Abstractions and Algorithms for Assembly Tasks with Large Numbers of Robots and Parts

Spring Berman and Vijay Kumar

***Abstract*— We present a decentralized, scalable approach to designing a reconfigurable manufacturing system in which a swarm of robots assembles heterogeneous parts into target amounts of products. The sequence of part assemblies is determined by interactions between robots in a decentralized manner in real time. Our methodology is based on deriving a continuous abstraction of the system from chemical reaction models and formulating the strategy as a problem of selecting rates of assembly and disassembly. The rates are mapped onto probabilities that define stochastic control policies for individual robots, which then produce the desired aggregate performance. We illustrate our approach using a physics-based simulator with examples involving 15 robots and two types of final products.**

I. Introduction

We consider a scenario in which a large supply of heterogeneous parts must be assembled into desired 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 arena, 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 are parts are uniformly distributed throughout the arena, the system can be modeled as being spatially homogeneous, which allows us to describe it as a Chemical Reaction Network (CRN). We control the equilibrium part populations by designing the robot probabilities of executing assemblies and disassemblies; when 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 robots.

Previous assembly systems that are based on random collisions between parts have also been modeled as CRN's [1]–[3]. In [3], modules that have combined can detach into different parts with certain probabilities, which are optimized to maximize the equilibrium yield of one assembly type. The optimization does not scale well with the number of parts because it requires the enumeration of all reachable states.

We synthesize the robot control policies using a "top-down" design methodology (Fig. 1) similar to that in [4], where we applied it to the problem of reallocating robots among a set of tasks. 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 a continuous-time Markov process, the *macro-discrete model*, whose states are discrete populations of parts and robots. When these populations are large, the system can be abstracted to an ordinary differential equation (ODE) model, the *macro-continuous model*, whose state variables are continuous amounts of parts and robots. We design the parameters of this model for fast production of target amounts of assemblies; this optimization is independent of the number of robots and parts. When the parameters are mapped onto robot probabilities of assembly and disassembly, the average evolution of products over time follows the prediction of the macro-continuous model.

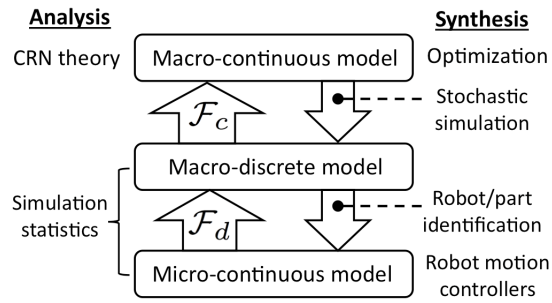

Fig. 1. Levels of abstraction of the assembly system with analysis and synthesis methodologies. The high-dimensional micro-continuous model is mapped to lower-dimensional models through the abstractions $\mathcal{F}_d$ and $\mathcal{F}_c$ using the theoretical justification in [9].

Recent work on modeling robot swarms [5]–[7] has also developed accurate macroscopic models of physical systems. Our novel contribution is our control synthesis methodology, which provides theoretical guarantees on performance. In the following sections, we illustrate our approach on an example system. For further details on this work, see [8].

II. Micro-continuous model

The assembly task is implemented in the robot simulator Webots [10], which uses the Open Dynamics Engine to accurately simulate physics. A group of Khepera III robots combines parts of four different types into larger parts according to the assembly plans in Fig. 2 to ultimately form final assemblies $F1$ and $F2$. The number of robots is at least the total number of scattered parts, and the arena is sufficiently large to prevent robot crowding. Each robot has infra-red distance sensors for collision avoidance and

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

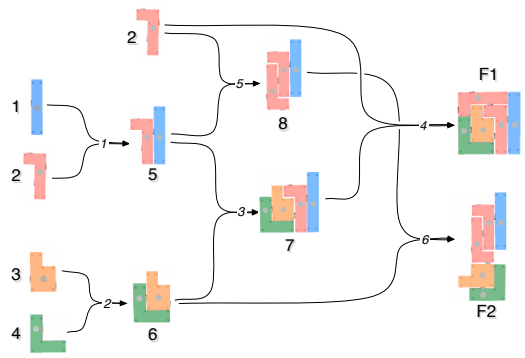

Fig. 2. Assembly plans for final assemblies $F1$ and $F2$.

