Distributed Estimation and Control For Stochastically Interacting Robots

Fayette W. Shaw and Eric Klavins

Abstract— **We introduce a distributed estimation algorithm for use by a collection of stochastically interacting agents. Each agent has both a discrete value and an estimate of the mean of that value taken over all agents. The estimates are updated according to a local rule when pairs of agents interact. In this paper we prove that the ensemble average of the estimates converges to the correct global average. We then use the estimate information to control the agents to a desired average value. Furthermore, we demonstrate the algorithm experimentally using the *Programmable Parts Testbed* [1].**

I. Introduction

We consider a kind of consensus problem motivated by the self-assembly of simple robotic parts [1] into a larger structure. As the parts form larger and larger subassemblies of the desired structure, controlling the relative numbers of each subassembly type becomes important for maximizing the yield of the process. For example, suppose an assembly of type A is formed from subassemblies of type B and C, which are each made up of some number of parts. To maximize the yield of A, it is desirable to balance the number of parts destined to compose each subassembly by the rate at which the subassemblies form. That is, the *stoichiometry* of B and C must be maintained at $1:1$. In this paper we capture the essence of this problem by supposing that in a population of stochastically interacting robots, each robot has a discrete state that is either zero or one and that, by switching between these values, they can control the ratio $\rho(t)$ of robots of state one to the total number of robots. In particular, we concentrate first on supplying each robot with an estimate of $\rho(t)$ so that it can decide how to change its discrete state to effect the evolution of $\rho(t)$.

The particular physical setting we have at hand is the *Programmable Parts Testbed* [1]. In this testbed, robotic parts are randomly mixed on an air-table to induce random collisions. We have programmed the parts so that, when two of them collide they briefly bind together to share information and then they detach. In other work we have experimentally verified that this process can be reasonably described by a continuous time, jump Markov process in which a vector of discrete states (the zeros and ones) and a vector of estimates (each entry maintained by a different robot) is updated upon random collisions between random pairs of robots.

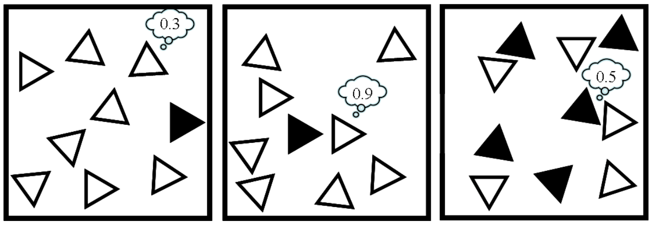


Fig. 1. Cartoon of Programmable Parts with discrete states zeros (black) and ones (white). Each robot has its own measurement, or estimate, of the ratio of white robots to total number of robots n, indicated by the number in the thought bubble. The goal is to drive the average of the discrete values to a reference, in this case 0.5, which is depicted in the last panel.

To some extent, existing results on consensus over randomly changing networks [2] apply here, except that we desire that the robots do not simply come to a consensus of their initial states, but that they *track* the time varying average of their discrete states (i.e. they track their stoichiometry). Thus, our approach uses both the estimates and discrete states of the robots when updating, which is similar to [3] except that we track a discrete signal in a stochastically changing network. The trade-off is that we can no longer guarantee that the estimate converges with probability one to the correct value, although we can come arbitrarily close to such an equilibrium distribution by tuning a "consensus" parameter.

The specific contributions of this paper are as follows. First, we introduce a simple update rule defining a distributed estimation scheme that balances consensus with tracking. We then show that the probability distribution of the vector of estimates obeys a *Master Equation* [4] and derive from it the first and second moment dynamics. From these we can compute the evolution of the mean estimate and its variance. We show that when the robots ignore the discrete state and simply do consensus, the estimates converge with probability one to the average of the initial values of the estimates. In addition, by appropriately weighting the discrete state in the update rule we can track the average ρ with an arbitrarily small variance at steady state. We demonstrate our algorithm with simulations and with experiments using the *P*rogrammable Parts Testbed. Finally, we show (in simulation) that a simple

This work is supported by NSF grants #0347955 and #0501628. Fayette W. Shaw is a graduate student in the Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, fayshaw@u.washington.edu. Eric Klavins is an Assistant Professor in the Department of Electrical Engineering, University of Washington, Seattle, WA 98195, klavins@ee.washington.edu.

distributed control law can be composed with the estimator and that it achieves the desired steady state stoichiometry as though it were using the actual value of $\rho(t)$.

