Stochastic Strategies for a Swarm Robotic Assembly System

Loïc Matthey, Spring Berman and Vijay Kumar

Abstract— **We present a decentralized, scalable approach to quickly assembling a group of heterogeneous parts into desired numbers of products using a swarm of robots. This activity forms the basis for a reconfigurable manufacturing system. The assembly plans are predetermined, and the system is implemented using a realistic robot simulator. Our approach is based on developing a continuous abstraction of the system 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 produce the desired aggregate behavior. We define methods for optimizing the rates to achieve fast convergence to target part amounts. We use the methods to produce continuous models that converge to different ratios of final products and test the resulting policies on a physical simulation with 15 robots and 15 parts.**

I. INTRODUCTION

We develop an approach to designing a reconfigurable manufacturing system in which a swarm of homogeneous robots assembles static parts into different types of products. The system must respond quickly to produce desired amounts of products, which can change at discrete points in time, from any initial set of parts. Since it is difficult, if not impossible, to efficiently control a large robot population through centralized algorithms, we employ a decentralized strategy in which robots operate autonomously and 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.

The robots move randomly within a closed arena, encountering parts and other robots at probability rates that are determined by the physical parameters of the system and by sensor and environmental noise. We model this behavior using a realistic 3D physics simulation, which we refer to as a *micro-continuous model* since it encompasses the continuous dynamics of individual robots. The system can be approximated as well-mixed, and the robot-part and robot-robot interactions are analogous to chemical reactions between molecules. Hence, like a set of chemical reactions, our system can be represented by the Stochastic Master Equation [1]. Various methods exist to numerically simulate the evolution of a system governed by this model [2], [3]. We call this the *complete macro-discrete model* because it describes a continuous-time Markov process whose states are the discrete numbers of system components. When there are large numbers of parts, the system can be abstracted to an ordinary differential equation (ODE) model, which we call a *macro-continuous model* since its state variables are continuous amounts of the parts.

Our work is similar in objective to studies on programmable self-assembly for modular robots [4]–[7]. The system in this work is a set of homogeneous triangular robots that move around randomly on an air table and assemble into components according to a plan that is constructed using *graph grammars*, which are rules that define interactions between robots. Graph grammars can be constructed automatically to produce a given predefined assembly [7]. The probabilities of a newly formed component to detach into different combinations of parts can be optimized to maximize the number of a desired assembly at equilibrium [5]. However, this optimization requires the enumeration of all system states reachable from the initial state.

Our approach to assembly system design allows for improved scalability and flexibility. We consider a scenario in which a heterogeneous set of parts is assembled into two types of final products. To provide theoretical guarantees on performance, we employ the "top-down" design methodology presented in [8]–[10] for reallocating a swarm of robots among a set of sites/tasks in a desired distribution. This methodology was applied to a linear model; here we extend it to a nonlinear (specifically, multi-affine) model with robot interactions. We construct this *complete macro-continuous model* of the system using the Chemical Reaction Network (CRN) framework [11], [12], which has been studied extensively for theoretical insight into biochemical systems. We compute reaction rate constants in the model from physical properties of the robots and environment and check that the model accurately predicts the results of the micro-continuous model. Then we simplify this abstraction to a *reduced macro-continuous model* with the same rates and use the model to optimize these rates for fast convergence to a target distribution of products, using an approach similar to [13], [14]. The optimization problem is independent of the number of parts and scales only with the number of rates. We simulate the reduced model with optimized rates for different target distributions, map the rates onto probabilities of assembly and disassembly which are used as control policies in the micro-continuous model, and observe this system's convergence behavior.

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.

Preprint submitted to 2009 IEEE International Conference on Robotics and Automation. Received September 15, 2008.

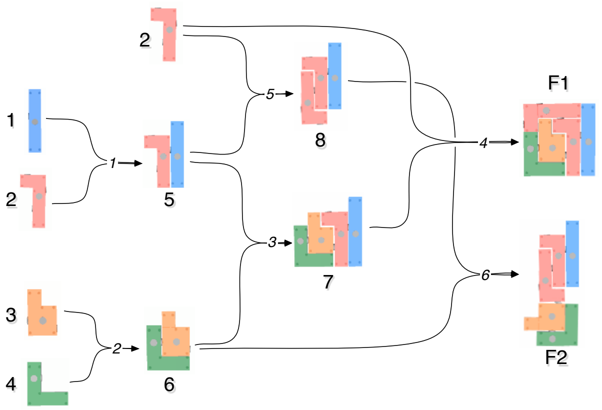

Fig. 1. Assembly plans for final assemblies F1 and F2.

