2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Parameter Estimation and Optimal Control of Swarm-Robotic Systems: A Case Study in Distributed Task Allocation

Nikolaus Correll

Abstract— **This paper presents a methodology for finding optimal control parameters as well as optimal system parameters for robot swarm controllers using probabilistic, population dynamic models. With distributed task allocation as a case study, we show how optimal control parameters leading to a desired steady-state task distribution for two fully-distributed algorithms can be found even if the parameters of the system are unknown. First, a reactive algorithm in which robots change states independently from each other and which leads to a linear macroscopic model describing the dynamics of the system is considered. Second, a threshold-based algorithm where robots change states based on the number of other robots in this state and which leads to a non-linear model is investigated. Whereas analytical results can be obtained for the linear system, the optimization of the non-linear controller is performed numerically. Finally, we show using stochastic simulations that whereas the presented methodology and models work best if the swarm size is large, useful results can already be obtained for team-sizes below a hundred robots. The methodology presented can be applied to scenarios involving the control of large numbers of entities with limited computational and communication abilities as well as a tight energy budget, such as swarms of robots from the centimeter to nanometer range or sensor networks.**

I. Introduction

Reactive, fully distributed coordination based on local interactions is a reasonable control scheme for multi-robot systems when the individual robotic units are limited in terms of computation and communication. Multi-robot systems obeying these control paradigm are commonly referred to as swarm-robotic systems [1]. Swarm-robotic systems provide a high level of robustness due to individual simplicity and because no single point of failure given by a centralized entity exists. Also, as coordination does not rely on global communication, swarm-robotic controllers scale well for large numbers of robots. Due to a high level of randomness (either implemented in the robots' controllers or coming "naturally" with sensor and actuator noise), the resulting state distribution of the robots is usually probabilistic and robust to changes in the number of robots.

A drawback of this coordination approach is that although the system will eventually show the desired behavior, it cannot be excluded that the system exhibits sub-optimal or even undesired behavior, in particular when the number of robots is low. A partial remedy to this problem is the development of powerful analysis tools that allow for better prediction of the systems performance; better predictability of the swarm performance can then be used to further improve the design of the controller and the system as a whole.

One way of analyzing swarm-robotic systems is to use population dynamic models, i.e. probabilistic rate equations that keep track of the number of individuals in a certain state [2]–[4]. Provided that the number of interactions among the robots is large, such models track well the *average* behavior of the system already for small teams. We wish to find a methodology to automatically identify the parameters of such models from experimental data and consequently use the refined models to find optimal control parameters for the individual robotic unit.

One of the advantages of population dynamic models is that control parameters of the individual robots are directly expressed in the macroscopic equations describing the system's dynamics. This property allows us to use these models as a design tool and calculate an optimal control to achieve a desired trajectory of the robot swarm in state space. For instance in [3], Martinoli et al. calculate an optimal control parameter that maximizes the collaboration rate in the "stick-pulling experiment". In [5], Correll et al. introduce dynamic optimization for finding an optimal dynamic collaboration policy for a swarm-robotic inspection case study. In [6], Milutinovic et al. use optimal control on partial differential equations describing the probability density function of the swarm's spatial location in order to drive the swarm into a desired area in the environment. Finally, Berman et al. consider swarms that switch between a series of controllers and proposes a methodology for finding optimal trajectories in state-space that is sub-divided into poly-topes each associated with a particular controller in [7].

As a model is only an approximation of a real system's dynamics, good quantitative agreement with model prediction and real robot experiments can usually only be achieved by calibrating some of the model parameters using real hardware and/or estimating parameters by model fitting [8]. In this paper, we propose a methodology to find both an optimal control as well as optimal system parameters at the same time. The process is illustrated using two distributed algorithms for the multi-robot task allocation problem, whose dynamics are described by a linear and a non-linear macroscopic model, respectively. Applications of the presented methodology are large-scale distributed systems consisting of entities with limited computational and communication capabilities as well as a tight energy budget such as miniature robots around and below the centimeter range or sensor networks.