II. Related Work

Our work is an extension of standard consensus [5] in that we control a group of agents to *track* a varying signal $\rho(t)$. In this sense, the present paper is quite similar to [6] except that we assume a stochastic network and the properties of the stochastic process that result. A randomly changing network is considered in [2] and the results there apply to our system when our consensus parameter $\zeta = 1$. However, the algorithm in [2] does not track a changing state. Our work also differs from [2] in that we investigate the continuous time *moment dynamics* of our the estimator instead of a discrete time system. This distinction is crucial for our system so that we may implement a controller that uses the estimate information and runs concurrently on a different time-scale. Investigating the second moment dynamics of our system also results in an alternative proof method for the probability-one convergence of consensus in [2] and also allows us to predict the non-zero equilibrium variance of the estimate when we track $\rho(t)$.

Distributed systems has its foundations in computer science [7] and has broadened to include many applications. Thus, distributed and decentralized estimation have many meanings in the literature. Often the terms refer to sensing and sensor fusion [3], where agents may independently perform Kalman filtering to fuse measurements. In the present paper, distributed estimation refers to agents in a network which each have a measurement of the global state, which get updated through local interactions.

In this paper, we also describe a controller that updates the discrete state based to force $\rho(t)$ to a desired reference. To achieve this, the robots switch roles in the network similar to examples of agents dynamically updating their roles in robotics and biology. For example, in [8], robots dynamically allocate roles to actuate highly coordinated actions. In nature, ants switch between foraging and nest repair [9] in such a way as to achieve a balance of responsibility. This type of controller could be used in more complex cooperative control scenarios where balancing activities based on an estimate of available resources or outstanding tasks is important.

III. Problem Setup and Notation

Consider a set of n robots similar to the programmable parts [1]. Each robot i has a discrete internal state $q_i(t) \in \{0,1\}$. Define

$$\rho(t) \triangleq \frac{1}{n}\sum_{i=1}^{n} q_i(t) \tag{1}$$

where n is the number of robots. Each robot also maintains an estimate $x_i(t) \in [0,1]$ of $\rho(t)$. It is assumed that each robot knows the value of n. The vector $\mathbf{q} = (q_1, ..., q_n)^T$ is defined to be the vector of internal states and $\mathbf{x} = (x_1, ..., x_n)^T$ to be the vector of estimates. The symbol $\langle\cdot\rangle$ denotes expected value and $\mathbb{1} = (\underbrace{1, \cdots, 1}_{n})^T$.

In our system, the robots are mixed together to induce collisions. We assume that in the next dt seconds there is a probability $k\ dt$ that any particular pair of robots i and j will collide and that any pair of robots is equally likely to interact next (that is, the system is well-mixed). We have experimentally verified this assumption and measured the rate k in other work [1]. When two robots collide, they are programmed to simply exchange their discrete states and estimates and update their estimates based on this information. Also, the robots concurrently update their discrete states so as to achieve a desired discrete state. We address the following three problems:

a) The Estimation Problem Define an estimator update $(x_i, x_j) \mapsto f(x_i, q_i, x_j, q_j)$ so that $x_i(t)$ converges to $\rho(t)$ as $t \to \infty$ with high probability.

b) The Control Problem Define a rate function $K_i(\rho, q_i)$ at which robot i switches from q_i to $1 - q_i$ so that $\rho(t)$ converges to a desired reference ratio r (a constant) with high probability *and* $K_i(\rho, q_i)$ converges to zero (the robots eventually stop switching).

c) The Simultaneous Control and Estimation Problem Demonstrate that the a solution to the *control problem* running concurrently with a solution to the estimation problem (that is, with $K_i(x_i, q_i)$ defining the rate at which robots switch states) drives $x_i(t)$ and $\rho(t)$ to r with high probability.

In this paper, we solve the first two problems formally, and demonstrate in simulation a working solution to the last problem.

IV. The Estimator

We consider the case in which the estimator update function is defined by a convex combination of the estimates and states of the interacting robots. In particular, if robot i interacts with robot j at time t then the robots update their estimates according to

$$\begin{aligned} x_i(t^+) &= f(x_i(t^-), q_i(t^-), x_j(t^-), q_j(t^-)) \\ x_j(t^+) &= f(x_j(t^-), q_j(t^-), x_i(t^-), q_i(t^-)) \\ x_k(t^+) &= x_k(t^-) \text{ for all } k \neq i, j, \end{aligned}$$

where $f(x_i(t^-), q_i(t^-), x_j(t^-), q_j(t^-))$ is defined by

$$\zeta\left(ax_i + (1-a)x_j\right) + (1-\zeta)\left(\frac{1}{n}q_i + \left(\frac{n-1}{n}\right)q_j\right).$$

Here $\zeta \in (0,1)$ is the *consensus parameter*, which is the weighting of the relative importance of the estimates and discrete states in the update rule; $\frac{1}{n}$ is the weighting of a robot's own discrete state; and $a \in (0,1)$ is the weighting on a robot's own estimate. The symbols t^- and t^+ denote the times immediately before and after the interaction, respectively. The last line of the above update rule represents the fact that robots not participating in the interaction do not update their estimates.