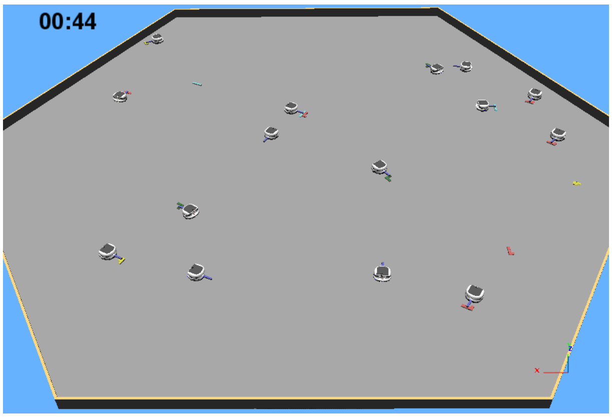

Fig. 2. Snapshot of the arena in the realistic physical simulation. Robots carry parts at the end of a protruding bar.

II. PROBLEM STATEMENT

A. Assembly task

There are four different types of parts, numbered 1 through 4, which can be combined to form larger parts according to the assembly plans in Fig. 1. Parts bond together through bi-directional connections at sites along their perimeters. The last step in each plan is the production of a final assembly, $F1$ or $F2$. The assembly task is executed by a group of robots in an arena that is sufficiently large to ignore the dynamics of small-scale interactions. Initially, robots and many copies of parts of type 1 through 4 are randomly scattered throughout the arena. There are exactly as many of these parts as are needed to create a specified number of final assemblies, and the number of robots equals the total number of scattered parts. Each robot knows the assembly plans a-priori and 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.

Our objective is to define robot controllers for moving around the arena and for picking up, assembling, and disassembling parts so that the robots produce target numbers of final assemblies as quickly as possible.

B. Micro-continuous model

We implement the assembly task in the robot simulator Webots [15], 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 a radio emitter and 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 the walled hexagonal arena shown in Fig. 2.

We use a control policy for the robots that is inspired by chemical processes: random movement patterns with probabilistic assemblies upon encounter, as well as random disassemblies. Our models assume that the system is well-mixed; to achieve this property, 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 starting an assembly and performing a disassembly.

III. MACRO-CONTINUOUS MODELS

A. Definitions

Interactions between parts and robots in the assembly system are modeled in the form of a Chemical Reaction Network (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*, which are 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 that complex i transforms into complex j, and is associated with a positive *reaction rate constant*.

Each part of type i in Fig. 1 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.

Preprint submitted to 2009 IEEE International Conference on Robotics and Automation. Received September 15, 2008.

B. Complete macro-continuous model

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

$$
\begin{array}{llll}
X_R + X_1^u \xrightarrow{e_1} X_1^c & & X_R + X_5^u \xrightarrow{e_5} X_5^c & \\
X_R + X_2^u \xrightarrow{e_2} X_2^c & & X_R + X_6^u \xrightarrow{e_6} X_6^c & \\
X_R + X_3^u \xrightarrow{e_3} X_3^c & & X_R + X_7^u \xrightarrow{e_7} X_7^c & \\
X_R + X_4^u \xrightarrow{e_4} X_4^c & & X_R + X_8^u \xrightarrow{e_8} X_8^c & \\
X_1^c + X_2^c \xrightarrow{k_1^+} X_5^c + X_R & & X_2^c + X_7^c \xrightarrow{k_4^+} X_{F1}^c + X_R & \\
X_3^c + X_4^c \xrightarrow{k_2^+} X_6^c + X_R & & X_2^c + X_5^c \xrightarrow{k_5^+} X_8^c + X_R & \\
X_5^c + X_6^c \xrightarrow{k_3^+} X_7^c + X_R & & X_6^c + X_8^c \xrightarrow{k_6^+} X_{F2}^c + X_R & \\
X_5^c \xrightarrow{k_1^-} X_1^c + X_2^u & & X_{F1}^c \xrightarrow{k_4^-} X_7^c + X_2^u & \\
X_6^c \xrightarrow{k_2^-} X_3^c + X_4^u & & X_8^c \xrightarrow{k_5^-} X_5^c + X_2^u & \\
X_7^c \xrightarrow{k_3^-} X_6^c + X_5^u & & X_{F2}^c \xrightarrow{k_6^-} X_8^c + X_6^u & (1)
\end{array}
$$

