The Role of Learning in Autonomous Robots

Rodney A. Brooks
MIT Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139
brooks@ai.mit.edu

Abstract

Applications of learning to autonomous agents (simulated or real) have often been restricted to learning a mapping from perceived state of the world to the next action to take. Often this is couched in terms of learning from no previous knowledge. This general case for real autonomous robots is very difficult. In any case, when building a real robot there is usually a lot of a priori knowledge (e.g., from the engineering that went into its design) which doesn't need to be learned. We describe the behavior-based approach to autonomous robots, and then examine four classes of learning problems associated with such robots.

1 INTRODUCTION

The purpose of this paper is to suggest some problem formulations for machine learning that would have relevance to physically instantiated autonomous robots. The key relevant aspects of such robots are:

- They may have many hundreds of input bits.
- There is no simple mapping between input bits and the state of the robot and the actual physical world.
- There are many possible actions (taking some hundreds of bits to specify) which can be taken at any given time step.
- Current (and perhaps all) learning algorithms have very little chance of coping under conditions of a single global reinforcement signal.
- Certain domain knowledge is readily available, and can help the learning process.
- As with animals, internal reinforcement signals are necessary for some types of learning.
- From a methodological point of view it is better to think about the robot as a system that interacts with the world already, and learning as a method of improving its performance. This contrasts with viewing the robot as a learning machine.

This paper does not pretend to be a comprehensive survey of the use of machine learning with autonomous robots. Instead, it is very deliberately limited to the robots developed within the mobile robot group at the MIT Artificial Intelligence Lab.

2 AUTONOMOUS ROBOTS

Over recent years a new approach has developed in Artificial Intelligence for controlling autonomous mobile robots. See [**Brooks 91b**] for more details but in essence the following tells the story.

2.1 INTELLIGENCE

Around 1984, a number of people started to worry about the problem of organizing intelligence. There was a requirement that intelligence be reactive to dynamic aspects of the environment, that a mobile robot operate on time scales similar to those of animals and humans, and that intelligence be able to generate robust behavior in the face of uncertain sensors, an unpredicted environment, and a changing world. None of the existing approaches at that time seemed to be able to satisfy these conditions.

Some of the key realizations about the organization of intelligence were as follows:

- Most of what people do in their day to day lives is not problem-solving or planning, but rather it is routine activity in a relatively benign, but certainly dynamic, world. Furthermore the representations an agent uses of objects in the world need not rely on a semantic correspondence with symbols that the agent possesses, but rather can be defined through interactions of the agent with the world. Agents based on these ideas have achieved interesting performance levels and were built from combinatorial circuits plus a little timing circuitry ([**Agre and Chapman 87**], [**Agre and Chapman 90**]).

- An observer can legitimately talk about an agent's beliefs and goals, even though the agent need not manipulate symbolic data structures at run time. A formal grounding in semantics used for the agent's design can be compiled away. Agents based on these ideas have achieved interesting performance levels and were built from combinatorial circuits plus a little timing circuitry ([**Rosenschein and Kaelbling 86**], [**Kaelbling and Rosenschein 90**]).
- In order to really test ideas of intelligence it is important to build complete agents which operate in dynamic environments using real sensors. Internal world models which are complete representations of the external environment, besides being impossible to obtain, are not at all necessary for agents to act in a competent manner. Many of the actions of an agent are quite separable—coherent intelligence can emerge from subcomponents interacting in the world. Agents based on these ideas have achieved interesting performance levels and were built from combinatorial circuits plus a little timing circuitry ([**Brooks 86**], [**Brooks 90b**], [**Brooks 91a**]).

A large number of others have also contributed to the approach. [**Maes 90**] is the most representative collection.

2.1.1 The Subsumption Architecture

In the mobile robot lab at the MIT Artificial Intelligence Laboratory we have been using a specific architecture for programming robots, known as the *subsumption architecture*. It incorporates many of the ideas presented above—a number of architectures similar in spirit but with varying details are used by other researchers. The subsumption architecture was arrived at by continuously refining attempts to program a robot to reactively avoid collisions in a people-populated environment, [**Brooks 86**]. It has been generalized so that its purpose is to program intelligent, situated, embodied agents.

