Abstract

Simulators used in research on autonomous mobile robots have been criticized for their tendency to change the nature of the problems the robot control architecture has to solve. In this paper we address those arguments, and find that under certain conditions simulators can be a valuable tool to supplement research with physical robots. We conclude with guidelines for the successful design and use of simulators in research on mobile robots.

Introduction

There is much active work in the field of robot control architectures for mobile robots. Some have chosen to supplement or replace work on physical robots with research using simulators—software programs designed to model the interaction of a robot with its environment. In some cases simulators are motivated strongly by the physical characteristics of a particular robot and environment. More often, simulators idealize and abstract certain parts of the problem.

The use of simulators as a substitute for experiments with physical robots has been roundly criticized [3; 4]. In this paper, we will consider these criticisms, as well as positive reasons for using simulators. We go on to discuss some existing simulators and the ways in which they have been used. We conclude with some principles for the appropriate design and use of simulators in mobile robot research.

Simulators Can Be Misused

Mobile robot simulators provide many features convenient for research, some of which will be discussed below. Despite their advantages simulators have been criticized for making it easy to solve some difficult problems, making it difficult to solve other easy problems, and for creating false decompositions of robot control problems. We shall consider these criticisms in turn.

In [3], Brooks argues that simulation “requires a constant feedback from real experiments to ensure that it is not being abused.” He goes on to say that simulators create a temptation to simulate the perceptual system, creating false decompositions which lead researchers to work on problems they claim will be integratable into a general framework. In [4], Brooks adds that cellular representations of space are problematic in simulations of physical robots.

We agree that real robots and environments must be used during the design of simulators which attempt to model them. It is plausible, however, that once a simulator has been demonstrated to model a certain robot and domain in many respects, that simulator could then be used for extended periods without reference to the physical robot. It is important that research which claims to be applicable to physical robots be tested on such robots. However, the advantages of working in simulation, discussed below, may make a substantial reliance on simulators a worthwhile alternative.

As for the criticism that simulators tempt researchers to create false decompositions into which their research will fit, this is a problem not just with simulators but with work in all fields on “part of a complete AI system,” including work on vision, robotics, planning, reasoning, and learning. Our sense, however, is that scientists in these fields are coming to understand better and better the need to make their claims accurate and not overgeneral. In some cases, such as Brooks’ work, this leads them to design different decompositions of the problem. In other cases, such as work on planning, this leads researchers to back off from their claims of generality to offer their work instead as solutions to more restricted problems.

Brooks’ argument that cellular representations of space are unsatisfactory is compelling. Certain computational domains derive much of their interest from just the interactions and constraints that make cellular representations unsuitable for mobile robot simulators. The use of these representations in such domains is fine, but some have tried to extend the use of cellular spaces to robot control problems, particularly in the field of Distributed AI (see, for example [5]). We agree with Brooks that such simulations are unlikely to model physical robots well enough to be useful research tools, and we recommend continuous space simulation.

We feel that it is important to credit scientists with the responsibility to use their tools wisely. A simulator is a powerful tool that clearly has the potential for misuse, but with increased familiarity, we believe researchers will come to understand the new responsibilities that come with the use of a simulator. In fact, we argue below that one of the advantages of simulators is that they create increased accountability by encouraging more independent verification and subsequent reuse of results. Note also that the use of simulators comes with much higher responsibility for clear separation between reporting of theoretical results, exper-
Another criticism of simulators is that it is very hard to write a good one. If we understood the domain well enough to build an accurate simulator, the argument goes, we could use that understanding directly to write programs to control robots in that domain.

This may actually be a good reason to build simulators, not to avoid them. If we can gain some understanding of the domain by working hard to simulate it accurately, that understanding can be valuable when it comes time to write control programs. This answer does not address the challenge of making good models. We argue that good programmers should be able to build good models, and that more important than simulating the world exactly is understanding the limitations of the simulation.

Simulators Can Be Useful

If the responsibilities mentioned above are heeded, simulators can be a powerful tool for the development of robot control architectures. Their use should increase the verification and incremental progress that will make work in this field science rather than engineering. Simulators will create an inexpensive, accessible development environment, and can be useful for experiments with new robot configurations before the robots are built. For all of these reasons, simulators can play an ever more important role in mobile robot research.

Scientists using simulators can share code, robot designs, and error models with one another. Research results can be more easily verified by other researchers with access to the same simulator software. Results can be reported in the literature with reference to the shared simulated environment with which other researchers are familiar. In this way, a well-designed simulator may provide some "benchmark" tests for mobile robot control architectures, useful for comparisons between different architectures.

One criticism we have of current work on the design of autonomous mobile robot architectures is the large quantity of completely novel architectures which are proposed each year. In most other sciences, great value is placed both on repeating experiments to verify the results of others, and on building incrementally on the results of others. One feature of work in mobile robots which may contribute to this problem is that results with physical robots are very expensive and difficult to reproduce. Another feature is that the standards for reporting are often not high enough for readers of an article to actually implement or perform the experiments suggested.

