Synthesis and Analysis of Distributed Ensemble Control Strategies for Allocation to Multiple Tasks

T. William Mather and M. Ani Hsieh

Abstract We present the synthesis and analysis of distributed ensemble control policies to enable a team of robots to control their distribution across a collection of tasks. We assume individual robot controllers are modeled as a sequential composition of individual task controllers. A macroscopic description of the team dynamics is then used to synthesize ensemble feedback control strategies that maintain the desired distribution of robots across the tasks. We present a distributed implementation of the ensemble feedback strategy that can be implemented with minimal communication requirements. Different from existing strategies, the approach results in individual robot control policies that maintain the desired mean and the variance of the robot populations at each task. We present the stability properties of the ensemble feedback strategy, verify the feasibility of the distributed ensemble controller through high-fidelity simulations, and examine the robustness of the strategy to sensing and/or actuation failures. Specifically, we consider the case when robots are subject to estimation and navigation errors resulting from lossy inter-agent wireless communication links and localization errors.

1 Introduction

In the last ten years, there has been significant interest in applying a swarming paradigm to the control and coordination of large robot teams where individuals are programmed with simple and identical behaviors that rely solely on limited onboard computational, communication, and sensing resources. This has lead to much progress in the development of collective motion control and/or consensus forming

T.W.Mather
Drexel University, e-mail: twm32@drexel.edu

M.A.Hsieh
Drexel University, e-mail: mhsieh1@drexel.edu

strategies for homogeneous teams that often come with rigorous theoretical guarantees [2,3,15,21]. However, applications, such as automated transportation, warehouse automation, environmental monitoring, and search and rescue, invariably require the allocation of the team across various subtasks. Existing swarm-inspired paradigms fail to address the effects caused by splitting the team across multiple tasks which inevitably leads to relatively small numbers of locally collaborating robots. The presence of these smaller scales motivates the need for multi-robot style techniques that remain amenable to whole-team analysis.

In this work, we address the dynamic allocation of a team of robots to a collection of spatially distributed tasks. This is similar to the multi-task (MT), single-robots (SR), time-extended assignment (TA) problem [7]. In the multi-robot domain, market-based approaches [5,6] have been successful and can be further improved when learning is incorporated [4]. Nevertheless, these methods often scale poorly in terms of team size and number of tasks [5,10]. Furthermore, in applications where inter-agent wireless communication is limited, *e.g.*, underwater applications, it is often difficult to devise reliable strategies to ensure timely communication of the various local costs and utilities required by these existing allocation approaches. Different from these existing works, we present a strategy where an appropriate macroscopic description of the ensemble dynamics is used to simultaneously address the allocation and controller synthesis problems for the team.

While macroscopic continuous models have been employed to model the ensemble dynamics of robotic self-assembly [13, 22] and robotic swarm systems [1, 14, 17, 18], existing strategies mostly focus on developing macroscopic models to analyze the effects of microscopic, or agent-level, behaviors on the overall performance of the team. In these works, the continuous population models are obtained by representing the individual robot controllers as probabilistic finite state machines and approximating the dynamics of the collective as a continuous-time Markov process. More recently, we have employed a macroscopic description of the ensemble dynamics to synthesize distributed agent-level control policies to dynamically allocate the team to the various tasks [19]. Different from existing strategies, this approach enables the team to actively shape both the mean and variance of the distribution of robots across the different tasks [19] and can be further extended to incorporate the dynamics of the communication network [20]. Since the complexity of the proposed strategy only depends on the number of tasks at runtime, the resulting decentralized robot controllers become invariant to team size.

Building on the multi-site allocation problem, first presented in [11], and our recent result [19], we consider the allocation problem in the presence of communication and sensing errors. We limit our discussions to homogeneous teams of robots[1] and represent the agent-level controller as a sequential composition of individual task controllers. Similar to [1,14], we design stochastic transition rules to enable the team to autonomously achieve a desired distribution across the various tasks. Different from [1,14], the proposed agent-level control policies set both the mean and the variance of the robot populations at each tasks [19]. Inspired by Klavins's work on

[1] For extensions to heterogeneous teams of robots, we refer the interested reader to [20].

controlling gene population dynamics [16], we model the simultaneous execution of spatially distributed tasks by a team of robots as a polynomial Stochastic Hybrid System (pSHS) and employ moment closure techniques to model the moment dynamics of ensemble distribution [12]. We analyze the stability of the resulting strategies and the robustness of ensemble strategy in the presence of communication and sensing errors. The main contribution is a team-size invariant approach towards the design of distributed agent-level control policies that can affect both the mean and the variance of the robot ensemble distribution. Since the strategies are stochastic, we show that they are also robust to communication and sensing errors. The result is a distributed allocation strategy that has the ability to respond to robot failures in a natural way, ensuring graceful degradation.

The paper is structured as follows: We formulate our approach in Section 2 and describe the synthesis and analysis of our ensemble feedback strategy in Section 3. Section 4 describes the distributed implementation of our macroscopically derived feedback controller in the presence of full communication and when robots rely solely on local communication. Section 5 presents our simulation results. We conclude with a discussion of our results and directions for future work in Sections 6 and 7 respectively.

2 Problem Formulation

Consider the assignment of N robots to execute M tasks each located at a different locale within the workspace. The objective is to synthesize decentralized single robot controllers that enable the team to autonomously distribute across the M tasks and maintain the desired allocation at the various locales. Different from [1, 14], the goal is to design distributed agent-level control policies that can affect the means and the variances of the robot populations at each task/site.

2.1 Individual Robot Controller

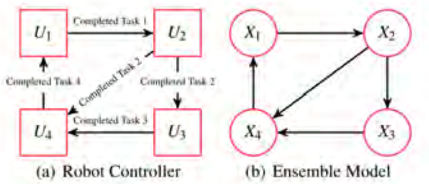
Given a collection of $\{1,\ldots,M\}$ tasks/sites, we use a directed graph, $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, to model the pairwise precedence constraints between the tasks. Each task is represented by a vertex in $\mathscr{V} = \{1,\ldots,M\}$. A directed edge exists between two vertices $(i, j) \in \mathscr{V} \times \mathscr{V}$ if task i immediately precedes task j and we denote this relation as $i \prec j$. Then, the set of edges, $\mathscr{E}$, is given by $\mathscr{E} = \{\forall (i, j) \in \mathscr{V} \times \mathscr{V} | i \prec j\}$. We assume $\mathscr{G}$ is a strongly connected graph, *i.e.*, a directed path exists for any $i, j \in \mathscr{V}$.[2]

Given the M tasks, we denote the set of task controllers for each task as $\{U_1,\ldots,U_M\}$ and assume that the single robot controller is obtained through the

[2] We assume the relation $\prec$ is not transitive so as to allow for cycles to exist in $\mathscr{G}$ such that $\mathscr{G}$ does not have to be a complete graph.

sequential composition of $\{U_1, \ldots, U_M\}$ such that the precedence constraints specified by $\mathscr{G}$ are satisfied. We represent the robot controller as a finite state automaton where each automaton state i is associated with a task controller U_i. Figure 1 shows an example robot controller where the arrows denote state transitions that satisfy the constraints specified in $\mathscr{G}$.

In this work, we consider the surveillance of M sites where the team must maintain some desired allocation of the robots across the various locations. Robots are tasked to monitor each site for a pre-specified amount of time, *i.e.*, execute U_i. Once the task is completed, robots must navigate to the next adjacent site based on the constraints encoded in $\mathscr{G}$. As such, we assume each robot has complete knowledge of $\mathscr{G}$, the ability to localize within the workspace, and is capable of navigating from one task/site to another while avoiding collisions with other robots in the workspace.



(a) Robot Controller

(b) Ensemble Model

Fig. 1 (a) The robot controller. The robot changes controller states based on guard conditions. (b) Graphical representation of the equivalent chemical reaction network for a robot ensemble executing the tasks.

