Computación y Sistemas
ISSN: 1405-5546
computacion-y-sistemas@cic.ipn.mx
Instituto Politécnico Nacional
México

Fuentes, Olac; Rao, Rajesh P.N.; Van, Michael
Hierarchical Learning of Reactive Behaviors in an Autonomous Mobile Robot
Computación y Sistemas, vol. 1, núm. 2, octubre-diciembre, 1997, pp. 71-75
Instituto Politécnico Nacional
Distrito Federal, México

HIERARCHICAL LEARNING OF REACTIVE BEHAVIORS IN AN AUTONOMOUS MOBILE ROBOT

Olac Fuentes
Laboratorio de Inteligencia Artificial
Centro de Investigación en Computación
Instituto Politécnico Nacional
México D.F., México
e-mail: fuentes@jsbach.cic.ipn.mx

Rajesh P. N. Rao and Michael Van Wie
Computer Science Department
University of Rochester
Rochester, New York
U.S.A.
e-mail: {rao,vanwie}@cs.rochester.edu

ABSTRACT

We describe an autonomous mobile robot that employs a simple sensorimotor learning algorithm at three different behavioral levels to achieve coherent goal-directed behavior. The robot autonomously navigates to a goal destination within an obstacle-ridden environment by using the learned behaviors of obstacle detection, obstacle avoidance, and beacon following. These reactive behaviors are learned in a hierarchical manner by using a simple hillclimbing routine that attempts to find the optimal transfer function from perceptions to actions for each behavior. We present experimental results which show that each behavior was successfully learned by the robot within a reasonably short period of time. We conclude by discussing salient features of our approach and possible directions for future research.

Keywords: Robot learning, behavior-based robotics, robot navigation.

1 INTRODUCTION

Traditionally, the task of developing a sensorimotor control architecture for a situated autonomous robot was left to the human programmer of the robot. Prewiring robot behaviors by hand however becomes increasingly complex for robots with large number of sensors and effectors, especially when

achieved by employing a *hierarchical behavior-b* *position* of the control architecture as originally s Brooks [1]. In addition, it is desirable in many cas the robot with the ability to adapt its constituent b line in response to environmental stimuli by allow *tonomously learn* the transfer function mapping s into motor commands.

For even moderately complex tasks and/or rob dimensionality of the sensorimotor space makes l ficult. One commonly used approach to make ro feasible despite the high dimensionality of the sen to run the learning algorithm on a simulated envi example, [6]). However, in many situations, it i impossible to gather enough knowledge about t its environment to build an accurate simulation some physical events, such as collisions, are ext cult to simulate even when there is complete kno these reasons, we believe that in order for the lea be applicable by the physical robot in its environ learning and experimentation has to be carried ou bodied physical robot itself. However, using a r some drawbacks: given the slowness of real worl tation and the limited computing power typically autonomous mobile robots, for the learning algo successfully applied, it is crucial that they conv reasonable number of trials and that they don't

havior. In particular, the robot solves the task of navigating to a goal destination (indicated by an infrared beacon) within an obstacle-ridden environment by using a set of learned behaviors for obstacle detection, obstacle avoidance, and beacon following. The behaviors themselves are learned individually by using a simple heuristic hillclimbing technique.

2 TASK DESCRIPTION

The task to be learned by the robot (figure 1) is one of navigation and obstacle-avoidance. Specifically, we expect the robot to learn appropriate sensorimotor strategies for navigating between two points in an obstacle-ridden environment.

Three classes of sensory input are available to the robot:

- **Bump Sensors**: Realized using digital microswitches, these sensors indicate whether the robot is physically touching an obstacle. Five of these sensors, placed at different locations around the robot, are used for learning the *obstacle-detection* behavior. In particular, the robot is expected to learn to back up when its front bump sensors are active, to turn left when the right bump sensor is active, and so on.

- **Photosensors**: Three shielded photoresistors placed in a tripodal configuration are used to give advance warning of an approaching obstacle, taking advantage of the fact that the obstacles have a darker color than the floor. The inputs from these sensors are used for learning the *obstacle-avoidance* behavior; they are expected to allow the robot to steer clear of obstacles detected in its path.