We believe that widespread use of simulators to supplement work with physical robots will help the field by making more experiments repeatable, facilitating dialogue about the details of the experiment and the assumptions it makes. It will also enable researchers to build on the implemented ideas of their peers directly, actually reusing simulated robots, domains, and control architectures.

Simulators provide an inexpensive development environment. This can be especially important for laboratories in which access to physical mobile robots is a very limited resource. Furthermore, hardware has lots of problems which it is perfectly reasonable to simulate away. These include physical connector problems, short battery life, and the difficulty of reconfiguring the sensors and effectors on a physical mobile robot.

Following this line, simulators may also be useful for experimenting with proposed designs for robots to see which are useful for certain tasks. An example is placement of sonars so a robot can effectively pass through a doorway. If the environment, mobile base, and sonars were well modelled, a simulator would be a good testbed for trying different robot configurations.

We have seen that simulators have the potential to be valuable as research tools. Let's go on to look at some existing simulators, and see the extent to which each meets the goals just outlined.

Existing Simulators

We discuss here a few simulators for mobile robots. This list is not meant to be complete or even representative. Rather, it suggests the character of some of what has been done with simulators in mobile robot research.

Realistic Simulators

The simulators described below have in common the goal of realistic simulation of particular physical mobile robots and their environment.

Erann Gat [6] implemented and used a simulator for some experiments as part of his Ph.D. thesis on his ATLANTIS architecture. In this case, the simulator modelled an existing physical robot. Gat provides a good discussion of the appropriate use of simulators, and follows his own advice and our recommendations in the clarity and quality of his reports on experiments performed with the simulator.

Jonas Karlsson and Patrick Teo have implemented a simulator called Botworld [9] for Nils Nilsson at Stanford University to demonstrate his work on Action Networks and Teleo-Reactive Sequences. This program uses a client-server model, allowing robot control programs to be written in other languages and run on different machines. The Botworld simulator is a good model of some aspects of a frictionless 2-d navigation and manipulation problem being studied with physical robots in the Aeronautics and Astroautics department at Stanford, but a simplistic model of other aspects. In this case, the physical robot system is so carefully engineered that a simulation can accurately model many aspects of the problem. So far, this work has not addressed the issue of the effectiveness of these strategies on low-level navigation of physical robots.

EDDIE is a testbed simulator written at Carnegie Mellon University for experiments on outdoor road-following and navigation. This simulator provides primitive sensor functions which approximate the mid-level output from vision and laser range-finding sensors. It has been advocated by its authors as a general purpose testbed which can be used for other work on mobile robots. While it is well-designed for its purpose, at least one researcher (Lynne Parker at MIT) has had difficulty adapting it to work in a different domain (multiple communicating mobile robots in an indoor office environment). This may teach us something about what is needed for a simulator to be truly reusable and general.
Lynn Stein has done some work on a program, MetaToto [10], which models a very rough approximation of the environment of the robot Toto built by Maja Mataric [8]. In her paper, Stein maintains that when coupled with a working robot system, rough simulation can help MetaToto to build an approximate landmark map which can be useful while the robot learns about a new physical topology. The simulation in MetaToto is not intended to accurately model the physical domain of Toto, or to be used in the initial design and testing of a mobile robot architecture. It is interesting, however, in that it tests the hypothesis that simulators can be designed well enough that the same robot control program runs on both the simulator and the physical robot being simulated.

In my own Master's research, I plan to extend the work of Maja Mataric [7; 8] on Toto to handle higher-level linguistically described cognitive features. Toward this end, I have begun to build a simulator for the sonar-based robot Toto in its indoor office environment. I have succeeded in making the simulator correct enough that the same robot control programs will drive both the simulator and the robot. The problem is made easier by the fact that the drive system, a holonomic Real World Interfaces B-12 base, is well designed. Sonar modelling appears to be the hardest aspect of this simulator. I am starting to experiment with probabilistic models based on data taken from literature on sonar experiments. Soon, I plan to take actual sonar data from the robot to use in the simulation, and to model such characteristics as specular reflection and different reflective properties of surfaces.

Idealized Simulators

Some simulators are not meant to simulate a real physical world at all, but merely test the kinds of decision making and problem solving that robots might need to go through. In some cases, these simulators abstract parts of the navigation problem by assuming a cellular space.

Rich Sutton's work on the DYNA learning architecture [11; 12] uses a simulation which models some aspects of a dynamically changing environment with spatial locality. As a simulator, this program suffers from the criticism that it may allow a false decomposition of the problem of intelligent action by allowing Sutton to focus on a reasoning problem in isolation. We argue that the jury is still out on the question of how this problem of intelligent action may be decomposed. For this reason, it is still worthwhile to pursue strategies for isolated subproblems such as reasoning and perception. In this particular case, Sutton is very clear about the limits of his work. Mark Drummond and Martha Pollack have a simulator called Tileworld which, though different in some details, shares the dynamic and unpredictable characteristics of Sutton's simulator.

