Stochastic Strategies for a Swarm Robotic Assembly System

Loïc Matthey, Spring Berman and Vijay Kumar

Abstract—We present a decentralized, scalable approach to assembling a group of heterogeneous parts into different products using a swarm of robots. While the assembly plans are predetermined, the exact sequence of assembly of parts and the allocation of subassembly tasks to robots are determined by the interactions between robots in a decentralized fashion in real time. Our approach is based on developing a continuous abstraction of the system derived from models of chemical reactions and formulating the strategy as a problem of selecting rates of assembly and disassembly. These rates are mapped onto probabilities that determine stochastic control policies for individual robots, which then produce the desired aggregate behavior. This top-down approach to determining robot controllers also allows us to optimize the rates at the abstract level to achieve fast convergence to the specified target numbers of products. Because the method incorporates programs for assembly and disassembly, changes in demand can lead to reconfiguration in a seamless fashion. We illustrate the methodology using a physics-based simulator with examples involving 15 robots and two types of final products.

I. INTRODUCTION

We develop an approach to designing a reconfigurable manufacturing system in which a swarm of homogeneous robots assembles static, heterogeneous parts into different types of products. The system must respond quickly to produce desired amounts of products from any initial set of parts. We employ a decentralized strategy in which robots operate autonomously and both robots and parts use local communication. The strategy can be readily implemented on resource-constrained robots, and it is scalable in the number of robots and parts and robust to changes in robot population.

We specify that robots move randomly inside a closed arena and pick up randomly scattered parts. The system can be modeled as being spatially homogeneous, and the robotpart and robot-robot interactions are analogous to chemical reactions between molecules. This allows us to model the system using the extensively studied Chemical Reaction Network (CRN) framework [1].

In the taxonomy [2] of macroscopic self-assembly systems, our objective and approach are most similar to those of [3], which considers a set of modules that bind through random collisions and detach into different parts according to programmed probabilities. The system in [3] is also modeled as a CRN, as is the one in [4], which predicts the yield of complete assemblies from passive, vertically stirred modules. In [3], an optimization problem is formulated to compute the detachment probabilities that maximize the yield of a particular assembly at equilibrium. The optimization is based on a Markov process model of the system and requires the enumeration of all reachable states, which does not scale well with the number of parts. Our use of robots to transport and join passive parts according to decentralized rules is similar to the setup in [5], which derives rules for building a single desired structure out of blocks.

In this paper, we use a "top-down" design methodology similar to [6], [7] to synthesize stochastic control policies for robots that cause them to produce target amounts of products as quickly as possible. Fig. 1 illustrates our methodology. As in recent work on modeling robot swarms [8]-[10], we develop an accurate macroscopic model of the physical system. We generate the continuous dynamics of individual robots using a realistic 3D physics simulation, the *micro-continuous model*. Since the system is spatially homogeneous, it can be represented by the Chemical Master Equation [11]. We call this the complete macro-discrete model because it describes a continuous-time Markov process whose states are the discrete populations of parts and robots. When these populations are large, the system can be abstracted to an ordinary differential equation (ODE) model, the complete macro-continuous model, whose state variables are continuous amounts of parts and robots. We compute rate constants in this model from geometrical properties of the robots and environment and check that the model predicts the behavior of the micro-continuous model. We simplify this abstraction to a more easily analyzable reduced macrocontinuous model with the same rates.

Our novel contribution to prior work on swarm modeling is our control synthesis methodology, which provides theoretical guarantees on system performance. Using an approach similar to [7], we optimize the rates in the reduced macrocontinuous model to minimize system convergence time while enforcing target quantities of all parts at equilibrium. The optimization problem is independent of the number of parts, scaling only with the number of rates. We then map the rates onto probabilities of assembly and disassembly which are used as robot control policies in the micro-continuous model. The trends in the evolution of part populations in this simulation are predicted by the designed behavior of the continuous model, including the faster convergence that results from using optimized versus non-optimal rates.

L. Matthey is with the DISAL Laboratory, Ecole Polytechnique Federale de Lausanne, Station 14, 1015 Lausanne, Switzerland, loic.matthey@epfl.ch

S. Berman and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104, USA, {spring, kumar}@grasp.upenn.edu

We gratefully acknowledge partial support from NSF grants CSR-CPS 0720801, IIS-0427313, NSF IIP-0742304, and IIS-0413138, ARO grant W911NF-05-1-0219, and ONR grant N00014-07-1-0829.



