

Distributed Estimation and State Assignment for Stochastically Interacting Robots

Fayette W. Shaw* Eric Klavins**

* Mechanical Engineering Department, University of Washington, Seattle,
WA (e-mail: fayshaw@u.washington.edu).

** Electrical Engineering Department, University of Washington, Seattle
WA (e-mail: klavins@ee.washington.edu)

Abstract: We consider a distributed tracking problem where agents interact locally with limited information. Each agent maintains both a discrete value and an estimate of the mean of that value taken over all agents. In earlier work, we designed an estimator that converged to the desired value with a finite variance and here, we derive a different estimator with zero variance. We design the controller and estimator separately, prove their simultaneous convergence and stability, finally demonstrate the results in simulation. While we present this work in the context of stochastic self-assembly, the algorithm can be applied other settings.

Keywords: Coordinated control and estimation over networks; Consensus problems; Decentralized algorithms for computation over sensor networks

1. INTRODUCTION

We address the problem of distributed estimation and control of agents in a stochastic environment. In this paper, distributed estimation refers to ensemble tracking of a changing value and distributed control refers to asynchronous and simultaneous state assignment to agents.

This work is motivated by, but not limited to, the Programmable Parts Testbed (PPT) (Klavins, 2007), which consists of triangular robots that are propelled by an external system of randomly actuated airjets. When two robots collide, they temporarily join using the embedded magnets in their chassises and exchange information via an infrared communication channel. Collision is the only mechanism for information exchange and occurs stochastically. Additionally, the robots can detach from one another by rotating magnets that are mounted to motors. Thus, the robots selectively reject undesirable configurations.

In this paper, each agent can be one of two discrete states, or *types*, and calculates an estimated *population fraction*, which is the fraction of robots in the system of a particular type. Additionally, the robots have the ability to transform from one type to another; the rate of switching is the control effort. Based on their estimates of the population fraction, agents switch their type to match a desired *reference* population fraction. This reference can be determined by how many robots should be allocated to certain tasks or, in the case of self-assembly, how many parts should in a sub-assembly.

The contribution in this paper is a distributed estimator of the population fraction that informs a stochastic controller and together results in a zero-variance algorithm. We introduce an estimator and controller, prove their con-

vergence separately and together, and demonstrate our results in simulation. While we present this work with the example of stochastic self-assembly, the theory can be generalized to be used in other examples such as swarms, both robotic (Halasz et al., 2007) and biological (Seeley, 1996).

2. RELATED WORK

The distributed estimation and control problem considered here has ties to existing research. In particular, the tools of distributed consensus are related to the estimation process we develop. Much of the literature addresses these problems using gossip algorithms (Boyd et al., 2006), sensor fusion (Xiao et al., 2005) (Spanos et al., 2005a), and distributed Kalman filtering (Olfati-Saber, 2005), but this work is most closely related to linear average consensus (Xiao and Boyd, 2003; Tsitsiklis et al., 1986).

Alignment refers to all agents reaching the same value, whereas *consensus* requires agreement at a specific value and *tracking* refers consensus of a dynamic value. The desired agreement value can be any convex combination of the initial states, but is often the average of the agents' initial states. To reach agreement in a distributed manner, the approach taken frequently in the literature is for each agent, k , to maintain a state variable, x_k , which is updated when interacting with neighboring agents

$$\dot{x}_k = \sum_{j \in \mathcal{N}_k(t)} f_k(x_j, x_k). \quad (1)$$

Here, $j \in \mathcal{N}_k$ signifies that agent k has access to x_j , the state variable of agent j . Note that the neighbor set $\mathcal{N}_k$ can be time varying. When the output of the update rule, f_k , is a linear combination of the input, the resulting protocol is referred to as linear consensus.

* This work is supported by NSF grants #0347955 and #0501628.

In the case of linear consensus, the agreement protocol can be written using the Laplacian matrix, L , of the graph representing the interaction topology,

$$\dot{x} = -Lx. \quad (2)$$

When interactions are symmetric ($j \in \mathcal{N}_k(t) \leftrightarrow k \in \mathcal{N}_j(t)$), the Laplacian matrix can be computed as $L = D - A$, where D is the diagonal degree matrix and A is the adjacency matrix, (Godsil and Royle, 2001).