The update equations can be written using matrices. For example, in a three-robot system in which robots 1 and 2 happen to interact at time t, the update rule is

$$\begin{aligned}\mathbf{x}(t^+) &= \zeta \begin{pmatrix} a & 1-a & 0 \\ 1-a & a & 0 \\ 0 & 0 & \frac{1}{\zeta} \end{pmatrix} \mathbf{x}(t^-) \\ &\quad + (1-\zeta) \begin{pmatrix} \frac{1}{n} & 1-\frac{1}{n} & 0 \\ 1-\frac{1}{n} & \frac{1}{n} & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{q}(0),\end{aligned}$$

or more compactly and generally

$$\mathbf{x}(t^+) = \zeta \mathbf{A}_{ij}\mathbf{x}(t^-) + (1-\zeta)\mathbf{B}_{ij}\mathbf{q}(0). \qquad (2)$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as follows:

$$\begin{array}{ll} \mathbf{A}_{ij}(i,i) = \mathbf{A}_{ij}(j,j) = a & \mathbf{B}_{ij}(i,i) = \mathbf{B}_{ij}(j,j) = \frac{1}{n} \\ \mathbf{A}_{ij}(i,j) = \mathbf{A}_{ij}(j,i) = 1-a & \mathbf{B}_{ij}(i,j) = \mathbf{B}_{ij}(j,i) = \frac{n-1}{n} \\ \mathbf{A}_{ij}(k,k) = \frac{1}{\zeta} & \end{array}$$

and all remaining matrix entries are 0.

For a process (Y,t), the change in probability can be expressed as a Master Equation [4]

$$\frac{d}{dt}P(y,t) = \int W(y|y')P(y',t) - W(y'|y)P(y,t)dy',$$

where $W(y|y')$ is the probability that the system transitions to y in time $t+dt$ given that state is y' at time t. The first term in the integral refers to the transitions going into the state and the second refers to the transitions leaving. Using (2) to derive the Master Equation for the estimator process we arrive at

$$\begin{aligned}\frac{d}{dt}P(\mathbf{x},t) = \frac{k}{\zeta}\sum_{i<j}\frac{1}{|\mathbf{A}_{ij}|}\, P\left(\frac{1}{\zeta}{\mathbf{A}_{ij}}^{-1}\left(\mathbf{x} - (1-\zeta)\mathbf{B}_{ij}\mathbf{q}\right), t\right) \\ - k\binom{n}{2}P(\mathbf{x},t), \qquad (3)\end{aligned}$$

where $P(\mathbf{x},t)$ is the probability of having the estimate vector $\mathbf{x}$ at time t and k is the rate of the robots' pairwise interaction.

A. First Moment Dynamics

The first moment of the estimator process are examined using the Master Equation (3). The dynamics of the expected value of the estimate vector $\langle \mathbf{x} \rangle$

$$\frac{d}{dt}\langle \mathbf{x}\rangle = k\left\langle \left(\zeta\sum_{i<j}\mathbf{A}_{ij} - \binom{n}{2}\mathbf{I}\right)\mathbf{x} + (1-\zeta)\sum_{i<j}\mathbf{B}_{ij}\mathbf{q}\right\rangle. \qquad (4)$$

This equation can be simplified as follows. Define

$$\mathbf{A} \triangleq \sum_{i<j}\mathbf{A}_{ij} \quad \text{and} \quad \mathbf{B} \triangleq \sum_{i<j}\mathbf{B}_{ij}.$$

For now, we assume that the discrete state $\mathbf{q}$ is constant. It can be shown that

$$\begin{aligned}\mathbf{A} &= (n-1)\left(a + \frac{n-2}{2\zeta}\right)\mathbf{I} + (1-a)\mathbb{1}\mathbb{1}^T \qquad (5)\\ \mathbf{B} &= \frac{n-1}{n}\mathbb{1}\mathbb{1}^T. \qquad (6)\end{aligned}$$

Therefore, equation (4) becomes

$$\frac{d}{dt}\langle \mathbf{x}\rangle = k\left[\left(\zeta\mathbf{A} - \binom{n}{2}\mathbf{I}\right)\langle \mathbf{x}\rangle + (1-\zeta)\mathbf{B}\mathbf{q}\right]. \qquad (7)$$

Theorem 1: The unique fixed point of $\frac{d}{dt}\langle \mathbf{x}\rangle$ is

$$\langle \mathbf{x}\rangle^* = \rho\mathbb{1}. \qquad (8)$$

That is, the estimates converge to the average value of the discrete states, assuming the discrete states are constant.

We confirm (8) in Appendix A and its uniqueness in Appendix B.

Theorem 2: The fixed point $\langle \mathbf{x}\rangle^* = \rho\mathbb{1}$ is stable.

