Principled synthesis for large-scale systems: task sequencing

Dylan A. Shell and Maja J Matarić

University of Southern California shell|mataric@usc.edu

Summary. This paper describes ongoing work toward a principled controller synthesis methodology for large-scale, minimalist multi-robot systems. The work's key objectives is to establish a set of programming primitives (processes) for which macroscopic behavior can be formally predicted. Such prediction is made possible by statistical physics techniques that use properties of time-invariant processes while exploiting the system's large size. This paper's focus is on the use of numerical and simulation methods during construction of the primitive process set. A computational method, developed by physicists, is used as a high-level simulation to characterize individual process behavior. The output, when interpreted qualitatively, guides distributed system design. In order to validate the approach, we consider a sequential inspection domain with a swarm of $400+$ simulated robots. Synchronization is achieved through processes analyzed with the methods described, and predictions are compared with behavior exhibited in a traditional multi-robot simulation. The two simulation tools play different roles in characterizing collective behavior, and these differences shed new light on the multi-robot controller synthesis problem.

1 Introduction

Robot swarms consist of many simple, small, and cheap units that exploit synergistic interactions to perform tasks and achieve robustness through massive redundancy. We consider the *synthesis* problem which requires derivation of local rules (robot control-laws and communication protocols) to enable performance of a pre-specified task. A method based on *composition of elementary processes* is proposed that yields controllers for large-scale systems. Most current methods are task specific, and finding a general solution to the problem remains a major challenge. This paper describes the use of simulation at different levels of detail to help mitigate this issue.

Researchers have begun seeking principled methods for design of minimalist systems. One approach is *analysis* of existing implementations (e.g., [5, 6]). Iterative improvements are offered for subsequent implementations and the overall-design process. Other work has focused on compilation of task-oriented rules [4], or automating the distribution of sensory information [7], to provide an *automated synthesis* methodology. Such approaches must be restricted to particular tasks or automate specific capabilities since the general problem is formidable. Analysis methods have been extrapolated to large systems (e.g., [6, pp. 12]) but formal synthesis methods have, thus far, only considered small groups.

The proposed synthesis method not only scales to hundreds of robots, but actually exploits large system size. Although not an automated procedure, the method makes use of predictive tools for guiding design. These tools produce coarse descriptions of behavior for a particular class of processes. Controller design then requires processes to be combined so as to achieve task-oriented behavior at the collective level. Many theoretical questions arise when considering synthesis based on composition. Most important is the robustness of predictions of the constituent processes, because models must–of necessity–ignore some details. Together with a description of the synthesis method itself (§2) and descriptions of the simulation techniques used (§4), this paper addresses the robustness issue in experiments with a simulated swarm applied to a sequential inspection task (§3).

Our broader research deals with two key questions: 1) How feasible (and tractable) is prediction of constituent processes as a guide for system design? 2) Given that we concentrate on a special class of computational processes, what are the computational capabilities of this class? We focus on the first question in this paper, specifically showing that the tools described are applicable to multi-robot system design for simple tasks. The second question is touched on empirically. Non-trivial capabilities are realizable with our method, as shown by considering synchronization, a canonical problem for groups of loosely-coupled asynchronous agents.

2 Analysis of individual processes

The collective behavior of a multi-robot system is difficult to predict since, in addition traditional distributed computing issues, robots have noisy sensors, imperfect actuators, and physical dynamics that constrain actions. Despite the many sources of complexity, formal synthesis and analysis methods are necessary. An important question to ask is: what information is needed *from* prediction? The level of detail required for a formal model is critically affected by the answer to this question. We accept sparse, qualitative predictions of collective behavior that ignore many low-level details. Most important from our perspective is the distinction between individual robot (*microscopic*) and group (*macroscopic*) descriptions. Explicitly modelling the system at these two levels becomes increasingly important as ever larger numbers of robots are considered.