2.2 The Ensemble Model

For a team of N robots, each executing the same sequentially composed controller, *e.g.*, the one in Figure 1, the ensemble dynamics can be represented by an equivalent chemical reaction network. This abstraction allows us to model the multi-task/site allocation as a *polynomial stochastic hybrid system* (pSHS) and to use moment closure techniques to model the time evolution of the distribution of the team across the various tasks/sites.

Let $X_i(t)$ denote the number of robots executing task i or located at site i. In this work, task execution happens at the site where the task is located and as such we will not distinguish between robots executing task i and robots located at the site where task i is located, *i.e.*, site i. Let $\bar{X}_i$ denote the desired number of robots for task i. The system state is given by $\mathbf{X}(t) = [X_1(t), \ldots, X_M(t)]^T$ with the desired distribution of the ensemble given by $\bar{\mathbf{X}} = [\bar{X}_1, \ldots, \bar{X}_M]^T$. Since the tasks are spatially distributed and robots must navigate from one site to another while avoiding collisions with other robots, we model the variability in robot arrival times at each task using *transition rates*. For every edge $(i, j) \in \mathscr{E}$, we assign constant $k_{ij} > 0$ such that

k_{ij} defines the transition probability per unit time for one agent from site i to go to site j.

Given $\mathscr{G}$ and the set of k_{ij}'s, we model the ensemble dynamics as a set of chemical reactions of the form:

$$X_i \xrightarrow{k_{ij}} X_j \quad \forall\,(i,j) \in \mathscr{E}. \tag{1}$$

The above reaction represents a stochastic transition rule with k_{ij} as the per unit reaction rate and $X_i(t)$ and $X_j(t)$ as discrete random variables. In the robotics setting, equation (1) implies that robots at site i will transition to site j with a rate of $k_{ij}X_i$. Further, we assume the ensemble dynamics is Markov which allow us to model the moment dynamics of the distribution as a set of linear differential equations. It is important to note that in general $k_{ij} \neq k_{ji}$ and k_{ij} encodes the inverse of the average time a robot spends at task/site i.

2.2.1 Moment Closure

Given the set of reactions in (1), the moment equations for the discrete random variable X_i is given by the extended generator of the system [12]. For a real-valued function $\psi(X_i)$, the extended generator is an expression for the time derivative of the expected value of ψ, *i.e.*, $\frac{d}{dt}\mathrm{E}[\psi(X_i)] = \mathrm{E}[L\psi(X_i)]$, and takes the form

$$L\psi(X_i) = \sum_j \left[(\psi(X_i - 1) - \psi(X_i))w_{ij} + (\psi(X_i + 1) - \psi(X_i))w_{ji}\right]. \tag{2}$$

The right hand side of (2) gives the continuous time derivatives of the system for a discrete change in the state X_i. The expression $[\psi(X_i - 1) - \psi(X_i)]$ represents the change in ψ given a unit change in the discrete variable X_i, while w_{ij} represents the rate at which the change occurs. For the system given by (1), $w_{ij} = k_{ij}X_i$. To obtain the rate of the change of the expected value of X_i, $\frac{d}{dt}\mathrm{E}[X_i]$, we let $\psi(X_i) = X_i$ in (2). Similarly, to obtain $\frac{d}{dt}\mathrm{E}[X_i^2]$, we let $\psi(X_i) = X_i^2$.

Example 1. Consider the case when $M = 2$, *i.e.*,

$$X_1 \underset{k_{21}}{\overset{k_{12}}{\rightleftharpoons}} X_2$$

where robots executing task 1, X_1, transition to task 2, X_2, and vice versa with rates $k_{12}X_1$ and $k_{21}X_2$ respectively. The first and second moment dynamics for X_1 are given by

$$
\begin{aligned}
\tfrac{d}{dt}\mathrm{E}[X_1] &= \mathrm{E}\left[((X_1+1)-X_1)k_{21}X_2 + ((X_1-1)-X_1)k_{12}X_1\right] \\
&= k_{21}\mathrm{E}[X_2] - k_{12}\mathrm{E}[X_1] \quad \text{and} \\
\tfrac{d}{dt}\mathrm{E}[X_1^2] &= \mathrm{E}\left[((X_1+1)^2-X_1^2)k_{21}X_2 + ((X_1-1)^2-X_1^2)k_{12}X_1\right] \\
&= -2k_{12}\mathrm{E}[X_1^2] + 2k_{21}\mathrm{E}[X_1X_2] + k_{21}\mathrm{E}[X_2] + k_{12}\mathrm{E}[X_1].
\end{aligned}
$$

When the w_{ij}'s are linear with respect to the system state $\mathbf{X}$, the moment equations are closed. This means that the time derivative for the first moment of X_i, $\frac{d}{dt}\mathrm{E}[X_i]$, is only dependent on the first moments of X_i for $i = 1,\ldots,M$, *i.e.*, $\mathrm{E}[X_1],\ldots,\mathrm{E}[X_M]$, the second moments are dependent on the first and second moments, and so on and so forth. This is important because when the moment equations are closed, the moment dynamics can be expressed as a linear matrix equation as follows:

$$
\frac{d}{dt}\begin{bmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \\ \mathrm{E}[X_1X_1] \\ \mathrm{E}[X_2X_2] \\ \mathrm{E}[X_1X_2] \end{bmatrix} = \begin{bmatrix} -k_{12} & k_{21} & 0 & 0 & 0 \\ k_{12} & -k_{21} & 0 & 0 & 0 \\ k_{12} & k_{21} & -2k_{12} & 0 & 2k_{21} \\ k_{12} & k_{21} & 0 & -2k_{21} & 2k_{12} \\ -k_{12} & -k_{21} & k_{12} & k_{21} & -k_{21}-k_{12} \end{bmatrix} \begin{bmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \\ \mathrm{E}[X_1X_1] \\ \mathrm{E}[X_2X_2] \\ \mathrm{E}[X_1X_2] \end{bmatrix}. \tag{3}
$$

The steady state solution to the above equation lies in the nullspace of the coefficient matrix. The solution is unique if we consider the conservation constraint $X_1 + X_2 = N$, *i.e.*, the number of robots within the system is constant. For this example, the steady state solution is a binomial distribution with the probability for robots to be at site 1 (or executing task 1) given by $p_1 = k_{21}(k_{12}+k_{21})^{-1}$, and the mean and variance given by, $\mathrm{E}[X] = Np_1$ and $\mathrm{E}[(X-\mathrm{E}[X])^2] = Np_1(1-p_1)$ respectively.

In general, the ensemble moment dynamics for the system with M tasks/sites is given

$$
\begin{aligned}
\tfrac{d}{dt}\mathrm{E}[X] &= \mathbf{K}\mathrm{E}[X] \\
\tfrac{d}{dt}\mathrm{E}[XX^T] &= \mathbf{K}\mathrm{E}[XX^T] + \mathrm{E}[XX^T]\mathbf{K}^T + \Gamma(\alpha,\mathrm{E}[X])
\end{aligned} \tag{4}
$$

where $[\mathbf{K}]_{ij} = k_{ji}$ and $[\mathbf{K}]_{ii} = -\sum_{(i,j)\in\mathscr{E}} k_{ij}$. It is important to note that $\mathbf{K}$ is a Markov process matrix and thus is negative semidefinite. This coupled with the conservation constraint $\sum_i X_i = N$ leads to exponentially stability of the system given by (4) [11, 16]. Each entry in the matrix of second moments is determined from the moment closure methods shown above where the entries of $\Gamma(\alpha,\mathrm{E}[X])$ are all linear with respect to the k_{ij}'s and the means $\mathrm{E}[X]$. For the two state example given by equation (3), $\Gamma(\alpha,\mathrm{E}[X])$ is defined as

$$
\Gamma(\alpha,\mathrm{E}[X]) = \begin{bmatrix} k_{12}\mathrm{E}[X_1]+k_{21}\mathrm{E}[X_2] & -k_{12}\mathrm{E}[X_1]-k_{21}\mathrm{E}[X_2] \\ -k_{12}\mathrm{E}[X_1]-k_{21}\mathrm{E}[X_2] & k_{12}\mathrm{E}[X_1]+k_{21}\mathrm{E}[X_2] \end{bmatrix}.
$$

Furthermore, the k_{ij}'s can be chosen to enable a team of robots to autonomously maintain some desired mean steady-state distribution of the team across the various tasks/sites [1, 11, 14]. In essence, the k_{ij}'s translate into a set of stochastic guard conditions for the single robot controllers. The result is a set of decentralized agent-level control policies that allow the team to maintain the steady-state mean of the ensemble distribution. Different from previous work, the focus of this paper is to use the ensemble moment dynamics to synthesize distributed control strategies to enable the team to maintain *both the mean and the variance* of the robot team distribution across the various tasks/sites. We describe the approach in the following sections.