Consensus protocols have received a large amount of attention recently (Jadbabaie et al., 2003). Bullo et al. (2008) and Olfati-Saber et al. (2007) present a comprehensive review of recent literature. However, nearly all consensus protocols are static and are insufficient for tracking. Once consensus is reached, agents forever maintain this value. Emerging applications require that the agreement value be non-stationary. One of the few works that does address tracking is (Spanos et al., 2005b), in which the authors examine a network that splits and merges and subnetworks of agents come to consensus on their initial conditions. Additionally, Tsitsiklis et al. (1986) addresses consensus with an exogenous measurement and we use a similar framework here.

In related previous work (Shaw and Klavins, 2008), we derived a stochastic controller and estimator for robots of two types. We proved that the estimation and control processes were separately stable and demonstrable together. However, the estimator converged to the population fraction with a finite variance. In the current work, we have modified the estimator to more accurately track the population fraction and provide a proof of the estimator and controller working together.

3. PRELIMINARIES

3.1 Problem Setup and Notation

Consider a set of n robots where each robot i has a discrete internal state $q_i(t) \in \{0, 1\}$. Define

$$x \triangleq \frac{1}{n} \sum_{i=1}^n q_i(t) \quad (3)$$

to be the population fraction. The constant n is the number of robots and is known by all the robots. Each robot maintains an estimate $\hat{x}_i(t) \in \mathbb{R}$. The vector $\mathbf{q} = (q_1 \dots q_n)^T$ is defined to be the vector of internal states and $\hat{\mathbf{x}} = (\hat{x}_1 \dots \hat{x}_n)^T$ to be the vector of estimates. For simplicity, we often omit the explicit dependency of x , $\hat{x}$, and q , on t . The symbols t^- and t^+ denote the times immediately before and after robot interactions, respectively. In the sequel, the symbol $\langle \cdot \rangle$ denotes expected value and $\mathbf{1} = (1 \dots 1)^T$.

We assume that the robots are well-mixed, that is, the pair of agents that interacts at any time is uniformly distributed. Thus, in the next dt seconds, any pair of robots is equally likely to interact with probability $k dt$. When two robots interact, we assume they are programmed to simply exchange and update their estimates. Robots concurrently update their discrete states to achieve a desired population fraction. In particular, we address the following problems:

(a) Distributed Estimation

Define an estimator function

$$(\hat{x}_i(t^+), \hat{x}_j(t^+)) = f(\hat{x}_i(t^-), \hat{x}_j(t^-))$$

so that $\hat{x}_i(t)$ converges to $x(t)$ as t approaches ∞ with high probability for all robots i .

(b) Distributed Control Problem

Define a rate function $K_i(x, q_i)$ at which robot i switches from $q_i(t)$ to $1 - q_i(t)$ so that x converges to a desired reference population fraction r (a constant) with high probability and $K_i(x, q_i)$ converges to zero (the robots eventually stop switching). We also define functions to update the state and estimate for a control event:

$$(\hat{x}_i(t^+), q_i(t^+)) = g(\hat{x}_i(t^-), q_i(t^-)).$$

(c) The Simultaneous Estimation and Control Problem

Prove that a solution to the *control problem* running concurrently with a solution to the *estimation problem* (that is, with switch rate $K_i(\hat{x}_i, q_i)$) drives $\hat{x}_i$ and x to r with high probability.

In this paper, we solve these three problems formally. Our estimator performs linear average consensus, we design a controller which also updates the estimate according to a change in state at a rate K_i , and prove that the estimator and controller work together.

3.2 Notation and Basic Results

We model our system as a continuous-time Markov Process over the compact state space Z and examine its embedded discrete-time Markov chain. The set $S \subset Z$ is measurable and invariant, T_S is the hitting time of S , and $P(\cdot)$ indicates probability. We write $\sigma = z_0 z_1 z_2 \dots z_n$ as a *path* of length n , or series of n states, that begins at state z_0 and ends at z_n and $K_{z_i z_j}$ as the transition rate from state z_i to z_j .