Proof: By (7) and since at equilibrium $\mathbf{q}$ is deterministic, it suffices to show that

$$\mathbf{H} \triangleq \zeta\mathbf{A} - \binom{n}{2}\mathbf{I}$$

is negative semi-definite. Note that each term in the sum $\mathbf{A}$ has the same eigenvalues since the $\mathbf{A}_{ij}$ matrices are permutations of each other. To show that any $\mathbf{A}_{ij} - \mathbf{I}$ has negative eigenvalues it suffices to show that the eigenvalues of $\mathbf{A}_{ij}$ are less than 1, as all zero eigenvalues represent robots that are not updating. Thus without loss of generality we can examine

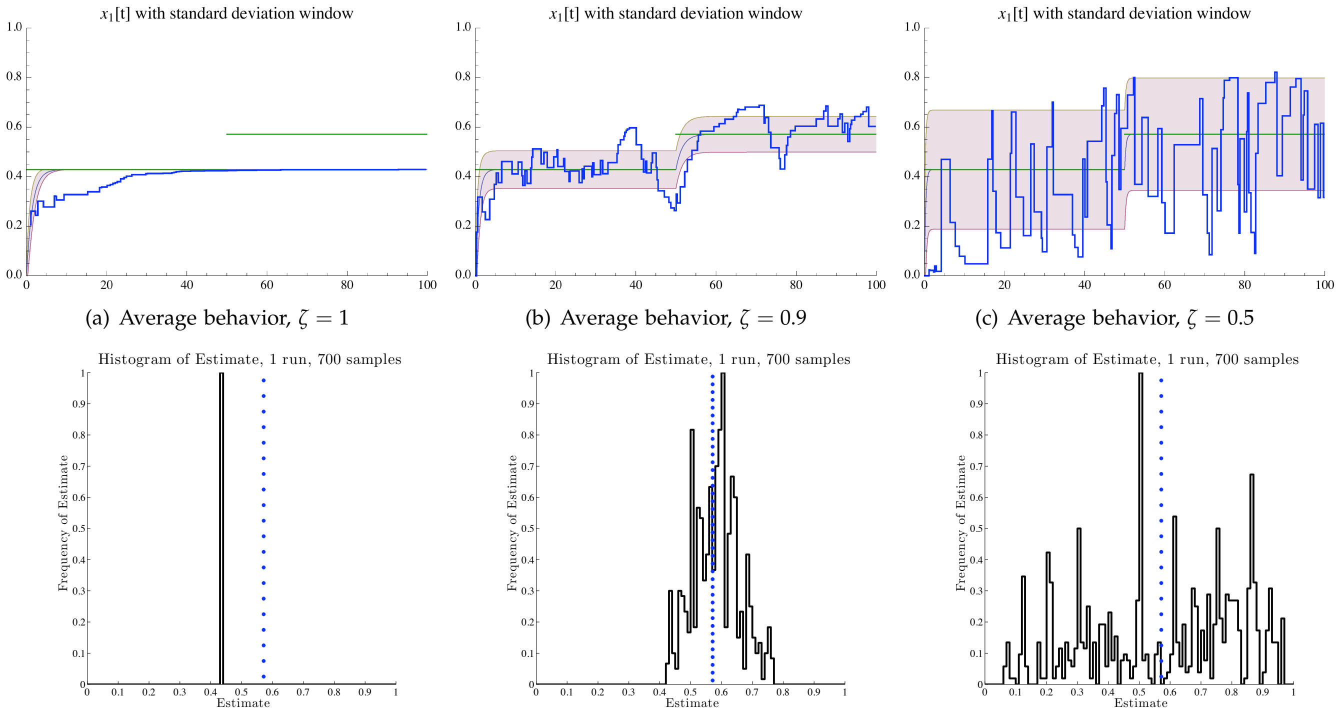
$$\mathbf{A}_{12} = \left(\begin{array}{cc|c} v & u & \mathbf{0} \\ u & v & \\ \hline \mathbf{0} & & \mathbf{I} \end{array}\right),$$

whose eigenvalues are $\{1, v-u, v+u\}$. The eigenvalue 1 has multiplicity $n-2$. Finally, all eigenvalues are less than or equal to 1, based on our assumptions on a and ζ. ■

B. Second Moment Dynamics

Similarly, we derive the dynamics for the second moment $\langle \mathbf{x}\mathbf{x}^T\rangle$. Using the Master Equation we have

$$\begin{aligned}\frac{d}{dt}\langle \mathbf{x}\mathbf{x}^T\rangle &= k\left\langle \sum_{i<j}\left(\mathbf{C}_{ij}\mathbf{x} + \mathbf{D}_{ij}\mathbf{q}\right)\left(\mathbf{C}_{ij}\mathbf{x} + \mathbf{D}_{ij}\mathbf{q}\right)^T\right\rangle \\ &\quad - k\binom{n}{2}\langle \mathbf{x}\mathbf{x}^T\rangle\end{aligned}$$