3 Ensemble Controller Design

As shown with equation (1), the rate in which agents in state X_i transition to X_j depends on the population in state X_i. As such, the more agents in state X_i, the faster they transition to X_j. However, Klavins recently showed that if we allow for both positive and negative transition rates, it is possible to shape both the mean and the variance of the ensemble distribution [16]. In other words, by introducing a negative feedback rate, it is possible to slow the population growth at a given state and thus affect the population variance in that state.

3.1 Controller Synthesis

Consider the following single reaction $X_1 \xrightarrow{\alpha_{12}} X_2$ with the corresponding moment equation for X_1 given by $\frac{d}{dt}\mathrm{E}[X_1] = -\alpha_{12}\mathrm{E}[X_1]$. If we add a negative feedback of the form $u = -\beta X_2$ such that the "closed-loop" reaction becomes $X_1 \xrightarrow{\alpha_{12}-\beta X_2} X_2$, then the moment dynamics with state feedback is given by $\mathrm{E}[X_1] = -\alpha_{12}\mathrm{E}[X_1] + \beta\mathrm{E}[X_1X_2]$ will depend on the covariant moment $\mathrm{E}[X_1X_2]$. Such a feedback control law breaks the linearity of the moment equations because the moment dynamics are not closed.

To ensure that the moment dynamics with state feedback remain closed, consider the following feedback controller

$$u = \beta\frac{X_2}{X_1}. \tag{5}$$

The reaction with state feedback becomes $X_1 \xrightarrow{\alpha_{12}-\beta\frac{X_2}{X_1}} X_2$ where (5) can be seen as a form of linearizing feedback control that inhibits transitions from X_1 to X_2 as X_2 becomes larger than X_1. For the two state system described in Example 1, the closed-loop reactions become

$$X_1 \xrightleftharpoons[\alpha_{21}-\beta_{21}\frac{X_1}{X_2}]{\alpha_{12}-\beta_{12}\frac{X_2}{X_1}} X_2 \tag{6}$$

and the corresponding closed-loop first and second moment dynamics for X_1 become

$$\begin{aligned}\tfrac{d}{dt}\mathrm{E}[X_1] =&(\alpha_{21}+\beta_{12})\mathrm{E}[X_2]-(\alpha_{12}+\beta_{21})\mathrm{E}[X_1],\\ \tfrac{d}{dt}\mathrm{E}[X_1^2] =&(\alpha_{21}-\beta_{12})\mathrm{E}[X_2]+(\alpha_{12}-\beta_{21})\mathrm{E}[X_1]\\ &+2(\alpha_{21}+\beta_{21})\mathrm{E}[X_1X_2]-2(\alpha_{12}+\beta_{12})\mathrm{E}[X_1^2].\end{aligned} \tag{7}$$

where the steady-state values of $\mathrm{E}[X_1]$ and $\mathrm{E}[X_1^2]$ can be independently set by adjusting parameters α and β.

In general, for the M state system described by (4), we propose the following ensemble feedback controller

$$\mathbf{u}=-\mathbf{K}_\beta\mathrm{E}[X] \qquad \mathbf{K}_\beta^{ij}=\begin{cases}\beta_{ji} & \forall(i,j)\in\mathscr{E}\\ -\sum_{(i,j)\in\mathscr{E}}\beta_{ji} & \forall i=j\\ 0 & \text{otherwise}\end{cases}, \tag{8}$$

resulting with the following closed-loop moment dynamics

$$\begin{aligned}\tfrac{d}{dt}\mathrm{E}[X] =&(\mathbf{K}_\alpha+\mathbf{K}_\beta)\mathrm{E}[X]\\ \tfrac{d}{dt}\mathrm{E}[XX^T] =&(\mathbf{K}_\alpha+\mathbf{K}_\beta)\mathrm{E}[XX^T]+\mathrm{E}[XX^T](\mathbf{K}_\alpha+\mathbf{K}_\beta)^T+\Gamma(\alpha,\beta,\mathrm{E}[X]).\end{aligned} \tag{9}$$

The above equations are obtained by simply substituting $k_{ij}=\alpha_{ij}-\beta_{ij}\frac{X_j}{X_i}$ in the reactions given by (1) and applying the extended generator to $\psi(X_i)=X_i$.

3.2 Analysis

In this section, we show the stability of the ensemble feedback controller.

Theorem 1. *The dynamics of the first moment of the system with ensemble feedback strategy given by* (9) *is stable.*

Proof. The first moment dynamics for the system with ensemble feedback is given by (9)

$$\tfrac{d}{dt}\mathrm{E}[X]=\mathbf{K}_\alpha\mathrm{E}[X]+\mathbf{K}_\beta\mathrm{E}[X]$$

.

Since both $\mathbf{K}_\alpha$ and $\mathbf{K}_\beta$ are Markov process matrices, they are negative semidefinite, each with a zero eigenvalue of multiplicity one. Furthermore, the eigenvector associated with the zero eigenvalue is the vector $\mathbf{1}$ such that $\mathbf{1}^T\mathbf{K}_{(\cdot)}=0$. The sum of two negative semidefinite matrices is still negative semidefinite and thus the first moment dynamics of the closed-loop system given by (9) is stable.

It is important to note that the rate of population exchange in the model allows for backwards flow when $\beta_{ij}X_j>\alpha_{ij}X_i$. In the systems that we are considering, we restrict this rate to be greater than or equal to zero. A rate of zero implies that no robots are executing that transition. This case requires a bit more work to show stability of the ensemble feedback strategy if only due to the saturation of the control inputs required to ensure $\beta_{ij}X_j \leq \alpha_{ij}X_i$.

4 Distributed Implementation

In this section we present the distributed implementation of the proposed ensemble feedback strategy given by equation (8) in the cases when robots have 1) full and infinite range communication and 2) local and finite range communication.

4.1 Full Communication

The feedback strategy (8) gives robots in state X_i the ability to set their own state transition rates to be independent from the number of robots in X_i. This, however, requires robots at task i to know how many robots are at adjacent sites, *i.e.*, robots in X_j where $(i,j) \in \mathscr{E}$. We begin with the assumption that individual robots have full knowledge of the ensemble states $\mathbf{X}(t) = [X_1(t), \ldots, X_M(t)]$. In practice, this can be achieved by endowing each task site the ability to track the number of robots at the site and the ability to communicate with adjacent task sites. To obtain timely estimates of the ensemble states, robots would only need to communicate with their current task site.

