Controlling a Mobile Robot Herd: Theory and Practice

Maja J Mataric
MIT Artificial Intelligence Laboratory
545 Technology Square #721
Cambridge, MA 02139
maja@ai.mit.edu

Abstract

This paper discusses the difficulties of analyzing and synthesizing intelligent collective behaviors for a collection of mobile robots. It reviews available approaches and presents a bottom-up experimental alternative consisting of a set of robust global behaviors based on simple local rules. These basic behaviors are designed to be sufficient and additive, in order to be combined into more complex collective task achieving behaviors. The paper also describes the role of interaction in a multi-agent environment, and ways of analyzing and predicting it. Finally, we present the experimental set up, consisting of twenty physical mobile robots, on which the interference studies were performed and the basic behavior repertoire was developed and tested.

Introduction

A methodology for principled synthesizing and analyzing robot behavior is one of the key unsolved problems in robotics and AI. Behavior synthesis in most cases consists of trial and error, and analysis is difficult since no good tools exist for accurately modeling real systems.

Our work is concerned with synthesizing and analyzing the behavior of not one but a collection of robots situated in the real world. The complexity of the problem is compounded by the interactions among the robots in addition to the interactions of the robots with their environment. In this paper we will show why a traditional top-down approach to designing behavior is not effective in the multi-robot domain, and present an alternative.

One of the goals of our work is to develop a set of operators for designing collective robot behaviors. These can serve as a basic behavior repertoire from which complex collective behaviors can be constructed.

In order to understand and formalize interactions among robots, it is necessary to observe them in a realistic environment. Theory about robot behavior cannot be developed without practice, nor can practice be generally useful without theory. Consequently, we are developing these two in parallel, and refining each with the other. This paper gives a qualitative description of the two aspects of the project.

Related Work

Designing intelligent, robust, and reliable behavior for a single agent is a topic of continued research in robotics and AI. As it remains a difficult problem, it is prudent to question whether it is useful to consider the multiple agent problem before the single agent case is better understood. In our research we hope to demonstrate not only that the multi-agent problem is relevant and manageable, but also that the existence of other agents can be used to simplify individual agent control.

The problem of controlling a collection of agents can be viewed at two levels: 1) the individual agent level, and 2) the collective level. The levels are interdependent as the design of one is strongly influenced by the other. This section describes the main approaches to individual agent control, and their extensions and applicability to the collective level.

At one extreme of the agent control spectrum lie traditional planner-based strategies which employ a centralized world model for verifying sensory information and generating actions in the world (e.g. [Giralt et al 84], [Chatila and Laumond 85], [Moravec and Cho 89]). The information in the world model is used by the planner to produce the most appropriate sequence of actions. As a compromise between purely reactive and planner-based approaches, hybrid architectures (e.g. [Arkin 89], [Payton 91], [Connell 91]) employ a reactive system for low-level control, and a planner for higher-level decision making. These approaches separate the control system into two or more communicating but otherwise independent parts. Following this framework, multi-agent systems have been developed by [Caloud et al 90] and [Noreils 90] applying a traditional planner-based control architecture for distributing the task over the agents.

In a state-based paradigm, extending from a single to a multi-agent problem results in expanding the global state space. Within this approach, a collection
of agents is a single system with each of the agents representing a part of the global state. However, the global state space is exponential in the number of agents. Additionally, the uncertainty in perceiving the state grows with the increased complexity of the environment. This exponential growth of the planning space makes global on-line planning intractable for any non-trivially sized collection of agents. Further, centralized planning requires communication between the agents and the controller, and the bandwidth grows with the number of agents. Consequently, traditional planner-based architectures for control do not appear well suited for problems involving multiple agents, in particular robots demanding real-time response based on uncertain sensory information.

In contrast to state space planner-based strategies, purely reactive approaches implement the control strategy as a collection of condition-action pairs with minimal state (e.g. [Brooks and Connell 87], [Agre and Chapman 87], [Connell 90]). These systems maintain no internal models, perform no search or planning, and merely lookup and command the appropriate action for each set of sensor readings. They rely on direct coupling between the sensors and actions, and a fast feedback loop through the world. Purely reactive strategies have been shown to be effective for a variety of problems which can be well specified at design time ([Schoppers 87]), but are inflexible at run-time due to their inability to store information dynamically ([Mataric 92]). A solution to a collective control problem in a purely reactive architecture would have no centralized controller at run time. It would consist of a collection of reactive agents each executing local task-related rules and relying only on local sensing and communication. Since all control in such a distributed system is local, it scales well with the number of agents. However, it requires a thorough understanding of the global consequences of the local interactions.