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 j, and k_i^- is the rate of disassembly process j. We theoretically estimate these rates as functions of the following probabilities:

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

p^e is the probability that a robot encounters a part or another robot. Using the assumption that robots and parts are distributed uniformly throughout the arena, we calculate p^e from the geometrical approach that is used to compute probabilities of molecular collisions [16], [17]: $p^e \approx vTw/A$, 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 $3\ m$), $w = 1.2\ m$, $v_R = 0.128\ m/s$ 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 [2]. This is illustrated in the next section. We numerically integrate this macro-continuous model with the rates we calculated and also use the StochKit toolbox [18] to efficiently perform a stochastic simulation of the macro-discrete model. We compare the results to those for the micro-continuous model in Fig. 3, using $p_i^+ = 1,\ p_i^- = 0\ \ \forall i$. The results for all

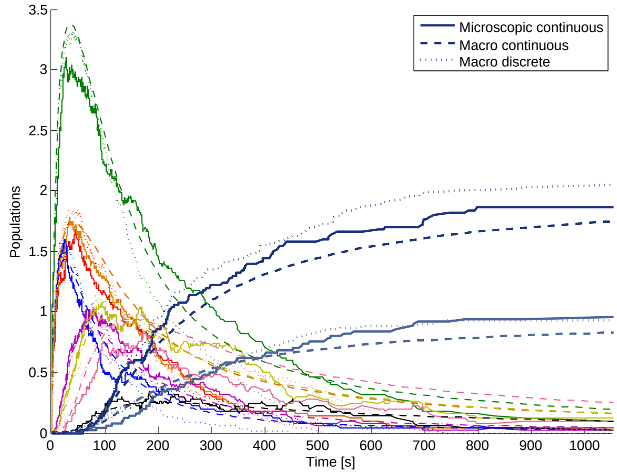

Fig. 3. Part populations in the complete macro-continuous, macro-discrete, and micro-continuous models for 3 final assemblies and 15 robots. Macro-discrete and micro-continuous results are averaged over 100 runs.

models are very similar, although discrepancies arise from two factors. First, certain events that happen in the physical simulation are not modeled: parts sticking together or being wedged against a wall, and deviations from the assumption of uniform spatial distribution. Second, the ODE approximation is valid only for large numbers of parts, and the system modeled had only 15 robots and 15 parts. Overall, the macro-continuous model accurately predicts the evolution of part populations, and hence we can use it to design the rates to direct the system's behavior, provided that the system has very large numbers of parts.

C. Reduced macro-continuous model

We simplify the complete model by abstracting away robots and retaining only interactions between parts, assuming that the time to find a part is small and there are more robots than parts:

$$
\begin{array}{lll}
X_1 + X_2 \underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} X_5 & X_2 + X_7 \underset{k_4^-}{\overset{k_4^+}{\rightleftharpoons}} X_{F1} & \\
X_3 + X_4 \underset{k_2^-}{\overset{k_2^+}{\rightleftharpoons}} X_6 & X_2 + X_5 \underset{k_5^-}{\overset{k_5^+}{\rightleftharpoons}} X_8 & \\
X_5 + X_6 \underset{k_3^-}{\overset{k_3^+}{\rightleftharpoons}} X_7 & X_6 + X_8 \underset{k_6^-}{\overset{k_6^+}{\rightleftharpoons}} X_{F2} & (3)
\end{array}
$$

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:

$$
\begin{aligned}
\mathbf{y}(\mathbf{x}) = \quad & [x_1x_2\ \ x_5\ \ x_3x_4\ \ x_6\ \ x_2x_7\ \ x_{F1} \\
& x_5x_6\ \ x_7\ \ x_2x_5\ \ x_8\ \ x_6x_8\ \ x_{F2}]^T \ . \qquad (4)
\end{aligned}
$$

