ABSTRACTIONS, ANALYSIS TECHNIQUES, AND SYNTHESIS OF SCALABLE CONTROL STRATEGIES FOR ROBOT SWARMS

Spring Melody Berman

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania in Partial

Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2010

Vijay Kumar, Professor of Mechanical Engineering and Applied Mechanics
Supervisor of Dissertation

Jennifer Lukes, Associate Professor of
Mechanical Engineering and Applied Mechanics
Graduate Group Chairperson

Dissertation Committee:

Vijay Kumar, Professor of Mechanical Engineering and Applied Mechanics

Mark Yim, Associate Professor of Mechanical Engineering and Applied Mechanics

Ali Jadbabaie, Associate Professor of Electrical and Systems Engineering

Eric Klavins, Assistant Professor of Electrical Engineering

Abstractions, Analysis Techniques, and Synthesis of Scalable

Control Strategies for Robot Swarms

To my parents

Acknowledgements

I would first like to thank my advisor and mentor, Prof. Vijay Kumar, for his excellent guidance and relentless encouragement throughout my graduate studies. I would also like to thank Prof. Ali Jadbabaie, Prof. Mark Yim, and Prof. Eric Klavins for serving on my dissertation committee and providing valuable comments on my work. Special thanks go to my collaborators Prof. Ádám Halász for mentoring me during my first years as a graduate student and Prof. Stephen Pratt for teaching me about the fascinating world of ants. I would also like to express my gratitude to the other collaborators who made this work possible: Prof. M. Ani Hsieh, Quentin Lindsey, Loïc Matthey, and Mahmut Selman Sakar. Finally, I am enormously grateful for the friendships that I made in the GRASP Laboratory and the boundless love and support of my mother, father, and grandmother.

ABSTRACT

ABSTRACTIONS, ANALYSIS TECHNIQUES, AND SYNTHESIS OF

SCALABLE CONTROL STRATEGIES FOR ROBOT SWARMS

Spring Melody Berman

Vijay Kumar

Tasks that require parallelism, redundancy, and adaptation to dynamic, possibly hazardous environments can potentially be performed very efficiently and robustly by a swarm robotic system. Such a system would consist of hundreds or thousands of anonymous, resource-constrained robots that operate autonomously, with little to no direct human supervision. The massive parallelism of a swarm would allow it to perform effectively in the event of robot failures, and the simplicity of individual robots facilitates a low unit cost. Key challenges in the development of swarm robotic systems include the accurate prediction of swarm behavior and the design of robot controllers that can be proven to produce a desired macroscopic outcome. The controllers should be scalable, meaning that they ensure system operation regardless of the swarm size.

This thesis presents a comprehensive approach to modeling a swarm robotic system, analyzing its performance, and synthesizing scalable control policies that cause the populations of different swarm elements to evolve in a specified way that obeys time and efficiency constraints. The control policies are decentralized, computed a priori, implementable on robots with limited sensing and communication capabilities, and have theoretical guarantees on performance. To facilitate this framework of abstraction and top-down controller synthesis, the swarm is designed to emulate a system of chemically reacting molecules. The majority of this work considers well-mixed systems when there are interaction-dependent task transitions, with some modeling and analysis extensions to spatially inhomogeneous systems.

The methodology is applied to the design of a swarm task allocation approach that does not rely on inter-robot communication, a reconfigurable manufacturing system, and a cooperative transport strategy for groups of robots. The third application incorporates observations from a novel experimental study of the mechanics of cooperative retrieval in *Aphaenogaster cockerelli* ants. The correctness of the abstractions and the correspondence of the evolution of the controlled system to the target behavior are validated with computer simulations. The investigated applications form the building blocks for a versatile swarm system with integrated capabilities that have performance guarantees.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Robots have been developed over the past fifty years to perform a variety of tasks that are undesirable for humans and to increase accuracy and productivity over human capabilities. However, there are many tasks that cannot be performed efficiently by a single robot, no matter how complex it is. This is particularly true for tasks that require many subtasks to be executed in parallel, such as the retrieval of sensor data over a large region. Redundancy may be a required feature to ensure that the task can be completed despite system failures, which may be frequent in hazardous environments. The system may have to readily adapt to a dynamically changing environment, which can range from outer space to inside of the human body. Security may be another issue: the system's mission should be undecipherable from a single component.

To accomplish these kinds of tasks, we consider the use of a *swarm robotic system*. A swarm would consist of hundreds or thousands of autonomous, relatively expendable robots with limited sensing, communication, and computation capabilities. The massive parallelism of a swarm makes its operation robust to robot failures and environmental disturbances and allows robots to be dynamically added or removed. Since swarm members have constrained resources, they have a low unit cost, which allows the system to be affordable. Recent advances in embedded processor, sensor,

actuator, and communication technologies are paving the way for the development of such systems. Although they have not yet been implemented in practice outside of academic research laboratories, there has been a growing interest over the past decade in developing them for applications including environmental monitoring, exploration, military surveillance and reconnaissance, battlefield and disaster area communication, chemical source localization, mine detection, search-and-rescue, medical monitoring and treatment, micromanipulation, nanoscale manufacturing, and even pollination [2, 20]. The swarm paradigm is particularly essential at the micro- and nano-scale, in which robotic systems will have extremely limited on-board power, sensing, and computational capabilities.

Given these factors, it is becoming more and more urgent to develop a rigorous theoretical framework for modeling robotic swarms, analyzing their collective behavior, and synthesizing robot controllers with performance guarantees. The work presented here employs a framework of this type for a broad range of systems. This thesis can largely be considered as a handbook of sorts for designing a robotic swarm: most of the work is devoted to formulating classes of models for a swarm, describing applicable analysis techniques and controller synthesis methods for each type of model, and then demonstrating these components on appropriate swarm systems. Chapter 8 contains a departure from this approach, in that it illustrates an experimental study of a biological example of a swarm and its subsequent modeling based on observations. Section 1.1 defines the problem that we want to solve and the challenges inherent in developing our framework, Section 1.2 gives an outline of our approach, and Section 1.3 enumerates the specific contributions of the thesis.

1.1   Problem Statement

We address the problem of controlling *populations* of different elements of a swarm in a way that obeys time and efficiency constraints on system performance and is easily

adaptable to changes in target populations. The swarm elements are determined by the application under consideration and may include robots at each of a set of tasks, objects of distinct types, and various forms of robot-object complexes. The controllable components of the system are the motion of the robots and their decisions to switch between tasks and interact with other elements.

The enormous populations in the systems that we consider present certain challenges in developing an approach to solve this problem. In order to predict the swarm behavior, it is imperative to create models of the system that can be simulated in a reasonable amount of time. The models should be amenable to analysis techniques that can establish theoretical guarantees on system performance over a range of parameters. Since the complexity of most centralized algorithms for coordination is at least quadratic in the number of agents [112], it is impractical to use centralized control for a group of hundreds or thousands of robots. A decentralized control approach is needed in order for the robot control policies to be scalable with the population. Decentralized algorithms are implementable on robots with limited capabilities, since they require only local information without knowledge of the global state of the system [111]. It is also important to restrict communication among robots, since bandwidth becomes a limiting factor in communication as the population size increases.

1.2 Approach

To satisfy the criteria in Section 1.1, we formulate the problem as the manipulation of trajectories of element populations in an *abstraction* of the system whose dimensionality is independent of the swarm size, and the subsequent *top-down* synthesis of robot controllers from the abstraction parameters. This approach requires a rigorous correspondence between the physical swarm and the abstraction in order for the resulting controllers to produce the desired evolution of element populations.

3

Such a correspondence has been established between *chemical systems of reacting molecules* and abstractions that take the form of continuous-time Markov chains and differential equations [45, 51, 75]. The well-developed bodies of theory for these mathematical tools offer a variety of techniques for simulation, analysis, optimization, and control. In addition, the actions of molecules are readily implementable on the resource-constrained members of a swarm. For these reasons, we choose to design the robot swarm to emulate such a chemical system. In the majority of this work (Chapters 3-8), we develop systems that can be modeled as a Chemical Reaction Network (CRN), which requires spatial homogeneity of the swarm elements if they interact with each other. Using this paradigm, we formulate the modeling, analysis, and controller synthesis components of our theoretical framework, which are illustrated in Figure 1.1. In Chapter 9, we extend the modeling and analysis methodology to spatially inhomogeneous systems with interactions by defining the swarm to behave like a reaction-diffusion chemical system that is diffusion-limited rather than rate-limited.

1.2.1 Methodology

The physical swarm system is represented by the *micro-continuous model*, which may range from a simple point-mass model to a realistic 3D multi-robot model at the highest level of fidelity. The robots' transitions between tasks are designed to imitate chemical reactions: they transition *stochastically*, either spontaneously or upon encountering certain objects or other robots, at rates that are determined by constants that are analogous to *reaction rate constants*. The rate constants are designed to produce a target macroscopic outcome for the swarm. The robot motion controllers are defined to produce the execution of the desired tasks and to create the physical conditions that facilitate the abstraction. If there are task transitions that can be initiated by encounters, the second factor is achieved by driving the robots with a random walk to enforce "mixing," which is superimposed on a deterministic

4
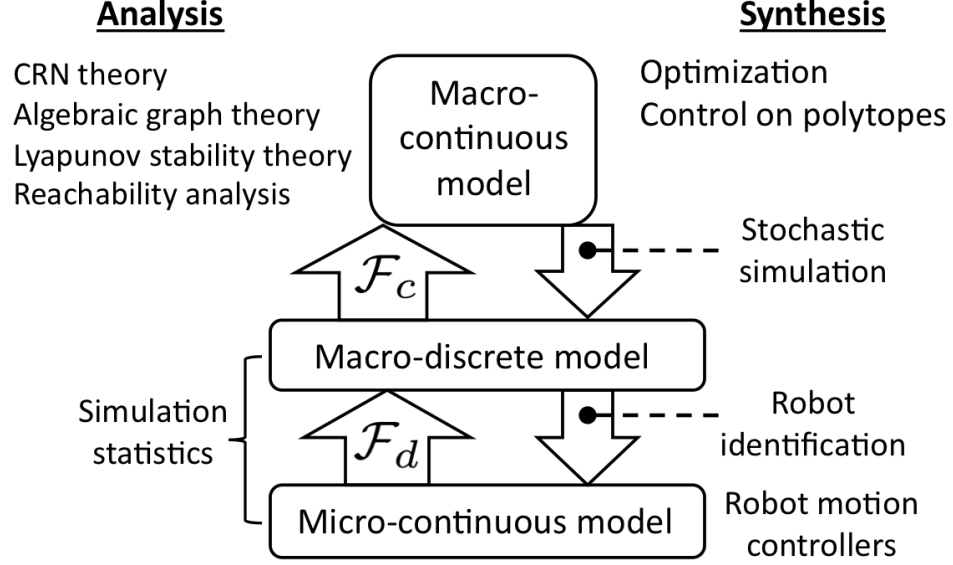
Figure 1.1: Levels of abstraction of a swarm robotic system with analysis and synthesis methodologies. The high-dimensional micro-continuous model is mapped to lower-dimensional representations, the macro-discrete and macro-continuous models, through the abstractions $\mathcal{F}_d$ and $\mathcal{F}_c$.

velocity in spatially inhomogeneous systems. For these systems, the deterministic and random components of the robot velocities are design parameters in addition to the rate constants. If all transitions are spontaneous, then there are no abstraction-motivated constraints on the motion controllers.

In practice, implementations of the micro-continuous model for N robots can be computationally expensive to simulate and intractable for analysis as N increases. The *macro-discrete model* is a lower-dimensional model of the system that can be simulated much more quickly while retaining some of the features of the micro-continuous model. This representation is a macroscopic model in the sense that it abstracts away the identities of the robots, and it considers a swarm as a collection of discrete agents whose integer populations evolve according to a Markov process.

The performance of the micro-continuous and macro-discrete models is analyzed by running simulations and examining the statistics of the resulting ensemble of trajectories or fields. In particular, realizations of the the macro-discrete model are

obtained with a stochastic simulation algorithm.

For very large populations, the system can be accurately abstracted to a *macro-continuous model*, which represents a swarm as a set of continuous populations governed by differential equations that are functions of the design parameters. Systems that can be modeled as CRN's are abstracted to ordinary differential equations (ODE's), while spatially inhomogeneous systems with interactions are abstracted to partial differential equations (PDE's). For PDE models, we employ a mesh-free Lagrangian computational method to obtain the numerical solution of populations over the domain. We developed the remainder of the methodology for ODE models.

The macro-continuous level analysis verifies whether the continuous model satisfies the system requirements. The model equilibria give the steady-state populations of the swarm elements, and they can be characterized using results from Chemical Reaction Network theory, algebraic graph theory, and Lyapunov stability theory. The model may be described as a *hybrid system* if it consists of discrete modes, corresponding to regions in the state space, that are each characterized by a different set of continuous dynamics. In this case, we can apply a *reachability analysis* technique to approximate the system behavior over time.

Using the macro-continuous model, the rate constants are designed to minimize system convergence time to a target distribution of element populations, possibly with a specification on efficiency. We formulate both convex optimization and stochastic optimization approaches to this problem. When the model is a hybrid system, control terms are added to the dynamics in order to guide the system trajectories through a specified sequence of modes, which in this work correspond to polytopes in the state space.

1.2.2 Implementation

We now discuss the design principles, control architecture, and robot capabilities that are needed to implement our top-down control strategy on a swarm robotic

system. The robots that we consider are unidentified, each identical in hardware and programmed with the same set of control algorithms. This enables the parallel execution of tasks, robustness to failures, and ease of adding and removing robots from the swarm. Using the asymmetric broadcast control (ABC) paradigm [112], a supervisory node computes the design parameters and broadcasts them to the swarm, along with information regarding the tasks they are to execute, the allowable transitions between tasks, and possibly the environment (such as a map). The supervisory node may obtain global state information about the swarm, such as its degree of convergence to the target element populations, and use this to update its computations, but it does not require knowledge of the robot population or individual robot actions.

To execute the task transitions and motion controllers, the robots must rely on local information that they obtain via sensing and/or communication. For an encounter-dependent transition, a robot must identify the type of another robot or object to determine whether it is a potential "reactant." This can be done using local wireless communication, in which the robots exploit communication links when they become available, or a camera with image processing software. The desired motion strategy may require localization and the ability to execute a random walk. To avoid collisions with other swarm elements and obstacles in the environment, a robot may use infrared proximity sensors, sonar sensors, or a laser rangefinder. Robots may attach to other elements with a gripper or a magnetic latching mechanism.

1.2.3 The Role of Bio-Inspiration

Swarms in nature are "proofs-of-concept" that complex macroscopic tasks can be achieved through the local interactions of numerous relatively simple, anonymous individuals. A prominent example is the self-organized behavior of social insects. Insect colonies achieve global objectives such as nest construction, colony emigration, foraging, hunting, and transport by switching between behaviors using local sensing

7

and physical contact. The outcomes are robust to perturbations in the environment and insect population and to noise in the information sensed by individual insects. Although it may seem to an observer that the members of a swarm are all following instructions from a leader, such as when birds exhibit incredibly coordinated flocking, in reality they are behaving in a decentralized manner. In a general sense then, natural swarms provide us with the principles for designing a robotic system that is massively parallel, robust to failure, flexible, and economical.

Our particular framework provides a concrete mechanism for incorporating concepts from biology: we can draw on CRN models of biological swarm behaviors that are useful for robotic systems to define the set of robot tasks and the structure of the reaction network that dictates the possible transitions. Such models are, most obviously, abundant in the literature on biochemical networks, for instance the *lac* operon in *E. coli* [157] and the mammalian circadian clock [41]. They are also employed in population biology; examples include the familiar Volterra-Lotka predator-prey model [134] and models of nest site selection in ants and honeybees [42]. We can select a model that describes a certain type of collective behavior, such as allocation among a set of sites, and adjust it using our controller synthesis methodology to produce a different macroscopic outcome that is more suitable for our application (see Chapter 6).

We may also be interested in the physical mechanisms of how a biological system accomplishes a task in order to define robot motion controllers for a novel decentralized approach to the task, as in the cooperative manipulation problem in Chapter 8. The controllers can be classified as bio-inspired rather than biomimetic because they are designed to emulate the strategy using the sensor information and actuation that are available to the robot, which is often not equipped to detect cues such as chemical signals that are vital to the biological agents.

1.3 Contributions

The main contribution of this thesis is the use of the theoretical framework described in Section 1.2 for designing the collective behavior of a robot swarm. This approach allows us to synthesize robot control policies that can be computed *a priori* and have provable guarantees on performance. In the analysis component of our methodology, we present a novel algorithm for the reachability analysis of hybrid systems with multi-affine continuous dynamics (see Section 4.3.2).

We use our modeling, analysis, and controller synthesis methodology to design systems for three types of applications. These applications can be combined to produce a versatile swarm system with the capabilities of distributing among a set of sites and subtasks, assembling products from raw materials and/or self-assembling, and transporting materials and products between sites.

In the first application, the robots are allocated among a collection of tasks, which are to be performed in parallel, that are located at different sites. In response to new levels of demand for robots at each task, the robots must quickly and efficiently reallocate themselves among the sites to occupy them according to a target population distribution. Since the tasks may be far apart from one another, communication between all robots may not be possible due to interference, obstruction, or power limitations, or it may be too risky, as in military applications. In light of these issues, we propose a strategy that does not use inter-robot communication.

In the second application, the robots must assemble target quantities of different types of products from a large supply of heterogeneous parts. In this case, the robots use local sensing and local communication to pick up parts and join them with parts carried by other robots. The advantage of this kind of manufacturing system over a traditional assembly line is its increased flexibility; the robots can respond quickly to assemble different product quantities when demand changes. We also model a part retrieval scenario in a spatially inhomogeneous assembly system.

The third application is a fully decentralized control strategy for robots in groups

of target sizes to cooperatively manipulate objects to specified locations while maneuvering past obstacles. Toward this end, we undertake an unprecedented experimental study of the mechanics of cooperative transport in groups of *Aphaenogaster cockerelli* ants and present a hybrid system model of the ant behavior based on observations.

1.4 Organization of this work

This work is organized as follows. Chapter 2 summarizes relevant literature in the areas of biological aggregations, swarm modeling, stochastic simulations, CRN theory, reachability analysis, controller synthesis, multi-robot task allocation, robotic assembly systems, and multi-robot transport systems. Chapter 3 describes the three swarm models shown in Figure 1.1. Chapters 4 and 5 present the analysis and controller synthesis techniques, respectively, that are applied to each model. Chapter 6 describes the application of the methodology to two task reallocation scenarios, a bio-inspired deployment strategy and a multi-site surveillance system. Chapter 7 discusses the swarm robotic assembly system application. Chapter 8 describes our experimental study and modeling of collective transport in ants. Chapter 9 extends the modeling and analysis approaches to spatially inhomogeneous swarms with encounter-dependent transitions. Finally, Chapter 10 suggests future directions for the development of our methodology and applications.

Chapter 2

Literature Review

This chapter outlines relevant literature on the modeling, analysis, and controller synthesis components of our methodology, as well as on the applications of multi-robot task allocation, swarm robotic assembly systems, and multi-robot collective transport.

2.1 Swarms in Biology

Diverse examples of emergent collective behavior can be witnessed in nature: the coordinated motion of bird flocks and fish schools, the process of nest site selection in ant and honeybee colonies, and the feat of mound construction in termite colonies, to name a few examples. I have contributed a catalog of references on collective behaviors in biology to the online repository at [1] and the online database [21].

2.2 Swarm Models

At the micro-continuous level, swarms have been simulated using agent-based models such as a biologically-inspired model of flocking behavior [127] and Brownian agent models of a wide variety of phenomena such as cells responding to stimuli,

trail formation in ants and pedestrians, urban aggregation, and voter opinion forma-
tion [136]. Our abstraction of the micro-continuous system to an accurate macro-
scopic model is similar to recent work on modeling robot swarms [4, 58, 97]. In the
first two references, it is assumed that robots and their stimuli are uniformly spatially
distributed. Identical robot controllers are defined with stochastic state transitions,
and they are averaged to obtain a set of difference equations [4] or ordinary differen-
tial equations [97]. In [58], the swarm is represented by a Fokker-Planck equation to
model spatial inhomogeneity. System performance is studied by running the mod-
els, which are validated through simulations and experiments, under many different
conditions. This constitutes a "bottom-up" controller synthesis procedure, in con-
trast to our "top-down" approach, which is less computationally expensive and gives
theoretical guarantees on performance.

2.3 Analysis Techniques

2.3.1 Stochastic Simulation Algorithms

A stochastic simulation algorithm is a procedure for computing numerical realizations
of molecular populations over time in a well-stirred chemical reaction network. [51]
provides a thorough survey of algorithms formulated by Gillespie and extensions
that were developed to increase the computational speed, some specifically designed
for stiff systems. A recent addition is the weighted stochastic simulation algorithm
(wSSA) [49, 87], which uses the technique of importance sampling to increase the
frequency of rare events of interest.

2.3.2 Chemical Reaction Network (CRN) Theory

The literature on CRN theory provides techniques for determining the existence,
multiplicity, and stability properties of equilibria of nonlinear dynamical systems of

certain classes. The results can be applied to networks with high-dimensional, complex dynamics and sometimes hold regardless of the values of the system parameters. The theory developed by Feinberg, Horn, and Jackson [37, 38] describes results that can be obtained for networks with a fairly general topology using a parameter called the *deficiency*; mass action kinetics are required to conclude the possibility of equilibria. The theory of monotone dynamical systems gives results for networks with a more restricted topology and *monotone* but otherwise arbitrary reaction kinetics [93]. The Deficiency theorems state results for equilibria and cyclic trajectories in the positive orthant of the real state space, while the theory on monotone systems extends results on stability and uniqueness to equilibria in the nonnegative orthant.

2.3.3 Reachability Analysis

In order to accurately approximate the global behavior of a set of hybrid system trajectories, or to verify that they do not enter an undesirable region, it is productive to consider *reachability analysis*, a well-known symbolic analysis technique [32, 62, 140]. A typical reachability problem is to determine whether a certain region of the state space can be reached by a system, starting from a given set of initial conditions. The reachability problem is decidable when the continuous dynamics are constant (timed and multirate automata), take values in a constant interval (rectangular automata) [62], or fall into certain classes of linear systems [90]. If the dynamics are not of these types, an overapproximation of the reachable set can be computed in one of two ways. One option is to pursue a discrete abstraction of the hybrid system via an *indirect method*. Alternatively, the reach set can be directly calculated the on the state space via a *direct method*.

In the indirect method, one generally partitions the continuous state space of the system into a finite number of sets and explores how states in one set may reach states in another set. Sets are usually convex regions of the state space; the exact representation of a set depends on a particular method. An example is the

multi-affine reachability algorithm developed in [7, 56, 81].

HyTech [62] and PHAVer [44] are tools for the verification of linear hybrid automata. This class of automata has piecewise constant bounds on the derivatives of the continuous state variables. HyTech and PHAVer overapproximate affine continuous dynamics by linear formulas over the derivatives. PHAVer also has the ability to partition reachable modes recursively along user-defined hyperplanes. MATISSE [52] and CheckMate [140] are reachability algorithms that use direct techniques for non-linear hybrid systems.

2.4   Controller Synthesis

Control paradigms for multi-robot systems can be classified as centralized or decentralized, and the robots may be individually identified or identical (either to all other robots in the group or to a subset of the group) and unidentified. As discussed earlier, the paradigm that we employ for a swarm system involves the decentralized control of many unidentified robots. Our top-down control synthesis approach is related in spirit to the work of [8], which presents a systematic approach to translate group behaviors, modeled as vector fields on a low-dimensional abstract manifold, into agent behaviors in a high-dimensional manifold derived from copies of an agent's state space. At the micro-continuous level, robot motion controllers for navigation through an environment with obstacles can be defined using navigation functions [128–130] and vector field design on a cell decomposition of the free space [30]. Vector field design may also be used to control the state of a hybrid macro-continuous model through a sequence of modes [7, 55]. Flocking control of multiple robots can be implemented in a distributed manner using artificial potential functions [95, 145], which may also be used for inter-robot collision avoidance alone.

14

2.5　Applications

2.5.1　Task Allocation for Robot Swarms

We apply our methodology to the design of a strategy for quickly and efficiently reallocating a swarm of robots among a set of tasks such that the steady-state populations at the tasks follow a target distribution. The tasks are to be performed in parallel, continuously, and independently of one another. This is an instance of the single-task robot, multi-robot task problem (ST-MR) [46], where the goal is to assign teams of robots to tasks in a way that maximizes the system's performance. This is known as the *coalition formation* problem when applied to software agents. Tractable approaches to this problem, which is NP-hard, rely on extensive agent cooperation that is not easily implemented in robot systems since communication can be costly and unreliable and resources are not transferrable [153]. The algorithm in [137] was adapted to the multi-robot domain in [152], but robots must compute all possible coalitions and agree on the best ones, and coalition sizes are limited. The ST-MR problem has recently been addressed with market-based techniques, although allocation strategies for robots have mostly considered the problem of assigning a single robot to each task [153]. Market-based approaches [35] require robots to execute complex bidding schemes based on perceived costs and utilities, and the computation and communication requirements often scale poorly as the number of robots and tasks increases.

These algorithms are not suitable for the systems that we consider, which have very large populations and do not employ inter-robot communication to allocate among the tasks. Instead, we adopt the decentralized paradigm of the approaches in [3, 84, 89], which are inspired by the self-organized behavior of social insects such as ants, and those in [4, 97]. In these approaches, robots switch between simple behaviors based on environmental stimuli and interactions with other robots. We note that the potential-based algorithm in [138] is also scalable, but it is designed

for tasks that are depleted and does not address the problem of allocating robots as quickly as possible.

2.5.2 Swarm Robotic Assembly Systems

We also apply our methodology to the design of a decentralized manufacturing system in which a swarm of mobile robots must quickly assemble target amounts of different products from any initial set of parts. In the taxonomy [54] of macroscopic self-assembly systems, our objective and approach are most similar to those of [80], which considers a set of modules that bind through random collisions and detach into different parts according to programmed probabilities [79]. As in our system, the interactions between elements are modeled as a CRN. Other assembly systems based on random collisions between parts are described in [67], which predicts the yield of complete assemblies from passive, vertically stirred modules, and in [6], which studies the dynamics of micro scale batch assembly using vibration; both of these systems are also modeled as CRN's. In [80], an optimization problem is formulated to compute the module detachment probabilities that maximize the equilibrium yield of one assembly type. The optimization, which is based on a Markov process model of the system, requires the enumeration of all reachable states, and is therefore suitable for a relatively small number of parts.

Our use of robots to transport and join passive parts according to decentralized rules is similar to the setups in [154], which derives rules for building a single desired structure out of blocks, and in [83], which presents an algorithm for assembling an object such as a truss-like structure out of different types of parts.

2.5.3 Multi-Robot Transport Systems

Many approaches to multi-robot manipulation of a load rely on leader/follower or centralized schemes and knowledge of the load geometry and possibly the contact force measurements [116]. Other approaches to multi-robot transport include towing

[39], caging [40], and ant-inspired box-pushing [85]. The first two works incorporate rigorously derived models and controllers with performance guarantees. [40] presents a decentralized approach in which sequentially composed vector fields are used to drive a group of robots to approach, surround, and push an object to a desired location, maintaining closure by orbiting around the object. [39] formulates a quasi-static model for the motion of a planar load that is towed by one, two, or three robots in the presence of dry friction and cable tension constraints. [85] implements a decentralized approach, inspired by collective transport in ants, to multi-robot box pushing in which robots switch between simple behaviors in response to locally sensed cues. The system behavior is investigated through multiple simulations and experiments; there are no theoretical guarantees on performance.

Chapter 3

Swarm Models

The material in this chapter was first presented in [11–13, 15, 57, 69, 109].

This chapter presents the three models of a swarm that we use in our analysis and controller synthesis methodology. Section 3.1 describes the *micro-continuous model*, which represents individual robots and other system elements in a physical environment, incorporating the geometry and dynamics of the robots and possibly modeling heterogeneity. The robots' transitions between tasks are modeled as a Chemical Reaction Network (CRN), and several types of transitions are specified in Section 3.1.1 and 3.1.2. The micro-continuous model is abstracted to the lower-dimensional *macro-discrete model*, outlined in Section 3.2, which models the time evolution of *integer* populations of robots at each task and other elements of different types according to the Chemical Master Equation [45, 48, 75]. The macro-discrete model can be abstracted to the *macro-continuous model*, described in Section 3.3, a set of coupled ordinary differential equations (ODE's) that model the time evolution of the *continuous* populations of robots at each task and other types of elements. This model is known as the *rate equation* in literature on CRN's. Sections 3.3.1, 3.3.2, and 3.3.3 present different classes of macro-continuous models that can be used to represent robotic systems with the transitions given in Sections 3.1.1 and 3.1.2.

3.1 Micro-Continuous Model

Consider a population of N robots moving in the continuous state space $Y_r \subset \mathbb{R}^d$, $d \in \{2, 3\}$. At any given time, a robot's actions are determined by one of a set Ω_r of P controllers. A controller $\omega \in \Omega_r$ causes a robot to fulfill a distinct *task*, or possibly a subdivision of activity within a task. The *micro-continuous model* is comprised of N hybrid automata, $H_r = \{Y_r, \Omega_r\}$, each describing a robot in terms of both its continuous and discrete dynamics. Borrowing CRN terminology, we refer to the S types of elements in the system as *species* and call a combination of species that appears before or after a reaction arrow a *complex*. A particular species, symbolized by X_i, represents a robot that is performing task i or an object of type i with which the robots interact.

The micro-continuous model can be implemented in various ways, depending on the desired level of realism. If it is important to generate the dynamics associated with friction, collisions, and other inter-robot or robot-environment interactions, then the model may be implemented in a 3D multi-robot simulator such as Gazebo [82] or Webots [113], both of which use the Open Dynamics Engine to simulate physics, or more accurately with the simulation tool daVinci Code [10] or the multi-body simulation algorithm in [77]. If such interactions do not play a significant role in the system performance, then each robot can be described by a point agent that is governed by a single-integrator or double-integrator model, depending on whether the motion control input is applied to the robot's velocity or acceleration. The applications presented in this thesis use both types of implementations.

In order to be able to abstract the system to the macro-discrete model, the robots' task transitions must be be executed in a way that conforms to the *fundamental hypothesis* of the stochastic formulation of chemical kinetics [47, 50]. This hypothesis states that a reaction that converts complex i into complex j is characterized by a

stochastic reaction constant c_{ij}, which is defined such that

$$c_{ij}\delta t \;\; = \;\; \text{the average probability that a particular combination of elements of the}$$

species types in complex i will transition to the species types in complex

j in the next infinitesimal time interval δt. (3.1)

The corresponding reaction rate constant, k_{ij}, is proportional to c_{ij} (see the following sections for its definition). The reaction *propensity*, a_{ij}, is defined such that $a_{ij}\delta t$ is the probability that the reaction that converts complex i into complex j will occur in the next δt. It is the product of c_{ij} with h_{ij}, the current number of distinct combinations of elements that can undergo the reaction.

In the following sections, we discuss how to ensure the validity of (3.1) in systems that contain two types of "reactions." In the first type, described in Section 3.1.1, robots switch between tasks upon encountering nearby robots or other elements of the system. In the second type, described in Section 3.1.2, robots switch spontaneously between tasks.

3.1.1 Interaction-Dependent Task Switching

In our systems, as in CRN's in general, it is rare that three or more elements encounter each other simultaneously. Hence, in this section we only consider "bimolecular" reactions, in which the reactants consist of two species, m and n. We specify that the robots are moving in a bounded domain with volume V (area A in two dimensions).

For premise (3.1) to be true, the system must be *well-mixed*. Gillespie [47, 50] defines a well-mixed chemical system as one in which molecules are uniformly randomly distributed throughout a container with fixed volume V. He notes that this property can be maintained by direct stirring or by a very high rate of nonreactive molecular collisions compared to that of reactive collisions. In our robotic system, one option is to design the robot controllers to emulate a reaction-diffusion system,

in which diffusion is implemented by driving the robots with Brownian motion (see Chapter 9 for a discussion of the class of models to which reaction-diffusion systems belong). When diffusive motion is the mechanism that drives mixing, the well-mixed property can be verified by checking that the diffusion rate is much higher than the reaction rates. This was done for testbeds of stochastically interacting robotic components in [24] and [117] using measurements of the diffusion coefficients and rate constants. Alternatively, the robots can be commanded to execute a random walk with fixed speed and we can verify that they are uniformly randomly distributed over the domain.

For bimolecular reactions, the probability $c_{ij}\delta t$ is the product of the probability that a random pair of reactant elements of complex i will encounter each other in the next time interval δt, denoted by $c_{ij}^e \delta t$, and the probability that these reactants will form complex j given that they are in close proximity, c_{ij}^p [48]. The probabilities c_{ij}^p are the parameters that we can design to achieve a macroscopic system objective. In molecular systems, these probabilities are functions of activation energy [50]. In order to satisfy the fundamental hypothesis (3.1), c_{ij}^e and c_{ij}^p must both be independent of δt [48]. We can design c_{ij}^p to be time-independent, and below we discuss cases in which c_{ij}^e can be verified to be a constant.

In chemical systems, the well-mixed condition implies that the probability that the center of any molecule lies in the "collision volume" δV_{coll} swept out by any other molecule in the infinitesimal time interval δt is given by $\delta V_{coll}/V$ [47, 48, 50]. Using this premise, Gillespie geometrically derives the average probability that a random pair of molecules 1 and 2, which we will suppose here are species in a complex i, will collide in the next time interval δt as

$$c_{ij}^e \delta t = E(\delta V_{coll}/V) = \frac{\pi}{V}\left(\frac{d_1 + d_2}{2}\right)^2 E(v_{12})\delta t, \qquad (3.2)$$

where $E(\cdot)$ denotes expected value, d_1 and d_2 are the diameters of the molecules (assumed to be spheres), and v_{12} is the speed of molecule 1 relative to molecule 2. If the system is in thermal equilibrium at absolute temperature T, then the molecules

have Maxwell-Boltzmann velocity distributions, which results in the scalar value

$$E(v_{12}) = (8k_BT/\pi m_{12})^{1/2}, \tag{3.3}$$

where k_B is Boltzmann's constant and $m_{12} = m_1m_2/(m_1 + m_2)$, with m_i denoting the mass of molecule i [48, 50]. Gillespie and van Kampen [75] (p. 171) note that the occurrence of a higher frequency of nonreactive collisions than reactive collisions maintains this type of velocity distribution. From the kinetic theory of gases, it is known that diffusing particles have Maxwellian velocity distributions, so robots with Brownian motion should satisfy Equation (3.3). This can be experimentally verified for a given system, as was done in [19] for a testbed of programmable robotic parts.

If the robots move according to a random walk with constant speed, and in addition all elements of the system are uniformly distributed throughout the domain, then c_{ij}^e can be calculated as [31]

$$c_{ij}^e = vw/V, \tag{3.4}$$

where v is the robot speed and w is the robot's detection width (assumed to be constant), which is the width of the area that the robot's sensing or communication range sweeps out while it is moving.

We now define the possible types of bimolecular reactions in our systems. Let N_i be the integer population of elements of species i. For bimolecular reactions, $c_{ij} = k_{ij}/V$ and $h_{ij} = N_mN_n$ if $m \neq n$, and $c_{ij} = 2k_{ij}/V$ and $h_{ij} = N_m(N_m - 1)/2$ if $m = n$ [50].

Let X_m denote either a robot performing task m or an object of type m, and let X_n represent the same. The interaction between these two elements upon encountering each other can induce one or both to adopt a new task (in the case of a robot) or identity (in the case of an object), labeled by o and p. This transition is described by the reaction

$$X_m + X_n \xrightarrow{k_{ij}} X_o + X_p . \tag{3.5}$$

The position of k_{ij} above the reaction arrow indicates that it is a reaction rate constant. Here, $X_m + X_n$ is complex i and $X_o + X_p$ is complex j. When $n = p$, the reaction describes a conversion of X_m to X_o that is "catalyzed" by an interaction with X_n.

Now set X_m to be a robot performing task m and X_n to be an object of type n, such as a component of a product in an assembly system. If the robot encounters the object, it can decide to connect to it. Let X_p, which will be labeled as complex j, symbolize the robot once it has bonded to the object and possibly started a new activity. The reaction that describes this connection is

$$ X_m + X_n \xrightarrow{k_{ij}} X_p \ . \tag{3.6}$$

Alternatively, this reaction may represent the bonding of objects of type m and n to form an object of type p.

In the implementation of these reactions, a robot in complex i that encounters another robot or other element with which it can "react" to form complex j computes a uniformly distributed random number $u \in [0, 1]$ and follows through with the transition if $u < c_{ij}^p$.

3.1.2   Spontaneous Task Switching

This section presents reactions that are classified as "unimolecular" since their reactants consist of one species. The fundamental hypothesis (3.1) is valid if a robot at task i undergoes the transition associated with the reaction at probability per unit time c_{ij}. For these types of reactions, $c_{ij} = k_{ij}$ and $h_{ij} = N_i$ [50].

Let X_i denote a robot that is performing task i, and suppose that the robot switches to task j at probability per unit time k_{ij}. Using CRN notation, this reaction is written as

$$ X_i \xrightarrow{k_{ij}} X_j \ . \tag{3.7}$$

For this type of reaction, k_{ij} will also be called a *transition rate*.

Now let X_i represent a robot that is executing a task while connected to an object of type n, labeled by X_n. The robot has a certain probability per unit time, k_{ij}, of disconnecting from the object. Let X_m symbolize the robot once it has disconnected and possibly started a new activity (even if it is the same activity, we still consider the detachment to initiate a new "task"), and label $X_m + X_n$ as complex j. Then the reaction that describes this disconnection is

$$X_i \quad \xrightarrow{k_{ij}} \quad X_m + X_n \; . \tag{3.8}$$

Alternatively, this reaction may represent the disassembly of an object of type i into objects of type m and n.

In the implementation of these reactions, a robot that is doing task i computes a uniformly distributed random number $u \in [0, 1]$ at each (very small) simulation timestep dt and executes the transition governed by k_{ij} (switching to task j or disconnecting from an object) if $u < k_{ij}dt$. The number of transitions governed by k_{ij} that occur in a time interval Δt has a Poisson distribution with parameter $k_{ij}\Delta t$.

3.2 Macro-Discrete Model

By applying the laws of probability to the fundamental hypothesis (3.1), it is possible to derive a time-evolution equation for $P(\mathbf{n}, t|\mathbf{n_0}, t_0) \equiv Prob\{\mathbf{N}(t) = \mathbf{n}$ given $\mathbf{N}(t_0) = \mathbf{n_0}\}$, where $\mathbf{N}(t) \in \mathbb{R}^S$ is the vector of integer species populations N_i at time t [48]. This is the *Chemical Master Equation* (CME), and it describes a continuous-time Markov process whose states are all the possible species population vectors $\mathbf{N}$. We refer to this as the *macro-discrete model* of the system.

The macro-discrete model can capture phenomena that occur in the physical system but do not appear in the macro-continuous model. This is because the stochastic formulation of a system has a more legitimate physical basis than the deterministic formulation, which does not account for correlations and fluctuations [50]. Examples of such phenomena include state fluctuations in relatively small

populations, potentially leading to stochastic transitions between equilibria of multistable systems [73, 133].

The Chemical Master Equation itself is not a useful tool for analysis; it cannot be solved analytically for more than a few simple cases, and it is often difficult to solve numerically due to the number of its independent variables (S species populations and the continuous variable t) [47, 51]. Instead of attempting to compute the probability density function of $\mathbf{N}(t)$, we can characterize statistical properties of the macro-discrete model by generating numerical realizations of $\mathbf{N}(t)$ with a stochastic simulation algorithm that is logically equivalent to the Chemical Master Equation, *i.e.*, it is also derived from hypothesis (3.1). One such algorithm is described in Section 4.2.

3.3 Macro-Continuous Models

For the types of systems that we consider, the time evolution of the expected values of the species concentrations, $N_i(t)/V$, is described deterministically by a set of ODE's which we call the *macro-continuous model*. When the system contains only unimolecular reactions, this can be shown by substituting the CME into the equation for $\partial E(\mathbf{N}(t))/\partial t$ and using the linearity of expectation [155] (Section 6.7). When there are bimolecular reactions present, it can be demonstrated by using van Kampen's system size expansion of the CME [75] (Ch. 10) if the set of ODE's has a unique, globally stable equilibrium [75] (p 355). First, the concentration of species i is defined as the sum of a deterministic value, x_i, and a fluctuating component that is scaled in inverse proportion to $\Omega^{1/2}$, where Ω is a system size parameter that in many cases is the system volume V. The rationale behind this is that for constant average concentrations, relative fluctuations will tend to decrease with $V^{-1/2}$ [36]. The CME is Taylor-expanded near trajectories of x_i, $i = 1, ..., S$, in powers of $V^{-1/2}$. The terms in the expansion of order $V^{1/2}$ give an ODE model governing the time

25

evolution of x_i, and the terms of order V^0 yield a linear Fokker-Planck equation (the "linear noise approximation") that can be used to characterize the moments of the fluctuations around the ODE solution.

In the thermodynamic limit, defined as the limit in which N_i, $i = 1, ..., S$, and V all approach infinity in such a way that the concentrations N_i/V approach finite values [51], the concentration fluctuations become negligibly small and the time evolution of the system is accurately described by the ODE model. We denote the vector of continuous species concentrations $x_i(t)$ by $\mathbf{x}(t) \in \mathbb{R}^S$. The continuous state space of these concentrations is labeled by $Y_p \subset \mathbb{R}^S$; the dimensionality of this space may be reduced by using conservation laws of the system to eliminate variables. Y_p may be divided into a set Ω_p of Q regions, called *population modes*, that are each associated with different continuous dynamics. Then, the system can be described by a hybrid automaton $H_p = \{Y_p, \Omega_p\}$.

The macro-continuous model is a function of the rate constants k_{ij} as well as the concentrations x_i. Each species i is assigned a target concentration, x_i^d, and the vector of these values, $\mathbf{x^d}$, is called the *target species distribution*. As described in Section 5.2, the macro-continuous model is used to design the rate constants k_{ij} that will cause the system to quickly converge to the target distribution, which is the unique, stable equilibrium of the model. We assume that a central controller determines $\mathbf{x^d}$, computes the k_{ij}, and broadcasts these rate constants to the robots.

In the most general case that we consider, the system contains both unimolecular and bimolecular reactions. The corresponding macro-continuous model is a *multi-affine* function of the species concentrations, defined by

$$\dot{\mathbf{x}} = \sum_{j=0}^{2^S-1} \mathbf{a}_j \, x_1^{i_1(j)} x_2^{i_2(j)} ... x_S^{i_S(j)} \; ; \quad \mathbf{a}_j \in \mathbb{R}^S \, , \tag{3.9}$$

where the concatenation

$$\Theta_j \equiv i_1(j) i_2(j) ... i_S(j), \quad \{i_1(j), ..., i_S(j)\} \in \{0, 1\}^S \tag{3.10}$$

is a binary representation of the integer j. It should be noted that the offset vector $\mathbf{a}_0$ is $\mathbf{0}$ in this model due to the structure of the reactions; however, nonzero components of $\mathbf{a}_0$ are introduced when conservation laws are used to eliminate state variables (see Section 6.1.2 for an example). If there are no bimolecular reactions in the network, then the macro-continuous model takes the form of an *affine* system,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{a} , \tag{3.11}$$

where $\mathbf{A} \in \mathbb{R}^{S \times S}$ and $\mathbf{a} \in \mathbb{R}^S$.

The multi-affine model is formulated as a matrix equation in Section 3.3.1, and Section 3.3.2 presents the reduction of the model to a linear form for the case of a "task reallocation" system, which has reactions of type (3.7) only. Section 3.3.2 also illustrates how to account for realistic distributions of task transition times within the linear ODE framework and describes models with transition rates that vary with the difference between the robot population at a task and a threshold value.

3.3.1 Multi-Affine Model

The set of reactions in a system can be modeled as a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of vertices, $\mathcal{V} = \{1, \ldots, C\}$, signifies the complexes, and the set of P directed edges, $\mathcal{E}$, represents the reaction pathways between the complexes. Complexes i and j are *adjacent*, denoted by $i \sim j$, if there is a reaction pathway in which complex i transforms into complex j. We denote this relation by the ordered pair $(i, j) \in \mathcal{V} \times \mathcal{V}$, with the set $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid i \sim j \}$. Each pathway (i, j) is associated with a reaction rate constant k_{ij}. Let R denote the number of *unordered* complex pairs (i, j) that are connected by at least one reaction pathway; each of these pairs, along with their corresponding pathway(s), is referred to as a *reaction* of the system.

Let $\mathbf{y}(\mathbf{x}(t)) \in \mathbb{R}^C$ define a vector in which entry $y_i(t)$ is the product of concentrations of the species in complex i at time t. If complex i consists of one species, X_j, then $y_i(t) = x_j(t)$, and if it consists of two species, $X_j + X_k$, then $y_i(t) = x_j(t)x_k(t)$.

The *reaction rate* of a reaction involving complexes i and j is defined as the difference between the *forward flux* $k_{ij}y_i(t)$ and the *reverse flux* $k_{ji}y_j(t)$, which is nonzero if $(j, i) \in \mathcal{E}$. Letting $\nu_i(t)$ denote the reaction rate of reaction i, the vector of reaction rates is $\nu(\mathbf{x}(t)) \in \mathbb{R}^R$.

We define a matrix $\mathbf{M} \in \mathbb{R}^{S \times C}$ in which each entry $\mathbf{M}_{ji}$, $j = 1, ..., S$, of column $\mathbf{m}_i$ is the coefficient of species type j in complex i (0 if absent). Finally, let $\mathbf{K} \in \mathbb{R}^{C \times C}$ be a matrix with the properties

$$\mathbf{K}^T \mathbf{1} = \mathbf{0} , \tag{3.12}$$

$$\mathbf{K}_{ij} \leq 0 \quad \forall (i, j) \in \mathcal{E} . \tag{3.13}$$

These two properties result in the following matrix structure:

$$\mathbf{K}_{ij} = \begin{cases} -k_{ji} & \text{if} \quad i \neq j , \ (j, i) \in \mathcal{E} , \\ 0 & \text{if} \quad i \neq j , \ (j, i) \notin \mathcal{E} , \\ \sum_{(i,l) \in \mathcal{E}} k_{il} & \text{if} \quad i = j . \end{cases} \tag{3.14}$$

Then the ODE abstraction of the system can be written in the following form [27]:

$$\dot{\mathbf{x}} = -\mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x}) . \tag{3.15}$$

Model (3.15) can equivalently be written in terms of $\nu(\mathbf{x})$ and the *stoichiometric matrix* $\mathbf{S} \in \mathbb{R}^{S \times R}$, for which each entry $\mathbf{S}_{ij}$ is the stoichiometric coefficient of species i in reaction j [61]:

$$\dot{\mathbf{x}} = \mathbf{S}\nu(\mathbf{x}) . \tag{3.16}$$

The ODE model is subject to $S - rank(\mathbf{S})$ linearly independent conservation constraints on the species concentrations, each of the form

$$\mathbf{c}^T \mathbf{x} = c, \quad \mathbf{c} \in \mathbb{R}^S , \ c \in \mathbb{R} , \tag{3.17}$$

where $\mathbf{c}$ is in the null space of $\mathbf{S}^T$ [61].

28

3.3.2 Linear Models

We consider the case in which the CRN representation of the system consists only of reactions of type (3.7). This models a scenario in which robots reallocate themselves among a set of tasks that are to be executed in parallel, continuously, and independently of one another. For instance, each task could be an activity at a physical site such as building surveillance, environmental monitoring, construction, or a search-and-rescue operation. To facilitate robot redistribution, we specify that robots may switch from any task to any other task, either directly or through a sequence of intermediate tasks; thus, no task acts as a source or a sink. Each quantity $x_i(t)$ now represents the continuous population of robots that is performing task i at time t. We specify $x_i(t)$ as a *population fraction*, the continuous population of species i divided by the swarm population N, which is practical for scaling as well as for applications in which losses of robots are common.

The robots must be capable of executing the tasks and transitions. For instance, if the tasks are situated at different sites, the robots must be able to localize themselves in their environment and navigate safely between sites. It is assumed that all robots have complete knowledge of the tasks to perform and the allowable transitions between them; this information can be preprogrammed and updated via a broadcast.

The graph $\mathcal{G}$ now models the precedence constraints between tasks; $\mathcal{V}$ denotes the collection of tasks, and $\mathcal{E}$ represents possible transitions between tasks. The adjacency relation $i \sim j$ signifies that a robot that is working on task i can switch directly to task j. For example, if each task i is an activity at a physical site i, then $\mathcal{G}$ models the site interconnection topology: $\mathcal{V}$ is the set of S sites and each edge (i, j) represents a one-way route that robots can travel from i to j. If there are P possible routes from i to j, $m = 1, \ldots, P$, each with transition rate $k_{ij,m}$, then they are represented by distinct edges $(i, j)_m$, and $k_{ij} = \sum_{m=1}^{P} k_{ij,m}$.

A *directed path* from task i to task j is a sequence of vertices, $\{v_0, v_1, ..., v_p\} \in \mathcal{V}$, such that $v_0 = i$, $v_p = j$, and $(v_{k-1}, v_k) \in \mathcal{E}$, $k = 1, \ldots, p$. Since we specify that the

robots can switch from any task to any other task, a directed path exists between any pair of distinct vertices in $\mathcal{G}$. This implies that $\mathcal{G}$ is *strongly connected*. We also consider the special case in which robots can switch from any task directly to any other task without first working on a set of intermediate tasks. In this case, each vertex in $\mathcal{G}$ is adjacent to every other vertex, so the graph is *fully connected*. For a strongly connected, but not necessarily fully connected, graph with bidirectional edges, in which $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, we explore the advantage of having a *reversible* Markov process, which is defined by the *detailed balance equations*:

$$k_{ij} x_i^d = k_{ji} x_j^d \quad \forall (i, j) \in \mathcal{E} . \tag{3.18}$$

A. Baseline Linear Model

Since each complex in the system consists of a single species, we have the relations $C = S$, $\mathbf{y}(\mathbf{x}(t)) = \mathbf{x}(t)$, and $\mathbf{M} = \mathbf{I}$. Thus, model (3.15) reduces to the linear model

$$\dot{\mathbf{x}} = -\mathbf{K}\mathbf{x} . \tag{3.19}$$

For this model, $\mathbf{S}$ is the incidence matrix of the graph $\mathcal{G}$. Since we specify that $\mathcal{G}$ is strongly connected, and thus has one connected component, by Theorem 8.3.1 of [53] $rank(\mathbf{S}) = S - 1$. Therefore, the null space of $\mathbf{S}^T$ is one-dimensional, yielding one conservation constraint. By Lemma 8.3.2 of [53], $\mathbf{S}\mathbf{S}^T$ is the Laplacian $\mathbf{L}$ of $\mathcal{G}$, for which $\mathbf{L}\mathbf{1} = \mathbf{0}$. Since this implies that $\mathbf{1}^T \mathbf{S}\mathbf{S}^T \mathbf{1} = ||\mathbf{S}^T \mathbf{1}||_2 = 0$, it follows that $\mathbf{S}^T \mathbf{1} = \mathbf{0}$. Hence, conservation constraint (3.17) for system (3.19) is

$$\mathbf{1}^T \mathbf{x} = 1 , \tag{3.20}$$

where $c = 1$ because the x_i are defined as population fractions. This equation simply states that the number of robots is conserved, which is an intuitive result since the system is closed; there is no inflow or outflow of robots.

Model (3.19) subject to (3.20) will be referred to as the *baseline linear model*, since it is the simplest expression of the task-allocation system. We can elaborate

30

on this point by inspecting the ODE for each component of $\dot{\mathbf{x}}$,

$$\dot{x}_i(t) = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}x_i(t) \ . \tag{3.21}$$

The flux from task i to j at time t is $k_{ij}x_i(t)$, which can be interpreted physically as the fraction of robots per unit time that are leaving i to switch to j. Hence, Equation (3.21) quantifies the rate of change of population fraction $x_i(t)$ as the difference between the total influx and total outflux of robots at task i. The model captures this effect in a simple way by representing robots as switching instantaneously from one task to another, ignoring the time that robots take to effect transitions. Because the k_{ij} are constant, robots still switch between tasks at equilibrium, when the net flux through each task is zero. This contributes to system robustness since the population at each task, which may be depleted by breakdowns, is constantly replenished. The persistent switching may also serve a useful function, such as patrolling or sampling between sites.