N. Correll is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, USA. This work has been sponsored by SNSF fellowship PBEL2–118737. nikolaus@csail.mit.edu

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

A. Case Study: Dynamic Role Allocation in Robot Swarms

The multi-robot task allocation [9] or role assignment problem is well illustrated by applications such as search and rescue missions [10] or foraging [11]. In a search and rescue mission, some robots are allocated for exploring the environment, some robots remain static in the environment for message passing, and some others are used as "markers" around points of interests [12]. In a foraging application, a task corresponds to foraging for a specific class of items requiring a specific tool, e.g., and the number of each robots performing a task is a function of the estimated number of items in each class.

In both scenarios, a reactive, fully distributed approach provides a robust and scalable solution, and is particularly competitive under the absence of global communication but sub-optimal when compared with deliberative, deterministic coordination approaches using global communication (see [11]–[13] for an analytical and experimental comparison of a series of task assignment algorithms in a robot swarm).

For illustrating the optimal control and parameter estimation framework, we consider first a non-collaborative reactive approach where robots switch to a certain task with a specific probability. It is then shown analytically, how optimal control parameters for achieving a certain steady-state task distribution can be found, even if the times required for task execution are unknown. Second, a threshold-based algorithm [14], [15] is considered, where the probability to switch to a task is a function of the number of robots already in this task.

II. Optimal Control on Population Dynamic Models

Consider $N(k+1) = f(N(k), \mu, \theta)$ describing the evolution of the number of robots in a certain state at time step k with the vector μ containing the *control parameters* of the individual robot, i.e. parameters that are hard-coded into the robot controller, and the vector θ contains the *system parameters* which are usually average quantities describing the environment, e.g. the average duration of a task, or the average encountering probability of a certain object [3].

With the system parameters θ being known, optimal control parameters μ^* for driving the system into a desired steady-state distribution π can be found by solving the optimization problem

$$\mu^* = \arg\min_{\mu} (\pi - N^*(\mu, \theta))^2 \tag{1}$$

where $N^*(\mu, \theta) = \lim_{k \to \infty} f(N(k), \mu, \theta)$.

Solutions to (1) can be obtained by applying the *necessary condition of optimality* (NCO), that is

$$\frac{\partial}{\partial \mu} (\pi - N(\mu, \theta)^*)^2 = 0 \tag{2}$$

where the derivative $\frac{\partial}{\partial \mu}$ is either calculated analytically (when possible) or by exciting an implementation of the system by a small perturbation of μ by slightly changing one parameter at a time [5].

A. Optimal Control under Uncertainty of System Parameters

In a real robotic system, not all of the model parameters can be known beforehand, either due to uncertainty of measurements or because the chosen model oversimplifies the system (e.g., ignoring friction or sensor noise). In this case, parameters need to be estimated from observations of the system $\hat{N}^*(\mu, \theta)$, leading to the following optimization problem [8]:

$$\theta^* = \arg\min_{\overline{\theta}} (N^*(\mu, \overline{\theta}) - \hat{N}^*(\mu, \theta))^2 \tag{3}$$

which aims at minimizing the error between model prediction $N^*(\mu, \overline{\theta})$ and observation $\hat{N}^*(\mu, \theta)$ by finding an optimal set of parameters θ^*. As for the optimal control problem in Section II, analytical solutions for (3) can be obtained by applying the NCO.

As either μ or θ is known (in a real experiment or in the model used for optimization, respectively), finding an optimal control and parameter estimation are indeed two separate optimization problems. An optimal control μ^* calculated using (1) based on an estimate $\overline{\theta}$ can then be used in an implementation at a lower level of abstraction, e.g. realistic, sensor-based simulation (e.g., Webots [16]) or real robot experimentation, which provide an observation $\hat{N}(\mu^*, \theta)$. This observation can then be used for solving the parameter estimation problem (3) and improving the optimal control by using θ^* instead of $\overline{\theta}$.