We also define a matrix $\mathbf{M} \in \mathbb{R}^{10\times 12}$ in which each entry 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

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

Then our ODE abstraction of the system can be written in the following form [19]:

$$\dot{\mathbf{x}} = \mathbf{MKy}(\mathbf{x}) \ . \tag{6}$$

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

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

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 $\mathbf{y}(\bar{\mathbf{x}}) = \mathbf{0}$ [19]. 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 $\mathbf{m}_i - \mathbf{m}_j$, $(i,j) \in \mathcal{E}$ [20]. 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 [21]. ∎

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 rates* k_i^+, k_i^-, $i = 1, ..., 6$, which define an *optimal rate matrix* $\mathbf{K}^*$ according to (5), that minimize the convergence time of the system to a vector of target part quantities, $\mathbf{x^d}$. Note that 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 $\mathbf{K}$,

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

Now we consider the aspect of minimizing the convergence time to $\mathbf{x^d}$. We quantify this time in terms of the system *relaxation times* τ_i, $i = 1, ..., 6$, the times in which different modes (dynamically independent variables) of the the system converge to a stable equilibrium after perturbation [22], [23]. 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 [24]. For instance, one such measure was minimized in the optimization of rates for the linear chain in [25].

To estimate the relaxation times, we first 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 $\mathbf{v}(\mathbf{x}) = [v_1 \ ... \ v_6]^T$ 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 [26]:

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

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 $G_{ij} = dv_i/dx_j$ [23]. 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 $\mathbf{x^d}$ [26],

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

Each reaction j in system (3) is of the form $X_k + X_l \underset{k_j^-}{\overset{k_j^+}{\rightleftharpoons}} X_m$. Thus, $v_j = k_j^+ x_k x_l - k_j^- x_m$, and the entries of column j in $\mathbf{S}$ are all 0 except for $s_{kj} = s_{lj} = -1$ and $s_{mj} = 1$. Then according to equation (11), the relaxation time for each reaction is

$$\tau_j = (k_j^+(x_k^d + x_l^d) + k_j^-)^{-1} \ . \tag{12}$$

Define $\mathbf{k} \in \mathbb{R}^{12}$ as the vector of all rates k_i^+, k_i^-. Using equation (12), we define two possible objective functions

Preprint submitted to 2009 IEEE International Conference on Robotics and Automation. Received September 15, 2008.

$f : \mathbb{R}^{12} \rightarrow \mathbb{R}$, the average τ_j^{-1} and the minimum τ_j^{-1}, to *maximize* in order to produce fast convergence to $\mathbf{x^d}$:

$$f_{ave}(\mathbf{k}) = \frac{1}{6}\sum_{j=1}^{6}\tau_j^{-1} , \quad (13)$$

$$f_{min}(\mathbf{k}) = \min\{\tau_1^{-1}, \ldots, \tau_6^{-1}\} . \quad (14)$$

Finally, we write the rates k_i^+, k_i^- in terms of the tunable probabilities p_i^+, p_i^- using equation (2) and define these probabilities as the optimization variables. Let $\mathbf{p} \in \mathbb{R}^{12}$ be the vector of all p_i^+, p_i^-. Then the optimization problem can be posed as **Problem P** below. It will be referred to as **Problem P1** when $f = f_{ave}$ and as **Problem P2** when $f = f_{min}$.

[P] maximize $f(\mathbf{k}(\mathbf{p}))$
subject to $\mathbf{MK}(\mathbf{p})\mathbf{y}(\mathbf{x^d}) = \mathbf{0}, \quad \mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$.

Problems P1 and P2 are both linear programs, which can be solved efficiently. To check that they do in fact minimize convergence time, we implemented a Monte Carlo method [27], which is more computationally expensive, to find the $\mathbf{k}(\mathbf{p})$ that directly minimizes this time. We measure the degree of convergence to $\mathbf{x^d}$ by $\Delta(\mathbf{x}) = ||\mathbf{y}(\mathbf{x}) - \mathbf{y}(\mathbf{x^d})||_2$ and say that one system converges faster than another if it takes less time for $\Delta(\mathbf{x})$ to decrease to some small fraction, 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 $\mathbf{N}$ defined such that $\mathbf{Nk} = \mathbf{MKy}(\mathbf{x^d}) = \mathbf{0}$. Once $\mathbf{k}(\mathbf{p})$ also satisfies $\mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$, it is used to simulate the reduced macro-continuous ODE model to find $\Delta(\mathbf{x})$ after some time. Since the system is stable by Theorem 1, $\Delta(\mathbf{x})$ always decreases monotonically with time, so a Newton scheme can be 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