To address problem (c) we introduce a theorem: if for all states there exists a finite path to an invariant set, then the probability that any path eventually reaches the invariant set S is 1, provided that the sum of the transition rates out of each state is bounded. The following theorem applies to discrete-time infinite-state Markov Processes.

Theorem 1. Assume $K_{z_i z_{i+1}} > \epsilon > 0$ and $\sum_{x \neq y} K_{xy} < K_{max}$. If $\forall z \in Z$ there exists a path starting at z and ending in S , whose length is $n \leq N < \infty$, then $P(T_S < \infty) = 1$.

Proof of Theorem 1. Consider a path of length N , $\sigma = z_i z_{i+1} \dots z_{i+N}$. By the assumptions of Theorem 1,

$$P(z_{i+N} \in S | z_i \notin S) \geq \delta = \left(\frac{\epsilon}{K_{max}} \right)^N. \quad (4)$$

Thus, the probability of its complement must be

$$P(z_{i+N} \notin S | z_i \notin S) \leq 1 - \delta. \quad (5)$$

We derive the probability $P(z_{i+mN} \notin S)$ as

$$P(z_{i+mN} \notin S | z_{i+(m-1)N} \notin S) P(z_{i+(m-1)N} \notin S) + P(z_{i+mN} \notin S | z_{i+(m-1)N} \in S) P(z_{i+(m-1)N} \in S), \quad (6)$$

where $P(z_{i+mN} \notin S | z_{i+(m-1)N} \in S)$ in (6) equals zero since the system cannot leave S . By iterating this substi-

tution, we arrive at the expression for $P(z_{i+mN} \notin \mathcal{S}) =$

$$\prod_{j=1}^m P(z_{i+jN} \notin \mathcal{S} | z_{i+(j-1)N} \notin \mathcal{S}) P(z_i \notin \mathcal{S}).$$

Substituting (5) above gives

$$P(z_{i+mN} \notin \mathcal{S}) < (1 - \delta)^m P(z_i \notin \mathcal{S}).$$

Our condition $P(T_S < \infty)$ is equivalent to

$$\begin{aligned} & 1 - \lim_{m \rightarrow \infty} P(T_S > mN) \\ & \geq 1 - \lim_{m \rightarrow \infty} \sum_{z \in Z} (1 - \delta)^m P(z_i = z) \\ & \geq 1 - 0 \\ & = 1. \end{aligned} \quad \blacksquare$$

4. SUMMARY OF ESTIMATOR AND CONTROLLER

4.1 Estimator

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

$$\begin{aligned} \hat{x}_i(t^+) &= a\hat{x}_i(t^-) + (1-a)\hat{x}_j(t^-) \\ \hat{x}_j(t^+) &= (1-a)\hat{x}_i(t^-) + a\hat{x}_j(t^-) \\ \hat{x}_k(t^+) &= \hat{x}_k(t^-) \text{ for all } k \neq i, j. \end{aligned} \quad (7)$$

Here $a \in (0, 1)$ is a design parameter defining the weight of a robot's own estimate. The last line of the above update rule represents robots not participating in the interaction. The interaction is written as matrices

$$\hat{x}(t^+) = A_{ij}\hat{x}(t^-), \quad (8)$$

where A_{ij} is defined for robot i and j according to equation (7).

4.2 Controller

For the controller, we define a rate K_i at which the robots change state so that (a) $x(t)$ approaches r and (b) the robots eventually stop switching. The update rule for robot i when changing state is

$$q_i(t^+) = 1 - q_i(t^-) \quad (9)$$

$$\hat{x}_i(t^+) = \hat{x}_i(t^-) + q_i(t^+) - q_i(t^-). \quad (10)$$

Equation (10) is necessary to preserve

$$\sum_i q_i = \sum_i \hat{x}_i \quad (11)$$

and allow the estimator to track the population with zero variance. Tsitsiklis et al. (1986) refer to Equation (10) (their Equation 2.1) as an exogenous measurement.