(a) Average behavior, $\zeta = 1$

(b) Average behavior, $\zeta = 0.9$

(c) Average behavior, $\zeta = 0.5$

(d) Histogram, $\zeta = 1$

(e) Histogram, $\zeta = 0.9$

(f) Histogram, $\zeta = 0.5$

Fig. 2. Comparisons of simulated data with various parameters for 7 robots. Figures 2(a), 2(b), and 2(c) show a step change of the discrete states from a mean of $\frac{3}{7}$ to $\frac{4}{7}$ indicated in green. The data is simulated from the perspective of a robot starting at a discrete state 0. The purple regions indicate the mean plotted with error bars. The blue lines indicate single estimate trajectories. The simulations vary parameter ζ in the update rule and result in these different plots. In Figure 2(a), ζ=1 and this is equivalent to the consensus algorithm presented in [2] and others. The mean of estimates converges to a delta function, as seen in Figure 2(d), to the incorrect mean discrete state; the correct stat is plotted as a dotted blue line. Figure 2(b) demonstrates our algorithm with parameter $\zeta = 0.9$. The mean converges close to the correct mean discrete state with a wider distribution than delta function. As the parameter ζ decreases, the variance becomes larger but convergence is faster.

where $\mathbf{C}_{ij} = \zeta \mathbf{A}_{ij}$ and $\mathbf{D}_{ij} = (1-\zeta)\mathbf{B}_{ij}$. The second moment dynamics can be expressed as

$$\begin{aligned}\frac{d}{dt}\, vec(\langle \mathbf{x}\mathbf{x}^T\rangle) = k\sum_{i<j}\Big[&\mathbf{C}_{ij}\otimes\mathbf{C}_{ij}\; vec(\langle \mathbf{x}\mathbf{x}^T\rangle)\\ &+ \mathbf{C}_{ij}\otimes\mathbf{D}_{ij}\; vec(\mathbf{q}\langle \mathbf{x}^T\rangle) + \;\mathbf{D}_{ij}\otimes\mathbf{C}_{ij}\; vec(\langle \mathbf{x}\rangle\mathbf{q}^T)\\ &+ \mathbf{D}_{ij}\otimes\mathbf{D}_{ij}\; vec(\mathbf{q}\mathbf{q}^T)\Big] - k\binom{n}{2} vec(\langle \mathbf{x}\mathbf{x}^T\rangle)\end{aligned}$$

where $\otimes$ is the Kronecker product and $vec(\cdot)$ is the vector representation of a matrix where matrix columns are concatenated vertically.

C. Second Moment Equilibrium

The equilibrium value of the second moment is a tedious function of $\mathbf{q}$ and the parameters ζ, a, and n. However, a simple expression for the equilibrium can be obtained when $\zeta = 1$, which corresponds to pure consensus. Our argument (below) amounts to an alternative proof that consensus in this setting converges to the average of the initial conditions of the estimates with probability one. Said differently, when $\zeta = 1$ the variance at equilibrium is 0 (even though the estimate is completely wrong).

It is also evident that as ζ decreases, the variances increase although the expected value of the estimate is correct. This is illustrated in Figure 2 where the estimator attempts to track a changing $\rho(t)$. When $\zeta = 1$, the estimator (i.e. pure consensus) fails. With $\zeta < 1$, the estimate can track $\rho(t)$, but with a non-zero variance at steady state.

When $\zeta = 1$, the expression for the second moment dynamics reduces to

$$\frac{d}{dt}\langle \mathbf{x}\mathbf{x}^T\rangle = \sum_{i<j} \mathbf{A}_{ij}\langle \mathbf{x}\mathbf{x}^T\rangle \mathbf{A}_{ij}. \tag{9}$$

Setting the derivative to zero and solving for $\langle \mathbf{x}\mathbf{x}^T\rangle$ yields $\langle \mathbf{x}\mathbf{x}^T\rangle = w\mathbb{1}\mathbb{1}^T$ where w is a scalar. Now, define

$$V = \mathbb{1}^T\langle \mathbf{x}\mathbf{x}^T\rangle\mathbb{1}.$$

Multiplying (9) by $\mathbb{1}^T$ on the left and by $\mathbb{1}$ on the right, noting that $\mathbb{1}^T\mathbf{A}_{ij} = \mathbb{1}^T$, and noting that $\mathbf{A}_{ij}\mathbb{1} = \mathbb{1}$ shows that $\frac{d}{dt}V = 0$.

