Toward a Multi-Robot Coordination Formalism

Chris Jones and Maja J Matarić

Computer Science Department
University of Southern California, Los Angeles, CA 90089-0781, USA
{cvjones|maja}@robotics.usc.edu

Abstract

Coordination is an essential characteristic of any system, either natural or artificial, that is composed of multiple interacting agents. The mechanism by which the coordination is achieved determines such properties as how robust the system is to environmental perturbations, how efficient it is in performing a given task, and, more generally, what types of coordinated tasks can be achieved. A formal understanding of these issues will permit more focused analysis of coordinated natural systems and more effective approaches to the design of coordinated artificial systems. Toward this end, we are systematically investigating coordination in the context of multi-robot systems. We present a formal framework for multi-robot system (MRS) coordination and use it to discuss and reason about coordination. Using this principled foundation, we are developing a suite of general methods by which to automatically synthesize controllers for robots constituting a MRS such that a given task is performed in a coordinated fashion. This paper presents a method for the automatic synthesis of a specific type of controller, one that retains some internal state (e.g., memory of past occurrences) but is not capable of communication. Understanding the capabilities and limitations of a coordinated system composed of individual agents with such properties contributes to the understanding of when memory alone is sufficient to achieve the desired coordination and when other capabilities, such as communication, may be necessary. We validate our formal approach to the study of coordination in a multi-robot construction task domain through the use of both physically-realistic simulations and real-robot demonstrations.

1 Introduction

Coordination is of fundamental importance in any system composed of multiple interacting agents. The survival of an ant colony, for example, depends on its ability to coordinate its division of labor in such tasks as foraging, tending the brood, and defending against invaders. Similarly, the effectiveness of a multi-robot system (MRS) on a given task is determined by the extent to which the robots are able to coordinate their actions. The study of coordination in the context of an artificial system, such as a MRS, allows one to systematically modify and augment the capabilities of the agents, the mechanisms by which coordination is achieved, and the types and complexity of tasks to be performed in order to gain insight into the relationships and dependencies among these characteristics. Insights gained through such a systematic study will result in improved methods for the design of coordinated MRS and may benefit the study and analysis of coordination in natural systems.

Toward this end, we have developed a formalism that provides a principled framework for precisely defining and reasoning about the intertwined entities involved in any task-achieving coordinated MRS – the world, task definition, and the capabilities of the robots, including action selection, sensing, maintenance of internal state, and inter-robot communication. Using this principled foundation, we are developing a suite of general methods by which to automatically synthesize controllers for robots constituting a MRS such that a given task is performed in a coordinated fashion. Each of these methods is directed toward the

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE **Toward a Multi-Robot Coordination Formalism**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Southern California,Computer Science Department,Los Angeles,CA,90089-0781**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

synthesis of a specific type of controller, which we taxonomize based on the following three characteristics: deterministic or probabilistic action selection (DA/PA), use of internal state or stateless (IS/NIS), and use of inter-robot communication (Comm/NComm). The resulting methods for principled controller synthesis across this taxonomy provide more than just pragmatic tools for building coordinated MRS. Given their formal grounding, they also serve as a means for systematically determining the fundamental limitations of each type of controller, understanding the inherent relationships among different controller types, and introducing novel insights into the general requirements necessary to achieve different forms of coordination. Thus, facilitating formal answers to fundamental questions such as: '*Under what conditions is it necessary to maintain internal state in order to achieve the desired coordination?*', '*Under what conditions is internal state alone insufficient?*', and '*When are the use of internal state and communication interchangeable?*'.

In our previous work, we have presented a method for the synthesis of DA-NIS-Comm controllers and defined situations in which communication is useful to achieve the necessary coordination [12]. Here, we present a novel method for the synthesis of a different type of controller, a DA-IS-NComm controller. We formally show when and why DA-IS-NComm controllers are useful in achieving the desired coordination and when they are insufficient. We present experimental validation of our formal approach to DA-NIS-Comm controller synthesis in a multi-robot construction task domain using extensive physically-realistic simulation experiments and limited real-robot demonstrations.

