Ensemble Synthesis of Distributed Control and Communication Strategies

T. William Mather and M. Ani Hsieh

Abstract—We present an ensemble framework for the design of distributed control and communication strategies for the dynamic allocation of a team of robots to a set of tasks. In this work, we assume individual robot controllers are sequentially composed of individual task controllers. This assumption enables the representation of the robot ensemble dynamics as a class of stochastic hybrid systems that can be modeled as continuous-time Markov jump processes where feedback strategies can be derived to control the team’s distribution across the tasks. Since the distributed implementation of these feedback strategy requires the estimation of certain population variables, we show how the ensemble model can be expanded to incorporate the dynamics of the information exchange. This then enables us to optimize the individual robot control policies to ensure overall system robustness given some likelihood of resource failures. We consider the assignment of a team of homogeneous robots to a collection of spatially distributed tasks and validate our approach via high-fidelity simulations.

I. INTRODUCTION

In the last ten years, there has been significant interest in applying a swarming paradigm towards the control and coordination of large robot teams where individuals are programmed with simple but identical behaviors that rely solely on limited on-board computational, communication, and sensing resources. This has lead to much progress in the development of collective motion control and/or consensus forming strategies for homogeneous teams that often come with rigorous theoretical guarantees [1]–[4]. 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 present an ensemble approach towards the modeling, design, and analysis of distributed control and communication strategies that takes into account the small number effects resulting from the distribution of many robots to a set of tasks. This is similar to the multi-task (MT) robots, single-robots (SR), time-extended assignment (TA) problem [5]. Unlike existing approaches which require the execution of complex bidding schemes by the robots coupled with careful selections of individual reward and penalty functions [6]–[8], we use an appropriate macroscopic description of the ensemble dynamics to simultaneously address the allocation and controller synthesis problems for the team. 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.

While macroscopic continuous models have been used to describe the ensemble dynamics of robotic self-assembly [9], [10] and robotic swarm systems [11]–[14], these works have mostly focused on the modeling side of the problem. Similar to [13], [14], we consider the design of stochastic transition rules to enable the team to autonomously achieve a desired distribution across a set of tasks. Different from these works, we build off recent results in [15], [16] to synthesize agent-level control policies to affect both the mean and the variance of the distribution of robots across the tasks. In this work, the simultaneous execution of spatially distributed tasks by a team of robots is modeled as a polynomial stochastic hybrid system and moment closure techniques are used to model the moment dynamics of the ensemble distribution and the information exchange [17]. We limit our discussion to teams of homogeneous robots, however, our methods can be extended to heterogeneous teams.

The main contribution is an integrated approach for designing distributed agent-level control and communication policies that are invariant to team size. By modeling both the robot task execution and the inter-robot communication as a stochastic hybrid system, the entire system can be cast into an optimization framework. The paper is structured as follows: We formulate our approach in Section II. The synthesis of our ensemble feedback control and communication strategy is presented in Section III. Section IV presents our simulation results. We conclude with a discussion and directions for future work in Sections V and VI respectively.

II. PROBLEM FORMULATION

Consider the assignment of $N$ robots to $M$ tasks each located at a different locale within the workspace. The objective is to synthesize a decentralized control strategy to enable the team to autonomously distribute across the $M$ tasks and maintain the desired allocation at the various locales. Building off the results in [16], we synthesize our distributed agent-level control policies to enable the team to maintain both the mean and the variance of the desired ensemble distribution taking into account the communication requirements needed to achieve the desired allocation. For
the sake of completeness, we briefly outline our assumptions and the development of our ensemble models and invite the interested reader to [16] for further details.

A. Individual Robot Controller

Given a collection of \( \{1, \ldots, M\} \) spatially distributed tasks, we use a strongly connected directed graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) to model the physical interconnection topology of the tasks. Each task is represented by a vertex in \( \mathcal{V} = \{1, \ldots, M\} \) and a directed edge \( e_{ij} \) exists between two vertices \( (i, j) \in \mathcal{V} \times \mathcal{V} \) if a road exists between two tasks in \( \mathcal{V} \). We assume roads are bidirectional and thus for every \( e_{ij} \), there is a corresponding \( e_{ji} \).

Given the \( M \) tasks, we denote the set of task controllers for each task as \( \{U_1, \ldots, U_M\} \). The single robot controller is obtained through the sequential composition of \( \{U_1, \ldots, U_M\} \) such that the physical constraints given by \( \mathcal{G} \) are satisfied. Fig. 1(a) shows an example robot controller represented as a finite state automaton where each automaton state \( i \) is associated with a task controller \( U_i \). The arrows in Fig. 1(a) denote state transitions that satisfy the constraints specified in \( \mathcal{G} \).