We solved optimization problems P1 and P2 for $\alpha \in \{0.01, 0.02, \ldots, 0.99\}$ using $\mathbf{x^0} = [60\ 120\ 60\ 60\ \mathbf{0}]^T$ and $\mathbf{x^d} = [0.5\ 2.5\ 1\ 1\ 0.5\ 1\ 1\ 1\ 57\alpha\ 57(1-\alpha)]^T$. As Table I shows, the computed rates are constant for each α except for those corresponding to assembly and disassembly processes 4 and 6. This means that the system is flexible enough to yield any α when only the rates of assembling and breaking apart the final assemblies are modified. We also ran the Monte Carlo program for $\alpha = 0.4$. For the optimal set of rates, which were very similar to those for Problem P1, $t_{0.1} = 4.69$, as compared to 4.68 for Problem P1 and 8.49 for P2. This provides evidence that Problem P1 is minimizing the system convergence time.

We integrated the reduced macro-continuous model for $\alpha = 0.4$ and $\alpha = 0.8$ using the optimized rates from Problems P1 and P2 and a set of non-optimal rates that were chosen to satisfy constraint (9) and $\mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$ but were not optimized for some objective. The evolution of the model for each set of rates is shown in Fig. 4, with time in log-scale.

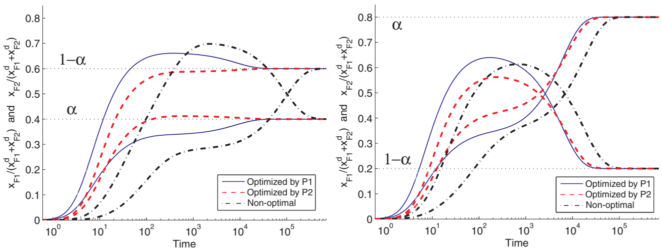

Fig. 4. Evolution of final assembly ratios in the reduced macro-continuous model for $\alpha = 0.4$ and $\alpha = 0.8$, using rates optimized by Problems P1 and P2 and non-optimal rates.

The optimized systems first quickly converge to an "intrinsic equilibrium" at $t \approx 10^2$ that is independent of α and then redistribute much more slowly to the target ratios, which are attained for all set of rates. For both α, the optimized models converge faster to equilibrium than the non-optimal model. For $\alpha = 0.4$, the non-optimal model displays a large over- and under-shooting of the target ratios, whereas the optimized models converge quickly with little or none of this effect. Problem P2 produces a more efficient system in this respect, although both optimized models converge at comparable rates.

B. Mapping rates onto the micro-continuous model

For $\alpha = 0.4$, we mapped the rates optimized by Problem P1 and the non-optimal rates onto the micro-continuous model to see 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. 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. 5 shows the time evolution of the micro-continuous model averaged over 30 runs for both sets of rates, using 15 robots and 15 parts (3 final assemblies). The non-optimal model converges to the target ratios but initially over- and under-shoots x_{F1}^d and x_{F2}^d, respectively, as in Fig. 4. The ratios in the optimized model rise quickly to the target ratios but then deviate from them toward 0.5.

The discrepancy between the optimized model results and the target ratios can be explained by inaccurate modeling of some low-level effects. In the micro-continuous model, when a robot disassembles a part, a nearby robot is likely to pick up the fallen component and quickly reassemble the original part with the other robot, which generally hasn't traveled far. This increases the rate of assembly of the part and violates the well-mixed assumption. In addition, robots that pick up dropped parts spend a significant amount of time carrying them before reassembly, a factor that was abstracted away in the reduced model. Adding more robots tends to alleviate this effect. Random failures at disassembly, incorrect internal state definitions by the robots and parts, and collision errors by the physics engine sometimes occur in the simulation

TABLE I

VALUES OF OPTIMIZED RATES FOR VARYING α. *Continuous* RATES EVOLVE CONTINUOUSLY WITH RESPECT TO α.