This insight allows for formulating the following recurrence equations for the parameter vectors $\mu(n)$ and $\theta(n)$. The index n corresponds to the number of the experiment that led to an observation of the steady state $\hat{N}(\mu, \theta, n)^*$.

$$\overline{\theta}(n+1) = \arg\min_{\overline{\theta}} \sum_{i=0}^{n} (N^*(\mu(i), \overline{\theta}) - \hat{N}^*(\mu(i), \theta, i))^2 \tag{4}$$

$$\mu(n+1) = \arg\min_{\mu} (\pi - N^*(\mu, \overline{\theta}(n+1)))^2 \tag{5}$$

with $\mu(0) = \mu_0$ an initial guess for the control parameters. Already one experiment ($n = 0$) conducted with an initial guess μ_0 then allows for estimating the real system parameters θ, if θ can be identified unambiguously. If this is not the case, i.e. there are multiple combinations of parameters possible that reproduce the observation (see [8]), multiple iterations of (4)-(5) might be necessary. In this case, the sum in (4) assures that $\overline{\theta}$ is in agreement with all conducted experiments.

III. A Probabilistic, Ergodic, Task Allocation Problem

Consider a robotic system in which each robot can engage in one out of m different tasks, as well as being idle. For a large system of robots, let the number of robots in either state be given by the state vector $N(k) = (N_0(k), N_1(k), \ldots, N_m(k))^T$, where $N_0(k)$ corresponds to the number of robots being idle. The index k corresponds to the number of time-steps of length T.

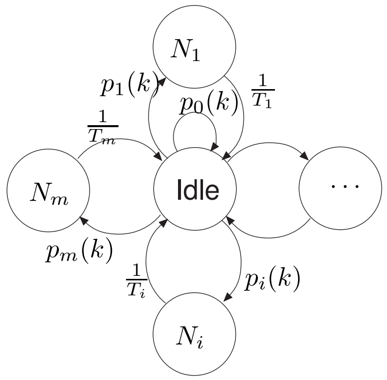

Fig. 1. Markov chain describing the individual robot system. A robot is *idle* or switches to task i with probability $p_i(k)$. For the linear controller, $p_i(k)$ is constant, whereas for the threshold-based controller $p_i(k)$ is a function of the number of robots in state i. Both times, a robot needs an average of T_i time-steps for executing task i and returning to *idle*.

Task allocation in the system is fully distributed, probabilistic, and does not require any sensors. When idle, at every time-step a robot will switch to a task i with probability p_i or will remain idle with probability p_0 such that $\sum_{i=0}^{m} p_i = 1$. It will then execute this task for an *average* time, T_i. Thus, when executing task i, a robot returns into idle with probability $\frac{1}{T_i}$. The Markov chain representing the individual robot is depicted in Fig. 1.

The *average* number of robots executing task i is then given by the following difference equation

$$N_i(k+1) = N_i(k) + p_i N_0(k) - \frac{1}{T_i} N_i(k) \tag{6}$$

The number of idle robots can be calculated from the fact that the number of robots is constant

$$N_0(k+1) = n - \sum_{i=1}^{m} N_i(k+1) \tag{7}$$

with $N(0) = (n, 0, \ldots, 0)^T$ the initial conditions.

Let M be the state transition probability matrix of the Markov chain depicted in Fig. 1, with its i-th row given by Equations 6 and 7. The system can then be summarized by

$$N(k+1) = M(\mu, \theta) N(k) \tag{8}$$

where $\mu = (p_1, \ldots, p_n)$ are the control parameters of the individual robot, and $\theta = (T_1, \ldots, T_n)$ the system parameters that are dependent on the average time a robot spends in a task in this example. As (8) is a linear system, this controller will be referred to as the *linear controller* in the reminder of this paper.

A. Optimal Control Problem

Assuming θ to be known, we want to find an optimal control μ^* such that $\lim_{k\to\infty} M(\mu, \theta)N(k) = N^* = \pi$, i.e. the steady-state role distribution matches a specific distribution π.