The remainder of this paper is organized as follows. In Section 2 we summarize relevant related work. In Section 3 we provide definitions and notation, including those for the task environment, task definition, and individual robot capabilities. In Section 4 we present a principled DA-IS-NComm synthesis procedure. In Section 5 we present experimental demonstration of our the synthesized controller in a multi-robot construction domain, both in physically-realistic simulation and in real-world robot experiments. In Section 6 we draw conclusions about this work and discuss future directions.

2 Related Work

The majority of work in MRS is empirical in nature. Formal work addressing synthesis and analysis of MRS coordination mechanisms includes Donald [6], which presents the derivation of information invariants aimed at defining the information requirements of a given task and how those requirements can be satisfied in a robot controller. Parker [20] extends the idea of information invariants by defining equivalence classes among task definitions and robot capabilities to assist in the choice of an appropriate controller class. Dudek et al. [7] present a taxonomy which classifies multi-robot systems based on communication and computational capabilities. Martinoli et al. [15] presents a general methodology by which the collective behavior of a group of mobile robots can be accurately studied using a simple probabilistic model. Balch [3] presents hierarchic social entropy, an information theoretic measure of robot team diversity in an effort to understand the role of heterogeneity in MRS coordination. Gerkey and Matarić [9] present a principled framework and an analysis methodology for the formal study of multi-robot task allocation. Lerman and Galstyan [14] present a mathematical model of the dynamics of collective behavior in a multi-robot adaptive task allocation domain. Alternative approaches to the synthesis of MRS controllers can be found in evolutionary methods [8] and learning methods [16, 19]. There also exist a number of MRS design environments, control architectures, and programming languages which assist in the design of task-achieving coordinated MRS [17, 2, 19, 1].

We validate our MRS controller synthesis method in a multi-robot construction domain. Related work in this area includes Bonabeau et al. [5], which uses a rule-based model in the construction of biologically-inspired structures. Bonabeau et al. [4] investigate the use of genetic algorithms to generate such rules and explores the relationship between the space of rules and resulting structures. In the area of construction by physical robots, Melhuish et al. [18] demonstrate how a group of minimalist robots can construct defensive walls using biologically-inspired templates. Wawerla et al. [21] present work on the comparison of different coordination strategies in the construction of simple 2D structures using a MRS. Our previous work Jones and Matarić [11] presents a method to automatically synthesize controllers for rule-based agents using local sensing and control in an intelligent self-assembly domain.

3 Definitions and Notation

We now provide necessary definitions. The *world* is the domain in which the MRS is expected to perform a defined task. We assume the world is Markovian and the state is an element of the finite set S of all possible states. The set of all robots is denoted by the finite set R. We assume all robots in the MRS are homogeneous. An action a_r performed in the world by a single robot r is drawn from the finite set A of all possible actions. An *observation* x made by robot r, drawn from the finite set of all observations X, consists of accessible information external to the robot and formally represents a subset of the world state. The world is defined by a probabilistic state transition function $P : S \times X \times A \times S \rightarrow [0, 1]$. That is, given a world state s at time t, a robot r making observation x and executing action a, and a world state s' at time $t + 1$, $P(s, x, a, s') = Pr(S^{t+1} = s'|S^t = s, X_r^t = x, A_r^t = a)$. We note that the world state transition function involves an observation because the tasks we consider are spatial in nature and the physical location where an action is performed is just as important as the action itself. In this representation, an observation x is equated with the spatial location where the action a is performed. Therefore, an action a executed upon the observation of x_i will transition the world differently than the same action a performed upon the observation of x_j. We define a *task*, assumed to be Markovian, as a set of n ordered world states $T_s = \{s_0, s_1, ..., s_n\}$ which must be progressed through in sequence. We assume the initial state of the world is s_0. We define *correct task execution* to be the case where, for all task states $s_i \in T_s$, $i < n$ the only actions executed by any robot are those that transition the world state to s_{i+1}. Once the world state is $s_n \in T_s$ the task is terminated. Therefore, we define an observation and action pair for a robot, x and a, to be correct for task state s_i if $P(s_i, x, a, s_{i+1}) > 0$. We assume that an observation x and action a cannot be correct for more than one task state. The probabilistic *observation function* $O(s, x) = Pr(X_r^t = x|S^t = s)$ gives the probability observation x will be made in state s by a robot r. Furthermore, we assume that an observation x may only be made at one physical location in the world in a state s. A robot's internal state value m at any time is a member of the finite set $M = \{m_0, ..., m_p\}$. Two probabilistic functions define a robot r's behavior in the world, known collectively as the robot's *controller*. The controller is comprised of an *action function* $A(x, m, a) = Pr(A_r^t = a|X_r^t = x, M_r^t = m)$ and an *internal state transition function* $L(m, x, m') = Pr(M_r^{t+1} = m'|X_r^t = x, M_r^t = m)$. Although the controller is modeled with probabilistic functions for generality, in this paper those functions are assumed to be binary; A and L will always be either 0 or 1.