Our principles of computation are:

- Computation is organized as an asynchronous network of active computational elements (they are *augmented finite state machines*—see [**Brooks 89**] for details[1]), with a fixed topology network of uni-directional connections.
- Messages sent over connections have no implicit semantics—they are small numbers (typically 8 or 16 bits, but on some robots just 1 bit) and their meanings are dependent on the dynamics designed into both the sender and receiver.
- Sensors and actuators are connected to this network, usually through asynchronous two-sided buffers.

These principles lead to certain consequences. In particular:

- The system can certainly have state—it is not at all constrained to be purely reactive.
- Pointers and manipulable data structures are very hard to implement (since the model is Turing equivalent it is of course possible, but hardly within the spirit).
- Any search space must be quite bounded in size, as search nodes cannot be dynamically created and destroyed during the search process.
- There is no implicit separation of data and computation, they are both distributed over the same network of elements.

In all the robots built in the mobile robot lab, the following principles of organization of intelligence have been observed:

- There is no central model maintained of the world. All data is distributed over many computational elements.
- There is no central locus of control.
- There is no separation into perceptual system, central system, and actuation system. Pieces of the network may perform more than one of these functions. More importantly, there is intimate intertwining of aspects of all three of them.
- The behavioral competence of the system is improved by adding more behavior-specific network to the existing network. We call this process *layering*. This is a simplistic and crude analogy to evolutionary development. As with evolution, at every stage of the development the systems are tested—unlike evolution there is a gentle debugging process available. Each of the layers is a behavior-producing piece of network in its own right, although it may implicitly rely on presence of earlier pieces of network.
- There is no hierarchical arrangement—i.e., there is no notion of one process calling on another as a subroutine. Rather the networks are designed so that needed computations will simply be available on the appropriate input line when needed. There is no explicit synchronization between a producer and a consumer of messages. Message reception buffers can be overwritten by new messages before the consumer has looked at the old one. It is not atypical for a message producer to send 10

[1] For programming convenience we use a higher level abstraction known as the *Behavior Language*, documented in [**Brooks 90c**]. It compiles down to a network of machines as described above.

messages for every one that is examined by the receiver.

- The layers, or behaviors, all run in parallel. There may need to be a conflict resolution mechanism when different behaviors try to give different actuator commands.
- The world is often a good communication medium for processes, or behaviors, within a single robot.

2.2 ROBOTS

We have built over ten robots that are programmed with the subsumption architecture. See [**Brooks 90b**] for a recent overview.

Some of these robots have had learning capabilities. But they have not followed the reinforcement learning techniques like Q learning of [**Watkins 89**], as has the work, with physical robots, of [**Kaelbling 90**] and [**Mahadevan and Connell 91**].

2.3 Sensors and Action

Real sensors are very noisy. They do not give the same sort of simple mapping from actual world state to a clean input vector that we are used to expecting after using simulated robots.

Simple sensors have characteristics which distort the mapping from world state to input vector in very discontinous and non-intuitive ways.

For instance, infrared proximity sensors are very sensitive to the size of the nearby object and its surface albedo—thus the paint color will greatly effect the distance from a wall at which the sensor will fire. But worse than that, it is quite common for two identically built sensors to have a sensitivity that varies by as much as a factor of two.

Simple sonars, used on many robots, respond to obstacles within a thirty degree cone. But there is a fairly large range of angles over which the sonar beam will bounce off rather than return to the emitter. Thus many ordinary looking situations appear as a maze of mirrors to sonar. Small changes in angle can give very discontinous range measurements. But worse, different materials respond differently to sonar (even in versions with multiple frequencies). In one instance we found that all the doors on one side of a corridor completely absorbed the sonar signal so that the doors always appeared open, but of course the robot was unable to go through them.