Statistical mechanics is concerned with the derivation of bulk material properties from molecular models. The theory typically considers systems with infinitely many constituents. Equilibrium methods also include other assumptions (e.g., slow changes). Simulation tools are necessary because, even with these assumptions, few models have analytical solutions. This paper is part of an ongoing research agenda to explore the limits of such assumptions, range of applicability, and appropriate generalizations for multi-robot (and multi-agent) systems. These methods require that a non-traditional view of distributed computation be considered.

A *process* describes a time-extended series of actions or events. The processes we consider result from the execution simple, local, computational rules. Each process is described by listing a range of possible states, and a (possibly non-deterministic) transition function on those states. Execution of a process is the repeated application of the transition function. In this paper, states are values of internal variables on each robot. Only homogeneous systems are considered. At some frequency the transition function maps from an existing state to a new state, based on the process state and the state of others within the communication distance. The sequence of state values generated by each robot is the *microstate evolution*; it gives the microscopic details for a process. A lower-level controller for obstacle avoidance, smoothing sensor readings, etc., has access to the robot's state variables. We are only concerned with the higher-level coordination and cooperative aspects for our processes.

We restrict the processes (and hence transition functions) to those that are *ergodic*. This implies the existence of a time-invariant probability measure on the state-space [8], since the process dynamics have a weak temporal structure. Operationally, this property allows for the time average of some quantity to be calculated by averaging over the state space.

To synthesize controllers for new problems, we envision a library of ergodic processes, each with a macroscopic description. This description is the mean of a (process dependant) characteristic function calculated over all possible system states; it need only be calculated once for each process. Multiple processes can be combined so that, provided time-scales are chosen appropriately, the resulting behavior can be inferred from the macroscopic descriptions of the constituent processes. Function values are simply calculated over the product of the two constituent state spaces.

Ergodicity is valuable for synthesis because well-defined parts of the controller can be independently considered and each can have the ergodic property *by construction*. Using the property for system analysis requires that the overall system behavior be ergodic. Such a claim is difficult to make. (Analysis that assumes the plausibility of a "stochastic series of events" interpretation, as in [6], is far less restrictive.) Existing work that exploits ergodic dynamics does so only for part of the overall system. For example, Jones and Matarić [4] consider controllers which perform an ergodic exploration of the environment, and implicitly use this fact in predicting system performance. White et al. [9] make an assumption about the nature of the environmental dynamics that amounts to an assumption of the ergodic property. No work has explicitly recognized the connection with ergodic theory.

Without inherent temporal structure, ergodic processes appear inadequate for robot controller design. Restricting design to ergodic processes certainly represents a significant shift in perspective. Typically, programming (of computers or robots) involves a decomposition of the task specification into a sequence of steps. The validation task domain explored in this paper is a sequential inspection task, and the presented controller shows that ergodicity can impose temporal ordering. The processes achieve this at a strictly macroscopic level.

3 Sequential inspection task

Given a large bounded environment with multiple sites of interest, consider the problem of having a robot swarm (in its entirety) visit these sites in a predetermined sequence. This situation arises when a mobile robot system supplements a network of static sensor nodes in order to provide higher-resolution sampling when some phenomenon is sensed or anticipated. We consider the case in which the entire swarm is tasked as a unit to perform the inspection of sites of interest in a specified order.

Decision making is non-trivial without a centralized leader. Synchronization of state across robots is necessary for sequential inspection because the group must collectively decide that the next site should be visited. This form of synchronization was first defined by Jennings and Kirkwood-Watts [3] in a multi-robot context. We consider individual robots that are extremely limited and show that two simple ergodic processes are sufficient for this cooperative decision making.

3.1 Process definition

The simulated robots have limited capabilities with noisy sensing and unreliable local communication. The sequential inspection controller consists of a low-level controller coupled to two ergodic processes. The processes are responsible for the decision of which site of interest to visit. The processes execute in a distributed fashion, sharing information with neighboring robots through a local-broadcast communication network (further details in §4.2).

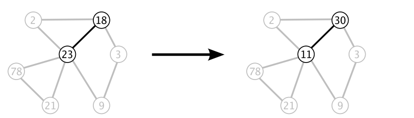

Fig. 1. Vertices represent robots and edges communication links. Numbers depict Process 1 state. A random value (12 here) is exchanged. No state information in grey is necessary, it is a strictly local interaction.