Fig. 1. Levels of abstraction of the assembly 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 using the theoretical justification in [12]. Under certain assumptions (see Section III-C), the complete macro-continuous model can be mapped to a lower-dimensional model via the abstraction \mathcal{F}_r .



Fig. 2. Assembly plans for final assemblies F1 and F2.

II. PROBLEM STATEMENT

A. 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 Fig. 2, culminating in final assemblies F1 and F2. 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. The assembly plans and control policies can be preprogrammed onto the robots and updated via a broadcast if product demand changes. 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.



Fig. 3. Snapshot of the arena in the realistic physical simulation. Robots carry parts at the end of a protruding bar.

B. Micro-continuous model

We implement the assembly task in the robot simulator Webots [13], which uses the Open Dynamics Engine to accurately simulate physics. 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 takes place inside a walled hexagonal arena. Fig. 3 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. 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 dropping one of the component parts on the ground. To control the outcome of part populations, we can directly modify the probabilities of robots starting an assembly and performing a disassembly.

III. MACRO-CONTINUOUS MODELS

A. Definitions

Interactions between parts and robots are modeled as a CRN. A set of reactions can be represented as a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of vertices, \mathcal{V} , signifies the *complexes*, the combinations of parts and/or robots that appear before and after reaction arrows. The set of directed edges, \mathcal{E} , represents the reaction pathways between the complexes. Each pathway is denoted by an ordered pair $(i, j) \in \mathcal{V} \times \mathcal{V}$, which means that complex *i* transforms into complex *j*, and is associated with a positive *reaction rate constant*.

Each part of type *i* in Fig. 2 is symbolized by X_i , and a robot is symbolized by X_R . X_i may be further 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. Let *M* be the number of these variables, or *species*, in a model of the system. Then $\mathbf{x}(t) \in \mathbb{R}^M$ is the vector of the species populations, which are represented as continuous functions of time *t*.

B. Complete macro-continuous model

We define a CRN that represents each possible action in the micro-continuous model of the assembly system:

$$\begin{aligned} X_{R} + X_{i}^{u} & \stackrel{e_{i}}{\to} & X_{i}^{c} , \qquad i = 1, ..., 8 \\ X_{1}^{c} + X_{2}^{c} & \stackrel{k_{1}^{+}}{\to} & X_{5}^{c} + X_{R} & X_{2}^{c} + X_{7}^{c} & \stackrel{k_{4}^{+}}{\to} & X_{F1}^{c} + X_{R} \\ X_{3}^{c} + X_{4}^{c} & \stackrel{k_{2}^{+}}{\to} & X_{6}^{c} + X_{R} & X_{2}^{c} + X_{5}^{c} & \stackrel{k_{5}^{+}}{\to} & X_{8}^{c} + X_{R} \\ X_{5}^{c} + X_{6}^{c} & \stackrel{k_{3}^{+}}{\to} & X_{7}^{c} + X_{R} & X_{6}^{c} + X_{8}^{c} & \stackrel{k_{6}^{+}}{\to} & X_{F2}^{c} + X_{R} \\ X_{5}^{c} & \stackrel{k_{1}^{-}}{\to} & X_{1}^{c} + X_{2}^{u} & X_{F1}^{c} & \stackrel{k_{4}^{-}}{\to} & X_{7}^{c} + X_{2}^{u} \\ X_{6}^{c} & \stackrel{k_{2}^{-}}{\to} & X_{3}^{c} + X_{4}^{u} & X_{8}^{c} & \stackrel{k_{5}^{-}}{\to} & X_{5}^{c} + X_{2}^{u} \\ X_{7}^{c} & \stackrel{k_{3}^{-}}{\to} & X_{6}^{c} + X_{5}^{u} & X_{F2}^{c} & \stackrel{k_{6}^{-}}{\to} & X_{8}^{c} + X_{6}^{u} & (1) \end{aligned}$$

In this CRN, e_i is the rate at which a robot encounters a part of type i, k_i^+ is the rate of assembly process i, and k_i^- is the rate of disassembly process i. We theoretically estimate these rates as functions of the following probabilities:

$$e_i = p^e$$
, $k_i^+ = p^e \cdot p_i^a \cdot p_i^+$, $k_i^- = p_i^-$. (2)

 p^e is the probability 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 from the geometrical approach that is used to compute probabilities of molecular collisions [9], [11]: $p^e \approx vTw/A$, where v is the average robot speed, T is a timestep, A is the area of the arena, 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_i^a is the probability of two robots successfully completing assembly process *i*; it depends on the part geometries.