Vision is even harder to use on a real robot. Given the variations in lighting we can not reliably recognize objects or get good three dimensional descriptions of the world. These things can be done under very restricted experimental set ups, but not in the context of a moving autonomous robot. Even simple things like recognizing the color of the wall in front of a robot are not doable. Color discrimination, while seemingly trivial for humans, is actually a complex problem and an open research question. The naive idea that we perceive colors based directly on the frequency of reflected light is simply false.

2.3.1 Example of Complexity

As an example of the raw complexity presented by these robots consider the case of Attila, [**Angle and Brooks 90**], a six legged walking robot.

A central processor sends commands and receives inputs at 10 Hz to and from a number of peripheral processors which package and unpackage these communications. The number of input and output bits from the point of view of the central processor (which simulates a distributed set of computations) is detailed below.

Each of Attila's six legs has three motor actuated degrees of freedom. There is also a body pitch motor, a pan and tilt motor for the head, and two motors to control an active antenna. Thus there are 23 controllable motors. Each motor can be give new 20 bit commands (8 position, 4 velocity, 4 force, 2 to select a gain table, and 2 to cause calibration actions to be taken) at 10 Hz. Another 14 bits can be sent to various sensors to switch them on and off, giving a total of 474 action bits.

The robot has 150 sensors. Not including the vision sensor these produce roughly 800 bits of sensing information at 10 Hz. These sensors include strain guages on the legs, 1 bit proximity senors, touch sensors, joint position sensors, antenna strains, range sensors, surface color sensors, and battery voltage monitors.

The vision sensor, a 165×192 pixel CCD camera, produces 253440 bits at 5 Hz for the vision processor. It summarizes this into about 128 bits which are delivered to the main processor at 10 Hz.

Using a standard state space model this robot has a state space of size 2^{928} with 2^{474} possible actions. Note however, that its perceived state, does not map simply into the world state, so there is added complexity.

3 LEARNING

The previous problems lead me to be pessimistic about the possibility of building learning systems which can operate on realistic robots and make use of a simple reward system—i.e, the thrust of much recent learning research seems not too useful.

There are two observations which can lead us to the sorts of things a robot might be able to learn, however.

First, as a result of engineering and building a robot we often know quite a lot *a priori* about how the robot can interact with the world. It makes sense to use this knowledge when it is easy to do so, rather than force the robot to learn it all afresh.

Second, a robot should have a number of internal reward functions built in, rather than relying on a single external reward function. Animals for instance rely on satiating hunger to learn certain tasks rather than using the single bit of reward from death, perhaps due to starvation but perhaps due to some other cause, as a learning reward. Likewise it makes sense to have a robot worry about its battery voltage and learn to keep that high, besides getting rewards for covering terrain, finding interesting rocks, or whatever its primary mission is designed to be.

With these points in mind we suggest the following as reasonable challenges for things that autonomous robots in the real world should learn:

1. representations of the world that help in some task
2. aspects of instances of sensors and actuators (this is sometimes called calibration)
3. the ways in which individual behaviors should interact
4. new behavioral modules

4 EXAMPLES

We now discuss some early exmaples of autonomous robots learning these classes of things. We restrict our attention to systems that have been implemented on real robots within the MIT Mobile Robot lab.

4.1 REPRESENTATIONS

In order to make sense of the vast number of input bits, an autonomous robot needs to develop internal representations.

[**Mataric 90, 91**] introduced *active-constructive representations* to subsumption in a sonar-based robot, *Toto*, which wandered around office environments building a map based on landmarks, and then used that map to get from one location to another. Her representations were totally decentralized and non-manipulable, and there is certainly no central control which build, maintains, or uses the maps. Rather, the map itself is an active structure which does the computations necessary for any path planning the robot needs to do.