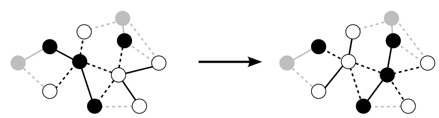

Fig. 2. A Process 2 transition. Filled vertices represent state +1, empty ones -1; solid edges depict alignment, broken edges misalignment. The number of solid (4) and broken (5) edges is conserved.

Process 1

The state of Process 1 (on each robot) is described by a positive integer. Force the following constraint to hold: the sum of the state values over all robots must equal $\mathcal{K}_1$. Then the state-space for the entire swarm consists of every possible permutation of $\mathcal{K}_1$ over the n robots. Although the summation constraint describes a global property, it is locally enforceable. Robot x in state x_i can transition to state x_{i+1} and maintain constraint by reaching an agreement with another robot, say y. Robot y must to transition from y_i to y_{i+1} so that $y_{i+1} + x_{i+1} = x_i + y_i$.

The dynamics function makes a transition from state x_i to x_{i+1} by setting apart a random portion of x_i and transmitting it to a randomly selected robot within communication range. The value of x_{i+1} is obtained by adding any portions received from neighbors. (The portion is only removed by the sender with confirmation of receipt.) Thus, the sum of state values remains $\mathcal{K}_1$. Fig. 1 shows the result of this transition.

Process 2

The second process is based on the ferromagnetic Ising model[1]. Each robot's process can be in one of two states: $\{-1, 1\}$. Two values on neighboring robots are *aligned* if they have the same state values, and *misaligned* otherwise. The total number of aligned and misaligned edges are written as N_{aligned} and $N_{\text{misaligned}}$ respectively. We write $\mathcal{K}_2 = -N_{\text{aligned}} + N_{\text{misaligned}}$.

The transition rules conserve $\mathcal{K}_2$. Two neighboring robots can calculate the effect of flipping local state (i.e., changing to the other state) by examining their immediate neighbors. This constraint can also be locally enforced. As before, the decision as to which neighbor, and whether to flip or not, is randomly selected. Process 1 is clear in terms of vertex states, Process 2 is more intuitive as operating on edges. See Fig. 2.

3.2 Process analysis

Next, consider the behavior of the processes over a range of $\mathcal{K}_1$ and $\mathcal{K}_2$ values.

Process 1

The transition function for Process 1 is a random walk of K_1 units on the robot communication graph. If robots maintain loose connectivity (i.e., disconnected subgraphs rejoin periodically) then symmetry suggest equal probabilities over each of the robots. This allows the mean and variance of expected states to be calculated. Both are trivial, with analytical solutions for any values of K_1 and n. Fig. 3 and 4 show a comparison between these theoretical predictions and data recorded from simulation runs.

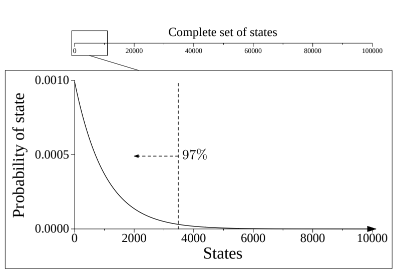

Fig. 3. The density function describing the probability of Process 1 being in a particular state at a random time. The plot of the full domain along the top and the broken line that delineates 97% of the probability mass, showing sharp peak in the distribution.

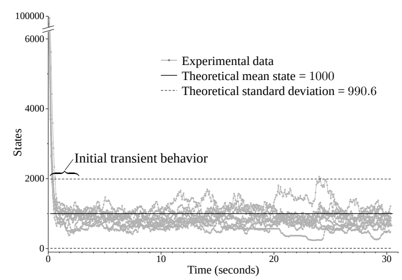

Fig. 4. Plots from ten experimental runs showing the state of a single robot ($n = 100$ and $\mathcal{K}_1 = 100000$) over time. In all cases, the plotted robot moved quickly into states well characterized by the theoretical mean and standard deviation.