B. Linear Chain Model

As mentioned previously, model (3.19) does not account for the fact that in reality, the influx of robots to task j from task i is delayed by the time taken to switch between the tasks, τ_{ij}. If we assume a constant transition time τ_{ij} for each edge (i, j), this effect can be included by rewriting Equation (3.21) as a delay differential equation (DDE):

$$\dot{x}_i(t) = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t - \tau_{ji}) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}x_i(t) \ . \tag{3.22}$$

Due to the finite τ_{ij}, there will be robots in the process of switching between tasks; thus, $\sum_{i=1}^{S} x_i(t) < 1$ for $t > 0$. Let $y_{ij}(t)$ be the population fraction of robots that are in transition from task i to j at time t. Then the conservation equation (3.20) becomes:

$$\sum_{i=1}^{S} x_i(t) + \sum_{i=1}^{S} \sum_{\forall j|(i,j)\in\mathcal{E}} y_{ij}(t) = 1 \ . \tag{3.23}$$

In practice, robots will complete a transition in different amounts of time, so model (3.22) can be made more realistic by defining the τ_{ij} as random variables, T_{ij}. In the case where robots effect transitions by traveling between sites, variations in τ_{ij} can arise from changes in navigation patterns caused by collision avoidance, congestion, and localization errors. For this case, we can estimate a reasonable form for the probability density of the T_{ij} from an analogous scenario in which vehicles deliver items along roads to different sites. Vehicle inter-site travel times have been modeled as following an Erlang distribution to capture the fact that the times have positive, minimum possible values and a small probability of being large due to accidents, breakdowns, and low energy, as well as the tendency of their distributions to be skewed toward larger values [131]. We assume that each T_{ij} follows this distribution with parameters $\omega_{ij} \in \mathbb{Z}^+$ and $\theta_{ij} \in \mathbb{R}^+$:

$$g(t; \omega_{ij}, \theta_{ij}) = \frac{\theta_{ij}^{\omega_{ij}} t^{\omega_{ij}-1}}{(\omega_{ij}-1)!} e^{-\theta_{ij}t} \ . \tag{3.24}$$

In practice, the parameters are estimated by fitting empirical transition time data to density (3.24).

Under this assumption, the DDE model (3.22) can be transformed into an equivalent ODE model of the form (3.21), which allows us to design the k_{ij} using the methods we develop for this type of model. We use the fact that T_{ij} has the same distribution as the sum of ω_{ij} independent random variables, $T_1, ..., T_{\omega_{ij}}$, with a common distribution $f(t; \theta_{ij}) = \theta_{ij} e^{-\theta_{ij}t}$ [59]. Each of the variables represents a portion of the transition time between tasks i and j. To model these portions of the transition, we define a directed path composed of a sequence of *virtual tasks*, $u = 1, ..., \omega_{ij}$, between the real tasks i and j. Assume that robots transition instantaneously from virtual task u to $u + 1$, which is task j when $u = \omega_{ij}$, at a constant probability per unit time, θ_{ij}. It follows that $f(t; \theta_{ij})$ is the distribution of the time that a robot spends doing virtual task u, and therefore, we can define T_u, $u \in \{1, ..., \omega_{ij}\}$, as this task execution time. The expected value of T_u is $E(T_u) = \theta_{ij}^{-1}$. Using the property $E(T_{ij}) = \sum_{u=1}^{\omega_{ij}} E(T_u)$, we see that $E(T_{ij}) = \omega_{ij}/\theta_{ij}$. The variance of T_{ij} is
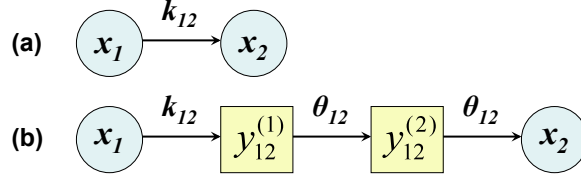
Figure 3.1: A labeled edge $(i, j) = (1, 2)$ that consists of (a) the real tasks, corresponding to model (3.21), and (b) both real and virtual tasks (for $\omega_{12} = 2$), corresponding to model (3.25).

$Var(T_{ij}) = E(T_{ij})^2/\omega_{ij}$.

We denote the population fraction that is doing virtual task u along edge (i, j) by $y_{ij}^{(u)}$. Then $\sum_{u=1}^{\omega_{ij}} y_{ij}^{(u)}$ represents y_{ij}, the fraction of robots in transition from task i to task j. Figure 3.1 illustrates how an edge from model (3.21) is expanded with two virtual states $y_{ij}^{(u)}$. As in the baseline model, the dynamics of the population fractions at all real and virtual tasks in the expanded system can be written as a set of linear ODE's:

$$
\begin{aligned}
\dot{x}_i(t) &= \sum_{j|(j,i)\in\mathcal{E}} \theta_{ji} y_{ji}^{(\omega_{ji})}(t) - \sum_{j|(i,j)\in\mathcal{E}} k_{ij} x_i(t) , \\
\dot{y}_{ij}^{(1)}(t) &= k_{ij} x_i(t) - \theta_{ij} y_{ij}^{(1)}(t) , \\
\dot{y}_{ij}^{(m)}(t) &= \theta_{ij} \left(y_{ij}^{(m-1)}(t) - y_{ij}^{(m)}(t) \right) , \\
&\qquad m = 2, ..., \omega_{ij} ,
\end{aligned}
\tag{3.25}
$$

where $i = 1, ..., S$ and $(i, j) \in \mathcal{E}$.

Let $\mathbf{y}$ be the vector of $y_{ij}^{(u)}$, $u = 1, ..., \omega_{ij}$, $(i, j) \in \mathcal{E}$. The system state vector is then $\mathbf{z} = [\mathbf{x} \, \mathbf{y}]^T$. We interpret each component of $\mathbf{z}$ as the population fraction at task $i \in \{1, ..., S'\}$, where S' is the sum of all real and virtual tasks. The interconnection topology of these tasks can be modeled as a directed graph, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \{1, ..., S'\}$ and $\mathcal{E}' = \{(i, j) \in \mathcal{V}' \times \mathcal{V}' \mid i \sim j \}$. Since $\mathcal{G}$ is strongly connected, so is $\mathcal{G}'$. Then the ODE model (3.25) can be written in the form of model (3.19),

$$
\dot{\mathbf{z}} = -\hat{\mathbf{K}}\mathbf{z} ,
\tag{3.26}
$$

where $\hat{\mathbf{K}} \in \mathbb{R}^{S' \times S'}$ has structure (3.14) with entries $\hat{k}_{ij}$ (in place of k_{ij}) defined by the corresponding coefficients in model (3.25). The conservation equation (3.23) can be written as

$$\mathbf{1}^T \mathbf{z} = 1 \ . \tag{3.27}$$

System (3.26) subject to (3.27) will be referred to as the *linear chain model*, since it incorporates a chain of virtual tasks between each pair of real tasks.

Remark: The modeling approach in this section can still be applied when the distribution of T_{ij} is complicated (e.g., multimodal) by approximating it as a combination of Erlang distributions; this is a topic for future work.

3.3.3 Quorum-Based Models

In the linear models of task reallocation, the robots switch between tasks independently of one another, which is an appropriate strategy if it is undesirable or unfeasible to implement decisions based on robot interactions. It is possible to improve system performance if the transition rates k_{ij} incorporate information about the current populations at the tasks. When the distribution of robots among the tasks is differs greatly from the target distribution $\mathbf{x^d}$, it makes sense for the robots to transition between tasks as quickly as possible to enable fast convergence to $\mathbf{x^d}$. Once the system attains $\mathbf{x^d}$, the robots can conserve the energy that they expend during transitions by switching between tasks at lower rates. (Section 5.2.1 elaborates on these points.)

We implement this system behavior through the decentralized mechanism of *quorum sensing*. Each task i is associated with a *quorum* q_i, a threshold robot population that is specified as a fraction of x_i^d. We assume that every robot knows the q_i as well as the x_i^d and can estimate the population fraction x_i through local sensing. The transition rate from task i to an adjacent task j is defined to increase from k_{ij}, the rate in the baseline linear model, to a maximum value when $x_i/x_i^d \geq q_i$. We then

refer to edge (i, j) as being *activated*. The transition rate decreases back to k_{ij} when $x_i/x_i^d < q_i$. This definition is inspired by ant colony house hunting behavior, detailed in Section 6.1.1, in which quorum sensing by individual ants speeds up emigration to a new nest site when it is sufficiently populated.

The following sections present two versions of the quorum-based model. The first represents the transition rates as continuous functions of x_i, which models uncertainty in robots' estimation of this quantity. The second represents a scenario with low uncertainty, in which transition rates can be abstracted as switching between discrete values, thus giving rise to a hybrid system.

Continuous Quorum-Based Model

We assume that $\mathcal{G}$ has bidirectional edges. The *continuous quorum-based model* is defined as the baseline linear model, subject to condition (3.18), with transition rates k_{ij} replaced by rates k_{ij}^q, which are defined as follows. We specify that the increase in k_{ij}^q is a continuous function of x_i through the use of an analytic switching function,

$$\sigma_i(x_i) = \left(1 + e^{\gamma\left(q_i - \frac{x_i}{x_i^d}\right)}\right)^{-1}, \tag{3.28}$$

where the constant $\gamma \gg 1$ is chosen such that $\sigma_i \approx 1$ when $x_i/x_i^d = q_i + \epsilon$, where $\epsilon > 0$ is small. Note that $\sigma_i \to 1$ as x_i/x_i^d increases from q_i and that $\sigma_i \to 0$ as x_i/x_i^d decreases from q_i. This is similar to threshold methods described by [22] and [3]. Using switch (3.28), we provide two alternative definitions for k_{ij}^q. Each edge $(i, j) \in \mathcal{E}$ is assigned a maximum possible rate k_{ij}^{max}, which is derived from the maximum allowable flux of robots switching between tasks i and j. In a system where tasks are located at different sites, this flux would be determined by the traffic capacity of the route from site i to site j. In the first definition, k_{ij}^q increases to this maximum:

$$k_{ij}^q = k_{ij} + \sigma_i(x_i)\left(k_{ij}^{max} - k_{ij}\right) . \tag{3.29}$$

35

In the second, k_{ij}^q increases to a multiple $\alpha > 0$ of k_{ij}, with $\max_{(i,j)\in\mathcal{E}} \alpha k_{ij} < \min_{(i,j)\in\mathcal{E}} k_{ij}^{max}$:

$$k_{ij}^q = k_{ij} + \sigma_i(x_i)(\alpha - 1)k_{ij} \ . \tag{3.30}$$

Switched Linear System

Another way to represent quorum dependencies in transition rates is to model the swarm as a hybrid system, specifically a *switched linear system* [100]. This model takes the form

$$\dot{\mathbf{x}} = -\mathbf{K}_\pi \mathbf{x} \ , \tag{3.31}$$

where $\{\mathbf{K}_p \mid p \in \mathcal{P}\}$ is a family of constant matrices that is parameterized by an index set $\mathcal{P}$, and $\pi : [0, \infty) \to \mathcal{P}$ is a piecewise constant function of time called the *switching signal*. Each $\mathbf{K}_p$ is associated with a population mode, $\omega \in \Omega_p$. If the switched system models the task reallocation scenario described in Section 3.3.2, then each $\mathbf{K}_p$ has structure (3.14). The transition rates (3.29) and (3.30) can be adapted to this model by replacing the analytic switch $\sigma_i(x_i)$ with $U(x_i/x_i^d - q_i)$, where U is the unit step function,

$$U(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ 0 & \text{if} \quad x \le 0. \end{cases} \tag{3.32}$$

Chapter 4

Analysis Techniques

The chapter presents analysis techniques that can be applied to the swarm models described in Chapter 3 to study their behavior over time.

4.1 Micro-Continuous Model

As described in Section 3.1, the micro-continuous model can be implemented in a realistic 3D simulator or, more simply, as a point-agent simulation. The time evolution of quantities of interest, such as species populations, can be obtained from the simulation, and statistical properties of the quantities can be derived from an ensemble of these trajectories.

4.2 Macro-Discrete Model

As explained in Section 3.2, a stochastic simulation algorithm that is equivalent to the Chemical Master Equation can be used to numerically compute trajectories of integer species populations. We use Gillespie's Direct Method [47,50,51] to perform stochastic simulations of the system. This method simulates a sequence of reactions in a CRN and their initiation times using the reaction rate constants k_{ij}. These

reactions describe robot transition events, as described in Section 3.1.

The Direct Method is implemented as follows. First, the initial number of elements of each type of species is stored in a counter and the propensities a_{ij}, defined in Section 3.1, are calculated. The next reaction is selected according to a uniform probability distribution over the propensities, and the time until its occurrence, $\Delta\tau$, is computed from an exponential distribution with $\sum_{(i,j)\in\mathcal{E}} a_{ij}$ as its parameter. The time is advanced by $\Delta\tau$ and the reaction is simulated by decrementing the species populations in complex i and incrementing the populations in complex j. The increment of product populations may occur immediately or at a future time that represents the completion of the reaction, such as the end of a robot's navigation between tasks at two different sites. Whenever the counter of species populations is updated, the propensities must be recalculated and a new reaction and $\Delta\tau$ are computed.

The Direct Method can be considered a centralized approach to generating the time evolution of species populations, since it uses a "global planner" to initiate reactions. However, the method is mathematically equivalent to a decentralized agent-based simulation in which the system elements probabilistically undergo transitions at every time step Δt according to the fundamental hypothesis. The advantage of the centralized simulation is its faster execution than the decentralized approach, which must loop through all system elements at each time step. The Direct Method can therefore be used to simulate systems with very large populations in less time than an agent-based simulation.

4.3   Macro-Continuous Models

4.3.1   Characterization of Equilibria

A general multi-affine system in $\mathbb{R}^S$ can have multiple equilibria. For the systems that we consider, which contain only constant, linear, and bilinear terms, these

equilibria can be computed in closed form using Sylvester's method of resultants for all S when the number of bilinear terms does not exceed $2S - 2$, and for $S \leq 16$ otherwise [120]. However, as we show in the following sections, the properties of the macro-continuous models defined in Chapter 3 can be used to prove that they each have a unique, stable equilibrium. This allows us to design a target distribution $\mathbf{x^d}$ to which the system converges from any initial distribution.

Trajectories of the species populations in the macro-continuous models can be numerically integrated using standard techniques such as the Runge-Kutta method.

Multi-Affine Model

We will discuss results from Chemical Reaction Network theory that can be obtained using a network parameter called the *deficiency* [37,38]. The results apply to closed, spatially homogeneous, constant-volume systems, although they may be extended to open systems by including "pseudoreactions" of the form $0 \rightarrow X_i$, $X_i \rightarrow 0$.

A. Definitions

A *linkage class* of a CRN is a set of complexes connected by reactions; in other words, a connected component of graph $\mathcal{G}$. Let L denote the number of linkage classes in the network. Recalling that the columns of matrix $\mathbf{M}$ in model (3.15) are denoted by $\mathbf{m}_i$, $i = 1, ..., C$, where C is the number of complexes in the system, the network *rank*, K, is defined as the rank of the matrix with rows $\mathbf{m}_i - \mathbf{m}_j$, $(i,j) \in \mathcal{E}$. The deficiency of the network, δ, is defined as

$$\delta = C - L - K \ . \tag{4.1}$$

A network is *weakly reversible* if whenever there is a directed arrow pathway from complex i to complex j, there is also one from j to i.

Each equilibrium of model (3.15), $\{\bar{\mathbf{x}} \mid \mathbf{MKy}(\bar{\mathbf{x}}) = \mathbf{0}\}$, can be classified as either a *positive* equilibrium $\bar{\mathbf{x}} > \mathbf{0}$ or a *boundary* equilibrium in which $\bar{x}_i = 0$ for some i, which can be found by solving $\mathbf{y}(\bar{\mathbf{x}}) = \mathbf{0}$ [27]. A *cyclic trajectory* is a periodic

solution of model (3.15): $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^S$, $\mathbf{x}(0) = \mathbf{x}(T)$. The cyclic trajectory is called *positive* if it is entirely contained in the positive orthant of $\mathbb{R}^S$.

The *stoichiometric subspace* for the network is defined as

$$Z = \{\mathbf{z} \in \mathbb{R}^S \mid \mathbf{z} = \sum_{(i,j) \in \mathcal{E}} \alpha_{ij}(\mathbf{m}_j - \mathbf{m}_i), \ \alpha_{ij} \geq 0\} . \tag{4.2}$$

The dimension of Z is K. The vector $\dot{\mathbf{x}}$, and thus $\mathbf{x}(t) - \mathbf{x}(0)$, is constrained to Z. This implies that a trajectory starting at $\mathbf{x}(0)$ must lie in the set $\mathbf{x}(0) \oplus Z$ ($\oplus$=Minkowski sum). The *positive stoichiometric compatibility class* (PSCC) is defined as this set intersected with the positive orthant of $\mathbb{R}^S$.

B. Theoretical Results

If $\delta = 0$, then the Deficiency Zero Theorem [37,38] provides the following results. If the network is not weakly reversible, then there is no positive equilibrium or positive cyclic trajectory. If the network is weakly reversible and has mass action kinetics, then each PSCC contains a unique, asymptotically stable equilibrium, and there is no nontrivial positive cyclic trajectory. Furthermore, by Theorem 4.1 of [139], this equilibrium is globally asymptotically stable (with respect to the $\mathbf{x}$ in its PSCC) if the network does not admit any boundary equilibria.

The Deficiency One Theorem [37, 38] gives conditions under which there is a unique equilibrium in each PSCC.

Linear Models

The results in this section were first presented in [12, 13].

The following result applies to the baseline linear model, (3.19) subject to (3.20).

Theorem 1. *If the graph $\mathcal{G}$ is strongly connected, then the baseline linear model has a unique, stable equilibrium.*

Proof. Since $\mathcal{G}$ is strongly connected, the rank of $\mathbf{K}$ is $S - 1$ [118]. The null space

40

of $\mathbf{K}$, $\mathbf{x^n}$, is therefore one-dimensional. This null space is intersected by the $(S-1)$-dimensional hyperplane described by constraint (3.20). Thus, system (3.19) subject to (3.20) has a unique equilibrium point, which we call $\bar{\mathbf{x}}^{\mathbf{n}} = [\bar{x}_1^n \ ... \ \bar{x}_S^n]^T$.

Now consider the matrix $\mathbf{T} = t\mathbf{I} - \mathbf{K}$, where $t > 0$ and $\mathbf{I} \in \mathbb{R}^{S \times S}$ is the identity matrix. Choose t large enough such that $\mathbf{T}$ is a nonnegative matrix. Since $\mathcal{G}$ is strongly connected, the matrix $-\mathbf{K}$, and therefore $\mathbf{T}$, is irreducible. Because $\mathbf{T}$ is nonnegative and irreducible, by the Perron-Frobenius theorem $\mathbf{T}$ has a real, positive, simple eigenvalue $\lambda_m(\mathbf{T})$ such that all other eigenvalues of $\mathbf{T}$, $\lambda(\mathbf{T})$, satisfy $|\lambda(\mathbf{T})| < \lambda_m(\mathbf{T})$. This eigenvalue also satisfies the inequalities $\min_j \sum_{i=1}^{S} \mathbf{T}_{ij} \leq \lambda_m(\mathbf{T}) \leq \max_j \sum_{i=1}^{S} \mathbf{T}_{ij}$ [118]. Since the columns of $\mathbf{K}$ sum to 0, both sides of these inequalities are t, so $\lambda_m(\mathbf{T}) = t$. Note that $\lambda(\mathbf{T}) = \lambda(-\mathbf{K}) + t$. Thus, the eigenvalue of $-\mathbf{K}$ corresponding to $\lambda_m(\mathbf{T})$ is 0, and all other eigenvalues of $-\mathbf{K}$ satisfy $|\lambda(-\mathbf{K})+t| < t$. It follows that $-\mathbf{K}$ has a simple zero eigenvalue and all its other eigenvalues satisfy $Re(\lambda(-\mathbf{K})) < 0$. Therefore, the equilibrium point $\bar{\mathbf{x}}^{\mathbf{n}}$ is stable. $\qquad\square$

This equilibrium can be calculated from $\mathbf{K}$ as [118]:

$$\bar{x}_i^n = K_{ii} / \sum_{j=1}^{S} K_{jj} \ , \quad i = 1, ..., S \ , \tag{4.3}$$

where K_{ij} is the cofactor of $\mathbf{K}$ obtained by deleting row i and column j.

Since the linear chain model, (3.26) subject to (3.27), has the same form as the baseline linear model, the next result immediately follows from Theorem 1.

Corollary 1. *If $\mathcal{G}$ is strongly connected, then the linear chain model has a unique, stable equilibrium.*

Denote the equilibrium of the linear chain model by $\bar{\mathbf{z}}^{\mathbf{n}} = [\bar{\mathbf{x}}^{\mathbf{n}} \ \bar{\mathbf{y}}^{\mathbf{n}}]^T$. At equilibrium in this model, the incoming and outgoing flux at each virtual task along the path from task i to j is $k_{ij}\bar{x}_i^n$, yielding the following equilibrium values of $y_{ij}^{(u)}$, $u = 1, ..., \omega_{ij}$:

$$\bar{y}_{ij}^{n(u)} = k_{ij}\bar{x}_i^n / \theta_{ij} \ . \tag{4.4}$$

41

Substituting $\bar{y}_{ij}^n = \sum_{u=1}^{\omega_{ij}} \bar{y}_{ij}^{n(u)}$ into Equation (3.23) gives the conservation equation for this system at equilibrium:

$$\sum_{i=1}^{S} \bar{x}_i^n \left(1 + \sum_{j|(i,j)\in\mathcal{E}} k_{ij}\omega_{ij}/\theta_{ij} \right) = 1 \tag{4.5}$$

Then the equilibrium values $\bar{x}_i^n$ can be shown to be [118]:

$$\bar{x}_i^n = K_{ii}/\sum_{p=1}^{S}(1 + \sum_{j|(p,j)\in\mathcal{E}} k_{pj}\omega_{pj}/\theta_{pj})K_{pp} , \quad i = 1, ..., S . \tag{4.6}$$

Comparing the equilibrium values (4.6) of the linear chain model with the values (4.3) of the corresponding baseline linear model, it is evident that the ratio between $\bar{x}_i^n$ at any two real tasks is the same in both models. However, since $k_{pj}\omega_{pj}/\theta_{pj} > 0$, the $\bar{x}_i^n$ of the chain model are *lower* than those of the baseline model.

Quorum-Based Models

The following result for the continuous quorum-based model is proven in [69] using Lyapunov stability theory.

Theorem 2. *The continuous quorum-based model with k_{ij}^q defined by Equation (3.30) and $q_i = q$, $i = 1, ..., S$, converges asymptotically to $\mathbf{x^d}$.*

The stability of a switched linear system (3.31) cannot be inferred from the stability of the system corresponding to each population mode. Lyapunov stability theorems for hybrid systems can be used to characterize an equilibrium of this type of system [148].

4.3.2 Reachability Analysis

The material in this section was first presented in [14].

If the macro-continuous model can be represented by a hybrid system, then reachability analysis can be used to determine whether the system can enter a certain

region of the continuous state space starting from a given set of initial conditions. Hence, this type of analysis can verify whether a swarm that switches between different sets of reaction rate constants can achieve (or avoid) a specified global objective, defined as a set of species populations. Here we consider hybrid systems in which the continuous state space associated with each mode is a hyper-rectangle.

This section presents MARCO (Multi-Affine Reachability analysis using Conical Overapproximations), a novel direct reachability analysis algorithm for a hybrid system whose modes are each characterized by multi-affine continuous dynamics. MARCO is shown to yield results that are superior to those of the indirect multi-affine reachability algorithm developed in [7, 56, 81], referred to here as the MAR1 algorithm. MAR1 exploits the convexity of multi-affine functions on hyper-rectangles and the fact that the vector field inside a hyper-rectangle is uniquely determined by its values at the vertices. Once a state is inside a hyper-rectangle, the algorithm considers the entire hyper-rectangle to be reachable. Because of this, the algorithm computes *conservative approximations* of the reachable set. While this approximation is guaranteed to include all reachable states, it can be overly conservative and in many simple cases (for example, constant vector fields along the diagonals of the hyper-rectangles) yield little insight into the actual behavior of the system.

MARCO computes less conservative reachable sets than MAR1 without sacrificing accuracy. Like MAR1, MARCO performs a computationally inexpensive reachability analysis within each mode by exploiting the convexity property of multi-affine vector fields on hyper-rectangles. However, MARCO determines a better conical approximation for the reachable set, thus providing a finer level of granularity for the reachable set without incurring a significantly higher penalty for computations. The technique used by MARCO for overapproximating the reachable set within a mode is similar to that used in HyTech [62] and PHAVer [44]. A higher degree of precision for the entire reachable set can be achieved by increasing the resolution of the rectangular partitions.

43

Due to the simplicity of its reachability operations, MARCO is suitable for multi-affine hybrid systems with many modes, such as a system that closely approximates a hybrid automaton with nonlinear dynamics. Thus, in principle, it is more readily applicable to such systems than existing reachability algorithms that use direct techniques for nonlinear hybrid systems, such as MATISSE [52] and CheckMate [140].

Definitions

A *hyper-rectangular multi-affine switched system (HMS)* is defined as the seven-tuple $H = (X, X_0, \Omega, I, F, T, A)$. $X \subset \mathbb{R}^n$ is the continuous space of state variables $\mathbf{x}$, $X_0 \subset X$ is a set of initial states, and Ω is a set of discrete modes. I maps the modes to subsets of X such that if the system is at mode $\omega \in \Omega$, then $\mathbf{x} \in I(\omega)$, the location invariant of ω. The location invariants are n-dimensional hyper-rectangles, which are defined as follows. For each dimension $j = 1, ..., n$, specify a strictly monotonically increasing sequence of values, $\{x_0^{(j)}, x_1^{(j)}, \cdots, x_{D_j}^{(j)}\}$. A mode ω is labeled by an n-dimensional coordinate vector $\omega = (k_1, \cdots, k_n)$, where $k_j \in \{1, ..., D_j\}$. Then $I(\omega)$ is the hyper-rectangle $[x_{k_1-1}^{(1)}, x_{k_1}^{(1)}] \times [x_{k_2-1}^{(2)}, x_{k_2}^{(2)}] \cdots \times [x_{k_n-1}^{(n)}, x_{k_n}^{(n)}]$. F is a map that assigns a continuous, autonomous vector field to each mode ω, $\dot{\mathbf{x}} = f_\omega(\mathbf{x}) \in \mathbb{R}^n$, where f_ω is a multi-affine function (3.9). T is a finite set of transitions between modes, each defined by a three-tuple $(\omega, \omega', g_{\omega,\omega'})$, in which $\omega, \omega' \in \Omega$ and $g_{\omega,\omega'} \subset \partial I(\omega)$ is a guard set. The transition from ω to ω' is enabled when $\mathbf{x} \in g_{\omega,\omega'}$. Each guard $g_{\omega,\omega'}$ of mode ω corresponds to a facet, denoted by $H(\omega, \omega')$, that $I(\omega)$ shares with $I(\omega')$. Finally, A is a finite set of symbols that label the transitions.

Definition 1 (Mode trajectory [150])**.** *A trajectory $(\omega, \tau, \mathbf{x}_\omega(t))$ associated with mode $\omega \in \Omega$ consists of a nonnegative time τ and a continuous and piecewise differentiable function $\mathbf{x}_\omega : [0, \tau] \to \mathbb{R}^n$ such that $\mathbf{x}_\omega(t) \in I(\omega)$ and $\dot{\mathbf{x}}_\omega(t) = f_\omega(\mathbf{x}_\omega(t))$ for all $t \in (0, \tau)$.*

Definition 2 (Trajectory of an HMS [150])**.** *A trajectory of an HMS starting from $\mathbf{x}_{\omega_0}(0) \in X_0 \subset I(\Omega_0)$, where $\Omega_0 \subset \Omega$, is defined as an infinite sequence of mode*

trajectories,

$$(\omega_0, \tau_0, \mathbf{x}_{\omega_0}(t)) \xrightarrow{a_0} (\omega_1, \tau_1, \mathbf{x}_{\omega_1}(t)) \xrightarrow{a_1} (\omega_2, \tau_2, \mathbf{x}_{\omega_2}(t)) \xrightarrow{a_2} \cdots \qquad (4.7)$$

such that at the event times $t_{\omega_j} = \sum_{i=0}^{j} \tau_i$, $\mathbf{x}_{\omega_j}(t_{\omega_j}) \in H(\omega_j, \omega_{j+1})$. *Since the HMS is defined to be a switched system,* $\mathbf{x}_{\omega_j}(t_{\omega_j}) = \mathbf{x}_{\omega_{j+1}}(0)$. *The* j^{th} *transition is labeled by* $a_j \in A$.

The ordered set of modes in Equation (4.7) after a finite number of transitions is represented by a *filiation sequence* of length $d \in \mathbb{N}$, $s = \{\omega_0, \omega_1, \cdots, \omega_{d-1}\}$. Now define a concatenation operation similar to that which is used for strings: $s * \{\sigma\} = \{\omega_0, \cdots, \omega_{d-1}, \sigma\}$. In the following definitions, ϕ_s designates an HMS trajectory whose first d modes comprise sequence s, given some $x_{\omega_0} \in X_0$.

Definition 3 (Footprint). *A footprint of degree d and filiation sequence s, $X_{s,\omega_d}^{(d)} \subset H(\omega_{d-1}, \omega_d)$, is the set consisting of $\mathbf{x}_{\omega_{d-1}}(t_{\omega_{d-1}})$ from each ϕ_s.*

Definition 4 (Forward reachable set of a mode). *The forward reachable set of mode ω_d from a set B, where $B = X_0$ if $d = 0$ and $B = X_{s,\omega_d}^{(d)}$ if $d > 0$, is $X_{r,\omega_d}(B) \subset I(\omega_d)$. It consists of the union of states*

$$\mathbf{x}_{\omega_{d-1}}(t_{\omega_{d-1}}) = \mathbf{x}_{\omega_d}(0) \ \cup \ \{\mathbf{x}_{\omega_d}(t) \mid t \in (0, \tau_d)\} \ \cup \ \mathbf{x}_{\omega_d}(t_{\omega_d}) = \mathbf{x}_{\omega_{d+1}}(0) \qquad (4.8)$$

from each ϕ_s for which $\omega_d \in s$.

Definition 5 (Forward reachable set of an HMS). *The forward reachable set X_r from an initial set X_0 of an HMS is the set of all continuous states $\mathbf{x}_\omega(t)$ associated with each ϕ_s.*

Definition 6 (Time-elapse cone). *The time-elapse cone C_ω for mode $\omega = (k_1, \cdots, k_n)$ is the cone generated by nonnegative linear combinations of the velocity vectors at the vertices of $I(\omega)$ (Θ_j is defined by Equation (3.10)):*

$$C_\omega \ = \ \{\sum_{j=0}^{2^n-1} \lambda_{\Theta_j} f_\omega(x_{k_0+(k_1-k_0)i_1(j)}^{(1)}, \cdots, x_{k_{n-1}+(k_n-k_{n-1})i_n(j)}^{(n)}) \mid \lambda_{\Theta_j} \geq 0\} \ . \qquad (4.9)$$

The following definitions specify the core steps of the MARCO algorithm.

Definition 7 (Overapproximated reachable set of a mode). *Consider a mode ω and a set $B \subset I(\omega)$. The overapproximated reachable set in mode ω with initial set B is defined as:*

$$R_\omega(B) = (B \oplus C_\omega) \cap I(\omega) \quad . \tag{4.10}$$

Definition 8 (Overapproximated footprint). *An overapproximated footprint of degree d and filiation sequence s, $F_{s,\omega_d}^{(d)} \subset H(\omega_{d-1}, \omega_d)$ is generated as follows.*

$$
\begin{aligned}
F_{\{\omega_0\},\omega}^{(1)} &= (X_0 \oplus C_{\omega_0}) \cap H(\omega_0, \omega) \\
F_{s*\{\omega_d\},\omega}^{(d+1)} &= (F_{s,\omega_d}^{(d)} \oplus C_{\omega_d}) \cap H(\omega_d, \omega)
\end{aligned}
\tag{4.11}
$$

The footprints and their corresponding overapproximated reachable sets form a tree structure, which in practical implementations is organized as a linked list. The sequence s distinguishes among repeated passages through the same mode during the reachability calculation.

Validity of Reachable Set Overapproximation

The following results demonstrate that the reachable set computed by MARCO contains the exact reachable set X_r.

Proposition 1 ([7]). *Let $f_\omega : I(\omega) \to \mathbb{R}^n$ be a multi-affine function (3.9) and let $\mathbf{x} \in I(\omega)$. Then $f_\omega(\mathbf{x})$ is a convex combination of the values of f_ω at the 2^n vertices of $I(\omega)$.*

Proposition 2. *Let $\mathbf{x}_\omega(t)$ be defined as in Definition 1. The displacement vector $\Delta\mathbf{x}_\omega(t) = \mathbf{x}_\omega(t) - \mathbf{x}_\omega(0)$, $t \in [0, \tau]$, is contained in the convex hull of the set of*

velocities at the vertices of $I(\omega)$, scaled by the elapsed time t. That is, $\exists \{\Lambda_{\Theta_j}\}$ where $\Lambda_{\Theta_j} \in [0, 1]$, $j = 0, ..., 2^n - 1$, and $\sum_{j=0}^{2^n-1} \Lambda_{\Theta_j} = 1$, such that:

$$\Delta\mathbf{x}_\omega(t) = t \sum_{j=0}^{2^n-1} \Lambda_{\Theta_j} f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) . \qquad (4.12)$$

Proof. The solution to $\dot{\mathbf{x}}_\omega(t) = f_\omega(\mathbf{x}_\omega(t))$, $t \in [0, \tau]$, is $\mathbf{x}_\omega(t) = \mathbf{x}_\omega(0) + \int_0^t f_\omega(\mathbf{x}_\omega(s))ds$. From Proposition 1, for $s \in [0, \tau]$, $\exists \{\lambda_{\Theta_j}(s)\}$ where $\lambda_{\Theta_j}(s) \in [0, 1]$, $j = 0, ..., 2^n - 1$, and $\sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) = 1$, such that:

$$f_\omega(\mathbf{x}_\omega(s)) = \sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) \qquad (4.13)$$

The existence of $\{\lambda_{\Theta_j}(s)\}$ is guaranteed but it is not unique; choose one set. The displacement vector $\Delta\mathbf{x}_\omega(t) = \mathbf{x}_\omega(t) - \mathbf{x}_\omega(0)$ at t is:

$$\begin{aligned}
\Delta\mathbf{x}_\omega(t) &= \int_0^t \sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s) f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)})ds \\
&= \sum_{j=0}^{2^n-1} f_\omega(x^{(1)}_{k_0+(k_1-k_0)i_1(j)}, \cdots, x^{(n)}_{k_{n-1}+(k_n-k_{n-1})i_n(j)}) \int_0^t \lambda_{\Theta_j}(s)ds \quad (4.14)
\end{aligned}$$

Define Λ_{Θ_j} as the integrated quantity divided by t:

$$0 \leq \Lambda_{\Theta_j} = \frac{1}{t} \int_0^t \lambda_{\Theta_j}(s)ds \leq 1 \qquad (4.15)$$

$$\sum_{j=0}^{2^n-1} \Lambda_{\Theta_j} = \sum_{j=0}^{2^n-1} \frac{1}{t} \int_0^t \lambda_{\Theta_j}(s)ds = \frac{1}{t} \int_0^t \sum_{j=0}^{2^n-1} \lambda_{\Theta_j}(s)ds = 1 . \ \square \qquad (4.16)$$

Corollary 2. *The set of continuous states $\mathbf{x}_\omega(t)$, $t \in [0, \tau]$, in a trajectory of mode ω is a subset of $\mathbf{x}_\omega(0) \oplus C_\omega$, the Minkowski sum of $\mathbf{x}_\omega(0)$ and the time-elapse cone.*

Proposition 3. $X_{r,\omega}(B) \subset R_\omega(B)$.

Proof. From Definition 4, $X_{r,\omega}(B)$ is the set of all states $\mathbf{x}_\omega(t)$ in the trajectory of mode ω such that $\mathbf{x}_\omega(0) \in B$, which by Corollary 2 is a subset of $\mathbf{x}_\omega(0) \oplus C_\omega$:

$$X_{r,\omega}(B) = \{\mathbf{x}_\omega(t) \mid \mathbf{x}_\omega(0) \in B, t \in [0, \tau]\} \subset \{\mathbf{x}_\omega(0) \oplus C_\omega \mid \mathbf{x}_\omega(0) \in B\} = B \oplus C_\omega$$

Since $X_{r,\omega}(B) \subset I(\omega)$ by definition, $X_{r,\omega}(B) \subset (B \oplus C_\omega) \cap I(\omega) = R_\omega(B)$. $\square$

Proposition 4 (Validity of overapproximation). *The set of states $\mathbf{x}_\omega(t)$ in the first d mode trajectories of an HMS trajectory ϕ_s with $\mathbf{x}_{\omega_0}(0) \in X_0$ is contained in the union of $R_{\omega_0}(X_0)$ with $R_{\omega_j}(F^{(j)}_{\{\omega_0,...,\omega_{j-1}\},\omega_j})$, $j = 1, ..., d-1$.*

Proof. By Proposition 3, $\mathbf{x}_{\omega_0}(t) \in R_{\omega_0}(X_0)$ for $\mathbf{x}_{\omega_0}(0) \in X_0$, $t \in [0, \tau_0]$. Therefore, by Definition 7, $\mathbf{x}_{\omega_0}(\tau_0) \in (X_0 \oplus C_{\omega_0}) \cap I(\omega_0)$. Also, $\mathbf{x}_{\omega_0}(\tau_0) = \mathbf{x}_{\omega_0}(t_{\omega_0}) \in H(\omega_0, \omega_1) \subset I(\omega_0)$. Thus, by Definition 8, $\mathbf{x}_{\omega_0}(\tau_0) = \mathbf{x}_{\omega_1}(0) \in F^{(1)}_{\{\omega_0\},\omega_1}$. By Proposition 3 again, $\mathbf{x}_{\omega_1}(t) \in R_{\omega_1}(F^{(1)}_{\{\omega_0\},\omega_1})$ for $t \in [0, \tau_1]$. The same set inclusions may be defined for the remaining modes in s. $\square$

Termination Conditions

There are two possible termination conditions for the algorithm.

Proposition 5 (Termination condition 1). *If $R_{\omega_d}(F^{(d)}_{s,\omega_d})$ is a subset of R_{ω_d}, the union of the reachable sets previously computed for mode ω_d, then all states $\mathbf{x}_\omega(t)$ in HMS trajectories with $\mathbf{x}_{\omega_0}(0) \in F^{(d)}_{s,\omega_d}$ are contained in R_{ω_d} and all reachable sets evolving from R_{ω_d}.*

Since the reachable set might grow by very small amounts for a long time, a second heuristic condition may be applied to ensure termination within a reasonable amount of time. Each iteration of the algorithm generates a new set of conical overapproximations and footprints; let $V(R_i)$ be the volume of the newly computed reachable set at iteration i and $V(S)$ be the volume of the state space.

Proposition 6 (Termination condition 2). *For a small constant ζ, stop if*
$$V(R_i) < V(R_{i-1}) \text{ and } V(R_i) < \zeta V(S).$$
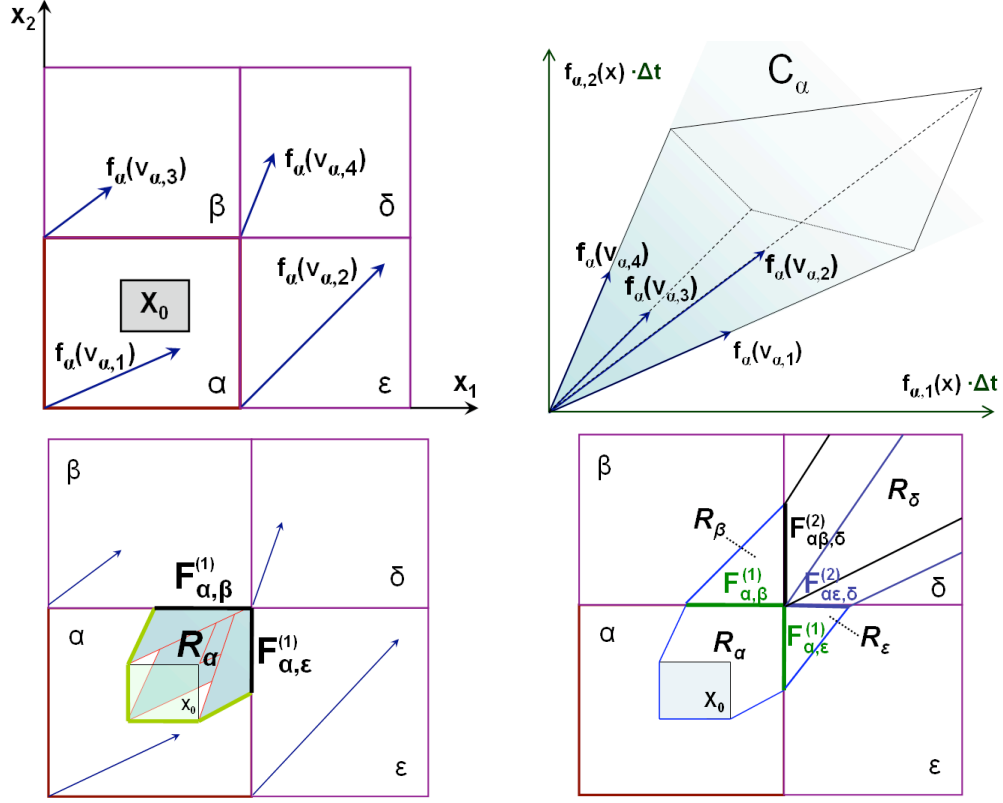
Figure 4.1: Illustration of the MARCO algorithm. (a) (upper left) Initial set X_0 and velocities at vertices of mode α; (b) (upper right) definition of the time-elapse cone C_α; (c) (lower left) computation of reachable set R_α and footprints $F^{(1)}_{\alpha,\beta}$ and $F^{(1)}_{\alpha,\epsilon}$ of adjacent modes; (d) (lower right) computation of R_β, R_δ, and R_ϵ.

Implementation

The MARCO algorithm is written in Matlab and uses the Multi-Parametric Toolbox (MPT) [88] for polyhedral operations. Figure 4.1 illustrates its steps for a two-dimensional state space, and Figure 4.2 gives an outline of the algorithm.

The user inputs the specifications of the hybrid system H. First, the set Ω of reachable modes is initialized with the modes $\Omega_0 \subset \Omega$ that contain the initial set X_0. These modes are identified as members of generation 0. In Figure 4.1a, $\Omega_0 = \alpha$. The portion of X_0 that intersects the mode invariant $I(\omega)$ for $\omega \in \Omega_0$ is the first incoming footprint of mode ω. For each mode in generation 0, a time-elapse cone C_ω is found according to Definition 6. Figure 4.1a-b shows the creation of cone C_α

from the velocities at the vertices of mode α. The cone is scaled to extend past the mode boundaries. C_ω is added to the mode footprint via a Minkowski sum and is then bounded by the mode facets to produce the overapproximated mode reachable set, $R_\omega(X_0 \cap I(\omega))$ (Figure 4.1c). Next, each adjacent mode ω' with a facet that has a nonempty intersection with $R_\omega(X_0 \cap I(\omega))$ is added to Ω if it is not already in the list, and the intersection is designated as the overapproximated incoming footprint of that mode, $F_{\{\omega\},\omega'}^{(1)}$. These modes are identified as members of the next generation. In Figure 4.1d, the footprints are $F_{\alpha,\beta}^{(1)}$, $F_{\alpha,\epsilon}^{(1)}$, and modes β, ϵ are in generation 1.

The algorithm repeats the reachable set overapproximation and footprint identification for modes in each consecutive generation. Note that a mode ω may have multiple footprints, as does mode δ in Figure 4.1d. Each footprint generates a reachable set, and the concatenation of these sets is the total reachable set within the mode. The algorithm terminates according to Proposition 5, Proposition 6, or when there are no new modes in the current generation, which occurs when the reachable set hits the boundary of the state space X, as in Figure 4.1d. The algorithm returns the total reachable set, stored as polyhedral subsets of mode invariants, that is attained from X_0.

Examples

The examples in this section illustrate the improvement of MARCO over MAR1. Figures 4.3 and 4.4 display reachable sets computed by MARCO and by a Matlab implementation of MAR1. The MARCO reachable sets are shown in dark gray or magenta, while the MAR1 sets consist of light gray boxes in the 2D examples and transparent boxes in the 3D and 4D examples. In each example, both algorithms used the same state space boundaries and mode partition. All examples were run on a standard 2 GHz laptop.

In Figure 4.3a, the dynamics in each mode consist of the constant vector field $\dot{x}_1 = 1$, $\dot{x}_2 = 0.5$, and the initial set is the box in the lower left corner. The reachable

Table 4.1: Definitions of P and Q in Equation (4.18)

$x_{1,ave}^{\omega}$	(-∞,0.3]	(0.3,0.5]	(0.5,0.7]	(0.7,0.8]	(0.8,1.0]	(1.0,1.3]	(1.3,1.6]	(1.6,∞)
P	0.1	-0.2	-0.7	-1.4	0.2	$0.5\bar{3}$	$0.9\bar{6}$	1.5
Q	0.0	1.0	2.0	3.0	1.0	$0.\bar{6}$	$0.\bar{3}$	0.0

set computed by MARCO is exact, while MAR1 predicts that all modes are reached.

Figure 4.3b displays a vector field whose integral curves are spirals with a steady state at the origin. The dynamics are given by

$$\dot{x}_1 = -x_1 + 2x_2 \qquad \dot{x}_2 = -2x_1 - x_2 \; . \tag{4.17}$$

The initial set is the box containing the steady state. The MARCO algorithm terminates and returns a conservative but finite reachable set around the equilibrium point; it essentially recognizes the presence of the steady state. The MAR1 method considers the entire space to be reached due to the velocity components pointing out of the center mode.

Figure 4.3c shows the computation of the reachable set for a three-dimensional vector field with integral curves that are helical spirals. The results are similar to those of Figure 4.3b.

Figure 4.4a shows a bistable vector field,

$$\dot{x}_1 = f(x_2) - x_1 \quad \dot{x}_2 = x_1 - x_2 \; , \quad f(x_2) = P + Qx_2 \; , \tag{4.18}$$

where $f(x_2)$ is a piecewise-linear approximation of a sigmoid-shaped function. P and Q for a mode ω depend on the particular x_1 interval that contains the average x_1 coordinate of the mode, $x_{1,ave}^{\omega}$, and are defined in Table 4.1. The initial set is located at a place where the vector field diverges. The MARCO reachable set correctly approaches and terminates at the two steady states while avoiding the unstable steady state. The MAR1 reachable set is much more conservative.

Table 4.2: Comparison of computation times and reachable set precision

Vector field	Time (sec)		Reached vol./State space vol.	
	MARCO	MAR1	MARCO	MAR1
2D constant	4.17	0.42	0.255	1.000
2D linear	2.83	0.42	0.329	1.000
3D linear	4.78	0.78	0.078	1.000
2D affine	7.27	0.94	0.266	0.714
4D multi-affine	130.31	2.53	0.022	0.061

Figure 4.4b illustrates the projection of a four-dimensional multi-affine system,

$$\dot{x}_1 = (x_1 - 10.5)(x_2 - 10.5)(x_3 - 10.5)(x_4 - 10.5)$$

$$\dot{x}_2 = (x_1 - 4.5)(x_2 - 7.5)(x_3 - 7.5)(x_4 - 1.5)$$

$$\dot{x}_3 = (x_1 - 7.5)(x_2 - 4.5)(x_3 - 1.5)(x_4 - 7.5)$$

$$\dot{x}_4 = (x_1 - 1.5)(x_2 - 1.5)(x_3 - 4.5)(x_4 - 4.5) , \tag{4.19}$$

onto the x_2, x_3, and x_4 dimensions. The system has 24 equilibria; in particular, the equilibrium $\mathbf{x}_{e1} = [10.5\ 7.5\ 1.5\ 4.5]^T$ is stable and the equilibrium $\mathbf{x}_{e2} = [10.5\ 7.5\ 4.5\ 7.5]^T$ is unstable. The initial set for the reachability computation is a box surrounding $\mathbf{x}_{e2}$. The reachable set diverges at $\mathbf{x}_{e2}$: one branch terminates at $\mathbf{x}_{e1}$, while the other runs into the state space boundary. Again, the MAR1 reachable set fails to attain the precision of the MARCO set under the same mode partition.

Table 4.2 compares the performance of the two algorithms in terms of the computation time and volume fraction of the state space reached for each example. Note that although MAR1 is faster on all examples, its overly conservative predictions of the reachable set cannot be refined with iterative partitioning.

Input: System dimension n, mode dividers, vertices of initial set I_0, dynamical parameters

Output: $R = \{R_{\omega_0}, ..., R_{\omega_N}\}$, $\omega_i \in \Omega$

$\Omega := \{\omega_i \mid I(\omega_i) \cap X_0 \neq \emptyset\}$

for all $\omega_i \in \Omega$: Generation$(\omega_i) = 0$; $R_{\omega_i} = \emptyset$

$G = -1$

do

    $G = G + 1$

    for all $\{\omega_i \mid \text{Generation}(\omega_i) = G\}$

        $R_{\omega_i}^{gen} = \emptyset$

        Calculate velocities at vertices of ω_i

        Create time-elapse cone C_{ω_i}

        Combine overlapping footprints of ω_i

        for all footprints $F_{s,\omega_i}^{(G)}$:

            $R_{\omega_i}(F_{s,\omega_i}^{(G)}) = (F_{s,\omega_i}^{(G)} \oplus C_{\omega_i}) \cap I(\omega_i)$

            for all $\{\omega_j \mid F_{s*\{\omega_i\},\omega_j}^{(G+1)} = (F_{s,\omega_i}^{(G)} \oplus C_{\omega_i}) \cap H(\omega_i, \omega_j) \neq \emptyset\}$

                if $\omega_j \notin \Omega$

                    $\Omega = \Omega * \{\omega_j\}$

                    $R_{\omega_j} = \emptyset$

                  Generation$(\omega_j) = G + 1$

                end

                $R_{\omega_i}^{gen} = R_{\omega_i}^{gen} * \{R_{\omega_i}(F_{s,\omega_i}^{(G)})\}$

        end

        if $R_{\omega_i}^{gen} \not\subset R_{\omega_i}$

            $R_{\omega_i} = R_{\omega_i} * \{R_{\omega_i}^{gen}\}$

    end

until $R_{\omega_i}^{gen} \subset R_{\omega_i} \ \forall \ \ \{\omega_i \mid Generation(\omega_i) = G\}$

Figure 4.2: MARCO reachability algorithm
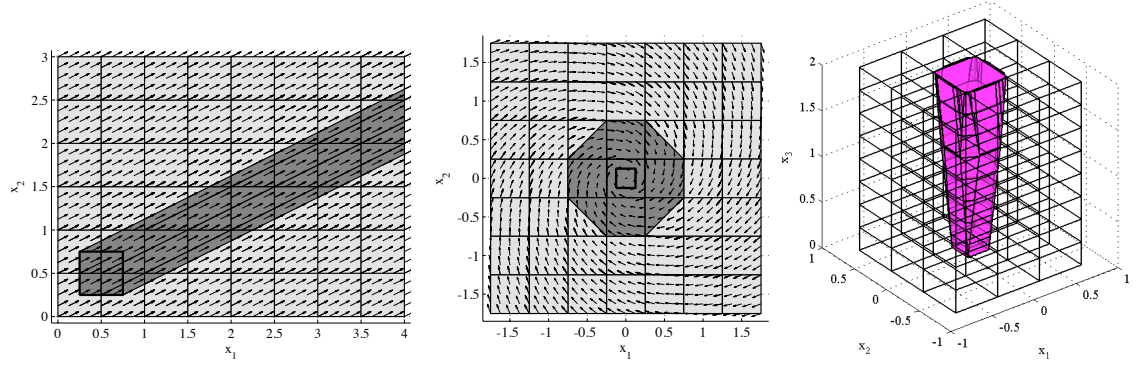
Figure 4.3: Reachable sets for (a) 2D constant field; (b) 2D linear field; (c) 3D linear field
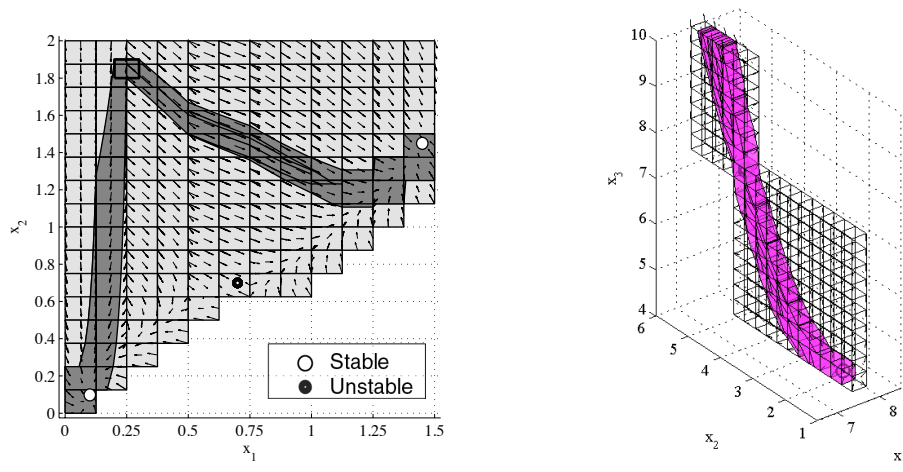
Figure 4.4: Reachable sets for (a) 2D affine field; (b) 4D multi-affine field

Chapter 5

Controller Synthesis

Section 5.1 describes the synthesis of robot motion controllers for navigation and inter-robot collision avoidance in the micro-continuous model. Section 5.2 describes the optimization of the parameters in the macro-continuous models for quick, efficient system convergence to the target equilibrium, as well as the synthesis of feedback control laws for steering the macroscopic system state through a sequence of population modes in a hybrid system.