Reaction i	1	2	3	4	5	6
P1 Optimized p_i^+	1.0					
P1 Optimized p_i^-	0.01885	0.00754	0.00377	*continuous*	0.00942	*continuous*
P2 Optimized p_i^+	0.36	0.666	1.0	*continuous*	0.4705	*continuous*
P2 Optimized p_i^-	0.006855	0.005027	0.00377	*continuous*	0.00443	*continuous*

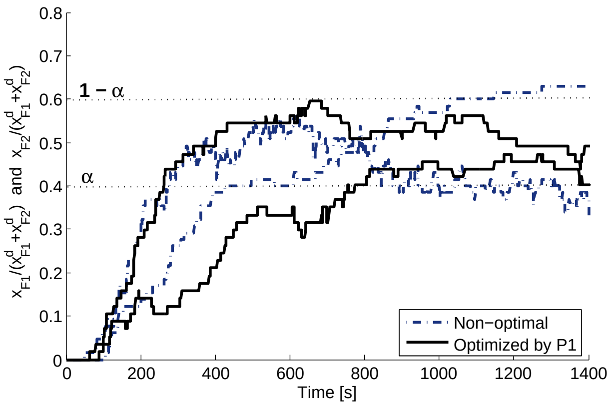

Fig. 5. Evolution of final assembly ratios in the micro-continuous model for $\alpha = 0.4$, using rates optimized by Problem P1 and non-optimal rates.

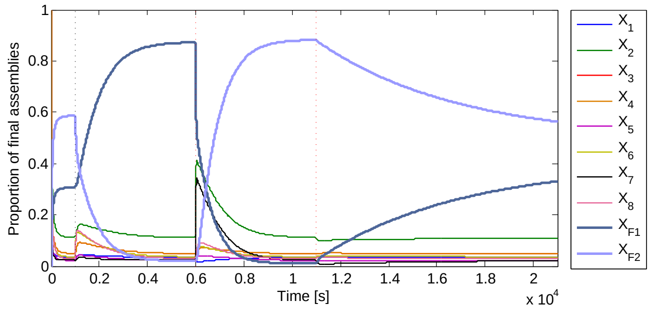

Fig. 6. Online adaptation of the reduced macro-continuous model to changes in target equilibrium, which occur at the vertical dashed lines.

but are not modeled. Finally, the continuous abstraction may not accurately capture the system behavior for such a low number of parts. However, it becomes more computationally expensive to simulate the system as the number of parts and robots increase.

C. Reconfigurable manufacturing

We apply our methodology to a "green manufacturing" task, in which finished products are recycled into new products in ratios that depend on the current demand. We perform a simulation of the reduced macro-continuous model that corresponds to a situation in which robots switch between sets of optimized rates at discrete points in time to produce different target numbers of parts. The sequence of target α is $0.4, 0.99, 0.01$, and 0.5, and $\mathbf{x^0}$ is the same as in Section V-A. The system evolution is shown in Fig. 6 using rates from Problem P1. The system quickly adapts to new target ratios, although some take longer to attain than others (e.g. $\alpha = 0.5$). The results demonstrate that high-level control of the system is possible in real-time.

VI. DISCUSSION AND FUTURE WORK

We have shown an approach to systematically deriving decentralized control policies for a swarm of robots to perform an assembly task with a desired outcome. We represent the system using a multi-level modeling methodology. A micro-continuous model that is implemented using a realistic physics simulator is abstracted to a macro-continuous ODE model that can be used to optimize the assembly and disassembly rates of the system. Our optimization scheme relies on global stability properties of a specific class of CRN to which the macro-continuous model belongs. We implement the optimization as a linear program with constraints on target amounts of parts at equilibrium. We propose two objective functions based on an estimate of the system convergence time to its equilibrium. We simulate the macro-continuous model and observe that it achieves target final assembly ratios faster with optimized rates than with non-optimal rates, and that it can quickly respond to changes in the targets. Finally, we map the rates onto probabilities of assembly and disassembly in the micro-continuous model. We find that the resulting system can produce the target ratios, although discrepancies arise due to violation of the well-mixed property, low part numbers, and failure to capture certain physical effects in the model.