4 Synthesis of DA-IS-NComm Controllers

In this section we present a systematic procedure by which to synthesize a DA-IS-NComm controller, a controller that uses deterministic action selection, maintains some amount of non-transient internal state, and does *not* have inter-robot communication capabilities. We also discuss the uses and limitations of such controllers in the facilitation of MRS coordination.

4.1 Synthesis

There are four high-level steps in the synthesis procedure: **1)** synthesize a baseline DA-NIS-NComm controller, **2)** initialize some important variables, **3)** identify situations in which internal state can be used to facilitate coordination, and **4)** appropriately augment the action and internal state transition functions given by the synthesized baseline DA-NIS-NComm controller. The full synthesis method is given by the procedure *Build_DA-IS-NComm_Controller* shown in Figure 1. All line numbers referenced in the remainder of this section refer to Figure 1.

Step 1: We synthesize a DA-NIS-NComm controller, a stateless, non-communicative controller with deterministic action selection, which we will augment with communication to synthesize the DA-NIS-Comm controller. The method to synthesize a DA-NIS-NComm controller is given by the procedure *Build_DA-NIS-NComm_Controller* in Figure 1. For each $s_i \in T_s$, the synthesis procedure adds a rule to the action function of the form $A(x, m_0, a) = 1$ such that x and a are correct for task state s_i. Since all actions require an internal state value of m_0, this is equivalent to a stateless controller.

However, such a DA-NIS-NComm controller contains room for error if x and a are correct for some task state s_i but there exists another task state s_j where x and a are *not* correct and $O(s_j, x) > 0$. In such situations, an MRS composed of robots with DA-NIS-NComm controllers cannot enforce the action sequence necessary for correct task execution. This is a common problem with purely reactive controllers in sequential task domains. In the DA-NIS-Comm synthesis steps that follow, we will incorporate the use of internal state to improve coordination in these situations. Due to sensing and action uncertainty, the addition of internal state cannot guarantee correct task execution, it increases its likelihood.

Step 2: Next, we initialize some relevant variables in lines 13-17. The variable $X_a(s_i)$ contains the set of all observations x for which there exists an action a such that x and a are correct for state s_i, for all $s_i \in T_s$. Informally, $X_a(s_i)$ contains the set of all observations that may be made in state s_i which will lead to an action. The set $V_a(s_i)$ will contain the index of the internal state value (i.e., 0 represents internal state value m_0) that a robot will need to have in order to execute an action in state s_i, for all $s_i \in T_s$. Initially, all values in V_a are assigned the same internal state value m_0 – this is equivalent to not using any internal state at all. Lastly, the set $O_a(s_i)$ will contain the observation, if any, that will be used to transition the internal state value in state s_i, for all $s_i \in T_s$. Initially, all values in O_a are assigned as $NULL$, since no internal state transitions are defined at that point.