5.1   Micro-Continuous Model

Control strategies in the micro-continuous model are used to physically guide the robots through their environment in a manner that allows them to safely complete their task. This section describes several different types of robot motion controllers. We represent each robot i as a planar agent governed by a kinematic model $\dot{\mathbf{q}}_i = \mathbf{u}_i$, where $\mathbf{q}_i \in \mathbb{R}^2$ denotes the robot's (x, y) coordinates and $\mathbf{u}_i \in \mathbb{R}^2$ is a control input.

5.1.1 Navigation

Navigation Functions

The navigation function control methodology [128–130] can be used to guide a robot to a destination point while avoiding obstacles in the environment. Navigation functions combine the three problems of path planning, trajectory planning, and robot control for trajectory tracking by providing a form for a bounded-torque feedback controller on a robot. The controller produces collision-free motion and convergence to the goal location, $\mathbf{q}^d$, the unique minimum of the function, from almost all initial free configurations.[1] To use navigation function controllers, we assume that the environment is stationary and that perfect information is available about the obstacles, geometric constants that are derived from them, and the topology of the free configuration space, $\mathcal{F}$. Additionally, each robot is assumed to have ideal sensors for position and velocity and ideal actuators that can deliver a bounded torque instantaneously.

Let $\mathcal{F} \subset \mathbb{E}^n$ be a compact, connected, analytic manifold with boundary. By Definition 1 in [130], a map $\varphi : \mathcal{F} \to [0,1]$ is a navigation function if it is smooth on $\mathcal{F}$ (at least a $C^{(2)}$ function), admissible on $\mathcal{F}$, a Morse function, and polar at a point $\mathbf{q}^d$ in the interior of $\mathcal{F}$. The configuration space is restricted to the class of generalized sphere worlds. Let $\mathcal{D}(\mathbf{q}, \rho)$ denote a Euclidean n-dimensional disk with center $\mathbf{q}$ and radius ρ. A *Euclidean sphere world* is a compact, connected submanifold of Euclidean n-space, $\mathbb{E}^n$, that is formed by removing from a large disk $\mathcal{D}_0(\mathbf{0}, \rho_0)$ M smaller, disjoint disks, $\mathcal{D}_j(\mathbf{q}_j, \rho_j)$ $(j = 1, ..., M)$, which represent obstacles. The complement of $\mathcal{D}_0$ is the zeroth obstacle. More complicated generalized sphere worlds can be deformed onto a Euclidean sphere world through a diffeomorphic mapping.

We describe the construction of φ on a Euclidean sphere world. Each obstacle is implicitly represented by an obstacle function β_i in the form $\mathrm{obs}_i = \{\mathbf{q} : \beta_i(\mathbf{q}) \leq 0\}$,

---

[1]Each obstacle introduces at least one saddle point of the navigation function, but in practice these points do not present a problem because their domain of attraction has an empty interior.

where $\beta_i = 0$ on the boundary of the obstacle. The obstacle functions are given by

$$\beta_0(\mathbf{q}) = -\|\mathbf{q} - \mathbf{q}_0\|^2 + \rho_0^2 \ , \qquad \beta_j(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_j\|^2 - \rho_j^2 \ , \quad j = 1, ..., M \ . \qquad (5.1)$$

The product of obstacle functions is $\beta = \prod_{i=1}^{M} \beta_i$.

The navigation function is an analytic switch, varying smoothly between 0 at $\mathbf{q}^d$ and 1 at the zeros of β, that has been "sharpened" by a function that makes $\mathbf{q}^d$ a nondegenerate critical point. The navigation function is defined as

$$\varphi_\kappa(\mathbf{q}, \mathbf{q}^d) = \frac{\|\mathbf{q} - \mathbf{q}^d\|^2}{[\|\mathbf{q} - \mathbf{q}^d\|^{2\kappa} + \beta(\mathbf{q})]^{1/\kappa}} \ , \qquad (5.2)$$

where $\kappa > 0$ is a parameter. Theorem 4 of [128] states that φ_κ is a navigation function on $\mathcal{F}$ if $\kappa \geq N$, where N is a positive integer that is a function of the geometric data. As κ increases, undesired local minima disappear. The control input $\mathbf{u}_i$ is set to be proportional to the negative gradient of $\varphi_\kappa(\mathbf{q}_i, \mathbf{q}^d)$.

Design of Vector Fields on Convex Polygons

Another provably correct way to integrate path planning and robot control for navigation is presented in [30]. The free space is decomposed into a collection of convex polygons, and a graph search algorithm is applied to the adjacency graph of the polygons to find a path from the polygon containing the initial location of the robot to the polygon containing the goal location. A potential function for each polygon is defined as the composition of the solution of Laplace's equation on the unit disk with a mapping from the polygon to the unit disk. When $\mathbf{q}_i$ is inside the polygon, the control input $\mathbf{u}_i$ is set to be the negative gradient of this potential function, which is orthogonal to the boundary of the polygon and free of local minima. In this way, the robot is directed from one polygon in the path to another through the common boundary of adjacent polygons until it reaches the goal polygon. The methodology can be extended to define control policies for robots with dynamical constraints.

5.1.2 Inter-Robot Collision Avoidance

In [145] and [95], artificial potential functions are used to define local robot controllers that enforce desired inter-robot distances within a group. The functions are shaped such that two neighboring robots will approach each other when the distance between them exceeds a specified value and move away from each other when the distance is less than this value. We can use these functions to implement inter-robot collision avoidance by activating the controller only when two robots are within the range of distances associated with repulsion.

The relative position vector between robots i and j is denoted by $\mathbf{q}_{ij} = \mathbf{q}_i - \mathbf{q}_j = [x_{ij} \; y_{ij}]^T$. Let $\mathcal{N}_i$ be the neighbor set of robot i, which may be the robots with which i is allowed to communicate (giving rise to a fixed network) or the robots that are physically proximal to i (a dynamic network). $V_{ij} = V_{ij}(\|\mathbf{q}_{ij}\|)$ is an artificial potential function that is differentiable, nonnegative, and radially unbounded. It attains its unique minimum when robots i and j are located at a specified distance, and it approaches infinity as $\|\mathbf{q}_{ij}\| \to 0$. An example V_{ij} is illustrated in Figure 5.1. The directional derivative of V_{ij} along the vector $\mathbf{q}_{ij}$ is computed as:

$$\nabla_{\mathbf{q}_{ij}} V_{ij} = \frac{\partial V_{ij}}{\partial \mathbf{q}_{ij}} = \frac{\partial V_{ij}}{\partial \|\mathbf{q}_{ij}\|} \frac{\partial \|\mathbf{q}_{ij}\|}{\partial \mathbf{q}_{ij}} = \frac{\partial V_{ij}}{\partial \|\mathbf{q}_{ij}\|} \frac{\mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|} \; , \tag{5.3}$$

where $\partial \|\mathbf{q}_{ij}\|/\partial \mathbf{q}_{ij} = [\partial \|\mathbf{q}_{ij}\|/\partial x_{ij} \; \partial \|\mathbf{q}_{ij}\|/\partial y_{ij}]^T$. $\nabla_{\mathbf{q}_i} V_{ij}$ and $\nabla_{\mathbf{q}_j} V_{ij}$ can be calculated using a similar application of the chain rule. The following relationship exists between the three directional derivatives:

$$\nabla_{\mathbf{q}_{ij}} V_{ij} = \nabla_{\mathbf{q}_i} V_{ij} = -\nabla_{\mathbf{q}_j} V_{ij} \; . \tag{5.4}$$

The input $\mathbf{u}_i$ can now be defined as

$$\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} \nabla_{\mathbf{q}_i} V_{ij} \; . \tag{5.5}$$
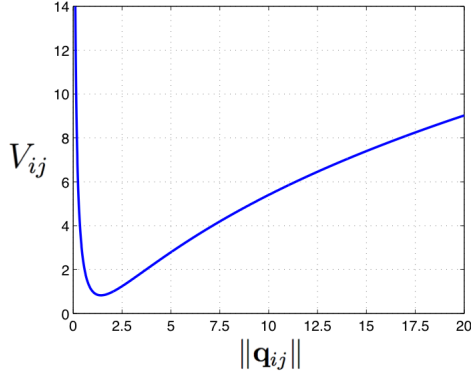
Figure 5.1: Example potential function, $V_{ij} = \ln^2(\|\mathbf{q}_{ij}\|) + \frac{1}{\|\mathbf{q}_{ij}\|}$

5.2 Macro-Continuous Model

5.2.1 Optimization

The material in this section was first presented in [13, 109].

The continuous models in Chapter 3 describe a swarm as a function of the reaction rate constants, k_{ij}, which are the designable parameters that control system performance. The k_{ij} must cause the system to converge to the target equilibrium in a reasonably short amount of time, perhaps while adhering to a specification on a metric of efficiency. We quantify the degree of convergence of a macro-continuous model to $\mathbf{x^d}$ by the *fraction of misplaced robots*,

$$\mu(\mathbf{x}) = ||\mathbf{x} - \mathbf{x^d}||_2 \ . \tag{5.6}$$

We say that one system converges faster than another if it takes less time t_f for $\mu(\mathbf{x})$ to decrease to some small fraction f of its initial value, $\mu(\mathbf{x^0})$.

This section defines optimization problems for computing the k_{ij} according to these objectives. An advantage of using the macro-continuous models to design the k_{ij} is that the optimization problems are then independent of the number of species, so they are scalable with the populations in the system.

Multi-Affine Model

Define $\mathbf{k} \in \mathbb{R}^{|\mathcal{E}|}$ as the vector of all k_{ij} in the system. We formulate an optimization problem to compute the $\mathbf{k}$ that minimizes the convergence time of model (3.15) to a target species distribution $\mathbf{x^d}$. If the model can be proven to have a unique, stable equilibrium, then we can set $\mathbf{x^d}$ to be this equilibrium through the following constraint on $\mathbf{K}$:

$$\mathbf{MKy}(\mathbf{x^d}) = \mathbf{0} \ . \tag{5.7}$$

There may be other constraints on the k_{ij} such as upper and lower bounds on their values,

$$\mathbf{b}_l \leq \mathbf{k} \leq \mathbf{b}_u \ . \tag{5.8}$$

It is possible to find the $\mathbf{k}$ that *directly* minimizes the system convergence time by using a stochastic optimization method to solve the optimization problem below. We chose to use a Monte Carlo method because of its simplicity and the fact that it yields reasonable improvements in t_f with moderate computing resources.

[$\mathbf{P_{MA}}$]   Minimize t_f subject to constraints (5.7) and (5.8).

Implementation: At each iteration, $\mathbf{k}$ is perturbed by a random vector and projected onto the null space of linearly independent rows of a matrix $\mathbf{N}$ defined such that $\mathbf{Nk} = \mathbf{MKy}(\mathbf{x^d}) = \mathbf{0}$. Once $\mathbf{k}$ also satisfies constraint (5.8), it is used to simulate model (3.15), and a Newton scheme is used to compute the exact time t_f when $\mu(\mathbf{x}) = f\mu(\mathbf{x^0})$.

It may be possible to formulate the problem as a convex optimization problem, which can be solved efficiently. Toward this end, we construct an analytical expression for the system convergence time to $\mathbf{x^d}$. We quantify this time in terms of the system *relaxation times* τ_i, $i = 1, ..., R$, the times in which different modes (dynamically independent variables) of the system converge to a stable equilibrium after perturbation [61, 71]. Various measures of the average relaxation time of a CRN have been defined, but they are applicable only under certain conditions, such as a

linear reaction sequence [60] [135]. A common approach to analyzing the dynamical properties of a CRN is to linearize the macro-continuous model of the system about an equilibrium point and study the properties of the associated Jacobian matrix $\mathbf{J} = \mathbf{SG}$, where $\mathbf{S}$ is the stoichiometric matrix defined in Section 3.3.1 and the entries of $\mathbf{G}$ are $\mathbf{G}_{ij} = d\nu_i/dx_j$ [71]. By constraint (5.7), the equilibrium about which the model is linearized is $\mathbf{x^d}$. Denoting the eigenvalues of $\mathbf{J}$ by λ_i, a common measure of relaxation time is $\tau_i = 1/|Re(\lambda_i)|$. Since the λ_i are negative at a stable equilibrium, one way to yield fast convergence is to choose rate constants that minimize the largest λ_i. An alternative estimate of relaxation time, also derived by linearizing the system around its equilibrium $\mathbf{x^d}$, is defined as [61]

$$\tau_j = \left(\sum_{i=1}^{S} (-\mathbf{S}_{ij}) \frac{d\nu_j}{dx_i} \right)^{-1}_{\mathbf{x}=\mathbf{x^d}}. \tag{5.9}$$

Possible objective functions in the optimization problem are the average τ_j^{-1} and the minimum τ_j^{-1}. These functions should be *maximized* to produce fast convergence to $\mathbf{x^d}$. If the selected objective function is a concave function of the k_{ij}, then the problem of maximizing the function subject to constraints (5.7) and (5.8) is a convex optimization problem.

Linear Models

We consider the problem of redeploying a swarm represented by the baseline linear model, (3.19) subject to (3.20), from an initial distribution among a set of tasks, $\mathbf{x^0}$, to a target distribution $\mathbf{x^d}$. Theorem 1 proves that the baseline linear model always converges to a single equilibrium $\mathbf{\bar{x}^n}$, which represents the steady-state distribution of population fractions among the S tasks. Hence, we can achieve a predefined $\mathbf{x^d}$ from any $\mathbf{x^0}$ by specifying that $\mathbf{\bar{x}^n} = \mathbf{x^d}$ through the following constraint on $\mathbf{K}$:

$$\mathbf{Kx^d} = \mathbf{0} . \tag{5.10}$$

When the k_{ij} are chosen such that the corresponding $\mathbf{K}$ matrix satisfies constraint

(5.10), a swarm of robots that use the k_{ij} as stochastic transition rules will redistribute from any $\mathbf{x^0}$ to $\mathbf{x^d}$. Since (3.19) is a linear system, the rate of convergence of $\mathbf{x}$ to $\mathbf{x^d}$ is governed by the real parts of the eigenvalues of $\mathbf{K}$, which are positive homogenous functions of the k_{ij} [144]. Thus, the rate of redistribution can be made arbitrarily fast by using high k_{ij}. However, in actual robotic systems there is often a substantial cost to using high k_{ij}. At equilibrium, the probability that any robot doing task i will start switching to task j in time step δt is $k_{ij} n_i^d \delta t$. Thus, raising k_{ij} increases the equilibrium "traffic" of robots transitioning between tasks i and j. This is also evident from Equation (4.4). This switching expends power; for instance, if the tasks are at different locations, the robots must travel between them and may experience delays due to congestion along the route.

Thus, when choosing the k_{ij}, we are faced with a tradeoff between rapid convergence to $\mathbf{x^d}$ and long-term system efficiency, i.e. few idle transitions between tasks once $\mathbf{x^d}$ is achieved. In light of this tradeoff, we compute the matrix $\mathbf{K}$ as the solution to an optimization problem that maximizes a measure of the convergence rate of system (3.19) to $\mathbf{x^d}$ subject to one of two possible constraints on task transitions at equilibrium. The first is a limit on the total equilibrium flux of robots switching between tasks:

$$\sum_{(i,j)\in\mathcal{E}} k_{ij} x_i^d \le c_{tot} \ . \tag{5.11}$$

Note that in practice, the total equilibrium flux is actually $\sum_{(i,j)\in\mathcal{E}} k_{ij} b x_i^d \le b c_{tot}$, where b is the population fraction at the real tasks (i.e., not in transit).

Constraint (5.11) does not dictate how the transitioning population is distributed among edges. An alternative constraint achieves this with a set of limits on the equilibrium flux between each pair of adjacent tasks:

$$k_{ij} x_i^d \le c_{ij}, \quad (i,j) \in \mathcal{E} \ . \tag{5.12}$$

We formulate several versions of this optimization problem, summarized in Table 5.1 (FC=fully connected, ROC= rate of convergence). Each version is tailored to

Table 5.1: **K** Optimization Problems

Problem	FC	$\mathbf{x^0}$	Objective
$\mathbf{P_L^1}, \mathbf{P_L^{1R}}$			maximize asymptotic ROC[a]
$\mathbf{P_L^2}$	✓		maximize overall ROC
$\mathbf{P_L^3}$		✓	minimize time to reach $0.1\mu(\mathbf{x^0})$
$\mathbf{P_L^4}$	✓	✓	maximize ROC along $\mathbf{x^d} - \mathbf{x^0}$

[a]Maximizes *all* nonzero eigenvalues of **K** when the Markov process on $\mathcal{G}$ is reversible and constraint (5.12) is used; see Section 5.2.1A.

an application with a particular combination of properties. The graph $\mathcal{G}$ will be fully connected, in addition to strongly connected, if there are no physical or logical constraints on the flow of robots between pairs of tasks, such as a path in a disaster area that is only wide enough for robots to travel in one direction. In addition, it may be possible to obtain $\mathbf{x^0}$, for instance by identifying robots in an image from an aerial camera.

Problem P_L^3 is solved using a Monte Carlo method that directly minimizes convergence time. The resulting system is used as a baseline to compare the systems computed by the other problems, which manipulate convergence time by maximizing functions of the eigenvalues of **K** using linear or semidefinite programs. Since these types of programs can be solved with methods that have polynomial complexity in the worst case [151], we can efficiently compute the $S \times S$ matrix **K** for large S. Thus, our allocation approach scales well with the number of tasks.

The **K** design methods can also be applied to the more realistic model (3.22) with Erlang-distributed τ_{ij} when it is expressed as the equivalent linear chain model, (3.26) subject to (3.27). We list the necessary modifications for the implementation of Problem P_L^3.

A. Maximizing the asymptotic rate of convergence

If $\mathcal{G}$ is strongly connected, but not necessarily fully connected, and $\mathbf{x^0}$ is unknown, we can designate the asymptotic rate of convergence of system (3.19) to $\mathbf{x^d}$ as the

quantity to maximize. Let $\lambda_i(\mathbf{K})$ signify the eigenvalue of $\mathbf{K}$ with the ith smallest real part of all the eigenvalues. By Theorem 1, $\lambda_1(\mathbf{K}) = 0$ and $\lambda_i(\mathbf{K}) > 0$ for $i = 2, ..., S$. Thus, the asymptotic rate of convergence is governed by $Re(\lambda_2(\mathbf{K}))$. Noting that $\mathbf{K}$ is usually not symmetric, we first find a symmetric matrix $\mathbf{S}$ such that $\lambda_2(\mathbf{S}) \leq Re(\lambda_2(\mathbf{K}))$. We replace the objective function $Re(\lambda_2(\mathbf{K}))$ by $\lambda_2(\mathbf{S})$. We can write this problem as a semidefinite program with a linear matrix inequality that arises from a variational characterization of $\lambda_2(\mathbf{S})$.

Theorem 3. *Define* $\mathbf{\Pi} = diag(\mathbf{x^d})$, *which is invertible since* $\mathbf{x^d} > \mathbf{0}$. *Let* $\mathbf{K}$ *be a matrix with the structure in* (3.14). *Define the matrices*

$$\mathbf{N} = \tfrac{1}{2}(\mathbf{\Pi}\mathbf{K}^T + \mathbf{K}\mathbf{\Pi}) , \tag{5.13}$$

$$\tilde{\mathbf{K}} = \mathbf{\Pi}^{-1/2}\mathbf{K}\mathbf{\Pi}^{1/2} ,$$

$$\mathbf{S} = \tfrac{1}{2}(\tilde{\mathbf{K}} + \tilde{\mathbf{K}}^T) = \mathbf{\Pi}^{-1/2}\mathbf{N}\mathbf{\Pi}^{-1/2}. \tag{5.14}$$

Then $\lambda_2(\mathbf{S}) \leq Re(\lambda_2(\mathbf{K}))$.

Proof. Define a convex, symmetric function $h : \mathbb{R}^S \to \mathbb{R}$,

$$h(\mathbf{x}) = -\min\{x_i + x_j\} , \quad i, j \in \{1, ..., S\} . \tag{5.15}$$

Let $\lambda(\mathbf{A})$ be the vector of the eigenvalues of a matrix $\mathbf{A}$. By Theorem 16.4 of [98], since h is convex and symmetric, $h(Re(\lambda(\mathbf{K})))$ is the infimum of $h(\tfrac{1}{2}\lambda(\mathbf{M} + \mathbf{M}^T))$ over all matrices $\mathbf{M}$ similar to $\mathbf{K}$. Thus, since $\tilde{\mathbf{K}}$ is similar to $\mathbf{K}$,

$$h(Re(\lambda(\mathbf{K}))) \leq h(\tfrac{1}{2}\lambda(\tilde{\mathbf{K}} + \tilde{\mathbf{K}}^T)) = h(\lambda(\mathbf{S})) , \tag{5.16}$$

where the equality on the right comes from Equation (5.14).

Now we evaluate both sides of inequality (5.16). By Theorem 1, $h(Re(\lambda(\mathbf{K}))) = -Re(\lambda_2(\mathbf{K}))$. We observe that $\lambda(\mathbf{S}) = Re(\lambda(\mathbf{S}))$ because $\mathbf{S}$ is symmetric. We now show that $\mathbf{S}$ is positive semidefinite, denoted by $\mathbf{S} \succeq 0$, which implies that $h(\lambda(\mathbf{S})) = -\lambda_2(\mathbf{S})$ and hence reduces (5.16) to the inequality $\lambda_2(\mathbf{S}) \leq Re(\lambda_2(\mathbf{K}))$. By Equation

64

(5.14), $\mathbf{S} \succeq 0$ if $\mathbf{N} \succeq 0$. Since $\mathcal{G}$ is strongly connected, $\lambda_2(\mathbf{N}) > 0$ (Lemma 10 of [156]). Using property (3.12) and constraint (5.10), $\mathbf{N1} = \frac{1}{2}(\mathbf{\Pi K}^T \mathbf{1} + \mathbf{Kx^d}) = \mathbf{0}$, and so $\lambda_1(\mathbf{N}) = 0$ with corresponding eigenvector $\mathbf{1}$. Therefore, $\mathbf{N} \succeq 0$. $\qquad\square$

Denote the vector of all k_{ij} by $\mathbf{k} \in \mathbb{R}^{S^2 - S}$, which is the optimization variable. Both constraints on transitions can be written in the form $f(\mathbf{k}) \leq 1$, where $f : \mathbb{R}^{S^2 - S} \to \mathbb{R}$ is defined as f_{tot} for constraint (5.11) and f_{ind} for constraint (5.12):

$$f_{tot}(\mathbf{k}) = \sum_{(i,j) \in \mathcal{E}} k_{ij} x_i^d, \quad f_{ind}(\mathbf{k}) = \max_{(i,j) \in \mathcal{E}} \{ k_{ij} x_i^d / c_{ij} \}. \tag{5.17}$$

Now we can state the optimization problem as: maximize $\lambda_2(\mathbf{S})$ subject to $f(\mathbf{k}) \leq 1$, $\mathbf{k} \geq \mathbf{0}$. We use an alternate formulation [144]: minimize $f(\mathbf{k})$ subject to $\lambda_2(\mathbf{S}) \geq 1$, $\mathbf{k} \geq \mathbf{0}$. The vector $\mathbf{q} = [(x_1^d)^{1/2} \; ... \; (x_S^d)^{1/2}]^T$ is the eigenvector of $\mathbf{\Pi}^{-1/2} \mathbf{N} \mathbf{\Pi}^{-1/2}$ corresponding to the zero eigenvalue. From Equation (5.14) and the characterization of eigenvalues in [66], the constraint $\lambda_2(\mathbf{S}) \geq 1$ can be expressed as:

$$\lambda_2(\mathbf{S}) = \inf_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T \mathbf{q}=0}} \mathbf{x}^T \mathbf{\Pi}^{-1/2} \mathbf{N} \mathbf{\Pi}^{-1/2} \mathbf{x} \geq \inf_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T \mathbf{q}=0}} \mathbf{x}^T (\mathbf{I} - \mathbf{qq}^T) \mathbf{x} \tag{5.18}$$

The problem can now be posed as **Problem $\mathbf{P_L^1}$**, in which the linear matrix inequality comes from (5.18).

[$\mathbf{P_L^1}$] minimize $f(\mathbf{k})$

subject to $\mathbf{\Pi}^{-1/2} \mathbf{N} \mathbf{\Pi}^{-1/2} \succeq \mathbf{I} - \mathbf{qq}^T, \;\; \mathbf{k} \geq \mathbf{0}$.

Denote the optimized vector of rates by $\mathbf{k}^*$. If constraint (5.11) is used, then we can achieve the maximum total flux by multiplying $\mathbf{k}^*$ by $c_{tot}/f_{tot}(\mathbf{k}^*)$. If constraint (5.12) is used, we can achieve the maximum flux for each edge by dividing $\mathbf{k}^*$ by $f_{ind}(\mathbf{k}^*)$.

Suppose that $\mathcal{G}$ is a strongly connected, but not necessarily fully connected, graph with bidirectional edges for which the two edges between each pair of adjacent tasks have equal flux capacities. For example, robots may travel between sites along

identical parallel roads, similar to a two-way highway. Then by condition (3.18), the Markov process on $\mathcal{G}$ is reversible. We adapt the problem of maximizing the asymptotic rate of convergence to this special case and call it **Problem P$_\mathbf{L}^\mathbf{1R}$**.

For constraint (5.11): Condition (3.18) implies that $\mathbf{K\Pi} = \mathbf{\Pi K}^T$, so $\mathbf{N} = \mathbf{K\Pi}$ in Equation (5.13). Substitute $\mathbf{K\Pi}$ for $\mathbf{N}$ in Problem P1 (with $f = f_{tot}$). Since $\mathbf{K} = \mathbf{N\Pi}^{-1}$, $\mathbf{K}$ is similar to $\mathbf{S}$, so the constraint $\lambda_2(\mathbf{S}) \geq 1$ becomes $\lambda_2(\mathbf{K}) \geq 1$. Thus, the problem constrains $Re(\lambda_2(\mathbf{K}))$ directly instead of a lower bound on this value.

For constraint (5.12): We can maximize *all* the nonzero eigenvalues of $\mathbf{K}$ by setting each transition rate to its maximum value subject to condition (3.18) and constraint (5.12):

$$k_{ij} = (1/x_i^d) \min(c_{ij}, c_{ji}) , \quad (i,j) \in \mathcal{E} .$$

This is evident by using the Courant-Fischer min-max theorem [66] to express each nonzero eigenvalue of $\mathbf{S}$, and therefore of $\mathbf{K}$, in terms of a quadratic form $\mathbf{x}^*\mathbf{Sx}$ ($\mathbf{x}^*$ is the conjugate transpose of $\mathbf{x}$), which is equal to

$$\sum_{(i,j)\in\mathcal{E}} k_{ij} x_i^d a_{ij} \bar{a}_{ij} , \quad a_{ij} = x_i(x_i^d)^{-1/2} - x_j(x_j^d)^{-1/2} ,$$

where $\bar{a}_{ij}$ is the complex conjugate of a_{ij}.

We investigated the effect of the connectivity of $\mathcal{G}$ on $\lambda_2(\mathbf{K})$ for several strongly connected, directed graphs on three tasks, labeled in Figure 5.2. We used Problem P$_\mathbf{L}^\mathbf{1R}$ to compute $\mathbf{K}$ for graph α with condition (3.18) and Problem P$_\mathbf{L}^1$ to compute $\mathbf{K}$ for graph α without this condition and for all other graphs. We modeled each edge in a graph as providing one unit of equilibrium flux capacity by defining $c_{ij} = 1$ for all $(i,j) \in \mathcal{E}$ in constraint (5.12) and $c_{tot} = N_\mathcal{E}$ in constraint (5.11). The target distribution was $x_1^d = 0.2$, $x_2^d = 0.3$, $x_3^d = 0.5$.

Table 5.2 gives the resulting $\lambda_2(\mathbf{K})$ of each graph for both constraints, with column 2 indicating whether condition (3.18) was imposed. The fully connected
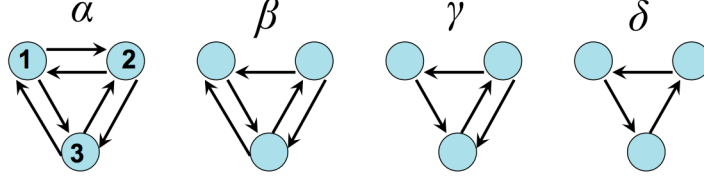
Figure 5.2: Graphs on three tasks

Table 5.2: Comparison of $\lambda_2(\mathbf{K})$ for graphs on three tasks

Graph	Rev.	$\lambda_2(\mathbf{K})$, constraint (5.11)	$\lambda_2(\mathbf{K})$, constraint (5.12)
α	yes	9.6774	7.7299
α	no	$9.6774 \pm 0.0026i$	7.7299
β	no	$8.0645 \pm 2.7936i$	$4.9588 \pm 1.6378i$
γ	no	$6.5729 \pm 2.9691i$	$4.6667 \pm 2.2111i$
δ	no	$5.1667 \pm 2.5766i$	$5.1667 \pm 2.5766i$

graph α yields the fastest convergence, which is expected since robots can switch from any task directly to any other task. Each removal of an edge from graph α lowers $\lambda_2(\mathbf{K})$, except in the case of constraint (5.12) applied to the 3-edge cycle δ. This is because the optimization problem maximized the equilibrium flux of *each* edge of graph δ (and did not for β and γ), which offset the stricter limits on task switching than in the other graphs.

B. Maximizing the overall convergence rate

The asymptotic rate of convergence only dictates the long-term system behavior. If $\mathcal{G}$ is fully connected and $\mathbf{x^0}$ is unknown, we can speed convergence of the faster modes by maximizing a measure of the overall convergence rate, which is a function of *all* the nonzero eigenvalues of $\mathbf{K}$, $\mathbf{\Lambda}(\mathbf{K}) = [\lambda_2(\mathbf{K}) \dots \lambda_S(\mathbf{K})]$. We define the quantity to be maximized as $\mathbf{1}^T\mathbf{\Lambda}$, which weights each eigenvalue equally. We use Equations (3.12) and (5.10) to write $\mathbf{k}$ as a linear function of $\mathbf{v} \equiv [\mathbf{\Lambda}(\mathbf{K}) \ \mathbf{0}]^T \in \mathbb{R}^{S^2-S}$. This allows us to formulate the optimization problem as a linear program with optimization variable $\mathbf{v}$ and objective function $\mathbf{1}^T\mathbf{v}$.

Let $\mathbf{K}$ be a matrix that satisfies Equation (3.12), which sets S constraints on the

S^2 entries of $\mathbf{K}$, and Equation (5.10), which sets $S - 1$ constraints. We now define the remaining $(S - 1)^2$ constraints on $\mathbf{K}$ in terms of the variable $\mathbf{\Lambda}(\mathbf{K})$. Since no extra constraints can be applied, no k_{ij} may be set to zero, which is why $\mathcal{G}$ must be fully connected.

Construct an orthonormal basis set in $\mathbb{R}^S$, $D = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{S-1}, \mathbf{x^d}/||\mathbf{x^d}||\}$. Define a matrix in $\mathbb{R}^{S \times S}$ as

$$\mathbf{A} \;=\; [\; \mathbf{d}_1 \; ... \; \mathbf{d}_{S-1} \; \mathbf{1} \;]^T \;\equiv\; [\; \tilde{\mathbf{A}}^T \;|\; \mathbf{1} \;]^T \;. \tag{5.19}$$

Since $\mathbf{1}^T \mathbf{x^d} = 1$ by Equation (3.20), $\mathbf{1}$ has a nonzero component in the direction of $\mathbf{x^d}$, so the rows of $\mathbf{A}$ are linearly independent. Thus, $\mathbf{A}$ is invertible. Let $\mathbf{B} = \mathbf{A}^{-1}$. Then

$$\mathbf{B} = \left[\; \tilde{\mathbf{A}}^T \;|\; \mathbf{x^d} \;\right] \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{1}^T \tilde{\mathbf{A}}^T & 1 \end{bmatrix} \equiv \left[\; \tilde{\mathbf{B}} \;|\; \mathbf{x^d} \;\right] \;.$$

Define $\mathbf{C} \in \mathbb{R}^{(S-1) \times (S-1)}$ as follows for some fixed $\tilde{\mathbf{A}}$:

$$\mathbf{C} = \tilde{\mathbf{A}} \mathbf{K} \tilde{\mathbf{B}} \;. \tag{5.20}$$

Also define $\hat{\mathbf{C}} \in \mathbb{R}^{S \times S}$ as $\mathbf{C}$ augmented with an added row of zeros and an added column of zeros.

Theorem 4. *A matrix $\mathbf{K}$ can be expressed as $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$ if and only if it satisfies Equations (3.12) and (5.10).*

Proof. $\mathbf{K}$ is similar to $\mathbf{P} \equiv \mathbf{M}\mathbf{K}\mathbf{N}$, where $\mathbf{M}, \mathbf{P} \in \mathbb{R}^{S \times S}$ and $\mathbf{N} = \mathbf{M}^{-1}$. Subdivide $\mathbf{M}$ as $[\tilde{\mathbf{M}}^T \;|\; \mathbf{m}]^T$ and $\mathbf{N}$ as $[\tilde{\mathbf{N}} \;|\; \mathbf{n}]$, where $\mathbf{m}, \mathbf{n} \in \mathbb{R}^{S \times 1}$. Then

$$\mathbf{M}\mathbf{N} = \begin{bmatrix} \tilde{\mathbf{M}}\tilde{\mathbf{N}} & \tilde{\mathbf{M}}\mathbf{n} \\ \mathbf{m}^T\tilde{\mathbf{N}} & \mathbf{m}^T\mathbf{n} \end{bmatrix} = \mathbf{I} \;, \tag{5.21}$$

$$\mathbf{M}\mathbf{K}\mathbf{N} = \begin{bmatrix} \tilde{\mathbf{M}}\mathbf{K}\tilde{\mathbf{N}} & \tilde{\mathbf{M}}\mathbf{K}\mathbf{n} \\ \mathbf{m}^T\mathbf{K}\tilde{\mathbf{N}} & \mathbf{m}^T\mathbf{K}\mathbf{n} \end{bmatrix} = \mathbf{P} \;. \tag{5.22}$$

Choose an $\mathbf{N}$ with $\mathbf{n} = \mathbf{x^d}$. It follows from Equation (5.21) that $\mathbf{m}^T\mathbf{x^d} = 1$, which by Equation (3.20) implies that $\mathbf{m} = \mathbf{1}$.

Suppose that $\mathbf{K}$ satisfies Equations (3.12) and (5.10). Since $\mathbf{m} = \mathbf{1}$ and $\mathbf{n} = \mathbf{x^d}$, these constraints applied to Equation (5.22) make the last row and last column of $\mathbf{P}$ both $\mathbf{0}$. To satisfy $\tilde{\mathbf{M}}\mathbf{n} = \tilde{\mathbf{M}}\mathbf{x^d} = \mathbf{0}$ in Equation (5.21), $\tilde{\mathbf{M}}$ can be set to $\tilde{\mathbf{A}}$. Then $\mathbf{M} = \mathbf{A}$, $\mathbf{N} = \mathbf{B}$, and $\mathbf{P} = \hat{\mathbf{C}}$, so it follows that $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$.

Now suppose that $\mathbf{K} = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}$. Since $\hat{\mathbf{C}}\mathbf{A}\mathbf{x^d} = \mathbf{0}$ and $\mathbf{1}^T\mathbf{B}\hat{\mathbf{C}} = \mathbf{0}$, $\mathbf{K}$ satisfies Equations (3.12) and (5.10). $\qquad\square$

From this result, $\mathbf{K}$ is similar to $\hat{\mathbf{C}}$, and so the eigenvalues of $\mathbf{C}$ are $\mathbf{\Lambda}(\mathbf{K})$. Thus, we can define $\mathbf{C}$ as:

$$\mathbf{C} \equiv diag(\mathbf{\Lambda}(\mathbf{K})) \ . \tag{5.23}$$

Now reformulate Equation (5.10) as $\mathbf{Fk} = \mathbf{0}$, where $\mathbf{F} \in \mathbb{R}^{S \times (S^2 - S)}$, and Equation (5.20) with $\mathbf{C}$ determined by (5.23) as $\mathbf{Gk} = \mathbf{g}$, where $\mathbf{G} \in \mathbb{R}^{(S-1)^2 \times (S^2 - S)}$ and $\mathbf{g} = [\mathbf{\Lambda}(\mathbf{K})\ \mathbf{0}]^T \in \mathbb{R}^{(S-1)^2}$. Define $\tilde{\mathbf{F}}$ as any $S - 1$ rows of $\mathbf{F}$. Then $\mathbf{k}$ can be written as

$$\mathbf{k} = [\mathbf{G}\ \ \tilde{\mathbf{F}}]^{-T}[\mathbf{g}^T\ \mathbf{0}]^T \equiv \mathbf{H}^{-1}\mathbf{v} \ . \tag{5.24}$$

Using definition (5.24) for $\mathbf{k}$, constraints (5.11) and (5.12) are

$$\mathbf{r}^T\mathbf{H}^{-1}\mathbf{v} \le c_{tot} \ , \quad \mathbf{H}^{-1}\mathbf{v} \le \mathbf{c} \ , \tag{5.25}$$

where the entries of $\mathbf{r} \in \mathbb{R}^{S^2 - S}$ are x_i^d and the entries of $\mathbf{c} \in \mathbb{R}^{S^2 - S}$ are c_{ij}/x_i^d. In addition, property (3.13) is

$$\mathbf{H}^{-1}\mathbf{v} \ge \mathbf{0} \ . \tag{5.26}$$

Note that while this property is not needed to prove Theorem 4, it is required to produce a valid $\mathbf{K}$. The optimization problem can now be posed as **Problem $\mathbf{P_L^2}$**.

[$\mathbf{P_L^2}$]   Maximize $\mathbf{1}^T\mathbf{v}$ subject to $\mathbf{v}_i = 0$ for $i = S, ..., S^2 - S$, Equation (5.26), and one of the constraints in (5.25).

C. Maximizing the convergence rate for a specified $\mathbf{x^0}$

If $\mathcal{G}$ is strongly connected, but not necessarily fully connected, and $\mathbf{x^0}$ is known, we can use a stochastic optimization method to *directly* minimize the time to converge from $\mathbf{x^0}$, quantified by t_f. We implement **Problem $\mathbf{P_L^3}$** below using a Monte Carlo method with $\mathbf{k}$ as the variable.

[$\mathbf{P_L^3}$] Minimize t_f subject to Equations (3.12), (3.13), (5.10), and constraint (5.11) or (5.12).

Implementation: At each iteration, $\mathbf{k}$ is perturbed by a random vector such that the resulting $\mathbf{K}$ matrix satisfies (3.12), (3.13), and (5.10). $\mathbf{k}$ is then scaled as in Problem P1 to satisfy constraint (5.11) or (5.12) while maximizing flux capacity. The resulting $\mathbf{K}$ is decomposed into its normalized eigenvectors and eigenvalues, system (3.19) is mapped into the space spanned by the normalized eigenvectors, and the appropriate transformation is applied to compute $\mathbf{x}(t)$ using $\exp(t \; diag([\mathbf{\Lambda}(\mathbf{K}) \; 0]))$. Since the system is stable by Theorem 1, $\mu(\mathbf{x})$ always decreases monotonically with time, so a Newton scheme can be used to calculate t_f. To compute $\hat{\mathbf{K}}$ for the linear chain model, this procedure is used with $\mathbf{z}$ in place of $\mathbf{x}$, $\mathbf{z^0} = [\mathbf{x^0}^T \; \mathbf{0}]^T$, and the target distribution $\mathbf{z^d}$ defined as the null space of $\hat{\mathbf{K}}$ at each iteration. The θ_{ij} are constants from the Erlang density (3.24).

To investigate the effect on the $\mathbf{k}$ optimization of accounting for task transition times, we implemented Problem P_L^3 for the baseline linear and linear chain models of a simple system with extreme variations in these times. $\mathcal{G}$ was defined as graph α in Figure 5.2, with an average transition time of 1000 for edge $(3, 2)$ and 10 for all other edges. All robots start at site 3, and the target distribution is $x_1^d = x_3^d = 0.05$, $x_2^d = 0.9$. The baseline model was assigned a relatively high k_{32} because sending robots along edge $(3, 2)$ is the most direct reallocation strategy. The chain model, which accounts for transition times, was assigned a k_{32} about 2×10^5 times lower than the baseline model value; most robots in this model switch from site 3 to site 1 and then site 2, avoiding the long route. A stochastic simulation was run using
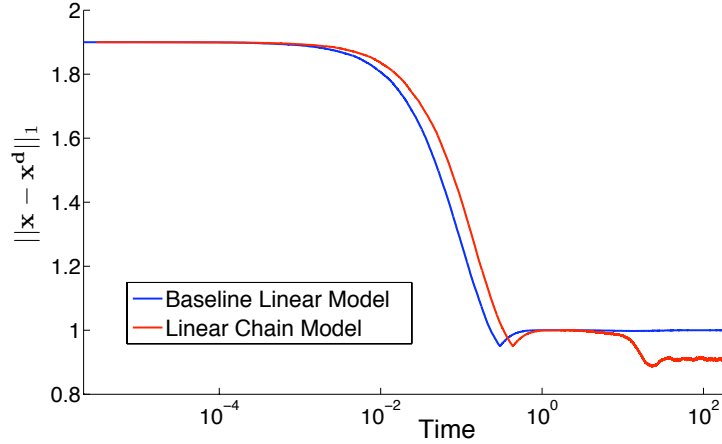
Figure 5.3: Fraction of misplaced robots for stochastic simulations using graph α in Figure 5.2 and $\mathbf{k}$ from the baseline linear and linear chain models, which are optimized using Problem $\mathrm{P}_{\mathrm{L}}^3$.

the method in Section 4.2 with 20000 robots and the $\mathbf{k}$ from both models. Figure 5.3 shows that the $\mathbf{k}$ from the chain model produces a more efficient system since it yields a lower fraction of misplaced robots at equilibrium. The system using the $\mathbf{k}$ from the baseline model cannot maintain many robots at site 2 because most are in transit on the long route $(3, 2)$ and leave site 2 relatively quickly.

If $\mathcal{G}$ is fully connected and $\mathbf{x^0}$ is known, then $\mathbf{K}$ can be computed such that $\Delta \equiv \mathbf{x^d} - \mathbf{x^0}$ is one of its eigenvectors with eigenvalue $\lambda > 0$. By maximizing λ, we maximize the convergence rate along the vector from $\mathbf{x^0}$ to $\mathbf{x^d}$, the most direct route in $\mathbb{R}^S$ to the target distribution. We use the decomposition of $\mathbf{K}$ from Theorem 4 to formulate the optimization problem as a linear program that maximizes λ.

Theorem 5. *Let $\mathbf{K}$ be a matrix that satisfies Equations (3.12) and (5.10); then by Theorem 4, $\mathbf{K} = \mathbf{B\hat{C}A}$. Let $\mathbf{d}_1 = \mathbf{d}$ in definition (5.19), where*

$$\mathbf{d} = \Delta'/||\Delta'||, \quad \Delta' = \Delta - \left(\mathbf{x^d}^T\Delta/||\mathbf{x^d}||^2\right)\mathbf{x^d} . \tag{5.27}$$

Then $\mathbf{K}\Delta = \lambda\Delta$ if and only if $\mathbf{C}$ from (5.20) is defined as

$$\mathbf{C} = [\ \mathbf{c}\ |\ \tilde{\mathbf{C}}\ ], \quad \mathbf{c}^T = [\lambda\ \mathbf{0}]\ , \tag{5.28}$$

71

where λ and $\tilde{\mathbf{C}}$ are unconstrained.

Proof. Suppose that $\mathbf{K}\Delta = \lambda\Delta$. Then

$$\mathbf{K}\Delta = \mathbf{B}\hat{\mathbf{C}}\mathbf{A}\Delta = \lambda\Delta \quad \Rightarrow \quad \hat{\mathbf{C}}\mathbf{A}\Delta = \lambda\mathbf{A}\Delta \ . \tag{5.29}$$

Using Equation (5.27) for $\mathbf{d}_1$ and the orthonormality of the $\mathbf{d}_i$:

$$\mathbf{d}_i^T\Delta = ||\Delta'||\mathbf{d}_i^T\mathbf{d}_1 + \left(\mathbf{x}^{\mathbf{d}^T}\Delta/||\mathbf{x}^{\mathbf{d}}||^2\right)\mathbf{d}_i^T\mathbf{x}^{\mathbf{d}} = 0 \tag{5.30}$$

for $i = 2, ..., S - 1$. From this equation and the fact that $\mathbf{1}^T\Delta = 0$ by constraint (3.20), $\mathbf{A}\Delta = [\ \mathbf{d}_1^T\Delta\ |\ \mathbf{0}\ ]^T$. Thus, Equation (5.29) is true if and only if $\mathbf{C}$ is defined as in (5.28). $\qquad\square$

We can now pose the optimization problem as **Problem $\mathbf{P_L^4}$**, in which property (3.13) and constraints (5.11) and (5.12) are defined in terms of the entries of $\mathbf{B}\hat{\mathbf{C}}\mathbf{A}$, with $\mathbf{d_1} = \mathbf{d}$ and $\mathbf{C}$ defined by (5.28). The optimization variables are λ and $\tilde{\mathbf{C}}$.

[$\mathbf{P_L^4}$] Maximize λ subject to Equation (3.13) and constraint (5.11) or (5.12).

5.2.2 Design of Vector Fields on Polytopes

The previous section discussed the optimization of the rate constants k_{ij} for a particular set of continuous dynamics defined by the multi-affine model (3.15) or the linear model (3.19). Now suppose that the macro-continuous model can be represented by a hybrid system, in which each population mode $\omega \in \Omega_p$ is associated with a different set of continuous dynamics. Recall that the systems we design must have a unique, stable equilibrium that represents the target distribution $\mathbf{x^d}$. Our control objective is to steer the trajectories originating in each mode through a sequence of modes that terminates in the mode containing this equilibrium. This section describes an approach to achieving this objective through the application of feedback control to the dynamics of each mode. We can also employ feedback control to place the target equilibrium inside the terminating mode (see the application in Section 6.3.1A), and

if the resulting dynamics of the mode are of the form (3.15) or (3.19), then we can optimize the k_{ij} using the methods in Section 5.2.1.

We consider the case in which each mode corresponds to a region of the continuous state space that is a full-dimensional polytope, $P_S \in \mathbb{R}^S$. The *implicit* description of P_S defines it as the intersection of a finite number of closed half-spaces. In this description, there is an integer $K \geq S + 1$, nonzero vectors $\mathbf{n}_1, ..., \mathbf{n}_K \in \mathbb{R}^S$, and scalars $\alpha_1, ..., \alpha_K$ such that [55]:

$$P_S = \{\mathbf{x} \in \mathbb{R}^S \mid \forall i = 1, ..., K : \mathbf{n}_i^T \mathbf{x} \leq \alpha_i\} . \tag{5.31}$$

The intersection of P_S with one of its supporting hyperplanes, $\{\mathbf{x} \in \mathbb{R}^S \mid \mathbf{n}_i^T \mathbf{x} = \alpha_i\}$, is called a *facet* F_i if the dimension of the intersection is $S - 1$. The vector $\mathbf{n}_i$ is the normal vector of F_i and, by convention, is of unit length and points out of P_S. A point $\mathbf{v}_j \in \mathbb{R}^S$, $j = 1, ..., M \geq S + 1$, is called a *vertex* of P_S if it cannot be written as a convex combination of the other points.

Given a time T_0, the system dynamics inside P_S are modified by a control input $\mathbf{u} : [0, T_0] \rightarrow U \subset \mathbb{R}^m$, where U is a polyhedral set, in the following way:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + G(\mathbf{x})\mathbf{u} , \tag{5.32}$$

where $f : P_S \rightarrow \mathbb{R}^S$ and $G : P_S \rightarrow \mathbb{R}^{S \times m}$. The objective is to calculate a function $\mathbf{u}$ that steers the state of system (5.32) to a facet of P_S, here assumed to be F_1, in finite time. This facet is the boundary between two adjacent polytopes that are associated with modes in the desired sequence. The velocity vector $\dot{\mathbf{x}}(T_0)$ must point out of P_S. The input $\mathbf{u}$ should be independent of the initial state, $\mathbf{x}(0)$, and take the form of a continuous feedback law, $\mathbf{u}(t) = g(\mathbf{x}(t))$, where $g : P_S \rightarrow U$.

Control of Affine Systems on Polytopes

Suppose that a mode is associated with linear dynamics (3.19) and the conservation constraint (3.20) is used to eliminate one variable; then $f(\mathbf{x})$ is defined by the affine

system (3.11). Let $G(\mathbf{x}) = \mathbf{B}$, a constant matrix. For this case, necessary and sufficient conditions for solvability of the control problem are given by [55]. Here we state the necessary conditions, which are used in an application in Section 6.3.1A. First, two index sets will be defined. For $i \in \{1, ..., K\}$, $V_i \subset \{1, ..., M\}$ is the index set of the vertices that belong to facet F_i. For $j \in \{1, ..., M\}$, the set $W_j \subset \{1, ..., K\}$ contains the indices of the facets connected to vertex $\mathbf{v}_j$. The necessary conditions consist of linear inequalities on the inputs at the polytope vertices, $\mathbf{u}_j = g(\mathbf{v}_j) \in U$, $j = 1, ..., M$ (Proposition 3.1 in [55]):

(1) $\forall j \in V_1$:

$$\mathbf{n}_1^T(\mathbf{A}\mathbf{v}_j + \mathbf{B}\mathbf{u}_j + \mathbf{a}) > 0 \tag{5.33}$$

$$\forall i \in W_j \backslash \{1\} : \mathbf{n}_i^T(\mathbf{A}\mathbf{v}_j + \mathbf{B}\mathbf{u}_j + \mathbf{a}) \leq 0 \tag{5.34}$$

(2) $\forall j \in 1, ..., M \backslash V_1$:

$$\forall i \in W_j : \mathbf{n}_i^T(\mathbf{A}\mathbf{v}_j + \mathbf{B}\mathbf{u}_j + \mathbf{a}) \leq 0 \tag{5.35}$$

$$\sum_{i \in W_j} \mathbf{n}_i^T(\mathbf{A}\mathbf{v}_j + \mathbf{B}\mathbf{u}_j + \mathbf{a}) < 0 \tag{5.36}$$

These inequalities describe a cone of possible velocity vectors at each of the polytope vertices. The sufficient conditions are given by Theorem 4.1 in [55] and assume that g is a Lipschitz-continuous function.

If the polytope is a *simplex*, a polytope in $\mathbb{R}^S$ with $S+1$ vertices and $S+1$ facets, then the necessary conditions are also sufficient. This results from the property that every point in a simplex can be described as a unique convex combination of its vertices. The input can be defined as the affine function

$$\mathbf{u} = \mathbf{F}\mathbf{x} + \mathbf{g} , \tag{5.37}$$

where $\mathbf{F} \in \mathbb{R}^{m \times S}$ and $\mathbf{g} \in \mathbb{R}^m$. This input produces an affine closed-loop system. If $\mathbf{u}_1, ..., \mathbf{u}_{S+1}$ can be found to satisfy the necessary conditions, then $\mathbf{F}$ and $\mathbf{g}$ can be calculated from a matrix equation that depends only on these input values and the vertices.

The construction of feedback controls on a general polytope is similar to the simplex case. A sufficient condition that effectively replaces necessary condition (5.36) is introduced; it is always satisfied if the inputs are unconstrained and if $\mathbf{B}$ is right invertible is introduced. Although any point in a general polytope can be expressed as a convex combination of the vertices, this choice is not necessarily unique. To construct a feedback law that remains continuous, a specific choice of vertices for each point must be fixed by triangulating the polytope. If $\mathbf{u}_1, ..., \mathbf{u}_M$ can be found that satisfy the necessary conditions and the sufficient condition, then the polytope is triangulated and $\mathbf{F}$ and $\mathbf{g}$ in feedback law (5.37) are computed for each simplex.