- **Infrared detectors**: These sensors, when used in conjunction with infrared detection software, indicate the strength of the modulated infrared light in a small spread along their lines of sight. Four of these sensors are used to learn the high-level behavior of navigating toward the goal position, which is a source of infrared transmission.

The above sensory repertoire is supplemented by two ef-

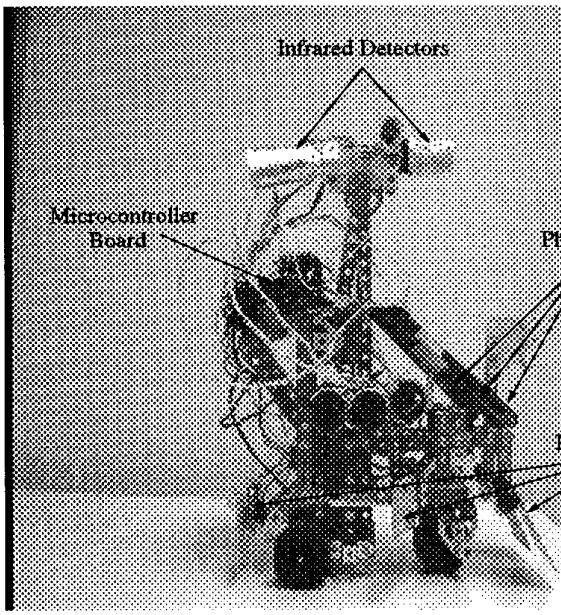


Figure 1: The robot used for the exper

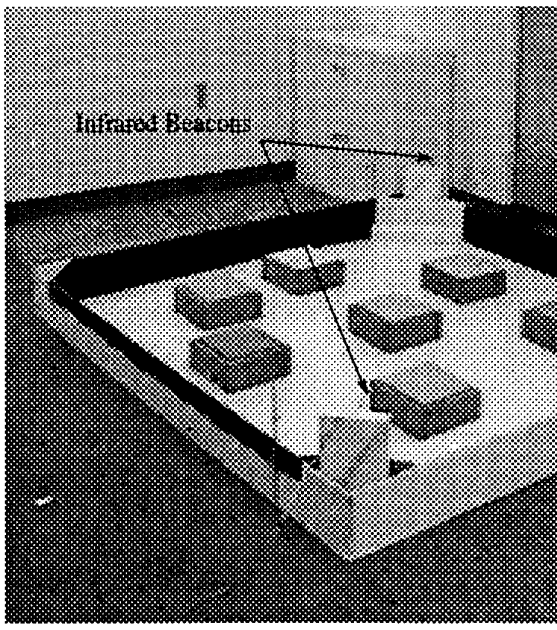
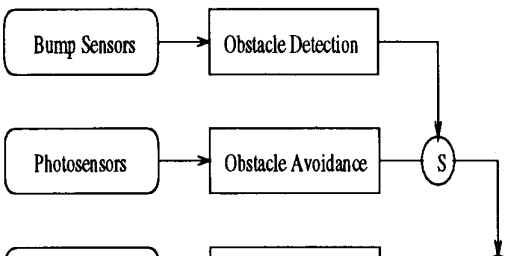


Figure 2: The robot arena



3 BEHAVIOR-BASED TASK DECOMPOSITION

Since the robot is equipped with twelve different sensors and two effectors, the learning task consists of finding a mapping from the 12-dimensional sensory space to the 2-dimensional motor space that optimizes the robot's performance of the task. The number of different perceptions the robot may encounter grows exponentially with the number of sensors it possesses, thereby making the task of hardwiring behaviors extremely cumbersome and error-prone.

One way of circumventing this "curse of dimensionality" is to divide the given task into several layers of control such that the first layer consists of an elementary level of performance (say avoid continuous contact with obstacles), with each subsequent layer improving upon the performance obtained by the previous ones. If we chose this partition carefully, we can also arrange things in such a way that the first layer uses only a subset of the sensors available and each subsequent layer uses a superset of the sensors used by the lower layers. This is reminiscent of Brooks' subsumption architecture [1]. Figure 3 illustrates the simple three-level hierarchical architecture used in our robot.