As the Markov chain of Fig. 1 is ergodic, a steady-state $N^*(\mu, \theta)$ exists, and $N^*(\mu, \theta) = \pi$ is the eigenvector of M with eigenvalue 1, i.e.

$$N^*(\mu, \theta) = M(\mu, \theta) N^*(\mu, \theta) \tag{9}$$

and solving

$$(M - I)\pi = 0 \tag{10}$$

with $\pi = (\pi_0, \ldots, \pi_m)^T$ leads to

$$\mu^* = \left(\frac{1}{T_1} \frac{\pi_1}{\pi_0}, \ldots, \frac{1}{T_n} \frac{\pi_m}{\pi_0} \right) \tag{11}$$

Alternatively, μ^* can be found by solving the following optimization problem

$$\mu^* = \arg\min_{\mu} (\pi - N^*(\mu, \theta))^2 \tag{12}$$

Using the necessary condition of optimality, we write

$$0 = \frac{\partial}{\partial \mu} (\pi - N^*(\mu, \theta))^2 = -2(\pi - N^*(\mu, \theta)) \frac{\partial N^*(\mu, \theta)}{\partial \mu} \tag{13}$$

which leads with $\frac{\partial N^*(\mu,\theta)}{\partial \mu} \neq 0$ to a similar Ansatz as (10). Solving for each component of N^* leads to

$$N_0^* = \frac{n}{1 + \sum_{i=1}^{m} p_i T_i} \tag{14}$$

and

$$N_i^* = T_i p_i N_0^*. \tag{15}$$

Substituting $N^*(p, T) = \pi$, (15) is equivalent to the solution obtained in (11).

B. Optimal Control under Uncertainty of System Parameters

We will now assume that the task-execution times $\theta = (T_1, \ldots, T_m)^T$ are unknown and need to be estimated along the optimization process. Using an initial guess for $\theta(0) = \overline{\theta}$ and known $\mu(0) = \mu_0$, a new estimate for T can be calculated using (4) and a new optimal control based on the improved estimate of $\overline{T}$ can be found using (5) which can both be solved using (11).

IV. A Threshold-Based Task Allocation Problem

Whereas the probability for executing a certain task was constant for the controller of Section III, we now assume that robots can estimate the fraction of robots performing a certain task and use this information for calculating the probability for executing this task. The fraction of robots performing a certain task can be estimated either by direct observation (e.g. using a camera), overhearing communications in the swarm, explicit local communication, or by observing changes in the environment that are a direct result of a certain task [13], [17]. This approach is known as a *threshold based* algorithm, and has been observed for a variety of social insects [14] and successfully been used for coordinating robot swarms in [13], [15], [17].

The threshold function is chosen such that the probability to execute a certain task decreases with the number of robots already performing this task, and is given by

$$p_x = \frac{\sigma_x}{\sigma_x + \tau_x} \tag{16}$$

where σ_x is the perceived stimulus for executing task x and τ_x is the threshold for a certain task. In this example, we use $\sigma_x(k) = \frac{1}{N_x(k)+1}$, i.e. the stimulus is anti-proportional to the perceived number of robots. As $\sigma_x(k)$ simply needs to be anti-proportional to $N_x(k)$ and the controller is stochastic, a local estimate of $N_x(k)$ that is proportional to the real value is sufficient. The time-varying probability for executing task x is thus given by

$$p_x(k) = \frac{1}{1 + \tau_x(1 + N_x(k))} \qquad (17)$$

As before, robots switch back to idle state after an average time given by $\theta = (T_1, \ldots, T_m)^T$. The system equations are given as follows

$$N_i(k+1) = N_i(k) + N_0(k)p_i(k) - \frac{1}{T_i}N_i(k) \qquad (18)$$

$$N_0(k+1) = n - \sum_{i=1}^{m} N_i(k+1) \qquad (19)$$

with $N(0) = (n, 0, \ldots, 0)^T$ the initial conditions, i.e. all robots being idle initially, and $\mu = (\tau_0, \ldots, \tau_m)$ the decision variable of the optimal control problem.

As the state transition probability matrix of the system is time-varying, analytical solutions to the optimization problems become unfeasible and the steady-state $N^*(k, \mu, \theta)$ is calculated numerically by iterating (18)-(19).