Control of Multi-Affine Systems on Rectangles

For control system (5.32), where f, G, and $\mathbf{u}$ are Lipschitz-continuous functions, the necessary and sufficient conditions for solvability of the control problem on a general polytope are identical to the conditions in [55] (substituting $f(\mathbf{x})$ for $\mathbf{Ax} + \mathbf{a}$ and $G(\mathbf{x})$ for $\mathbf{B}$) [7]. In [7], the sufficient conditions are adapted to the case where $f(\mathbf{x})$ is defined by the multi-affine function (3.9) and P_S is an S-dimensional rectangle. If $G(\mathbf{x}) = \mathbf{B}$ and there exist control inputs at all vertices of P_S that satisfy the sufficient conditions, then a continuous multi-affine feedback law $\mathbf{u}$ can be constructed as a convex combination of these inputs to solve the control problem. This feedback produces a multi-affine closed-loop system.

Chapter 6

Applications: Bio-Inspired Deployment and Multi-Site Surveillance

The work in this chapter was first presented in [11–16, 69].

In this chapter, the modeling, analysis, and controller synthesis methodologies are used to design stochastic control policies for robots to allocate themselves among a network of sites according to a target occupancy distribution for the swarm. The robots execute the transitions either independently of one another or using local sensor information, without relying on inter-robot communication.

One of our site allocation scenarios is inspired by a dynamical model of ant "house hunting," a decentralized process in which a colony of ants chooses a new nest from several candidate sites and emigrates there through quorum-dependent recruitment mechanisms. During the selection process, ants transition at experimentally measurable rates between simple behaviors that arise from local sensing and physical contact, and the pattern of transition rates ensures that the highest quality nest is usually chosen, with no ants stranded in a lower-quality nest. The quorum sensing

mechanism speeds up emigration to a site when it is sufficiently populated, a reflection of many individual decisions on the site's quality. The resulting group behavior is robust to environmental noise and to changes in colony population.

From a robotics perspective, an analogy can be drawn between the ants and robotic agents with limited communication and sensing capabilities that must distribute themselves and/or transport objects optimally among several locations. The features of ant house hunting suggest that a deployment model with quorum-based recruitment will produce a quick, robust distribution of resources. We develop an extension of the house hunting model in [42] to allow resource allocation throughout an arbitrary number of sites. Although we try to reflect ant behavior as accurately as possible, our goal is not to create a new description of ant house hunting, which has already been modeled in considerable detail [124], but rather to synthesize robot controllers that will produce ant-like activity. We implement this model for a scenario in which the robots and their transported items deploy to the better of two sites. We also add control terms to the original house hunting model that cause the swarm to split between two sites in a predefined ratio. Both models are hybrid systems because the swarm switches between different sets of behaviors based on the existence of a quorum.

We applied the task allocation approach discussed in earlier chapters to two scenarios in which each task is the surveillance of a particular site, and task transitions are effected by navigation between sites. In the first scenario, a moderate-sized swarm must redistribute among the perimeters of four buildings to achieve a target distribution as quickly as possible while adhering to a limit on inter-site traffic at equilibrium. Using the optimization methods from Section 5.2.1, we designed the robot control policies for two site connectivity graphs and both with and without knowledge of the initial robot distribution. We also emulated realistic inter-site travel time distributions by augmenting the baseline linear model of the system with virtual sites, representing the progress of traveling robots, to create a linear chain

model, and obtained control policies by optimizing this model. In the second scenario, a very large swarm reallocates to a target distribution among 42 sites by using quorum-dependent switching between maximum transition rates and rates from the baseline linear model. As in the ant-inspired models, this strategy is intended to speed up the allocation process using only local information and no communication.

6.1 Modeling

6.1.1 Micro-Continuous Model

Implementation

All micro-continuous models were implemented in C or Matlab, with the robots represented as point-mass agents governed by kinematic models. Gillespie's Direct Method, described in Section 4.2, is used to generate the sequence of robot transition events and their initiation times.

A. Bio-Inspired Deployment

To define tasks for the robot deployment scenario, we use the model of ant house hunting behavior presented in [42]. This model, constructed from experimental observations of *Temnothorax albipennis* ants, predicts the behavior of a colony of ants that is faced with a choice between two new nest sites, labeled 1 and 2, following the destruction of its original nest, site 0. Site 2 is a higher quality nest than site 1. A fraction p of the colony is actively involved in house hunting, and the remainder consists of brood items and other "passive" ants that must be carried to a new nest. The "active" ants perform the following tasks: they may be *naive* ants that search for a new site, *assessors* of site 1 or 2, or *recruiters* to site 1 or 2. Recruiters to site i bring ants from site 0 to i, and both their method of recruitment and their target recruitee depend on whether their population fraction has reached a *quorum value*, q. If the population of recruiters to site i is less than q, then the recruiters limit

themselves to using *tandem runs* to lead naive ants to assess site i. If the recruiter population exceeds q, then the recruiters use the faster method of *transport* to carry the passive ants at site 0 to site i.

In a robotics context, the active ants are analogous to robots that organize the distribution of resources or other robots, the "passive" items, among multiple sites. We will describe the house hunting model in this context from here on.

We define an extension of the house hunting model in which a swarm can distribute itself across an arbitrary number of sites and transfer passive items anywhere in the site network. Like the original model, this extended model includes realistic ant behaviors. Each robot has knowledge of at most two sites, one of which it considers its "home base." The assessing tasks are subdivided according to the home base of the robots, and recruiting tasks are subdivided to account for the recruiter's starting site. A recruiter starting at the destination site j performs tandem runs or transports depending on the entire population at j (not just the recruiter population); a recruiter starting at the source site i recruits via transports until it can determine the population at j. The home base of a recruiter starting at j changes from i to j when the population at j exceeds q. Assessors as well as naive robots may be recruited during tandem runs, and passive items may be transferred from any site to any other site. When recruiters to j find that there are no more passive items to transport from i, they "forget" this site and become naive robots at j. We reduce the extended house hunting model to the 3-site scenario of the original model.

In both models, each robot is represented as an entity that stores knowledge of its task, site 0, another site, position, speed, type of robot it is recruiting (in the extended model), and whether it is navigating to a site. We assume that the robots can estimate the population at each site using local sensing, possibly through their encounter rate with other robots at the site, which is how ants measure population [126]. When a transition is generated, a random robot in the appropriate task state that is not already *en route* to a site is selected to attempt recruitment or switch

tasks, either immediately or after traveling.

The robots are modeled at the scale of ants. Each of the three sites is represented as a circle of radius 0.02 m; a robot is considered inside the site once it enters the circle. For the purpose of navigation control, a robot's destination is defined as the center of a site circle. These destinations are 65 cm apart, the inter-site distance used in experiments to derive the site discovery and recruitment rates [125]. We consider an environment that lends itself to the construction of navigation functions, described in Section 5.1.1, and the abstraction of a circular boundary with three circular obstacles (see Figure 6.1). Robots performing tandem runs move at 1.5 mm/sec, while all other robots move at 4.6 mm/sec, the transport speed [42]. The transition rate units are $\min^{-1}$.

We note that aside from its specification of navigation controllers, the micro-continuous model is still a coarse-grained representation [96] since it abstracts away robot behaviors such as quorum estimation, recruiter-recruitee communication, and inter-robot collision avoidance. Thus, the model still requires more detail in order to constitute an executable robot controller. We point out that the quorum dependency does not pose a theoretical impediment to synthesizing such a controller. In the model, only the robots that visit a site know whether it has attained a quorum population. From the perspective of transition dynamics, a robot that has perceived a quorum is in a different state than a robot that has not, but the two robots are otherwise identical. Therefore, the quorum condition does not violate the Markov property of the model.

B. Multi-Site Surveillance

In this scenario, each task is defined as the surveillance of a particular site. We define two different environments: one with a few sites and a moderate-sized swarm to demonstrate the incorporation of realistic robot motion controllers, and one with many sites and a very large swarm to illustrate the scalability of our task allocation approach.
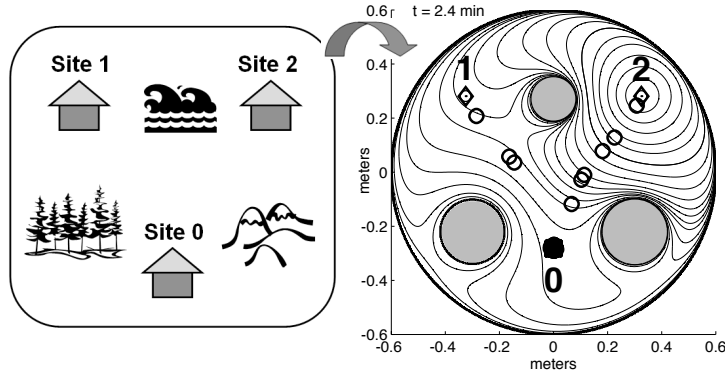
Figure 6.1: A three-site environment with obstacles (left) and the model with the contours of a navigation function (right) [○ = *naive robot*, ◇ = *assessor*].

In the first environment, the sites to be monitored are four buildings, numbered and highlighted in Figure 6.2, on the University of Pennsylvania campus. Two different site connectivity graphs $\mathcal{G}$, shown in Figure 6.3, are defined on these buildings. We assume that the robots can localize themselves on the campus and sense neighboring robots. Robots that are monitoring a building i circulate around the building perimeter while maintaining their distance from the robot ahead of them. A transition from building i to building j is randomly assigned to one of these robots. The selected robot continues to track the perimeter of i until it reaches a point that is designated as the start of the route from i to j. The robot exits i at this point and navigates to j while avoiding collisions with other robots, and it begins tracking the perimeter of j at a designated entrance point. Figure 6.4 illustrates the integration of switching initiations, perimeter surveillance, and navigation in the simulation.

The robot sensing radius ρ was set to 46 m, which is within the capabilities of some laser rangefinders. The navigation speed v_n is 1.3 m/s, which is attainable by some mobile robots that are particularly suited to surveillance tasks, such as PatrolBot® and Seekur®. The perimeter surveillance speed v_p is 4.5 times slower.

The second environment is a grid of 42 sites whose interconnection topology is shown in Figure 6.5. We assume that robots can detect the robot population at
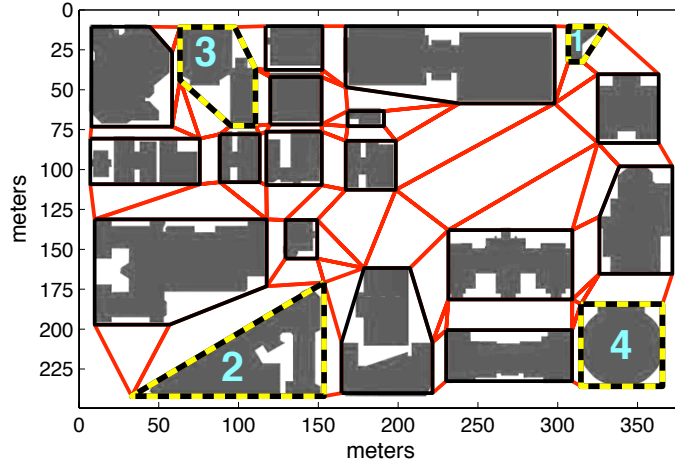
Figure 6.2: Campus map with cell decomposition of the free space used for navigation (see Section 6.3.2B).
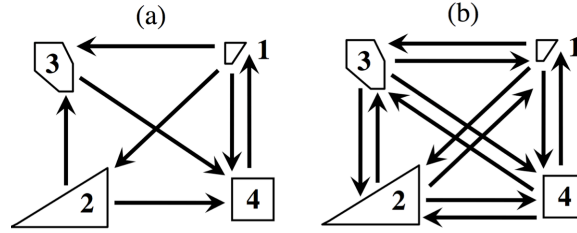
Figure 6.3: Numbering and connectivity of surveyed buildings for (a) a strongly connected but not fully connected graph; (b) a fully connected graph.
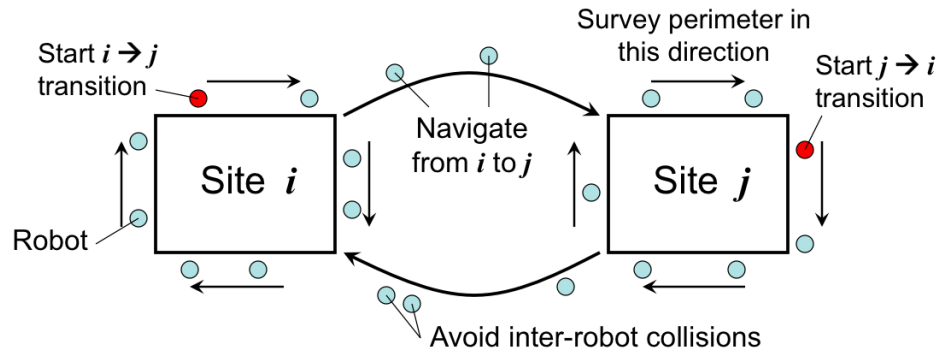
Figure 6.4: Robot activities in the surveillance simulation.
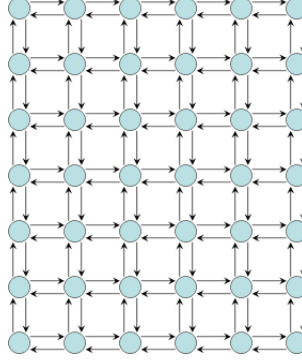
Figure 6.5: A graph of 42 sites with bidirectional edges.

their current site, and they use this information to determine the rate at which they switch to neighboring sites. Robots aggregate at sites and travel between them at a constant velocity; no collision avoidance behaviors are implemented.

CRN Descriptions

A. Bio-Inspired Deployment

We first describe the *original house hunting CRN* from [42]. A naive robot is symbolized by X, an assessor of site $i \in \{1, 2\}$ by Z_i, a recruiter to site $i \in \{1, 2\}$ by Y_i, and a passive item at site $i \in \{0, 1, 2\}$ by B_i. We use transition rates that were empirically determined from observations of ant colonies [125]. Naive robots discover site i at rate μ_i. Assessors become recruiters to site i at rate k_i, which is directly related to the quality of the site. λ_i and ϕ_i are the rates at which recruiters perform tandem runs and transports to site i, respectively. ρ_{ij} is the rate at which assessors and recruiters at site i encounter site j and switch their allegiance by becoming assessors of that site.

Each reaction that describes recruitment is switched on or off depending on whether there are recruitees available and whether the population at the destination site is above the quorum q. We define these switches using the unit step function U in Equation (3.32). In each recruitment reaction, the flux of recruited robots has the

form $\kappa y U(w) U(\Delta(q, y))$, where κ is a recruitment rate, w is the population fraction of recruitees, y is the population fraction of recruiters, and $\Delta(q, y)$ is either $q - y$ or $y - q$. To formulate recruitment as a reaction of type (3.7) with transition rate β and a recruitee as the reactant, we set the flux associated with such a reaction, βw, equal to $\kappa y U(w) U(\Delta(q, y))$ and solve for β. In this way, we define the transition rates β_i^r and β_i^t, which are associated with tandem runs and transports, respectively:

$$
\begin{aligned}
\beta_i^r &= \lambda_i y_i U(x) U(q - y_i)/x, \\
\beta_i^t &= \phi_i y_i U(b_0) U(y_i - q)/b_0, \quad i = 1, 2.
\end{aligned}
\tag{6.1}
$$

Now the original house hunting CRN can be written as:

$$
\begin{aligned}
X &\xrightarrow{\mu_1 + \beta_1^r} Z_1 & X &\xrightarrow{\mu_2 + \beta_2^r} Z_2 \\
Z_1 &\xrightarrow{k_1} Y_1 & Z_2 &\xrightarrow{k_2} Y_2 \\
Z_1 &\xrightarrow{\rho_{12}} Z_2 & Y_1 &\xrightarrow{\rho_{12}} Y_2 \\
B_0 &\xrightarrow{\beta_1^t} B_1 & B_0 &\xrightarrow{\beta_2^t} B_2
\end{aligned}
\tag{6.2}
$$
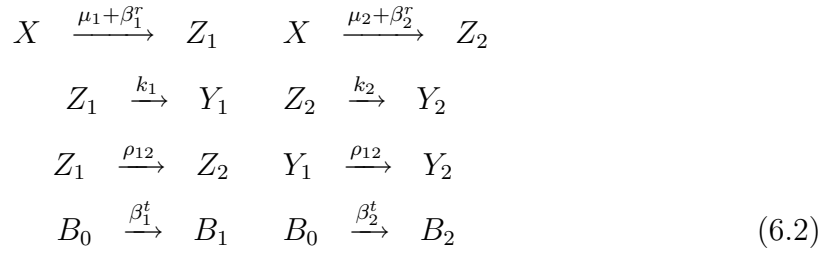
Note that transitions between the active robot tasks are unaffected by the passive item quantities.

In Section 6.3.1 we define a controlled house hunting model given by (6.22). The CRN corresponding to this model is the same as the original house hunting CRN, except that the transition rate in the $Y_1 \to Y_2$ reaction is $\rho_{12c} U(y_1 - q)$ and the reaction $Y_2 \to Y_1$ is added with transition rate $(\rho_{21c} + d/y_2) U(y_2 - q)$.

We now construct the *extended house hunting CRN* for a network of $M + 1$ sites. Y_i denotes a naive robot that considers site i its home base and leaves this site to search for a new site. Z_{ij} represents a robot that regards site i as its home base and is assessing site j. $Y_{ij,n}$ represents a robot that is located at site $n \in \{i, j\}$ and leaves to recruit other robots from i to j. B_i still denotes a passive item at site i.

Let y_i, z_{ij}, $y_{ij,n}$, and b_i be the robot population fractions corresponding to these

states. The population fraction at a site j is

$$p_j = y_j + b_j + \sum_{\substack{i=0 \\ i \neq j}}^{M} (y_{ij,j} + y_{ji,j} + z_{ij}) \; .$$

The transition rates associated with tandem runs and transports are derived in the same way as those in (6.1) and are defined as

$$
\begin{aligned}
\beta_{ij}^r &= \lambda_j y_{ij,j} U(y_i) U(q - p_j)/y_i \; , \\
\beta_{ijk}^r &= \lambda_k y_{jk,k} U(z_{ij}) U(q - p_k))/z_{ij} \; , \\
\beta_{ij}^t &= \phi_j y_{ij,j} U(b_i)(1 - U(q - p_j))/b_i + \phi_j y_{ij,i} U(b_i)/b_i \; , \\
& i, j, k \in \{0, ..., M\}.
\end{aligned}
\tag{6.3}
$$

All other transition rates are the same as in the original model (6.2).

The extended house hunting CRN is:

$$Y_i \xrightarrow{\mu_j + \beta_{ij}^r} Z_{ij}$$

$$Z_{ij} \xrightarrow{k_i} Y_{ji,j} \qquad Z_{ij} \xrightarrow{k_j} Y_{ij,j} \qquad Z_{ij} \xrightarrow{\rho_{jk} + \beta_{ijk}^r} Z_{ik}$$

$$Y_{ij,j} \xrightarrow{\rho_{jk} U(q - p_j)} Z_{ik} \qquad Y_{ij,j} \xrightarrow{\rho_{jk}(1 - U(q - p_j))} Z_{jk} \qquad Y_{ij,j} \xrightarrow{\phi_j(1 - U(q - p_j))U(b_i)} Y_j$$

$$Y_{ij,i} \xrightarrow{\phi_j U(b_i)} Y_{ij,j} \qquad Y_{ij,i} \xrightarrow{\phi_j(1 - U(b_i))} Y_j$$

$$B_i \xrightarrow{\beta_{ij}^t} B_j \tag{6.4}$$
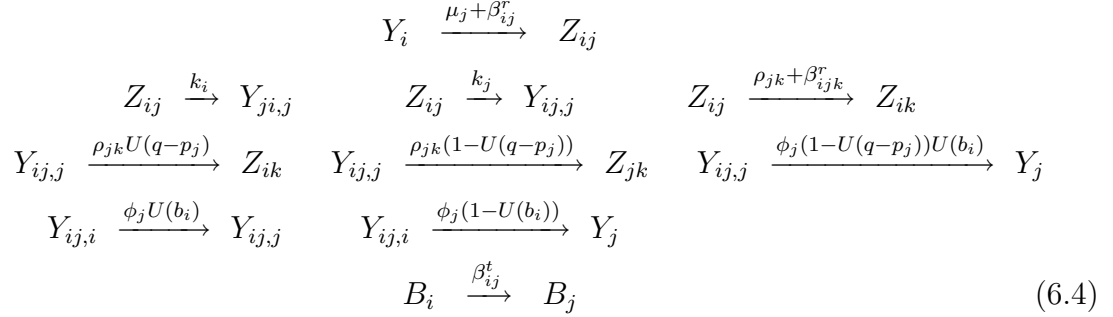
where $i, j, k \in \{0, ..., M\}$.

B. Multi-Site Surveillance

The CRN's corresponding to the surveillance scenarios each have a set of reactions of type (3.7) only, with the reaction pathways defined by the edges of the strongly connected graphs in Figures 6.3a,b and Figure 6.5. The transition rates k_{ij} in the 4-site environment are constant, and the rates in the 42-site environment obey the detailed balance condition (3.18) and are given by

$$k_{ij}^q = k_{ij} + U(x_i/x_i^d - q)(k^{max} - k_{ij}) \; , \tag{6.5}$$

where q is the quorum for each task.

6.1.2 Macro-Continuous Models

Bio-Inspired Deployment

For the original house hunting CRN, the species vector is $\mathbf{x} = [x\ y_1\ y_2\ z_1\ z_2\ b_0\ b_1\ b_2]^T$, where the entries are the population fractions of robots performing the different tasks. If we ignore the time that robots take to travel between sites, then this CRN can be abstracted to the following macro-continuous model:

$$
\begin{aligned}
\dot{x} &= -(\mu_1 + \mu_2)x - \lambda_1 y_1 U(x)U(q - y_1) - \lambda_2 y_2 U(x)U(q - y_2) \\
\dot{y}_1 &= k_1 z_1 - \rho_{12} y_1 \\
\dot{y}_2 &= k_2 z_2 + \rho_{12} y_1 \\
\dot{z}_1 &= \mu_1 x + \lambda_1 y_1 U(x)U(q - y_1) - \rho_{12} z_1 - k_1 z_1 \\
\dot{z}_2 &= \mu_2 x + \lambda_2 y_2 U(x)U(q - y_2) + \rho_{12} z_1 - k_2 z_2 \\
\dot{b}_0 &= -\phi_1 y_1 U(b_0)U(y_1 - q) - \phi_2 y_2 U(b_0)U(y_2 - q) \\
\dot{b}_1 &= \phi_1 y_1 U(b_0)U(y_1 - q) \\
\dot{b}_2 &= \phi_2 y_2 U(b_0)U(y_2 - q)
\end{aligned}
\tag{6.6}
$$

The model is subject to a conservation constraint on the active robot population:

$$
x + y_1 + y_2 + z_1 + z_2 = p \ .
\tag{6.7}
$$

Due to the unit step functions, which switch the terms describing recruitment on and off, model (6.6) is a hybrid system H_p. We consider only the five active robot state variables x, y_1, y_2, z_1, z_2, which are decoupled from the three passive robot state variables b_0, b_1, b_2. Since the active robot fraction p is constant, x can be eliminated through constraint (6.7). The system therefore evolves on the four-dimensional state space

$$
Y_p = \{\mathbf{x} \in \mathbb{R}^4 \mid \mathbf{x} \geq \mathbf{0},\ \mathbf{1}^T \mathbf{x} \leq p\} \ .
\tag{6.8}
$$

The state space is divided into four population modes by the hyperplanes $y_1 = q$

and $y_2 = q$. The set of modes is defined as $\Omega_p = \{NN, NQ, QN, QQ\}$, where

$$NN : \; y_1 < q \; , \; y_2 < q \qquad QN : \; y_1 \geq q \; , \; y_2 < q$$

$$NQ : \; y_1 < q \; , \; y_2 \geq q \qquad QQ : \; y_1 \geq q \; , \; y_2 \geq q \; . \tag{6.9}$$

When Equation (6.7) is used to replace x with $p - y_1 - y_2 - z_1 - z_2$ in model (6.6), the dynamics of each mode can be written as an affine model (3.11), where $\mathbf{x} = [y_1 \; y_2 \; z_1 \; z_2]^T$, $\mathbf{A} \in \mathbb{R}^{4 \times 4}$, and $\mathbf{a} \in \mathbb{R}^4$.

The entries of the species vector $\mathbf{x}$ for the extended house hunting CRN are y_i, z_{ij}, $y_{ij,i}$, $y_{ij,j}$, and b_i, where $i, j, k \in \{0, ..., M\}$. To illustrate the inclusion of inter-site robot travel times in the ODE abstraction, we write the corresponding macro-continuous model as a set of delay differential equations, as described in Section 3.3.2. The state variables in the model represent population fractions that are physically located at one of the $M + 1$ sites. Each time delay τ_{ij} is estimated as the average of a set of times, obtained from the simulation of the micro-continuous model, that robots take to travel from site i to site j. If i and j are written in bold, unitalicized font in a delay τ_{ij}, then the trip is a tandem run; otherwise, it is a transport or a solitary journey (which is conducted at the speed of a transport). The measured time delays are $\tau_{\mathbf{01}} = \tau_{\mathbf{02}} = 6$ min, $\tau_{01} = \tau_{02} = 2.2$ min, $\tau_{10} = \tau_{20} = 2.5$ min, $\tau_{\mathbf{12}} = \tau_{\mathbf{21}} = 7.84$ min, and $\tau_{12} = \tau_{21} = 2.48$ min.

We define $\tau_{ji+ij} \equiv \tau_{ji} + \tau_{ij}$, $x \equiv x(t)$ for each population fraction x, and $x[\tau_{ij}] \equiv x(t - \tau_{ij})$. We also define n_{tand} as the number of types of robots Y_i and $Z_{k,i}$ that a recruiter robot $Y_{ij,j}$ can lead in a tandem run $(k \neq i, j)$. Now the macro-continuous model corresponding to the extended house hunting CRN can be written as:

87

$$
\begin{aligned}
\dot{y}_i \;=\;& \sum_{\substack{j=0\\j\neq i}}^{M}[\phi_i(1-U(q-p_i[\tau_{ij+ji}]))(1-U(b_j[\tau_{ji}]))y_{ji,i}[\tau_{ij+ji}]\\
+\;& \phi_i(1-U(b_j[\tau_{ji}]))y_{ji,j} - \sum_{\substack{j=0\\j\neq i}}^{M}[\lambda_j U(q-p_j[\tau_{ji}])U(y_i)y_{ij,j}[\tau_{ji}] + \mu_j y_i]\\[4pt]
\dot{z}_{ij} \;=\;& \mu_j y_i[\tau_{ij}] - (k_i+k_j)z_{ij} + \sum_{\substack{k=0\\k\neq i,j}}^{M}[\rho_{kj}z_{ik}[\tau_{kj}] - \rho_{jk}z_{ij}]\\
+\;& \sum_{\substack{k=0\\k\neq i,j}}^{M}[\rho_{ij}(1-U(q-p_i[\tau_{ij}]))y_{ki,i}[\tau_{ij}] + \rho_{kj}U(q-p_k[\tau_{kj}])y_{ik,k}[\tau_{kj}]]\\
+\;& \sum_{\substack{k=0\\k\neq i,j}}^{M}[\lambda_j U(q-p_j[\tau_{jk+\mathbf{kj}}])U(z_{ik}[\tau_{\mathbf{kj}}])y_{kj,j}[\tau_{jk+\mathbf{kj}}]\\
-\;& \lambda_k U(q-p_k[\tau_{kj}])U(z_{ij})y_{jk,k}[\tau_{kj}]] + \lambda_j U(q-p_j[\tau_{ji+\mathbf{ij}}])U(y_i[\tau_{\mathbf{ij}}])y_{ij,j}[\tau_{ji+\mathbf{ij}}]\\[4pt]
\dot{y}_{ij,i} \;=\;& k_j z_{ji} - \phi_j y_{ij,i}\\[4pt]
\dot{y}_{ij,j} \;=\;& k_j z_{ij} + n_{tand}[-\lambda_j U(q-p_j)y_{ij,j} + \lambda_j U(q-p_j[\tau_{ji+\mathbf{ij}}])y_{ij,j}[\tau_{ji+\mathbf{ij}}]]\\
-\;& \phi_j(1-U(q-p_j))y_{ij,j} + \phi_j(1-U(q-p_j[\tau_{ji+ij}]))U(b_i[\tau_{ij}])y_{ij,j}[\tau_{ji+ij}]\\
+\;& \phi_j U(b_i[\tau_{ij}])y_{ij,i}[\tau_{ij}] - \sum_{\substack{k=0\\k\neq i,j}}^{M}\rho_{jk}y_{ij,j}\\[4pt]
\dot{b}_i \;=\;& \sum_{\substack{j=0\\j\neq i}}^{M}[\phi_i(1-U(q-p_i[\tau_{ij+ji}])U(b_j[\tau_{ji}])y_{ji,i}[\tau_{ij+ji}] + \phi_i U(b_j[\tau_{ji}])y_{ji,j}[\tau_{ji}]]\\
-\;& \sum_{\substack{j=0\\j\neq i}}^{M}[\phi_j(1-U(q-p_j[\tau_{ji}])U(b_i)y_{ij,j}[\tau_{ji}] + \phi_j U(b_i)y_{ij,i}] \qquad (6.10)
\end{aligned}
$$

Like model (6.6), the total population fraction of active robots is conserved, and the model can be represented as a hybrid system H_p due to the terms that include the unit step function.

Suppose that model (6.10) is formulated without time delays. When written in matrix form, both this model and model (6.6) can be viewed as switched linear systems described by Equation (3.31), in which each $\mathbf{K}_p$ is subject to constraint (3.12) but not constraint (3.13).

Multi-Site Surveillance

The two CRN's for the 4-site environment that correspond to the interconnection graphs in Figure 6.3 are each abstracted to a baseline linear model, Equation (3.19)

subject to (3.20). In order to investigate the effect of accounting for inter-site travel times, the CRN for the graph in Figure 6.3a is also abstracted to two versions of the linear chain model, Equation (3.26) subject to (3.27). To determine the parameters of a *full linear chain model* that would most accurately emulate the travel time distributions in the simulation of the micro-continuous model, we collected a set of $750 - 850$ τ_{ij} from the simulation for each edge (i, j), plotted a histogram of the τ_{ij}, and then fit an Erlang distribution (3.24) to the histogram to obtain the distribution parameters ω_{ij} and θ_{ij}. Figure 6.6 shows a sample fitting of an Erlang distribution to τ_{ij} data for one edge, and Table 6.1 lists $E(T_{ij})$ (the average τ_{ij}) and ω_{ij} for each edge. The optimized $\hat{\mathbf{K}}$ for this chain model is called $\hat{\mathbf{K}}_{full}$. For comparison, we also computed a optimized $\hat{\mathbf{K}}$, called $\hat{\mathbf{K}}_{one}$, for a *one-site linear chain model* in which each $\omega_{ij} = 1$ and each θ_{ij} is $1/E(T_{ij}) = \theta_{ij}/\omega_{ij}$ from the full chain model. In this case, the Erlang distribution reduces to an exponential distribution with the same mean value.

Note that each travel time τ_{ij} is measured as the the sum of τ_{ij}^a, the time for a robot to reach the exit on building i from the position at which it commits to the transition, and τ_{ij}^b, the travel time from the exit to building j's entrance. The robots at i are uniformly distributed around the perimeter (see Section 6.3.2B) and are randomly selected for transitions; therefore, τ_{ij}^a has a uniform distribution. The distribution of τ_{ij}^b is affected by the congestion on the roads and at the target sites, which determines the amount of time spent avoiding collisions.

The predictive value of the linear chain model depends on how well the travel time distributions are characterized. The effects of crowding, localization errors, collision avoidance, and quorum estimation can be readily incorporated into the linear ODE framework if we are able to model the distribution of the resulting delays.

The CRN that describes the 42-site environment is abstracted to a switched linear system (3.31), in which each $\mathbf{K}_p$ has the structure (3.14).
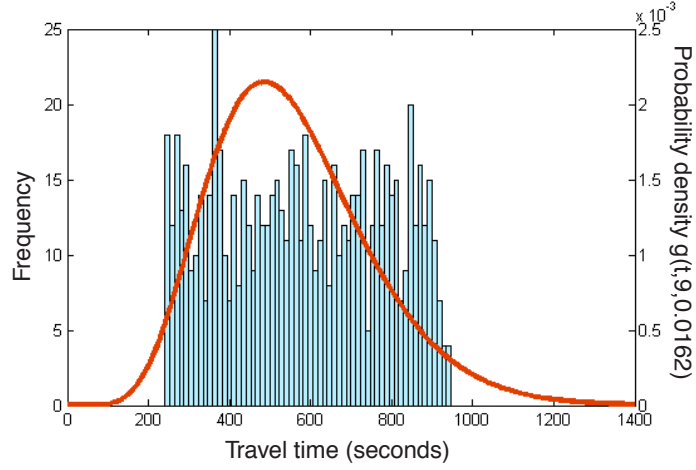
Figure 6.6: Histogram of the travel times from site 1 to site 4 (758 data points) and the approximate Erlang distribution.

Table 6.1: Data for Erlang distribution parameters

(i, j)	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)	(4,1)
$E(T_{ij})$	757	738	556	1507	1628	1228	1072
ω_{ij}	14	15	9	6	5	7	6

6.2 Analysis

6.2.1 Micro-Continuous and Macro-Discrete Models

We verify that for each environment, the macro-continuous model accurately predicts the system performance. To do this, we numerically integrate quantities in the macro-continuous model and compare these trajectories to those of averaged quantities in simulations of the micro-continuous and macro-discrete models. Gillespie's Direct Method, described in Section 4.2, was used to simulate the macro-discrete models and, as mentioned in Section 6.1.1, the task transitions and their times in the micro-continuous models.

For all transitions in the surveillance scenarios, the number of "product" robots in the state counter is incremented when the robot finishes navigating between sites. For transitions in the bio-inspired deployment scenario, the number of "product"

robots is incremented either immediately, as in the the transition from assessor to recruiter, or once the robot completes inter-site navigation.

Transitions that are enabled or disabled by unit step functions are implemented in a way that can be realized on robots that rely on local sensing and do not have information about robots at other sites. A transition that depends on the population at a site is enabled based on the population size relative to a quorum value, which as stated before can be measured by individual robots. In the bio-inspired deployment scenarios, a transition that depends on the recruitee population at a site is initiated independently of the recruitee availability, which is not known to a recruiter at another site. When a transition associated with recruitment is initiated, the number of robots in the appropriate recruiter state is decremented in the state counter to reflect the start of a tandem run or transport. If any recruitees are present when the recruiter arrives at their site, then their population is decremented in the counter. At the end of the recruiter's round-trip journey, the counter is updated to reflect the recruiter's success or failure at bringing another robot or passive item to the site.

For all environments, the simulation of the micro-continuous model is run in time steps Δt to implement the robots' incremental navigation through their environment. The completion of inter-site navigation is checked at the beginning of every time step, and a transition at time τ is initiated when $t \leq \tau \leq t + \Delta t$. In the macro-continuous and macro-discrete models, the time delays τ_{ij} due to inter-site navigation are measured from this simulation.

Figures 6.10, 6.12, and 6.16 show that the micro-continuous and macro-discrete trajectories match the macro-continuous model fairly closely. As the robot population approaches infinity, the standard deviations of ensembles of these trajectories should decrease to zero, as illustrated in Figure 6.13.

6.2.2 Macro-Continuous Model

A. Bio-Inspired Deployment

We apply reachability analysis to investigate certain properties of the original house hunting model. We consider the dynamics of the active robots only, so that the state space is defined by (6.8) with population modes (6.9). Since the dynamics of each mode in this model consist of the affine model (3.11), which is a special case of the multi-affine model (3.9), we can use the MARCO reachability algorithm described in Section 4.3.2.

First, we determine whether a quorum of recruiters at site 1 will ever be reached for a certain value of k_1, which reflects the quality of site 1, for the situation in which all robots start as naive. The initial set is the four-dimensional unit cube, and we set $N = 52$ robots and $qN = 10$, according to the values in [42]. Figure 6.7a shows the new reach set volume per iteration of the algorithm as a fraction of the total state space volume. The algorithm was set to terminate according to Proposition 6 with $\zeta = 0.05$. Figure 6.7b shows the projection of the reach set onto the $y_1 N - y_2 N$ plane. The curved black lines are the solutions of the macro-continuous model starting at the vertices of the initial set. From comparison with these solutions, the reachable set correctly predicts that site 1 will never achieve a quorum of 10 robots. The large reach set projection to the right of $y_1 N = 4$ resulted from defining some relatively large modes and from covering footprints with bounding boxes to reduce polyhedral complexity.

Second, we identified sets of initial conditions that guarantee that a particular site reaches a quorum before the other site. This analysis was conducted in mode NN, for which the state space is defined as $y_1, y_2 \in [0, 0.0481]$, $z_1, z_2 \in [0, 0.0721]$. This space was divided into modes of dimension $0.0120 \times 0.0120 \times 0.0144 \times 0.0144$ for refinement of the reachable set. Initial set A is defined as $y_1 \in [0.0337, 0.0385]$, $y_2 \in [0, 0.00481], z_1, z_2 \in [0.0288, 0.0337]$; initial set B is $y_1 \in [0, 0.00481]$, $y_2 \in$
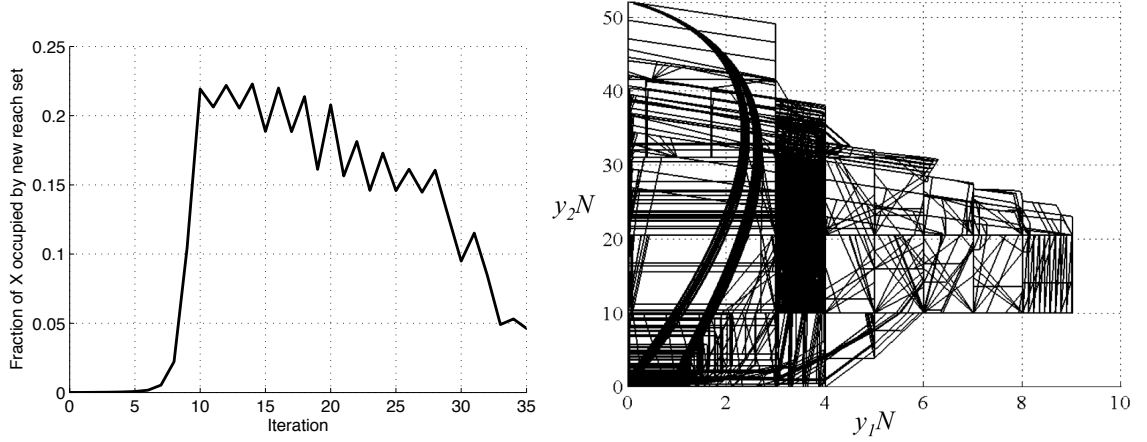
Figure 6.7: (a) Increase in reachable set volume at each iteration divided by state space volume as a function of the number of iterations. (b) Projection of 4-D reachable set for $k_1 = 0.0025$ (run time = 9251 sec).

$[0.0240, 0.0288], z_1, z_2 \in [0.0288, 0.0337]$.

In Figure 6.8, the unions of gray polygons are two-dimensional projections of the reachable set from each initial set. The computation took 33.5 minutes and consisted of 8 generations for box A and 22.3 minutes, 9 generations for box B. Each four-dimensional box has 16 vertices, which are projected onto the $y_1 - y_2$ plane. The black lines are the solutions of the macro-continuous model starting at these vertices. As shown by comparison with these solutions, both reachable sets correctly predict the first site to achieve a quorum of 0.0481. The reachability results show that all system trajectories starting inside box A and box B will first cross the quorum for site 1 and site 2, respectively. The algorithm guarantees this without computing any of the actual trajectories.

B. Multi-Site Surveillance

By Theorem 1 and Corollary 1, the baseline linear and linear chain models of the 4-site environment each converge to a unique, designable distribution of population fractions at each task starting from any initial distribution $\mathbf{x^0}$. We can arrive at the same result for the linear switched system model of the 42-site environment using

93
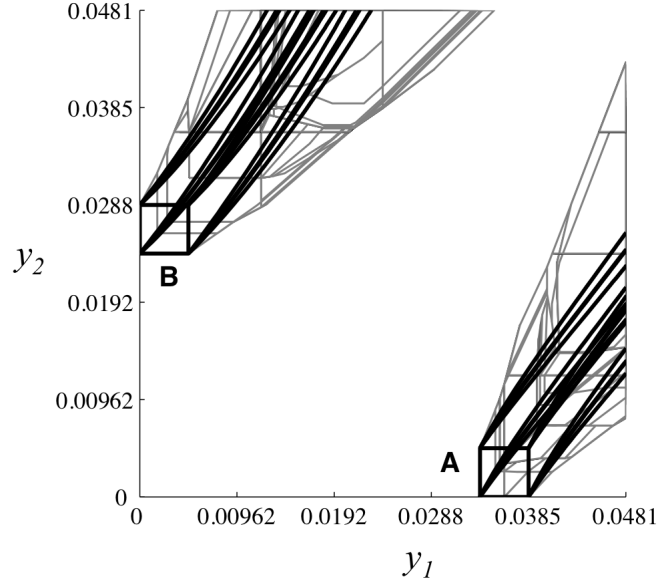
Figure 6.8: Two-dimensional projection of reachable sets; $p = 0.25$, $q = 0.0481$, $\mu_1 = \mu_2 = 0.013$, $\lambda_1 = \lambda_2 = 0.033$, $\rho_{12} = 0.004$, $k_1 = 0.019$, $k_2 = 0.020$ (values are from [42], [125]).

the analysis in [69], where this model is represented as a hybrid system with two modes, a *quorum* mode in which $x_i/x_i^d > q$ for some i and a *linear* mode in which $x_i/x_i^d < q$ $\forall i$.

6.3　Controller Synthesis

6.3.1　Macro-Continuous Model

A. Bio-Inspired Deployment

Using results from Section 5.2.2, we add control terms to the original house hunting model (6.6) to cause the swarm to split between two available sites at a *target occupancy ratio* $\alpha \geq 1$. We consider the active robots only and require that the system has one equilibrium at:

$$x = z_1 = z_2 = 0 \ , \quad y_1 = \frac{p}{1+\alpha} \ , \quad y_2 = \frac{\alpha p}{1+\alpha} \ , \tag{6.11}$$

94

Recall that the population modes of the system are defined by (6.9). We require that the equilibrium (6.11) is inside mode QQ; that is, $(1+\alpha)q \leq p \leq \alpha p$. The state space associated with this mode is defined by a four-dimensional simplex.

We first add controls to the dynamics of mode QQ so that it contains equilibrium (6.11) and no trajectories leave the mode. We redefine the dynamics of QQ, given by (3.11), as the control system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{a}$, where $\mathbf{u}$ is the affine feedback law (5.37). The controlled dynamics are thus:

$$\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{B}\mathbf{F})\mathbf{x} + (\mathbf{a} + \mathbf{B}\mathbf{g}) \ . \tag{6.12}$$

To enforce the conservation law (6.7), we dictate that the controls must result in a balance of terms among the differential equations. We only add controls to the recruiter dynamics, since the recruiter fractions alone determine the current mode and the steady state:

$$\begin{aligned} \dot{y}_1 &= k_1 z_1 - \rho_{12c} y_1 + \rho_{21c} y_2 + d \\ \dot{y}_2 &= k_2 z_2 + \rho_{12c} y_1 - \rho_{21c} y_2 - d \end{aligned} \tag{6.13}$$

Y_2 robots can now switch allegiance to Y_1 at rate ρ_{21c}. ρ_{12c} may differ from ρ_{12} in the original model, and d is a constant.

To ensure the desired equilibrium, we set $\dot{y}_1 = 0$, $\dot{y}_2 = 0$ and substitute the values in Equation (6.11) for the variables. This results in the equation:

$$(\alpha \rho_{21c} - \rho_{12c})\frac{p}{1+\alpha} + d = 0 \ . \tag{6.14}$$

To prevent trajectories from escaping mode QQ, we apply conditions (5.34) and (5.35) at the facets $F_1 = \{\mathbf{x} \in \mathbb{R}^4 \mid \mathbf{n}_1^T \mathbf{x} = -q\}$ and $F_2 = \{\mathbf{x} \in \mathbb{R}^4 \mid \mathbf{n}_2^T \mathbf{x} = -q\}$, where $\mathbf{n}_1 = [-1\ 0\ 0\ 0]^T$ and $\mathbf{n}_2 = [0\ -1\ 0\ 0]^T$ are the normal vectors of F_1 and F_2, respectively. The vertices of the simplex corresponding to mode QQ are:

$$\begin{aligned} \mathbf{v}_1 &= [q\ q\ 0\ 0]^T & \mathbf{v}_4 &= [q\ (p-q)\ 0\ 0]^T \\ \mathbf{v}_2 &= [q\ q\ (p-2q)\ 0]^T & \mathbf{v}_5 &= [(p-q)\ q\ 0\ 0]^T \\ \mathbf{v}_3 &= [q\ q\ 0\ (p-2q)]^T \end{aligned} \tag{6.15}$$

Setting $f(\mathbf{x}) \equiv \dot{\mathbf{x}}$ from Equation (6.12) and noting that $\dot{\mathbf{x}} = [\dot{y}_1 \; \dot{y}_2 \; \dot{z}_1 \; \dot{z}_2]^T$, the conditions to be satisfied are:

$$\mathbf{n}_1^T f(\mathbf{v}_i) \le 0, \quad i \in \{1, 2, 3, 4\} \quad \Rightarrow \quad \dot{y}_1 \ge 0 \tag{6.16}$$

$$\mathbf{n}_2^T f(\mathbf{v}_j) \le 0, \quad j \in \{1, 2, 3, 5\} \quad \Rightarrow \quad \dot{y}_2 \ge 0 \;, \tag{6.17}$$

where $\dot{y}_1$ and $\dot{y}_2$ are evaluated at the designated vertices using the equations in (6.13). The resulting set of inequalities is satisfied if conditions (6.16) and (6.17) are satisfied when evaluated only at vertex $\mathbf{v}_1$:

$$\dot{y}_1 \;=\; -\rho_{12c}q + \rho_{21c}q + d \ge 0 \tag{6.18}$$

$$\dot{y}_2 \;=\; \rho_{12c}q - \rho_{21c}q - d \ge 0 \tag{6.19}$$

$$\Rightarrow \quad d = (\rho_{12c} - \rho_{21c})q \;. \tag{6.20}$$

The relationship between ρ_{12c} and ρ_{21c} may be derived by substituting the expression for d from Equation (6.20) into Equation (6.14):

$$\frac{\rho_{12c}}{\rho_{21c}} = \frac{\alpha p - q(1 + \alpha)}{p - q(1 + \alpha)} \;. \tag{6.21}$$

We now modify the dynamics of the other three modes so that they contain no attractors and trajectories starting inside these modes follow the pattern: $NN \rightarrow NQ$ or QN, $QN \rightarrow QQ$, $NQ \rightarrow QQ$. We do this by replacing the recruiter dynamics in these modes with the equations in (6.13) with switches that prevent states from flowing in the $-y_1$ direction in modes NN, NQ and in the $-y_2$ direction in modes NN, QN.

The *controlled house hunting macro-continuous model* is thus defined as:

$$\dot{x}, \dot{z}_1, \dot{z}_2 \text{ from model (6.6)}$$

$$\dot{y}_1 \;=\; k_1 z_1 - \rho_{12c}y_1 U(y_1 - q) + (\rho_{21c}y_2 + d)U(y_2 - q)$$

$$\dot{y}_2 \;=\; k_2 z_2 + \rho_{12c}y_1 U(y_1 - q) - (\rho_{21c}y_2 + d)U(y_2 - q) \tag{6.22}$$

We replace ρ_{12} in the $\dot{z}_1$, $\dot{z}_2$ equations with ρ_{12c}.
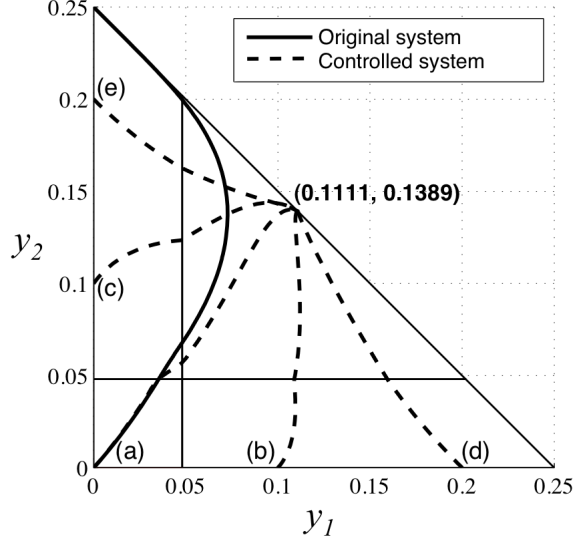
Figure 6.9: Trajectories of the original and controlled house-hunting models with $p = 0.25$, $q = 0.0481$, $\mu_1 = \mu_2 = 0.013$, $\lambda_1 = \lambda_2 = 0.033$, $k_1 = 0.019$, $k_2 = 0.020$ (values are from [42] [125]); $\alpha = 1.25$, $\rho_{12c} = 0.01$, $\rho_{21c} = 0.0069$, $d = 0.000147$. The dashed lines are trajectories beginning at (a) $[0\ 0\ 0\ 0]^T$, (b) $[0.1\ 0\ 0.1\ 0]^T$, (c) $[0\ 0.1\ 0\ 0.1]^T$, (d) $[0.2\ 0\ 0\ 0]^T$, and (e) $[0\ 0.2\ 0\ 0]^T$.

Figure 6.9 displays numerically integrated trajectories of models (6.6) and (6.22) on a 2-D projection of the state space. The thick solid line is the trajectory of model (6.6) beginning at $\mathbf{x} = [0\ 0\ 0\ 0]^T$. The dashed lines are sample trajectories of model (6.22) for $p = 0.25$, $\alpha = 1.25$. The figure shows that the original model converges to the equilibrium $[0\ 0.25\ 0\ 0]^T$, whereas the controlled model converges to the equilibrium (6.11).

B. Multi-Site Surveillance

For the 4-site environment, we compared the convergence of the micro-continuous model to a target distribution $\mathbf{x^d}$ for different sets of transition rates $\mathbf{k}$, each computed from one of the optimization problems in Section 5.2.1 using the baseline linear model. Problems P_L^1 and P_L^3 were used to compute rates for the system with graph Figure 6.3a, and Problems P_L^{1R}, P_L^2, P_L^3, and P_L^4 were used for the system with graph Figure 6.3b. To investigate the utility of the linear chain model in optimizing the

97

transition rates, we compared the performance of the micro-continuous model with $\mathbf{k}$ computed from Problem P_L^3 using the baseline linear model, the one-site linear chain model, and the full linear chain model.

For the 42-site environment, we compared the convergence of the micro-continuous model to $\mathbf{x^d}$ for three different $\mathbf{K}$: $\mathbf{K}_n$, a non-optimal $\mathbf{K}$ in the baseline linear model; $\mathbf{K}_o^{max}$, a $\mathbf{K}$ in the baseline model optimized by Problem P_L^3; and $\mathbf{K}_n^q$, which defines a linear switched system with rates given by (6.5), in which the k_{ij} are from $\mathbf{K}_n$, $k^{max} = 12$, and $q = 1.05$. $\mathbf{K}_o^{max}$ is subject solely to constraints $k_{ij} \leq k^{max}$ with no constraints on inter-site traffic at equilibrium; this means that it is the optimal $\mathbf{K}$ with respect to system convergence rate.

6.3.2 Micro-Continuous Model

A. Bio-Inspired Deployment

Navigation functions, described in Section 5.1.1, are used to direct robot travel between sites while preventing collisions with obstacles. The position of a robot k is updated at each time step by numerically integrating the equation

$$\dot{\mathbf{q}}_k = -\frac{v}{||\nabla \varphi_\kappa(\mathbf{q}_k, \mathbf{q}_k^d)||_2} \nabla \varphi_\kappa(\mathbf{q}_k, \mathbf{q}_k^d) \ , \tag{6.23}$$

where v is the robot's speed, $\mathbf{q}_k^d$ is its current destination site, and φ_κ is defined by Equation (5.2). The φ_κ of each robot share a common κ, which was selected empirically to be high enough to make φ_κ a valid navigation function and to eliminate local minima. Various combinations of v and $\mathbf{q}_k^d$ are used to produce different robot motion controllers; for example, one $\omega \in \Omega_r$ is navigating from site 0 to site 1 at the tandem-running speed.

B. Multi-Site Surveillance

This section describes the robot motion controllers for perimeter tracking, inter-site navigation, and robot collision avoidance in the 4-site environment.

Suppose that the boundary of a building m is parameterized by a vector $\mathbf{s}(s) \in \mathbb{R}^2$ that maps arc length s to (x, y) coordinates. A robot k monitoring the perimeter of m moves in the direction of a unit vector tangent to this boundary, $\hat{\mathbf{n}}_m(\mathbf{s}) \in \mathbb{R}^2$. To create an approximately uniform distribution of robots around the perimeter, we specify that the robot k slows down by a fraction ζ of its perimeter-tracking speed v_p if its distance q_{kl} from the robot l in front of it is less than p_m/N_m, where p_m is the perimeter length and N_m is the site population. The robot kinematics are then defined as

$$\dot{\mathbf{q}}_k = (1 - \sigma(q_{kl}, p_m, N_m)\zeta) \; v_p \; \hat{\mathbf{n}}_m(\mathbf{q}_k) \; ,$$

where $\sigma(q_{kl}, p_m, N_m) = 1$ if $q_{kl} < p_m/N_m$ and 0 otherwise.