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

We compute p_i^a and the parameters for p^e using measurements from the micro-continuous model (Webots simulations): $A = 23.4 m^2$ (hexagon of radius 3 m), w = 1.2 m, v = 0.128 m/s from an average over 50 runs, and $\mathbf{p^a} = [0.9777 \ 0.9074 \ 0.9636 \ 0.9737 \ 0.8330 \ 1.0]$ (entries follow the numbering of the associated reactions) from averages over 100 runs. We set T = 1 s.

In the thermodynamic limit, which includes the condition that populations approach infinity, the physical system represented by (1) can be abstracted to an ODE model [12]. This is illustrated in the next section. We numerically integrate



Fig. 4. Evolution of part populations for 3 final assemblies and 15 robots. Error bars show standard deviations. (a) All part populations in the microcontinuous model, averaged over 100 runs. (b) Final assembly populations in the complete macro-continuous, macro-discrete, and micro-continuous models. The latter two models are each averaged over 100 runs.

this macro-continuous model with the rates we calculated and also use the StochKit toolbox [14] to efficiently perform a stochastic simulation of the macro-discrete model. We compare the results to those for the micro-continuous model in Fig. 4, using $p_i^+ = 1$, $p_i^- = 0$ $\forall i$. Discrepancies among the models arise from several factors. The system modeled has relatively low numbers of robots and parts (15 each), while the ODE approximation is most accurate for large populations. In the simulation, 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, which are most detrimental to systems with few final products, or the small local effect of a higher availability of parts where they are dropped, which often leads to the recreation of broken assemblies. The discrepancies caused by these factors can be reduced by increasing the robot and part populations; however, it becomes more computationally expensive to simulate the system as the populations increase. Nevertheless, the macrocontinuous model appears to predict the evolution of part populations fairly accurately, and hence we can use it to design the rates to direct the system's behavior.

C. Reduced macro-continuous model

We simplify the complete model 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:

$$X_{1} + X_{2} \stackrel{k_{1}^{+}}{\underset{k_{1}^{-}}{\rightleftharpoons}} X_{5} \quad X_{3} + X_{4} \stackrel{k_{2}^{-}}{\underset{k_{2}^{-}}{\rightleftharpoons}} X_{6} \quad X_{5} + X_{6} \stackrel{k_{3}^{-}}{\underset{k_{3}^{-}}{\rightleftharpoons}} X_{7}$$
$$X_{2} + X_{7} \stackrel{k_{4}^{+}}{\underset{k_{4}^{-}}{\rightleftharpoons}} X_{F1} \quad X_{2} + X_{5} \stackrel{k_{5}^{+}}{\underset{k_{5}^{-}}{\rightleftharpoons}} X_{8} \quad X_{6} + X_{8} \stackrel{k_{6}^{+}}{\underset{k_{6}^{-}}{\rightleftharpoons}} X_{F2} \quad (3)$$

The rates are also defined by equation (2).

We define a vector $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^{12}$ in which entry y_i is the part or product of parts in complex *i*:

$$\mathbf{y}(\mathbf{x}) = \begin{bmatrix} x_1 x_2 & x_5 & x_3 x_4 & x_6 & x_2 x_7 & x_{F1} \\ x_5 x_6 & x_7 & x_2 x_5 & x_8 & x_6 x_8 & x_{F2} \end{bmatrix}^T .$$
(4)

We also define a matrix $\mathbf{M} \in \mathbb{R}^{10 \times 12}$ in which each entry \mathbf{M}_{ji} , j = 1, ..., 10, of column \mathbf{m}_i is the coefficient of part type j in complex i (0 if absent). We relabel the rate associated with reaction $(i, j) \in \mathcal{E}$ as k_{ij} and define a matrix $\mathbf{K} \in \mathbb{R}^{12 \times 12}$ with entries

$$\mathbf{K}_{ij} = \begin{cases} k_{ji} & \text{if } i \neq j , \quad (j,i) \in \mathcal{E} ,\\ 0 & \text{if } i \neq j , \quad (j,i) \notin \mathcal{E} , \\ -\sum_{(i,l)\in \mathcal{E}} k_{il} & \text{if } i = j . \end{cases}$$
(5)

Then our ODE abstraction of the system can be written in the following form [15]:

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x}) \ . \tag{6}$$

One set of linearly independent conservation constraints on the part quantities is:

$$\begin{array}{rcl}
x_3 - x_4 & = & N_1 \\
x_1 + x_5 + x_7 + x_8 + x_{F1} + x_{F2} & = & N_2 \\
x_2 + x_5 + x_7 + 2(x_8 + x_{F1} + x_{F2}) & = & N_3 \\
x_3 + x_6 + x_7 + x_{F1} + x_{F2} & = & N_4
\end{array}$$
(7)

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

Theorem 1: System (6) subject to (7) has a unique, stable equilibrium $\bar{\mathbf{x}} > \mathbf{0}$.