We choose a control scheme where robot i toggles its state at the rate

$$K_i \triangleq |q_i - r| |x - r|. \quad (12)$$

To show convergence of the controller, we first consider perfect knowledge of x and later, we substitute the estimate, $\hat{x}_i$, into the expression for K_i . Note that K_i is 0 when $x = r$.

4.3 Moment Equations

Our system has a deterministic update function for events that occur at random times and we can describe it us-

ing Stochastic Hybrid System (SHS) formalism. To reason about the dynamic behavior of a SHS, we use the following theorem.

Theorem 2. (Hespanha, 2006) Let $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be an arbitrary function. Then

$$\frac{d}{dt} \langle \psi(x) \rangle = \langle \mathcal{L}\psi(x) \rangle, \quad (13)$$

where $\mathcal{L}$ is the extended generator of the SHS (defined in (Hespanha, 2006)).

Our system has neither continuous flow nor time dependence, thus we do not use the entire formalism described in Hespanha (2006). In the present case, the extended generator $\mathcal{L}$ is given by

$$\mathcal{L}\psi(\hat{x}, q) = \sum_i \lambda_i (\psi(\phi_i(\hat{x}, q)) - \psi(\hat{x}, q)), \quad (14)$$

where ψ is the test function, ϕ_i is the function for a particular change of state i , and λ_i is the associated rate. To examine the expected value of the estimates, we choose $\psi(\hat{x}) = \hat{x}$ and substitute into (14) the estimate function (8) with rate k and the discrete value function (10) with rate K_i , which gives

$$\frac{d}{dt} \langle \hat{x} \rangle = k \left\langle \sum_{i < j} A_{ij}(\hat{x}) - \hat{x} \right\rangle + \left\langle \sum_{i=1}^n K_i (1 - 2q_i) \right\rangle, \quad (15)$$

where where the indices $i < j$ refer to the possible interactions between robots i and j . Since a is constant and the next reaction pair i, j is uniformly distributed over all pairs, then $\langle A_{ij}\hat{x} \rangle = A \langle \hat{x} \rangle$, where

$$A \triangleq \left\langle \sum_{i < j} A_{ij} \right\rangle = \sum_{i < j} A_{ij},$$

which can be shown to be

$$\left(-(n-1)(1-a) + \binom{n}{2} \mathbf{I} \right) + (1-a)(\mathbf{1}\mathbf{1}^T - \mathbf{I}). \quad (16)$$

Thus we derive the dynamics for the moments of the estimate and discrete value:

$$\frac{d}{dt} \langle \hat{x} \rangle = k \left(A - \binom{n}{2} \mathbf{I} \right) \langle \hat{x} \rangle + \left\langle \sum_{i=1}^n K_i (1 - 2q_i) \right\rangle \quad (17)$$

and

$$\frac{d}{dt} \langle q \rangle = \left\langle \sum_{i=1}^n K_i (1 - 2q_i) \right\rangle.$$

5. STABILITY ANALYSIS

5.1 Estimator Stability in Isolation

To examine the estimator stability, we consider the estimator without control; that is the discrete value q_i is constant for all i and $K_i = 0$. Equation (17) becomes

$$\frac{d}{dt} \langle \hat{x} \rangle = k \left(A - \binom{n}{2} \mathbf{I} \right) \langle \hat{x} \rangle.$$

Relationship to Laplacian Dynamics

Proposition 1. The dynamics of the estimator, without control, is a function of the Laplacian matrix for a complete graph.

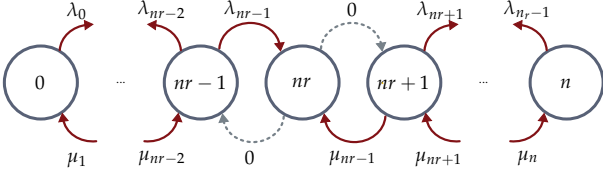


Fig. 1. Birth-death process modeling the controller. States are labeled with the number of robots with discrete state 1, or nx . Note state nr is invariant.