Process 2

The symmetry in Process 2 is far less obvious than in the previous case. The implementation does not bias one configuration over another. Given enough time, we can expect any two configurations with the equal $\mathcal{K}_2$ to be equally likely if we assume that average network connectivity is stable (or slow changing compared with execution of Process 2 itself).

We simplify the problem by considering a model of robots placed on a 21×21 square lattice with each of the 441 robots placed on the grid and connected to nearest neighbours. To calculate the probability of an ergodic process occurring in a particular state one must construct a measure over all possible states. For n robots this means integrating over the 2^n states. Process 2's transition function was simulated using MMMC (see $\S 4.1$ for details). The mean state value M is calculated for each configuration and must lie on $[-1, 1]$. The two extremal values are given for ordered states, those midway ($M \sim 0$) are disordered. The result is a characterization of the number of given states for a given $\mathcal{K}_2$. The logarithm of this number gives S the

Boltzmann entropy of the configuration (in natural units). Fig. 5 shows a construction of the entropy surface for different values of M and E. The E axis is $\mathcal{K}_2$ normalized to fall between -1 for minimum alignment and $+1$ for complete alignment. The figure shows the log conditional probability (S) of an M state given an E value.

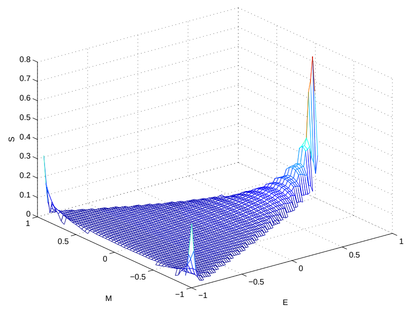

Fig. 5. The entropy S as a function of M for a given E. Since S is a logarithm of the number of states, the function is sharply peaked. This implies a high degree of certainty of the process state for extremum values of E.

For values of $E \sim +1$ there is one clear peak: the system can be expected to exhibit behavior with $M \to 0$, or an equal number of -1 and $+1$ spin values. When $E \sim -1$ the system occurs in one of two states, either many $+1$ or many -1 values. If the system is moved from a state of high E to toward low E, then it will exhibit a spontaneous symmetry-breaking phase-transition.

Coupling the two processes

Thus far, only considered constant $\mathcal{K}_1$ and $\mathcal{K}_2$ conditions have been considered. Now we couple the processes by allowing Process 1 to slowly increase (decrease) $\mathcal{K}_1$ provided that $\mathcal{K}_2$ decreases (increases) by the same amount. This is the same type of conservation used in defining the processes, but now applied to process parameters. Such a coupling can be modelled as a transition function operating on process parameters while maintaining $\mathcal{K}_1 + \mathcal{K}_2 = \mathcal{C}$. It is essentially a composite process with a combined state-space. We term this a *macroscopic degree-of-freedom*.

The entropy (and hence probability distribution) of the composite process can be calculated by considering the entropy function of each individual process. The exact method depends on the relative "size" of the entropy function, but a full discussion is beyond the scope of this paper. For Process 1 we can see that the first process results in a mean value of $\frac{\mathcal{K}_1}{n}$ per robot. If we permit free exchange of one unit of $\mathcal{K}_1$ for one unit of $\mathcal{K}_2$, then a decrease in $\mathcal{K}_1$ results in an increase in $\mathcal{K}_2$. With appropriate values, the decrease in $\mathcal{K}_1$ can result in a rapid transition from $M \sim 0$ to $M = -1$ or $M = +1$. This transition is the result of the average behavior of the system. We describe how this is used for the sequential inspection task below.

4 Simulation tools

Trade-offs in run-time, space, and fidelity require the careful choice of simulation software. We used two software tools, one to characterize a process's behavior, the other to simulate the robots executing the appropriate composite controllers. The first simulates pseudo-dynamics during controller construction, while the second is a traditional sensor-based simulator used for testing and validation.