Similar to previous work, we consider the case where the team must maintain a desired allocation of the robots across the \( M \) locations to ensure task completion. The tasks at each location can be executed in parallel and completely asynchronously. At each task, robots execute the corresponding \( U_i \) and upon completion navigate to the next adjacent site based on \( \mathcal{G} \). Thus, we assume each robot has complete knowledge of \( \mathcal{G} \), is able to localize itself in the workspace, and navigate from one site to another while avoiding collisions with other robots.

B. The Ensemble Model

For a team of \( N \) robots, each executing the same sequentially composed controller similar to the one in Fig. 1(a), we model the allocation problem as a polynomial stochastic hybrid system (pSHS) where moment closure techniques are used to describe the time evolution of the distribution of the team across the various tasks. Let \( X_i(t) \) denote the number of robots executing task \( i \) or at site \( i \). The system state is given by \( X(t) = [X_1(t), \ldots, X_M(t)]^T \) with the desired distribution of the ensemble given by \( \bar{X} \). We assume robots executing task \( i \), i.e. co-located at the same site, have the ability to communicate and form a consensus on \( \bar{X}_i \) in a distributed fashion. For control purposes, let \( \bar{X}_{ij} \) denote the estimate of \( X_i \) by robots executing task \( j \) based on information provided by robots who came from task \( i \).

Since robots must navigate from one site to another and avoiding collisions with one another, the variability in robot arrival times at each task is modeled by transition rates. For every edge \( e_{ij} \in \mathcal{E} \), we assign a constant \( k_{ij} > 0 \) to denote the transition probability per unit time for one agent from site \( i \) to go to site \( j \). Given \( \mathcal{G} \) and the set of \( k_{ij} \)'s, we obtain a set of stochastic transition rules of the form \( X_i \xrightarrow{k_{ij}} X_j \) for all \( e_{ij} \in \mathcal{E} \). These rules imply that robots at site \( i \) will move to site \( j \) with a rate of \( k_{ij}X_i \). In general \( k_{ij} \neq k_{ji} \) and \( k_{ij} \) encodes the inverse of the average time a robot spends at task/site \( i \).

If we assume the ensemble dynamics is Markov, then the moment dynamics of the distribution can be modeled as a set of linear differential equations. For the discrete random variable \( X_i \), the moment equations is given by the extended generator of the system [17]. For a real-valued function \( \psi(X_i) \), the extended generator is an expression for the time derivative of the expected value of \( \psi \), i.e., \( \frac{d}{dt}E[\psi(X_i)] = E[\psi'(X_i)] \). To obtain the rate of the change of the expected value of \( X_i \), \( \frac{d}{dt}E[X_i] \), we let \( \psi(X_i) = X_i \). Similarly, to obtain \( \frac{d}{dt}E[X_i X_j] \), we let \( \psi(X_i) = X_i^2 \).

For a team of \( N \) robots executing \( M \) tasks, the ensemble dynamics are given by

\[
\begin{align*}
\frac{d}{dt}E[X] &= KE[X] \\
\frac{d}{dt}E[XX^T] &= KE[XX^T] + E[XX^T]K^T + \cdots \\
\Gamma(&\alpha, E[X])
\end{align*}
\]

where \( [K]_{ij} = k_{ij} \) and \( [K]_{ii} = -\sum_{(i,j) \in \mathcal{E}} k_{ij} \) [16]. It is important to note that \( K \) is a Markov process matrix and thus is negative semidefinite. This coupled with the conservation constraint \( \sum X_i = N \) leads to exponentially stability of the system given by (1) [16]. Furthermore, the entries of \( \Gamma(\alpha, E[X]) \) are all linear with respect to the \( k_{ij} \)'s and the means \( E[X] \).

We note that \( k_{ij} \)'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 [13]. In essence, the \( k_{ij} \)'s translate into a set of stochastic guard conditions for the single robot controllers which result in decentralized agent-level control policies that shape the steady-state distribution of the ensemble.