Primitive layers of control let Toto wander around following boundaries (such as walls and furniture clutter) in an indoor environment. A layer which detects landmarks, such as flat clear walls, corridors, etc., runs in parallel. It informs the map layer as its detection certainty exceeds a fixed threshold. The map is represented as a graph internally. The nodes of the graph are computational elements (they are identical little subnetworks of distinct augmented finite state machines). Free nodes arbitrate and allocate themselves, in a purely local fashion, to represent a new landmark, and set up topological links to physically neighboring nodes (using a limited capacity switching network to keep the total virtual 'wire length' between finite state machines to be linear in the map capacity). These nodes keep track of where the robot is physically, by observing changes in the output of the landmark detector, and comparing that to predictions they have made by local message passing, and by referring to other more primitive (magnetic compass based) coarse position estimation schemes.

When a higher layer wants the robot to go to some known landmark, it merely 'excites', in some particular way the particular place in the map that it wants to go. The excitation (this is an abstraction programmed into the particular finite state machines used here—it is not a primitive—as such there could be many different types of excitation co-existing in the map, if other types of planning are required) is spread through the map following topological links, estimating total path link, and arriving at the *landmark-that-I'm-at-now* node (a deictic representation) with a recommendation of the direction to travel right now to follow the shortest path. As the robot moves so to does its representation of where it is, and at that new node the arriving excitation tells it where to go next. The learned map thus bears a similarity to the *internalized plans* of [**Payton 90**], but it represented by the same computational elements that use it—there is no distinction between data and process. Furthermore Mataric's scheme can have multiple simultaneously active goals—the robot will simply head towards the nearest one.

The next step in this work will be to generalize active-constructive representations to more classes of use.

4.2 CALIBRATION

[**Viola 90**] demonstrated calibration of a complex head-eye system modeling the primate vestibulo-ocular system. The problem is to keep a camera platform steady even as the robot moves about. This lets the visual processing routines have a better chance of working.

In this system there is one fast channel between a gyroscope and a high performance pan-tilt head holding the camera, and a slower channel using vision which produces correction signals for the gyroscope channel. The task is to learn an open loop transfer function between the gyroscopes and the motors. The vision channel provides a feedback signal that is too slow to servo the camera but can be used to update the estimate of the transfer function. It monitors the images from the camera and returns independent signals for horizontal and vertical image slip. In the first case the possible outputs are *none*, *left*, or *right*, and in the second case they are *none*, *up*, or *down*.

The same system was used to learn how to accurately saccade to moving stimuli, where in this case the transfer function was from a desired target for the center of the fovea to a set of motor commands to ballistically move the fovea to that location. The error signal in this case was the relative position of the moving stimulus after the camera movement, again as three possible outcomes for each of vertical and horizontal.

In both cases the functions to be learned are not shown to the system, but instead only crude overshoot and undershoot correction signals are provided. THe functions learned, however, are real valued functions on one and two dimensional domains.

4.3 INTERACTION

Lastly, [**Maes and Brooks 90**] programmed an early six legged robot to learn to walk using the subsumption architecture along with the behavior activation schemes of [**Maes 89**]. Independent behaviors on each leg monitored the activity of other behaviors and correlated that, their own activity state, and the results from a belly switch which provided negative feedback, as input to a local learning rule which learned under which conditions it was to operate the behavior. After about 20 trials per leg, spread over a total of a minute or two, the robot reliably learns the alternating tripod gait—it slowly seems to emerge out of initially chaotic flailing of the legs.

This example is the closest to the machine learning ideas in reinforcement learning. It was also a very simple case, where we carefully provided just a few bits of state information (six). It remains to be seen how such an approach might scale up.

4.4 NEW BEHAVIORS

This is a difficult area, and we have not seen any non-trivial new behaviors learned by any existing approaches. In all cases they have been constrained so much by the experimenters that there is very little to learn—the experimenters have solved all the difficult interaction and coupling problems already.

5 CONCLUSION

There is a rich domain for theoretical learning research in autonomous mobile robots. At the moment the two fields are far apart. It will take patience on both sides to come to a middle ground where the autonomous robot community is able to articulate the problems they find interesting, and the theoretical learning community is able to provide algorithms robust enough to operate in the complex real world environments provided by the robots.

Acknowledgements

Support for this research was provided in part by the University Research Initiative under Office of Naval Research contract N00014–86–K–0685, in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014–85–K–0124, in part by the Hughes Artificial Intelligence Center, in part by Siemens Corporation, and in part by Mazda Corporation.

