

HAMR: A Hybrid Multi-Robot Control Architecture

Daylond Hooper and Gilbert Peterson

Department of Electrical and Computer Engineering

Air Force Institute of Technology*

Wright-Patterson AFB, OH 45433

Email: {daylond.hooper, gilbert.peterson}@afit.edu

Abstract

Highly capable multiple robot architectures often resort to micromanagement to provide enhanced cooperative abilities, sacrificing individual autonomy. Conversely, multi-robot architectures that maintain individual autonomy are often limited in their cooperative abilities. This article presents a modified three layer architecture that solves both of these issues. The addition of a Coordinator layer to a three-layered approach provides a platform-independent interface for coordination on tasks and takes advantage of individual autonomy to improve coordination capabilities. This reduces communication overhead versus many multi-robot architecture designs and allows for more straightforward resizing of the robot collective and increased individual autonomy.

Introduction

For a multi-robot system (MRS) to provide retention of individual capabilities and the addition of capabilities for meeting multi-robot tasks, a control architecture must emphasize both individual autonomy (independence) with collaboration and cooperation capabilities (coordination). Heterogeneous MRSs provide a significant benefit if coordination is adequately provided, since the work is distributed among the robots in the group and each agent may possess unique capabilities. Many multi-robot control architectures focus on coordination, often at the cost of individual autonomy ((Simmons et al. 2000) for example). Decentralized architectures of this nature still have a single point of failure, which makes for a less robust system. Attempts to provide both coordination and individual autonomy often use extensive communication (Klavins 2002). This monopolizes communication channels and risks broadcasting of information to undesired parties. Furthermore, many multi-robot systems are scalable, but unable to handle the addition of a previously unknown robot type. To provide a heterogeneous system that

emphasizes individual autonomy, retains low communication overhead, and is well suited to resizing of the collective, we present HAMR, the Hybrid Architecture for Multiple Robots.

HAMR Implementation

HAMR builds on the three layer robot architecture paradigm (Gat 1998), allowing for high-level planning and behavior-based execution of tasks. The traditional three-layer paradigm is composed of Controller, Sequencer, and Deliberator layers. The Controller layer manages behaviors and their implementation. The Sequencer enables and disables behaviors at the appropriate time, maintains internal state to alert the Deliberator if a new plan is needed, and generates the replanning signal for the Deliberator. The Deliberator performs high-level reasoning tasks that include task decomposition, task allocation, and planning. The Deliberator also generates new tasks from sensor data processing that occur during the performance of a task. For a multirobot system, the task allocation responsibility of the Deliberator needs knowledge of the task allocations of other group members. Each robot, upon receipt of the other robot's utilities, individually performs the processing to generate global task or resource assignments and individually determines their own allocation, communicating the results to the other members. The Coordinator expands on these capabilities, aiding in prioritization of tasks and managing negotiations for multi-robot applications.

Coordinator

The Coordinator is a new component that enables multi-robot control and contributes to a robust collection of mostly independent robots. It acts as the central point for managing communication: broadcasting the robot's task utilities, transmitting state data, and receiving and processing messages from other robots. In addition to communication, the Coordinator 1) generates utility values for tasks to aid the Deliberator in decision making for task allocation, 2) monitors the world state for state changes which justify an update to the system's task allocation, and 3) maintains the world state as it pertains to the multi-robot aspects, including modeling of the other group members, environmental data from other group members, and global task and resource allocation records.

*This research was sponsored by an AFRL Lab Task 06SN02COR from the Air Force Office of Scientific Research, Lt Col Scott Wells, program manager. The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense or the U.S. Government.
Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Multi-robot task allocation provides robots with the ability to perform multiple tasks simultaneously. The Coordinator uses the Sequencer-generated behavior sets, expected resource expense, and general fitness estimates stored in the state to assign a utility to the task. This utility describes the robot's estimate of its own usefulness, based upon the robot's evaluation of its capabilities and other task-dependent costs. When the utility is generated, the Coordinator broadcasts it to the other group members. Once all utilities are collected, the Deliberator evaluates the task and determines appropriate allocation of tasks and resources using a "highest utility wins" approach. The task is stored in the state with an associated collection of utility values and the robot assigned to each task. This assignment is calculated independently in each robot's Deliberator, and plans for execution are generated upon assignment.