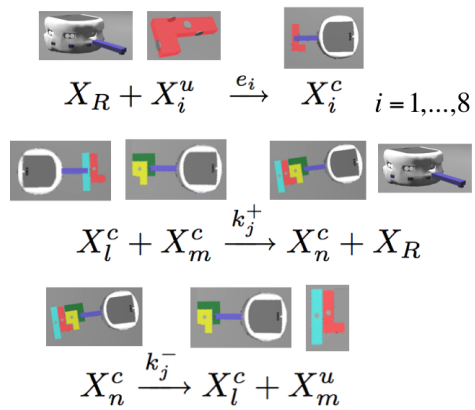

Fig. 3. CRN representing all actions in the assembly system.

a protruding bar with a rotational servo at the tip. Both robots and parts have an infra-red emitter/receiver for local communication and for computing relative bearing. A robot that detects a part approaches it until a magnet on the servo bonds to a magnet on the part's top face. The servo rotates the part into the correct orientation for assembly. When the robot finds a robot that is carrying a compatible part, the two robots align their parts and approach each other until the parts bond via magnets on their side faces. One robot carries off the newly assembled part; 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.

III. Macro-Continuous Model

Interactions between parts and robots are modeled as a CRN. The *complexes* are the combinations of parts and/or robots that appear before and after reaction arrows. Each reaction pathway is associated with a positive *rate constant*. X_i denotes a part of type i, and X_R symbolizes a robot. X_i may be further classified as X_i^u, an unclaimed part, 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 the system. Then $\mathbf{x}(t) \in \mathbb{R}^M$ is the vector of species populations, which are represented as continuous functions of time t.

The CRN in Fig. 3 models all possible actions in the system. The rate constants are estimated as functions of the following probabilities:

$$e_i = p^e \ , \qquad k_j^+ = p^e \cdot p_j^a \cdot p_j^+ \ , \qquad k_j^- = p_j^- \ . \qquad (1)$$

p^e is the probability that a robot encounters a part or another robot. The uniform distribution of robots and parts in the arena allows us to calculate p^e from the geometrical approach that is used to compute probabilities of molecular collisions [11]: $p^e \approx vTw/A$, where v is the average robot speed, T is a timestep, A is the arena area, and w is twice a robot's communication radius. p_j^a is the probability of two robots successfully completing assembly process j; it depends on the part geometries. Finally, 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.

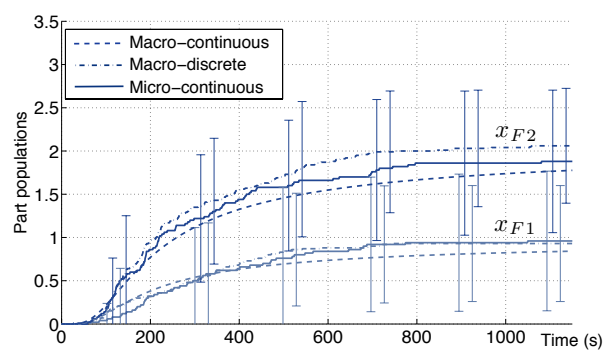

Fig. 4. $F1$, $F2$ populations in three models; the macro-discrete and micro-continuous models are each averaged over 100 runs. The system has 15 robots and parts for 3 final assemblies. Error bars show standard deviations.