The video game domains Pengo and Amazon used by Phil Agre and David Chapman in their work on Pengi [1] and Sonja [2] are another example of a less than realistic simulator. Agre argues that the domain shares many of the problems of situated action found in the physical world, while finessing certain hard problems such as vision. The perception research this assumes is not as farfetched as a "magic recognition box," but it is well beyond the capabilities of current artificial vision systems.

MICE is an experimental testbed offered by Durfee and Montgomery for distributed artificial intelligence research and it contains many features which make it attractive for experiments with multiple communicating robots with heterogeneous skills. It uses multiple asynchronous processes to simulate each robot, so the simulation of multiple robots may potentially start many processes. Its cellular representation of space may make it unsuitable for detailed physical simulation. Its separation of the simulation from the implementation of the agent control programs is a positive contribution.

These idealized simulators are less interesting from the perspective of mobile robot research, as they make no attempt to accurately model the problems of navigation physical robots encounter. Still, it is important to evaluate them with respect to their author's claims about the realistic nature of the problems they model.

So far no simulator, realistic or otherwise, is a competitive substitute for experiments on physical robots. What would it take to build such a simulator?

Principles for Design

We conclude by presenting a set of principles which may be followed in the design and use of mobile robot simulators. The powerful simulator we call for here has its origins in the discussion of better simulators as a future direction Erann Gat's dissertation [6]. Our hope is that by making some features of this proposed simulator concrete, we may encourage others to join our efforts to develop such a simulator.

First and foremost, a simulator must come with a clear sense of the assumptions made about the domain it models, the capabilities and limits of the simulation. Claims about the generality, scalability and usefulness of results demonstrated on a simulator must always be made with respect to the relationship of the simulator to the real world.

Another important characteristic of an effective simulator is modularity and extensibility. As Gat describes, an ideal simulator will "allow the user to construct customized robots by mixing and matching sensors and actuators which are actually software objects." Models of error used in these devices should also be customizable by the user. These components should be described in an easy to understand language, either declarative, procedural, or some combination, so that they can be shared between different implementations of the simulator running on different platforms.

The simulator should provide models of the types of sensors commonly used on mobile robots, including sonars, pyroelectric and infrared sensors, microphones and photocells, all preprocessed in various ways. For effectors, holonomic bases such as the RWI B-12 base and non-holonomic bases such as four and six-wheeled cars, treads vehicles such as bulldozers and tanks, and legged robots should all be offered. In addition to "canned" sensors and actuators, submodules should also be provided at a variety of levels to facilitate implementation of new sensors and actuators. It should be possible to configure simulated robots in any reasonable way, including the mounting of sensors on articulated parts such as arms or actuated pivots.

The user should have as many choices as possible, both
at design time—when constructing the robot and the environment—and at run time, in the characteristics and performance of the simulation. The simulator should provide both precise algorithms and faster approximation algorithms, and allow the user to choose between them to trade off precision against speed in the simulation. The user should be able to control the granularity of the simulation in both space and time. Continuous motion simulation should be used wherever possible, as unintended effects from discrete simulation have been documented [4].

Pseudorandom sequences used to generate data for simulated sensors should be repeatable exactly. If the simulator has control over the robot control architecture, and if that architecture is deterministic, this will allow a repeatability not possible in the physical world that can reveal elusive bugs in the control programs or in the simulator itself.

Surfaces and obstacles in the simulated environment should be able to have different response characteristics with respect to each sensor, to model things like different specular reflective response to sound or light. Different surfaces should be able to have different responses to the same sensor, to model the differences between walls of different colors, textured objects, and so on.

An interactive graphical interface is desirable for ease of use, development, and debugging user robot control programs, new robot models, and other new simulator modules. The simulator should provide for multiple possibly heterogeneous robots, including “drones” or moving obstacles, controlled by the user at run time or by simple programs. Different robots should be able to be driven by different robot control architectures, potentially even implemented in different programming languages. A client-server model or other distributed systems approach may be helpful here. This can foster separation of the simulation from the robot control programs, as well as potentially distributing the computational load over several computers.

Finally, in using the simulator the scientific community should work toward generally accepted “toolkit parts,” including error models, environments, sensors and effectors, parts of robots and whole robots which the community agrees are reasonable models of actual physical robots in particular domains.

Conclusion

Simulators can be a powerful tool for the development and testing of robot control architectures. We hope they will continue grow in number, power, flexibility and acceptance as the field working on autonomous mobile robots comes to understand their strengths and to use them wisely.

Acknowledgements

My thanks to Erann Gat and to an open letter on the Usenet group comp.robotics from a past head of NSF grant awards for starting my thinking about what it takes to make a good simulator, and when they can be useful. Thanks also to my adviser Lynn A. Stein, and to Nils Nilsson, Rod Brooks, Lynne Parker and Barney Pell for fruitful discussions.

References