Proof: Each equilibrium of the system, $\{\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 $\mathbf{y}(\bar{\mathbf{x}}) = \mathbf{0}$ [15]. From definition (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 *deficiency* δ of a reaction network is the number of complexes minus the number of linkage classes, each of which is a set of complexes connected by reactions, minus the network rank, which is the rank of the matrix with rows $\mathbf{m}_i - \mathbf{m}_j$, $(i, j) \in \mathcal{E}$ [1]. Network (3) has 12 complexes, 6 linkage classes, and rank 6; hence, $\delta = 0$. Also, the network is *weakly reversible* because whenever there is a directed arrow pathway from complex *i* to complex *j*, there is one from *j* to *i*. 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 [16].

IV. RATE OPTIMIZATION

We consider the problem of designing the system described by model (6) subject to (7) to produce desired 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 quantities, \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 (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 quantities 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

$$\alpha \equiv x_{F1}^d / (x_{F1}^d + x_{F2}^d) . \tag{8}$$

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) and definition (8) 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 1 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 the following constraint on \mathbf{K} ,

$$\mathbf{MKy}(\mathbf{x}^{\mathbf{d}}) = \mathbf{0} \ . \tag{9}$$

We quantify the time to converge to $\mathbf{x}^{\mathbf{d}}$ in terms of the system *relaxation times* τ_i , i = 1, ..., 6, the times in which different modes (dynamically independent variables) of the system converge to a stable equilibrium after perturbation [17], [18]. 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 [19] [20]. To estimate the τ_i , we reformulate the system in terms of new variables. Define v_i , i = 1, ..., 6, as the difference between the forward and reverse fluxes associated with reaction *i* in system (3). For example, $v_1 = k_1 x_1 x_2 - k_2 x_5$. Let $\mathbf{v}(\mathbf{x}) = [v_1 \dots v_6]^T$ and let $\mathbf{S} \in \mathbb{R}^{6 \times 10}$ denote the stoichiometric matrix of the system, which is defined such that model (6) can be written as [17]:

$$\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x}) \ . \tag{10}$$

The dynamical properties of a CRN are often analyzed by linearizing the ODE model of the system about an equilibrium and studying the properties of the associated Jacobian matrix $\mathbf{J} = \mathbf{SG}$, where the entries of \mathbf{G} are $\mathbf{G}_{ij} = dv_i/dx_j$ [18]. 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 rates that minimize the largest λ_i . However, in our system it is very difficult to find analytical expressions for the λ_i . We use an alternative estimate of relaxation time that is also derived by linearizing the system around its equilibrium \mathbf{x}^{d} [17],

$$\tau_j = \left(\sum_{i=1}^{10} (-\mathbf{S}_{ij}) \frac{dv_j}{dx_i}\right)_{\mathbf{x}=\mathbf{x}^{\mathbf{d}}}^{-1} .$$
(11)

Each reaction j in system (3) is of the form $X_k + X_l \rightleftharpoons_{k_j^-}^{k_j^+} X_m$. Thus, $v_j = k_j^+ x_k x_l - k_j^- x_m$, and the entries of column j in **S** are all 0 except for $\mathbf{S}_{kj} = \mathbf{S}_{lj} = -1$ and

$$\mathbf{S}_{mj} = 1$$
. Then according to equation (11), the relaxation time for each reaction is
 $\tau_i = (k_i^+ (x_k^d + x_l^d) + k_i^-)^{-1}$. (12)

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 (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:

$$\begin{array}{ll} \textbf{[P]} & \mbox{maximize} & \frac{1}{6}\sum_{j=1}^{6}\tau_{j}^{-1} \\ & \mbox{subject to} & \mbox{MK}(\mathbf{p})\mathbf{y}(\mathbf{x}^{\mathbf{d}}) = \mathbf{0}, \quad \mathbf{0} \leq \mathbf{p} \leq \mathbf{1} \ . \end{array}$$