Algorithm 1 describes the implementation of the individual robot controller shown in Figure 1 where $\mathscr{U}(0,1)$ denotes a continuous uniform distribution between 0 and 1. The algorithm consists of two parts. In Lines 1-14, the robot executes task i and determines the time it stays with task i, T_i, from the set of k_{ij}'s. In Lines 16-19, the robot determines the next task to execute after completing task i. Lines 16-19 of Algorithm 1 is necessary for the general case when task i can be followed by multiple tasks. To determine T_i, robots first calculate the sum of all the exit rates, $\sum_{\forall i} k_{ij}$, from i (Line 3). This total rate represents the rate a single robot leaves task i for j and is used to calculate the time a robot spends executing task i, *i.e.*, T_i. If a new robot arrives at task/site i or an existing one leaves, the robot recalculates T_i. The feedback controller given by Equation (8) is implemented by assigning $k_{ij} = \alpha_{ij} - \beta_{ij}\frac{X_j}{X_i}$ in Line 3 for all i,j pairs and X_i, X_j is obtained through local communication between the robot and site i.

We note that from the ensemble view, the parallel execution of Algorithm 1 by the team of N robots is akin to the parallel execution of N stochastic simulation algorithms first proposed by Gillespie in [8]. Since the times a robot spends at a task/site are all exponentially distributed and memoryless, the process of gaining and losing robots at the site is a birth-death process which allows for the constant resetting of T_i.

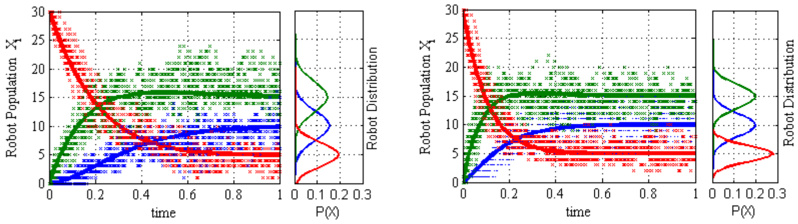
Algorithm 1 Robot Controller w/ Full Communication

```
 1: initialize $taskController = U_i$
 2: initialize all $\alpha_{ij}$s and $\beta_{ij}$s
 3: $k_{\Sigma i} = \sum_{\forall (i,j) \in \mathscr{E}} k_{ij}$
 4: $T_i = \sum_{\forall \{i,j\} \in \mathscr{E}} \frac{1}{k_{\Sigma i}} \ln \frac{1}{\mathscr{U}(0,1)}$
 5: **repeat**
 6:     execute $taskController$
 7:     **if** a new robot arrives at current site $i$ **then**
 8:         $X_i = X_i + 1$
 9:         goto 1
10:     **end if**
11:     **if** a robot leaves current site $i$ **then**
12:         $X_i = X_i - 1$
13:         goto 1
14:     **end if**
15: **until** $t \leq T_i$
16:
17: j = sample$[\frac{k_{i1}}{\sum_{\forall (i,j) \in \mathscr{E}} k_{ij}}, \cdots, \frac{k_{ij}}{\sum_{\forall (i,j) \in \mathscr{E}} k_{ij}}, \cdots]$
18: $taskController = U_j$
19: $X_j = X_j + 1$
20: $i = j$
21: goto 3
```

Example 2. Consider the example when $M = 3$. The equivalent chemical reaction network representation of the ensemble model is shown in Figure 7 in the Appendix. The full equation of the closed-loop first and second moment dynamics are is given by Equation (11), also in the Appendix. The values for the α_{ij}'s and β_{ij}'s were chosen to achieve the desired population distribution mean of $\mathrm{E}[X] = [10, 15, 5]$ and variance of $Var[X] = [4, 4, 2]$ at each site. These values are shown in Table 4 in the Appendix. By adjusting the ratios of the α_{ij}'s and β_{ij}'s, we were able to maintain the same mean behavior while reducing the on-site population variance and simultaneously speeding up the closed-loop system's convergence rate. Figure 2 shows the distribution of the ensemble at each of the three sites with and without the ensemble feedback strategy given by Equation (8).

4.2 Local Communication

In practice, is it often unreasonable to assume global communication among the robots, especially when tasks are distributed across vast geographic regions or in located in environments where long-range communication is difficult/impossible, *e.g.*, underground/underwater environments. In this section, we present a decentralized implementation of the proposed ensemble feedback control strategy given by equation (8) that relies solely on local inter-robot communication. We assume robots have finite communication ranges and can only communicate with other robots that



(a)

(b)

Fig. 2 Results of the system in Figure 7. These plots compare the steady state distributions and the convergence rate of the system with and without ensemble feedback. Each left side plot shows the transient behavior from an initial condition of $X = [0,0,30]$. The solid lines denote the numerical solutions of the first moment dynamics and the data points are 10 representative stochastic simulation runs. The right side plots are the steady state distributions represented as Gaussians. Note how the system with ensemble feedback has faster convergence and smaller variance on its populations.

are co-located at the same site and/or within each other's communication range. As robots move from one site to another and exchange information with other robots they encounter, each robot can construct their own estimates of the population levels at the various sites.

To more faithfully represent the underlying agent-based system, we expand the ensemble model to take into account the navigation controller executed by individual robots as they move from one task/site to another. Given the set of M tasks and task controllers $\{U_1,\ldots,U_M\}$, let U_{ij} denote the navigation controller executed by a robot to travel from site i to j. Let λ_{ij} denote the mean per robot arrival rates at site j for robots traveling from sites i. In other words, $1/\lambda_{ij}$ denote the expected travel time between sites i and j whose variability can be affected by the number of robots "on the road". Let $Y_{ij}(t)$ denote the number of robots traveling between sites i and j. Similar to the $X_i(t)$ variables which denote the number of robots at site i, $Y_{ij}(t)$ are discrete random variables.

Applying (2) with $\psi(X_i) = X_i$, we obtain the following first moment dynamics

$$\begin{aligned}\tfrac{d}{dt}\mathrm{E}[X_i] &= \sum_{\forall e_{ji}\in\mathscr{E}} \lambda_{ji}\mathrm{E}[Y_{ji}] - \sum_{\forall e_{ij}\in\mathscr{E}} k_{ij}\mathrm{E}[X_i],\\ \tfrac{d}{dt}\mathrm{E}[Y_{ij}] &= k_{ij}\mathrm{E}[X_i] - \lambda_{ij}\mathrm{E}[Y_{ij}]\end{aligned} \tag{10}$$

for all $i,j = 1,\ldots,M$. Due to space considerations, we omit the equations for the second moment dynamics. Similar to previous examples, we employ the feedback control strategy given by (8). The moment dynamics with state feedback can be obtained by substituting $k_{ij} = \alpha_{ij} - \beta_{ij}X_jX_i^{-1}$ into the above equations.

To achieve online estimation of the on-site robot population as well as the robot population at neighboring sites, robots exchange information with other robots that are co-located at the same site. Robots arriving from site i from site j delivers an estimate of X_j which we denote as $\hat{X}_{j|i}$ and refer to it as the estimated value of X_j

at task *i*. Since the proposed ensemble feedback strategy relies on information of robot populations at adjacent sites, the decentralized implementation requires two way reactions between sites to achieve variance control.

Algorithm 2 presents the decentralized single robot controller with ensemble feedback. The significant change in the decentralized scheme is the updating of transition rates in a discrete fashion that allows for the state dependent rates to be maintained. Each robot sets its own timer by randomly sampling an exponential distribution. This allows each robot to actively reset its timer while not effecting the expected transition time of the robot. This is imperative because the transition times must be updated for all changes in the current population and the population estimates.