4.1 Microcanonical Metropolis Monte-Carlo (MMMC)

Microcanonical Metropolis Monte-Carlo (MMMC) is a numerical method described by Gross [2] as a variation of the well-known Metropolis-Hastings algorithm. The both methods approximate a probability distribution by drawing a sequence of random samples. The numerical methods are particularly useful when explicitly interested in the thermodynamics of finite systems. The need for MMMC arose in nuclear physics because distributions over unconventional thermodynamic variables (conserved extensive variables) were considered. The resulting distributions are extremely peaked, with values ranging many orders of magnitude. Naïve sampling fails to obtain sufficient data for the lower probability states because they arise so infrequently. The MMMC algorithm decomposes the distribution surface into local patches called *knots*. Sampling then proceeds locally for each knot. Partial derivatives of the surface is calculated at each knot, and the function is then constructed through integration of those values.

Values at each knot are calculated by collecting statistics by a pseudo-dynamics simulation of the model from given initial value within the knot. The emphasis is on collecting numerical data based transitions within the knot itself. The simulation does not attempt to construct long temporal explorations of the states like a typical simulation would do. We believe that this approach, rather than simulation for plausible runs, is important for characterising the processes. From the perspective of the entropy surface, such an approach allows far more information to captured, which is necessary of a complete characterization of process behavior.

We used MMMC to construct the entropy surface for a finite Ising model. Even with 100 robots it is infeasible to explore all available states so a randomized algorithm must be used. Fig. 5 shows the logarithm of the probability normalized for each E value as calculated by MMMC. This function shows the extreme range of probabilities that can arise.

4.2 Microscopic simulation

Our sensor-based microscopic simulation uses an environment model in order to produce artificial readings for virtual sensors. We use an efficient Delaunay triangulation data-structure for representing positions of the robots as well as obstacles within the environment. Sensor readings are generated from this data-structure rather than ray-casting in a bitmapped rendering. Robots are updated asynchronously.

The experiments used simulated robots with a velocity control interface. Linear and angular velocities were corrupted by three noise terms: multiplicative, additive and additive biased. The first two were drawn from normal distributions at each timestep, the last term represented a systematic bias and was drawn once at initialization. These values were different for each robot. The three terms for linear velocity were drawn from $\mathrm{N}(0, 0.01^2)$, $\mathrm{N}(0, 0.03^2)$, and $\mathrm{N}(0, 0.002^2)$; similarly for angular velocities $\mathrm{N}(0, 0.01^2)$, $\mathrm{N}(0, 2.5^2)$, $\mathrm{N}(0, 0.15^2)$; units are meters and degrees respectively.

The robots used three sensors: 1) a distance sensor with 12 radial rays, each with a range of 0.5 meters, with multiplicative $\mathrm{N}(0, 0.02^2)$, additive $\mathrm{N}(0, 0.05^2)$, and a additive bias for each ray $\mathrm{N}(0, 0.05^2)$; 2) a compass with four-bits of information, and added noise $\mathrm{N}(0, 15.0^2)$ and added bias $\mathrm{N}(0, 2.0^2)$; 3) a single-bit sensor that responded to the sites of interest, that returned false-negatives with probability 0.15 and false-positives with probability 0.08. Only the compass provided global information. Distance readings returned from other robots and obstacles were indistinguishable.

The simulator included a model of a local-broadcast communications. Each robot could send messages that may be heard within a disk of radius 2m centered at the sender. Messages arrived at a robot distance d from the sender with a probability given by $0.02d^2 - 0.18d + 1.0$. We also provided a local point-to-point network: robots could provide an intended recipient. These packets were locally broadcast within the same 2m disk, but automatically discarded by non-matching recipients. The sender was notified of a successful transmission.

4.3 Comparison of methods

Each run of the microscopic simulation produces a single temporal evolution of the robot system. It allows for comparison of task performance over time, with each independent execution offering new evidence. If the simulation is sufficiently realistic, it offers a forecast of performance with physical robots, and is thus important for validation purposes.