Now, initially $\langle \mathbf{x}\mathbf{x}^T\rangle = \mathbf{x}(0)\mathbf{x}^T(0)$ (i.e. the variances are zero initially since we start with deterministic values for $\mathbf{x}$). Thus, $V^* = V(0)$. Also, the equilibrium value for $\langle \mathbf{x} \rangle$ (only when $\zeta = 1$) is

$$\eta \triangleq \frac{1}{n}\mathbb{1}^T \mathbf{x}(0)$$

so that $V(0) = \mathbb{1}^T \mathbf{x}(0)\mathbf{x}^{\mathbf{T}}(0)\mathbb{1} = \eta^2 n^2$. Thus, writing out $V(0) = V^*$ we have

$$\eta^2 n^2 = wn^2$$

so that $w = \eta^2$. Thus, the covariance matrix at equilibrium is

$$\mathbf{C} = \langle \mathbf{x}\mathbf{x}^T \rangle^* - \langle \mathbf{x} \rangle^* \langle \mathbf{x}^T \rangle^* = w\mathbb{1}\mathbb{1}^T - \eta^2 \mathbb{1}\mathbb{1}^T = \mathbf{0}.$$

D. Second Moment Stability

Theorem 3: The equilibrium of the second moment dynamics is stable.

Since we consider $\mathbf{q}$ to be deterministic , we examine the following matrix to determine stability of $\langle \mathbf{x}\mathbf{x}^T \rangle$:

$$\mathbf{M} \triangleq k\sum_{i<j} \mathbf{A}_{ij} \otimes \mathbf{A}_{ij} - k\binom{n}{2}\mathbf{I} = k\sum_{i<j}(\mathbf{A}_{ij} \otimes \mathbf{A}_{ij} - I).$$

Call λ_i the eigenvalues of $\mathbf{A}$ and μ_i the eigenvalues of $\mathbf{B}$. Then the eigenvalues of $\mathbf{A} \otimes \mathbf{B} = \lambda_i \mu_i$. We showed in Theorem 2 that the eigenvalues of $\mathbf{A}_{ij}$ were $\{1, v-u, v+u\}$. Thus, the eigenvalues $\mathbf{A}_{ij} \otimes \mathbf{A}_{ij}$ are $\{1, v-u, v+u, (v-u)^2, (v-u)(v+u), (v+u)^2\}$, which are all less than 1, based on our assumptions on a and ζ.

As in the proof for Theorem 2 each term in this sum is symmetric and negative semi-definite, from which it can be concluded that $\mathbf{M}$ itself is negative semi-definite.

V. Demonstration of the Distributed Estimator in Simulation and Experiment

We demonstrate the algorithm by directly simulating the system using the *Stochastic Simulation Algorithm* [10]. Figure 2 shows simulations of the update rule with various parameters. Note that 2(a) reduces our formulation to the consensus algorithm presented in [2]. Various choices of parameters result in different convergence speed and variance.

The PPT has been adapted to make internal estimation states observable by an overhead camera. Each robot computes its estimate with 7-bit accuracy and displays it as a 5-bit number using LEDs. Each robot has a bright blue LED in its center and a green LED to indicate the lowest bit of the estimate, and continuing clockwise to indicate the binary estimate as depicted in Figure 3. The estimates indicated by the LEDs were automatically extracted from images using MATLAB. The robots display a quantization error of 2^{-7} in Figure 4, which appears to be a steady state error.

VI. Distributed Control of Stoichiometry

We now address the *control problem* discussed in Section III. That is, we define a rate at which the robots should flip their discrete states from 0 to 1 and *vice versa* so that (a) the $\rho(t) \to r$ (a constant) and (b) the robots eventually stop switching. For now we assume that the robots have perfect knowledge of $\rho(t)$. Later, we replace this knowledge with the estimated value computed in the previous section. The update rule for robot i when changing its discrete state is simply

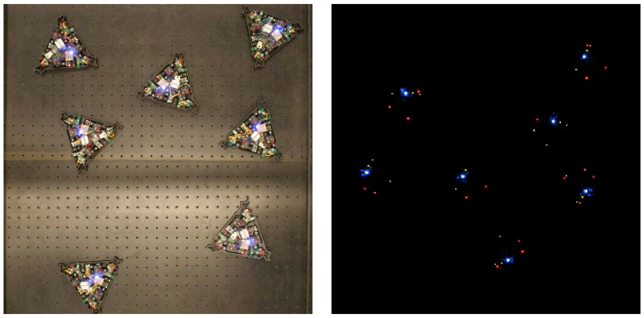
$$q_i(t^+) = 1 - q_i(t^-).$$

There are many possible control schemes. Here is a simple one: robot i toggles its state at the rate

$$K_i \triangleq |q_i - r||\rho - r|.$$

Consider the random variable $N(t) = \sum_{i=1}^n q_i(t)$. Define $\mu_i dt$ to be the rate at which N transitions from $N = i$ to $N = i+1$ in the next dt seconds. Similarly, let $\lambda_i\, dt$ to be the rate at which N transitions from $N = i$ to $N = i-1$ in the next dt seconds. Then, we have a birth-death chain where

$$\mu_i = i(1-r)\left|\frac{i}{n} - r\right| \quad \text{and} \quad \lambda_i = (n-i)r\left|\frac{i}{n} - r\right|.$$



(a) Photo of Programmable Parts (b) LEDs indicating estimates

Fig. 3. Programmable Parts Testbed. 3(a) shows the robots. 3(b) shows the lighting conditions for runs. On each robot the blue LED points to the lowest significant bit to indicate a binary estimate.