III. ENSEMBLE DESIGN

Since the rate in which agents in state \( X_i \) transition to \( X_j \) depends on \( X_i \), the more agents in \( X_i \), the faster they transition to \( X_j \). However, 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 [15], [16].
A. Synthesis

To ensure that the closed-loop moment dynamics remain closed, consider the feedback controller first proposed in [16]

\[ u = -K_{\beta} E[X] \quad \text{s.t.} \quad K_{\beta}^{ij} \begin{cases} \beta_{ji} & \forall e_{ij} \in \mathcal{E} \\ -\sum_{(i,j) \in \mathcal{E}} \beta_{ji} & \forall i = j \\ 0 & \text{otherwise} \end{cases} \]

resulting with the following closed-loop moment dynamics

\[
\begin{align*}
\frac{d}{dt} E[X] &= (K_{\alpha} + K_{\beta}) E[X] \\
\frac{d}{dt} E[XX^T] &= (K_{\alpha} + K_{\beta}) E[XX^T] + \cdots \\
E[XX^T] &= (K_{\alpha} + K_{\beta})^T + \Gamma(\alpha, \beta, E[X]).
\end{align*}
\]

The above equations are obtained by simply substituting \( k_{ij} = \alpha_{ij} - \beta_{ij}X_{ij} \) and applying the extended generator to \( \psi(X_i) = X_i \).

B. Modeling Distributed Communications

The feedback strategy given by (2) gives robots at task \( i \), the ability to set their own state transition rates to be independent from \( X_i \), the number of robots at task \( i \). The trade-off is that robots at task \( i \) must estimate the number of robots at adjacent tasks, \( i.e., X_j \) for every \( e_{ij} \in \mathcal{E} \). However, as robots transition from one task to another, this presents an opportunity for information exchange with robots located at different tasks. If we assume that robots with finite communication ranges have the ability to communicate with other robots that are executing the same task, then it is possible for task \( i \) robots to estimate the number of robots at \( i,j \). Furthermore, as robots arrive from adjacent tasks \( j \), the robots at \( i \) can also estimate the number of robots at \( j \), \( X_{ji} \).

In this case, both \( \dot{X}_i \) and \( \dot{X}_{ji} \) can be estimated via some distributed consensus protocol. Additionally, \( \dot{X}_i \) and \( \dot{X}_{ji} \) are updated at discrete time intervals based on the arrival rates of robots from neighboring tasks. The only difference between \( \dot{X}_i \) and \( \dot{X}_{ji} \) is that \( \dot{X}_{ji} \) can experience larger jumps since the estimate of the number of robots at task \( j \) can jump from 3 to 6. The similarity between the ensemble state variables \( X_e \) and their corresponding estimates \( \dot{X}_e \) implies that we can model the communication dynamics as a second Markov jump process. Different from [16], this is the main contribution of this work.

Applying the extended generator [16] to \( \dot{X}_{ij} \) gives

\[
\frac{d}{dt} E[X_{ij}] = E[(X_i - \dot{X}_{ji})\lambda_{ij}] = \lambda_{ij}E[X_i] - \lambda_{ij}E[\dot{X}_{ji}]
\]

where \( \lambda_{ij} \) represents the per robot transit rate from task \( i \) to \( j \). In fact, when tasks are physically located in different parts of the workspace, a more appropriate model for the communication dynamics is the time delayed version of (4) given by

\[
\frac{d}{dt} E[X_{ij}] = \lambda_{ij}E[X_i](t - \tau_{ij}) - \lambda_{ij}E[\dot{X}_{ji}](t)
\]

where \( \tau_{ij} \approx k_{ij} \) since it takes a non-zero amount of time for robots to complete task \( i \).

In the examples below, we introduce a set of intermediate navigation tasks for every \( e_{ij} \in \mathcal{E} \). In other words, we simply expand the individual robot controllers to include the navigation state between task \( i \) and \( j \) as shown in Fig. 2(a). This results in an expansion of the ensemble states to include \( Y_{ij} \) for every \( e_{ij} \in \mathcal{E} \) such that \( Y_{ij} \) is the number of robots traveling between tasks. These navigation tasks are denoted with a different letter because we assume that the robots cannot share information or estimates while “en route”. The expanded ensemble dynamics is then given by

\[
\begin{align*}
X_i &\xrightarrow{\alpha_{ij} - \beta_{ij}X_{ij}} X_j \\
Y_{ij} &\xrightarrow{\lambda_{ij} X_{ij}} Y_{ij} \xrightarrow{\text{Est}} \alpha_{ij} - \beta_{ij}X_{ij} \xrightarrow{\lambda_{ij} X_{ij}} \emptyset
\end{align*}
\]

C. From Ensemble to Individual Robots

Since the ensemble dynamics is derived assuming single robot controllers are given by a sequential composition of the individual task controllers \( U_i \), the closed-loop ensemble feedback model given by (5) provides the transition probabilities and the ensemble state estimation strategy for the robots to allocate across and execute the various tasks. We refer the interested reader to [16] for the details on the distributed implementation of the closed-loop ensemble feedback strategy given by (5).