We constructed the macro-continuous model (see below for details) and numerically integrated it with p_e, p_j^a measured from the micro-continuous model and $p_j^+ = 1$, $p_j^- = 0 \ \forall j$. We also used the StochKit toolbox [12] to efficiently perform a stochastic simulation of the macro-discrete model. Fig. 4 compares the evolution of final products in all three models. Discrepancies among the models are due in part to the fact that the ODE model is most accurate for very large populations, while the system has relatively low numbers of parts and robots (15 each) so that it would not be too computationally expensive to simulate. If the populations were increased, the micro- and macro-continuous models should correspond more closely. Also, assembly disruption sometimes occurs in Webots due to erroneous collisions and robot interference; we do not model these failures.

Nevertheless, the macro-continuous model predicts the $F1$, $F2$ populations fairly accurately, so we can use it to design p_j^+, p_j^- to direct the system behavior. In order to be able to prove convergence properties of the model, we reduce its dimensionality by abstracting away robots, 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 CRN becomes a set of reactions of the form $X_l + X_m \underset{k_j^-}{\overset{k_j^+}{\rightleftharpoons}} X_n$, $j = 1, ..., 6$. 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:

$$\mathbf{y}(\mathbf{x}) = \quad [x_1x_2 \;\; x_5 \;\; x_3x_4 \;\; x_6 \;\; x_2x_7 \;\; x_{F1} \; x_5x_6 \;\; x_7 \;\; x_2x_5 \;\; x_8 \;\; x_6x_8 \;\; x_{F2}]^T \; . \quad (2)$$

We also define a matrix $\mathbf{M} \in \mathbb{R}^{10\times 12}$ whose entries are the coefficients of part types in the complexes and a matrix $\mathbf{K} \in \mathbb{R}^{12\times 12}$ constructed from k_j^+ and k_j^-. Then the macro-continuous model can be written in the following form:

$$\dot{\mathbf{x}} = \mathbf{MKy}(\mathbf{x}) \; . \quad (3)$$

The model is subject to four linearly independent conservation constraints on the x_i. Results from CRN theory [13] can be used to show that the model has a unique, positive, globally asymptotically stable equilibrium. Thus, we can achieve a target distribution of part quantities, $\mathbf{x^d}$, from any initial distribution $\mathbf{x^0}$ by computing $\mathbf{K}$ to satisfy $\mathbf{MKy}(\mathbf{x^d}) = \mathbf{0}$, which defines $\mathbf{x^d}$ as the model's equilibrium.

IV. Controller Synthesis

We defined $\mathbf{x^d}$ to contain a large number of final assemblies compared to other part quantities. A set of k_j^+, k_j^- was randomly selected to satisfy the above constraint on $\mathbf{K}$ and the constraint that $p_j^+, p_j^- \in [0,1]$ in equation (1). Two other sets of k_j^+, k_j^- were computed as optimization variables that minimize a measure of the convergence time of model (3) to $\mathbf{x^d}$ subject to these constraints. One set was obtained from a linear program that maximizes the average inverse *relaxation time*, the time for a system mode to converge to equilibrium after perturbation, which was estimated by linearizing the model around $\mathbf{x^d}$ [14]. Another set was computed using a Monte Carlo method that directly minimizes the time for the model to reach a certain level of convergence.

We calculated random and optimized k_j^+, k_j^- for $\alpha \equiv x_{F1}^d/(x_{F1}^d + x_{F2}^d) = 0.1, 0.5, 0.9$ and mapped them onto the micro-continuous model as follows. A robot carrying a part that can be disassembled according to process j computes a uniformly distributed random number $R \in [0,1]$ at each timestep Δt (32 ms in Webots) and disassembles the part if $R < p_j^- \Delta t$. A robot about to begin assembly process j computes R and executes the assembly if $R < p_j^+ \Delta t$.

Fig. 5 shows the evolution of final product fractions in the micro- and macro-continuous models for the different sets of k_j^+, k_j^-. For each α, the models that use optimized k_j^+, k_j^- converge faster to the target $F1$, $F2$ fractions than the models that use random k_j^+, k_j^-. This demonstrates that the simple ODE model can indeed be used to compute control policies that improve the yield rate when used in a realistic system model. The Monte Carlo k_j^+, k_j^- lead to faster production than those from the linear program; however, on a standard 2 GHz laptop, the Monte Carlo optimization takes about 10 hours to converge to a solution, whereas the linear program computes k_j^+, k_j^- in less than a second.