To implement inter-site navigation, we use an approach similar to that described in Section 5.1.1. First, we decomposed the free space into a tessellation of convex cells, shown in Figure 6.2. Each edge (i, j) was defined as a sequence of cells to be traversed by robots moving from an exit point on the perimeter of building i to an entry point on the perimeter of j. Dijkstra's algorithm was used a priori to compute the sequence with the shortest cumulative distance between cell centroids, starting from the cell adjacent to the exit at i and ending at the cell adjacent to the entrance at j. The robots are provided with the cell sequence corresponding to each edge.

Define $\mathcal{N}_k$ as the set of robots within the sensing radius ρ of robot k and v_n as the navigation speed. The robot kinematics for navigation are

$$\dot{\mathbf{q}}_k = \frac{v_n}{||\mathbf{n}_g(\mathbf{q}_k) + \mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k)||_2} \; (\mathbf{n}_g(\mathbf{q}_k) + \mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k)) \; ,$$

where vector $\mathbf{n}_g(\mathbf{q}_k)$ is computed from local potential functions to ensure arrival at the goal cell and vector $\mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k)$ is computed from repulsive potential functions to achieve inter-robot collision avoidance, as described in Section 5.1.2.

Suppose that $\mathbf{q}_k$ is in cell c. Let $\hat{\mathbf{n}}_e^c$ be the unit vector pointing out of c orthogonal to its exit facet. Let $\hat{\mathbf{n}}_{f_1}^c$, $\hat{\mathbf{n}}_{f_2}^c$ be unit vectors pointing into c orthogonal to each facet adjacent to the exit facet, and define d_{k1}, d_{k2} as the distance from robot k to each

of these facets. Also define $\eta, \upsilon, \kappa > 0$. Then

$$\mathbf{n}_g(\mathbf{q}_k) = \eta \; \hat{\mathbf{n}}_e^c + \upsilon \; (1/d_{k1}^\kappa \; \hat{\mathbf{n}}_{f_1}^c + 1/d_{k2}^\kappa \; \hat{\mathbf{n}}_{f_2}^c) \; .$$

In the last cell in the sequence, this vector is replaced with one pointing from $\mathbf{q}_k$ to the perimeter entrance point.

Let $q_{kl} = ||\mathbf{q}_{kl}|| = ||\mathbf{q}_k - \mathbf{q}_l||$ and $\xi > 0$. Define a sum of vectors that point away from each neighboring robot,

$$\mathbf{n}_n(\mathbf{q}_k, \mathcal{N}_k) = \sum_{l \in \mathcal{N}_k} -\frac{1}{\xi^2 q_{kl}^2} \left(2 \ln (\xi q_{kl}) - \frac{1}{\xi q_{kl}} \right) \mathbf{q}_{kl} \; .$$

This is derived from the example potential function given in Figure 5.1, with the added parameter ξ that, when lowered, increases the range of repulsion between robots. Finally,

$$\mathbf{n}_a(\mathbf{q}_k, \mathcal{N}_k) = \frac{||\mathbf{n}_g(\mathbf{q}_k)||_2}{||\mathbf{n}_n(\mathbf{q}_k, \mathcal{N}_k)||_2} \mathbf{n}_n(\mathbf{q}_k, \mathcal{N}_k) \; .$$

6.4   Results

6.4.1   Bio-Inspired Deployment

We simulated the macro-continuous, macro-discrete, and micro-continuous models of the extended and controlled house hunting systems. The extended house hunting system is defined for the case $M = 2$, the 3-site scenario of the controlled system. All robots and passive items are initially located at site 0, and all active robots begin as naive.

Figure 6.10 shows the population fractions at sites 1 and 2 in all three models of the extended house hunting system. The robot population is $N = 832$. In all models, both sites achieve a quorum prior to 30 min and initially experience population growth. Site 2 outpaces site 1 in growth because robots commit to site 2 more quickly ($k_2 > k_1$) and are more willing to switch allegiance from site 1 to 2 than vice versa ($\rho_{12} > \rho_{21}$). By ~130 min, all passive items have been transported from
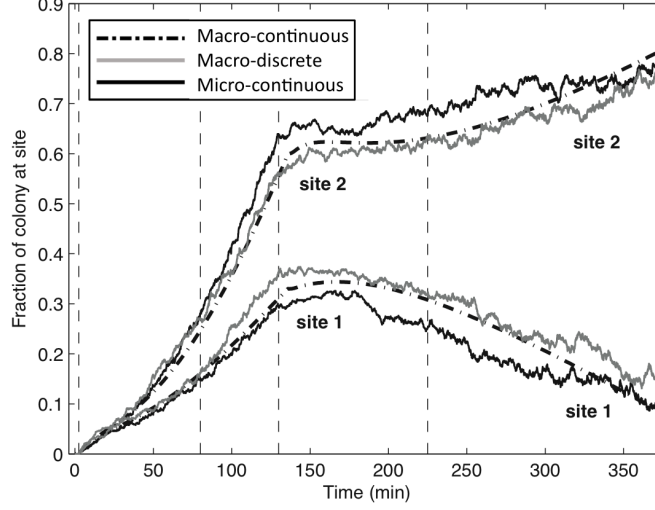
Figure 6.10: Population fractions at sites 1 and 2 in extended house hunting system; $p = 0.25$, $q = (10/208)N$, $\mu_1 = \mu_2 = 0.013$, $\lambda_1 = \lambda_2 = 0.033$, $\rho_{12} = 0.008$, $k_1 = 0.016$, $k_2 = 0.020$, $\phi_1 = \phi_2 = 0.099$ (values are from [42], [125]); $\rho_{21} = 0.002$, $\kappa = 2.7$. Dashed vertical lines correspond to the times of the snapshots in Figure 6.11.

site 0, and recruiters "forget" this site. The newly naive robots at site 1 or 2 repeat the process of finding, assessing, and recruiting to the other available site; however, now they can recruit *from* the site as well. Assessors at either site are more likely to recruit to the site of higher quality, which results in a net transport of passive items to site 2. By ~376 min in the macro-continuous and micro-continuous models, all passive items at site 1 have been removed to be reunited with those at site 2; only active robots remain at site 1. Due to stochastic fluctuations, some passive items still remain at site 1 in the macro-discrete model.

Figure 6.11 shows snapshots of the micro-continuous simulation at times indicated by the vertical lines in Figure 6.10. The curvature in the robot paths is due to the shape of the navigation functions, one of which is displayed in Figure 6.11a. The snapshots correspond to the initial searching and assessing phase (6.11a), the period of transport from site 0 (6.11b), the realization that site 0 contains no passive items (6.11c), and the period of transport between sites 1 and 2 (6.11d).

Figure 6.12 displays the recruiter fractions at sites 1 and 2 in all three models of

101
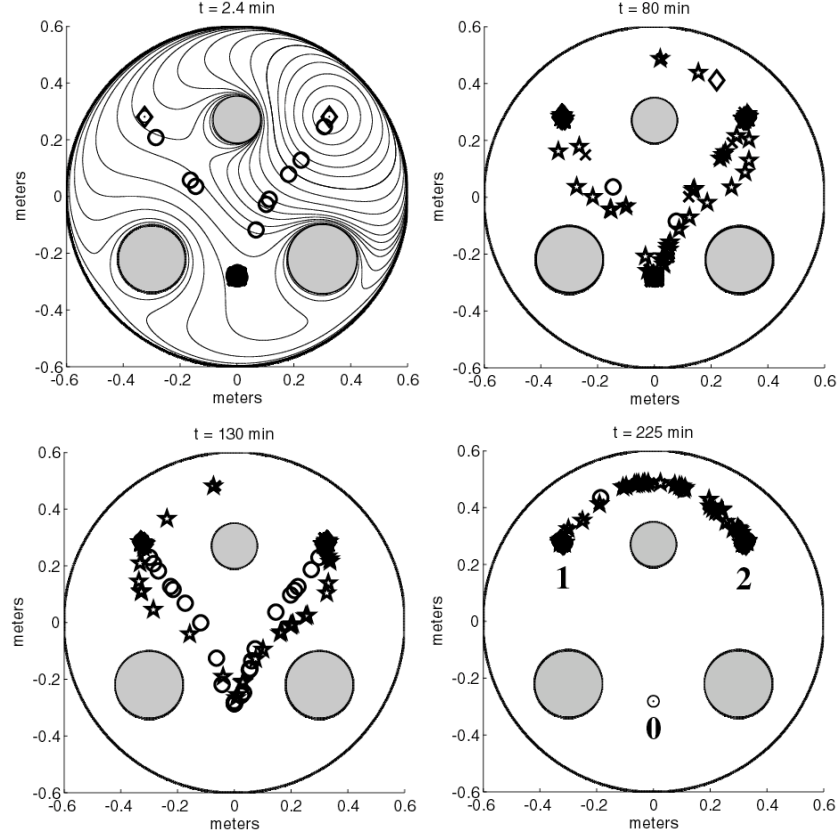
Figure 6.11: Micro-continuous simulation snapshots for extended house hunting system ($\bigcirc$ = *naive*; $\Diamond$ = *assessor*; $\bigstar$ = *recruiter*; $\times$ = *passive*) showing the swarm at (a) 2.4 min (top left); (b) 80 min (top right); (c) 130 min (bottom left); and (d) 225 min (bottom right). The navigation function that is used in a navigation controller (6.23) with $\mathbf{q}_k^d$ at site 2 is shown at the top left.

the controlled house hunting system. The target occupancy ratio is set to $\alpha = 1.25$ and the active robot population is $pN = 208$. At 700 min in the macro-continuous model, $y_1 = 0.1089$ and $y_2 = 0.1357$, which fall short of the equilibrium values in (6.11) by the fraction of recruiters that are traveling between sites 1 and 2. The final values of y_1 and y_2 in the other two models are close to these fractions. Thus, the equilibrium occupancy ratio in the macro-discrete and micro-continuous models approximates $\alpha = 1.25$.

To investigate the effect of robot population on the equilibrium recruiter fractions, macro-discrete simulations were run for active robot populations of 52, 208, and 832.

For each run, y_1 and y_2 were sampled at intervals as close as possible to 2 min from 1000 to 5000 min. Figure 6.13 shows the resulting frequency distributions of y_1 and y_2. The vertical lines near the centers of the distributions mark the macro-continuous equilibrium values. The mean and standard deviation of each variable are also recorded in the figure. The standard deviation decreases as the number of robots increases: it is less than 15% of the mean for a relatively modest population of 52 active robots and less than 4% for a larger but still realistic population of 832.

An interesting question is whether the features of ant house hunting actually provide an advantage in fulfilling the deployment task. For example, consider the simpler linear model, which does not rely on quorum sensing or switching between controllers but has the desired equilibrium fractions of x, y_1, and y_2 from (6.11):

$$\begin{aligned}
\dot{x} &= -(\mu_1 + \mu_2)x \\
\dot{y_1} &= \mu_1 x - \rho_{12c}y_1 + \rho_{21c}y_2 + d \\
\dot{y_2} &= \mu_2 x + \rho_{12c}y_1 - \rho_{21c}y_2 - d \ ,
\end{aligned} \tag{6.24}$$

where ρ_{12c}, ρ_{21c} and d satisfy Equation (6.14). It would be possible to use (6.24) as our macro-continuous model and synthesize robot behaviors from it using our methodology. Indeed, for the particular set of parameters used in our simulations, the difference between models (6.22) and (6.24) is not substantial. However, if the discovery rates μ_1 and μ_2 in both models are reduced by a factor of 10, the importance of tandem runs becomes apparent. Fig. 6.14 shows simulation results[1] of $z_i + y_i$, $i = 1, 2$, for the controlled model (6.22) and y_i, $i = 1, 2$, for the simple model (6.24) in this situation. The controlled model converges significantly faster to the target equilibrium. This illustrates a possible advantage of the strategy employed by ants over the more obvious linear model (6.24): the recruitment mechanisms can significantly speed up the deployment process to compensate for a low discovery rate arising from environmental constraints.

---

[1]Both models were simulated as delay differential equations with the previously defined time delays to include the effect of navigation.
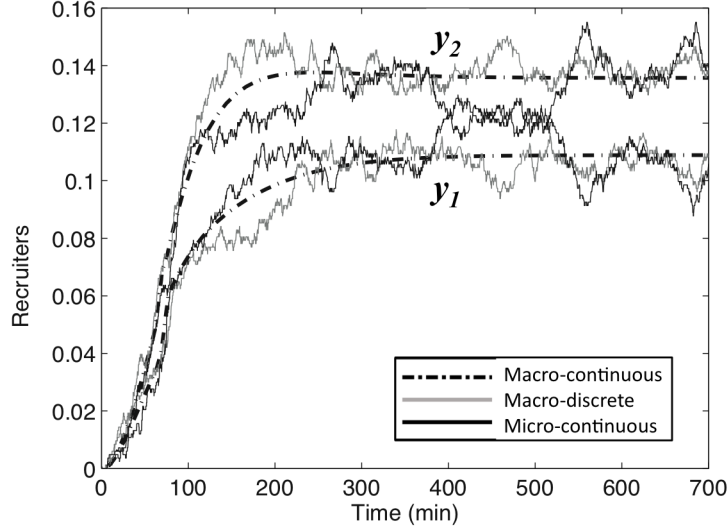
Figure 6.12: Recruiter fractions y_1, y_2 in controlled house hunting system; parameters are the same as in Figure 6.9.

6.4.2 Multi-Site Surveillance

4-Site Environment

We simulated the macro-continuous and micro-continuous models of the 4-site surveillance scenario. The swarm is initially split equally between buildings 3 and 4 in all simulations. In the optimization problems, the total equilibrium flux capacity c_{tot} for all possible edges (graph Fig. 6.3b) was set to 0.175 robots/s and distributed among the edges in proportion to the cumulative distance between the centroids of their associated cells.

In the comparison of the system with different optimized $\mathbf{k}$ from the baseline model, we used a population of 250 robots and the target distribution $x_1^d = 0.1$, $x_2^d = 0.4$, $x_3^d = 0.2$, and $x_4^d = 0.3$. The snapshots in Figure 6.15 illustrate the robot redistribution for one trial. In Figure 6.16, we compare performance in terms of $||\mathbf{x} - \mathbf{x^d}||_1$ for 40 simulation runs of the micro-continuous model and the macro-continuous DDE model (3.22) with the same $\mathbf{k}$. Each time delay τ_{ij} in the DDE model was estimated as the average of $750 - 850$ robot travel times at equilibrium

104
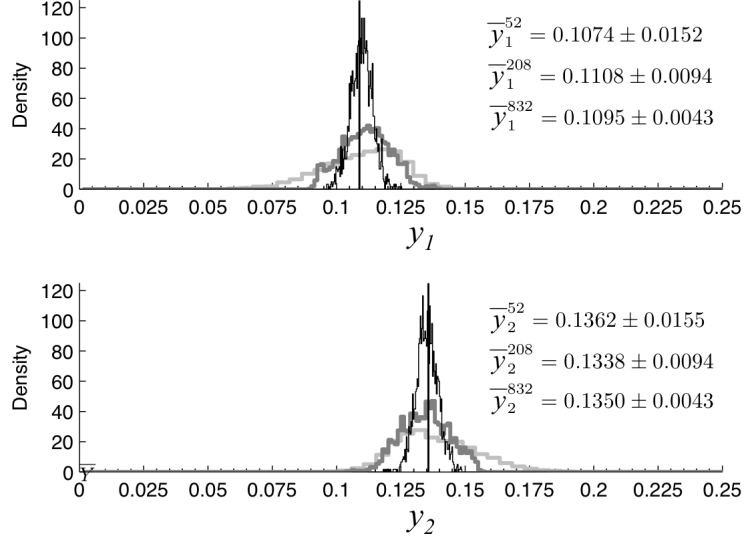
Figure 6.13: Histograms of equilibrium y_1 and y_2 in controlled house hunting system for $pN = 52$ (light gray), 208 (dark gray), and 832 (black), where $p = 0.25$. Bar width is $1/N$ for each value of N. Vertical lines mark $y_1 = 0.1089$ and $y_2 = 0.1357$.

from site i to j, collected from a micro-continuous simulation using site graph Fig. 6.3b. The simulation runs average to a plot that is close to the DDE trajectory and display little variability, even though the swarm is only of moderate size. We now discuss several key points from the micro-continuous simulation results.

Tradeoff between convergence rate and equilibrium traffic: Figures 6.17 and 6.18 compare system performance for different $\mathbf{k}$ in terms of the *distance from equilibrium*,

$$\nu(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{x^d}||_1 - \mathbf{1}^T\mathbf{y} . \tag{6.25}$$

This quantity decreases to zero at equilibrium because then the fraction of travelers, $\mathbf{1}^T\mathbf{y}$, accounts entirely for all the discrepancies $|x_i - x_i^d|$, $i = 1, ..., 4$. Each plot is an average over 40 micro-continuous simulation runs, and the bold numbers beside the legends are the average traveler fractions at equilibrium for each $\mathbf{k}$. (Standard deviations are not shown to avoid cluttering the figures; the maximum standard deviation over all plots is 0.078.) The data in these figures verify that there is a tradeoff between rapid convergence to equilibrium and the number of idle transitions

105
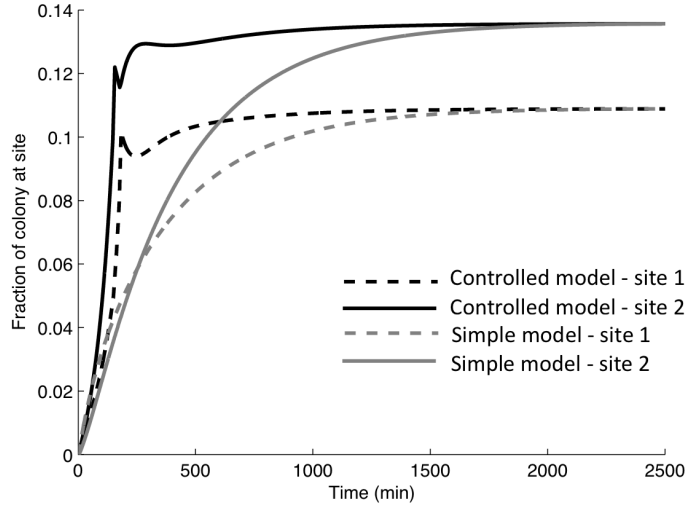
Figure 6.14: Population fractions at sites 1 and 2 in the controlled house hunting macro-continuous model and simple model (6.24); parameters are the same as in Figure 6.9 except for $\mu_1 = \mu_2 = 0.0013$.

between sites at equilibrium. For instance, the runs in Figure 6.17b are the slowest to converge, and they yield the lowest equilibrium traffic fractions. It is notable that this tradeoff can occur to different degrees depending on the flux constraint, (5.11) or (5.12). The Problem P_L^2 plot converges slightly faster in Figure 6.18b than in Figure 6.18a, but it has a *lower* equilibrium traffic fraction.

Faster convergence with increased site connectivity: Figures 6.17 and 6.18 show that for both flux constraints, (5.11) and (5.12), the runs for graph Figure 6.3b converge faster to equilibrium than those for graph Figure 6.3a. This is due to the difference in allowable pathways between the initial and final distributions. In Figure 6.3b, robots can travel directly from sites 3 and 4 to sites 1 and 2, while in Fig 6.3a, they can only reach sites 1 and 2 via the path $3 \to 4 \to 1 \to 2$, which prolongs the redistribution process. The greater number of edges in Figure 6.3b also reduces the impact on convergence of limiting each edge's flux capacity. The range of convergence times to equilibrium for Figure 6.3b are similar for both constraints, while the convergence times for Figure 6.3a increase significantly when constraint (5.12) is applied.

106
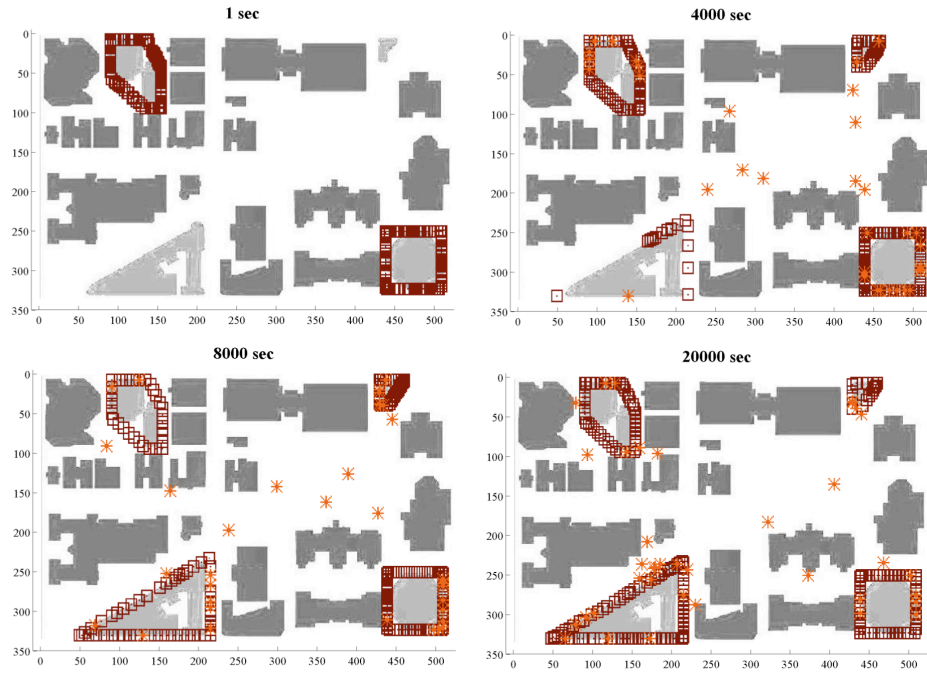
Figure 6.15: Snapshots of a run using **k** from Problem P_L^1 with constraint (5.12). The red robots (□) are not engaged in a transition; the orange robots (∗) have committed to travel to another site or are in the process of traveling.
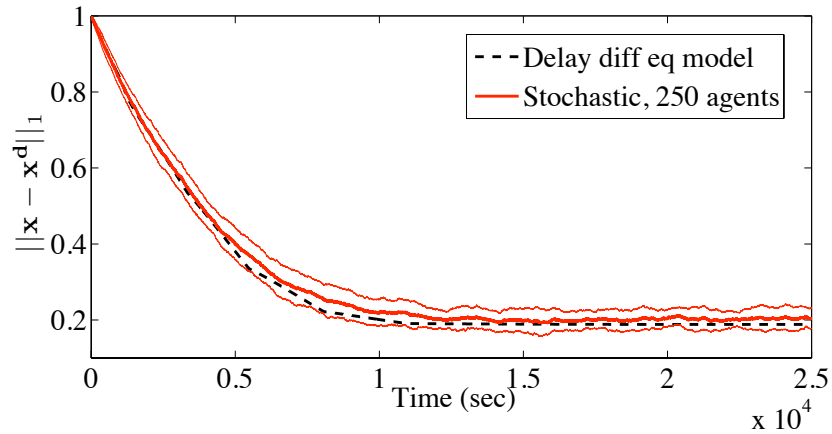
Figure 6.16: DDE macro-continuous model and micro-continuous simulations using **k** from Problem P_L^1 with constraint (5.12). Micro-continuous plots show the average over 40 runs ± standard deviation.

Limits on edge flux capacities eliminate the advantage of knowing $\mathbf{x^0}$*:* Since the $\mathbf{k}$ produced by Problems P_L^3 and P_L^4 are optimized for a specific $\mathbf{x^0}$, it seems likely that for any given $\mathbf{x^0}$ the Problem P_L^3 and Problem P_L^4 runs will converge at least as fast as the runs corresponding to other problems, which optimize $\mathbf{k}$ for the entire domain of $\mathbf{x^0}$. As Figures 6.17a and 6.18a show, this is true if constraint (5.11) is used. This is because the flux capacity can be distributed among edges in any way as long as total capacity does not exceed a limit. However, when constraint (5.12) is used, limits are placed on edges that, if left unconstrained, would be allocated a higher flux capacity to redistribute robots from $\mathbf{x^0}$ to $\mathbf{x^d}$. The problems that are independent of $\mathbf{x^0}$ are more robust to these limitations; their corresponding runs converge as fast as the runs that rely on $\mathbf{x^0}$ or outperform them.

$\mathbf{K}$ *from convex optimization is competitive compared to* $\mathbf{K}$ *from stochastic optimization:* The fastest-converging runs that use $\mathbf{k}$ from Problems P_L^1, P_L^{1R}, P_L^2, and P_L^4 attain equilibrium at least as quickly as the corresponding runs that use $\mathbf{k}$ from Problem P_L^3. Hence, we can use efficient convex optimization techniques to compute a $\mathbf{k}$ that yields the same system performance as a $\mathbf{k}$ from a much more time-consuming stochastic optimization approach.[2] This facilitates the quick computation of $\mathbf{k}$ in real-time task allocation scenarios.

In the comparison of the system with optimized $\mathbf{k}$ from both the baseline and chain models, we used a population of 240 robots and the target distribution $x_1^d = x_2^d = x_3^d = x_4^d = 0.25$. In Problem P_L^3, the same flux constraint (5.11) is applied to both the baseline and chain models so that the models have the same equilibrium traveler fraction, which is necessary to have a basis for comparing the system convergence rates due to the tradeoff between these properties that was discussed earlier.

---

[2]On a standard 2 GHz laptop, one Metropolis optimization run used for graph Figure 6.3b took about 15 minutes for $t_{0.1}$ to decrease slowly enough with each iteration for $\mathbf{K}$ to be deemed close enough to optimal, while all the convex optimization programs computed an optimal $\mathbf{K}$ in less than a second.
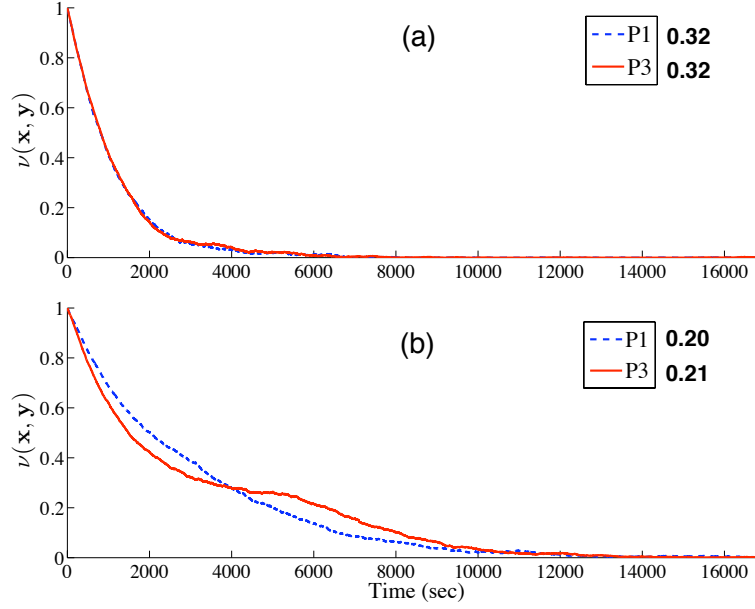
Figure 6.17: Distance from equilibrium for micro-continuous simulations using graph Figure 6.3a with (a) constraint (5.11) and (b) constraint (5.12). Each plot is an average over 40 runs that use **k** from the problem labeled in the legend. The bold number to the right of each legend entry is the equilibrium traveler fraction averaged over 1000 data points of the corresponding plot.

Figures 6.19(b),(d) show that starting at $\sim$ 15000 sec, each average traveler fraction over 40 micro-continuous simulation runs oscillates close to the average equilibrium value from the simulations using **K**, the optimized matrix in the baseline model. Thus, **K**, $\hat{\mathbf{K}}_{one}$, and $\hat{\mathbf{K}}_{full}$ yield approximately the same equilibrium inter-site traffic. Figures 6.19(a),(b) show that **K** and $\hat{\mathbf{K}}_{one}$ produce very similar trajectories for the average fraction of misplaced robots $\mu(\mathbf{x})$, average traveler fraction, and associated standard deviations. The same can be said of the results for **K** and $\hat{\mathbf{K}}_{full}$ in Figures 6.19(c),(d), since the relatively high standard deviations indicate that any disparities may not be significant.

This result arises from the fact that the average τ_{ij} for the edges are only within a factor of 3 of each other (see Table 6.1), as opposed to a factor of 100 in the scenario giving rise to Figure 5.3, so there is not much advantage in terms of convergence to rerouting robots away from a "long" route. Also, the average τ_{ij} is at most only

109
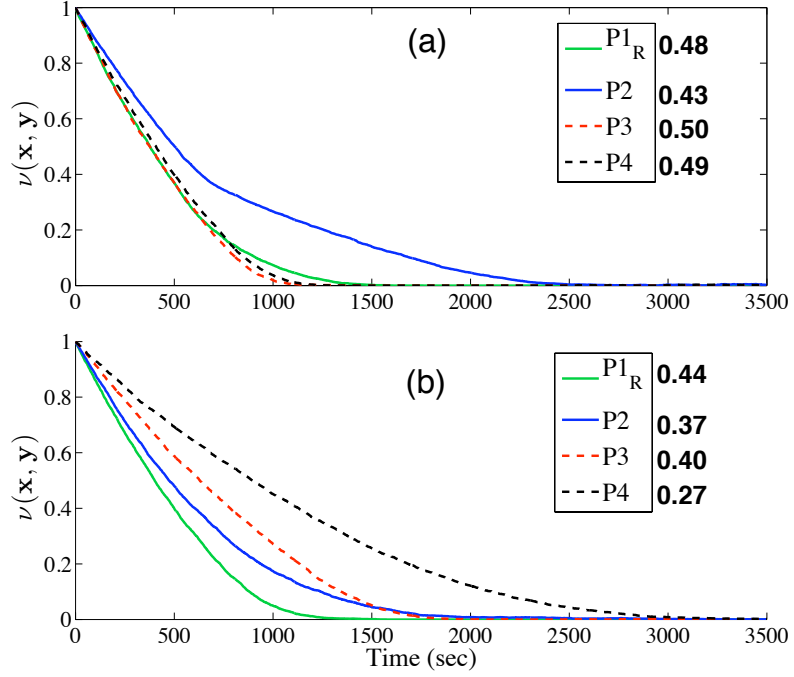
Figure 6.18: The same quantities as in Figure 6.17 for runs using graph Figure 6.3b.

about a third of the average robot waiting time at site i, k_{ij}^{-1}, which indicates that the travel times do not in general contribute as much to system convergence time as site occupancy times.

The similarity among the results for the micro-continuous simulations run with transition rates from the baseline and chain models indicates that for the purpose of controller synthesis, the baseline model is a sufficiently accurate representation of the system in this case. Hence, we can simply optimize the matrix $\mathbf{K}$ and do not have to incur the greater computational expense that is needed to optimize the larger matrix $\hat{\mathbf{K}}$.

42-Site Environment

This work was done in collaboration with M. Ani Hsieh and Ádám Halász.

We simulated the macro-continuous and micro-continuous models of the 42-site scenario using a robot population of $20,000$. The robots are initially distributed

110
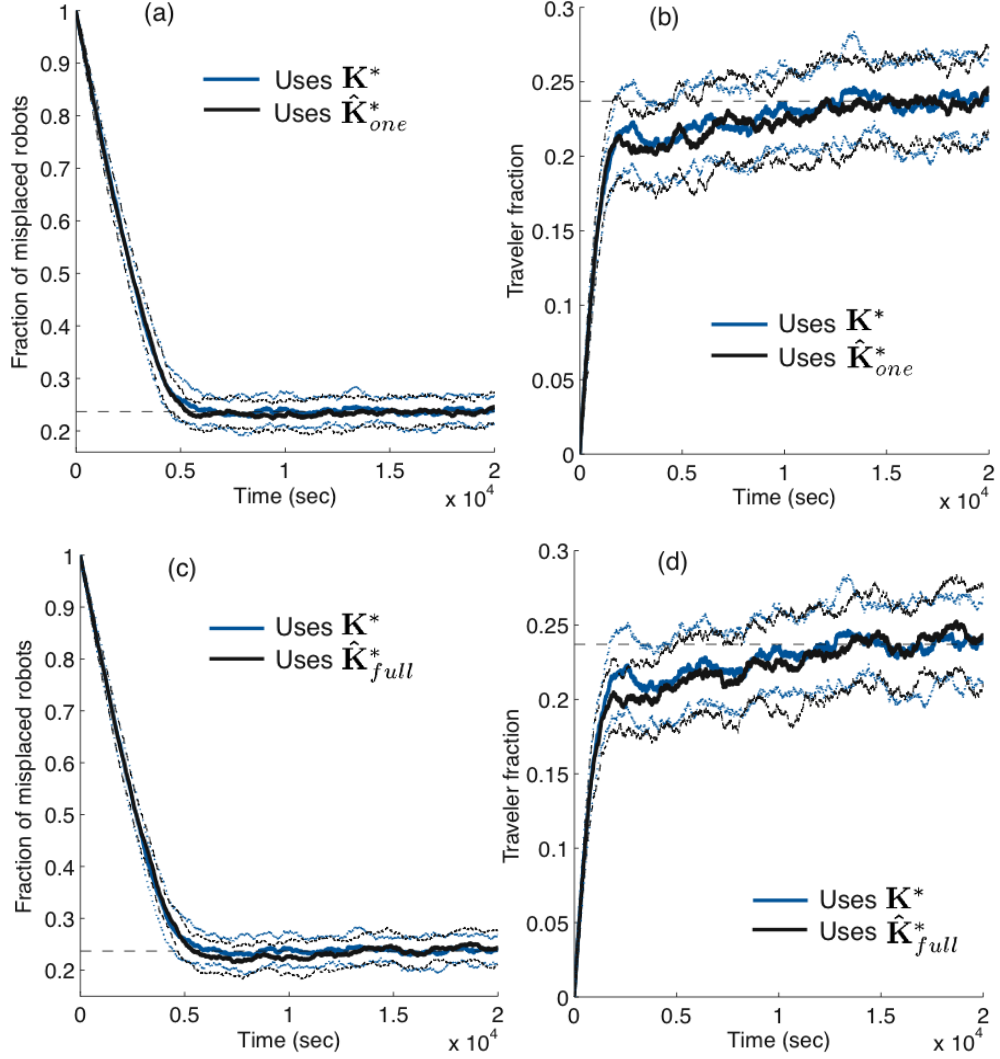
Figure 6.19: (a),(c) Fraction of misplaced robots $\mu(\mathbf{x})$ and (b),(d) fraction of travelers vs. time for micro-continuous simulations using $\mathbf{K}$, $\hat{\mathbf{K}}_{one}$, and $\hat{\mathbf{K}}_{full}$. Thick lines are averages over 40 simulation runs; thin lines mark the standard deviations. The horizontal dashed lines mark the mean equilibrium traveler fraction, 0.237, measured from the $\mathbf{K}$ runs.

111
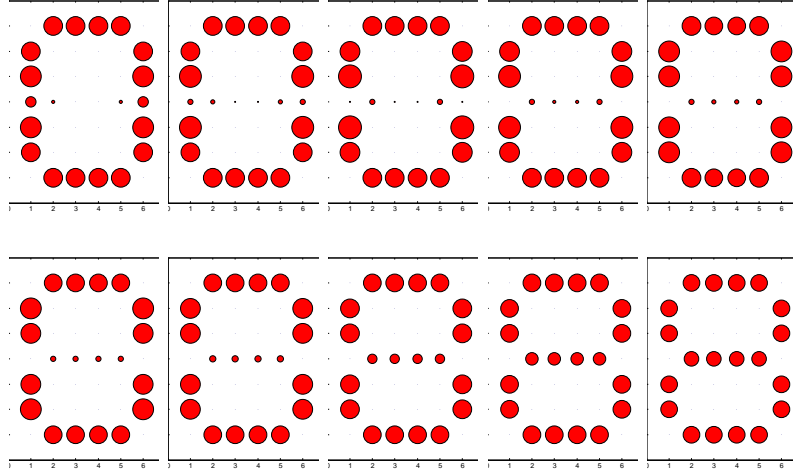
Figure 6.20: Snapshots, sequenced from top left to bottom right, of a simulation in which $20,000$ robots use the rates in $\mathbf{K_n}$ to redistribute to form the design specification for the number 8. *Created by Ádám Halász; reproduced from Figure 4 in [69].*

among sites that form the number 0; they must redistributed to another set of sites that form the number 8. Snapshots of the simulation are shown in Figure 6.20, in which the red circles represent the number of the robots at each site. The larger the circle, the higher the robot population.

Figure 6.21 plots $||\mathbf{x} - \mathbf{x^d}||_1$ over time for micro-continuous simulations that use transition rates from each of the three $\mathbf{K}$. The figure shows that the quorum-based model allows us to maximize transient traffic between sites without sacrificing the limit on the number of idle trips at equilibrium.

Figure 6.22 shows snapshots of a second, more radical redistribution of $20,000$ robots among the 42 sites. The robots are initially located at the four corner sites, and the task is to redistribute them to another set of sites that form the letter S. Robots are denoted by dots that are arranged into circles at each site.

112
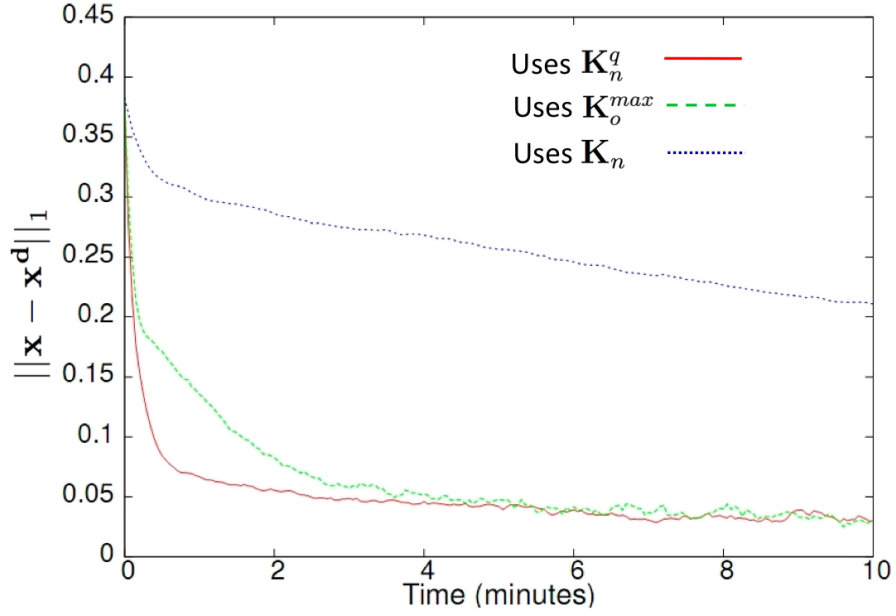
Figure 6.21: A measure of the fraction of misplaced robots vs. time for micro-continuous simulations using $\mathbf{K}_n$, $\mathbf{K}_o^{max}$, and $\mathbf{K}_n^q$. *Created by Ádám Halász; reproduced from Figure 6a in [69].*
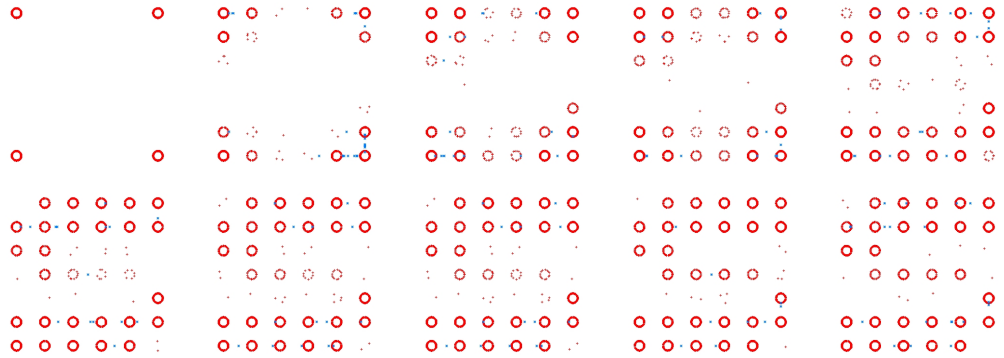
Figure 6.22: Snapshots, sequenced from top left to bottom right, of a simulation in which $20,000$ robots use the rates in $\mathbf{K}_n^q$ redistribute to form the design specification for the letter S. *Created by Ádám Halász; reproduced from Figure 7 in [69].*

113

6.5 Conclusions

We have presented an approach to the problem of redistributing a swarm of robots in a decentralized manner among a network of sites without using inter-robot communication. We designed ant-inspired behaviors that cause robots to converge to the best site or split between two sites in a specified ratio, possibly while transferring resources. We also demonstrated our task reallocation approach on a realistic surveillance scenario with 4 sites and a much larger system with 42 sites. In the 4-site scenario, we compared system performance for the k_{ij} from the baseline linear model, which was designed using the optimization methods from Section 5.2.1 for producing fast convergence to a target distribution and long-term efficiency. We accounted for inter-site travel times, measured from the micro-continuous simulation, by constructing a linear chain model, which did not yield significantly different performance in this case. The 42-site scenario employed ant-inspired quorum-based control policies that, when used with non-optimal k_{ij}, outperformed both optimized and non-optimal baseline models.

Chapter 7

Application: Swarm Robotic Assembly System

The work in this chapter was first presented in [17, 109] and was done in collaboration with Loïc Matthey; see individual sections for the division of work.

We consider a scenario in which a large supply of heterogeneous parts must be assembled into target amounts of different products. The assembly strategy should be scalable in the number of parts, easily modeled to facilitate the optimization of appropriate parameters for fast production, and quickly adjustable when product demand changes.

We fulfill these criteria by using a swarm of autonomous mobile robots to execute the assembly task in a decentralized fashion. The robots move randomly inside an assembly workspace, identify and pick up randomly scattered parts, and combine them according to a predefined assembly plan. These actions are performed using local sensing and local communication with parts and with other robots. Since robots and parts are uniformly randomly distributed throughout the workspace, the system can be modeled as being well-mixed, which allows us to abstract it to an ODE macro-continuous model according to the justification in Chapter 3. We optimize the reaction rate constants in the corresponding macro-continuous model to minimize

system convergence time while enforcing target quantities of all parts at equilibrium. It should be emphasized that the rate constants are optimized with respect to the particular assembly plan that has been chosen; the plan itself may be optimized as well, although this problem is not addressed in this thesis. We then map these parameters onto probabilities of executing assemblies and disassemblies, which are used as robot control policies in the micro-continuous model. When product demand changes, the only adjustment needed is the update of these probabilities, such as via a broadcast. This strategy can be readily implemented on resource-constrained robots and is scalable in the number of parts and robots.

We use the modeling, analysis, and controller synthesis methodologies to apply this strategy to the following assembly task. There are four types of parts, numbered 1 through 4, which are combined to form larger parts according to the assembly plans in Figure 7.1, culminating in final assemblies $F1$ and $F2$. These plans were chosen to allow subassemblies to be created in parallel, contributing to fast production of $F1$ and $F2$. The assembly plans, like the control policies, can be preprogrammed onto the robots and updated via a broadcast if they are changed. Parts bond through bi-directional connections at sites along their perimeters. The assembly task is executed by a group of robots in an arena that is sufficiently large to prevent robot crowding. Initially, robots and many copies of parts 1 through 4 are randomly scattered throughout the arena. There are exactly as many parts as are needed to create a specified number of final assemblies, and the number of robots is at least the total number of scattered parts. Each robot has the ability to recognize part types, pick up a part, combine it with one that is being carried by another robot, and disassemble a part it is carrying.

116
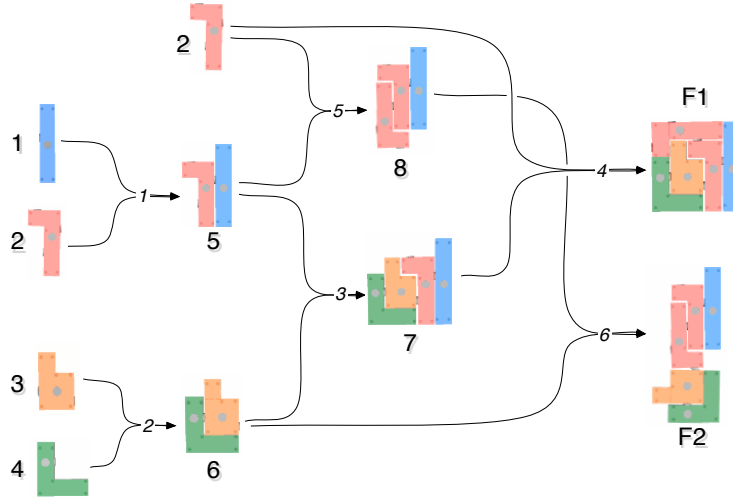
Figure 7.1: Assembly plans for final assemblies $F1$ and $F2$.

7.1 Modeling

7.1.1 Micro-Continuous Model

Implementation

The robot and part controllers were initially coded by Spring Berman and then modified by Loïc Matthey to improve scalability, accommodate arbitrary assembly plans, and implement the desired robot motion patterns. Spring Berman designed the robot arms and parts, and Loïc Matthey designed the arena environment.

We implement the assembly task in the robot simulator Webots [113]. We use the robot platform Khepera III, which has infra-red distance sensors for collision avoidance. Each robot is outfitted with a protruding bar with a rotational servo at the tip. A magnet on the servo bonds to a magnet on the top face of a part, and the servo is used to rotate the bonded part into the correct orientation for assembly. Parts bond to each other via magnets on their side faces. Magnets can be turned off to deactivate a bond. Robots and parts are equipped with an infra-red emitter/receiver for local communication and for computing relative bearing, which is used to align robot and part magnets and to rotate a part for assembly. The task

117
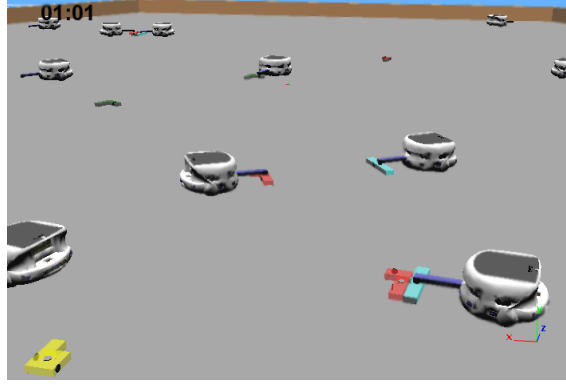
Figure 7.2: Snapshot of the arena in the realistic physical simulation. Robots carry parts at the end of a protruding bar.

takes place inside a walled hexagonal arena. Figure 7.2 shows a snapshot of the simulation.

To achieve the spatial homogeneity that we assume in our models, robots move according to a random walk, and we verify that the space is uniformly covered (Figure 7.3). Robots and parts switch between action states based on information they receive via local sensing and communication. When a robot encounters a part on the ground, it approaches and bonds to it and starts searching for a robot that is carrying a compatible part, according to the assembly plans. When one is found, the two robots align their parts and approach each other to join the parts. One robot carries off the newly assembled part while the other resumes searching for a part on the ground. A robot can disassemble a part it is carrying by deactivating a magnetic part bond, which releases one of the component parts. To control the outcome of part populations, we directly modify the probabilities of robots starting an assembly and performing a disassembly.

CRN Descriptions

Loïc Matthey developed the CRN's in this section.

Each part of type i in Figure 7.1 is symbolized by X_i, and a robot is symbolized by
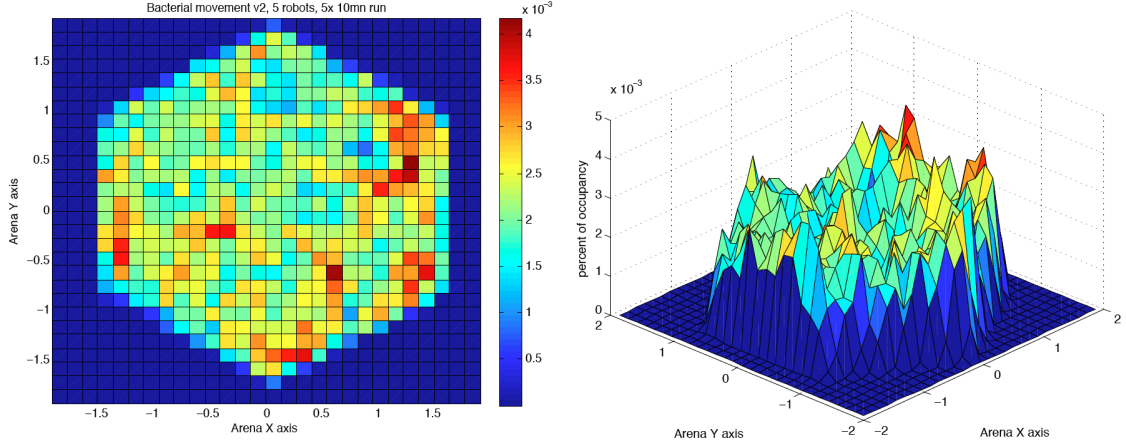
Figure 7.3: Average coverage of the arena by 5 robots over 5 runs of 10 minutes each [108]. The distribution of robots over the arena is approximately uniform.

X_R. X_i may be classified as X_i^u, an unclaimed part on the ground, or as X_i^c, a claimed part i and the robot that is carrying it. We define a *complete CRN*, illustrated in Figure 7.4, that represents each possible action in the micro-continuous model:

$$
\begin{aligned}
X_R + X_i^u \;&\xrightarrow{e_i}\; X_i^c\,, \qquad i = 1, ..., 8 \\
X_1^c + X_2^c \xrightarrow{k_1^+} X_5^c + X_R \qquad & X_2^c + X_7^c \xrightarrow{k_4^+} X_{F1}^c + X_R \\
X_3^c + X_4^c \xrightarrow{k_2^+} X_6^c + X_R \qquad & X_2^c + X_5^c \xrightarrow{k_5^+} X_8^c + X_R \\
X_5^c + X_6^c \xrightarrow{k_3^+} X_7^c + X_R \qquad & X_6^c + X_8^c \xrightarrow{k_6^+} X_{F2}^c + X_R \\
X_5^c \xrightarrow{k_1^-} X_1^c + X_2^u \qquad & X_{F1}^c \xrightarrow{k_4^-} X_7^c + X_2^u \\
X_6^c \xrightarrow{k_2^-} X_3^c + X_4^u \qquad & X_8^c \xrightarrow{k_5^-} X_5^c + X_2^u \\
X_7^c \xrightarrow{k_3^-} X_6^c + X_5^u \qquad & X_{F2}^c \xrightarrow{k_6^-} X_8^c + X_6^u
\end{aligned}
\tag{7.1}
$$
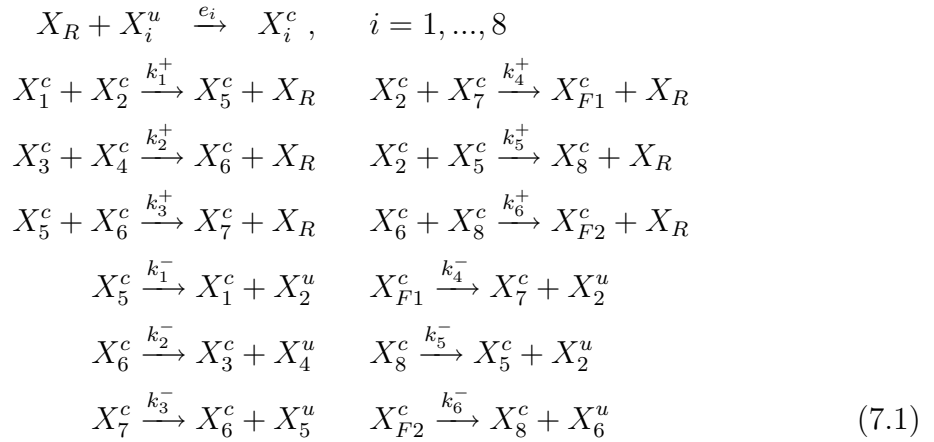
In this CRN, e_i is the rate constant at which a robot encounters a part of type i, k_j^+ is the rate constant of assembly process j, and k_j^- is the rate constant of disassembly process j. Using the decomposition of the stochastic reaction constant c_{ij} given in Section 3.1.1, we estimate these rate constants as functions of the following

119
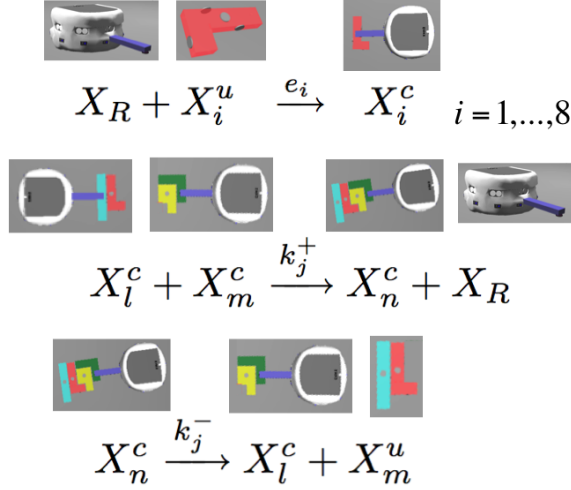
Figure 7.4: CRN representing all actions in the assembly system.

probabilities:

$$e_i = A \, p^e \, , \qquad k_j^+ = A \, p^e \, p_j^a \, p_j^+ \, , \qquad k_j^- = p_j^- \, , \tag{7.2}$$

where A is the area of the arena. A is included because it is the proportionality constant between the k_{ij} and c_{ij} of a bimolecular reaction.

p^e is the probability per unit time that a robot encounters a part or another robot. Since our arena size yields a low robot density, this probability is modeled as being independent of the robot population. The property that robots and parts are distributed uniformly throughout the arena allows us to calculate p^e as c_{ij}^e, given by Equation (3.4). In this equation, v is defined as the average robot speed and w is twice a robot's communication radius, since this is the range within which a robot detects a part or robot and initiates an assembly process.

p_j^a is the probability of two robots successfully completing assembly process j; it depends on the part geometries.

p_j^+ is the probability of two robots starting assembly process j, and p_j^- is the probability per unit time of a robot performing disassembly process j. These are the *tunable parameters* of the system.