To show Proposition 1, it is equivalent to show that

$$H \triangleq A - \binom{n}{2} I = -wL,$$

where w is a weight and $L = (n-1)I + (\mathbb{1}\mathbb{1}^T - I)$. Using the definition for A from (16) gives

$$H = (1-a) \left(-(n-1)I + \mathbb{1}\mathbb{1}^T - I \right).$$

Thus, $H = -wL$, where $w = (1-a)$, and the results are consistent with linear average consensus algorithms, which obey (2). The union of the graphs induced by the random interactions of the robots is *jointly connected*, (Jadbabaie et al., 2003) and, in fact, complete. Thus, we observe that the estimator dynamics is governed by the Laplacian for a complete graph. ■

Because we have a complete graph, the associated Laplacian is positive semi-definite and there exists a single stable equilibrium (Olfati-Saber et al., 2007). This corresponds to the mean estimate for all agents converging to the correct population fraction x with a variance of 0, and thus all estimates $\hat{x}_i$ are equal.

5.2 Controller Stability in Isolation

To analyze the stability of the controller, we consider the control rate K_i informed by a perfect estimate of the system. Consider $nx \in \{0, 1, \dots, n\}$, representing the number of agents with state 1. Define $\mu_i dt$ to be the rate at which nx transitions from $nx = i$ to $nx = i+1$ in the next dt seconds. Similarly, let $\lambda_i dt$ to be the rate at which nx transitions from $nx = i$ to $nx = i-1$ in the next dt seconds. Then, we have a birth-death chain where

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

are rates derived using (12). To understand the expression for the (birth) rate μ_i note that when $nx = i$, there are $n-i$ robots left that can transition from 0 to 1 and they each do so at the same rate $|q_j - r||x - r| = r \left| \frac{i}{n} - r \right|$, since $q_j = 0$ and $x = \frac{i}{n}$. The expression for λ_i is similar. Note that $\mu_{nr} = \lambda_{nr} = 0$ and the state nr is invariant: all of the probability masses of nx eventually flows to nr and x approaches r . This is illustrated in Figure 1.

We now analyze the estimator and controller together.

5.3 Simultaneous Estimation and Control Stability

To combine the estimator and controller and ensure stability, we must adjust the controller. The controller stability in the previous section is dependent upon the es-

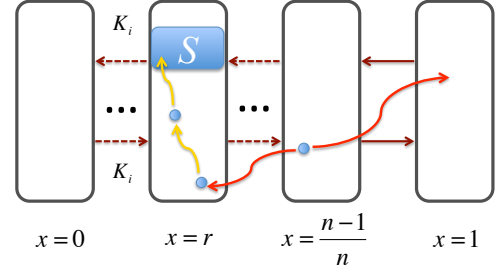


Fig. 2. Levels labeled with the population fraction x . The state $x = r$ includes the sink set S . Dashed arrows and ellipses indicate omitted states. Red arrows indicate control events and yellow arrows indicate estimation events. Curved arrows specify a trajectory and blue dots are states along this trajectory.

timates $\hat{x}_i$ being equal to the actual x . Since the space of $\hat{x}$ is continuous, this will never occur. Thus, we alter our controller so that rate K_i becomes 0 when $\hat{x} \approx r$:

$$K_i = \begin{cases} 0 & \text{if } r - \epsilon \leq x_i \leq r + \epsilon, \forall i; \\ |q_i - r||x_i - r| & \text{otherwise.} \end{cases} \quad (18)$$

Note that $\epsilon < \frac{1}{2n}$ should be chosen to be sufficiently small to preserve the conservation property (11).

We divide the state space $X \times Q$ into equivalence classes where x is constant in each class. We call these classes *levels* and define them as $\chi(x) = \{(\hat{x}, q) | \frac{1}{n} q^T \mathbb{1} = x\}$. Refer to Figure 2 for a graphical description. We prove that there is an invariant S in level $\chi(r)$, where $\frac{1}{n} q^T \mathbb{1} = r$ and $\hat{x}_i \in [r - \epsilon, r + \epsilon] \forall i$.