Step 3: Lines 18-27 identify situations in which internal state can be used to improve coordination and assign appropriate values to the sets V_a and O_a. The basis for determining when internal state can be used to improve coordination is in identifying task states where an observation x in $X_a(s_j)$, where x and some action a are correct for s_j, can also be made in some earlier task state s_i for which x and a are not correct. We note that our synthesis method does not deal with the situation where there exists a task state s_p which occurs *later* than s_j and $O(s_p, x) > 0$, where x and a are not correct for s_p. This situation is addressed in Section 5.3.

In order to use internal state to disambiguate the observation of x between s_i and s_j, we identify an observation z that can only be made in task states which occur later than s_i and in at least one state before s_{j+1}. This observation, or absence thereof, can now be used to sufficiently disambiguate the observation of x in s_i and s_j. If x is currently being observed and z has been previously observed, then the task state must be s_j and the appropriate action can be performed. Otherwise, if x is currently being observed and z has *not* been previously observed, then the task state could either be s_i or s_j and the action x may or may not be correct.

Step 4: We synthesize the DA-IS-NComm controller by augmenting the DA-NIS-NComm controller synthesized in Step 1. This is accomplished by adding the internal state transition function and appropriately

```
procedure Build_DA-NIS-NComm_Controller()
  for all $a \in A, x \in X, m \in M$ do
    $A(x, m, a) = 0$
  endfor
  for all $s_i \in T_s$ do
    for all $a \in A, x \in X(O(s_i, x) > 0 \wedge P(s_i, x, a, s_{i+1}) > 0)$ do
      $A(x, m_0, a) = 1$
    endfor
  endfor
end procedure Build_DA-NIS-NComm_Controller

procedure Build_DA-IS-NComm_Controller()
  Build_DA-NIS-NComm_Controller()

  for all $s_i \in T_s$ do
    $X_a(s_i) = \{x_0, x_1, ..., x_n\}$ s.t. $\forall x \in X_a(s_i) \exists a(O(s_i, x) > 0 \wedge A(x, m_0, a) = 1)$
    $V_a(s_i) = 0$
    $O_a(s_i) = NULL$
  endfor

  for all $s_i, s_j \in T_s(i < j)$ do
    if $\exists x \in X_a(s_j)(O(s_i, x) > 0)$ then
      for $s_k = s_j$ downto $s_{i+1}$ do
        if $\exists z \nexists s_u(O(s_k, z) > 0 \wedge u < (i+1) \wedge O(s_u, z) > 0)$ then
          $O_a(s_k) = z$
          $\forall s_w(w >= k) \rightarrow V_a(s_w) = V_a(s_w) + 1$
        endif
      endfor
    endif
  endfor

  for all $s_i \in T_s$ do
    if $O_a(s_i) \neq NULL$ then
      $L(m_{V_a(s_i)-1}, O_a(s_i), m_{V_a(s_i)}) = 1$
    endif
    for all $x \in X_a(s_i), a \in A(A(x, m_0, a) = 1)$ do
      $A(x, m_0, a) = 0$
      $A(x, m_{V_a(s_i)}, a) = 1$
    endfor
  endfor
end procedure Build_DA-IS-NComm_Controller
```

Figure 1: Procedure for synthesizing a DA-IS-NComm controller.

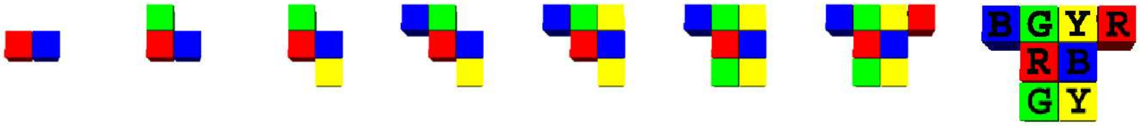

Figure 2: The sequence of world states, s_0 to s_6, defining a construction task, as seen from overhead (a view not available to the robots). On the far right is the final task state s_6 labeled with the brick colors.