Algorithm 2 Robot Controller w/ Local Communication

```
Initialize Robot Controller: $taskController = U_i$
loop
    execute $taskController$
    while $taskController == U_{ji}$ do
        if $q \in B(i)$ then
            $taskController = U_i$
            Send estimate $\{\hat{X}_j\}$ to other robots executing $i$
            $T = setExponentialTimer(\alpha, \beta, X_i, \hat{X}_{j|i})$
        end if
    end while
    while $taskController == U_i$ do
        if $\Delta X_i \neq 0$ then
            $\hat{X}_i = \hat{X}_i + \Delta X_i$
            $T = setExponentialTimer(\alpha, \beta, X_i, \hat{X}_{j|i})$
        end if
        if $T == 0$ then
            $\{X_i\}_k = X_i - 1$
            $j = \text{sample}[\frac{k_{i1}}{\sum_{\forall(i,j)\in\mathscr{E}} k_{ij}}, \cdots, \frac{k_{ij}}{\sum_{\forall(i,j)\in\mathscr{E}} k_{ij}}, \cdots]$
            $taskController = U_{ij}$
        end if
    end while
end loop
```

4.3 In the Presence of Errors

In this work, we consider the performance of our proposed distributed ensemble strategy in the presence of sensing and communication failures. Specifically, we consider two types of errors: bad estimates of ensemble variables due to communication failures and sensing errors that result in localization errors leading to erroneous navigation strategies.

To simulate communication failures, we included a probabilistic rate at which the robots exchange information with each other when co-located at the tasks. Recall that robots must know the estimates of the robot populations at its current task, $\hat{X}_i$, and at neighboring sites, $\hat{X}_{j|i}$, before deciding to proceed to the next task since $k_{ij} = \alpha_{ij} - \beta ij \hat{X}_{j|i} X_i^{-1}$. We note that in the previous section, we assume robots have perfect information of the number of robots co-located at the same task. As such, we did not distinguish between X_i and $\hat{X}_i$. Here, we consider the impact of missed updates on X_i on the overall performance and thus denote a robot's estimate of X_i as $\hat{X}_i$. From Algorithm 2, $\hat{X}_i$ is updated every time a robot arrives and leaves the site, while $\hat{X}_{j|i}$ is only updated when a new robot arrives at the site. Two types of communication errors are considered: failure to receive updates on $\hat{X}_i$ and $\hat{X}_{j|i}$. In one scenario, robots probabilistically fail to receive updates on $\hat{X}_i$ and in another, robots probabilistically fail to receive updates on $\hat{X}_{j|i}$.

To simulate localization errors where individual robots fail to reach the desired task locations, "lost" robots are tasked to navigate to a "dummy" holding site that is separate from the task locations. Every time a robot leaves a site for a new task, it has a non-zero probability of arriving at the dummy location. These "lost" robots are eventually stochastically reintroduced back into the system by returning to one of the active task sites. Upon arrival, these robots will communicate an out-of-date estimate of $\hat{X}_j$ or a zero value. This is to simulate localization errors that can result in prolonged navigation times and/or localization errors that result in erroneous estimates of the population at another site because the robot believed it was at a given site when it was effectively lost.

The results for the following scenarios are presented in the next section: (I) the system with no ensemble feedback, (II) robots executing Algorithm 1, (III) robots executing Algorithm 2, (IV) robots executing Algorithm 2 with communication errors, and (V) robots executing Algorithm 2 with navigation errors.

5 Results

To verify the validity of our robot controller presented in Algorithm 2, we employed a multi-level simulation strategy. At the top level are the *macro-continuous* simulations where the linear moment closure equations are numerically solved. At the intermediate level are the *macro-discrete* simulations which are conducted using the Stochastic Simulation Algorithm (SSA) which is mathematically equivalent to an agent-based simulation [9]. At the lowest level are the *micro-discrete* simulations which are agent-based simulations using a team of mSRV-1 robots in USARSim [23].

Simulations at all three levels were run for a team of 30 robots for the four site example shown in Figure 3. The macro-discrete simulations were ran for approximately $30,000$ transitions to ensure the system has reached steady-state for the following Scenarios: (I) the system with no ensemble feedback, (II) robots executing Algorithm 1, (III) robots executing Algorithm 2, (IV) robots executing Algorithm 2 with communication errors, and (V) robots executing Algorithm 2 with navigation errors. For Scenario (IV), we consider the case where communication failures results in incorrect estimates of the robot population at neighboring or off-site tasks only (IVa) and incorrect estimates of the on-site robot populations only (IVb). Similarly, for Scenario (V), we consider the case where localization/navigation errors lead to robots passing on out-of-date information (Va) or robots passing on a population estimate of 0 robots at a given task (Vb). A snapshot of our USARSim micro-discrete simulations for a 4-site surveillance task is shown in Figure 4.

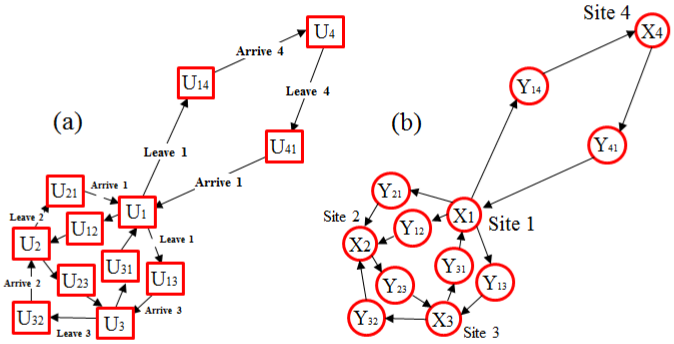


Fig. 3 (a) Robot controller for the 4-task problem with navigation. U_i denotes task controllers at site i and U_{ij} denotes navigation controllers between sites i and j. (b) The corresponding ensemble model.

The macro-continuous, macro-discrete, and micro-discrete results for Scenario (I) are shown in Figures 5-6. Since the macro-continuous and macro-discrete population distributions always show good correspondence in all scenarios, we only show macro-continuous and micro-discrete (agent-base) solutions for the remaining scenarios. These are shown in Figures 8-13 located in the Appendix. In these figures, the macro-continuous results are represented as normal distributions and shown by the dotted lines.

The values for α_{ij}'s and β_{ij}'s in the ensemble feedback strategy are shown in Table 5 and were chosen such that $\mathrm{E}[X] = (6,6,6,4)$ for each site while reducing $Var[X_i]$. The values for the expected travel times are shown in Table 6. The travel times are found by running initial experiments and solving for what the associated road rates would have to be to achieve the associated distribution. This is then compared with the actual average travel time of the robots. These tables are located in the Appendix.

The first and second moments of our results for Scenarios (I)-(III) are summarized in Table 1. Each row of data within the table corresponds to a full micro-discrete simulation that lasted approximately 5 hours or 5000 state transitions. Additionally, the results for Scenarios (IVa) and (IVb) are summarized in Tables 2 and 2 respectively. Finally, Table 3 summarizes the mean and variance of the robot populations for Scenario (Va) and (Vb). Additional results showing the probability distributions for the various scenarios are shown in the Appendix.

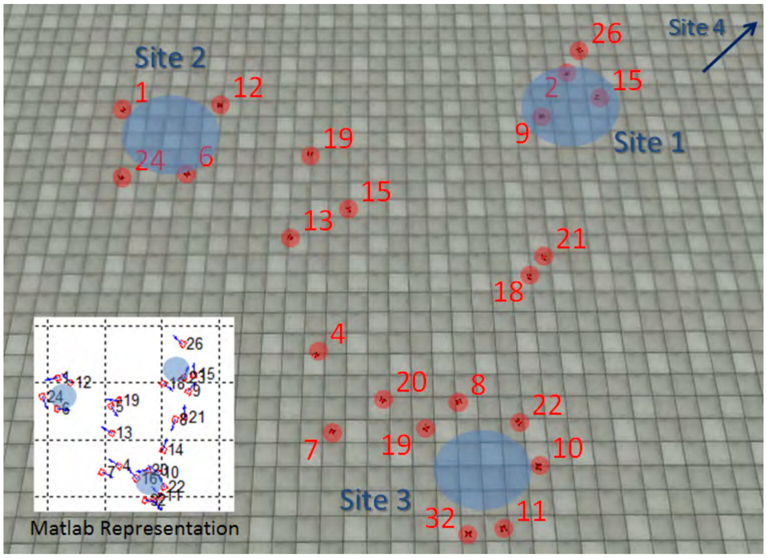


Fig. 4 A still of the 30 robot USAR simulation. While executing the survey task, the robots would circle the sites. Due to the planar size of the simulation area,this still only shows tasks sites 1,2 and 3. Site 4 is relatively far removed to the upper right.