Using the following lemmas with proofs outlined below, we show that there exists an invariant set S , where

$$S = \{(\hat{x}, q) | r - \epsilon \leq \hat{x}_i \leq r + \epsilon, \forall i = 1, \dots, n\} \text{ for } \epsilon < \frac{1}{2n}.$$

Lemma 1. S is invariant with respect to the estimator and controller actions. In particular,

- (a) No estimation event takes an agent out of S .
- (b) No control event takes an agent out of S .

Proof of Lemma 1. For any robots i and j , estimates $\hat{x}_i$ and $\hat{x}_j$ are updated by the convex combination in (7). Without loss of generality, assume $\hat{x}_i < \hat{x}_j$. Then

$$\hat{x}_i \leq \hat{x}'_i \leq \hat{x}'_j \leq \hat{x}_j, \quad (19)$$

where $\hat{x}'_i$ and $\hat{x}'_j$ are the updated estimate values. No estimation event can take the agents' estimates outside of any level $\chi(x)$. In particular, $S \in \chi(r)$ is invariant with respect to estimation events.

Similarly, by construction, no control event can take an agent's state out of S . Refer to the definition of the modified controller (18). ■

Lemma 2. The rate $K_i = 0$ for all i only when $(\hat{x}_i, q_i) \in S$.

Proof of Lemma 2 All invariant states are in the sink set S . We prove this lemma by contradiction and show that if $\hat{x} \in [r - \epsilon, r + \epsilon] \forall i$ then $x = r$.

Assume that $\hat{x} \in [r - \epsilon, r + \epsilon] \forall i$ but $x \neq r$. Thus, the sum $\sum_i \hat{x}_i \in [n(r - \epsilon), n(r + \epsilon)]$. We note by definitions (3) and (11) that $x = \frac{1}{n} \sum_i \hat{x}_i$ and $x \in [r - \epsilon, r + \epsilon]$. The values x and r can be written as fractions: $x = \frac{j}{n}$ and $r = \frac{k}{n}$ where $j, k \in \{0, 1, \dots, n\}$. Since ϵ can be no larger than $\frac{1}{2n}$, $j \in [k - \frac{1}{2}, k + \frac{1}{2}]$. There is only one integer in that interval, so $j = k$ and $x = r$, which is a contradiction.

Thus, if $\hat{x} \in [r - \epsilon, r + \epsilon] \forall i$, then $K_i = 0 \forall i$ robots and the agents estimates are within the invariant set $\mathcal{S}$. ■

Lemma 3. $\mathcal{S}$ is reachable from any initial condition in finite time.

- (a) From any level, there is a path of m_c steps to $\chi(r)$.
- (b) From within $\chi(r)$, there is a path of m_e steps to $\mathcal{S}$.

In other words, $\mathcal{S}$ is globally attracting.

Proof of Lemma 3. Using the results of Lemmas 1 and 2 and Theorem 1 we prove Lemma 3, that our system always reaches a desired state in finite time. We use Lemma 1 to prove that the set $\mathcal{S}$ is invariant and Lemma 2 to prove that the set $\mathcal{S}$ appears in only one level.

From Theorem 1, to show that the probability that any infinite path will end up in $\mathcal{S}$ is 1, it is sufficient to construct a finite path from any state $(\hat{x}, q)$ to $\mathcal{S}$ provided bounded transition rates. First, we find a bound, K_{max} , on the sum of transition rates out of any state by summing the controller and estimator rates.

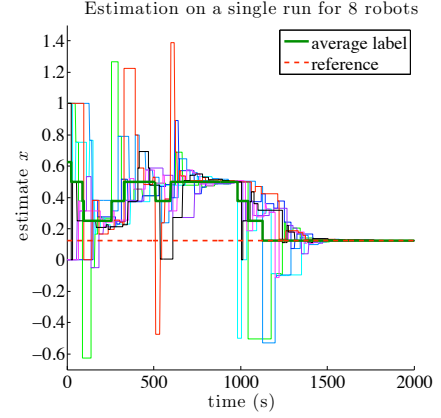
$$K_{max} > \sum_{i=1}^n |q_i - r| |\hat{x}_i - r| + \binom{n}{2} k.$$

Each control rate is bounded, that is $|q_i - r| |\hat{x}_i - r| < 2$ and the rate k is measured from the physical testbed, (Burden et al., 2006). Thus, $K_{max} = 2n + \binom{n}{2} k$.