In order to be able to prove convergence properties of the system, we reduce its

dimensionality by abstracting away robots and retaining only interactions between parts, assuming that the time for a robot to find a part is small and that there are at least as many robot as parts. Then the complete CRN becomes the *reduced CRN*,

$$X_1 + X_2 \underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} X_5 \quad X_3 + X_4 \underset{k_2^-}{\overset{k_2^+}{\rightleftharpoons}} X_6 \quad X_5 + X_6 \underset{k_3^-}{\overset{k_3^+}{\rightleftharpoons}} X_7$$

$$X_2 + X_7 \underset{k_4^-}{\overset{k_4^+}{\rightleftharpoons}} X_{F1} \quad X_2 + X_5 \underset{k_5^-}{\overset{k_5^+}{\rightleftharpoons}} X_8 \quad X_6 + X_8 \underset{k_6^-}{\overset{k_6^+}{\rightleftharpoons}} X_{F2} \tag{7.3}$$
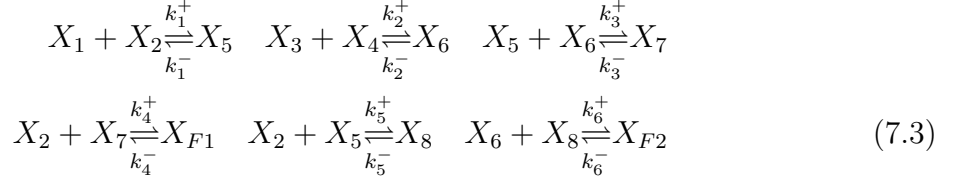
The rate constants in this CRN are also defined by Equation (7.2).

7.1.2 Macro-Continuous Model

Spring Berman defined the model in this section.

Both the complete CRN and the reduced CRN contain bimolecular reactions. Therefore, the corresponding macro-continuous models take the form of the multi-affine model (3.15). The complete CRN abstracts to the *complete macro-continuous model*, and the reduced CRN abstracts to the *reduced macro-continuous model*. Here we describe the latter model. The reduced CRN has $S = 10$ species, $C = 12$ complexes, and $R = 6$ reactions. The species vector $\mathbf{x} = [x_1 \ \ldots \ x_8 \ x_{F1} \ x_{F2}]^T$ contains the concentrations of different part types. The vector of complexes is

$$\mathbf{y}(\mathbf{x}) = [x_1 x_2 \ x_5 \ x_3 x_4 \ x_6 \ x_2 x_7 \ x_{F1} \ x_5 x_6 \ x_7 \ x_2 x_5 \ x_8 \ x_6 x_8 \ x_{F2}]^T . \tag{7.4}$$

The rate constants in the CRN can be relabeled as k_{ij} according to their associated reaction pathways (i, j), where complexes i and j are numbered according to their positions in $\mathbf{y}(\mathbf{x})$ (i.e., complex 3 is $X_3 + X_4$), and used to define $\mathbf{K}$ according to (3.14). One set of linearly independent conservation constraints on the x_i is:

$$\begin{aligned}
x_3 - x_4 &= N_1 \\
x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} &= N_2 \\
x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) &= N_3 \\
x_3 + x_6 + x_7 + x_{F1} + x_{F2} &= N_4
\end{aligned} \tag{7.5}$$

where N_i, $i = 1, ..., 4$, are computed from the initial part quantities.

7.2 Analysis

7.2.1 Micro-Continuous and Macro-Discrete Models

Loïc Matthey ran the simulations described in this section.

To confirm that the complete macro-continuous model is an accurate representation of the system, we compare the evolution of products in this model to simulations of the micro-continuous and macro-discrete models. The forward rate constants e_i and k_j^+ contain the parameters that are measured from the Webots simulations; therefore, we only simulate assemblies ($p_j^+ = 1$, $p_j^- = 0$ $\forall j$) to verify that the system with these rate constants can be abstracted correctly. We set $T = 1$ s, and the other parameters for p^e are $A = 23.4$ m^2 (hexagon of radius 3 m), $w = 1.2$ m, and $v = 0.128$ m/s from an average over 50 runs. The p_i^a were measured as $\mathbf{p^a} = [0.9777\ 0.9074\ 0.9636\ 0.9737\ 0.8330\ 1.0]$ (entries follow the numbering of the associated reactions) from averages over 100 runs. We numerically integrated the complete macro-continuous model with the calculated e_i and k_j^+, and we used the StochKit toolbox [99] to efficiently perform a stochastic simulation of the macro-discrete model.

Figure 7.5a shows all part populations, averaged over 100 Webots runs, for a system of 15 robots and 15 parts of types 1 through 4. The unlabeled trajectories are the quantities of the subassemblies, which increase as they are created and then decrease as they are combined to form final products $F1$ and $F2$. The standard deviations are relatively large because the system can create only 3 final products from the 15 parts.

Figure 7.5b compares the evolution of final products in all three models. Discrepancies among the models arise from several factors. The ODE model is most accurate for very large populations, while the system has relatively low numbers of parts and robots so that it would not be too computationally expensive to simulate. If the robot and part populations were increased, the averaged simulations
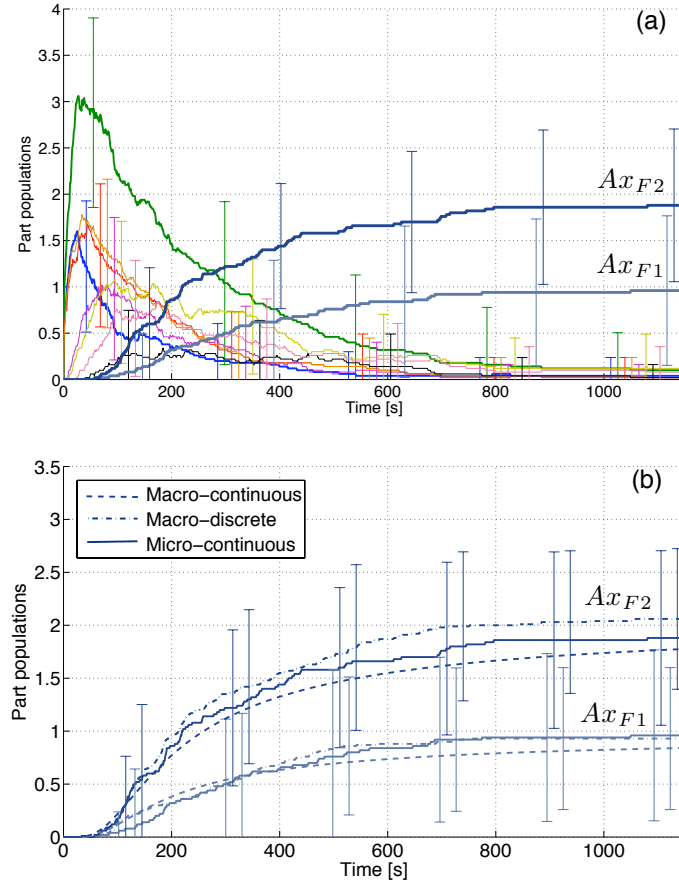
Figure 7.5: Evolution of part populations for a system with 15 robots and parts for 3 final assemblies. Error bars show standard deviations. (a) All part populations in the micro-continuous model, averaged over 100 runs. (b) $F1$, $F2$ populations in the complete macro-continuous, macro-discrete, and micro-continuous models. The latter two models are each averaged over 100 runs.

would correspond more closely to the macro-continuous model. Also, in the Webots simulations, an assembly is occasionally prevented by robot collisions with walls, the interference of another robot, or erroneous part collisions. We do not model these failures or the small local effect of a higher availability of parts where they are dropped, which often leads to the recreation of broken assemblies. Nevertheless, the macro-continuous model predicts the $F1$ and $F2$ populations fairly accurately, and hence we can use it to design p_j^+, p_j^- to direct the system behavior.

7.2.2 Macro-Continuous Model

Spring Berman and Loïc Matthey collaborated on this section.

Using the results from CRN theory outlined in Section 4.3.1, we can prove that reduced macro-continuous model converges to a designable distribution of part quantities from any initial distribution $\mathbf{x^0}$.

Theorem 6. *The reduced macro-continuous model, subject to conservation equations (7.5), has a unique, globally asymptotically stable equilibrium $\bar{\mathbf{x}} > \mathbf{0}$.*

Proof. From definition (7.4) of $\mathbf{y}(\mathbf{x})$, it can be concluded that in each boundary equilibrium, all $x_i = 0$ except for one of the four combinations of variables (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4). Since we only consider systems that can produce x_{F1} and x_{F2}, it is not possible for the system to reach any of these equilibria; each one lacks two part types needed for the final assemblies.

The reduced CRN (7.3) has 12 complexes, 6 linkage classes, and rank 6; hence, its deficiency, defined by Equation (4.1), is $\delta = 0$. Also, the network is weakly reversible. Because the network has deficiency 0, is weakly reversible, and does not admit any boundary equilibria, it has a unique, globally asymptotically stable positive equilibrium according to Theorem 4.1 of [139]. □

7.3 Controller Synthesis

7.3.1 Macro-Continuous Model

Spring Berman developed the material in this section.

We consider the problem of designing the reduced macro-continuous model, subject to Equations (7.5), to produce target quantities of parts as quickly as possible. The objective will be posed as the design of *optimal probabilities* p_i^+, p_i^-, $i = 1, ..., 6$,

that minimize the convergence time of the system to a vector of target part concentrations, $\mathbf{x^d}$. We formulate an optimization problem in which these probabilities are written in terms of the rates k_i^+, k_i^-, $i = 1, ..., 6$, using Equation (7.2). Although only the amounts of the final assemblies $F1$ and $F2$ may need to be specified in practice, our optimization problem requires that target concentrations of *all* parts be defined.

We first specify $x_1^d, x_2^d, x_3^d, x_5^d, x_8^d$ and a parameter

$$\alpha \equiv x_{F1}^d / (x_{F1}^d + x_{F2}^d) \ . \tag{7.6}$$

Then we compute the dependent variables x_4^d, x_6^d, x_7^d, and $x_{F1}^d + x_{F2}^d$ from the conservation equations (7.5) and definition (7.6) and check that they are positive to ensure a valid $\mathbf{x^d}$. In this way, we can keep $x_{F1}^d + x_{F2}^d$ and the target non-final part quantities constant while adjusting the ratio between x_{F1} and x_{F2} using α.

Theorem 6 shows that we can achieve $\mathbf{x^d}$ from any initial distribution $\mathbf{x^0}$ by specifying that $\bar{\mathbf{x}} = \mathbf{x^d}$ through constraint (5.7) on $\mathbf{K}$. We quantify the time to converge to $\mathbf{x^d}$ in terms of the system relaxation times τ_i, $i = 1, ..., 6$, described in Section 5.2.1. Since it is very difficult to find analytical expressions for the eigenvalues of the Jacobian matrix $\mathbf{J}$ of our system, we use the alternative estimate of relaxation time given by Equation (5.9). Each reaction j in CRN (7.3) is of the form $X_k + X_l \underset{k_j^-}{\overset{k_j^+}{\rightleftharpoons}} X_m$. Thus, the reaction rates are $v_j = k_j^+ x_k x_l - k_j^- x_m$, and the entries of column j in the stoichiometric matrix $\mathbf{S}$ are all 0 except for $\mathbf{S}_{kj} = \mathbf{S}_{lj} = -1$ and $\mathbf{S}_{mj} = 1$. Then, according to Equation (5.9), the relaxation time for each reaction is

$$\tau_j = (k_j^+ (x_k^d + x_l^d) + k_j^-)^{-1} \ . \tag{7.7}$$

Define $\mathbf{k} \in \mathbb{R}^{12}$ as the vector of all k_i^+, k_i^- and $\mathbf{p} \in \mathbb{R}^{12}$ as the vector of all p_i^+, p_i^-. Note that according to Equation (7.2), $\mathbf{k} = \mathbf{k}(\mathbf{p})$. We define the objective function as the average τ_j^{-1}, which should be *maximized* to produce fast convergence to $\mathbf{x^d}$. The optimization problem can now be posed as a linear program, which can be solved efficiently:

[**P**] maximize $\quad \frac{1}{6}\sum_{j=1}^{6}\tau_j^{-1}$

subject to $\quad \mathbf{MK(p)y(x^d)} = \mathbf{0}, \quad \mathbf{0 \leq p \leq 1}$.

For comparison, we implemented Problem P_{MA} in Section 5.2.1 using the Monte Carlo method presented in that section to find the $\mathbf{k(p)}$ that *directly* minimizes the convergence time. Constraint (5.8) was defined as $\mathbf{k(0) \leq k(p) \leq k(1)}$, and the parameter f in the convergence time t_f was set to 0.1.

7.3.2 Micro-Continuous Model

The robots execute a random walk that is based on the run-and-tumble motion of bacteria. Obstacle avoidance is implemented with a Braitenberg algorithm [23], which computes the angular speed of each wheel as a weighted sum of the distance sensor values.

7.4 Results

7.4.1 Macro-Continuous model

Spring Berman produced the results in this section.

To investigate the effect of optimization on the convergence time of the reduced macro-continuous model, we generated non-optimal $\mathbf{k}$, which satisfy constraint (5.7) and $\mathbf{0 \leq p \leq 1}$ but are not optimized for some objective, and computed $\mathbf{k}$ using Problem P and Problem P_{MA}. The non-optimal $\mathbf{k}$ and $\mathbf{k}$ from Problem P were calculated for $\alpha \in \{0.01, 0.02, \dots, 0.99\}$, and the $\mathbf{k}$ from Problem P_{MA} for $\alpha \in \{0.1, 0.2, \dots, 0.9\}$. We set $\mathbf{x^0} = [60\ 120\ 60\ 60\ \mathbf{0}]^T$ and $\mathbf{x^d} = [0.5\ 2.5\ 1\ 1\ 0.5\ 1\ 1\ 1\ 57\alpha\ 57(1-\alpha)]^T$. Problem P produced the same rate constants for each α (for instance, $p_i^+ = 1\ \forall i$) except for the rate constants of disassembly processes 4 and 6, which vary with α. This shows that the system is flexible enough to yield any α when only the rates of breaking apart the final assemblies are modified.

126

Figure 7.6 compares the convergence time $t_{0.1}$ of the reduced macro-continuous model with these different $\mathbf{k}$. It is evident that optimized $\mathbf{k}$ produce much faster convergence times than the average $t_{0.1}$ (over 100 values) using non-optimal $\mathbf{k}$. The $\mathbf{k}$ from Problem P_{MA} consistently yield the fastest convergence but are time-consuming to compute: on a standard 2 GHz laptop, it takes about 10 hours for $t_{0.1}$ to decrease slowly enough with each Monte Carlo program iteration for $\mathbf{k}$ to be considered close enough to optimal. The $\mathbf{k}$ from Problem P, which for each α are computed in less than a second, yield $t_{0.1}$ that dip close to the Monte Carlo times for $\alpha = 0.2 - 0.5$ but increase up to two orders of magnitude outside this range. This shows that linearizing the ODE model for optimization is most effective within a certain range of target product distributions.

We numerically integrated the reduced macro-continuous model for $\alpha = 0.1$, 0.5, and 0.9 using both optimized and non-optimal $\mathbf{k}$. The evolution of the model for each $\mathbf{k}$ is shown in Figure 7.7. Consistent with Figure 7.6, the optimized models converge faster to the target final product fractions than the non-optimal model for each α.

7.4.2   Micro-Continuous model

Loïc Matthey produced the results in this section.

To confirm that optimizing $\mathbf{k}$ using the reduced macro-continuous model does indeed increase the convergence rate of the micro-continuous model, we mapped the optimized and non-optimal $\mathbf{k}$ onto robot actions in the Webots simulation for $\alpha = 0.1, 0.5, 0.9$. This mapping, described in general terms in Sections 3.1.2 and 3.1.1, was done as follows. A robot carrying a part that can be disassembled according to process j computes a uniformly distributed random number $u \in [0, 1]$ at each timestep Δt (32 ms in Webots) and disassembles the part if $u < p_j^- \Delta t$. A robot about to begin assembly process j computes u and executes the assembly if $u < p_j^+ \Delta t$.

127
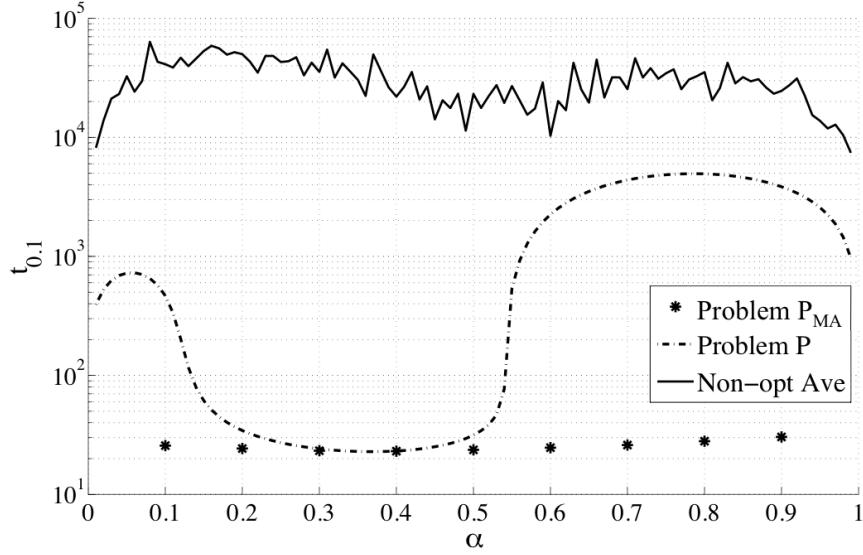
Figure 7.6: Time for the reduced macro-continuous model to converge to $0.1\Delta(\mathbf{x^0})$ vs. α for optimized and non-optimal $\mathbf{k}$. "Non-opt Ave" is the average of 100 $t_{0.1}$ corresponding to different random feasible $\mathbf{k}$ for each α.

Figures 7.8 and 7.9 plot the time evolution of final product fractions in the micro-continuous and macro-continuous models for the different $\mathbf{k}$. The results in Figure 7.8 are from a system with 15 robots and 15 parts, and those in Figure 7.9 are from a larger system with 50 robots and 50 parts. The figures show that the averaged trajectories from the simulations follow the same trends as the corresponding ODE trajectories; discrepancies are due to the factors described in Section 7.2.1. For each α, the simulation runs using the $\mathbf{k}$ from Problem P_{MA} make the most progress toward the target product fractions, the runs using the non-optimal $\mathbf{k}$ make very little progress, and the runs using the $\mathbf{k}$ from Problem P display intermediate performance. Figure 7.9 illustrates that adding more robots and parts decreases the standard deviations of the final product fractions for simulations using optimized $\mathbf{k}$. However, simulations with non-optimal $\mathbf{k}$ produce final assemblies very infrequently, resulting in large error bars. For comparison, the average numbers of $F1$, $F2$ assemblies at $t = 1200$ sec are 0.25, 0.80 when using non-optimal $\mathbf{k}$; 2.90, 2.60 for $\mathbf{k}$ from Problem P; and 2.41, 2.50 for $\mathbf{k}$ from Problem P_{MA}. All of these results demonstrate that a
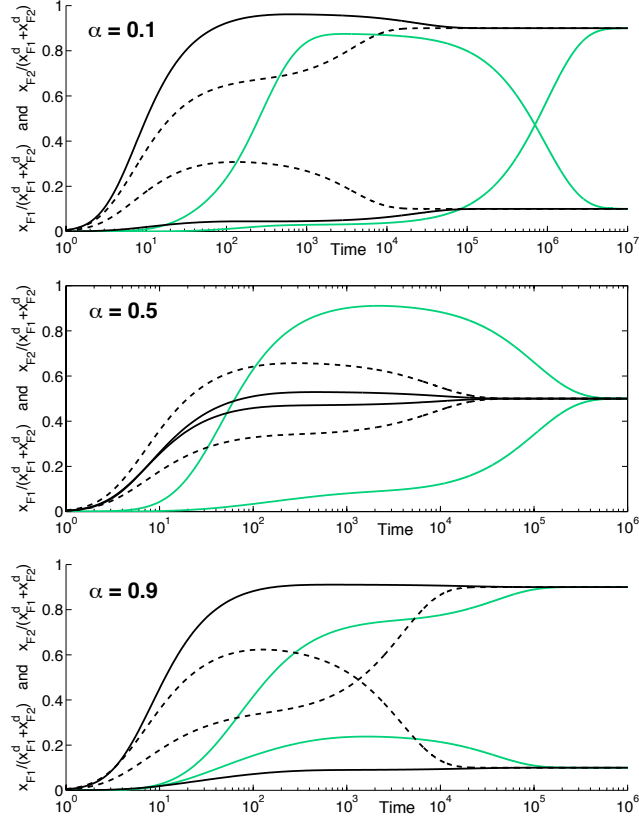
Figure 7.7: Evolution of final product fractions in the reduced macro-continuous model for $\alpha = 0.1, 0.5, 0.9$ using non-optimal **k** (light solid lines) and **k** optimized by Problem P_{MA} (dark solid lines) and Problem P (dashed lines).

simple ODE model can indeed be used to compute control policies that improve the yield rate when used in a realistic system model.

7.5   Conclusions

We have presented a method to systematically derive decentralized, stochastic control policies for a swarm of robots to quickly manufacture different products in response to varying demand. The collective behavior of the system is abstracted to an ODE model whose parameters, the rate constants of assembly and disassembly, govern the control policies running on individual robots for executing the assembly
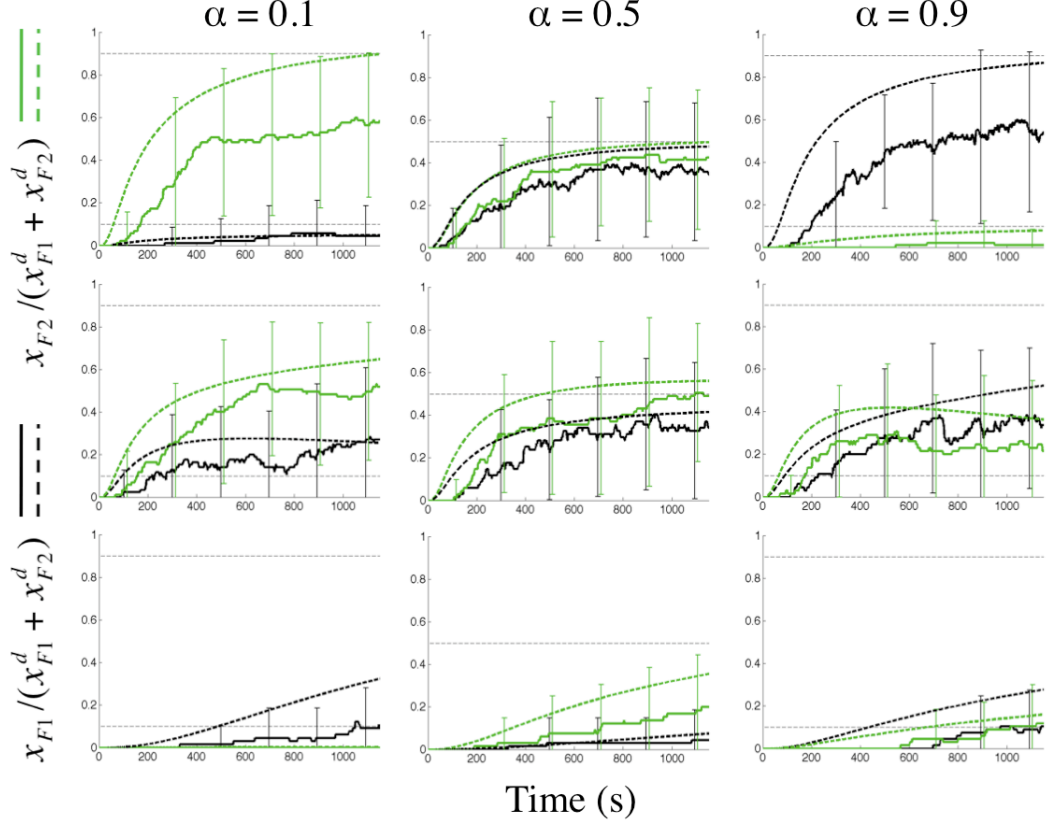
Figure 7.8: $F1$, $F2$ fractions in the micro-continuous (solid lines) and reduced macro-continuous (dashed lines) models using **k** optimized by Problem P_{MA} (top row) and Problem P (center row) and non-optimal **k** (bottom row). Micro-continuous models are each averaged over 30 runs and have 15 robots and parts for 3 final assemblies. $x_{F1}^d + x_{F2}^d$ was computed as the equilibrium $x_{F1} + x_{F2}$ of model (3.15) with $\mathbf{x^0} = [3\ 6\ 3\ 3\ \mathbf{0}]^T$.

task. By tuning these rates, we tune the performance of the system. This optimization relies on global stability properties of a specific class of ODE's to which the macro-continuous model belongs, and it is independent of the number of parts and robots. We map the rates onto robot probabilities of executing assemblies and disassemblies and find that the average evolution of products over time in the resulting system follows the prediction of the macro-continuous model. It is expected that performance closer to the ODE model would be observed in a micro-continuous model with more accurate simulation of rigid body contacts.
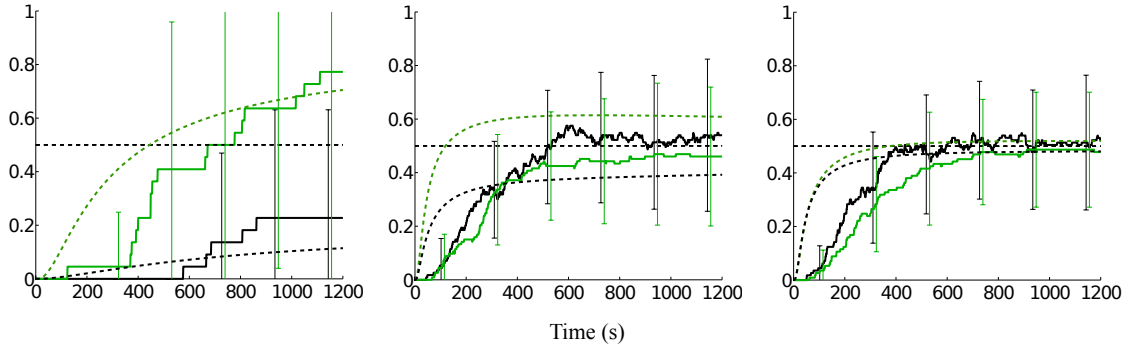
Figure 7.9: $F1$, $F2$ fractions (see legend in Figure 7.8) in the micro-continuous (solid lines) and reduced macro-continuous (dashed lines) models for $\alpha = 0.5$ using non-optimal $\mathbf{k}$ (left) and $\mathbf{k}$ optimized by Problem P (center) and Problem P_{MA} (right). Micro-continuous models are each averaged over 20 runs and have 50 robots and parts for 10 final assemblies. $x_{F1}^d + x_{F2}^d$ was computed as the equilibrium $x_{F1} + x_{F2}$ of model (3.15) with $\mathbf{x^0} = [10\ 20\ 10\ 10\ \mathbf{0}]^T$.

Chapter 8

Application: Bio-Inspired Collective Transport

The work in this chapter was first presented in [18] and was done in collaboration with Prof. Stephen Pratt, Quentin Lindsey, Mahmut Selman Sakar, Prof. David Cappelleri, and Travis Van Schoyck; see individual sections for the division of work.

While there are various approaches to cooperative robotic manipulation, there are very few completely decentralized approaches. Such approaches can provide a greater degree of flexibility and robustness in construction and manufacturing applications. Cooperative transport in ants is a striking example of a fully decentralized manipulation strategy that is scalable in the number of transporters and successful for a wide range of payloads (food items) and environments, including ones with obstacles. This biological phenomenon offers inspiration for the analogous problem in robotics, in which a group of robots, each equipped with a gripper, must manipulate an arbitrarily-shaped payload that is too heavy for a single robot to move through an obstacle-filled environment. Figure 8.1 shows an example robotic platform for this task: a SCARAB differential drive robot [111], developed in the GRASP Laboratory, equipped with a gripper designed by Quentin Lindsey, Michael Shomin, and Travis Van Schoyck. The robots must employ a scalable transport strategy that uses local
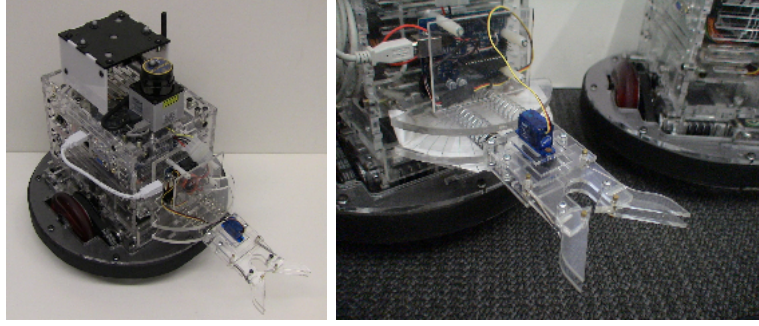
Figure 8.1: A SCARAB robot equipped with a single-actuator gripper with passive compliance.

sensing and no inter-robot communication, and they do not have *a priori* knowledge about the payload or obstacles.

To extract the rules that govern ant transport behavior and better understand the mechanisms of successful collective transport, we study this phenomenon in *Aphaenogaster (ex-Novomessor) cockerelli*, a monomorphic ant species that is common in the deserts of the southwestern United States. To our knowledge, this is the first work that investigates the mechanics of cooperative transport in ants. We fabricate elastic two-dimensional structures that we use as vision-based ant force sensors and videotape ant retrieval of the structures. We track the structure deformations produced by the ants, the structure configuration over time, and the evolution of the ant population in the retrieval group. Using this data, we quantitatively illustrate salient features of ant transport, including the emergence of consensus, that can be used as metrics to validate a model of the system. In a step toward implementation on multi-robot systems, we develop a micro-continuous model of the ant transport behavior from qualitative observations of the videos of transport trials. We also outline an approach to synthesizing control policies that will arrange the robots into transport groups of target sizes in a scenario with multiple payloads.

8.1 Collective Transport in Ants

The phenomenon of *collective transport* in ants occurs when a group of ants cooperatively retrieves a food item that is too heavy for a single ant to bring back to the nest on its own. This behavior reduces the worker force required to retrieve a given weight of food [64, 114], enhancing the colony's efficiency at food collection. Collective transport is limited to species with effective recruitment communication, and it has evolved independently among most subfamilies of ants. It is best developed in ants of the genera *Aphaenogaster* (funnel ants), *Pheidologeton* (marauder ants), *Eciton* (army ants), and *Oecophylla* (weaver ants) [65].

Among the most impressive collective transporters are the group-raiders *Eciton burchelli* [43] and *Pheidolegeton diversus* [114], which form massive swarms that can retrieve exceptionally large prey items. For instance, *P. diversus* workers have been observed to carry objects as much as 5000 times their weight and 10000 times their volume in groups of as many as 100 ants [114]. The small colonies of *A. cockerelli* form no such raids, but they are also skilled at group retrieval [64]. When a solitary forager finds an item too large to move, she summons a team of nestmates within a few minutes. Foragers recruit over a short range by releasing poison gland contents into the air; if this is inadequate, they initiate longer-range recruitment by laying a pheromone trail back to their nest [64]. These chemical signals can be enhanced by vibrational signals, and fast running by ants acts as a local recruitment stimulus [107]. In this way, a group of *A. cockerelli* can quickly retrieve the food before the arrival of more aggressive, mass-recruiting competitors.

Collective transport is a stigmergic process in the sense that the medium of communication between individuals is the progress of the transport itself [142]. The tractive resistance of the prey stimulates orientational and positional changes by the ants as well as recruitment behavior. When a prey item becomes stuck, ants realign their bodies without releasing their grasp, and if that fails, they release their grasp and reposition themselves on the prey [141, 143].

The variability in individual ant forces can aid the transport group in circumventing obstacles. In [110], it was observed that a group of *Formica polyctena* ants could move a prey out of a fold in a piece of cardboard more quickly when the prey was 30 m from the nest than at 20 m. This was explained as a result of the ants' increased difficulty in orienting at this distance, which raised their probability of moving in the direction opposite the nest, a tendency that helped them overcome the obstacle.

The process of collective transport has been characterized as having an "uncoordinated" phase in which ants arrive at the prey, either by chance or via a recruitment process, and pull on it in all directions, essentially immobilizing the prey, and a "coordinated" phase during which a preferred direction emerges from the ants' exertions and transport occurs [102], [147]. The transition to the coordinated phase has been attributed to spatial rearrangements [114], a decrease in number of ants, realignments and rotation of the prey [143], and tensions acting as positive feedback when associated with success [102].

There has been some debate over the existence of cooperation among ants during transport. On one hand, the ants often appear to behave as though they were isolated, at times pulling on the load in opposite directions. Indeed, ants often appear to exhibit the same behaviors in solitary transport and group transport [141, 143]. However, the concurrent behaviors of multiple ants during group transport produces some cooperative features: groups can move prey significantly faster [26], develop higher mean power [143], and exert greater forces than single ants [141]. And, of course, a group can successfully transport the prey item to the nest. The consensus is that this apparent collaboration does not arise from conscious deliberation, learning, or communication, but is rather a juxtaposition of uncoordinated individual forces on the prey with a strong component toward the nest, the common destination of the ants [102]. Essentially, the ants agree on the direction of the transport, but disagree on how to maneuver the prey into that direction. This strategy allows the ants to

135

move different types of prey over a variety of surfaces [142].

8.2 Elastic Vision-Based Force Sensor

8.2.1 Design

We initially observed that a group of 5 to 8 ants was able to transport a piece of fig of mass $m = 1.0 - 1.5$ g (Figure 8.2). Next, trials were conducted with plastic disk-shaped loads, which were coated with fig paste to induce retrieval. A group of about 15 ants was able to transport a disk of mass $m = 1.6$ g and diameter $d = 4.6$ cm (Figure 8.3). The ants were better at transport when the perimeter of the disk had small tabs, measuring 2 mm long and 0.5 mm wide, for their mandibles to grasp. More than 20 ants participated in carrying a larger disk, with $m = 3.4$ g and $d = 6.9$ cm, although they were not as effective as the group carrying the smaller disk.

Using these dimensions as guidelines, we developed and tested several planar elastic structures, shown in Figure 8.4. Each structure consisted of a circular load ringed with several springs that were tipped with thin tabs (see the labels on the structure in Figure 8.4e). In all of the structures, some ants gripped the springs themselves rather than the tabs; it was difficult to dissuade them from doing this. However, their contributions to the spring deflections did not seem significant since most ants gripped the tabs and bars, which were smeared with fig paste.

The first prototype, Figure 8.4a ($m = 1.3$ g, $d = 8.5$ cm), was laser cut from 0.75-mm-thick Lexan material. A group of 12-15 ants carried the structure effectively once the edges were peeled back with a knife to create 0.5 mm $\times$ 2 mm tabs for the ants to latch onto more easily (Figure 8.5). However, they produced negligible spring deformations, even when several ants pulled on one spring.

To obtain measurable deformations, we switched to fabricating 3.2-mm-thick structures out of the soft material polydimethylsiloxane, or PDMS (see Section 8.2.2 for details). We chose PDMS for its mechanical compliance, manufacturability, and
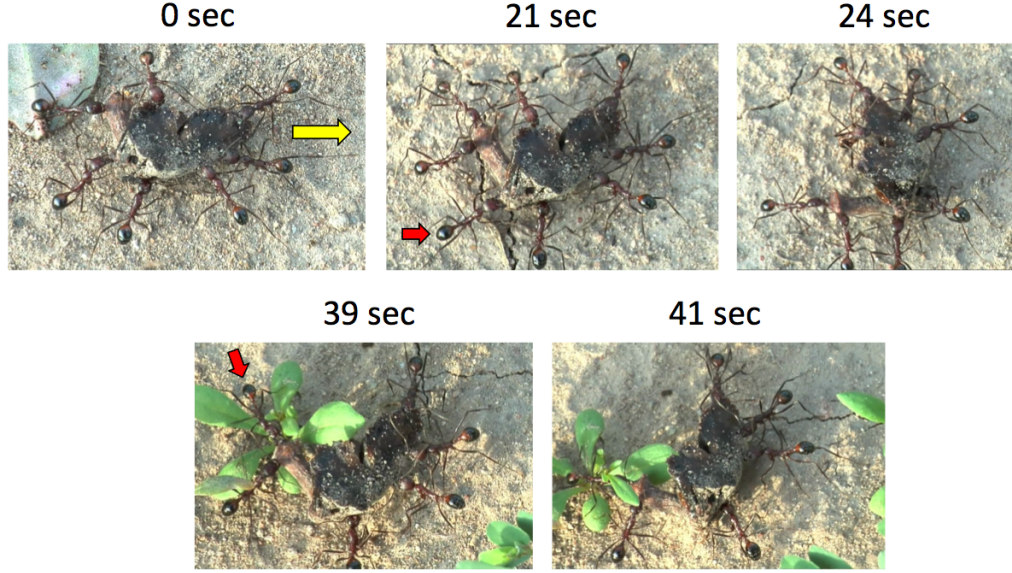
Figure 8.2: Video frames of ants transporting a piece of fig. The yellow arrow shows the direction of movement for all frames. At 21 sec, the ant indicated by the red arrow arrives at the base of the fig stem and induces a counterclockwise rotation of the fig before it leaves at 24 sec. At 39 sec, the ant indicated by the red arrow encounters a plant, and seemingly in response, the ants maneuver such that the fig rotates counterclockwise and the blocked ant can move freely again.

biocompatibility [158]. The prototype in Figure 8.4b ($m = 2.6$ g, $d = 4.5$ cm) was dragged by 3 to 5 ants (Figure 8.6). The transport was slow, indicating that the structure was subject to a high frictional force. The ants deformed the springs significantly in the radial (but not tangential) direction; however, adjacent springs often interfered with each other during transport. The structure in Figure 8.4d ($m = 2.3$ g, $d = 6.1$ cm), which we call structure 1, was designed to reduce this interference and to measure the force exerted by each individual ant. The springs were tipped with wide curved bars in an effort to focus the ants' attention on the tabs. This structure showed sizable spring deformations in the tangential (but not radial) direction. For the structure in Figure 8.4e ($m = 1.6$ g, $d = 7.7$ cm), which we call structure 2, we modified the spring geometry to have more similar radial and tangential stiffnesses. Since the μ for PDMS is relatively large regardless of surface, we chose to reduce this quantity by applying DuPont$^{\text{TM}}$ Teflon® tape to the bottom
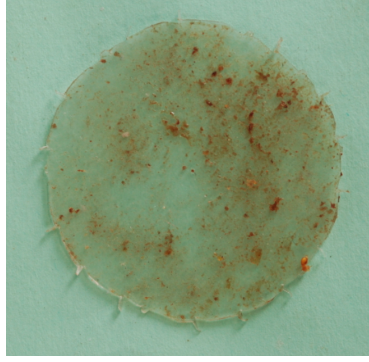
Figure 8.3: Plastic disk coated in fig paste. *Photo by Prof. Stephen Pratt.*

of the loads of structures 1 and 2.

We also tested a 1.6-mm-thick version of structure 2 without the curved bars, shown in Figure 8.4c. This structure was very fragile; three of its springs broke off during manufacturing. It was light enough to be dragged by one to three ants, and as the bottom video frame in Figure 8.7 shows, ants grabbed the middle of the springs as well as the tips and sometimes lifted the springs out-of-plane.

8.2.2 Fabrication

Mahmut Selman Sakar did the fabrication described in this section.

The elastic structures were fabricated using the transfer molding procedure outlined in Figure 8.8. The process started with the fabrication of molds of the structure shape, which was drawn as a CAD file. The substrate used for the mold was acrylic (Lucite International) because of its strength, elasticity, and ease of manufacturing. First, a flatbed laser cutter (Universal Laser Systems X-660) was used to cut the negative structure features (Figure 8.8a). Then the mold and a 5 cm × 5 cm glass plate were silanized by vapor deposition of (tridecafluoro-1,1,2,2-tetrahydrooctyl)-1-trichlorosilane for 1h at room temperature in a vacuum chamber. To complete the construction of the molding setup, the acrylic mold was fixed on the glass plate.

The PDMS elastomer base (Dow Corning, MI) was polymerized by mixing 10:1 (w/w) ratio with curing agent. The solution was stirred and degassed under vacuum
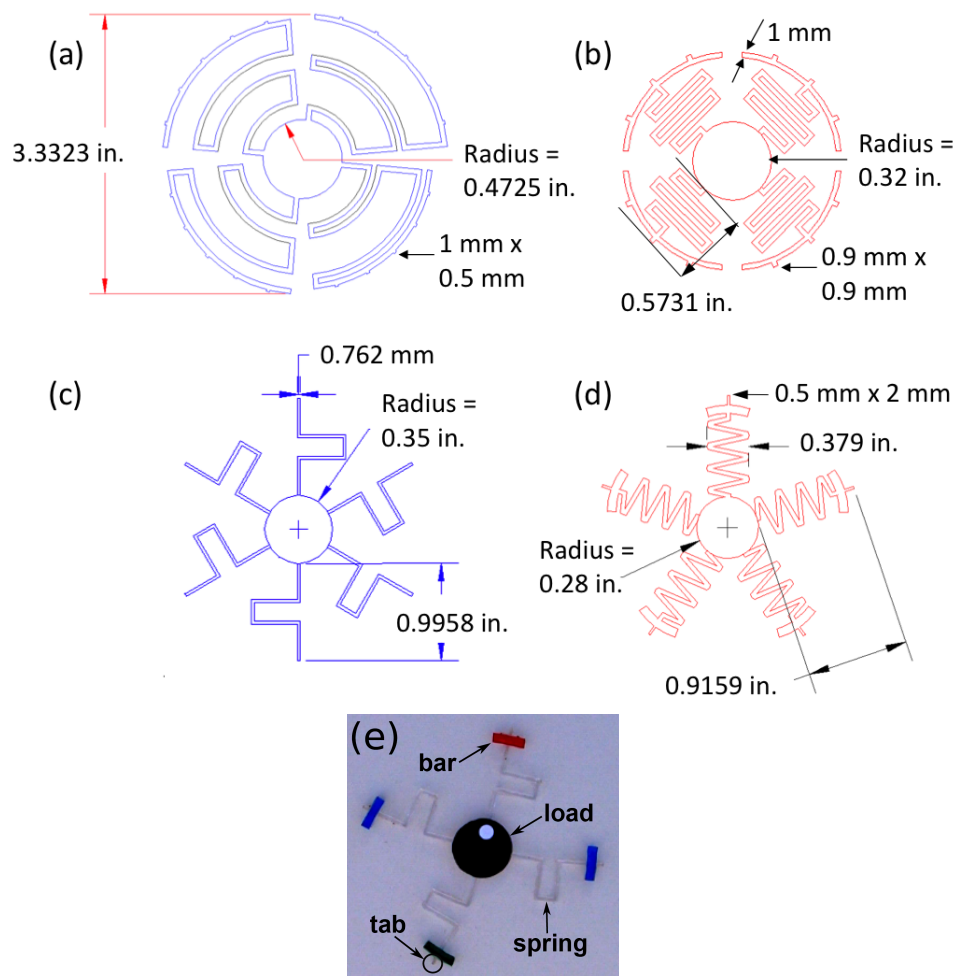
Figure 8.4: Artificial load designs: (a) and (b) were designed by David Cappelleri; (c)-(e) were designed by Spring Berman. In (e), the load and spring dimensions are the same as in (c) and the bar and tab dimensions are the same as in (d).

for 20 min, and the mold was filled with the viscous solution. Excess solution was scraped off the surface of the mold with a clean razor blade (Figure 8.8b). The polymer was cured for 2h at 80°C on a hot plate (Fig 8.8c). Once cured, the structure was carefully removed with a pair of tweezers.

8.2.3 Calibration

Quentin Lindsey and Travis Van Schoyck performed the calibration.
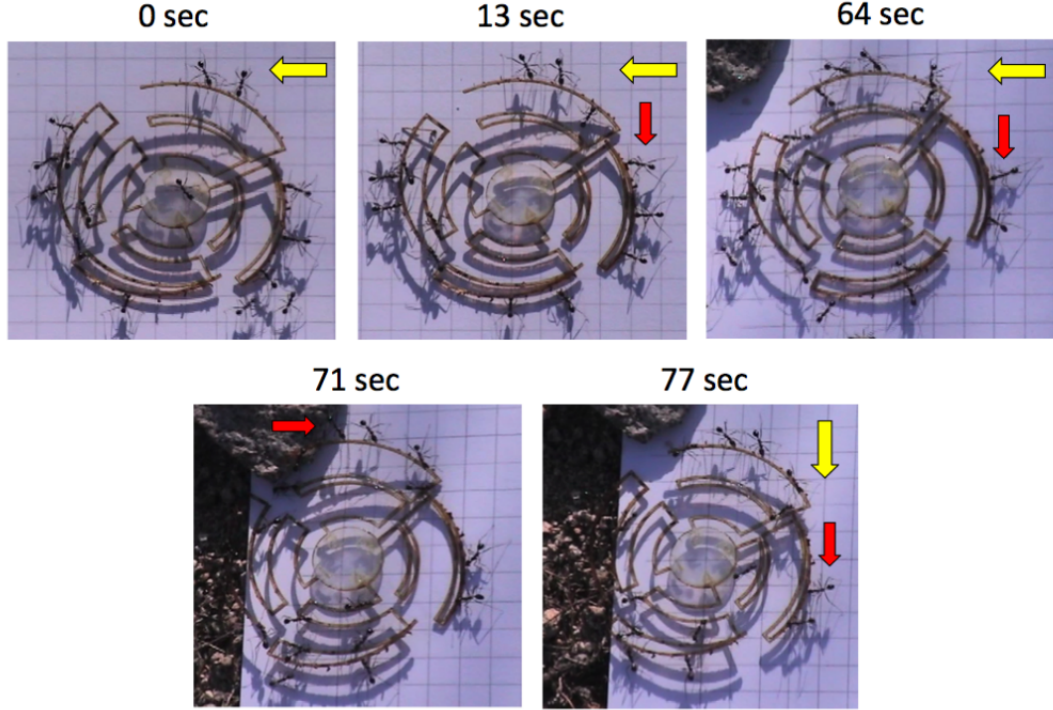
Figure 8.5: Video frames of ants transporting the Lexan load. The yellow arrow shows the direction of movement in each frame. As in the fig transport trials, the ants can negotiate obstacles by rotating the load. At 71 sec, the ant indicated by the red arrow climbs onto a rock, and the leftward movement of the load is redirected downward. The load is slowly rotated clockwise while it is pulled downward, and the ant, which had resisted the motion at 13 sec and 64 sec (as evidenced by its stretched hind legs), now contributes to the rotation by stepping sideways.

The calibration setup uses a micro/macro manipulation apparatus to create a lookup table of applied force vs. spring tip displacement. The load of the elastic structure is clamped to an acrylic mount such that one spring is in the appropriate configuration. This mount is attached to a motorized rotary stage (New Focus$^{\text{TM}}$), which is mounted on a H107 ProScan$^{\text{TM}}$II xy stage with 1 micron repeatability, step size of 5 microns, and travel of 4 in. $\times$ 3 in. A needle, which is attached to a load cell (Transducer Techniques®) in compression configuration and allowed to rotate freely, is inserted into the bar at the tip of the spring. Using an automated routine, the stage moves to predetermined locations on a rectangular grid, and a

Figure 8.6: Video frames showing configurations of ants that resulted in motion of the first PDMS prototype. The yellow arrow shows the direction of movement in each frame. Note that the ants are predominantly arranged on one side of the load, in contrast to the arrangement of ants on both sides of the lighter Lexan load.

force measurement is taken using the load cell at each location. To determine the radial spring force, the longitudinal axis of the spring is aligned parallel to the axis of the load cell. After the routine has swept the entire grid, the spring is returned to its rest configuration and rotated with the rotary stage 90 deg clockwise so that its latitudinal axis is aligned with the load cell in order to measure the tangential forces. Half of the grid is swept in that configuration so that the load cell is in compression. The spring is then rotated 180 deg counterclockwise so that the other half of the grid is swept.

8.2.4 Model

Spring Berman developed the model in this section.

We develop a simple model of the springs that can be used to quickly estimate their radial and tangential stiffnesses and to design a spring geometry with desired stiffness properties. Since the springs consist of flexible members that can undergo large deflections, we apply a *pseudo-rigid-body model*, a type of model that describes the behavior of compliant mechanisms [68].

Figure 8.9 shows the most general case of the spring model for structure 2. The lengths of the spring members in mm are $l_1 = 9.8$, $l_2 = l_4 = 11.9$, $l_3 = 5.6$, $l_5 = 10.9$, and the spring width is $h = 0.8$ mm. Each member i is represented as a rigid beam

141
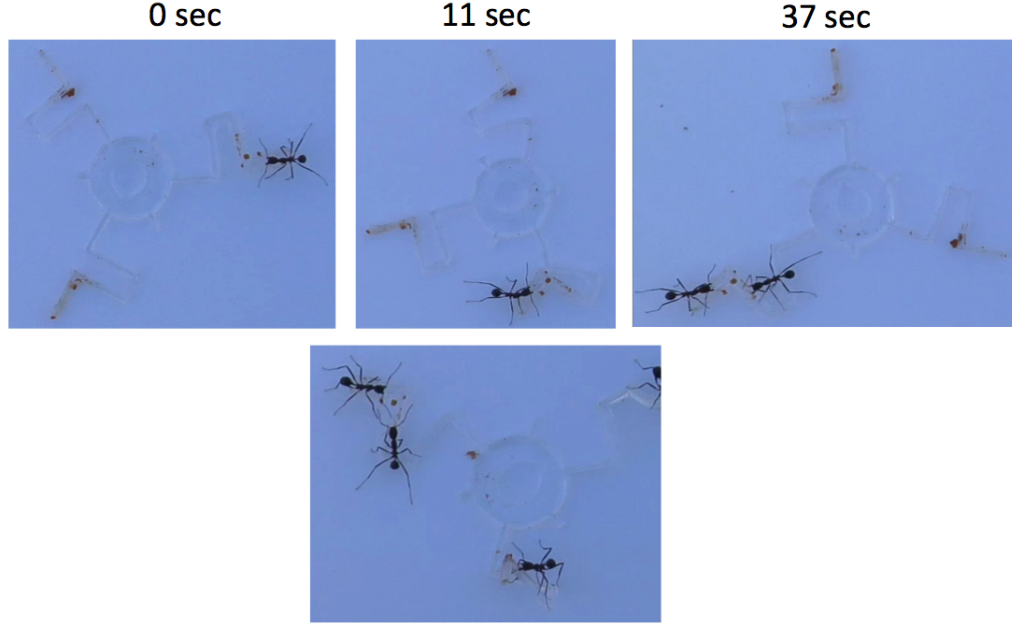
Figure 8.7: Video frames of ants transporting a thinner version of structure 2. In the top frames, an ant rotates the load so that it can pull it while walking backward toward the nest, a behavior observed by Sudd [143].

attached at a pin joint, where a torsional spring with spring constant k_i models the beam's resistance to deflection. Since the members are fixed-fixed segments, we define $k_i = 2\gamma K_\theta EI/l_j$ [68], where $j = 1$ for $i = 1$ and $j = 2$ otherwise. Here $\gamma = 0.85$, $K_\theta = 2.68$, E is Young's modulus, and I is the moment of inertia of the beam. Let $\Delta\theta_i$ be the angular displacement of beam i from its undeflected configuration. The moment at each pin joint is $\mathbf{T_i} = -k_i\Psi_i\hat{\mathbf{k}}$, where $\Psi_1 = \Delta\theta_1$, $\Psi_2 = \Delta\theta_2 - \Delta\theta_1$, $\Psi_3 = \Delta\theta_2 - \Delta\theta_3$, $\Psi_4 = \Delta\theta_4 - \Delta\theta_3$, $\Psi_5 = \Delta\theta_4 - \Delta\theta_5$.