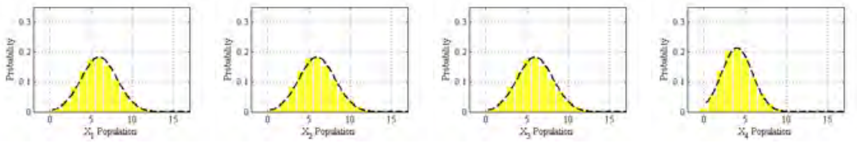


Fig. 5 Macro-continuous and macro-discrete probability distributions of the robot ensemble at each site with no ensemble feedback, *i.e.*, Scenario (I). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

6 Discussion

The results shown in Figure 5-9 demonstrate the ability of our ensemble feedback strategy to simultaneously affect the mean and the variance of the on-site robot populations. More interestingly, our fully decentralized implementation where robots

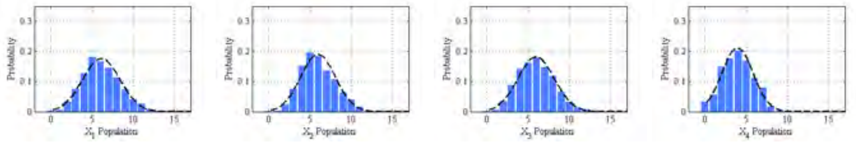


Fig. 6 Macro-continuous and micro-discrete probability distributions of the robot ensemble at each site with no ensemble feedback, *i.e.*, Scenario (I). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

	Sim Type	E[X_1]	E[X_2]	E[X_3]	E[X_4]	**Var**[X_1]	**Var**[X_2]	**Var**[X_3]	**Var**[X_4]
(I) no	Micro-Discrete	6.13	5.99	5.98	3.95	5.20	4.45	4.89	3.58
feedback	Macro-Discrete	6.00	6.02	6.00	3.95	4.76	4.83	4.83	3.36
$\beta_{ij} = 0$	Macro-Continuous	5.98	5.99	5.99	4.01	4.79	4.79	4.79	3.48
(II)	Micro-Discrete	6.01	5.79	5.93	3.84	3.04	2.36	2.63	1.60
Algorithm 1	Macro-Discrete	5.97	5.96	5.97	4.03	2.54	2.56	2.52	1.45
	Macro-Continuous	6.01	5.97	5.99	4.01	2.49	2.53	2.52	1.42
(III)	Micro-Discrete	5.88	5.86	5.93	4.24	3.42	3.48	3.18	2.10
Algorithm 2	Macro-Discrete	5.96	5.96	5.97	4.11	3.03	3.06	3.03	1.82
	Macro-Continuous	5.83	5.79	5.80	4.65	3.29	3.61	3.60	2.50

Table 1 Summary of the steady state means and variances for the 4-site surveillance task as shown Figure 3 for Scenarios (I) No Ensemble Feedback ($\beta = 0$), (II)Robots executing Algorithm 1, and (III) Robots executing Algorithm 2.

	E[X_1]	E[X_2]	E[X_3]	E[X_4]	**Var**[X_1]	**Var**[X_2]	**Var**[X_3]	**Var**[X_4]
(III)	5.88	5.86	5.93	4.24	3.42	3.48	3.18	2.10
(IVa)	6.06	6.02	5.99	4.00	3.05	3.11	2.75	2.20
(IVb)	5.72	6.03	5.80	3.98	3.20	3.74	3.33	1.90

Table 2 Summary of the steady state means and variances for the 4-site surveillance task for Scenarios (III) Robots executing Algorithm 2 w/ no failures, (IVa) Errors in off-site $X_{j|i}$ estimates, (IVb) Erros in on-site X_i estimates.

	E[X_1]	E[X_2]	E[X_3]	E[X_4]	**Var**[X_1]	**Var**[X_2]	**Var**[X_3]	**Var**[X_4]	E[X_L]	**Var**[X_L]
(III)	5.88	5.86	5.93	4.24	3.42	3.48	3.18	2.10	0	0
(Va)	5.26	5.32	5.33	3.99	3.21	2.87	2.89	1.99	2.63	2.56
(Vb)	5.38	5.38	5.20	4.00	3.62	3.35	3.00	2.13	2.53	3.32

Table 3 Summary of the steady state means and variances for the 4-site surveillance task for Scenarios (III) Robots executing Algorithm 2 w/ no failures, (Va) Navigation errors leading to out-of-date estimates, and (Vb) Navigation errors leading to estimates of 0 robots.

rely solely on at-site communication to estimate the various population variables achieves similar performance predicted by the ensemble model (see Figure 9).

However, we note that in our example, the mean task completion rates were slower than the travel transitions rates between sites, $\alpha_{ij}\mathrm{E}[X_i] - \beta_{ij}\mathrm{E}[X_j] < \lambda_{ij}Y_{ij}$. This led to a system where the effects due to the travel delays were overshadowed by the delays introduced by robots spending time at a site. For systems with larger travel times and lower site transition rates, the ensemble may encounter some instability due to poor estimation of the site populations. This issue is alleviated by the natural connection between the travel rate and the communication rate. To further reduce these effects, we could provide the robots the ability to estimate both X_j and Y_{ij}. This is not entirely unreasonable since estimation of Y_{ij} could be achieved as robots encounter other robots as they travel between sites.

While there is some disagreement between the population means and variances predicted by the linear ensemble model and the actual robot simulations, we believe that the main source of the model error is due to the minimum value of the single robot transition rate. Each reaction expression has a rate of zero when either the number of robots at the associated site is zero, or the difference $\alpha_{ij}\mathrm{E}[X_i] - \beta_{ij}\mathrm{E}[X_j] < 0$. Since all of the rates are one-way, they only govern robots leaving a site and thus the transition rate will go to zero if $\alpha_{ij}\mathrm{E}[X_i] - \beta_{ij}\mathrm{E}[X_j] < 0$. The linear approximation does not take into account this saturation effect.

More interestingly, Figures 10-11 show that our strategies are robust to communication failures that lead to erroneous robot population estimates. Lastly, while we note a slight shift in the mean of the results for Scenario (V) (see Figures 12-13), these are due to the fact that at any time an average of 10 robots are always "lost". However, looking at Table 3, we note that when compared to the results of Scenario (III), differences in means and variances are small. In general, poor estimates of the site populations would lead to tighter distributions and this is confirmed by our data. This is because the initial estimate the robot has is $X_i = 1$ which maximizes the feedback term, $\propto X_i^{-1}$, causing the transition rates to slow down with respect to the populations at other sites, which leads to a shrinking of the variance.

When considering localization/navigation errors, we note that robots traveling on longer paths have higher likelihoods of getting lost. This results in a lack of good information getting through to sites that are far away which alters the behavior of those subgroups. This makes intuitive sense since errors tend to accumulate and as more robots get lost, there are less robots to occupy the other sites resulting in a small drop in mean population. However, Site 4, the site further away does not experience this drop since the effective estimates of Site 4's robot population at Sites 1 and vice-versa are regularly misled by robots who eventually make their way back to these sites.

Of significant importance is that our results show that the distributed implementation of the proposed ensemble feedback strategy is in fact robust to both communication and localization errors as long as the time to communicate between robots at site is significantly less than the time it takes the robots to finish the task and move one to another site. From Table 2(c), we note that the average delay introduced by the communication errors at Site 1 is about 2 seconds while the Site 1 changes population about every 8 seconds. In most applications, it is safe to assume that on average communication occurs at a much faster rate than execution of the task by the robot even when communication is lossy. As such, our results show the feasibility and validity of using the ensemble dynamic models to synthesize distributed ensemble feedback strategies.