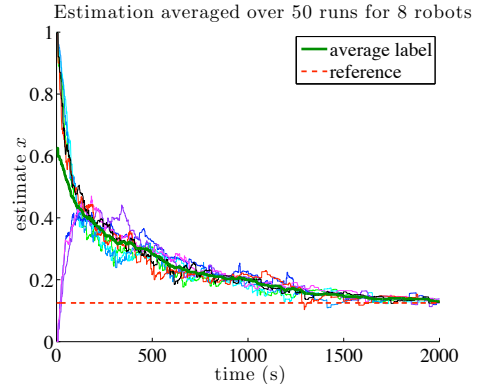
Second, we construct a path from any point in any level to the set $\mathcal{S}$ using m_c control actions and m_e estimation actions. From any level, the minimal path length to level $\chi(r)$ via control actions is $m_c = |rn - q^T \mathbf{1}|$. Similarly, from any location in $\chi(r)$, a path can be constructed in m_e steps via estimation events to $\mathcal{S}$. We choose two agents with maximal difference to update their estimates. Since $\mathcal{S}$ is measurable and estimation events are contracting (19), then estimates $\hat{x}_i$ move closer to each other and closer to r for all i . Therefore, all $\hat{x}_i$ must be within ϵ of r at a finite time thereby reaching $\mathcal{S}$ in a finite number of steps. ■

6. DEMONSTRATION OF ESTIMATOR AND CONTROLLER IN SIMULATION

We demonstrate the estimator and controller working together by directly simulating the system using the *Stochastic Simulation Algorithm*, (Gillespie, 1977). Figure 3 shows the behavior for 8 robots in a single run and averaged over 50 runs. In Figure 3(a), note that all estimates converge to the reference with zero variance. When averaged over many runs as in Figure 3(b) we see asymptotic convergence. The series of histograms in Figure 4 shows the evolution of the distribution over time. For these simulations the parameter ϵ is chosen to be 0.05 for the controller and this fulfills the constraint that $\epsilon < \frac{1}{2n}$.



(a) Single run.



(b) Fifty averaged runs.

Fig. 3. Demonstration of the estimator and controller working together. In both (a) and (b), thin lines represent data for single robots and thick green lines represent the average label. In both figures, the dashed red lines represent the reference 0.125.

7. DISCUSSION AND FUTURE WORK

In this paper we demonstrate and prove convergence for a distributed estimator and controller with zero variance. However, this algorithm is not robust to arbitrary initial conditions or any scenario where the agents' mean estimate deviates from the population fraction and violates (11). For example, if a message is dropped between two agents or if a robot turns off, the mean of the discrete values is not preserved and the system will not converge to the intended mean. We do have earlier work that can address this problem (Shaw and Klavins, 2008), but the algorithm in this previous work converges with finite variance. In the future, we will develop a notion of robustness for our system to compare algorithms.

While the example presented in this paper is a small-scale testbed with non-locomoting agents, we can apply our distributed control and estimation framework to a number of applications. For example, many task assignment algorithms require all-to-all communication, fixed graphs, or leader election. This often results in sequential task assignment, an approach that does not scale well. Since our controller employs simultaneous and dis-