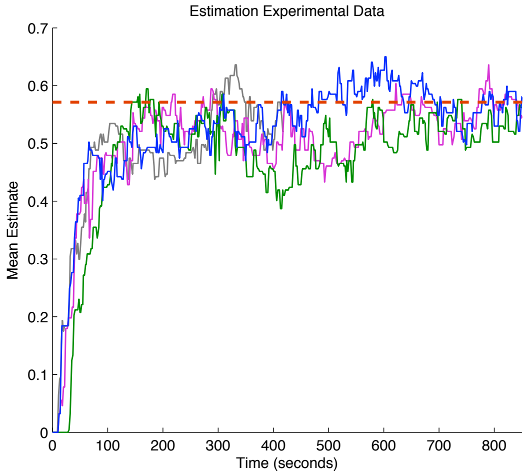
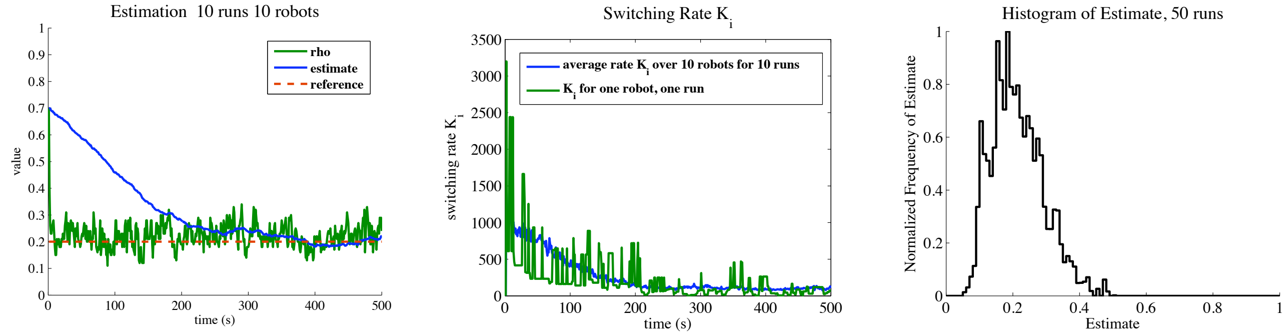


Fig. 4. Experimental data on PPT. Each line represents a run with all estimates averaged together. (Current setup does not allow for individual tracking). The red dashed line indicates the average discrete state in all runs. The parameters for this run are $\zeta = 0.8$ and $a = 0.5$.



(a) Average Behavior

(b) Rate K_i.

(c) Histogram.

Fig. 5. Controller with estimator. Figure 5(a) is a plot of 50 runs with 10 robots. The green line indicates the average of discrete states ρ, the thick blue line is the average of estimates over all robots, and the red dashed line is the reference for this trajectory. Parameters are $\zeta = 0.9$ and $a = 0.9$. Note that the controller using the estimate information does not stop as the perfect controller does. Figure 5(c) shows a histogram of the last hundred timesteps over all runs.

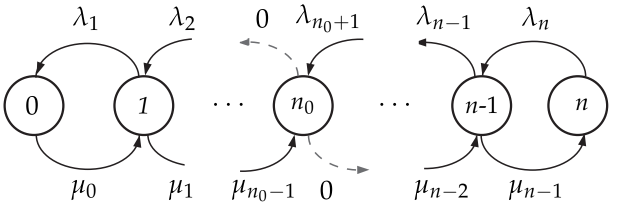


Fig. 6. Birth-death process modeling the controller. States denote ρ or number of robots with discrete state 1. Rates μ_i indicate increases in ρ while λ_i indicate decreases. Note that $n_0 = \rho n$ is a sink state and there are rates of 0 leaving it.

To understand the expression for μ_i note that when $N(t) = i$, there are i different robots that could transition from 0 to 1 and they each do so at the same rate $|q_j - r||\rho - r| = (1 - r)\left|\frac{i}{n} - r\right|$ since $q_j = 0$ and $\rho = \frac{i}{n}$. The expression for λ_i is similar.

Now, choose $r = \frac{n_0}{n}$ for some $0 \leq n_0 \leq n$. Then $\mu_{n_0} = \lambda_{n_0} = 0$ and the state n_0 is absorbing: all of the probability mass of $N(t)$ eventually flows to n_0. Since $n_0 = rn$ we get $\rho(t) \to r$. This is illustrated in Figure 6.