In contrast, MMMC doesn't simulate a single plausible trajectory. For each knot it will simulate a brief temporal sequence and reinitialize to a value within the knot's parameter values. Also, that brief temporal sequence may fail to follow the same strict dynamics rules as the robot system might. In the case of the Ising model, random flips are permitted, even if they're not neighboring. The key aspect is that the method provides a high-level characterization of the statistical aspects of the system.

Microscopic simulation is often used for iterative system design: insights from simulation (often supplemented by analysis) provide incremental improvements. Such methods operate in an end-to-end manner. The probabilistic characterization, like MMMC, provides a complementary approach.

5 Results

Experiments were performed in a $50\text{m}\times 50\text{m}$ arena with two disks of radius 3m placed at the North-East (NE) and South-West (SW) corners, 2m from the sides. The disks represent the sites of interest and can be sensed by robots that are positioned over them. Without localization information and equipped with a noisy 4-bit compass, many robots reach the arena corner having missed or failed to sense the disk. Independent sensing by the many robots within the swarm lessens the effects of the position and sensor uncertainty. Robots were initially placed in the arena center and were tasked with visiting first the NE site, then the SW site. The critical issue is in synchronization of the decision to advance from one site to the next.

Synchronization is achieved by coupling Processes 1 and 2 together as described above. On each robot the low-level controller (for obstacle avoidance, navigation, sensor processing, etc.) uses a variable to tracks task-state. The value of this variable affects the interpretation of the compass readings and hence steering. The low-level controller is coupled to the two synchronization processes in two ways:

1. Input through gradual perturbation of the Process 1's state space. A robot detecting that it is over a site, will increase the value of it's Process 1 state (by 200 units). Observation of thrashing or lack of progress (by measuring odometry movement of less that 0.25m in 5s) results in the Process 1's state being reset to zero.
2. Output produced by monitoring average values of Process 2's the slow changing state variables. Values are averaged over a 12 second window. When the value is within a threshold of zero (we used 0.02) a flag was set indicating that the task-state variable would soon change. When the mean approaches either -1 or 1 (we used 0.98), the task-state variable is altered to show the beginning of the next task (on -1) or return to the previous task (on 1).

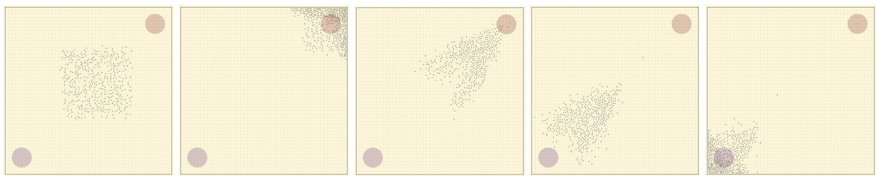

Fig. 6. The screen-shots of a run with 441 robots running for ~ 520 seconds.

Fig. 6 shows simulation snap-shots of robots moving from the arena center to the first and second inspection sites. Fig. 7 gives connectivity information for the same run. The increasing density at each inspection location had a marked effect on the number of neighbors each robot had. Fig. 8 gives a plot of $\mathcal{K}_1$ and the average M value for three experimental runs. The figure shows Process 2's state being switched from -1 to $+1$ throughout the system. These three runs show similar behavior because the symmetry was broken in identical ways for each case. In 10 total runs, 4 cases transitioned back to -1 and the robots stayed at the NE site. This is expected as symmetry breaking occurred in an unbiased fashion; adding a bias would stop such cases from having to undergo the phase transition multiple times in order to reach the decision for exploration of the next site.

6 Discussion and Conclusion

The experiments show that the gross simplifications made in modelling the individual processes (especially Process 2) do not invalidate the general features of the processes behavior. For example, the mean connectivity is shown to be significantly higher than the value of 4 used in MMMC simulations. Also, the linearization arguments based on global constraints (e.g., in assuming fixed $\mathcal{K}_1$), statements about "slow" couplings, and ignorance of sparse node or network failures, are all only plausible to a limited extent. The experiments suggest that such assumptions are valid provided only very coarse features of the process behaviors are considered.