State monitoring allows for detection and addressing of failures, along with streamlining of reassignment upon a failure. Specifically, the state monitoring responsibility of the Coordinator stores positions communicated from the other robots, task assignments, and resource allocations of the other group members. The capabilities of the other robots are not stored since each robot determines its own capabilities independently, enhancing the mutual independence of the system members. These functions are covered by the state monitoring and management component of the Coordinator, as shown in Figure 1. Whenever the state changes in an appreciable manner to prevent completion of the task at hand or to the global task allocation, the Coordinator alerts the Deliberator and a new allocation is determined.

The third responsibility of the Coordinator is maintaining the modeling of the other group members, important environmental data, and maintain global task and resource allocation records. This is an internal processing of the communication messages only, since sensor data are handled by the Controller. These aspects are handled by both the communication management and state monitoring components in Figure 1. Lists of tasks, along with their associated plans, allocated resources, and the robots assigned to them are kept in the state. The tasks are removed upon notification of completion, and any associated resources are freed. The models of the other group members include their task allocations and the estimated status of the robot in general (active or disabled).

The Coordinator also maintains a list of group members. Each new robot subscribes to the group by announcing its existence to the current members. The group members then incorporate the new robot's utilities into their task allocation analysis. After a certain time passes with no response (removal or failure of a robot), the robot is removed from the active robots list and tasks are reassigned if necessary. If the removed robot's failure is temporary, it can re-subscribe to the group and continue activity as normal. This mechanism allows for dynamic scalability. A robot with any physical construction and any set of behaviors can subscribe to the group, volunteer its utility, and be included in any task allocation that occurs. Each robot can then be developed separately and deployed to the group when ready without

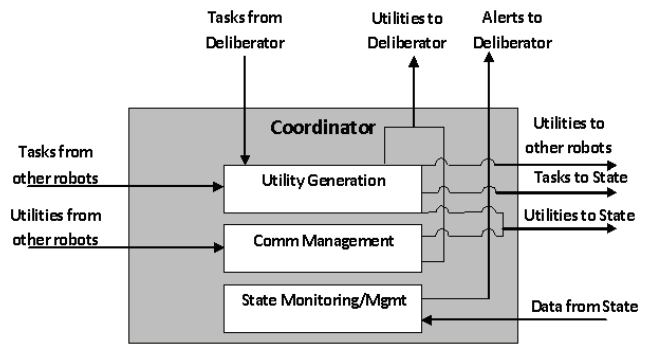


Figure 1: The internal structure of the Coordinator.

changing the rest of the group.

Conclusion

HAMR provides a number of key advancements: it provides coordination capabilities through an emphasis on individual autonomy, it contributes to coordination with low communication requirements, it provides a system that can be tasked, and enables mutual independence for all the robots in the collective. It takes advantage of this independence to provide a straightforward mechanism for continued operation when the collective size is changed. In addition, HAMR provides low communication overhead since all communications are conducted through the Coordinator and the only required communications are those used to coordinate on tasks and assign utilities. Each robot in HAMR is highly independent, yet the collective is cooperative. The robots can coordinate tightly since communication loads are reduced. The architecture is robust, since any task that falls within the skillset of the remaining agents will be completed eventually, and the removal of one or more agents from the group does not cause group failure. Finally, HAMR is extensible. The modularity of the underlying architecture provide a straightforward mechanism for introducing new skills to agents, and the other agents need not be informed of these changes. HAMR is more robust than the previous approaches in the face of robot failure, changing of the collective size or composition, and congested communication channels. It appropriately handles both loosely and tightly coupled tasks without sacrificing dynamic scalability.

References

- Gat, E. 1998. *Artificial Intelligence and Mobile Robots*. AAAI. chapter 8: On Three-Layer Architectures.
- Klavins, E. 2002. Communication complexity of multi-robot systems. In *Fifth International Workshop on the Algorithmic Foundations of Robotics*, 275–292. Springer-Verlag.
- Simmons, R.; Singh, S.; Hershberger, D.; Ramos, J.; and Smith, T. 2000. First results in the coordination of heterogeneous robots for large-scale assembly. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*.