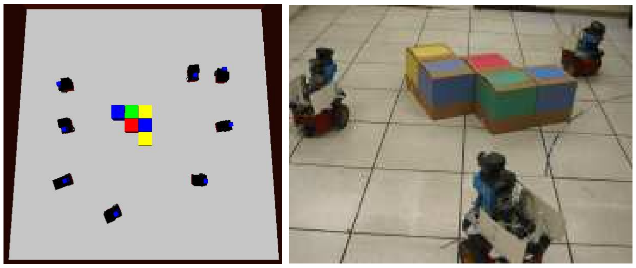

Figure 3: Left: Snapshot of an 8-robot experiment in simulation. Right: Snapshot of a 3-robot real-world experiment.

modifying the action function such that an action is not executed unless the required internal state value is present. The internal state transition function is constructed (Figure 1, lines 29-31) by mapping the internal state value $m_{V_a(s_i)-1}$ and observation $O_a(s_i)$ to the next internal state value of $m_{V_a(s_i)}$, for all $s_i \in T_s$. The action function is modified (Figure 1, lines 32-35) such that for each rule of the action function $A(x, m_0, a) = 1$ where x and a are correct for a state s_i is modified to become $A(x, m_{V_a(s_i)}, a) = 1$, where $m_{V_a(s_i)}$ is the required internal state value for task state s_i as determined in step 3. All probabilities not explicitly declared for the controller are 0.

We clarify that although in the synthesis procedure we discuss observations and internal state values in terms of specific task states, this does not mean that the resulting controller uses any explicit reasoning of underlying specific states. Rather, the resulting action and internal state transition functions do not make any use of task state values.

5 Validation: Coordination in Multi-Robot Construction

We experimentally demonstrated and validated our approach to the synthesis of coordinated MRS through the use of DA-IS-NComm controllers in a multi-robot construction task, both in physically-realistic simulation and on real robots. The construction task requires the sequential placement of a series of cubic colored bricks into a planar structure. For all examples used in this section, a brick's color is denoted by the letters `R`, `G`, `B`, or `Y`, which stand for `Red`, `Green`, `Blue`, and `Yellow`, respectively. The construction task starts with a seed structure, a small number of initially placed bricks forming the core structure.

Our simulation experiments were performed using Player and the Gazebo simulation environment. Player [10] is a server that connects robots, sensors, and control programs over the network. Gazebo [13] simulates

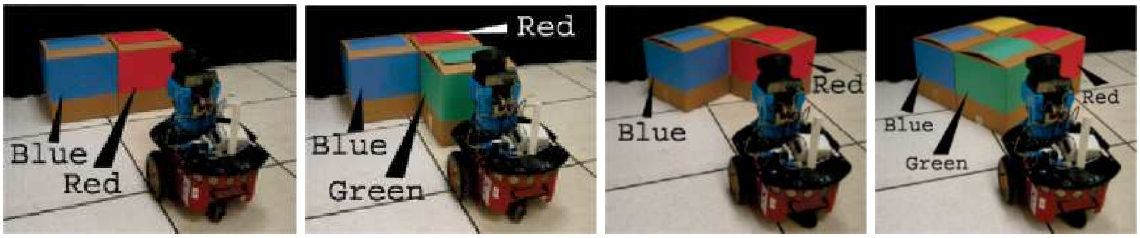

Figure 4: Example observations and actions in the construction domain. Top left: Robot in position to make observation `<FLUSH R B>`. Top right: Immediately after robot performs action `<G RIGHT FLUSH R B>`. Bottom left: Robot in position to make observation `<CORNER R B>`. Bottom right: Immediately after robot performs action `<G CORNER R B>`.

a set of Player devices in a 3-D physically-realistic world with full dynamics. Together, the two represent a high-fidelity simulation tool for individual robots and robot teams which has been validated on a collection of real-world robot experiments using Player control programs transferred directly to physical Pioneer 2DX mobile robots. In all simulation experiments, 8 robots were used, and in all real-world experiments, 3 robots were used. In simulations, the robots were realistic models of ActivMedia Pioneer 2DX mobile robots, in real-world experiments those physical robots were used. Each robot, approximately 30 cm in diameter, is equipped with a differential drive, a forward-facing 180 degree scanning laser rangefinder, and a forward-looking color camera with a 100-degree field-of-view and a color blob detection system. The bricks are taller than the robot's sensors, so the robots can only sense the local bricks on the periphery of the structure (i.e., robots do not have a birds-eye-view of the entire structure). Figure 3 shows snapshots of our simulation and real-world experimental setup.