V. Conclusions

We have presented a top-down methodology to derive decentralized, stochastic control policies for a swarm of robots to quickly manufacture target quantities of different products. Possible future work includes optimizing the assembly plan and using inter-robot communication to introduce a positive feedback mechanism that increases the yield rate.

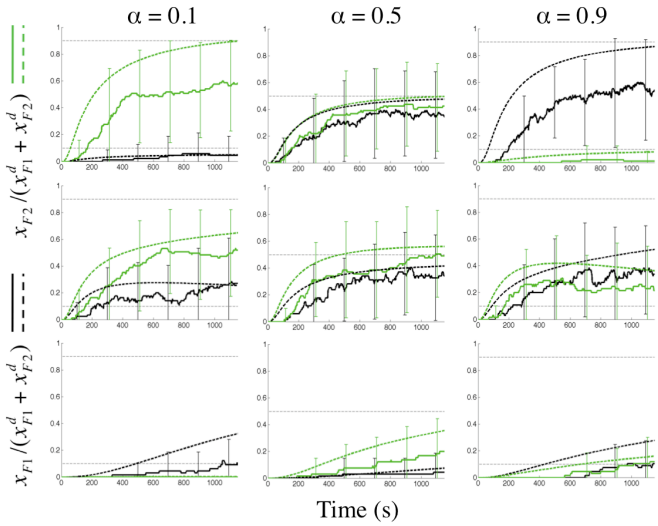

Fig. 5. $F1$, $F2$ fractions in the micro-continuous (solid lines) and macro-continuous (dashed lines) models using k_j^+, k_j^- optimized by the Monte Carlo method (top row) or linear program (center row) or randomly selected (bottom row). Micro-continuous models are each averaged over 30 runs and have 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) with $\mathbf{x^0} = [3\ 6\ 3\ 3\ \mathbf{0}]^T$.

VI. Acknowledgments

This work was supported by NSF grants IIS-0413138 and CSR-CPS 0720801, ARO grant W911NF-05-1-0219, ONR MURI grant N00014-08-1-0696, and ARL grant W911NF-08-2-0004.

References

[1] R. Baskaran, J. H. Hoo, B. Cheng, and K. Bohringer, "Catalyst enhanced micro scale batch assembly," in *Micro Electro Mechanical Systems (MEMS)*, 2008, pp. 1069–1072.

[2] K. Hosokawa, I. Shimoyama, and H. Miura, "Dynamics of self-assembling systems: Analogy with chemical kinetics," *Artificial Life*, vol. 1, no. 4, pp. 413–427, 1994.

[3] E. Klavins, S. Burden, and N. Napp, "Optimal rules for programmed stochastic self-assembly," *Robotics: Science and Systems II*, pp. 9–16, 2007.

[4] 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.

[5] 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.

[6] K. Lerman, C. V. Jones, A. Galstyan, and M. J. Mataric, "Analysis of dynamic task allocation in multi-robot systems," *Int'l. J. Robotics Research*, vol. 25, no. 4, pp. 225–242, 2006.

[7] 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.

[8] L. Matthey, S. Berman, and V. Kumar, "Stochastic strategies for a swarm robotic assembly system," in *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, 2009.

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

[10] O. Michel, "Webots: Professional mobile robot simulation," *Int'l. J. Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[11] 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.

[12] H. Li, Y. Cao, and L. Petzold, "StochKit: A stochastic simulation toolkit," *http://cs.ucsb.edu*, Dept. of Computer Science, University of California, Santa Barbara.

[13] D. Siegel and D. MacLean, "Global stability of complex balanced mechanisms," *J. Mathematical Chemistry*, vol. 27, pp. 89–110, 2000.

[14] R. Heinrich and S. Schuster, *The Regulation of Cellular Systems*. New York, NY: Chapman & Hall, 1996.