V. Results

Results are provided for a set of desired steady-state distributions and serve as an example for the proposed methodology. Notice that the accuracy of the numerical results is a function of the chosen optimization method (*fmincon* from the Matlab optimization toolbox), which has been sufficient for solving the problems considered in this paper. Simulations of the swarm are obtained by stochastic simulation of the underlying Markov chain, which has yielded accurate predictions when compared with real robot experiments in a series of case studies [3], [18].

A. Optimal control with known system parameters

We consider a system with three tasks with durations given by $\theta = \overline{\theta} = (2, 4, 2)^T$, i.e. the system parameters θ are known at time of optimization. We are interested in finding an optimal control for reaching the steady-state distributions π_1 and π_2 given by

$$\pi_1 = (40\%, 10\%, 20\%, 30\%)^T$$
$$\pi_2 = (40\%, 30\%, 20\%, 10\%)^T$$

which have been arbitrary chosen for demonstrational purposes (the set of theoretically feasible distributions is a function of θ, compare Equation 11).

Using (11) for calculating the optimal control for the linear controller, yields

$$\mu_1^* = (0.125, 0.125, 0.375, 0.375)^T$$
$$\mu_2^* = (0.375, 0.125, 0.125, 0.375)^T$$

For the non-linear controller, we solve the optimization problem from (12) for the system given by (18)–(19) numerically, where $N^*(\mu, \theta)$ is calculated by iterating (18)–(19) until a steady-state is reached, leading to

$$\mu_{1'}^* = (0.5025, 0.1107, 0.0831, 1.0000)^T$$
$$\mu_{2'}^* = (0.5073, 1.0000, 0.0833, 0.1104)^T$$

and a residual error in the order of 10^{-7} both times.

In order to study the transient behavior of the system, we performed a series of experiments where all robots switch to μ_2^* and $\mu_{2'}^*$ at $k = 26$, and use μ_1^* and $\mu_{1'}^*$ for the first 25 time-steps, for the linear and non-linear controllers, respectively. Fig. 2 shows simulation results of the macroscopic model and stochastic simulation for the linear (top row) and non-linear controller (bottom row). Stochastic simulations show the average over 100 runs including the 95% confidence interval, i.e. 95% of the trajectories stayed inside the envelope generated by the error bars. In order to show the impact of the team-size on the solution quality, two different team-sizes are simulated (the prediction of the macroscopic model is independent of the team-size for the models considered here).

B. Optimal control with unknown system parameters

We now consider a robotic-swarm with threshold-based task allocation with three tasks. The task duration is given by $\theta = (2.5, 4.3, 1.9)^T$, which is assumed to be unknown. For such a system, the optimal control $\mu_{1'}$ and $\mu_{2'}$ calculated using the estimate $\overline{\theta} = (2, 4, 2)^T$ will not lead to the desired outcome. Let the steady-state distribution observed on the real system (using θ and not $\overline{\theta}$) be $\hat{N}_1^*(\mu_{1'}, \theta)$ and $\hat{N}_2^*(\mu_{2'}, \theta)$, respectively

$$\hat{N}_1^*(\mu_{1'}, \theta) = (0.3937, 0.1113, 0.2053, 0.2897)^T$$
$$\hat{N}_2^*(\mu_{2'}, \theta) = (0.3770, 0.3266, 0.2014, 0.0951)^T$$

Solving the optimization problem from (4) yields $\theta = (2.5, 4.3, 1.9)^T$ for both $\hat{N}_1^*$ and $\hat{N}_2^*$ using the analytical solution and numerical optimization, respectively. θ finally being known, the improved optimal control can be calculated to

$$\hat{\mu}_{1'}^* = (0.5072, 0.0932, 0.0722, 1.0000)^T$$
$$\hat{\mu}_{2'}^* = (0.5947, 1.0000, 0.0735, 0.1508)^T$$

as in Section V-A, leading to the desired state distributions π_1 and π_2 with residual error less than 10^{-7}.