We note that our robots do not have the ability to independently manipulate bricks during the construction process in simulation or with physical robots. To address this issue in simulation, when a robot wants to execute a brick placement action, it commands the simulator to place a brick of a given color at a given location relative to the robot's current pose. In real-robot experiments, we manually placed the appropriate brick in response to the robot's audible command (e.g., "Place yellow brick in the corner formed by the red and blue bricks directly in front of my position").

5.1 Formal Definitions for Construction Task

In order to ground the construction task in the formal framework presented in Section 3, we now define the world, task definitions, observations, and actions in the construction task domain. The *world* state is defined as a specific spatial configuration of bricks, including the color of each brick. A construction *task* is defined as a *sequence* of brick configurations (i.e., world states), providing a specific construction sequence.

Observations in the construction domain are made up of the spatial configuration and color of bricks in the field-of-view of the robot's laser rangefinder and color camera and within an appropriate range and bearing. Two categories of observations can be made. The first is two adjacent, aligned bricks. A situation in which such an observation would be made is shown in Figure 4 and is denoted as `<FLUSH R B>`. The second is two adjacent bricks forming a corner. A situation in which such an observation would be made is shown in Figure 4 and is denoted as `<CORNER R B>`. The observations `<FLUSH R B>` and `<FLUSH B R>` constitute two different observations in which the spatial relationship between the Red and Blue bricks are

Action Function	Internal State Transition Function
A(<FLUSH R B>, m_0, <G RIGHT FLUSH R B>) = 1	L(m_0, <FLUSH Y B>, m_1) = 1
A(<FLUSH B R>, m_1, <Y RIGHT FLUSH B R>) = 1	L(m_1, <FLUSH B Y>, m_2) = 1
A(<FLUSH R G>, m_2, <B LEFT FLUSH R G>) = 1	L(m_2, <FLUSH B G>, m_3) = 1
A(<CORNER G B>, m_3, <Y CORNER G B>) = 1	L(m_3, <FLUSH G Y>, m_4) = 1
A(<CORNER Y R>, m_4, <G CORNER Y R>) = 1	L(m_4, <FLUSH Y G>, m_5) = 1
A(<FLUSH Y B>, m_5, <R RIGHT FLUSH Y B>) = 1	

Table 1: Synthesized action and communication functions for the construction task shown in Figure 2. $m_0, m_1, ..., m_5 \in M$. All robots' initial internal state value is m_0. All probabilities not shown are 0.

switched. A similar point holds for the observations `<CORNER R B>` and `<CORNER B R>`.

Actions are the placement of individual bricks to the growing structure. We do not consider construction tasks in which robots may remove bricks from the structure nor those in which sub-structures consisting of multiple bricks may be connected together. Other actions performed by the robots, such as moving through the environment, do not affect the world state and therefore do not need to be explicitly considered. Three categories of actions can be executed. The first is the placement of a brick on the right side (from the perspective of the acting robot) of a pair of adjacent, aligned bricks. The immediate result of such an action is demonstrated in Figure 4 and is denoted as `<G RIGHT FLUSH R B>`. The second is identical to the first except that the brick is placed on the left side of a pair of adjacent, aligned bricks. This action is denoted as `<G LEFT FLUSH R B>`. The third is the placement of a brick in the corner formed by two other bricks. The immediate result of such an action is demonstrated in Figure 4 and is denoted as `<G CORNER B R>`.

5.2 Synthesized Controller

We applied our systematic method for synthesizing DA-IS-NComm controllers to the construction task shown in Figure 2. The synthesized action and internal state transition functions are given in Table 1. Figure 5 shows how the action and internal state transition functions are integrated into the controller. Since the `Avoid` and `Random Walk` behaviors do not change the world state, they do not impact the controller synthesis procedure.

