From: AAAI-98 Proceedings. Copyright © 1998, AAAI (www.aaai.org). All rights reserved.

A Framework for Reinforcement Learning on Real Robots

William D. Smart and **Leslie Pack Kaelbling**
Computer Science Department
Brown University
Providence, RI 02912
(401) 863-7667
{wds, lpk}@cs.brown.edu

Robots are now being used in complex, unstructured environments, performing ever more sophisticated tasks. As the task and environmental complexity increases, the need for effective learning on such systems is becoming more and more apparent. Robot programmers often find it difficult to articulate their knowledge of how to perform a given task in a form suitable for robots to use. Even when they can, the limitations of robot sensors and actuators might render their intuitions less effective. Also, it is often not possible to anticipate (and code for) all environments in which the robot might find itself having to perform a certain task. Therefore, it seems useful to have the robot be able to learn to act, in the hope of overcoming these two difficulties.

One promising approach to learning on real robots that is attracting considerable interest at the moment is *reinforcement learning*. The programmer must supply a *reward function* which maps states of the world onto a scalar reward, essentially saying how good or bad it is to be in a given state. Once given this function, the robot can, in theory, learn an appropriate action policy to maximize some measure of reward over time. Designing such a function seems more intuitive than writing code to perform the task, since we can do so in more general terms.

However, there are some significant problems when we attempt to use this strategy on a real, physical robot, including (but not limited to) the following.

Low data rate Since sensor data (and hence reward data) are being generated by physical processes, the rate at which we can gather such information is typically low, with rates of 10Hz or lower being common. Combining this with the often extreme size of the state space means that we have a very sparse coverage of this space, making learning much more difficult.

Exploration is dangerous To gather information about the utility of taking actions from a given state, a reinforcement learning system will typically take actions about which it knows little or nothing. In many applications, this is allowable, but on a physical system can be extremely dangerous. For example, the only way for the robot to find out the utility of falling down the stairs is to actually fall down them, something that we would hope to avoid. This is especially relevant in the early stages of learning, when little is known and most actions are exploratory.

Copyright ©1998, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

In order to overcome these problems, we propose a new framework for reinforcement learning on real robots. The framework has three main components; a provided control policy and reward function, a policy learning process and a value learning process. Initially, the robot is under the control of the supplied policy. As it performs its task, the policy learner and value learner are both passively observing the sensor readings and effector commands issued by the supplied policy. The policy learner attempts to learn the mapping embodied in the supplied control policy, while the value learner attempts to learn a predictive version of the reward function.

Once the learned policy is performing as well as the supplied one, control passes over to it. From this point onwards, a reinforcement learning algorithm uses the learned value function in an attempt to improve the learned control policy. Eventually, the learned control policy will converge on one which is optimal with respect to the supplied reward function.

By supplying an initial control policy, we can bootstrap the reinforcement learning process and give is a useful bias. In the initial phase, we can gather information on the value function without performing exploratory actions and can also focus attention on the part of the state space that we will actually be using for the task at hand. Limiting ourselves to a part of the possible state space makes the learning tasks easier since, although we have a sparse data distribution over the whole space, it tends to be dense in the areas in which we are interested.

There are several questions that we hope to address in this work. What are the special requirements on the learning algorithms? How good does the initial policy have to be, and how much can we improve on it? How do we combine information from more than one initial policy? What sort of speed and performance benefits can be realized with this approach?