In Figure 5(a) we show a simulation of the control scheme where

$$K_i \triangleq |q_i - r||x_i - r|.$$

This suggests that the estimation algorithm described above does indeed yield a value that can be used to control the system. We are currently working on a proof that this composition of the estimator and the control indeed results in the correct behavior.

VII. Discussion

A. Contributions

In this paper, we described and verified a distributed estimation and control algorithm that allows a stochastically interacting group of agents to form local estimates of the stoichiometry and control it to a desired, stable fixed point. Convergence of the first and second moments of the estimate are proven formally and demonstrated empirically. Furthermore, the algorithm is applicable in the case of a changing global property. Current work in consensus utilizes discrete time arguments to prove convergence but cannot address concurrent processes occurring at different rates.

B. Future work

In future work, we will improve the estimator's variance by inserting a feedforward term and develop a proof that the estimator and controller work together. Also, we plan to extend the algorithm so that estimates of other global quantities can be tracked (e.g. the relative concentrations of arbitrary numbers of assembly types). The analysis will be extended to describe both higher order moments and the dynamics of subpopulations of agents having different discrete states.

VIII. Acknowledgments

The authors would like to thank Daniel Georgiev and Maryam Fazel for aid in model formulation, Samuel Burden for assistance in simulation, and Nils Napp and Alex Leone for collecting experimental data.

References

[1] E. Klavins, "Programmable self-assembly," *Control Systems Magazine*, vol. 24, pp. 43–56, August 2007.

[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, 2003.

[3] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed sensor fusion using dynamics consensus," in *IFAC World Congress*, 2005.

[4] N. G. V. Kampen, *Stochastic Processes in Physics and Chemistry*. Elsevier, 2007.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proceedings of the IEEE*, vol. 95, pp. 215–233, January 2007.

[6] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus for mobile networks," in *IFAC World Congress*, 2005.

[7] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc, 1996.

[8] L. Chaimowicz, V. Kumar, and M. F. M. Campos, "A paradigm for dynamic coordination of multiple robots," *Journal Autonomous Robots*, vol. 17, pp. 7–21, July 2004.

[9] D. M. Gordon, *Information Processing in Social Insects*, ch. Interaction patterns and task allocation in ant colonies. BirkhŁuser, 1999.

[10] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, 1977.

Appendix

A. First moment equilibrium solution

We show that (8) is an equilibrium solution for the first moment dynamics (7). Since we are interested in the solution for $\frac{d}{dt}\langle \mathbf{x} \rangle = 0$ it is equivalent to show

$$\left(\zeta\mathbf{A} - \binom{n}{2}\mathbf{I}\right)(\rho\mathbb{1}) = -(1-\zeta)\mathbf{Bq}. \tag{10}$$

Using the definition of $\mathbf{A}$ in (5) the left-hand side of (10) becomes

$$\rho(n-1)\left(\zeta\left(\frac{n-2}{2\zeta}+1\right)-\frac{n}{2}\right)\mathbb{1},$$

which reduces to

$$\rho(n-1)(\zeta-1)\mathbb{1}.$$

Substituting the definitions of $\mathbf{B}$ and ρ into the right-hand side gives

$$(1-\zeta)\frac{n-1}{n}\mathbb{1}\mathbb{1}^T\mathbf{q} = -\rho(n-1)(\zeta-1)\mathbb{1}.$$

Thus, (10) is true. ■

B. Proof of Uniqueness for Equilibrium Solution

Since (7)

$$\langle x \rangle * = (1-\zeta)\left(\binom{n}{2}I - \zeta\mathbf{A}\right)^{-1}\mathbf{Bq}.$$

To prove that this is a unique solution, we show that the matrix

$$\mathbf{H} = \binom{n}{2}I - \zeta\mathbf{A}$$

is invertible in the parameter region of interest. The matrix $\mathbf{H}$ is singular when

$$\mathbf{H} = u\mathbb{1}\mathbb{1}^T \text{ and } \mathbf{H} = -(n-1)\mathbf{I} + (\mathbb{1}\mathbb{1}^T - \mathbf{I}).$$

However the values for which this holds,

$$\{v = 0\} \text{ and } u = 1, v = -n$$

The values that make $v = 0$ are $n = 1$, which is a trivial estimation problem, and $\zeta a = -1$, which is not allowable by our parameter choices. To examine the second condition, we compute $\mathbf{H}\mathbb{1} = (n-1)(1-\zeta)\mathbb{1}$. For this to hold true with $v = -n$ and $u = 1$, the parameter $\zeta = 1$, which we already know reduces to the traditional consensus case and doesn't converge to our desired fixed point. Thus, $\mathbf{H}$ is invertible in the parameter region of interest. ■