References

[**Agre and Chapman 87**] "Pengi: An Implementation of a Theory of Activity", Philip E. Agre and David Chapman, *AAAI-87*, Seattle, WA, 1987, 268–272.

[**Agre and Chapman 90**] "What Are Plans for?", Philip E. Agre and David Chapman, *in* [**Maes 90a**], 1990, 17–34.

[**Angle and Brooks 90**] "Small Planetary Rovers", Colin M. Angle and Rodney A. Brooks, *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Ikabara, Japan, 1990, 383–388.

[**Brooks 86**] "A Robust Layered Control System for a Mobile Robot", Rodney A. Brooks, *IEEE Journal of Robotics and Automation, RA-2*, April, 1986, 14–23.

[**Brooks 89**] "A Robot that Walks: Emergent Behavior from a Carefully Evolved Network", Rodney A. Brooks, *Neural Computation 1:2*, 1989, 253–262.

[**Brooks 90a**] "Challenges for Complete Creature Architectures", Rodney A. Brooks, *Proc. First Int. Conf. on Simulation of Adaptive Behavior, MIT Press*, Cambridge, MA, 1990, 434–443.

[**Brooks 90b**] "Elephants Don't Play Chess", Rodney A. Brooks, *in* [**Maes 90**], 1990, 3–15.

[**Brooks 90c**] "The Behavior Language; User's Guide", Rodney A. Brooks, *MIT A.I. Lab Memo* 1227, 1990.

[**Brooks 91a**] "Intelligence Without Representation", Rodney A. Brooks, *Artificial Intelligence, 47*, 1991, 139–160.

[**Brooks 91b**] "Intelligence Without Reason", Rodney A. Brooks, *IJCAI-91*, Sydney, August, 1991.

[**Kaelbling 90**] "Learning in Embedded Systems", Leslie P. Kaelbling, *PhD thesis, Stanford University*, 1990.

[**Kaelbling and Rosenschein 90**] "Action and Planning in Embedded Agents", Leslie Pack Kaelbling and Stanley J. Rosenschein, *in* [**Maes 90**], 1990, 35–48.

[**Maes 89**] "The Dynamics of Action Selection", Pattie Maes, *IJCAI-89*, Detroit, MI, 1989, 991–997.

[**Maes 90**] "Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back", Pattie Maes *(ed)*, *MIT Press*, Cambridge, MA, 1990.

[**Maes and Brooks 90**] "Learning to Coordinate Behaviors", Pattie Maes and Rodney A. Brooks, *AAAI-90*, Boston, MA, 1990, 796–802.

[**Mahadevan and Connell 91**] "Automatic Programming of Behavior-based Robots using Reinforcement Leraning", Sridhar Mahadevan and Jonathan Connell, *Proceedings, AAAI-91*, July 1991.

[**Mataric 90**] "Navigation with a Rat Brain: A Neurobiologically-Inspired Model for Robot Spatial Representation", Maja J Mataric, *Proc. First Int. Conf. on Simulation of Adaptive Behavior, MIT Press*, Cambridge, MA, 1990, 169–175.

[**Mataric 91**] "Behavioral Synergy Without Explicit Integration", Maja J Mataric, *special issue of SIGART on Integrated Intelligent Systems*, July, 1991.

[**Payton 90**] "Internalized Plans: A Representation for Action Resources", David W. Payton, *in* [**Maes 90a**], 1990, 89–103.

[**Rosenschein and Kaelbling 86**] "The Synthesis of Machines with Provable Epistemic Properties", Stanley J. Rosenschein and Leslie Pack Kaelbling, *Proc. Conf. on Theoretical Aspects of Reasoning about Knowledge, Joseph Halpern* (ed)*, Morgan Kaufmann*, Los Altos, CA, 1986, 83–98.

[**Watkins 89**] "Learning from Delayed Rewards", Christopher Watkins, *Ph.D. Thesis*, King's College, Cambridge, 1989.