IV. RESULTS

To validate our ensemble feedback control and communication strategies, 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 based on the Stochastic Simulation Algorithm (SSA) which is mathematically equivalent to an agent-based simulation [18]. At the lowest level are the micro-discrete simulations which are agent-based simulations using a team of mSRV-1 robots in USARSim [19].
A. Controller Gain Selection

It is important to note that the rate of population exchange in the model allows for backwards flow when \( \beta_{ij}X_j > \alpha_{ij}X_i \). To prevent this, we restrict this rate to be greater than or equal to zero. In essence, we saturate our feedback controller at runtime such that the rate never goes below zero. The effect at the agent-level is that robots would simply remain at their current tasks and never move on to neighboring tasks. This saturation effect is highly non-linear and leads to a mismatch between the actual ensemble dynamics and those predicted by the moment dynamics (5). To ensure that the actual system remains within the linear regime we selected the gains \( \alpha_{ij} \) and \( \beta_{ij} \) such that the variance of the transfer rates is reduced. This was achieved by selecting the gains such that the following expression is minimized

\[
C_{\alpha\beta}(\alpha_{ij}, \beta_{ij}) = \sum_{i,j} \left[ E[\alpha_{ij}X_i - \beta_{ij}X_j] \right] - \cdots \\
2 \sqrt{\text{Var}(\alpha_{ij}X_i - \beta_{ij}X_j)}.
\]

(5)

B. Scenarios

In this work, we consider four scenarios each with \( M = 3 \) and \( N = 15 \). The scenarios are defined as follows:

S 1 Uniformly Maximizing \( E[X_i] \): This scenario minimizes \( C_{\alpha\beta} \), while maximizing the mean \( X_i \) for all \( i \). These results are shown in plots 3(a) and 3(b).

S 2 Stabilizing to \( E[\hat{X}] = [4, 5, 3]^T \): This scenario minimizes the \( C_{\alpha\beta} \) with the constraint \( E[\hat{X}] = [4, 5, 3]^T \). These results are shown in 3(c) and 3(d).

S 3 Unequal mean travel times between tasks: This scenario is the same as the previous one except the distance between Task 1 and Task 2 is increased by a factor of ten. These results are shown in 4(a).

S 4 Minimizing \( \text{Var}(X) \): This scenario attempts to tightly control the task population variance of a uniform distribution. The controller gains for this scenario are chosen to minimize the likelihood that \( C_{\alpha\beta} \) is minimized. The objective is to show that even outside of the linear regime, the ensemble feedback strategies derived from our linear models still performs relatively well. These results are shown in 4(b) and 4(c).

By minimizing (5), the objective is to select the \( \alpha_{ij} \)'s and \( \beta_{ij} \)'s such that the mean rate of each transition is within two standard deviations above zero. We used MATLAB’s \texttt{fmincon()} to compute the set of controller gains to minimize (5) while satisfying any additional constraints. Given \( E[\hat{X}] \), (1) is used to compute an initial set of \( \alpha_{ij} \)'s and \( \beta_{ij} \)'s to initialize the optimization routine.

V. DISCUSSION

By minimizing the likelihood that the mean rates of transition for scenarios 1-3 operate within the saturation range, our results show strong agreement between the distribution of the team across the 3 tasks and those predicted by moment dynamics given by (5). In scenario 1 (S1), despite a slight discrepancy between the means of the micro-discrete and the macro results, overall, the steady state probability distribution shows good qualitative matching between the two. The reason for this discrepancy is likely due to the small number of micro-discrete simulations that were conducted. More interestingly is the difference between the variances in \( \hat{X}_{ij} \) determined by the macro-continuous and macro-discrete simulations which differs by about 12%. Since updates to \( \hat{X}_{ij} \) are limited to \( \pm 1 \), this suggest that the communication is more faithfully modeled by the macro-continuous simulations when changes in \( \hat{X}_{ij} \) at every update step can be any integer value less than \( N \).