C. Optimal control with unknown system parameters for small teams

Using a real robotic system in the optimization loop, it is unlikely that the swarm accurately tracks the predicted average state distribution, but rather oscillates around the steady-state, in particular if the number of robots is low. In this case, it is unclear whether unexpected observations stem from randomness in the system, a wrong estimate of the system parameters when calculating the optimal control,

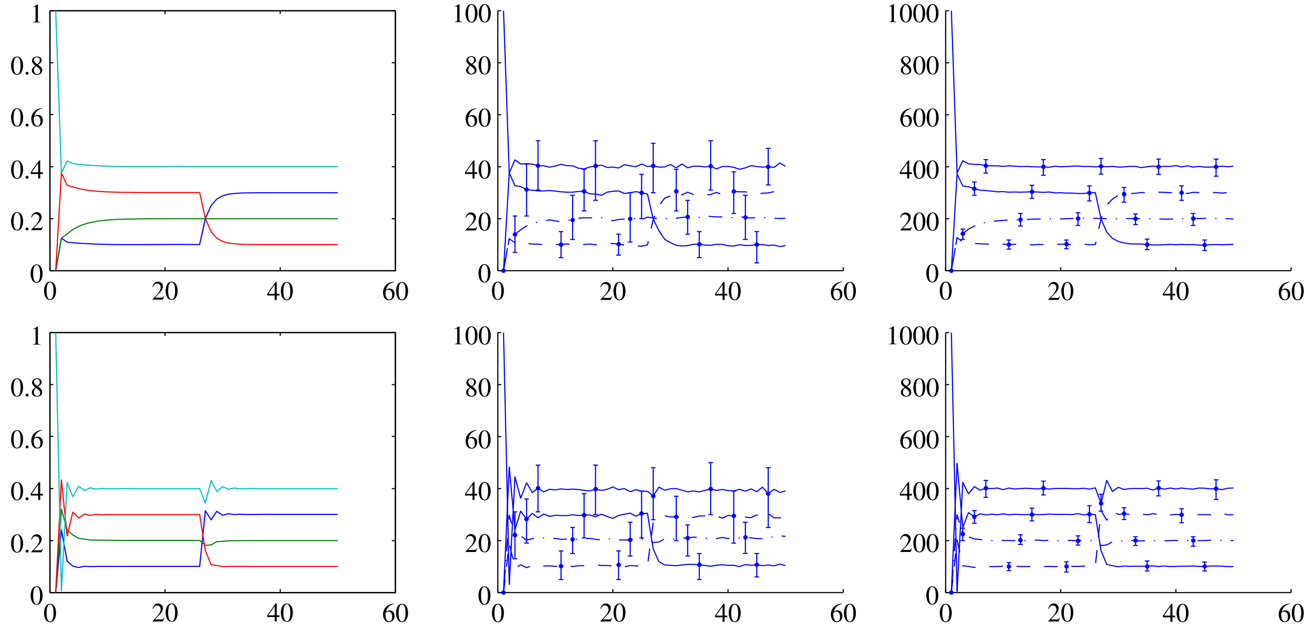

Fig. 2. Optimal control for swarms of 100 and 1000 robots. The swarms quickly reach the desired distribution [0.4 0.1 0.2 0.3] from the initial configuration [1 0 0 0] (all robots idle), and converge towards [0.4 0.3 0.2 0.1] after time-step 25 using the non-collaborative (*top row*) and the collaborative, threshold-based algorithm (*bottom row*). Macroscopic difference equations (*left*) and stochastic simulation (100 runs) for 100 and 1000 robots (*middle* and *right*). Error bars depict the 95% confidence interval.

or from intrinsic divergences between macroscopic representation of a system with small numbers of robots.

Fig. 3 shows the average steady-state estimate of 1000 stochastic simulations for 25 robots using the linear controller (left), and for 25 and 100 robots using the threshold-based controller (middle and right, respectively). The steady-state is estimated by averaging over the distribution from time-step 10 to 50. Results show that the minimal and maximal error (shown by the error-bars) is reasonable small already for small teams. For the threshold-based controller, however, results show a systematic error when the team-size is small. This is a typical artifact in stochastic simulation of non-linear systems. Further experiments (not shown) estimating the steady-state over a longer period of time, confirm this result.