Hierarchical partitioning of the sensory space allows the robot to learn the sensorimotor mapping corresponding to each layer independent of the other layers. This greatly reduces the search space and allows for an implementation where all the learning can be done by physically experimenting with the world, instead of relying on simulation. By not relying on simulation we avoid the danger of learning a policy that works well only in simulation and can not be transfered to the real world.

4 LEARNING REACTIVE BEHAVIORS USING PERCEPTUAL GOALS

To learn each constituent behavior, we use a relatively simple hillclimbing technique. Since we only keep in memory a policy that encodes a series of statements of the form *perception* $\rightarrow$ *action*, the method can be implemented using very little memory. This is in contrast to some machine learning techniques recently applied to mobile robotics (for example, genetic programming [4], reinforcement learning [2], and neural

defines the action $m(p) \in A$ to be taken when co
the sensory stimulus $p \in P$.

To every perception p we also assign a nu
v_p measuring the desirability or "goodness" of t
were that perception normally occurs. For exam
tual input indicating one or more pressed bump
low v, since its occurs in the undesirable situat
robot crashes into an obstacle, while having no b
pressed will have a high v, since it indicates the

The task of the learning mechanism is to le
m that will take the robot from "bad" to "good"
and maintain it in good perceptions when they a
achieve this by computing a heuristic metric $h($
sures how often, on average, the action taken i
has resulted in perceptions that are more desirabl
every *perception-action* pair in the current polic
heuristic value h and replace those entires in th
are judged to be inadequate (*i.e.* for which h falls
specified threshold.)

The heuristic hillclimbing learning algorithm
level can be defined as follows:

1. Randomly initialize m
2. Initialize heuristic value and occurrence co
   $(\forall p \in P)h(p) = 0, n(p) = 0$
3. Repeat until convergence
   (a) Get perceptual input p from sensors
   (b) Perform action $m(p)$
   (c) Get resulting perceptual input r from
   (d) Adjust heuristic value
   $h(p) = \frac{n(p)}{n(p)+1}h(p) + \frac{1}{n(p)+1}(\alpha(v_r -$
   (e) Update occurrence counter
   $n(p) = n(p) + 1$
   (f) if $h(p) < threshold$ replace m_p by a ra
   sen action $q \in A$ and reinitialize $h(p)$

At the obstacle-detection and obstacle avoi
$v(p) = 1$ if p represents a perception where th
crashing into an obstacle (*i.e.* none of the bu

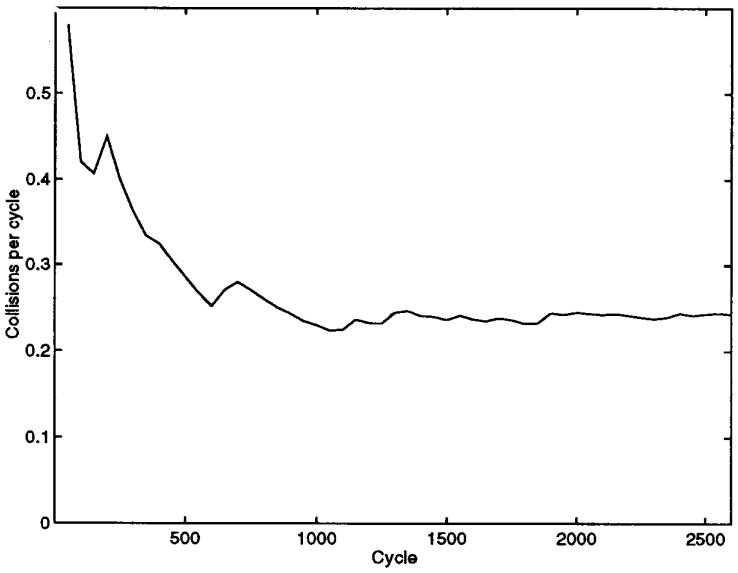


Figure 4: **Obstacle Detection.** The plot shows the average collisions per cycle as a function of the number of cycles.