7 Conclusion and Future Outlook

In this work, we presented a method for synthesizing distributed ensemble feedback control strategies through the development and analysis of an appropriate macroscopic description of the ensemble dynamics. Moment closure techniques were used to derive the ensemble dynamics and through this analysis a linearizing ensemble feedback strategy was obtained. We presented a distributed implementation of the proposed ensemble feedback strategy which can be implemented on robots with limited communication range. The resultant agent-level control policies enabled the team to affect both the mean and the variance of the ensemble population across the various spatially distributed tasks. Furthermore, we showed how the agent-level strategies are robust to communication and navigation errors as long as the average time for communicating between robots is faster than the average time a robot spends at a task.

While the results are positive, it would be interesting to conduct a more complete sensitivity analysis of the distribute ensemble strategy to the various failure modes. From our results, we believe it is possible to select the appropriate feedback gains to not only affect the mean and the variance of the on-site populations, but also to improve the systems overall convergence rate. In general, that the feedback terms, *i.e.*, the β_{ij} terms, look like "backward reactions" in the first moment and does increase the convergence rate of the population means as seen in Figure 2(b). However, simply scaling β_{ij} up would inhibit the at task exit rates from the perspective of a single robot since $k_{ij}X_i = \alpha_{ij}X_i - \beta_{ij}X_j$. As such, while the time to converge would be reduced, the overall effect would be higher mean populations at the tasks and less robots transitioning into the Y_{ij} states. The end result is the team's inability to maintain the desired mean populations at the various tasks. As such, further studies is needed to better understand the relationship between the length of the travel times and time required for individual robots to achieve a stable estimate of the various population variables and how this affects the overall convergence rate. Lastly, it would be interesting to see the extension of these synthesis techniques to a wider range of complex group behaviors.

Acknowledgements

This work is partially supported by the National Science Foundation under Grant No. CNS-1143941.

References

1. S. Berman, A. M. Halasz, M. A. Hsieh, and V. Kumar. Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites. In *Proc. 2008 IEEE Conf. on Decision & Control (CDC'08)*, pages 4376–4381, Cancun, Mexico, 2008.
2. N. Correll and A. Martinoli. Towards multi-robot inspection of industrial machinery - from distributed coverage algorithms to experiments with miniature robotic swarms. *IEEE Robotics and Automation Magazine*, 2008.
3. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'02)*, pages 1327–1332, Washington, DC, 2002.
4. T. S. Dahl, M. J. Mataric, and G. S. Sukhatme. A machine learning method for improving task allocation in distributed multi-robot transportation. In D. Braha, A. Minai, and Y. Bar-Yam, editors, *Understanding Complex Systems: Science Meets Technology*, pages 307–337. Springer, Berlin, Germany, June 2006.
5. M. B. Dias, R. M. Zlot, N. K., and A. Stentz. Market-based multirobot coordination: a survey and analysis. *Proc. of the IEEE*, 94(7):1257–1270, July 2006.
6. B. P. Gerkey and M. J. Mataric. Sold!: Auction methods for multi-robot control. *IEEE Trans. on Robotics & Automation*, 18(5):758–768, Oct 2002.
7. B. P. Gerkey and M. J Mataric. A formal framework for the study of task allocation in multi-robot systems. *Int. J. of Robotics Research*, 23(9):939–954, September 2004.
8. D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. of Computational Physics*, 22(4):403–434, 1976.
9. D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. of Physical Chemistry*, 81:2340–2361, 1977.
10. M. Golfarelli, D. Maio, and S. Rizzi. Multi-agent path planning based on task-swap negotiation. In *Proc. 16th UK Planning & Scheduling SIG Workshop*, Durham, England, 1997. PlanSIG.
11. A. Halasz, M. A. Hsieh, S. Berman, and V. Kumar. Dynamic redistribution of a swarm of robots among multiple sites. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'07)*, pages 2320–2325, San Diego, CA, Oct.–Nov. 2007.
12. J. P. Hespanha. Moment closure for biochemical networks. In *Proc. of the 3rd Int. Symp. on Control, Communications and Signal Processing*, Mar. 2008.
13. K. Hosokawa, I. Shimoyama, and H. Miura. Dynamics of self assembling systems: Analogy with chemical kinetics. *Artificial Life*, 1(4):413–427, 1994.
14. M. A. Hsieh, A. H., S. Berman, and V. Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, December 2008.
15. Ali Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, June 2003.
16. E. Klavins. Proportional-integral control of stochastic gene regulatory networks. In *Proc. 2010 IEEE Conf. on Decision & Control (CDC'10)*, 2010.
17. K. Lerman, C. Jones, A. Galstyan, and M. J. Mataric. Analysis of dynamic task allocation in multi-robot systems. *Int. J. of Robotics Research*, 2006.
18. A. Martinoli, K. Easton, and W. Agassounon. Modeling of swarm robotic systems: a case study in collaborative distributed manipulation. *Int. J. of Robotics Research: Special Issue on Experimental Robotics*, 23(4-5):415–436, 2004.
19. T. W. Mather and M. A. Hsieh. Distributed robot ensemble control for deployment to multiple sites. In *2011 Robotics: Science and Systems*, Los Angeles, CA USA, Jun/Jul 2011.
20. T. W. Mather and M. A. Hsieh. Ensemble synthesis of distributed control and communication strategies. In *in the Proc. IEEE Int. Conf. on Robotics and Automation (ICRA2012)*, Minneapolis, MN USA, May 2012.
21. Nathan Michael, Calin Belta, and Vijay Kumar. Controlling three dimensional swarms of robots. In *IEEE Int. Conf. on Robotics & Automation (ICRA) 2006*, pages 964–969, Orlando, FL, April 2006.

22. N. Napp, S. Burden, and E. Klavins. Setpoint regulation for stochastically interacting robots. In *Robotics: Science and Systems V*. MIT Press, 2009.
23. USARSim. Unified system for automation and robot simulation. http://usarsim.sourceforge.net, 2007.

Appendix – Supplemental Data

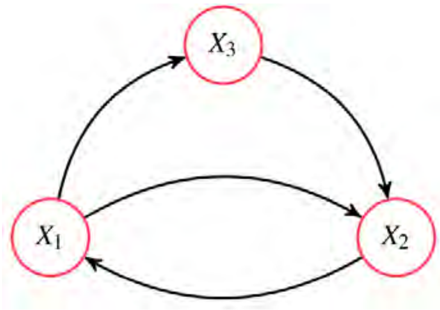


Fig. 7 A Graphical representation of a chemical reaction network representation of an example 3-site surveillance problem where X_i denotes the number of robots at site i executing surveillance task U_i.