VI. Discussion

In the proposed control architecture, robots fluctuate between states even when the system is at steady-state. These fluctuations are a function of the switching probabilities between robot states and are proportional to the speed with, which a system reaches the desired average distribution.

Although the chosen case study exemplifies the proposed methodology for a linear and non-linear model well and provides promising results, it is no guarantee for the general applicability of the method. Whether the proposed methodology is applicable for designing a distributed controller is dependent on three conditions. First, the system needs to be able to generate an observable that is constant and has small variance, e.g. a specific trajectory in state space or a steady-state. In the task allocation case study, such a metric is provided by a constant steady-state distribution, which can be robustly estimated by averaging over a certain time when the system is in steady-state regime. Second, a model that is able to provide quantitative agreement with the chosen metric is required. In this paper, we make use of probabilistic population dynamic models that can be systematically developed using the methodology brought forward in [3], [19], [20], as it allows to track the statistic mean of swarm dynamics. Third, a suitable optimization method is required to solve the resulting optimization problems. In particular for highly non-linear systems, finding an optimal control might be difficult, if not impossible.

For both the parameter estimation as well as for the optimization step, a reachability analysis of the model can answer whether optimal parameters that provide quantitative agreement of the model with experimental data exist, and whether a desired state is reachable. In our case study, the set of reachable distributions is constrained by the choice of θ (consider the extreme cases $T_i = 0$ or $T_i = \infty$ leading to $\pi_i = 0$ and $\pi_i = 1$, respectively). The set of reachable distributions thus defines whether an optimal control for a specific distribution exists or not, as well as whether model parameters exist that lead to quantitative agreement of the metric of interest with an observable of the real system.

The examples in this paper all aim at finding an optimal control for a specific steady-state of a system, which corresponds to the *average* system behavior. There is thus no guarantee for the system to not exhibit undesired behavior or to violate upper and lower bounds on the performance. Probabilistic simulation of the system, however, can help to estimate confidence intervals on the expected performance, which can potentially be used as constraints in the optimal

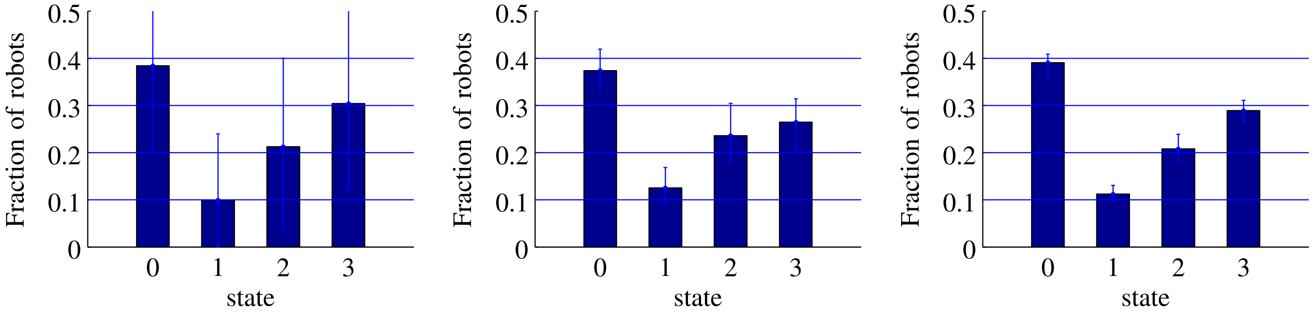

Fig. 3. Stochastic simulation of swarms of 25 using the linear controller (*left*), and of 25 and 100 robots (*middle* and *right*) for the non-linear controller for an optimal control leading to $\pi = (0.1, 0.2, 0.3, 0.4)^T$. Bars show the steady-state distribution from time-step 10–50. Error bars depict minimum and maximal values over 1000 experiments. Small team sizes show systematic errors.

control problem.