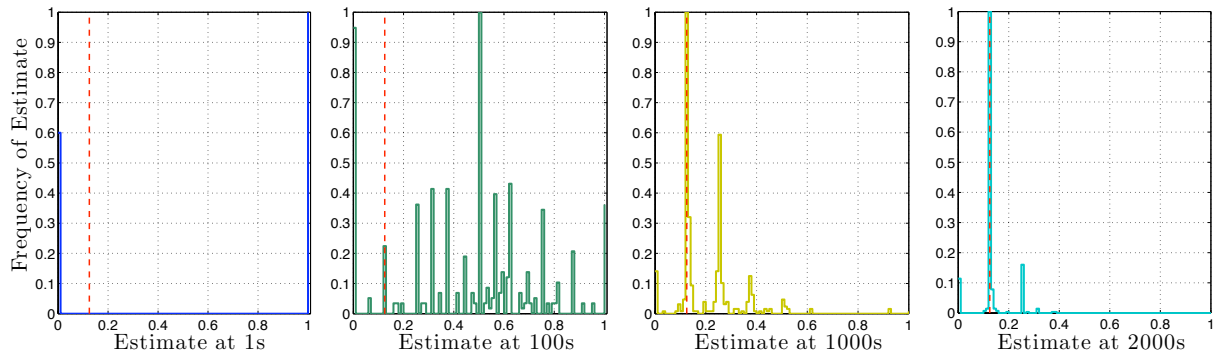


Fig. 4. Normalized histogram data at various times. The histogram at the far left shows the initial distribution of estimates where 5 of 8 robots start with discrete state 1. The histogram on the right shows the distribution of estimates at 2000s, with a spike about the reference 0.125.

tributed task assignment, it can be applicable to large-scale systems.

Furthermore, our stochastic framework can include disturbances that can be difficult to model by including agent failure at a particular rate. Thus, we can compose different processes occurring at various rates and maintain distributed, asynchronous algorithms that are based on local information exchange. These characteristics are appealing for decentralized algorithms and have implications for applications of larger scale.

REFERENCES

- Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE ACM Transactions on Networking*, 52(6), 2508–2530.
- Bullo, F., Cortes, J., and Martinez, S. (2008). *Distributed Control of Robotic Networks*. Princeton University Press, Princeton, NJ. URL <http://coordinationbook.info>. Manuscript under contract. Electronically available at <http://coordinationbook.info>.
- Burden, S., Napp, N., and Klavins, E. (2006). The statistical dynamics of programmed robotic Self-Assembly. In *Proceedings of the 2006 IEEE International Conference on Robotic Automation*, 1469–1476.
- Gillespie, D.T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*.
- Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*. Springer.
- Halasz, A., Hsieh, M., Berman, S., and Kumar, V. (2007). Dynamic redistribution of a swarm of robots among multiple sites. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2320–2325.
- Hespanha, J.P. (2006). Modeling and analysis of stochastic hybrid systems. *IEE Proceedings of Control Theory and Applications*, 153(5), 520–535.
- Jadbabaie, A., Lin, J., and Morse, A.S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48, 988–1001.
- Klavins, E. (2007). Programmable Self-Assembly. *Control Systems Magazine*, 24(4), 43–56.
- Olfati-Saber, R., Fax, J., and Murray, R. (2007). Consensus and cooperation in networked Multi-Agent systems. *Proceedings of the IEEE*, 95(1), 215–233.
- Olfati-Saber, R. (2005). Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, 8179–8184.
- Seeley (1996). *The Wisdom of the Hive*. Harvard University Press.
- Shaw, F. and Klavins, E. (2008). Distributed estimation and control in stochastically interacting robots. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 1895–1901.
- Spanos, D.P., Olfati-Saber, R., and Murray, R.M. (2005a). Distributed sensor fusion using dynamic consensus. In *Proceedings of the 16th IFAC World Congress*.
- Spanos, D.P., Olfati-Saber, R., and Murray, R.M. (2005b). Dynamic consensus for mobile networks. In *Proceedings of the 16th IFAC World Congress*.
- Tsitsiklis, J.N., Bertsekas, D.P., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9), 803–812.
- Xiao, L. and Boyd, S. (2003). Fast linear iterations for distributed averaging. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 5, 4997–5002.
- Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, 63–70.

ACKNOWLEDGEMENTS

The authors would like to thank Daniel J. Klein, Nils Napp, and David Thorsley, and Paulina Varshavskaya for many insightful discussions regarding the framework, approach, and proofs for this problem.