The differences in population variances between the macro-discrete and macro-continuous models also appear in the scenario 2 (S2) results. Once again, this is likely due to the overestimation of the variances of the \( X_{ij} \) variables. For scenario 3 (S3) where one of the roads (between tasks 1 and 2) is very long, the results are intuitive. In order to maintain the same distribution as in S2, the robots eventually stopped switching between tasks 1 and 2 which resulted in larger population variances at these two tasks. With only one viable path between tasks 1 and 2 (via task 3) there was only one legitimate information pathway to transfer relevant information without incurring significant delay. Another effect of the longer road length between tasks is the slow updates of \( \hat{X}_{12} \) and \( \hat{X}_{21} \). Without the ensemble feedback strategy and appropriate controller gain selection, this delay would have increased the likelihood that more robots traveled on the longer road.

For scenario 4 (S4), we purposely selected controller gains outside of the linear regime since rate saturation, \( k_{ij} = \max \{ \alpha_{ij}X_i - \beta_{ij}X_{ij}, 0 \} \), can be of significant concern when population levels are low, e.g., when population levels are approximately in the single digits. For the actual multi-agent robotic system, the addition or removal of a single robot at a task can significantly change the transfer rates between tasks especially when the number of robots at a task is low. The relative magnitudes of the \( \alpha_{ij} \) and \( \beta_{ij} \) values were chosen to exhibit rate saturation in S4 and our results show that the saturation tends to drive the means higher. This is because as \( X_i \) decreases, the rate of robots leaving the task falls as the magnitude of \( \alpha_{ij}X_i \) decreases in comparison to \( \beta_{ij}X_{ij} \). As such, the system tends to operate at or close to the saturation point since the \( \alpha_{ij}X_i - \beta_{ij}X_{ij} \) cannot be negative and thus disallowing populations to drop below zero and thus artificially skewing the mean towards a higher value. This effect can be seen in Figures 4(b) and 4(c) where \( X_i \) never drops below 3 but occasionally reaches 7 in the micro-discrete simulations. However, despite the presence of the saturation effects, the ensemble strategy derived from the macro-continuous models performs quite well as shown in Table I.

In this work, we assumed that the robots at each task can instantly reach consensus whenever a robot leaves or arrives at the task site. The additional time it takes to reach consensus can be integrated and simulated within our current framework. This can be achieved in various ways depending on whether we allow robots to transition in and out of a task while consensus is being established. However, how well
TABLE I

STEADY STATE MEANS AND VARIANCES FOR THE 4 SCENARIOS PRESENTED.

<table>
<thead>
<tr>
<th>Case</th>
<th>Decryption</th>
<th>E[X\textsubscript{1}]</th>
<th>E[X\textsubscript{2}]</th>
<th>E[X\textsubscript{3}]</th>
<th>Var[X\textsubscript{1}]</th>
<th>Var[X\textsubscript{2}]</th>
<th>Var[X\textsubscript{3}]</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Uniformly</td>
<td>4.13</td>
<td>4.20</td>
<td>4.11</td>
<td>2.21</td>
<td>2.24</td>
<td>1.89</td>
</tr>
<tr>
<td></td>
<td>Maximizing</td>
<td>4.32</td>
<td>4.32</td>
<td>4.30</td>
<td>2.07</td>
<td>2.08</td>
<td>2.06</td>
</tr>
<tr>
<td></td>
<td>E[X]</td>
<td>4.32</td>
<td>4.32</td>
<td>4.32</td>
<td>2.34</td>
<td>2.34</td>
<td>2.34</td>
</tr>
<tr>
<td>S2</td>
<td>Stabilizing</td>
<td>3.87</td>
<td>4.82</td>
<td>2.84</td>
<td>2.44</td>
<td>2.51</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>E[X] = [4, 5, 3]</td>
<td>4.01</td>
<td>4.98</td>
<td>2.98</td>
<td>2.19</td>
<td>2.27</td>
<td>2.19</td>
</tr>
<tr>
<td></td>
<td>Macro-Discrete</td>
<td>4.00</td>
<td>5.00</td>
<td>3.00</td>
<td>2.44</td>
<td>2.67</td>
<td>2.35</td>
</tr>
<tr>
<td>S3</td>
<td>Unequal</td>
<td>Micro-Discrete</td>
<td>3.98</td>
<td>4.80</td>
<td>3.11</td>
<td>2.14</td>
<td>2.24</td>
</tr>
<tr>
<td></td>
<td>Travel Times</td>
<td>Macro-Discrete</td>
<td>4.00</td>
<td>4.98</td>
<td>3.00</td>
<td>2.54</td>
<td>2.92</td>
</tr>
<tr>
<td>S4</td>
<td>Minimizing</td>
<td>Micro-Discrete</td>
<td>4.19</td>
<td>4.25</td>
<td>4.24</td>
<td>0.69</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>Macro-Discrete</td>
<td>4.13</td>
<td>4.13</td>
<td>4.14</td>
<td>0.65</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Macro-Continuous</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>0.67</td>
<td>0.67</td>
</tr>
</tbody>
</table>