Behavior-based strategies lie between the purely reactive and the planner-based extremes [Brooks 86], [Rosenschein and Kaelbling 86], [Agre and Chapman 87, 90], [Maes 89]. Although often confused in the literature, purely reactive and behavior-based strategies differ fundamentally. Much like reactive systems, behavior-based systems consist of a collection of parallel, concurrently executing behaviors devoid of a centralized reasoner, but the behaviors are more powerful than purely reactive rules. While behavior-based systems embody some of the properties of reactive systems, their computation is not limited to lookup but may both store and compute with arbitrary amounts of internal representation [Mataric 92a]. Scaling the behavior-based approach to a collection of agents is similar to the reactive solution in that no global controller exists. The individual agents programmed in behavior-based style can be more computationally powerful that reactive agents, thus making the analysis more difficult.

This paper describes an approach to synthesizing global collective behaviors that applies to both reactive and behavior-based control at the individual agent level.

Behavior Analysis of Multi-Agent Systems

Analyzing and predicting the behavior of a single situated agent is an unsolved problem in robotics and AI. Even highly constrained domains have been shown to be intractable, and realistic worlds usually do not contain the structure, determinism and predictability necessary for formal analysis [Lozano-Pérez et al 84], [Canny 88], [Brooks 90b, 91].

Predicting the behavior of a multi-agent system is even more complex than the single-agent case. The difficulty in analyzing comes from two properties intrinsic to complex systems:

- the actions of an agent depend on the states/behaviors of other agents,
- the behavior of the system as a whole is determined by the interactions among the agents rather than by individual behavior.

In general, no satisfactory solution exists for predicting the behavior of a system with nontrival interactions components. In contrast to physical particle systems which consists of large numbers of simple elements, multi-agent robotic systems are defined by comparatively small groups of much more complex agents. Statistical methods used for analyzing particle systems do not apply as they require minimal interactions among the components [Weisbuch 91]. While systems with large numbers of simple and simply interacting components can be analyzed this way [Wiggins 90], no tools are available for systems consisting of comparatively few but more complex components with complex interactions.

Instead of attempting to analyze arbitrary complex behaviors, this work focuses on providing a taxonomy of analyzable behavior primitives. We begin by designing a set of tools for behavior synthesis and analysis for a particular type of an interacting system: multiple autonomous agents moving in the plane. The primitives provide a type of a programming language for designing complex control programs.

Basic Interaction Operators

This work builds on the simplest, most basic types of inter-robot interactions, and their global consequences. Interactions between individual agents need not be complex to generate complex results. In our experiments to date we have designed the following behaviors:

- Collision Avoidance: the ability of an agent to avoid colliding with anything in the world. Two distinct strategies can be devised; one for other agents
of the same kind, and another for everything else that the robot might perceive.

- **Following**: the ability to stay behind or along side of another agent without colliding.

- **Dispersion**: the ability of a group of agents to spread out over an area in order to establish and maintain some predetermined separation.

- **Aggregation**: the ability of a group of agents to gather in order to establish and maintain some predetermined distance. This behavior is a dual of dispersion.

- **Homing**: the ability of one or a group of agents to reach a goal region or location.

- **Flocking**: the ability of a group of agents to move as a coherent aggregate without prespecified leaders and followers. Flocking includes components of collision avoidance, following, dispersion and aggregation.

The above collective behavior primitives serve as building blocks for designing more complex behaviors. For example, herding can be constructed as a combination of flocking and homing. Similarly, a foraging task involving finding and collecting pucks in an open area can be constructed from dispersion and homing, with the use of following and flocking in areas of high density. Collection can then be used as a subtask in another application, such as cleaning very large spaces with the use of following and flocking in areas of high density. Collection can then be used as a subtask in another application, such as cleaning very large spaces or buildings with separate rooms.

Two key requirements must be satisfied by the basic behavior set:

- It must be complete or at least sufficient.
- It must be simply additive.

A basic behavior set is complete if it can generate all possible more complex global behaviors. Since this set is infinite, the property is difficult to establish analytically. A behavior set is sufficient if it can produce all attempted global behaviors for the tested domain. The behaviors we described are sufficient for generating goal-driven collective behaviors within our domain.

A behavior set is simply additive if the constituent behaviors can be combined into temporal sequences without producing further higher-level interactive effects. While it is in general difficult to guarantee this property, it can be specifically designed into a system, as it is in the case of our basic behavior set. The behaviors cannot be strictly temporally isolated, and will overlap. For example, a part of a multi-agent system may be performing a following behavior, while another part may be dispersing. If the two parts come into contact, they must either preserve their individual behaviors or one must take precedence over the other. In our work, the global behaviors are additive because they are designed as a part of a fixed precedence hierarchy implemented by the agents' local rules. Consequently, for each local interaction, only one global consequence is possible. The task determines the structure of the particular hierarchy.

The design of the individual primitives and their combinations is heavily influenced by the system property we call interference.

**Prediction of Interference**

Interference is one of the main problems in programming agents in finite space. It is defined as a particular type of interaction that forces an agent to deviate from the optimal, goal-driven behavior, in order to deal with immediate, local influence. For example, an agent is being interfered with if it is heading toward a goal but turns away in order to avoid another agent.