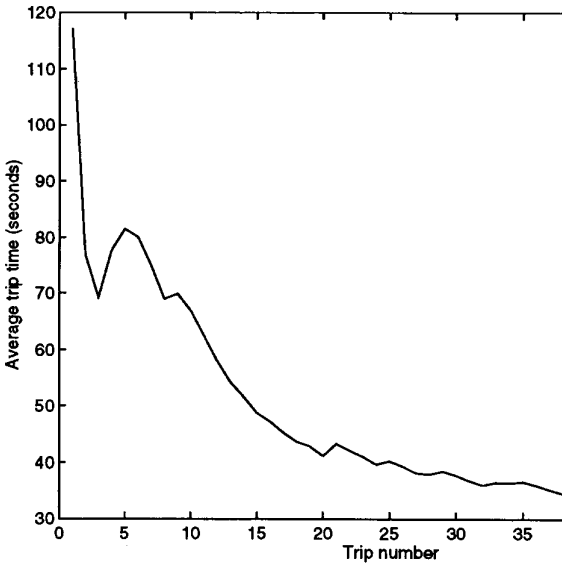


Figure 6: **Beacon Following.** The plot shows th per trip from one beacon to the other as a func number.

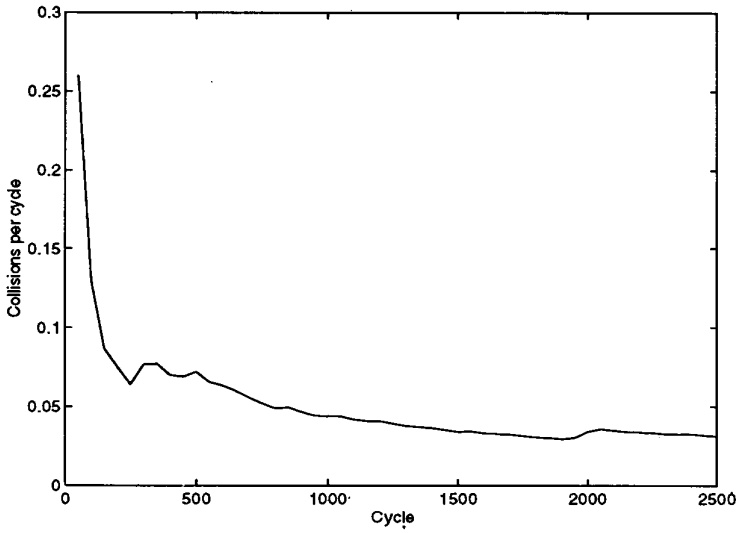


Figure 5: **Obstacle Avoidance.** The average collisions per cycle plotted as a function of the number of cycles.

5 Experimental results

In our experiments, the robot runs through the three levels of behavior, first learning to detect collisions with obstacles, then learning to avoid such collisions, and, finally, learning to navigate from goal to goal. Once the algorithm obtains adequate performance as specified by pre-set criteria, it switches behaviors and begins learning at the next level. For example,

ure 4 plots collisions per robot control cycle ver lowest level behavior, where the robot can feel stacles; figure 5 plots collisions-per-cycle vers second level behavoir, where the robot can bo obstacles; and figure 6 plots average time-per-t number during the beacon-following behavior.

Figure 4 shows the performance of the obstacle-detection behavior. Collisions-p sharply until it reaches a stable value of appr beyond which point the (blind) robot cannot ter a few hundred cycles, the robot has learnec ate actions to take when it crashes into an o the robot has achieved a good level of perfo obstacle-detection behavior, it switches to the havior, obstacle-avoidance. The results for thi shown in figure 5. Collisions-per-cycle again starting this time with the final value from the and eventually reaching a new minimum of ab the previous case, after a few hundred cycles cessfully learns a policy that results in significa lisions. It should be noted that given the finite of the robot and the cluttered environment, co be completely eliminated.