Problem P is a linear program, which can be solved efficiently. For comparison, we also implemented a Monte Carlo method to find the $\mathbf{k}(\mathbf{p})$ that *directly* minimizes the convergence time. We measure the degree of convergence to $\mathbf{x}^{\mathbf{d}}$ by $\Delta(\mathbf{x}) = ||\mathbf{x} - \mathbf{x}^{\mathbf{d}}||_2$ and say that one system converges faster than another if it takes less time t_β for $\Delta(\mathbf{x})$ to decrease to some small fraction β , here defined as 0.1, of its initial value. At each iteration, $\mathbf{k}(\mathbf{p})$ is perturbed by a random vector and projected onto the null space of linearly independent rows of a matrix N defined such that $\mathbf{N}\mathbf{k} = \mathbf{M}\mathbf{K}\mathbf{y}(\mathbf{x}^{\mathbf{d}}) = \mathbf{0}$. Once $\mathbf{k}(\mathbf{p})$ also satisfies $\mathbf{0} \le \mathbf{p} \le \mathbf{1}$, it is used to simulate model (6), and a Newton scheme is used to compute the exact time $t_{0,1}$ when $\Delta(\mathbf{x}) = 0.1\Delta(\mathbf{x}^0)$.

V. RESULTS

A. Optimization of rates $\mathbf{k}(\mathbf{p})$

To investigate the effect of rate optimization on the convergence time of model (6), we generated non-optimal rates, which satisfy constraint (9) and $\mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$ but are not optimized for some objective, and computed rates using Problem P and the Monte Carlo method. The non-optimal and Problem P rates were calculated for $\alpha \in \{0.01, 0.02, \ldots, 0.99\}$, and the Monte Carlo rates for $\alpha \in \{0.1, 0.2, \ldots, 0.99\}$. We set $\mathbf{x}^{\mathbf{0}} = [60\ 120\ 60\ 60\ \mathbf{0}]^T$ and $\mathbf{x}^{\mathbf{d}} = [0.5\ 2.5\ 1\ 1\ 0.5\ 1\ 1\ 1\ 57\alpha\ 57(1-\alpha)]^T$. Problem P produced the same rates for each α (for instance, $p_i^+ = 1$ $\forall i$) except for the rates 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.

Fig. 5 compares the system convergence time $t_{0.1}$ for these different sets of rates. The Monte Carlo rates 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 program



Fig. 5. Time for model (6) to converge to $0.1\Delta(\mathbf{x^0})$ vs. α for optimized and non-optimal **k**. "Non-opt Ave" is the average of 100 $t_{0.1}$ corresponding to different random feasible **k** for each α .



Fig. 6. Evolution of final assembly ratios in model (6) for $\alpha = 0.1, 0.5, 0.9$ using non-optimal rates (light solid lines) and rates optimized by the Monte Carlo method (dark solid lines) and Problem P (dashed lines).

iteration for k to be considered close enough to optimal. The rates from Program 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. However, these $t_{0.1}$ are still much lower than the average $t_{0.1}$ (over 100 values) of the system using non-optimal rates.

We numerically integrated model (6) for $\alpha = 0.1, 0.5, 0.9$ using both optimized and non-optimal rates. The evolution of the model for each set of rates is shown in Fig. 6. For each α , the optimized models converge faster to the target assembly ratios than the non-optimal model.

B. Mapping rates onto the micro-continuous model

For $\alpha = 0.1, 0.5, 0.9$, we mapped the optimized and non-optimal rates onto the micro-continuous model to see



Fig. 7. Evolution of final assembly ratios in the micro-continuous model for $\alpha = 0.1, 0.5, 0.9$ using rates optimized by the Monte Carlo method (top row) and Problem P (center row) and non-optimal rates (bottom row). $x_{F1}^d + x_{F2}^d$ was computed as the equilibrium $x_{F1} + x_{F2}$ of model (6) with $\mathbf{x}^0 = [3 \ 6 \ 3 \ 3 \ 0]^T$.

whether the physical system would behave similarly to the reduced continuous model. We did this in the following way. Let R be a uniformly distributed random number between 0 and 1 and let Δt be the simulation timestep (32 ms). A robot carrying a part that can be disassembled according to process i computes R at each timestep and disassembles the part if $R < p_i^- \Delta t$. A robot about to begin assembly process i computes R and executes the assembly if $R < p_i^+ \Delta t$. Fig. 7 shows the time evolution of the micro-continuous model averaged over 30 runs for all sets of rates, using 15 robots and 15 parts (3 final assemblies).