$$\frac{d}{dt}\begin{bmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \\ \mathrm{E}[X_3] \\ \mathrm{E}[X_1^2] \\ \mathrm{E}[X_2^2] \\ \mathrm{E}[X_3^2] \\ \mathrm{E}[X_1X_2] \\ \mathrm{E}[X_1X_3] \\ \mathrm{E}[X_2X_3] \end{bmatrix} = \begin{bmatrix} -\alpha_{12}-\alpha_{13}-\beta_{21} & \alpha_{21}+\beta_{12} & \beta_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_{12}+\beta_{21} & -\alpha_{21}-\beta_{12}-\beta_{32} & \alpha_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_{13} & \beta_{32} & -\alpha_{32}-\beta_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_{12}+\alpha_{13}-\beta_{21} & \alpha_{21}-\beta_{12} & -\beta_{13} & -2\alpha_{12}-2\alpha_{13}-2\beta_{21} & 0 & 0 & 2\alpha_{21}+2\beta_{12} & 2\beta_{13} & 0 \\ \alpha_{12}-\beta_{21} & \alpha_{21}-\beta_{12}-\beta_{32} & \alpha_{32} & 0 & -2\alpha_{21}-2\beta_{12}-2\beta_{32} & 0 & 2\alpha_{12}+2\beta_{21} & 0 & 2\alpha_{32} \\ \alpha_{13} & -\beta_{32} & \alpha_{32}-\beta_{13} & 0 & 0 & -2\alpha_{32}-2\beta_{13} & 0 & 2\alpha_{13} & 2\beta_{32} \\ \beta_{21}-\alpha_{12} & \beta_{12}-\alpha_{21} & 0 & \alpha_{12}+\beta_{21} & \alpha_{21}+\beta_{12} & 0 & \substack{-\alpha_{12}-\alpha_{13}-\alpha_{21} \\ -\beta_{12}-\beta_{21}-\beta_{32}} & \alpha_{32} & \beta_{13} \\ -\alpha_{13} & 0 & \beta_{13} & \alpha_{13} & 0 & \beta_{13} & \beta_{32} & \substack{-\alpha_{12}-\alpha_{13}-\alpha_{32} \\ -\beta_{13}-\beta_{21}} & \alpha_{21}+\beta_{12} \\ 0 & \beta_{32} & -\alpha_{32} & 0 & \beta_{32} & \alpha_{32} & \alpha_{13} & \alpha_{12}+\beta_{21} & \substack{-\alpha_{21}-\alpha_{32} \\ -\beta_{12}-\beta_{13}-\beta_{32}} \end{bmatrix} \begin{bmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \\ \mathrm{E}[X_3] \\ \mathrm{E}[X_1^2] \\ \mathrm{E}[X_2^2] \\ \mathrm{E}[X_3^2] \\ \mathrm{E}[X_1X_2] \\ \mathrm{E}[X_1X_3] \\ \mathrm{E}[X_2X_3] \end{bmatrix} \tag{11}$$

Rates	α_{12}	α_{13}	α_{21}	α_{32}	β_{12}	β_{21}	β_{13}	β_{21}
w/o (8)	1.66	1.97	2.41	3.94	0	0	0	0
w/ (8)	2.93	2.52	2.82	6.98	0.85	1.01	0.60	1.01

Table 4 Values of α_{ij}'s and β_{ij}'s for the system given by equation (11) with and without the ensemble feedback strategy given by equation (8). These values were chosen such that $\mathrm{E}[X] = [10, 15, 5]$ and $Var[X] = [4, 4, 2]$ with the ensemble feedback strategy.

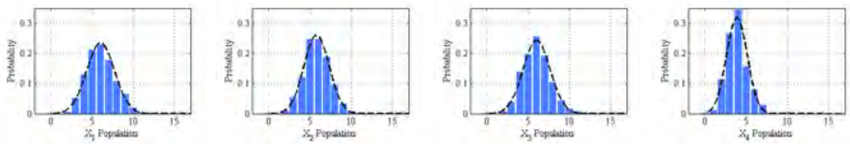


Fig. 8 Macro-continuous and micro-discrete probability distribution of the robot ensemble at each task site with robots executing Algorithm 1, *i.e.*, Scenario (II). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

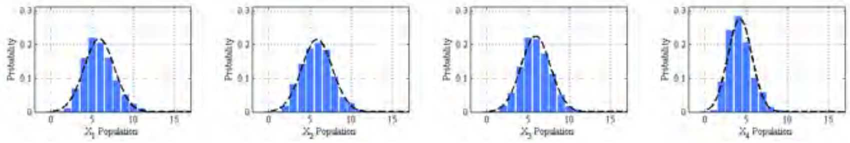


Fig. 9 Macro-continuous and micro-discrete probability distribution of the robot ensemble at each task site with robots executing Algorithm 2, *i.e.*, Scenario (III). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

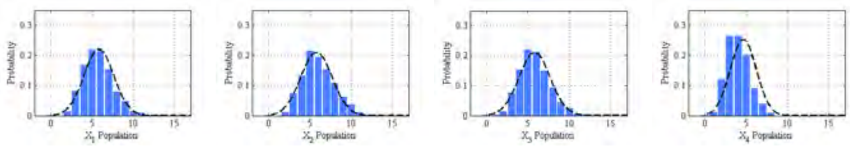


Fig. 10 Macro-continuous and micro-discrete probability distribution of the robot ensemble at each task site with robots executing Algorithm 2 with communication errors leading to incorrect off-site population estimates, *i.e.*, Scenario (IVa). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

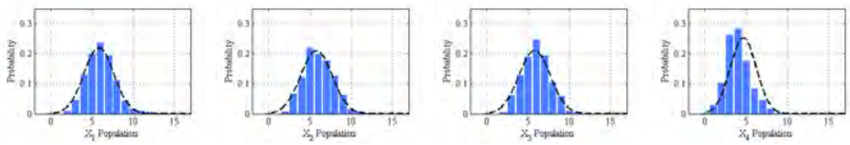


Fig. 11 Macro-continuous and micro-discrete probability distribution of the robot ensemble at each task site with robots executing Algorithm 2 with communication errors leading to erroneous on-site population estimates, *i.e.*, Scenario (IVb). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

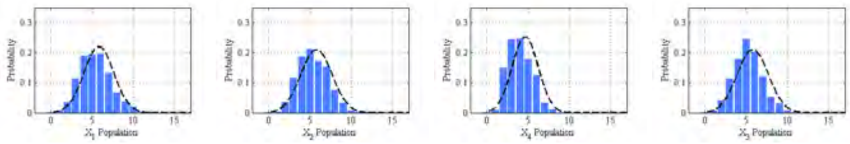


Fig. 12 Macro-continuous and micro-discrete probability distribution of the robot ensemble at each task site with robots executing Algorithm 2 with localization/navigation errors and with old/erroneous non-zero off-site population estimates, *i.e.*, Scenario (Va). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

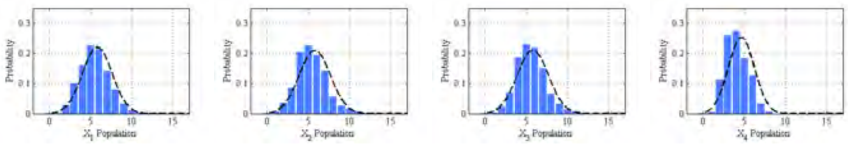


Fig. 13 Macro-continuous and micro-discrete probability distribution of the robot ensemble at each task site with robots executing Algorithm 2 with localization/navigation errors leading to estimates of off-site populations set to zero, *i.e.*, Scenario (Vb). Macro-continuous results obtained from numerically solving the moment closure are plotted as a dashed line.

Proportional rate	α_{12}	α_{13}	α_{14}	α_{21}	α_{23}	α_{31}	α_{32}	α_{41}
Rates $(10^{-3}/sec)$	65.0	61.8	39.7	64.1	58.7	62.6	57.9	15.3
Feedback rate	β_{12}	β_{13}	β_{14}	β_{21}	β_{23}	β_{31}	β_{32}	β_{41}
Rates $(10^{-3}/sec)$	23.0	23.0	34.5	23.0	23.0	23.0	23.0	15.3

Table 5 The transition rate gains that control the wait times at the tasks. $k_{ij} = \alpha_{ij} - \beta_{ij}\hat{X}_{j|i}\hat{X}_i^{-1}$

Intersite Travel Times	λ_{12}^{-1}	λ_{13}^{-1}	λ_{14}^{-1}	λ_{21}^{-1}	λ_{23}^{-1}	λ_{31}^{-1}	λ_{32}^{-1}	λ_{41}^{-1}
$\mathrm{E}[T_{ij}]$ (sec)	41.6	45.2	107.0	39.3	45.2	41.2	46.7	99.1

Table 6 Expected travel times, or inverse rates, λ_{ij}^{-1}, between tasks.