In order to solve the multi agent problem for any given task the designer must find a way to effectively distribute the task over multiple agents, so as to ensure that the interference among agents does not overwhelm the benefit of distributing the task. Predicting interference is an integral part of both synthesizing and analyzing collective behavior.

Attempting to precisely predict the amount of interaction is equivalent to trying to predict the exact behavior of the system. However, it is possible to make an approximate estimate of the amount of interference expected in a system based on the number of agents and their footprint. The footprint is the sphere of influence, or the range in which an agent can influence another, usually based on the agent's physical shape, motion constraints, and the sensor range and configuration.

Estimating interference is important for determining a number of key constraints on the feasibility of a particular task and agent configuration. For instance, a simple interference computation can predict how much physical space is necessary for a system to perform the task.

Accurate estimation of interference allows for determining which parts of the task will produce the most interaction among the agents, and consequently, which parts of the task require interactive and possibly cooperative control strategies. Homing illustrates this point; the level of interference increases drastically around the goal region. Consequently, it is in this area that collision avoidance protocols are most critical.

We have shown that a greedy solution to homing will degrade with the growing number of agents in the environment, precisely due to interference. To decrease interference, some cooperative behavior, in this case flocking, outperforms the individualistic approach. Although flocking behavior involves a compromise between individual and group goals, which may make an individual agent's path suboptimal, the global solution is more efficient.

Estimating interference is fundamentally a space occupancy issue. Using only the number and size of the agents, a simple mean free path computation can be used to estimate how many collisions are expected.
between the agents executing a random walk. This measure can be used to estimate how much space, on average, is required for the system, irrespective of its task. By bringing the spatial constraints of the task into the computation, we can estimate the distribution of the spatial requirements over the different parts of the task, i.e. plot the expected interference over the lifetime of the task. This computation indicates what parts of the behavior must be best designed to deal with interference. For example, during the homing behavior interference is highest during the later stages when a sufficiently large number of the agents has aggregated around the goal region. In contrast, interference is most pronounced during the initial stages of dispersion.

In the next section we describe the experimental environment used to investigate interference effects as well as the basic interaction operators on a collection of physical robots.

The Experimental Environment

In order to test the validity and the effectiveness of the proposed basic behavior set, we implemented it on a collection of twenty physically identical mobile robots. Each robot is a 12"-long four-wheeled vehicle, equipped with piezo-electric bumper sensors, infra-red collision and puck detection sensors, and a two-pronged forklift gripper for picking up, carrying, and stacking pucks (figure 1).

The robots are also equipped with radio transmitters and receivers for triangulating their position and exchanging data. The radio system is used for data gathering, as well as for simulating additional sensors. In particular, radios are used to distinguish robots from other objects in the environment, an ability which cannot be implemented with the on-board IR sensors. The flexibility of the radio system allows for testing a variety of communication parameters.

All experiments are run fully autonomously with all of the processing and power on board. The robots are programmed in the Behavior Language, a parallel programming language that compiles into a collection of augmented finite state machines [Brooks 91]. All of the above described collective behaviors have been implemented and tested on the robots. Detailed descriptions of each of the basic behaviors and related experimental data are presented in [Mataric 92b].

Summary

Behavior synthesis is necessarily a process of trial and error. Programs are generated, empirically tested, and incrementally improved. The exact behavior of an agent situated in a nontrivial world containing real error and noise, and executing even the simplest of programs is impossible to predict. The problem of unpredictability becomes even more acute in multi agent systems since the combinatorial space of possible global behaviors is exponential in the number of agents.

Establishing a taxonomy of basic types of interactions will provide a tool for predicting the qualitative state of a multi agent system. From the given behavior primitives, the types of results that can be expected for each type of interaction can be tabulated. Consequently, the process of behavior design can be made more systematic by selecting the appropriate primitives based on the task and agent specifications.

We have discussed the role of interaction in multi-agent systems, and some tools for predicting it. Further, we have described a set of basic behaviors which can be used to structure and simplify the process of generating useful collective behaviors for the mobile robot domain. The given basic behaviors offer a set of qualitative operators with associated global outcomes, but in order to make them useful for a particular system, they must be grounded in the system's mechanics, sensors, and the control program. We have demonstrated them on a collection of physical mobile robots, and are currently testing their sufficiency and their additive properties by implementing a number of more complex behaviors.

Acknowledgements

The research reported here was done at the MIT Artificial Intelligence Laboratory. Support for this research was provided in part by the Jet Propulsion Laboratory contract 959333 and in part by the Advanced Research Projects Agency under Office of Naval Research grant N00014–91–J–4038.
Many thanks to Matt Marjanovic and Owen Wessling for making the unruly robot herd behave in spite of many hardware hurdles. Thanks to Mike Bolotski for insightful comments on an earlier draft of this paper.

References