Future work includes tuning the macro-continuous model to more closely match the physical system. Also, we can assess the performance of our optimization results by continuing to compare them to results from other optimization techniques. A way to enforce target amounts of only the parts of interest, rather than all part populations, would allow more flexibility in system design. Finally, we note that in this work we use a subset of all possible assembly and disassembly steps and do not address the discrete optimal design of the assembly plans themselves. We wish to study whether we can apply our continuous optimization scheme to find the pathways that are promoted and inhibited in an expanded assembly plan to produce target amounts of products as quickly as possible.

REFERENCES

[1] D. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Comput. Phys.*, vol. 22, pp. 403–434, 1976.

[2] D. T. Gillespie, "Stochastic simulation of chemical kinetics," *Annual Review of Physical Chemistry*, vol. 58, pp. 35–55, Jan 2007.

[3] J. Puchalka and A. Kierzek, "Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks," *Biophysical Journal*, Jan 2004.

[4] J. Bishop, S. Burden, E. Klavins, and R. Kreisberg, "Programmable parts: a demonstration of the grammatical approach to self-organization," *Intelligent Robots and Systems*, Jan 2005.
[5] E. Klavins, "Programmable self-assembly," *Control Systems Magazine*, Jan 2007.
[6] J. McNew, E. Klavins, and M. Egerstedt, "Solving coverage problems with embedded graph grammars," *Hybrid Systems: Computation and Control*.
[7] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," *Urbana*.
[8] S. Berman, A. Halasz, V. Kumar, and S. Pratt, "Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system," *Swarm Robotics SAB 2006 International Workshop*, Jan 2006.
[9] S. Berman, Á. Halász, V. Kumar, and S. Pratt, "Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, 2007, pp. 2318–2323.
[10] A. Halasz, M. Hsieh, S. Berman, and V. Kumar, "Dynamic redistribution of a swarm of robots among multiple sites," *Intelligent Robots and Systems*, Jan 2007.
[11] M. Feinberg, "Lectures on chemical reaction networks," *Notes of lectures given at the Mathematics Research Center*, 1979.
[12] D. J. Wilkinson, "Stochastic modelling for systems biology," p. 254, Jan 2006.
[13] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Navigation-based optimization of stochastic deployment strategies for a robot swarm to multiple sites," accepted to 2008 Conf. on Dec. and Control (CDC).
[14] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimal stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, 2008, under review.
[15] O. Michel, "Webotstm: Professional mobile robot simulation," *Arxiv preprint cs.RO*, Jan 2004.
[16] D. Gillespie, "A rigorous derivation of the chemical master equation," *Physica A*, Jan 1992.
[17] N. Correll and A. Martinoli, "System identification of self-organizing robotic swarms," *Proc. of the Eight Int. Symp. on Distributed Autonomous ...*, Jan 2006.
[18] H. Li, Y. Cao, and L. Petzold, "Stochkit: A stochastic simulation toolkit," *cs.ucsb.edu*.
[19] M. Chaves, E. D. Sontag, and R. J. Dinerstein, "Steady-states of receptor-ligand dynamics: a theoretical framework," *Journal of theoretical biology*, vol. 227, no. 3, pp. 413–28, Apr 2004.
[20] M. Feinberg, "The existence and uniqueness of steady states for a class of chemical reaction networks," *Archive for Rational Mechanics and Analysis*, Jan 1995.
[21] D. Siegel and D. MacLean, "Global stability of complex balanced mechanisms," *Journal of Mathematical Chemistry*, vol. 27, pp. 89–110, 2000.
[22] R. Heinrich and S. M. Rapoport, "Metabolic regulation and mathematical models," *Prog. Biophys. Molec. Biol.*, vol. 32, pp. 1–82, 1977.
[23] 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.
[24] 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.
[25] S. Schuster and R. Heinrich, "Time hierarchy in enzymatic reaction chains resulting from optimality principles," *Journal of theoretical biology*, vol. 129, no. 2, pp. 189–209, Nov 1987.
[26] R. Heinrich and S. Schuster, *The Regulation of Cellular Systems*, 1996.
[27] D. P. Landau and K. Binder, *A guide to Monte-Carlo simulations in statistical physics*. Cambridge Univ. Press, 2000.