The controllers for the construction task shown in Figure 2 were implemented on a group of 8 simulated robots. A total of 300 experimental trials were conducted in simulation. As expected, due to significant uncertainty in sensing and imperfect actions, each trial did not result in correct task execution. Over the 300 experiments, correct task execution was achieved in 31.5% of the cases. This represents a significant improvement over the stateless DA-NIS-NComm controller, which resulted in only 0.9% of experiments being correctly executed. For real-robot verification of the feasibility of the synthesized DA-NIS-Comm controller, we also implemented it on a group of three actual Pioneer 2DX mobile robots and successfully performed a limited number of real-world experiments. We note that the real-robot experiments were primarily performed in order to verify that the assumptions in the formalism and synthesis procedure are reasonable and realistic. The experiments were also performed to show that our formalism and synthesis method are not merely abstract concepts but successfully capture the difficult issues involved in real-world embodied MRS, thus validating a grounded and pragmatic tool for the formal description and synthesis of coordinated MRS.

```
procedure Execute_DA-IS-NComm_Controller()
  $m \leftarrow m_0$
  repeat forever
    $x \leftarrow$ current observation
    if $\exists m'(L(m, x, m') > 0)$ then
      $m \leftarrow m'$ with prob. $L(m, x, m')$
    else if obstacle nearby then
      execute obstacle avoidance
    else if $\exists a(A(x, m, a) > 0)$ then
      execute action $a$ with prob. $A(x, m, a)$
    else
      execute a random walk
    endif
  endrepeat
end procedure Execute_DA-IS-NComm_Controller
```

Figure 5: High-level DA-IS-NComm controller integrating the synthesized action and internal state transition functions for the construction task domain.

5.3 Discussion

A DA-IS-NComm controller synthesized by the procedure in Figure 1 is only one, and certainly not the only, way in which internal state can be used to facilitate coordination. However, it is a representative means of using internal state that is effective in a variety of situations. From the perspective of identifying and understanding the fundamental requirements of coordination, MRS composed of robots which make use of internal state but are not capable of communicating are quite interesting. In this way, we can begin to isolate and formally describe the uses and limitations of internal state in MRS coordination, thereby justifying the use of additional mechanisms such as communication.

In Section 4.1 we noted that our synthesis method does not deal with the situation where there exists a task state s_p that occurs *later* than s_j and $O(s_p, x) > 0$, where x and a are not correct for s_p. In such a situation, even in the absence of sensing and action uncertainty, the use of internal state cannot guarantee correct task execution; however, it could potentially be used to probabilistically approach correct task execution. This is due to the fundamental unobservability of the underlying world state resulting from the local nature of each robot's sensing capabilities. Internal state can be used to record observations that have been made, but it cannot be used to record observations that have not been made. This later case is what is required in order to disambiguate the observation of x between s_j and s_p.

The goal of our formal approach to coordination is both to serve as a practical tool for the synthesis of coordinated MRS and to aid in understanding the uses and limitations of various controller characteristics, such as the use of internal state or communication, toward more principled study and analysis of coordination in natural and artificial systems. We presented experimental results in a multi-robot construction domain that is similar in nature to the biologically-plausible wasp nest construction algorithms presented in [5]. Through our formal analysis of coordination as applied to the construction domain, we believe there can be a useful transfer of knowledge to the study of construction in wasp colonies. Leveraging formal knowledge gained about the uses and limitations of internal state in such a domain, one can begin to infer the mechanisms that may be used by the wasps to achieve coherent coordination in the construction process. As such, we can gain insight into the situations and characteristics in which such coordination mechanisms are effective and when they are likely to break down. This may be useful, for example, in explaining mistakes in a given construction process and understanding the range of constructions that are theoretically possible given the individual wasps' sensing and control capabilities.