VII. Conclusion

We presented a methodology for finding optimal control parameters for a robot swarm in order to achieve a desired state distribution at steady-state, even if the system parameters are unknown. The approach requires that the system dynamics can be modeled by probabilistic population dynamic models, and that (numerical) solution methods to the arising optimization problem exist. For controllers whose collective dynamics can be described by a linear model, analytical solutions to the optimization problem can be found, whereas non-linear dynamics usually require numerical solutions by an appropriate numerical optimization algorithm. The methodology is illustrated using a task-allocation case study with a reactive and a threshold-based algorithm which lead to robust task allocation. The resulting population dynamics of both approaches have been captured by linear and non-linear macroscopic models. It was shown using stochastic simulations of swarms of various sizes that reliable steady-state estimates can already be achieved for small teams.

References

[1] E. Şahin, *Swarm Robotics*, ser. Lecture Notes in Computer Science. Springer Verlag, 2005, vol. 3342, ch. Swarm Robotics: From Sources of Inspiration to Domains of Application, pp. 10–20.

[2] K. Sugawara, M. Sano, I. Yoshihara, and K. Abe, "Cooperative behavior of interacting robots," *Artificial Life and Robotics*, vol. 2, pp. 62–67, 1998.

[3] A. Martinoli, K. Easton, and W. Agassounon, "Modeling of swarm robotic systems: A case study in collaborative distributed manipulation," *Int. J. of Robotics Research*, vol. 23, no. 4, pp. 415–436, 2004.

[4] K. Lerman, A. Martinoli, and A. Galystan, "A review of probabilistic macroscopic models for swarm robotic systems," in *Proc. of the SAB 2004 Workshop on Swarm Robotics*, Santa Monica, CA, USA, 2005, pp. 143–152, Lecture Notes in Computer Science Vol. 3342, Springer-Verlag, Berlin.

[5] N. Correll and A. Martinoli, "Towards optimal control of self-organized robotic inspection systems," in *8th Int. IFAC Symp. on Robot Control (SYROCO)*, Bologna, Italy, September 2006.

[6] D. Milutinovic and P. Lima, "Modeling and centralized optimal control of a large-size robotic population," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1280–1285, 2006.

[7] S. Berman, A. Halasz, V. Kumar, and S. Pratt, "Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system," in *Proc. of the SAB 2006 Workshop on Swarm Robotics*, Rome, Italy, September/October 2006.

[8] N. Correll and A. Martinoli, "System identification of self-organized robotic swarms," in *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*. Minneapolis, MN, USA (2006): Springer Distributed Autonomous Robotic Systems, 2006, pp. 31–40.

[9] B. Gerkey and M. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[10] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *PERVASIVE Computing*, pp. 24–33, October 2004.

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

[12] J. McLurkin and D. Yamins, "Dynamic task assignment in robot swarms," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[13] K. Sugawara and T. Mizuguchi, "A study on proportion regulation model for multi-robot system," in *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*. Minneapolis, MN, USA (2006): Springer Distributed Autonomous Robotic Systems, 2006, pp. 217–226.

[14] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies," *Proc. of the Roy. Soc. London B*, no. 3, pp. 191–209, 1997.

[15] W. Agassounon, A. Martinoli, and K. Easton, "Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes," *Autonomous Robots*, vol. 17, no. 2–3, pp. 163–191, 2004, special Issue on Swarm Robotics.

[16] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[17] M. J. B. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," *Robotics & Autonomous Systems*, vol. 30, no. 1-2, pp. 65–84, 2000.

[18] A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots," *Robotics & Autonomous Systems*, vol. 29, pp. 51–63, 1999, special Issue on Distributed Autonomous Robotic Systems.

[19] K. Lerman, A. Galstyan, A. Martinoli, and A.-J. Ijspeert, "A macroscopic analytical model of collaboration in distributed robotic systems," *Artificial Life*, vol. 7, no. 4, pp. 375–393, 2001.

[20] N. Correll, "Coordination schemes for distributed boundary coverage with a swarm of miniature robots: Synthesis, analysis and experimental validation," Ph.D. dissertation, École Polytechnique Fédérale Lausanne, 2007.