TABLE II

OPTIMIZED CONTROLLER GAINS - $\alpha_{ij}$, $\beta_{ij}$ AND NAVIGATION RELATED TRANSITION RATES - $\lambda_{ij}$

<table>
<thead>
<tr>
<th>Case</th>
<th>$\alpha_{12}$</th>
<th>$\alpha_{13}$</th>
<th>$\alpha_{21}$</th>
<th>$\alpha_{23}$</th>
<th>$\alpha_{31}$</th>
<th>$\alpha_{32}$</th>
<th>$\beta_{12}$</th>
<th>$\beta_{13}$</th>
<th>$\beta_{21}$</th>
<th>$\beta_{23}$</th>
<th>$\beta_{31}$</th>
<th>$\beta_{32}$</th>
<th>$\lambda_{12}$</th>
<th>$\lambda_{13}$</th>
<th>$\lambda_{21}$</th>
<th>$\lambda_{23}$</th>
<th>$\lambda_{31}$</th>
<th>$\lambda_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>714</td>
<td>556</td>
<td>714</td>
<td>556</td>
<td>556</td>
<td>556</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>118</td>
<td>93</td>
<td>116</td>
<td>84</td>
<td>94</td>
<td>95</td>
<td>23</td>
<td>31</td>
<td>55</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>714</td>
<td>556</td>
<td>714</td>
<td>556</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>13</td>
<td>102</td>
<td>11</td>
<td>98</td>
<td>97</td>
<td>95</td>
<td>3</td>
<td>43</td>
<td>5</td>
<td>70</td>
<td>2</td>
<td>1</td>
<td>71</td>
<td>556</td>
<td>71</td>
<td>556</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>714</td>
<td>556</td>
<td>714</td>
<td>556</td>
<td>556</td>
<td>556</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. Steady state distribution for scenarios S1 and S2. (a) and (c) show the macro-continuous steady state displayed as a continuous Gaussian distribution, and (b) and (d) shows the macro-discrete simulation results as a discrete probability density function.

VI. CONCLUSION AND FUTURE WORK

In this work, we presented a framework for synthesizing distributed ensemble feedback control and communication strategies to address the dynamic allocation of an ensemble of robots to a collection of tasks. Different from existing methods, our approach does not require robots to execute complex bidding schemes and careful selection of individual reward and penalty functions. Instead, we derive a suitable macroscopic description of the ensemble dynamics by assuming individual robot controllers can be obtained through a sequential composition of the individual task controllers. This approach allows us to solve both the allocation and controller synthesis problem for the team such that the resulting robot control strategies are decentralized and invariant to team size. We believe this approach is a step towards the development of a general framework that can incorporate techniques that address local collaboration between relatively small numbers of robots which remain amenable to team-level analysis.

There are many directions for future work. From our results, we believe there exist a trade-off between the team’s ability to reduce the variances in the number of robots executing a task and the robustness of the ensemble in the presence of catastrophic failures. This line of investigation is of particular interest since the proposed feedback strategy has the potential to drive the population variances to zero which prevents robots from switching from one task to another. Such an approach would result in cutting off the entire communication network for the ensemble resulting in no information exchange between locations. Under these
circumstances, the ensemble would then be unable to respond to any changes in robot population levels at given tasks should robots fail or disappear.

In our current formulation, we have essentially tied our resources with our communications capability which limits the classes of allocation strategies that can be modeled and synthesized within this framework. Specifically, our current approach only considers the assignment of robots to a collection of tasks that can executed in parallel and asynchronously. However, we would like to extend this framework to include the execution of complex tasks where strong precedence constraints between subtasks must be satisfied at all times. Finally, as mentioned in our problem formulation, this framework can be extended to include heterogeneous teams of robots. The moment dynamics of heterogeneous robot ensembles will invariably be nonlinear and break the moment closure property. As such, one must determine the appropriate moment closure techniques to employ for the ensemble dynamics to enable synthesis of feedback and communication strategies.

**References**