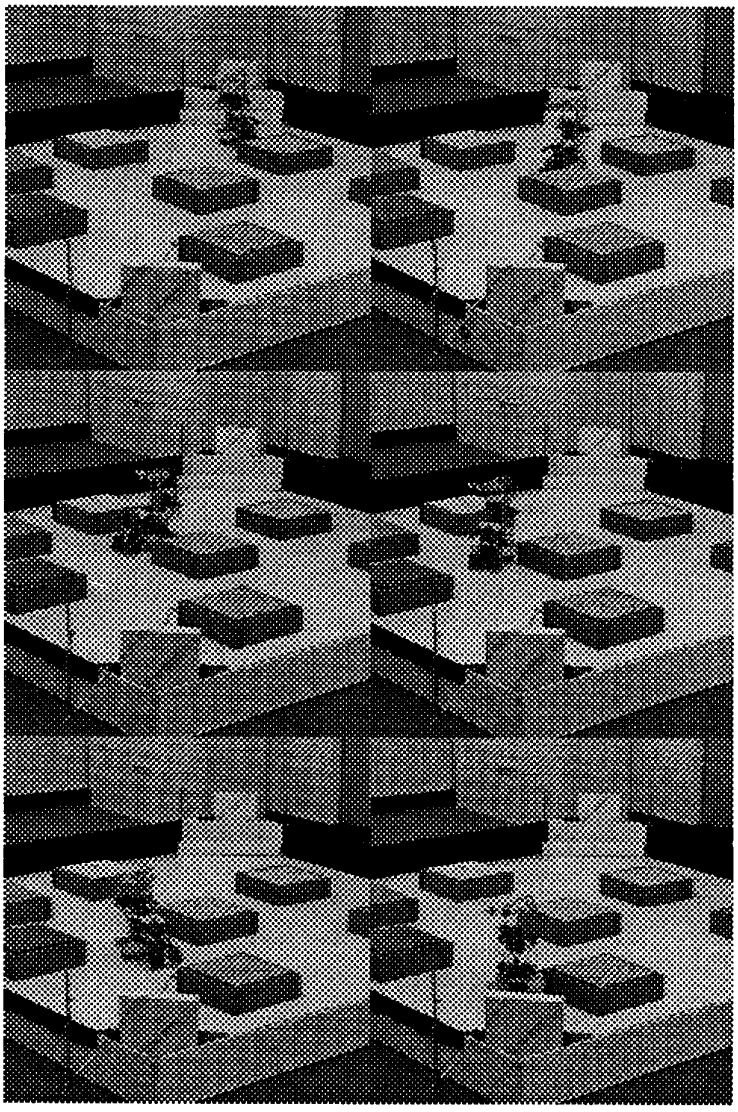


Figure 7: The Obstacle Avoidance/Beacon following behavior of the robot after hierarchical learning.

task of homing to the location of the goal beacon).

Figure 7 depicts the behavior toward the end of the learning period, when all three behaviors are active.

6 Conclusions

We have shown that a simple heuristic hillclimbing strategy can be effectively used for learning useful reactive behaviors in an autonomous mobile robot. Our method results in considerable savings of memory space over other learning methods such as genetic programming, reinforcement learning and neural networks since we require the storage of only a small history of perceptions for determining credit assignment fol-

tional behaviors, and possible autonomous learni
dination between behaviors (cf. [5]).

References

[1] Rodney A. Brooks. A robust layered control s
mobile robot. *IEEE Journal of Robotics and A*
2(1):14–22, April 1986.

[2] D. Gachet, M.A. Salchis, L. Moreno, and
mentel. Learning emergent tasks for an auton
bile robot. In *IEEE/RSJ International Confer*
telligent Robots an Systems, pages 290–297, 1

[3] Ben J. A. Krose and Marc Eecen. A self-organi
sentation of sensor space for mobile robot nav
IEEE/RSJ International Conference on Intelli
an Systems, pages 9–14, 1994.

[4] M. A. Lewis, A. H. Fagg, and A. Solidum. G
gramming approach to the construction of a
work for control of a walking robot. In *Pro*
the 1992 IEEE International Conference on R
Automation, Nice, France, 1992.

[5] Pattie Maes and Rodney A. Brooks. Learning
nate behaviors. In *Proceedings of the 1990 AA*
ence, 1990.

[6] David Pierce and Benjamin Kuipers. Lea
climbing functions as a strategy for generating
in a mobile robot. In *From Animals to Animats*
ings of the First International Conference or
lation of Adaptive Behavior, pages 327–336. C
MA: MIT Press, 1991.

Olac Fuentes. *Recived his Ph. D. in Computer Science fr*
of Rochester in May of 1997. His research interests inc
learning, robotics and computer vision. Dr. Fuentes is a prof
de Investigación en Computación, of the Instituto Politécr
in Mexico City, Mexico.