6 Conclusions

Coordination is of fundamental importance in any system composed of independent, interacting agents. Whether the system is natural, such as an insect society, or artificial, such as a team of robots in a distributed multi-robot system (MRS). In order to effectively describe, analyze, or synthesize coordinated systems, a formal understanding of how coordination is achieved is necessary. In this paper, we described a framework for formally describing and reasoning about the issues involved in a coordinated system required to execute a sequential task. Using this formal grounding, we presented a principled method for automated synthesis of coordinated MRS. Specifically, we synthesized controllers for robots in a MRS which may maintain internal state but are not capable of communication. These controllers, when executed by all robots in a MRS, correctly execute a given sequential task. We formally explained how the use of internal state can be used to improve coordination and identified limitations on the effectiveness of internal state. We experimentally validated our synthesis procedure using a multi-robot construction domain, in extensive physically-realistic simulations and in limited real-robot experiments.

7 Acknowledgments

This work was supported by Defense Advanced Research Projects Agency (DARPA) Grant F30602-00-2-0573.

References

[1] R. Alur, R. Grosu, Y. Hur, V. Kumar, and I. Lee. Modular specification of hybrid systems in charon. In *Proceedings of the 3rd International Workshop on Hybrid Systems: Computation and Control*, pages 6–19, Pittsburgh, PA, Mar 2000.

[2] R. Arkin and T. Balch. Aura: Principles and practive in review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9:175–189, 1997.

[3] T. Balch. Measuring robot group diversity. In T. Balch and L. E. Parker, editors, *Robot Teams: From Diversity to Polymorphism*, pages 93–135. AK Peters, 2002.

[4] E. Bonabeau, M. Dorigo, and G. Théraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[5] E. Bonabeau, G. Théraulaz, E. Arpin, and E. Sardet. The building behavior of lattice swarms. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 307–312. MIT Press, 1994.

[6] B. R. Donald. Information invariants in robotics. *Artifical Intelligence*, 72(1–2):217–304, 1995.

[7] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots 3*, (4):375–397, 1996.

[8] D. Floreano. Patterns of interactions in shared environments. In *Proceedings of the Second European Conference on Artificial Life*, pages 347–366, Brussels, Belgium, May 1993.

[9] B. Gerkey and M. Matarić. Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3862–3867, Taipei, Taiwan, Sep 2003.

[10] B. Gerkey, R. Vaughan, K. Stoey, A. Howard, G. Sukhatme, and M. Matarić. Most valuable player: A robot device server for distributed control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1226–1231, Maui, Hawaii, Oct 2001.

[11] C. Jones and M. Matarić. From local to global behavior in intelligent self-assembly. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 721–726, Taipei, Taiwan, Sep 2003.

[12] C. Jones and M. Matarić. Automatic synthesis of communication-based coordinated multi-robot systems. Technical report, University of Southern California Center for Robotics and Embedded Systems, CRES-04-007, 2004.

[13] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. Technical report, USC Center for Robotics and Embedded Systems, CRES-04-002, 2004.

[14] K. Lerman and A. Galstyan. Macroscopic analysis of adaptive task allocation in robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1951–1956, Las Vegas, Nevada, Oct 2003.

[15] A. Martinoli, A. Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: From probabilistic modeling to experiments with real robots. *Robotics and Autonomous Systems*, 29:51–63, 1999.

[16] M. Matarić. Designing and understanding adaptive group behavior. *Adaptive Behavior*, 4:51–80, Dec 1995.

[17] M. Matarić. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2–4):321–331, Dec 1995.

[18] C. Melhuish, J. Welsby, and C. Edwards. Using templates for defensive wall building with autonomous mobile ant-like robots. Technical report, Towards Intelligent Mobile Robots, University of Manchester, Department of Computer Science, UMCS-99-3-1, Bristol, UK, Mar 1999.

[19] L. Parker. Alliance: An architecture for fault-tolerant multi-robot cooperation. *Transactions on Robotics and Automation*, 14(2):220–240, Apr 1998.

[20] L. Parker. Toward the automated synthesis of cooperative mobile robot teams. In *Proceedings of SPIE Mobile Robots XIII*, volume 3525, pages 82–93, Boston, MA, 1998.

[21] J. Wawerla, G. S. Sukhatme, and M. Matarić. Collective construction with multiple robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2696–2701, Lausanne, Switzerland, Sep 2002.