The predictions of process behavior, even with major modelling simplifications, can be useful provided the designer seeks a qualitative understanding of the behavior. We believe the model of the synchronisation processes presented in this paper is robust because it was based on the topological properties of the entropy surface. Our ongoing work is considering other processes that result in particular properties of the entropy surface.

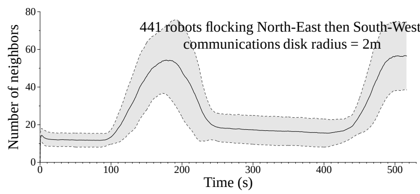

Fig. 7. The number of neighbouring robots within communication range for 441 robots. The average over all robots plus/minus one standard deviation.

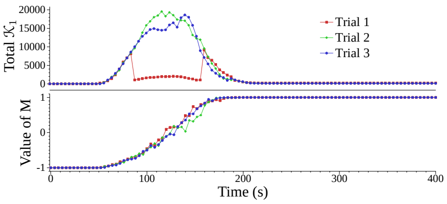

Fig. 8. Plot of synchronisation processes internal state for the first 400 seconds of a run with 441 robots. The transition of M values is clearly visible.

Fig. 7 suggests that a smaller communication disk may suffice for the sequential inspection in the environment we considered. An alternative approach (perhaps without ergodic processes) could use the number of robots within communication range in order to switch behavior directly. An interesting question is whether such a switch would occur as abruptly as our approach.

In conclusion, we have demonstrated that the composition of ergodic processes is a feasible approach for tackling the synthesis problem, at least in the case of simple tasks. Synchronization, as a basis for sequencing, can be achieved by such processes. Composition of processes is useful because each can be independently analyzed in a manner that remains valid for combinations of ergodic processes. The analysis attempts to understand the processes' macroscopic behavior by considering the entropy surface. Numerical tools are useful for expanding the set of processes which are amenable to analysis, and, in particular, for dealing with issues that arise within finite systems. We view analysis of composable processes as complementary to traditional methods that iterate controller design at the complete system level. This approach appears relatively robust to modelling errors, provided the description is used for qualitative understanding of system behavior.

Acknowledgements: This work was partially funded by National Science Foundation, Grant No. IIS-0413321. The authors thank Emily Mower her detailed comments on an early draft.

References

1. M. E. Fisher. The theory of equilibrium critical phenomena. *Reports on Progress in Physics*, 30:615–730, 1967.
2. D.H.E. Gross. Microcanonical thermodynamics and statistical fragmentation of dissipative systems – the topological structure of the n-body phase space. *Physics Reports*, 279:119–202, 1997.
3. J. Jennings and C. Kirkwood-Watts. Distributed mobile robotics by the method of dynamic teams. In *DARS*, pages 46–56, Karlsruhe, Germany, May 1998.
4. C. V. Jones and M. J. Matarić. Automatic Synthesis of Communication-Based Coordinated Multi-Robot Systems. In *IEEE/RSJ IROS*, pages 381–387, Sendai, Japan, September 2004.

5. K. Lerman and A. Galstyan. Mathematical Model of Foraging in a Group of Robots: Effect of Interference. *Autonomous Robots*, 13(2):127–141, 2002.
6. A. Martinoli, K. Easton, and W. Agassounon. Modeling of Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation. *IJRR*, 23(4):415–436, 2004.
7. L. E. Parker, M. Chandra, and F. Tang. Enabling Autonomous Sensor-Sharing for Tightly-Coupled Cooperative Tasks. In *Proceedings of the NRL Workshop on Multi-Robot Systems*, pages 119–230, Washington, DC, USA, March 2005.
8. K. Petersen. *Ergodic Theory*. Cambridge University Press, Cambridge, England, 1983.
9. P. White, V. Zykov, J. Bongard, and H. Lipson. Three dimensional stochastic reconfiguration of modular robots. In *Proceedings of Robotics: Science and Systems*, pages 161–168, Cambridge, MA, USA, June 2005.