We define three cases of the applied force $\mathbf{F}$ in polar coordinates, given in Table 8.1. We use the principle of virtual work to solve for the force-displacement relations. In each case, we model the spring as a 1-DOF system with generalized coordinate θ_q and include a subset $\mathcal{I}$ of the torsional springs, corresponding to members that display the most noticeable compliance over the calibration displacement range. We compare the force-displacement models to data from seven calibration trials on the same spring. The calibrated forces are obtained as x and y components and must
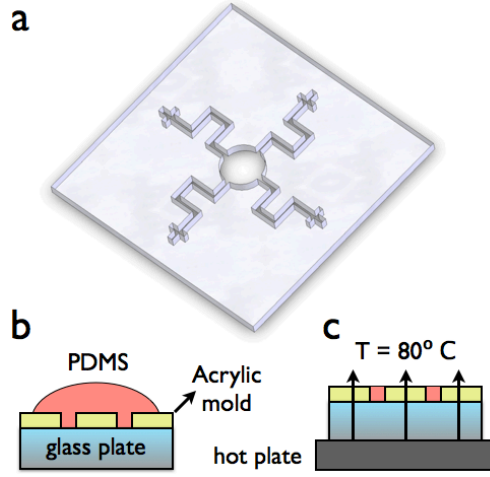
Figure 8.8: Elastic structure fabrication process. *Figure by Mahmut Selman Sakar.*

Table 8.1: Spring model parameters for different loading conditions

	$\mathbf{F}$	θ_1	θ_2	θ_3	θ_4	θ_5	$\mathcal{I}$
1	$F_r \mathbf{e}_r,\ \theta = 0$	0	θ_q	0	$2\pi - \theta_q$	0	$\{2,...,5\}$
2	$F_\theta \mathbf{e}_\theta,\ \theta \geq 0$	0	θ_q	$\theta_q - \frac{\pi}{2}$	$\theta_q + \pi$	$\theta_q - \frac{\pi}{2}$	$\{2\}$
3	$F_\theta \mathbf{e}_\theta,\ \theta < 0$	θ_q	$\theta_q + \frac{\pi}{2}$	θ_q	$\theta_q + \frac{3\pi}{2}$	θ_q	$\{1\}$

be transformed into F_r, F_θ beforehand. For cases 2 and 3, the F_θ measurements are interpolated along the circles $(x - l_1)^2 + y^2 = (l_3 + l_5)^2$ and $x^2 + y^2 = (l_1 + l_3 + l_5)^2$, respectively.

Case 1

Let r denote the spring displacement from its undeformed state. We solve for F_r in terms of θ_2, which is related to r geometrically:

$$F_r = \frac{\pi/2 - \theta_2}{2l_2 \sin(\theta_2)} \sum_{i=2}^{5} k_i, \quad \theta_2 = \cos^{-1}\left(\frac{r}{2l_2}\right). \tag{8.1}$$

Using calibration data, the least squares estimate of E is 1.66 MPa, which is within the typical range for PDMS (360 kPa to $\sim$3 MPa) [25]. Figure 8.10a shows that Equation (8.1) closely matches the calibration curve.

143
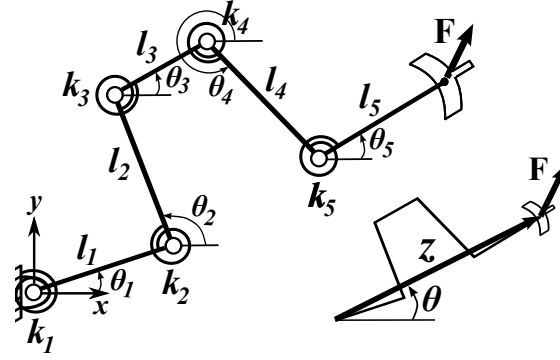
Figure 8.9: General pseudo-rigid-body model of a spring on structure 2.

Case 2

We solve for F_θ in terms of θ_2. Then from $l_1 + (l_3 + l_5)\sin\theta_2 = z\cos\theta$, where $z = ||\mathbf{z}||$:

$$F_\theta = \frac{k_2}{z - l_1\cos\theta}\cos^{-1}\left(\frac{z\cos\theta - l_1}{l_3 + l_5}\right), \quad \theta \geq 0. \tag{8.2}$$

As Figure 8.10b shows, Equation (8.2) matches the averaged calibration data fairly well. Note that the measured F_θ has much greater variability than F_r.

Case 3

We solve for F_θ in terms of θ_1, which due to the constraints on the other members is equal to θ:

$$F_\theta = k_1\theta/z, \quad \theta < 0. \tag{8.3}$$

Figure 8.10b shows that the model underestimates $|F_\theta|$ in this case. A closer fit to the data can be achieved by multiplying k_1 by 5, as shown by the light dashed plot.

144
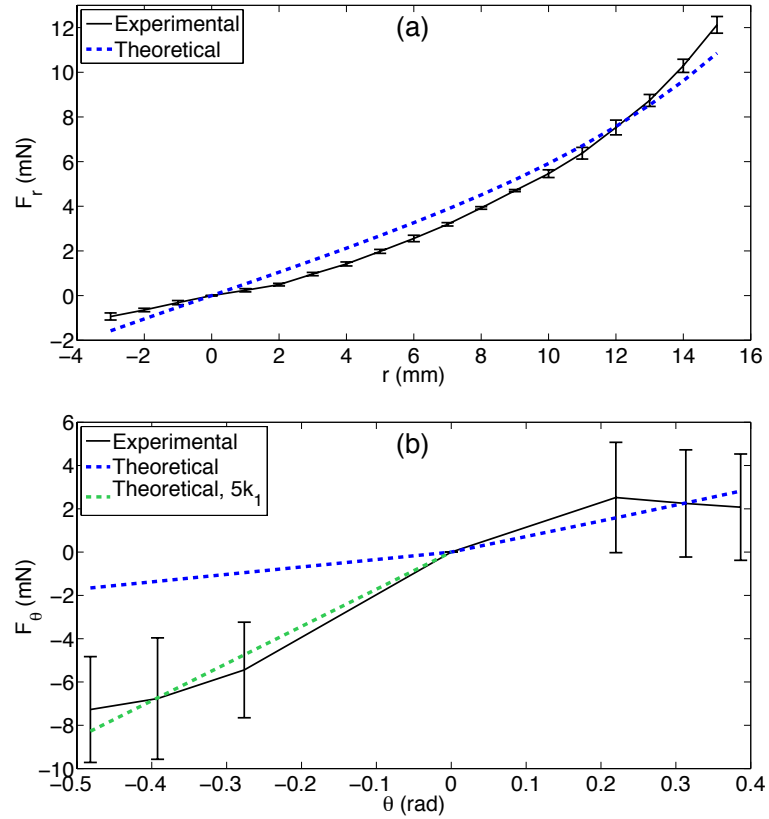
Figure 8.10: Experimental and theoretical force-displacement curves: (a) case 1, (b) cases 2 and 3. Measurements are averaged over seven calibration trials. Error bars show standard deviations.

8.3 Experiments

8.3.1 Video recording of group retrieval

Prof. Stephen Pratt conducted the experiments.

Experiments were performed in *A. cockerelli*'s natural Sonoran desert environment in South Mountain Park, Phoenix, Arizona using the setup shown in Figure 8.11. Ants were induced to retrieve an elastic structure placed approximately one meter from an active nest entrance. The tabs at the end of the springs were made attractive to the ants by smearing them with fig paste. The structure was placed on a flat wooden board (50 cm × 40 cm) lined with ivory copy paper. This provided a

Figure 8.11: Experimental setup for transport trials with ants.

smooth surface across which the ants could drag the structure, as well as a featureless background to aid in automated video tracking. A high definition video camera (Panasonic HVX200 or Canon HG20) was fixed above the structure on the vertical post of a copy stand. Its field of view encompassed a region of approximately 35 cm × 20 cm. A screen blocked direct sunlight from this region, to avoid shadows that could hinder video tracking. Fig baits in the vicinity of the structure were used to attract foragers. Once an ant discovered the structure, she soon initiated recruitment after unsuccessful attempts at moving it herself. A group of four to six ants then cooperated to move the structure toward the nest.

8.3.2 Video data image processing

Quentin Lindsey did the image processing.

In order to extract the structure configuration and the spring deformations from the video recordings of the transport, certain features of the structure were tracked, including the load and the tips of the springs. The load was covered with a black

circular label marked with a small white circle offset from the center. Thresholding and blob detection were employed using Swistrack [106], a general purpose vision processing software, to label and track the load center and the dot. The load orientation was obtained by comparing the relative position of the centers of the dot and the load. If the orientation could not be determined for a particular time step, the corresponding data was disregarded because the load orientation is a key feature used to track the spring deformations. The bar on each spring was marked with a red, blue, or green label and was tracked using similar methods. The deformation of a spring was calculated from the position of its label, the load pose, and other geometric quantities.

8.3.3   Results

Mechanics of transport

We found that typical inertial forces $m\ddot{\mathbf{r}}$ were on the order of 10^{-4} mN, which was negligible compared to the friction force on the structures, $\mu mg = 22$ mN, where $\mu = 1.4$ was measured from inclined plane tests with the structures. This indicates that the structures undergo quasi-static motion. The structures were often observed to move by stick-slip, which resulted in noisy data on load speed. The average ant force $||\mathbf{F}_i||$ was calculated to be 9.2 ± 6.2 mN (sample size $n = 7177$).

Cooperative features of transport

Figure 8.12 and 8.14 show snapshots of recorded transports with structures 1 and 2, respectively, as well as the evolution of the load configuration over both trials. Figure 8.13 shows the load speed and angular velocity over time for the trial in Figure 8.12. Figure 8.15 displays the sum of the ant *interaction forces* [86] for the trial in Figure 8.14, each defined as the projection of the difference in forces applied by two ants

onto their relative position vector:

$$||\mathbf{F}_{ij}^{int}|| = (\mathbf{F}_i - \mathbf{F}_j) \cdot (\mathbf{r}_i - \mathbf{r}_j)/||\mathbf{r}_i - \mathbf{r}_j|| \ . \tag{8.4}$$

This quantity is a measure of the degree of cooperation between two ants; it is zero when ants are perfectly cooperating.

We see evidence of an initial transport phase of low coordination among the ants followed by a more highly coordinated phase, as has been documented in the literature on group transport in ants (see Section 8.1). Figure 8.13 shows that structure 1 initially moves slowly on average, and then its speed sharply increases to significantly higher values at $t = 68$ s. As Figure 8.12 reveals, the transition between these phases of speed is initiated by the ants' reorientation of the structure a few degrees counterclockwise. The trial with structure 2 indicates that the phases can also be characterized in terms of the sum of the interaction forces, with the transition occurring as a sizable reduction in this quantity. Figure 8.15 shows that this sum decreases to about half its initial value on average after $t = 32$ s. From the snapshots in Figure 8.14, this increase in cooperation is apparent in the reorientation of the ants to pull with larger force components toward the nest during the second phase.

Load speed saturation with increased group size

Figure 8.16 shows that three ants are needed to move structure 2, and that when more than five ants participate in transport, there is on average no increase in the load speed. This is due to increased interference between ants during transport with higher populations.

8.4 Micro-Continuous Model

Spring Berman and Quentin Lindsey defined the model in this section.

The transport strategy is represented as a hybrid system with probabilistic transitions between different task modes. There is a precedent for such a model in [102],
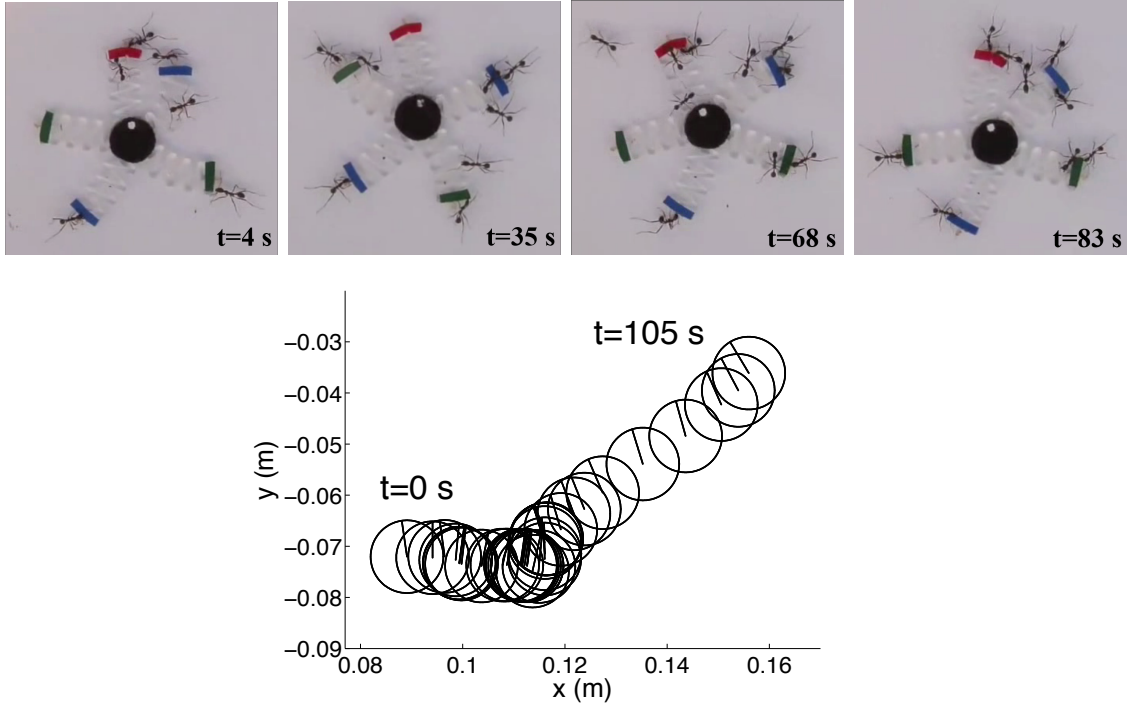
Figure 8.12: Snapshots of ant transport (top); load position and orientation (bottom).
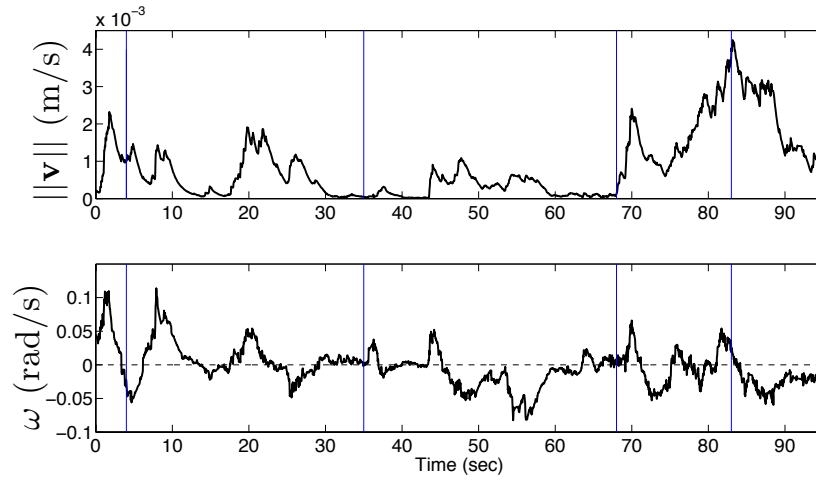
Figure 8.13: Load speed and load angular velocity over time for the trial in Figure 8.12. Blue lines indicate the times of the snapshots.
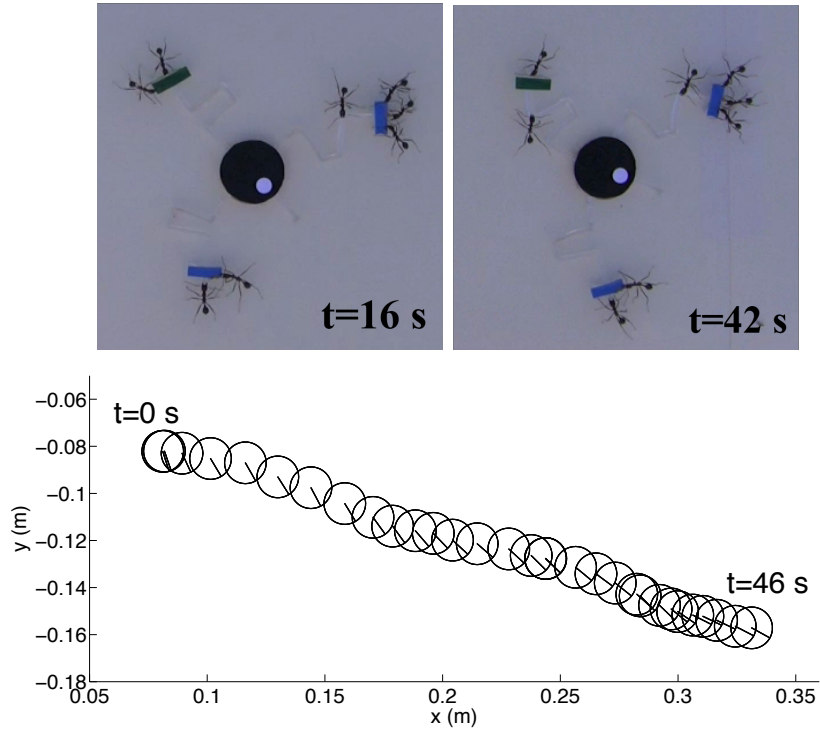
Figure 8.14: Snapshots of ant transport (top); load position and orientation (bottom).
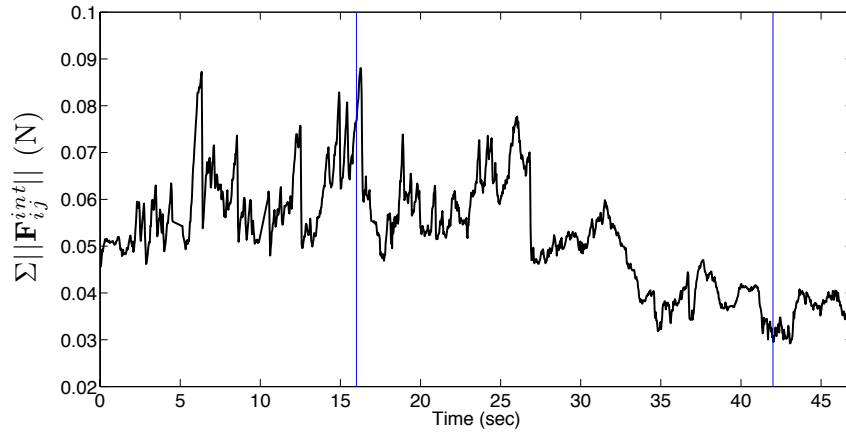
Figure 8.15: Sum of interaction force magnitudes over time for the trial in Figure 8.14. Blue lines indicate the times of the snapshots.
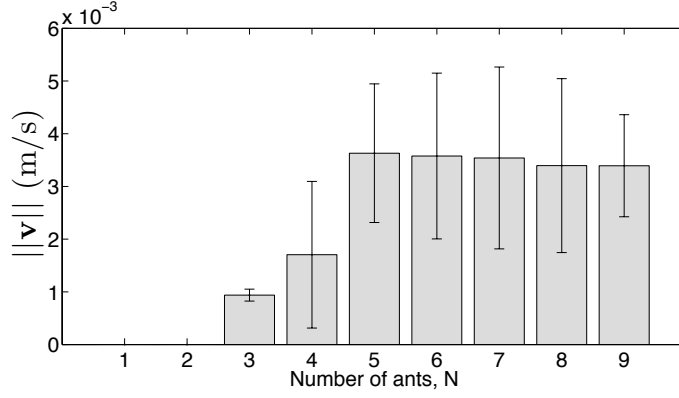
Figure 8.16: Average load speed as a function of transport group size for structure 2. Sample sizes for $N = 4 - 9$ are $n = 561, 453, 4154, 2448, 1866, 182$. Data for $N = 1 - 3$ were obtained manually. Error bars show standard deviations.

where collective transport in *Oecophylla longinoda* ants is modeled as a Markov chain using experimentally determined parameters. Figure 8.17 illustrates the notation that we use in describing the model.

We consider a population of N ants, each represented as a point-mass agent governed by a kinematic model,

$$\dot{\mathbf{q}}_i = \mathbf{u}_i \tag{8.5}$$

where $\mathbf{q}_i = [\mathbf{r}_i \; \theta_i]^T = [x_i \; y_i \; \theta_i]^T$ denotes the position of a point associated with ant i and the ant's orientation, both in an inertial frame, and $\mathbf{u}_i = [u_i^x \; u_i^y \; u_i^\theta]^T$ is the vector of control inputs.

We model the ants as switching between two behavioral modes, each corresponding to a different $\mathbf{u}_i$. In the behavior *Search for grasp point*, an ant i moves toward the structure in search of a protrusion to latch onto with its mandibles. If the outermost tip of a spring enters the sensing range of the ant, the ant latches onto the spring with probability p_1. Once attached, the ant is in the *Transport* mode and pulls on the spring in a manner that directs the structure toward its estimate of the nest. The ant reverts to mode *Search for grasp point* with a probability per time step $p_{2,i}(t)$, which is defined as a sigmoid function of the magnitude of the force $\mathbf{F}_i(t)$
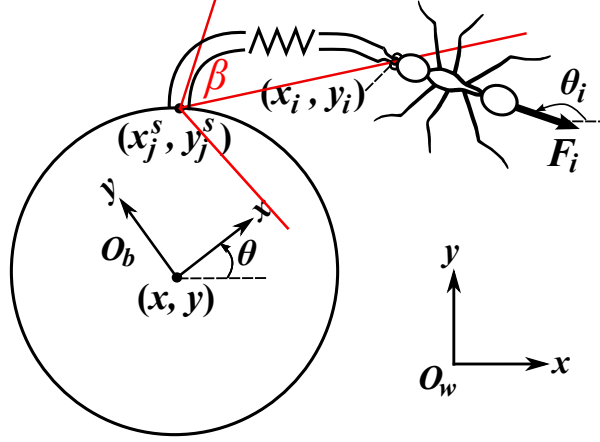
Figure 8.17: Notation for elastic structure with transporting ant (only one spring is shown).

that the ant applies to the spring at time t:

$$p_{2,i}(t) = \left(1 + e^{-\gamma(||\mathbf{F}_i(t)||-0.8F_{max})}\right)^{-1} ,$$ (8.6)

where F_{max} is the maximum force that the ant can apply, measured from the video data to be 150 mN, and $\gamma > 1$. We chose a sigmoid function to reflect the ants' tendencies to release the spring more often at large deflections and to occasionally lose their grip or lose interest even at low deflections. The ants stop moving when $||\mathbf{r} - \mathbf{r}_n|| \leq \rho$, where $\mathbf{r} = [x\ y]^T$ and $\mathbf{r}_n = [x_n\ y_n]^T$, the position of the nest, for some small constant ρ.

Search for grasp point

We define v_i and ω_i as the forward velocity and angular velocity, respectively, of ant i. An ant's estimates of the structure position (x, y), given by $(\hat{x}, \hat{y})$, are defined as values of the random variables $\hat{X} \sim \mathcal{N}(x, \sigma^2)$ and $\hat{Y} \sim \mathcal{N}(y, \sigma^2)$, where σ^2 is a specified variance. These values are updated at each time step. The ant's desired orientation θ_i^d is defined as the angle of the vector from (x_i, y_i) to $(\hat{x}, \hat{y})$. A

proportional controller is used to steer the ant's orientation to θ_i^d:

$$\omega_i = k e_i^\theta , \quad e_i^\theta = (\theta_i^d - \theta_i) \in [-\pi, \pi] . \tag{8.7}$$

A feedback linearization scheme is used to relate v_i and ω_i to the linear velocities of (x_i, y_i). Hence the control inputs are

$$\begin{bmatrix} u_i^x \\ u_i^y \\ u_i^\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -r \sin \theta_i \\ \sin \theta_i & r \cos \theta_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} , \tag{8.8}$$

where r is a small offset distance along the longitudinal axis of the ant.

Transport

An ant i in this mode pulls on its attachment point, whose coordinates are given by (x_i, y_i), with a force $\mathbf{F}_i$ that is directed along its longitudinal axis. Let $\mathbf{n}_i^f$ be the unit vector defining the direction of this force, and let $\mathbf{n}_i^n$ be the unit vector from (x_i, y_i) toward the nest coordinates, (x_n, y_n). The angles of these vectors in the inertial frame are $\theta_i + \pi$ and θ_i^n, respectively. We also define θ_j^s as the angle in the inertial frame of the vector from (x_j^s, y_j^s), the point where spring j, the spring that the ant is grasping, is attached to the load, to (x_i, y_i). To emulate realistic ant behavior, we specify that at each time t, the ant can begin to reorient itself to any angle between $\theta_i^{min}(t) = \theta_j^s(t) - \beta$ and $\theta_i^{max}(t) = \theta_j^s(t) + \beta$, where $\beta \leq \pi/2$. Similarly to Equation (8.7), a proportional controller steers θ_i:

$$\omega_i = k e_i^\theta , \quad e_i^\theta = (\theta_i^d - (\theta_i + \pi)) \in [-\pi, \pi] . \tag{8.9}$$

The desired orientation θ_i^d at a given time depends on the direction of the nest relative to the cone of possible ant orientations defined by β. Let $\mathbf{R}(\theta)$ be the 2D rotation matrix. Then θ_i^d is defined as

$$\theta_i^d = \begin{cases} \theta_i^n & \text{if} \quad \mathbf{n}_i^f \cdot \mathbf{n}_i^n \geq \cos \beta \\ \theta_i^{max} & \text{if} \quad \mathbf{R}(\frac{\pi+\beta}{2}) \mathbf{n}_i^f \cdot \mathbf{n}_i^n \geq \cos \frac{\pi-\beta}{2} \\ \theta_i^{min} & \text{if} \quad \mathbf{R}(-\frac{\pi+\beta}{2}) \mathbf{n}_i^f \cdot \mathbf{n}_i^n \geq \cos \frac{\pi-\beta}{2} \end{cases} \tag{8.10}$$

153

The control inputs are defined as follows:

$$\mathbf{u}_i = [-v_i \cos \theta_i \quad -v_i \sin \theta_i \quad \omega_i]^T \ . \tag{8.11}$$

8.5 Abstraction, Analysis, and Synthesis

Spring Berman developed the material in this section.

We consider a scenario in which a swarm of robots, such as the ones in Figure 8.1, is available to transport multiple identical payloads throughout an arena with area A. Our objective is to synthesize controllers for the robots to quickly form transport groups in a target distribution of sizes and manipulate the loads to specified locations. For instance, we may want the groups to contain the minimum number of robots needed to produce a certain payload speed, so as to free robots for other tasks. The maximum allowable number of robots in a group is denoted by N_T.

The robots' motion is defined by the micro-continuous model in Section 8.4. Note that compliance in the robot gripper mechanisms replaces compliance in the attachment points (the springs, which are specific to the loads in the ant experiments). We assume that robots can determine the number of robots in a transport group through local sensing. In order to abstract the system to an ODE model as described in Chapter 3, we specify that the robots and payloads are uniformly randomly distributed throughout the arena. Toward this end, the robots in *Search for grasp point* mode execute a random walk with constant speed until the payload enters their sensing range, at which point they follow the controllers (8.7), (8.8).

To achieve our objective, we can design the stochastic control policies in the micro-continuous model using our modeling, analysis, and synthesis framework. A robot in *Search for grasp point* mode is denoted by species Y, and the combination of a payload and a group of $i \in \{0, ..., N_T\}$ robots manipulating it in *Transport* mode is denoted by species Z_i. We define a CRN that describes the robot actions in the

154

micro-continuous model as follows,

$$Y + Z_{i-1} \overset{k_i^+}{\underset{k_i^-}{\rightleftharpoons}} Z_i, \quad i = 1, ..., N_T. \tag{8.12}$$
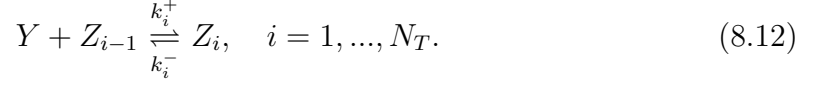
k_i^+ is the rate constant at which a robot finds and latches onto a payload being transported by $i - 1$ robots, and k_i^- is the rate at which the robot decides to detach from the payload. These rate constants are defined similarly to the assembly and disassembly rate constants in Equation (7.2):

$$k_i^+ = A \ p^e \ p_i^a \ p_i^+ \ , \quad k_i^- = p_i^- \ . \tag{8.13}$$

p^e is the probability per unit time that a robot encounters a payload. The uniform distribution of robots and payloads throughout the arena allows us to calculate p^e as c_{ij}^e, defined by Equation (3.4). p_i^a is the probability of a robot successfully latching onto the payload and can be measured from simulations. p_i^+ is the probability of a robot deciding to grasp a payload with a transport group of $i - 1$ robots, and p_i^- is the probability per unit time of a robot deciding to detach from such a payload. These two probabilities are the tunable parameters of the system.

Since the CRN (8.12) contains bimolecular reactions, it abstracts to a multi-affine macro-continuous model given by Equation (3.15). The species vector $\mathbf{x} = [y \ z_0 \ z_1 \ ... \ z_{N_T}]^T$ contains the concentrations of searching robots and payloads with transport groups of different sizes, and $\mathbf{y}(\mathbf{x}) = [yz_0 \ z_1 \ ... \ yz_{N_T-1} \ z_{N_T}]^T$. We can show that this model has a unique, globally asymptotically stable equilibrium using a proof similar to the one in Section 7.2.2. The proof uses the property that the CRN is deficiency zero since it has $2N_T$ complexes, N_T linkages classes, and rank N_T, which is determined by observing the linear independence of the N_T vectors $\mathbf{m}_i - \mathbf{m}_j$ associated with the *unordered* complex pairs (i, j) (recall that $\mathbf{m}_i$ is column i of matrix $\mathbf{M}$ in model (3.15)). Given this result, we can specify a target distribution $\mathbf{x^d}$ and use optimization methods similar to the ones in Section 7.3.1 to compute the probabilities p_i^+, p_i^-, $i = 1, ..., N_T$, that minimize the system convergence time to $\mathbf{x^d}$.

8.6 Conclusions

We have presented a study of the mechanics of cooperative prey retrieval in *A. cockerelli* ants. We conducted experimental trials with vision-based force sensors and developed a behavioral model of ant transport based on observations. The experimental data show that this type of distributed transport system is characterized by an initial disordered phase that transitions to a coordinated phase of increased payload speed and a higher degree of cooperation among the transporting agents. The data also demonstrate that additional transporters do not affect the progress of the task after a threshold population is reached. Our micro-continuous model of ant transport can be used to define controllers that produce a fully decentralized cooperative manipulation strategy in a multi-robot system. We can apply our modeling, analysis, and synthesis methodology to control populations of transport groups in a swarm robotic system with many payloads. Plans for continued study of cooperative retrieval in ants and an expanded transport model, as well as further approaches to analysis and synthesis for this type of system, are discussed in Chapter 10.

Chapter 9

Extension to Spatially Inhomogeneous Systems

We now remove the assumption that the system is well-mixed and consider systems in which robots and objects with which they interact are arbitrarily distributed throughout a domain, the robots have a deterministic velocity field superimposed on their diffusive motion, and the robot movement is slow relative to the rates of the reactions that define task transitions. Now the species concentrations x_s, $s = 1, ..., S$, are functions of position, denoted by $\mathbf{q} \in Y_r \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, as well as time, t. The macro-continuous model of the system is given by a set of *advection-diffusion-reaction* (ADR) equations [33], which describe the conservation of chemical species in a fluid:

$$\frac{\partial x_s}{\partial t} + \nabla \cdot (\mathbf{v}(\mathbf{q}) x_s) = D_s \nabla^2 x_s + R_s(\mathbf{x}), \quad s = 1, ..., S \qquad (9.1)$$

Here $\mathbf{v}(\mathbf{q})$ is a velocity field that specifies the deterministic robot motion, D_s is the diffusion coefficient of species s, and $R_s(\mathbf{x})$ is the function of reaction rates ν_i given by row s of Equation (3.16), evaluated at local concentrations $\mathbf{x}(\mathbf{q})$. When $R_s(\mathbf{x}) = 0$ this equation is known as the *advection-diffusion equation*, and when $\mathbf{v} = \mathbf{0}$ it is known as the *reaction-diffusion* equation.

To synthesize a target collective behavior in such a system, our design parameters

are now $\mathbf{v}(\mathbf{q})$ and D_s in addition to the reaction rate constants k_{ij}. In this chapter, we address the modeling and analysis components of our methodology for a fixed set of these parameters. In order to define a micro-continuous model that abstracts to the ADR equations (9.1) in the limit of infinite species populations, we specify the robots' motion using the technique of random walk particle tracking [34, 132], a method from statistical physics, and implement the reactions that govern robot task-switching using the formulation given in [9]. The general ADR equations that define the macro-continuous model cannot be solved analytically, and instead must be solved using numerical techniques [70]. A widely applied technique is a grid or mesh-based method such as finite difference, finite volume, and finite element methods [29]. A grid-based method may be Eulerian, in which the grid is fixed in space, Lagrangian, in which the grid moves and deforms with the material, or employ a combination of features from both. Grid-based methods are unsuitable for systems that consist of discrete physical entities (such as robots) rather than a continuum [104], and they can suffer from numerical dispersion and artificial oscillations when simulating advection-dominated problems [63, 132], which we may want to design. A more appropriate technique to use for our type of system is a meshfree method [103], which solves integral equations and PDE's with all kinds of boundary conditions using a set of arbitrarily distributed nodes or particles without imposing connectivity among these elements with a mesh. We numerically solve the ADR equations using smoothed particle hydrodynamics, a meshfree particle method that was originally developed to simulate astrophysical phenomena and has been extended to a variety of problems in computational fluid dynamics and solid mechanics [104, 105].

As an illustration of our methodology, we develop micro- and macro-continuous models of a spatially inhomogeneous part retrieval scenario for an assembly system and compare the species concentration fields from both models, where the latter is solved using smoothed particle hydrodynamics.

9.1 Micro-Continuous Model

Random walk particle tracking (RWPT) is a Lagrangian approach to solving advection-diffusion problems. This technique divides the mass of a system among a collection of N particles, each of which is assigned a mass m_i. We consider a particle i to represent a robot, which is modeled as a point kinematic agent with position $\mathbf{q}_i \in Y_r$, and $m_i = m \ \forall i$ to be an arbitrary quantity associated with each robot. The displacement of a particle i is governed by a drift term related to advection and a Brownian motion that drives diffusion. It can be computed using an Itō-Taylor integration scheme [45]:

$$\mathbf{q}_i(t + \Delta t) = \mathbf{q}_i(t) + \mathbf{v}(\mathbf{q}_i, t)\Delta t + \mathbf{B}\mathbf{Z}(t)\sqrt{\Delta t} \ , \tag{9.2}$$

where Δt is the time step, $\mathbf{Z} \in \mathbb{R}^d$ is a vector of independent, normally distributed random variables with zero mean and unit variance, and $\mathbf{B}$ satisfies $2D\mathbf{I} = \mathbf{B}\mathbf{B}^T$, where D is the diffusion coefficient. This scheme is defined to yield equivalence between the advection-diffusion equation and the Fokker-Planck equation, which describes the time evolution of the probability density function of the position of a particle, in the limit $N \to \infty$ [34, 132].

The particle masses are mapped to concentration values as described in [5, 149]. The concentration distribution $x(\mathbf{q}, t)$ can be approximated by the smoothed integral interpolation

$$x(\mathbf{q}, t) = \int_{Y_r} \tilde{x}(\mathbf{q}', t) \, w(\mathbf{q} - \mathbf{q}') \, d\mathbf{q}', \qquad \tilde{x}(\mathbf{q}, t) = \sum_{i=1}^{N} m_i \delta(\mathbf{q} - \mathbf{q}_i(t)), \tag{9.3}$$

where $\delta(\mathbf{q})$ is the Dirac delta function and $w(\mathbf{q})$ is a projection function with finite support that satisfies $\int_{Y_r} w(\mathbf{q}) d\mathbf{q} = 1$ and is ideally invariant with respect to coordinate transformations. A particle approximation for $x(\mathbf{q}, t)$ can be defined as

$$x(\mathbf{q}, t) = \sum_{i=1}^{N} m_i(t) w(\mathbf{q} - \mathbf{q}_i(t)). \tag{9.4}$$

Most implementations use $w(\mathbf{q}) = 1/V$ for points within a small cube of volume V around $\mathbf{q}$ and $w(\mathbf{q}) = 0$ otherwise [63]. This corresponds to dividing the domain Y_r

into a grid of cubic cells with volume V_c, finding the mass of all particles inside each cell c, M_c, and defining the concentration at all points $\mathbf{q}$ inside cell c as $x(\mathbf{q},t) = M_c/V_c$.

The unimolecular reactions (3.7) and (3.8) are implemented the same way as in the spatially homogeneous case. For the bimolecular reactions (3.5) and (3.6), the probability c_{ij} that a random pair of reactants in complex i will react to become the product(s) in complex j in the next infinitesimal time interval dt is, as in the homogeneous case, a function of the probability of the reactants encountering each other in this time interval. The relative displacement of the reactants must now be taken into account in the encounter probability. Let $\mathbf{r} = \mathbf{q}_o(t) - \mathbf{q}_p(t)$ be the initial relative displacement of two reactants o and p and let $\Delta\mathbf{q} = \mathbf{q}_p(t+dt) - \mathbf{q}_o(t+dt) + \mathbf{r}$. The probability of the reactants occupying the same position at time $t+dt$ is $P(\Delta\mathbf{q} = \mathbf{r})$. Defining $p(\mathbf{r})$ as the probability density of $\Delta\mathbf{q}$, c_{ij} is given by [9]

$$c_{ij} = k_{ij} m p(\mathbf{r}). \tag{9.5}$$

We denote by d the radius of a spherical region around a robot that contains the vast majority of reactants that the robot is likely to reach in the next dt time units. This region may be a subset of the robot's sensing or communication range if it is large enough. In this case, a bimolecular reaction is implemented by having the robot compute c_{ij} for each potential reactant within the region, along with a uniformly distributed random number $u \in [0,1]$, and execute the reaction with the reactant for which $u < c_{ij} dt$, if any. Otherwise, this computation is executed as a simulation of the robot's random movement within the region and its subsequent decision whether to interact with a reactant that is then within its sensing or communication range. Upon reaction, a robot becomes another species, corresponding to a different type of particle, and may follow a different displacement equation of the form (9.2).

9.2 Macro-Discrete Model

Stochastic simulation algorithms for spatially inhomogeneous systems [91, 92, 94] are designed to simulate trajectories of species populations according to the Multivariate or Reaction-Diffusion Master Equation, defined in [45, 75]. These approaches extend the algorithms proposed by Gillespie [51] to systems that are divided into subvolumes, each assumed to be spatially homogeneous, with diffusion across neighboring subvolumes modeled as unimolecular reactions.

9.3 Macro-Continuous Model

This section discusses the numerical solution of the ADR equations using smoothed particle hydrodynamics (SPH). In this method, a function $f(\mathbf{q}, t)$ over a domain is represented in terms of its values at a set of arbitrarily distributed particles. This converts the governing PDE equations of a system into a set of ODE's, each of which describes the time evolution of a variable associated with a particle. The value of a variable at a particle is influenced by values at particles within a local neighborhood only.

The SPH formulation is derived by first approximating $f(\mathbf{q})$ with a smoothed integral interpolation, similarly to Equation (9.3):

$$f(\mathbf{q}) = \int_{Y_r} f(\mathbf{q}') \, W(\mathbf{q} - \mathbf{q}', h) \, d\mathbf{q}' \ . \tag{9.6}$$

Here $W(\mathbf{q} - \mathbf{q}', h)$ is a differentiable kernel function with smoothing length h and compact support. W satisfies the properties

$$\lim_{h \to 0} W(\mathbf{q} - \mathbf{q}', h) = \delta(\mathbf{q} - \mathbf{q}') \ , \tag{9.7}$$

$$\int_{Y_r} W(\mathbf{q} - \mathbf{q}', h) \, d\mathbf{q}' = 1 \ , \tag{9.8}$$

$$W(\mathbf{q} - \mathbf{q}', h) = 0 \quad \text{for} \quad ||\mathbf{q} - \mathbf{q}'|| > \kappa h \ , \tag{9.9}$$

where κ is a constant that defines the support domain of the kernel of point $\mathbf{q}$. The

kernel function is frequently defined as a Gaussian or a spline that approximates a Gaussian [104].

The integral (9.6) can be approximated using a Monte Carlo integration scheme as follows. The spatial distribution of the mass in the system is represented by a set of M particles. Particle i has position $\mathbf{r}_i$, mass m_i, and density ρ_i, where m_i/ρ_i is the volume associated with the particle. Replacing $d\mathbf{q}'$ in Equation (9.6) by this volume at the particle locations, the particle approximation for $f(\mathbf{q})$, which is performed at each time step, is given by

$$f(\mathbf{q}) = \sum_{i=1}^{M} \frac{m_i}{\rho_i} f(\mathbf{r}_i) W(\mathbf{q} - \mathbf{r}_i, h) . \qquad (9.10)$$

The error in approximating Equation (9.6) by Equation (9.10) is $O(h^2)$ [115]. Computing the gradient of $f(\mathbf{q})$ using the particle approximation entails differentiating the kernel W rather than the function itself.

In recent years, the SPH technique has been used to define decentralized motion control laws for robot swarms at the *micro-continuous* level to achieve pattern generation in obstacle-filled environments [122,123] as well as deployment, sensor coverage, patrolling, flocking, and formation control behaviors [119, 121]. These works model a swarm as a fluid, which may be subjected to an external force, and represent each robot as an SPH particle that has an associated position, velocity, mass, density, energy, and pressure. The robots use the SPH formulations of the governing equations for the conservation of mass, momentum, and energy, along with an equation of state for pressure, to update their associated quantities and compute their control law. The approach is scalable because these updates only require information from robots within a local neighborhood that corresponds to the support domain of W.

In contrast, our application of the SPH method at the *macro-continuous* level does not associate particles with robots, but rather employs the particles as computational elements that track concentration values at points in space. We use an SPH scheme similar to those that have been recently applied to model solute transport in

porous media [63, 146]. The velocity of particle i is given by the velocity field in the advection term of Equation (9.1),

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}(\mathbf{r}_i). \tag{9.11}$$

The diffusion term in Equation (9.1), $D_s\nabla^2 x_s$, could be evaluated by differentiating the interpolated concentration twice. However, the resulting expression contains the second derivative of W, which can be noisy and sensitive to particle disorder, so instead we use an SPH discretization of the Laplace operator that involves only first order derivatives of W [72]. This is derived by taking the Taylor series expansion for $x_s(\mathbf{r})$ in the neighborhood of $x_s(\mathbf{r}')$, neglecting terms of third and higher orders, and multiplying $x_s(\mathbf{r}) - x_s(\mathbf{r}')$ and the expansion by

$$F(\Delta\mathbf{r}) = \frac{(\Delta\mathbf{r}) \cdot \nabla_{\Delta\mathbf{r}} W(\Delta\mathbf{r}, h)}{||\Delta\mathbf{r}||^2}, \tag{9.12}$$

where $\Delta\mathbf{r} = \mathbf{r} - \mathbf{r}'$ and $\nabla_{\Delta\mathbf{r}} W(\Delta\mathbf{r}, h)$ is the directional derivative of W along $\Delta\mathbf{r}$, as given by Equation (5.3). Then, integrating over all $\mathbf{r}$, it can be found that the expansion reduces to $\nabla^2 x_s|_{\mathbf{r}'}$, and so we arrive at the equation

$$\nabla^2 x_s|_{\mathbf{r}'} = -2 \int_{Y_r} (x_s(\mathbf{r}) - x_s(\mathbf{r}')) \, F(\Delta\mathbf{r}) \, d\mathbf{r}. \tag{9.13}$$

The particle approximation (9.10) of this integral is

$$\nabla^2 x_s|_{\mathbf{r}_i} = -2 \sum_{j=1}^{M_s} \frac{m_j}{\rho_j} \, (x_s(\mathbf{r}_j) - x_s(\mathbf{r}_i)) \, F(\mathbf{r}_{ij}), \tag{9.14}$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. We replace ρ_j/m_j in this sum by n_j, the number of particles per unit volume, which is calculated as [146]

$$n_i = \sum_{j=1}^{M_s} W(\mathbf{r}_{ij}, h). \tag{9.15}$$

For a set of irregularly spaced particles, $n_i \neq n_j$ in general, which produces an asymmetry in the magnitude of the contribution of concentration from particle i to particle j and vice versa. To rectify this, n_j is replaced by either the arithmetic or

163

harmonic average of n_i and n_j [63]. Choosing the harmonic average and multiplying Equation (9.14) by D_s, the particle approximation of the diffusion term becomes

$$D_s \nabla^2 x_s|_{\mathbf{r}_i} = D_s \sum_{j=1}^{M_s} \frac{n_i + n_j}{n_i n_j} \left(x_s(\mathbf{r}_i) - x_s(\mathbf{r}_j) \right) F(\mathbf{r}_{ij}). \qquad (9.16)$$

The species concentrations in the reaction rate term of the ADR equations are SPH approximations evaluated at each particle. Thus, the SPH formulation of the ADR equation for species s at each particle $i = 1, ..., M_s$ is

$$\left. \frac{dx_s}{dt} \right|_{\mathbf{r}_i} = D_s \nabla^2 x_s|_{\mathbf{r}_i} + R_s(x_1(\mathbf{r}_i), ..., x_S(\mathbf{r}_i)). \qquad (9.17)$$

The SPH method is implemented by initializing the particle positions and concentrations and then numerically integrating Equations (9.11) and (9.17) using standard techniques such as Runge-Kutta methods and the Velocity Verlet scheme [104].

9.4 Application: Assembly System Part Retrieval

We consider a spatially inhomogeneous assembly scenario in two dimensions with robots, labeled species A, and one type of part, species B. A cluster of N_A robots are initially located to the left of a supply of N_B parts. The robots search for parts by moving with a deterministic component to the right superimposed with a random walk; the diffusive motion enables the swarm to cover a larger area. Upon encountering a part, a robot picks it up with a predefined probability, becoming species C, and then travels deterministically to a location to the right of the part supply. Sections 9.4.1 and 9.4.2 describe the micro-continuous and macro-continuous models of the time evolution of the species concentrations, $x_A(\mathbf{q}, t)$, $x_B(\mathbf{q}, t)$, and $x_C(\mathbf{q}, t)$, using the framework discussed in Sections 9.1 and 9.3. Section 9.4.3 evaluates the accuracy of the two models for a system with diffusion only, and Section 9.4.4 compares the models of the part retrieval to verify that the macro-continuous model is an accurate abstraction.

9.4.1 Micro-Continuous Model

We model the robots and parts as kinematic agents according to the strategy in Section 9.1. The positions of the robots and parts are denoted by $\mathbf{q}_i^s \in \mathbb{R}^2$, where $s \in \{A, B, C\}$ and $i \in 1, ..., N_s$, and are updated at each time step Δt according to variations on Equation (9.2):

$$\mathbf{q}_i^A(t + \Delta t) \;=\; \mathbf{q}_i^A(t) + \mathbf{v}\Delta t + \mathbf{BZ}\sqrt{\Delta t} \tag{9.18}$$

$$\mathbf{q}_i^B(t + \Delta t) \;=\; \mathbf{q}_i^B(t) \tag{9.19}$$

$$\mathbf{q}_i^C(t + \Delta t) \;=\; \mathbf{q}_i^C(t) + \mathbf{v}\Delta t. \tag{9.20}$$

A robot's retrieval of a part is represented by the reaction $A + B \rightarrow C$, with the rate constant denoted by k. Since the parts are stationary, the function $p(\mathbf{r})$ in Equation (9.5) is the probability density of $\Delta\mathbf{q} = \mathbf{q}_i^A(t+dt) - \mathbf{q}_i^A(t) = \mathbf{v}dt + \mathbf{BZ}\sqrt{dt}$, using Equation (9.18). This is a bivariate normal distribution with mean $\mathbf{v}dt$ and covariance matrix $2D_A dt\mathbf{I}$. The radius d is defined as $2(2D_A dt)^{1/2}$, twice the standard deviation of the robot's motion in the x and y directions.

9.4.2 Macro-Continuous Model

The evolution of the species concentrations over time is governed by the following ADR equations,

$$\frac{\partial x_A}{\partial t} + \nabla \cdot (\mathbf{v}x_A) \;=\; D_A\nabla^2 x_A - kx_A x_B \tag{9.21}$$

$$\frac{\partial x_B}{\partial t} \;=\; -kx_A x_B \tag{9.22}$$

$$\frac{\partial x_C}{\partial t} + \nabla \cdot (\mathbf{v}x_C) \;=\; kx_A x_B. \tag{9.23}$$

Our SPH formulation of the system uses two sets of particles, each of which is arranged on a lattice with particle spacing $\Delta r = h/a$, where a is a constant ([104], p. 130 suggests $a = 1.2$). One set has M_A particles with positions $\mathbf{r}_i$ that move with velocity $\mathbf{v}$, and the other has M_B particles with positions $\mathbf{s}_j$ that remain stationary.

The velocity $\mathbf{v}$ is defined as $[\Delta r/\Delta t \;\; 0]^T$, so that the positions of moving particles coincide with those of stationary particles when the lattices overlap. This allows the moving particles to use the value of x_B at overlapping stationary particles in the reaction rate computations, and vice versa for the value of x_A. The SPH equations (9.11) and (9.17) for the system are

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}, \quad i = 1, ..., M_A; \qquad \frac{d\mathbf{s}_j}{dt} = \mathbf{0}, \quad i = 1, ..., M_B \tag{9.24}$$

$$\left.\frac{dx_A}{dt}\right|_{\mathbf{r}_i} = D_A \nabla^2 x_A|_{\mathbf{r}_i} - \sum_{j=1}^{M_B} k x_A(\mathbf{r}_i) x_B(\mathbf{s}_j) \delta(\mathbf{r}_i - \mathbf{s}_j), \quad i = 1, ..., M_A \tag{9.25}$$

$$\left.\frac{dx_B}{dt}\right|_{\mathbf{s}_j} = \sum_{i=1}^{M_A} k x_A(\mathbf{r}_i) x_B(\mathbf{s}_j) \delta(\mathbf{r}_i - \mathbf{s}_j), \quad j = 1, ..., M_B \tag{9.26}$$

$$\left.\frac{dx_C}{dt}\right|_{\mathbf{r}_i} = \sum_{j=1}^{M_B} k x_A(\mathbf{r}_i) x_B(\mathbf{s}_j) \delta(\mathbf{r}_i - \mathbf{s}_j), \quad i = 1, ..., M_A. \tag{9.27}$$

Let $R = ||\mathbf{r}_{ij}||/h$. We define the kernel W as a cubic spline with compact support and a shape similar to a Gaussian:

$$W(\mathbf{r}_{ij}, h) = \gamma \cdot \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3 & \text{if} \quad 0 \le R < \kappa/2 \\ \frac{1}{6}(2 - R)^3 & \text{if} \quad \kappa/2 \le R < \kappa \\ 0 & \text{if} \quad R \ge \kappa \end{cases} \tag{9.28}$$

where $\gamma = \frac{15}{7\pi h^2}$ in two dimensions and $\kappa = 2$. This function has been widely used in the SPH literature [104].

At each time step, the particle positions and concentrations are computed by numerically integrating Equations (9.24)-(9.27) using the Euler method.

9.4.3 Model Validation for Diffusion

The accuracy of the concentration distributions produced by the micro-continuous model and the SPH solution of the macro-continuous model were tested for a one-species system undergoing diffusion only, for which an analytical solution can be

found. A similar comparison was done between concentration profiles from particle tracking and SPH methods in [63]. We select A as the species to be studied and set $\mathbf{v} = \mathbf{0}$, $k = 0$. The corresponding PDE, Equation (9.21), reduces to the heat equation. The initial concentration distribution is set to

$$x_A(\mathbf{q}, 0) = \exp \frac{||\mathbf{q} - \mathbf{a}||^2}{2\alpha}, \quad \mathbf{a} = [2.5 \ \ 2.5]^T, \ \alpha = 0.09 \tag{9.29}$$

The solution to the heat equation on an infinite domain with initial condition (9.29) is

$$x_A(\mathbf{q}, t) = \frac{\alpha}{\alpha + 2D_A t} \exp \frac{-||\mathbf{q} - \mathbf{a}||^2}{2(\alpha + 2D_A t)}. \tag{9.30}$$

The computations were run on the spatial domain $Y_r = [0 \ 5] \times [0 \ 5]$ with $D_A = 1$ and $\Delta t = 0.001$. In the SPH method, the particles were placed on an $\sqrt{M_A} \times \sqrt{M_A}$ lattice, and the initial values of $x_A(\mathbf{r}_i)$, $i = 1, ..., M_A$, were defined according to Equation (9.29). In the micro-continuous model, the domain was divided into a $g \times g$ grid. To compute the mass associated with each of the N_A robots, we evaluated the initial concentration (9.29) at the center $\mathbf{q}^c$ of each grid cell c and set $m = \frac{1}{N_A} \sum_{c=1}^{g^2} x_A(\mathbf{q}^c, 0) V_c$. To initialize the robot positions, we computed the number of robots per cell as $n_c = x_A(\mathbf{q}^c, 0) V_c / m$, rounded this number to the nearest integer, and distributed the n_c robots uniformly randomly inside the cell.