Since the simulations are time-consuming to run, they were stopped well before the ODE models' end times in Fig. 6. This is why they do not attain full convergence. However, their qualitative behavior follows the same trends as Fig. 6. The runs using the Monte Carlo rates make the most progress toward the target ratios, the runs using the non-optimal rates make very little progress, and the runs using the Problem P rates display intermediate performance. Thus, in this realistic system model, the rates that are optimized in a much simpler model do indeed produce faster convergence than the non-optimal rates. Interestingly, Fig. 6 also predicts the crossing between final assembly amounts that occurs in the Problem P, $\alpha = 0.9$ run. Discrepancies between the simulations and model (6) are due to the factors described in Section III-B.

VI. CONCLUSIONS AND FUTURE WORK

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 task. By tuning these rates, we tune the performance of the system. This optimization relies on global stability properties of a specific class of CRN to which the ODE model belongs, and it is independent of the number of parts and robots. We map the rates onto probabilities of assembly and disassembly followed by the robots and find that the behavior of the resulting system is qualitatively predicted by the ODE model; it is expected that performance closer to the model would be observed for larger robot and part populations. One avenue of future work is to investigate the synthesis of the discrete assembly plans and incorporate feedback into the process. This direction draws inspiration from bio-molecular pathways in which intermediate subassemblies or molecules can promote or inhibit chemical reactions. We would like to be able to optimize the discrete assembly plan by constructing feedback loops to improve the yield rate.

REFERENCES

- M. Feinberg, "The existence and uniqueness of steady states for a class of chemical reaction networks," *Archive for Rational Mechanics* and Analysis, vol. 132, pp. 311–370, 1995.
- [2] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1490–1508, 2008.
- [3] E. Klavins, S. Burden, and N. Napp, "Optimal rules for programmed stochastic self-assembly," *Proc. Robotics: Science and Systems II*, pp. 9–16, 2007.
- [4] K. Hosokawa, I. Shimoyama, and H. Miura, "Dynamics of selfassembling systems: Analogy with chemical kinetics," *Artificial Life*, vol. 1, no. 4, pp. 413–427, 1994.
- [5] J. Werfel and R. Nagpal, "Three-dimensional construction with mobile robots and modular blocks," *Int'l. J. Robotics Research*, vol. 27, no. 3-4, pp. 463–479, 2008.
- [6] Á. Halász, M. A. Hsieh, S. Berman, and V. Kumar, "Dynamic redistribution of a swarm of robots among multiple sites," *Proc. Int'l. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2320–2325, 2007.
- [7] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Trans. on Robotics*, 2009, accepted.
- [8] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *Int'l. J. Robotics Research*, vol. 23, no. 4, pp. 415–436, 2004.
- [9] N. Correll and A. 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.
- [10] H. Hamann and H. Wörn, "A framework of space-time continuous models for algorithm design in swarm robotics," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 209–239, 2008.
- [11] D. Gillespie, "A rigorous derivation of the chemical master equation," *Physica A*, vol. 188, no. 1-3, pp. 404–425, 1992.
- [12] —, "Stochastic simulation of chemical kinetics," Annual Review of Physical Chemistry, vol. 58, pp. 35–55, 2007.
- [13] O. Michel, "Webots: Professional mobile robot simulation," Int'l. J. Advanced Robotic Systems, vol. 1, no. 1, pp. 39–42, 2004.
- [14] 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.
- [15] M. Chaves, E. D. Sontag, and R. J. Dinerstein, "Steady-states of receptor-ligand dynamics: a theoretical framework," *J. Theoretical Biology*, vol. 227, no. 3, pp. 413–28, 2004.
- [16] D. Siegel and D. MacLean, "Global stability of complex balanced mechanisms," J. Mathematical Chemistry, vol. 27, pp. 89–110, 2000.
- [17] R. Heinrich and S. Schuster, *The Regulation of Cellular Systems*. New York, NY: Chapman & Hall, 1996.
- [18] N. Jamshidi and B. O. Palsson, "Formulating genome-scale kinetic models in the post-genome era," *Molecular Systems Biology*, vol. 4, no. 171, pp. 1–10, 2008.
- [19] R. Heinrich, S. Schuster, and H.-G. Holzhutter, "Mathematical analysis of enzymic reaction systems using optimization principles," *Eur. J. Biochem.*, vol. 201, pp. 1–21, 1991.
- [20] S. Schuster and R. Heinrich, "Time hierarchy in enzymatic reaction chains resulting from optimality principles," *J. Theoretical Biology*, vol. 129, no. 2, pp. 189–209, 1987.