Figure 9.1 compares the maximum x_A computed over the domain by the micro-continuous model for different combinations of N_A and g, the SPH method for different combinations of M_A and h, and the solution (9.30) at the corresponding lattice points or cell centers. The maximum x_A values that were computed numerically are close to the theoretical values. Figure 9.2 plots the normalized error $\mu_A = \frac{1}{P} ||\mathbf{x}_A^{num} - \mathbf{x}_A^{theor}||$, where $P = g^2$ for the micro-continuous model and M_A for the SPH method, $\mathbf{x}_A^{num}$ is the vector of concentrations computed at each lattice point or cell center, and $\mathbf{x}_A^{theor}$ is the vector of the corresponding values of (9.30). These figures show that increasing N_A for fixed g and increasing M_A for fixed h results in more accurate concentration distributions. For $M_A = 10^4$, raising h, which increases
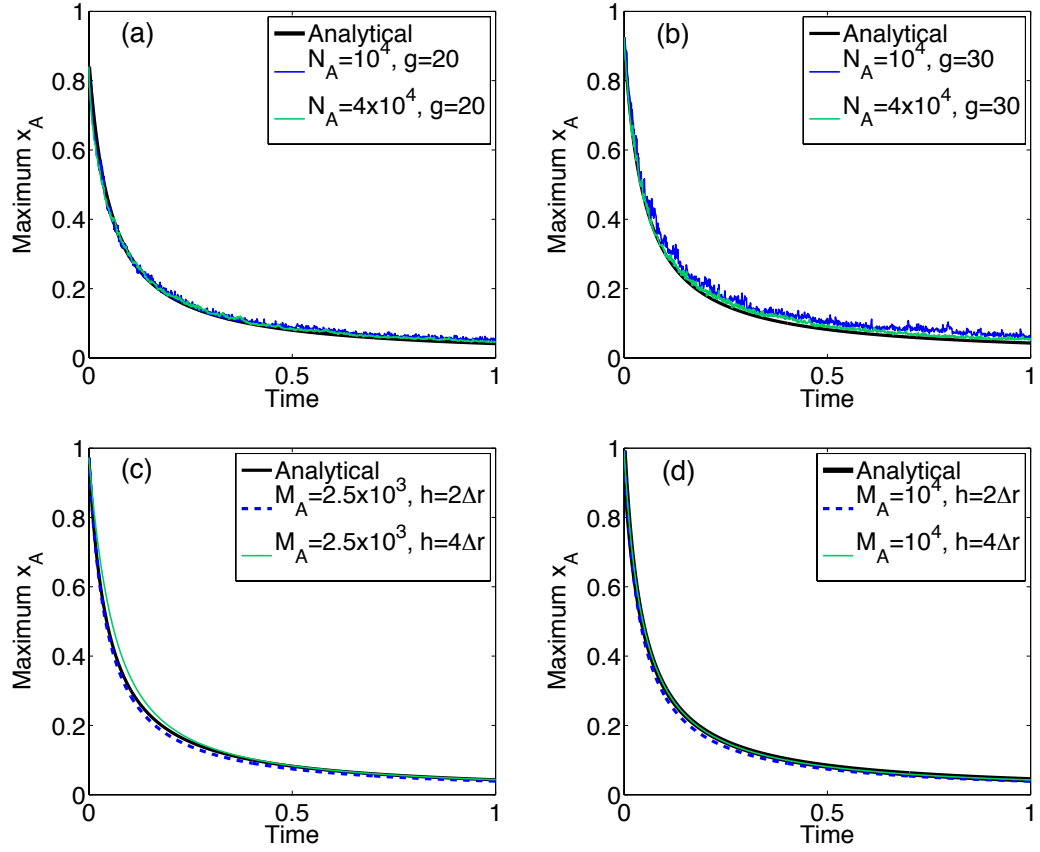
Figure 9.1: Maximum x_A over time computed by (a),(b) the micro-continuous model for four combinations of N_A, g and by (c),(d) the SPH method for four combinations of M_A, h. Each plot also shows the maximum value of the analytical solution (9.30) over time.

the number of particles in the support domain of W, increases accuracy at all points in time.

9.4.4 Model Comparison for Part Retrieval System

We ran the micro-continuous model and the SPH solution for the retrieval system described by Equations (9.21)-(9.23) with parameters $\mathbf{v} = [50 \;\; 0]^T$, $D_A = 1$, and $k = 350$. We use the initial concentration distributions (9.29) and

$$x_B(\mathbf{q}, 0) = \exp \frac{||\mathbf{q} - \mathbf{b}||^2}{2\beta}, \quad \mathbf{b} = [7.5 \;\; 2.5]^T, \; \beta = 0.09 \qquad (9.31)$$

$$x_C(\mathbf{q}, 0) = 0. \qquad (9.32)$$
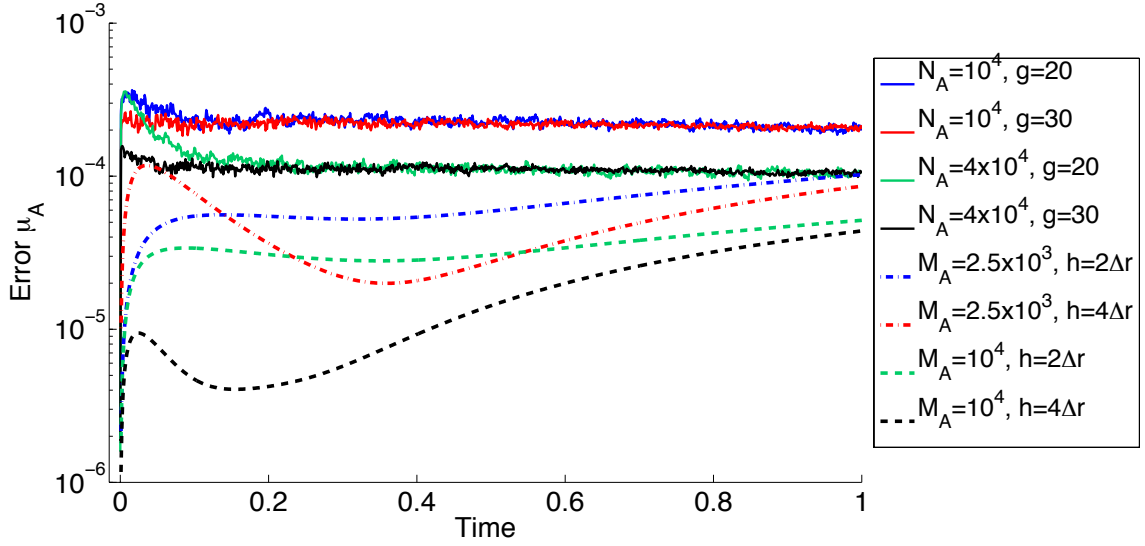
Figure 9.2: Error μ_A versus time of x_A computed by the micro-continuous model for four combinations of N_A, g and by the SPH method for four combinations of M_A, h.

We used the parameters in Figure 9.2 that yielded the lowest μ_A over time for each model: $M_A = 10^4$, $h = 4\Delta r$, $N_A = 4 \times 10^4$, and $g = 30$ (used to define the grid resolution). We set $M_B = M_A$ and $N_B = N_A$. In the SPH method, the moving particles were initialized as in Section 9.4.3, and the stationary particles were initialized similarly on the domain $[5\ 10] \times [0\ 5]$ with $x_B(\mathbf{s}_j, 0)$, $j = 1, ..., M_B$, defined by Equation (9.31). In the micro-continuous model, the positions of the robots and parts were initialized as described in Section 9.4.3 over a 90×30 grid. The time step was set to $\Delta t = 0.001$.

Figure 9.3 shows a sequence of snapshots of the micro-continuous model, and Figures 9.4-9.6 compare the $x_A(\mathbf{q}, t)$, $x_B(\mathbf{q}, t)$, and $x_C(\mathbf{q}, t)$ distributions computed by the micro-continuous model and the SPH method at the corresponding times. There is a close match between the two sets of concentration distributions, which demonstrates that macro-continuous model can be used to design the parameters $\mathbf{v}$, D_A, and k that will produce desired collective behaviors in the physical system.
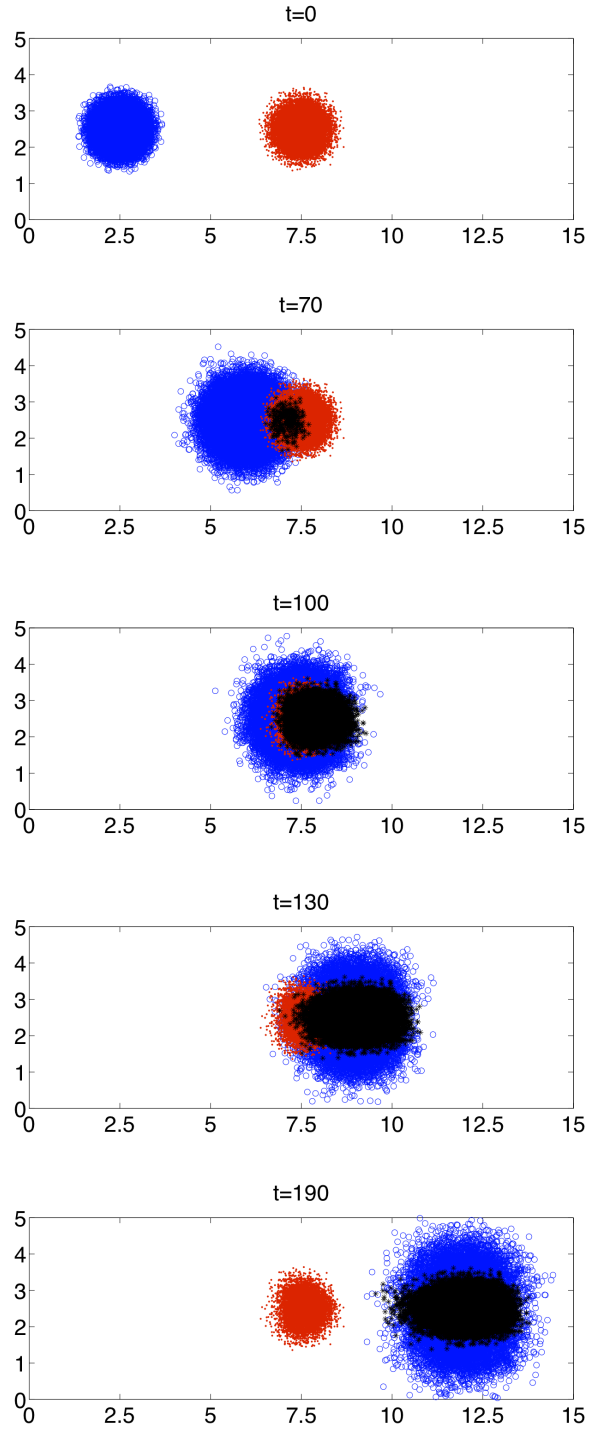
Figure 9.3: Snapshots of the micro-continuous model with $N_A = N_B = 10^4$ at $t = 0, 70, 100, 130, 190$. Blue circles are robots searching for a part, red dots are parts, and black stars are robots carrying parts.
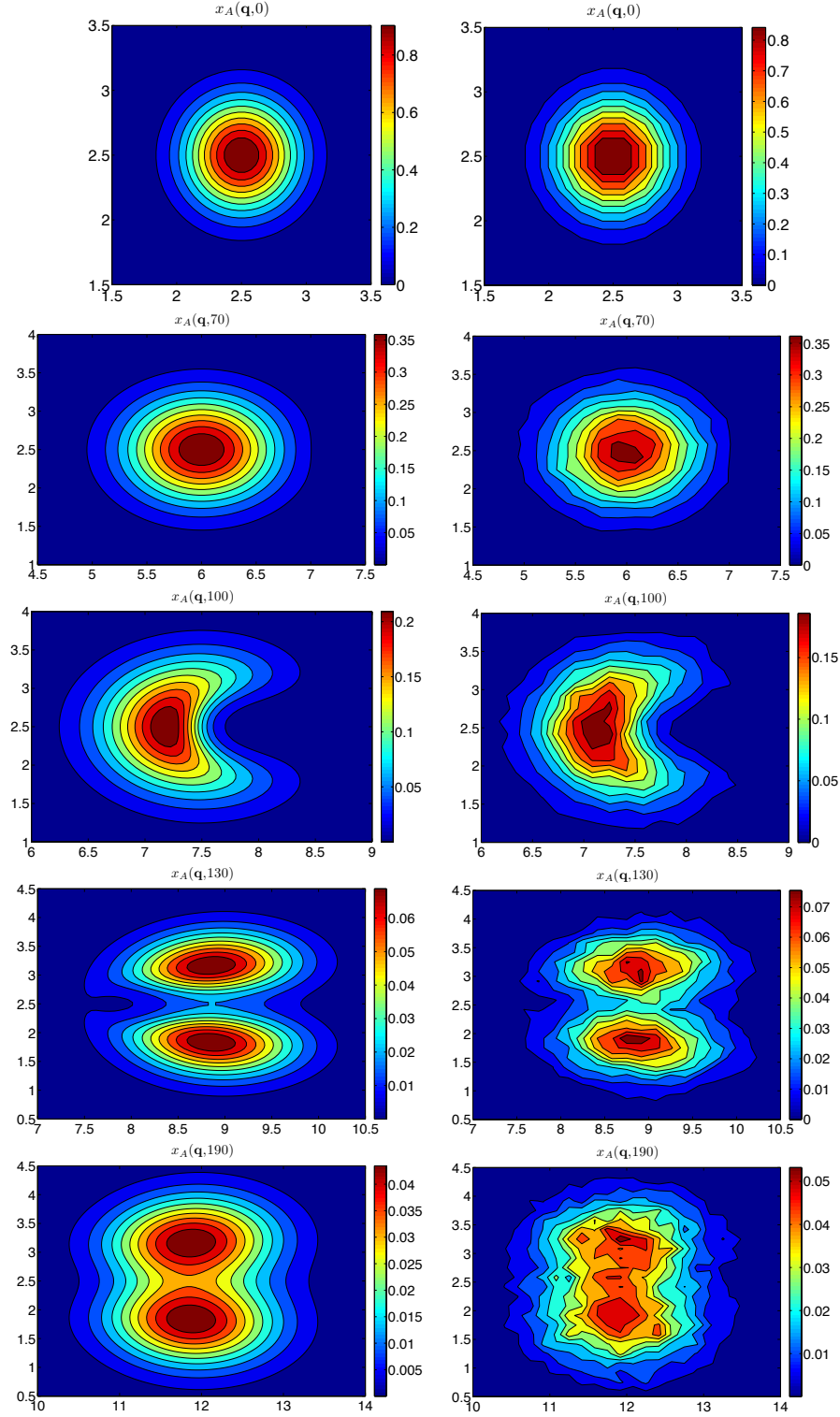
Figure 9.4: Snapshots of $x_A(\mathbf{q}, t)$ computed using the SPH method (left column) with $M_A = 10^4$ and the micro-continuous model (right column) with $N_A = 4 \times 10^4$ at $t = 0, 70, 100, 130, 190$.
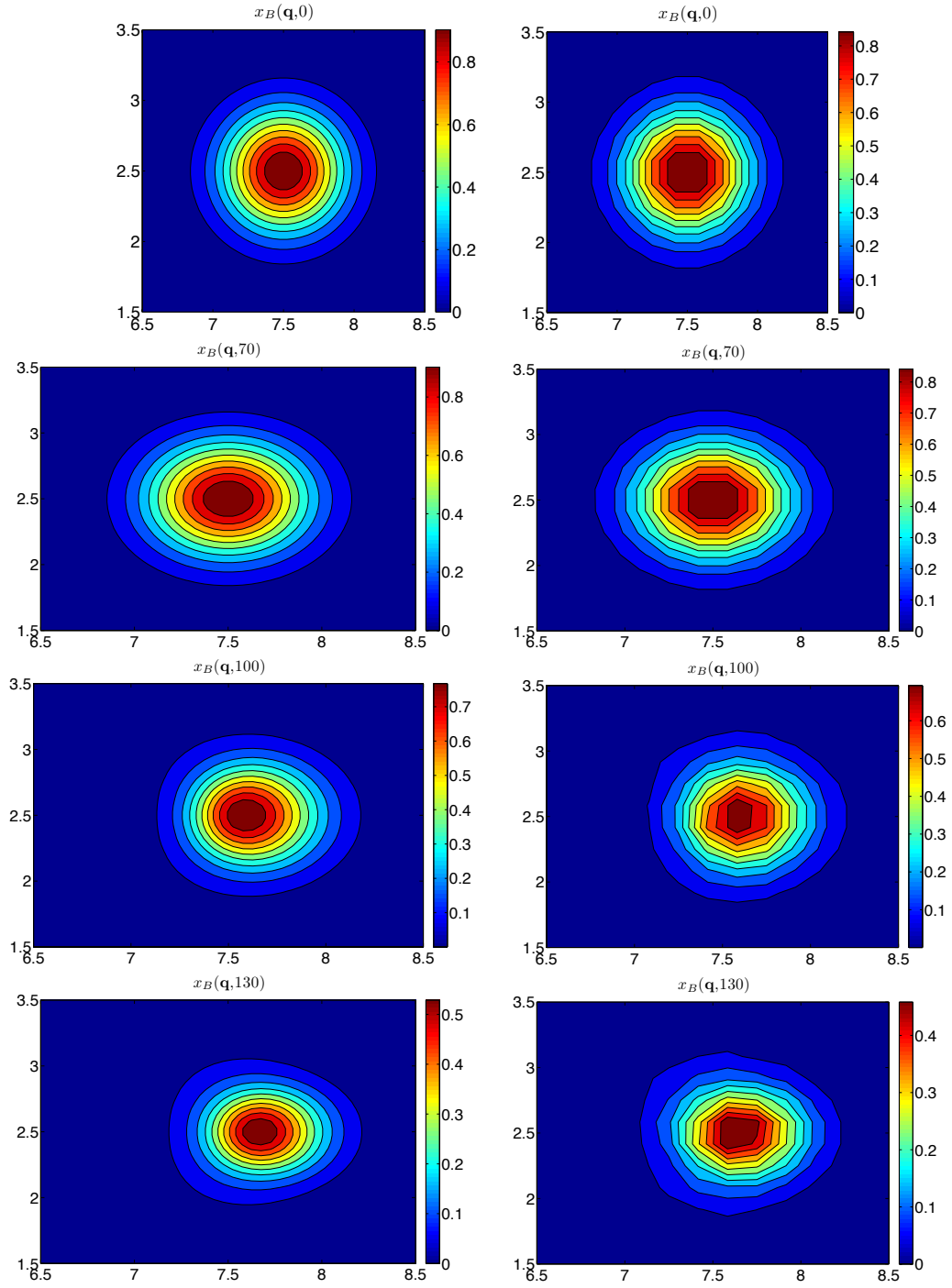
Figure 9.5: Snapshots of $x_B(\mathbf{q}, t)$ computed using the SPH method (left column) with $M_A = 10^4$ and the micro-continuous model (right column) with $N_A = 4 \times 10^4$ at $t = 0, 70, 100, 130$.
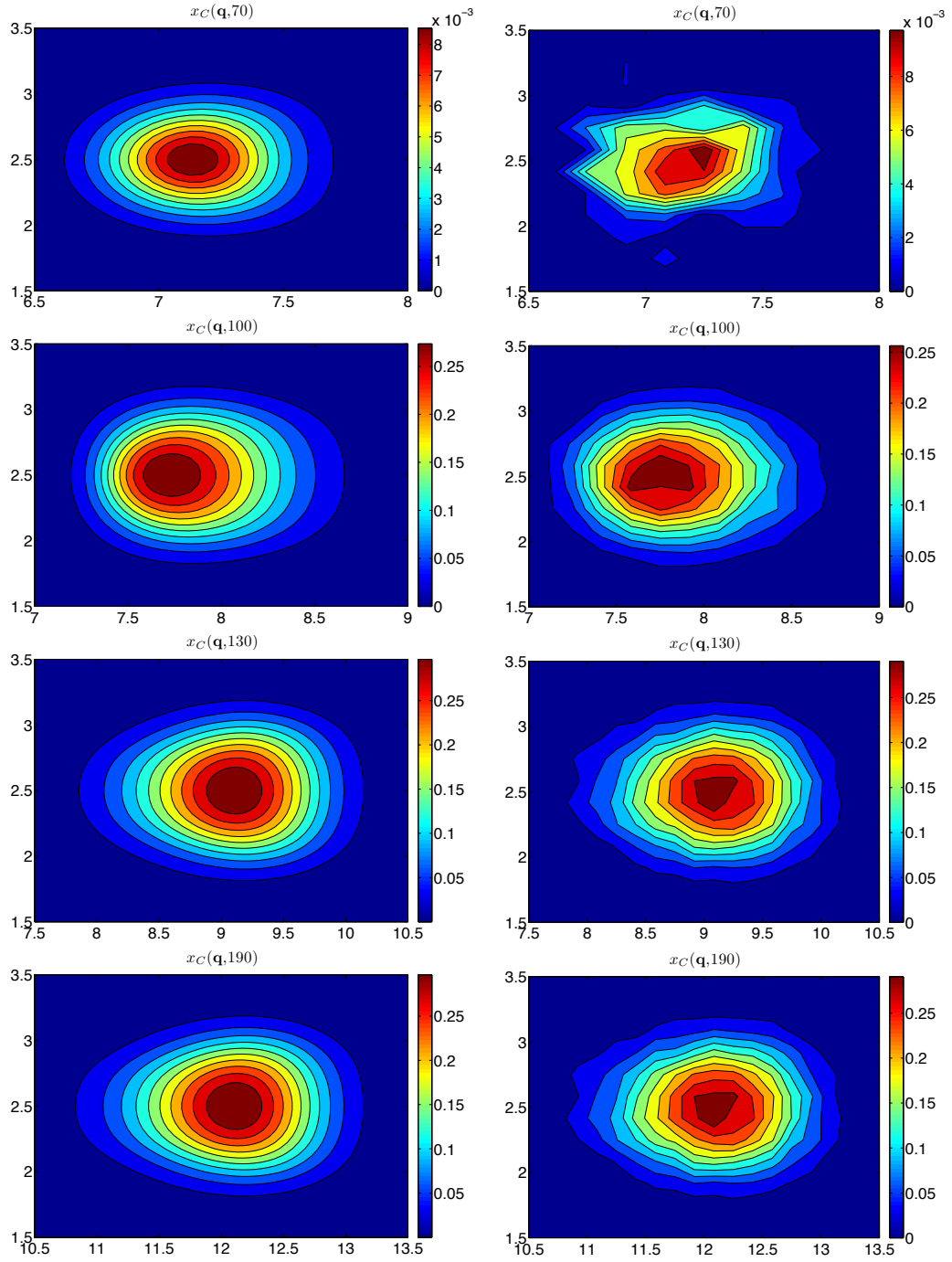
Figure 9.6: Snapshots of $x_C(\mathbf{q}, t)$ computed using the SPH method (left column) with $M_A = 10^4$ and the micro-continuous model (right column) with $N_A = 4 \times 10^4$ at $t = 70, 100, 130, 190$.

Chapter 10

Conclusion

10.1 Summary of Contributions

This thesis has presented a theoretical framework for the top-down synthesis of collective behaviors for swarm robotic systems. The framework can be used to design decentralized, scalable robot control policies for a broad range of applications with spontaneous and encounter-dependent task transitions. Here the framework was used to synthesize strategies for swarm task allocation, product assembly, and cooperative transport. Additional contributions of this thesis are a new reachability analysis algorithm for multi-affine hybrid systems and a novel experimental study of cooperative retrieval in ants.

10.2 Future Work

10.2.1 Methodology

In order to develop a wider range of swarm behaviors with theoretical guarantees, analysis and controller synthesis methodologies should be further developed for spatially inhomogeneous swarms. Approaches may include nonlinear stability analysis [74], nonlinear and robust control methods for hyperbolic (advection-reaction)

and parabolic (reaction-diffusion) PDE's [28], and Monte Carlo optimization techniques. When the macro-continuous equations are not available in closed form, a multiscale computational framework called *equation-free modeling* [78] can be employed to conduct analysis and controller synthesis by applying wrapper algorithms to appropriately initialized micro-continuous simulations on short time and length scales. In addition, methodologies may also be investigated for subgroups of swarms whose collective behavior is more accurately described by stochastic differential equations (SDE's) than by ODE's, using simulation techniques, analytical tools, and control approaches that have been developed for SDE's [45, 76].

10.2.2 Extensions to Applications

There are various possible avenues of future work in the design of the macro-continuous models for task reallocation in a swarm of robots. In the optimization of the linear models, the constraints on the flux of robots switching between tasks may be applied to the flux at all times, not only at equilibrium, in order to lower energy expenditure and prevent congestion during the transient phase of reallocation. It may also be fruitful to investigate a time-optimal control approach to the reallocation problem. The models can incorporate robots' estimation of whether a quorum exists from their own observations, and a quorum-based switching strategy can be used to halt robots from transitioning between tasks (and hence expending energy) once they detect that a task is close enough to the target population. The task transition rates, k_{ij}, may be designed to be time-dependent in order to redistribute the swarm according to a trajectory of desired distributions, $\mathbf{x^d}(t)$. Communication between robots may be incorporated, and nonlinear models can be used to represent robot interactions. For example, motifs in biomolecular networks can be used as inspiration to design nonlinear transition rates that produce interesting phenomena such as spontaneous switching as a result of an external input. In the particular case of reallocation among a set of physical sites, an aspect to consider is the dependence of inter-site

travel times on the robot population, which may lead to a generalized definition of "traffic capacity" that is not well described by a linear model.

For the swarm robotic assembly system presented in Chapters 7, two avenues of future work are to investigate the optimization of the discrete assembly plans and to use inter-robot communication to introduce a positive feedback mechanism that increases the product yield rate. Another is the synthesis of assembly systems with spatial dependencies, as an extension to the application in Chapter 9 which considered part retrieval only.

Future objectives for the cooperative transport application include a continued study of this phenomenon in ants, refinement of the ant behavioral model, development of a dynamic and behavioral simulation of transport, and synthesis of decentralized controllers for transport that will be implemented on a multi-robot testbed. Further experiments with the ants can be conducted to extract the ants' changing roles, examine the mechanics of lifting during transport, and investigate the role of sustained interaction forces in helping to avoid obstacles and overcome deadlocks. The transport behavior of *A. cockerelli* can be compared to that of another ant species that is not as effective at group retrieval in order to better understand the factors that make transport successful. The experimental trials should use an elastic structure with reduced friction; for instance, the springs may be elevated slightly so that they do not drag. The ant behavioral model can be improved by tracking the positions and orientations of individual ants in the video frames and using this data to estimate parameters that govern behavioral rules. Since the elastic structures undergo quasi-static motion during transport (Section 8.3.3), this model can be combined with a quasi-static dynamic manipulation model similar to the one in [39], replacing cables with compliant attachment points, to fully replicate the transport phenomenon in simulations. Such a dynamic model has been created and validated using the SCARAB robots in Figure 8.1 [101]. The simulated control policies can be

implemented on these robots for transporting an object to a location while maneuvering it around obstacles. The simulation and experiments can be used to study whether ant transport is optimal in some sense, such as minimizing the forces that individuals apply.

Bibliography

[1] *Scalable sWarms of Autonomous Robots and Mobile Sensors (SWARMS) Project*. http://www.swarms.org/.

[2] *Robobees Project*. http://robobees.seas.harvard.edu/, 2009.

[3] W. Agassounon and A. Martinoli. Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems. In *Proc. First Int'l. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AA-MAS)*, pages 1090–1097, 2002.

[4] William Agassounon, Alcherio Martinoli, and Kjerstin Easton. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2-3):163–191, 2004.

[5] A. Bagtzoglou, A. Tompson, and D. Dougherty. Projection functions for particle-grid methods. *Numer. Meth. Part. Diff. Eq.*, 8(4):325–340, 1992.

[6] R. Baskaran, J. H. Hoo, B. Cheng, and K. Bohringer. Catalyst enhanced micro scale batch assembly. In *Micro Electro Mechanical Systems (MEMS)*, pages 1069–1072, 2008.

[7] C. Belta, L. Habets, and V. Kumar. Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks. In *41st IEEE Conference on Decision and Control*, Las Vegas, NV, 2002.

[8] C. Belta and V. Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875, 2004.

[9] D. A. Benson and M. M. Meerschaert. Simulation of chemical reaction via particle tracking: Diffusion-limited versus thermodynamic rate-limited regimes. *Water Resour. Res.*, 44:W12201, 2008.

[10] Stephen Berard, Jeff Trinkle, Binh Nguyen, Benjamin Roghani, Vijay Kumar, and Jonathan Fink. davinci code: A multi-model simulation and analysis tool for multi-body systems. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, volume 23, pages 2588–2593, 2007.

[11] Spring Berman, Ádam Halász, and M. Ani Hsieh. Ant-inspired allocation: Top-down controller design for distributing a robot swarm among multiple tasks. In Y. Xiao and F. Hu, editors, *Bio-inspired Computing and Communication Networks*. Auerbach Publ., CRC Press, Apr. 2010. To appear.

[12] Spring Berman, Ádam Halász, M. Ani Hsieh, and Vijay Kumar. Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites. In *Proc. Conf. on Decision and Control (CDC)*, pages 4376–4381, Cancun, Mexico, 2008.

[13] Spring Berman, Ádam Halász, M. Ani Hsieh, and Vijay Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937, Aug. 2009.

[14] Spring Berman, Ádám Halász, and Vijay Kumar. MARCO: A reachability algorithm for multi-affine systems with applications to biological systems. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Int'l. Conf. on Hybrid Systems: Computation and Control (HSCC)*, pages 76–89, 2007.

[15] Spring Berman, Ádám Halász, Vijay Kumar, and Stephen Pratt. Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system. In

E. Sahin, W. M. Spears, and A. F. T. Winfield, editors, *Swarm Robotics*, volume 4433 of *LNCS*, pages 56–70. Springer, 2006.

[16] Spring Berman, Ádám Halász, Vijay Kumar, and Stephen Pratt. Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, pages 2318–2323, Rome, Italy, 2007.

[17] Spring Berman and Vijay Kumar. Abstractions and algorithms for assembly tasks with large numbers of robots and parts. In *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*, Bangalore, India, 2009.

[18] Spring Berman, Quentin Lindsey, Mahmut Selman Sakar, Vijay Kumar, and Stephen Pratt. Study of group food retrieval by ants as a model for multi-robot collective transport strategies. 2010. Accepted to *Robotics: Science and Systems*.

[19] J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. Nguyen. Programmable parts: A demonstration of the grammatical approach to self-organization. In *Proc. IEEE/RSJ Int'l. Conf. on Intell. Robots and Syst. (IROS)*, pages 3684 – 3691, 2005.

[20] Robert Bogue. Swarm intelligence and robotics. *Industrial Robot: An Int'l. Journal*, 35(6):488–495, 2008.

[21] Susan Bolden and David Skelly. *Group Behavior Database.* Yale University School of Forestry and Environmental Studies, New Haven, CT, 2006. http://alliance.seas.upenn.edu/∼swarms/dynamic/.

[22] Eric Bonabeau, Andrej Sobkowski, Guy Theraulaz, and Jean-Luis Deneubourg. Adaptive task allocation inspired by a model of division of labor in social insects. In *Biocomputing and Emergent Computation*, pages 36–45. World Scientific, 1997.

[23] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA, 1984.

[24] S. Burden, N. Napp, and E. Klavins. The statistical dynamics of programmed robotic self-assembly. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, pages 1469–1476, 2006.

[25] D. Cappelleri, G. Piazza, and V. Kumar. Two-dimensional, vision-based μN force sensor for microrobotics. In *Proc. Int'l. Conf. on Robotics and Automation*, Kobe, Japan, 2009.

[26] R. Chauvin. Le transport de proies chez les fourmis. Y-a-t-il entr'aide? *Behaviour*, 2(4):249–256, 1950.

[27] Madalena Chaves, Eduardo D Sontag, and Robert J Dinerstein. Steady-states of receptor-ligand dynamics: a theoretical framework. *J. Theoretical Biology*, 227(3):413–28, 2004.

[28] Panagiotis D. Christofides. *Nonlinear and Robust Control of PDE Systems.* Birkhauser, Boston, 2001.

[29] T. J. Chung. *Computational Fluid Dynamics.* Cambridge Univ. Press, Cambridge, 2002.

[30] David C Conner, Alfred Rizzi, and Howie Choset. Composition of local potential functions for global robot control and navigation. In *Proc. Int'l Conf. on Intelligent Robots and Systems (IROS)*, pages 3546–3551, 2003.

[31] Nikolaus Correll and Alcherio Martinoli. Modeling self-organized aggregation in a swarm of miniature robots. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA), Workshop on Collective Behaviors inspired by Biological and Biochemical Systems*, 2007.

[32] T. Dang and O. Maler. Reachability analysis via face lifting. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 96–109. Springer Verlag, Berlin, 1998.

[33] William M. Deen. *Analysis of Transport Phenomena*. Oxford University Press, New York, 19983.

[34] F. Delay, P. Ackerer, and C. Danquigny. Simulating solute transport in porous or fractured formations using random walk particle tracking: A review. *Vadose Zone Journal*, 4:360–379, 2005.

[35] M Bernardine Dias, Robert Michael Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: a survey and analysis. *Proc. of the IEEE*, 94(7):1257 – 1270, 2006.

[36] Johan Elf and Mans Ehrenberg. Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Res.*, 13:2475–2484, 2003.

[37] M Feinberg. Chemical reaction network structure and the stability of complex isothermal reactors. I: The deficiency zero and deficiency one theorems. *Chemical Engineering Science*, 42(10):2229–2268, 1987.

[38] M Feinberg. The existence and uniqueness of steady states for a class of chemical reaction networks. *Archive for Rational Mechanics and Analysis*, 132:311–370, 1995.

[39] Jonathan Fink, Peng Cheng, and Vijay Kumar. Cooperative towing with multiple robots. In *Proc. ASME IDETC/CIE*, pages 1471–1476, New York, NY, 2008.

[40] Jonathan Fink, M. Ani Hsieh, and Vijay Kumar. Multi-robot manipulation via caging in environments with obstacles. In *Proc. Int'l. Conf. on Robotics and Automation*, pages 1471–1476, Pasadena, CA, 2008.

[41] D. B. Forger and C. S. Peskin. A detailed predictive model of the mammalian circadian clock. *PNAS*, 100(25):14806–14811, 2003.

[42] N. Franks, S. Pratt, E. Mallon, N. Britton, and D. Sumpter. Information flow, opinion polling and collective intelligence in house-hunting social insects. *Phil. Trans. Roy. Soc. London*, B(357):1567–1584, 2002.

[43] N. R. Franks, A. B. Sendova-Franks, J. Simmons, and M. Mogie. Convergent evolution, superefficient teams and tempo in old and new world army ants. *Proc. Royal Soc. B*, 266:1697–1701, 1999.

[44] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. In Manfred Morari and Lothar Thiele, editors, *Proc. 5th Int'l Workshop on Hybrid Systems: Computation and Control (HSCC)*, volume 3414 of *LNCS*, pages 258–273, 2005.

[45] Crispin Gardiner. *Stochastic Methods: A Handbook for the Natural and Social Sciences.* Springer, 4th edition, 2009.

[46] Brian Gerkey and Maja Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int'l. J. of Robotics Research*, 23(9):939–954, 2004.

[47] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81:2340–2361, 1977.

[48] D Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188(1-3):404–425, 1992.

[49] D. T. Gillespie, M. Roh, and L. R. Petzold. Refining the weighted stochastic simulation algorithm. *J. Chem. Phys.*, 130:174103, 2009.

[50] Dan Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.

[51] Dan Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.

[52] A. Girard and G. Pappas. Approximate bisimulations for nonlinear dynamical systems. In *Proc. 44th IEEE Conf. on Decision and Control*, Seville, Spain, 2005.

[53] C. Godsil and C. Royle. *Algebraic Graph Theory*. Springer, New York, 2001.

[54] R. Groß and M. Dorigo. Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9):1490–1508, 2008.

[55] L.C.G.J.M. Habets and J. H. van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*, 40:21–35, 2004.

[56] Á. M. Halász, V. Kumar, M. Imielinski, C. Belta, O. Sokolsky, S. Pathak, and H. Rubin. Analysis of lactose metabolism in E.coli using reachability analysis of hybrid systems. *IET Systems Biology*, 1(2):130–148, 2007.

[57] Ádam Halász, M. Ani Hsieh, Spring Berman, and Vijay Kumar. Dynamic redistribution of a swarm of robots among multiple sites. In *Proc. Int'l. Conf. on Intelligent Robots and Systems (IROS)*, pages 2320–2325, San Diego, CA, 2007.

[58] Heiko Hamann and Heinz Wörn. A framework of space-time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2-4):209–239, 2008.

[59] B. Harris. *Theory of Probability*. Addison-Wesley, Reading, MA, 1966.

[60] R Heinrich, S Schuster, and H-G Holzhutter. Mathematical analysis of enzymic reaction systems using optimization principles. *Eur. J. Biochem.*, 201:1–21, 1991.

[61] Reinhart Heinrich and Stefan Schuster. *The Regulation of Cellular Systems*. Chapman & Hall, New York, NY, 1996.

[62] T. Henzinger, P. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1):110–122, 1998.

[63] P. A. Herrera, M. Massabo, and R. D. Beckie. A meshless method to simulate solute transport in heterogeneous porous media. *Advances in Water Resources*, 32:413–429, 2009.

[64] B. Holldobler, R. C. Stanton, and H. Markl. Recruitment and food-retrieving behavior in Novomessor (Formicidae, Hymenoptera: I. Chemical signals). *Behav. Ecol. Sociobiol.*, 4:163–181, 1978.

[65] Bert Holldobler and Edward O. Wilson. *The Ants*. Harvard University Press, Cambridge, MA, 1990.

[66] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge Univ. Press, 1985.

[67] Kazuo Hosokawa, Isao Shimoyama, and Hirofumi Miura. Dynamics of self-assembling systems: Analogy with chemical kinetics. *Artificial Life*, 1(4):413–427, 1994.

[68] L. Howell. *Compliant Mechanisms*. John Wiley & Sons, Inc., New York, 2001.

[69] M. Ani Hsieh, Ádam Halász, Spring Berman, and Vijay Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2):121–141, 2008.

[70] W. Hundsdorfer and J. G. Verwer. *Numerical Solutions of Time-Dependent Advection-Diffusion-Reaction Equations.* Springer-Verlag, New York, 2003.

[71] Neema Jamshidi and Bernhard O Palsson. Formulating genome-scale kinetic models in the post-genome era. *Molecular Systems Biology*, 4(171):1–10, 2008.

[72] M. Jubelgas, V. Springel, and K. Dolag. Thermal conduction in cosmological sph simulations. *Mon. Not. Roy. Astron. Soc.*, 351:423–435, 2004.

[73] A. Julius, Á. Halász, V. Kumar, and G. Pappas. Finite state abstraction of a stochastic model of the lactose regulation system of *escherichia coli.* In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, Dec. 2006.

[74] E. W. Justh and P. S. Krishnaprasad. Pattern-forming systems for control of large arrays of actuators. *J. Nonlinear Sci.*, 11:239–277, 2001.

[75] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier, New York, 3rd edition, 2007.

[76] D. Kannan and V. Lakshmikantham, editors. *Handbook of Stochastic Analysis and Applications.* CRC, 2001.

[77] Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3):946–956, 2005.

[78] I. G. Kevrekidis and G. Samaey. Equation-free multiscale computation: Algorithms and applications. *Annu. Rev. Phys. Chem.*, 60:321–344, 2009.

[79] Eric Klavins. Programmable self-assembly. *Control Systems Magazine*, 24(4):43–56, 2007.

[80] Eric Klavins, Samuel Burden, and Nils Napp. Optimal rules for programmed stochastic self-assembly. *Robotics: Science and Systems II*, pages 9–16, 2007.

[81] M. Kloetzer and C. Belta. Reachability analysis of multi-affine systems. In J. P. Hespanha and A. Tiwari, editors, *Proc. 9th Int'l Workshop on Hybrid Systems: Computation and Control (HSCC)*, volume 3927 of *LNCS*, pages 348–362, 2006.

[82] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an opensource multi-robot simulator. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, 2004.

[83] Seung kook Yun, Mac Schwager, and Daniela Rus. Coordinating construction of truss structures using distributed equal-mass partitioning. In *Proc. Int'l. Symposium on Robotics Research (ISRR)*, 2009.

[84] M. J. B. Krieger, J.-B. Billeter, and L. Keller. Ant–like task allocation and recruitment in cooperative robots. *Nature*, 406:992–995, 2000.

[85] C. Ronald Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30:85–101, 2000.

[86] Vijay Kumar and Kenneth Waldron. Force distribution in closed kinematic chains. *IEEE J. of Robotics and Automation*, 4(6):657–664, 1988.

[87] Hiroyuki Kuwahara and Ivan Mura. An efficient and exact stochastic simulation method to analyze rare events in biochemical systems. *J. Chem. Phys.*, 129:165101, 2008.

[88] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. http://control.ee.ethz.ch/∼mpt/.

[89] Thomas H. Labella, Marco Dorigo, and Jean-Louis Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Trans. Auton. Adapt. Syst.*, 1(1):4–25, 2006.

[90] G. Lafferriere, G. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *Hybrid Systems: Computation and Control (HSCC)*, volume 1569 of *LNCS*, pages 137–151, 1999.

[91] Sotiria Lampoudi, Dan Gillespie, and Linda Petzold. The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. *J. Chemical Physics*, 130(9):094104–1–094104–16, 2009.

[92] Paola Lecca and Lorenzo Dematté. Stochastic simulation of reaction-diffusion systems. *Int'l. J. of Biological and Medical Sciences*, 4(4):211–231, 2009.

[93] Patrick De Leenheer, David Angeli, and Eduardo Sontag. Monotone chemical reaction networks. *J. Mathematical Chemistry*, 41(3):295–314, 2007.

[94] Caroline Lemerle, Barbara Di Ventura, and Luis Serrano. Space as the final frontier in stochastic simulations of biological systems. *Federation of European Biochem. Soc. Letters*, 579:17891794, 2005.

[95] Naomi Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 2968–2973, Orlando, FL, 2001.

[96] K. Lerman, A. Martinoli, and A. Galstyan. A review of probabilistic macroscopic models for swarm robotic systems. In E. Sahin and W. Spears, editors, *Swarm Robotics Workshop: State-of-the-art Survey*, volume 3342 of *LNCS*, pages 143–152, 2005.

[97] Kristina Lerman, Chris V. Jones, Aram Galstyan, and Maja J. Mataric. Analysis of dynamic task allocation in multi-robot systems. *Int'l. J. of Robotics Research*, 25(4):225–242, 2006.

[98] A. Lewis and M. Overton. Eigenvalue optimization. *Acta Numerica*, 5:149–190, 1996.

[99] H Li, Y Cao, and L Petzold. StochKit: A stochastic simulation toolkit. *http://cs.ucsb.edu*. Dept. of Computer Science, Univ. of California, Santa Barbara.

[100] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999.

[101] Quentin J. Lindsey, Michael Shomin, and Vijay Kumar. Cooperative quasi-static planar manipulation with multiple robots. 2010. Submitted to ASME IDETC/CIE.

[102] Arnaud Lioni. *Chapter VII: Le transport collectif chez Oecophylla longinoda. From: Auto-assemblage et transport collectif chez Oecophylla. Unpublished.* PhD thesis, Universite Libre de Bruxelles, Belgium, 2000.

[103] G. R. Liu. *Meshfree Methods: Moving Beyond the Finite Element Method.* CRC Press, Boca Raton, FL, 2nd edition, 2010.

[104] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method.* World Scientific Publishing Company, 2003.

[105] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics: An overview and recent developments. *Arch. Comp. Methods Eng.*, 17:25–76, 2010.

[106] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems. In *Proc. IEEE/RSJ 2008 Int'l. Conf. on Intelligent Robots and Systems (IROS 2008)*, pages 4004–4010. IEEE, 2008.

[107] H. Markl and B. Holldobler. Recruitment and food-retrieving behavior in Novomessor (Formicidae, Hymenoptera: II. Vibration signals). *Behav. Ecol. Sociobiol.*, 4:183–216, 1978.

[108] Loïc Matthey. Hybrid reactions modeling for top-down design framework. *Ecole Polytechnique Federale de Lausanne*, 2008. Master's thesis.

[109] Loïc Matthey, Spring Berman, and Vijay Kumar. Stochastic strategies for a swarm robotic assembly system. In *Proc. of the Int'l. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[110] J.-A. Meyer. Transport de proies déposées sur une piste de Formica polyctena. II. étude de leur dispersion. Expériences de détour. *Insectes Sociaux*, 16:27–38, 1969.

[111] N. Michael, J. Fink, and V. Kumar. Experimental testbed for large multirobot teams: verification and validation. *IEEE Robotics and Automation Magazine*, 15(1):53–61, 2008.

[112] Nathan Michael, Jonathan Fink, Savvas Loizou, and Vijay Kumar. Architecture, abstractions, and algorithms for controlling large teams of robots: Experimental testbed and results. *Proc. Int'l. Symposium of Robotics Research (ISRR)*, 2007.

[113] O Michel. Webots: Professional mobile robot simulation. *Int'l. J. Advanced Robotic Systems*, 1(1):39–42, 2004.

[114] Mark W. Moffett. Cooperative food transport by an asiatic ant. *National Geographic Research*, 4(3):386–394, 1988.

[115] J. J. Monaghan. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.*, 30:543–574, 1992.

[116] G. Montemayor and J. Wen. Decentralized collaborative load transport by multiple robots. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, pages 372–377, 2005.

[117] N. Napp, S. Burden, and E. Klavins. Setpoint regulation for stochastically interacting robots. In *Robotics: Science and Systems*, 2009.

[118] H. Othmer. *Analysis of Complex Reaction Networks, Lecture Notes*. University of Minnesota, 2003.

[119] M. R. Pac, A. M. Erkmen, and I. Erkmen. Control of robotic swarm behaviors based on smoothed particle hydrodynamics. In *Proc. IEEE/RSJ Int'l. Conf. on Intell. Robots and Syst. (IROS)*, pages 4194–4200, 2007.

[120] Saurav Pathak. *Sylvester's Method of Resultants*. University of Pennsylvania, December 2004.

[121] J. R. Perkinson and B. Shafai. A decentralized control algorithm for scalable robotic swarms based on mesh-free particle hydrodynamics. In *Proc. IASTED International Conference on Robotics and Applications*, pages 1–6, 2005.

[122] L. Pimenta, M. Mendes, R. Mesquita, and G. Pereira. Fluids in electrostatic fields: An analogy for multirobot control. *IEEE Trans. on Magnetics*, 43(4):1765–1768, 2007.

[123] L. Pimenta, N. Michael, R. Mesquita, G. Pereira, and V. Kumar. Control of swarms based on hydrodynamic models. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, pages 1948–1953, 2008.

[124] S. Pratt, D. Sumpter, E. Mallon, and N. Franks. An agent-based model of collective nest choice by the ant *temnothorax albipennis*. *Animal Behav.*, 70:1023–1036, 2005.

[125] S.C. Pratt, E. B. Mallon, D. J. T. Sumpter, and N. R. Franks. Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant Leptothorax albipennis. *Behavioral Ecology and Sociobiology*, 52:117–127, 2002.

191

[126] Stephen Pratt. Quorum sensing by encounter rates in the ant Temnothorax albipennis. *Behavioral Ecology*, 16(2):488–496, 2005.

[127] Craig Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, volume 21, pages 25–34, 1987.

[128] E. Rimon and D. Koditschek. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11:412–442, 1990.

[129] E. Rimon and D. Koditschek. The construction of analytic diffeomorphisms for exact robot navigation on star worlds. *Transactions of the American Mathematical Society*, 327(1):71–115, 1991.

[130] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[131] R. Russell and T. Urban. Vehicle routing with soft time windows and Erlang travel times. *J. Operational Research Society*, 59(9):1220–1228, 2008.

[132] P. Salamon, D. Fernandez-Garcia, and J. J. Gomez-Hernandez. A review and numerical assessment of the random walk particle tracking method. *J. of Contaminant Hydrology*, 87:277–305, 2006.

[133] Hana El Samad, Mustafa Khammash, Linda Petzold, and Dan Gillespie. Stochastic modelling of gene regulatory networks. *Int. J. Robust Nonlinear Control*, 15:691–711, 2005.

[134] S. Sastry. *Nonlinear systems: analysis, stability, and control*. Springer-Verlag, New York, 1999.

[135] S Schuster and R Heinrich. Time hierarchy in enzymatic reaction chains resulting from optimality principles. *J. Theoretical Biology*, 129(2):189–209, 1987.

[136] Frank Schweitzer. *Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences.* Springer, New York, 2003.

[137] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101:165–200, 1998.

[138] Onn Shehory, Sarit Kraus, and Osher Yadgar. Emergent cooperative goal-satisfaction in large-scale automated-agent systems. *Artificial Intellig.*, 110(1):1–55, 1999.

[139] D. Siegel and D. MacLean. Global stability of complex balanced mechanisms. *J. Mathematical Chemistry*, 27:89–110, 2000.

[140] B.I. Silva, K. Richeson, B.H. Krogh, and A. Chutinan. Modeling and verifying hybrid dynamic systems using CheckMate. In *Proc. 4th Conference on Automation of Mixed Processes*, pages 323–328, 2000.

[141] J. H. Sudd. The transport of prey by an ant, Pheidole crassinoda. *Behaviour*, 16:295–308, 1960.

[142] J. H. Sudd. How insects work in groups. *Discovery*, 24(6):15–19, 1963.

[143] J. H. Sudd. The transport of prey by ants. *Behaviour*, 25:234–271, 1965.

[144] J. Sun, S. Boyd, L. Xiao, and P. Diaconis. The fastest mixing Markov process on a graph and a connection to the maximum variance unfolding problem. *SIAM Review*, 48(4):681–699, 2006.

[145] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Flocking in fixed and switching networks. In *IEEE Transactions on Automatic Control*, volume 52, pages 863–868, 2007.

[146] A. M. Tartakovsky, T. D. Scheibe, G. Redden, P. Meakin, and Y. Fang. Smoothed particle hydrodynamics model for reactive transport and mineral

precipitation. In *Int'l Conference on Computational Methods in Water Resources (CMWR XVI)*, Copenhagen, Denmark, 2006.

[147] Etienne Toffin. *Étude expérimentale du transport collectif chez la fourmi Pheidole pallidula.* PhD thesis, Université Libre de Bruxelles, 2003.

[148] Claire Tomlin. *Lecture Notes 6: Stability of Hybrid Systems; Course AA278A: Hybrid Systems: Modeling, Analysis, and Control.* Stanford University, 2005.

[149] A. F. B. Tompson. Particle-grid methods for reacting flows in porous media with application to fisher's equation. *Appl. Math. Modelling*, 16:374–383, 1992.

[150] A. J. van der Schaft and J. M. Schumacher. *An introduction to hybrid dynamical systems.* Springer-Verlag, Berlin, 2000.

[151] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[152] Lovekesh Vig and Julie Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.

[153] Lovekesh Vig and Julie Adams. Coalition formation: From software agents to robots. *J. Intellig. and Robotic Syst.*, 50(1):85–118, 2007.

[154] Justin Werfel and Radhika Nagpal. Three-dimensional construction with mobile robots and modular blocks. *Int'l. J. Robotics Research*, 27(3-4):463–479, 2008.

[155] D James Wilkinson. *Stochastic Modelling for Systems Biology.* Chapman & Hall/CRC, 2006.

[156] Chai Wah Wu. On Rayleigh-Ritz ratios of a generalized Laplacian matrix of directed graphs. *Linear Algebra and its Applications*, 402:207–227, 2005.

[157] N. Yildirim and M. C. Mackey. Feedback regulation in the lactose operon: A mathematical modeling study and comparison with experimental data. *Biophys. Journal*, 84:2841–2851, 2003.

[158] Y. Zhao, C. C. Lim, D. B. Sawyer, L. Ronglih, and X. Zhang. Cellular force measurements using single-spaced polymeric microstructures: isolating cells from base substrate. *J. Micromech. Microeng.*, 15:1649–1656, 2005.