# Achieving Rapid Adaptations in Robots by Means of External Tuition*

**Ulrich Nehmzow*** **and Brendan McGonigle**

Laboratory for Cognitive Neuroscience at the Department of Psychology
University of Edinburgh
GB - Edinburgh EH8 9LE
e-mail:B.O.McGonigle@ed.ac.uk

*Department of Computer Science
University of Manchester
GB - Manchester M13 9PL
e-mail:u.nehmzow@cs.man.ac.uk

## Abstract

Experiments with the *Edinburgh R2* mobile robot are presented that show how robots can be *taught* to accomplish various different tasks, without the need for re-programming the controller, and without using self-tuition.

In an externally supervised teaching process — not unlike the process of "shaping" known in animal learning — the robot acquires competences such as obstacle avoidance, contour following, box pushing, phototaxis or route learning (mazes). The learning is fast.

## 1 Introduction

It has been shown that competences fundamental to robot control, such as obstacle avoidance or contour following can, in behaviour-based robotics, be achieved through the interaction of independent, so-called behavioural modules ([Malcolm *et al.* 89, Brooks 85]). The question of how to re-combine and orchestrate several of those fundamental competences in order to synthesize new, more complex global behaviours is as yet an unresolved problem. The experiments presented here show one way to approach this: the *Edinburgh R2* mobile robot acquires fundamental motor competences, *as well as* combinations of them, through an externally supervised learning process.

This method of providing external feedback to accelerate a learning process is known as "shaping" in the field of animal learning, but is as yet relatively rarely used in robotics (see, for example, [Colombetti & Dorigo 93, Shepanski & Macy 87]). Rather, the predominant methods used for controller design have rested either on complete knowledge installation (an approach we consider unsuitable for all but the most basic behaviours), or self-tuition. The latter involves *autonomous* acquisition of task-achieving competences, and has successfully been used to accomplish tasks such as obstacle avoidance and contour following ([Nehmzow *et al.* 89, Nehmzow 91]), box pushing ([Mahadevan & Connell 91]), leg coordination ([Maes & Brooks 90]) or phototaxis ([Kaelbling 90, Colombetti & Dorigo 93]).

The advantage of providing external feedback (shaping) is, firstly, the considerable acceleration of the learning process. The fundamental robot behaviours described in this paper, for example, are all acquired in a couple of learning steps, taking less than a minute in real time; more complex behaviours such as maze learning are still obtained in tens of learning steps. Secondly, one and the same robot can be used for a number of different tasks, without any alteration to its control program or control parameters, merely using the training process to define the robot's behaviour.

## 2 Experimental Setup

The experiments described in this paper were conducted using an R2 mobile robot, the *Edinburgh R2*, shown in figure 1.

*Published in D. Cliff, P. Husbands, J.A. Meyer and S. Wilson (eds), From Animals to Animats 3, MIT Press 1994. 

Figure 1: The *Edinburgh R2*

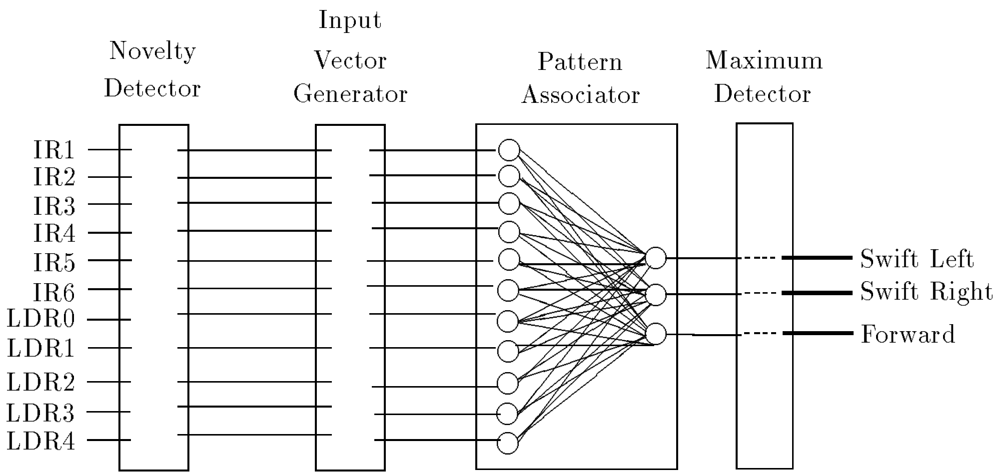


Figure 2: Controller structure

The robot has two differentially driven wheels, which allow the robot to move in any direction and perform turns on the spot. The robot's maximum speed is $40cm/s$, however the speed travelled at in the experiments described here does not exceed $8cm/s$. Its sensors comprise eight infrared sensors (IRs), which are used for obstacle detection and approximate range sensing (three range bins), six light-sensitive sensors based on passive light dependent resistors (LDRs), which are usually used for orientation and navigation, and five bumper sensors. The arrangement of the infrared and light sensors is shown diagrammatically in figure 3. The robot also has a gripper, which was not used for the experiments reported here.

## 2.1 Controller Architecture

The controller used for the experiments described here is shown in figure 2.

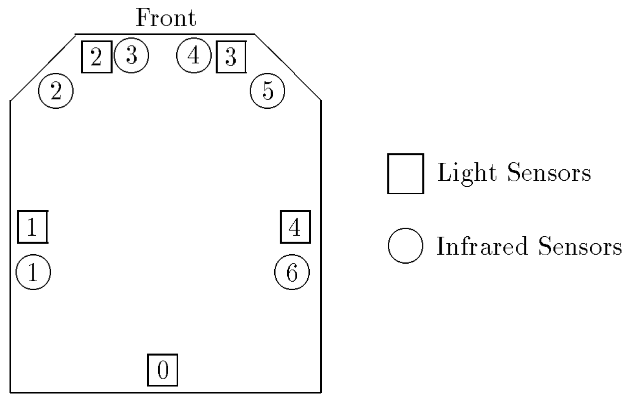


Figure 3: Sensor arrangement in the *Edinburgh R2*

Motor actions of the robot are associated with sensor signals by means of an artificial neural network (described in more detail later in this paper). Sensor signals, coming from both the infrared and the light sensors, are used as input to the network. The responses of the robot to each particular sensory stimulus — i.e. the motor actions performed by the robot — are stored as associations in the network: the three output nodes of the network stand for three distinct motor actions of the robot (swift left turn, swift right turn, and forward motion).

## 2.2 Representation of Network Input Space

A fresh input vector is generated whenever the novelty detector of the controller detects that either the general heading of the robot has changed[1] or the sum of all current IR readings differs by more than the numerical value of 2 from the previous one. If neither of these conditions apply, no input vector is generated and the robot continues performing the action it is currently performing.

Once an input vector is generated, it is fed into the artificial neural network serving as associative memory. The input vector contains information from both the obstacle detecting and range sensing infrared sensors, as well as from the orientation sensing light sensors. It is shown in figure 4.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| IR1 | IR2 | IR3 | IR4 | IR5 | IR6 | LDR0 | LDR1 | LDR2 | LDR3 | LDR4 |
| Range signal from IRs | | | | | | Signals from light sensors | | | | |

Figure 4: Input vector used

Each of the infrared sensors of the Edinburgh R2 returns a numerical value between '0' ('nothing detected') and '3' ('object seen at close range' — about 15 cm in the case of a white wall). Light sensors return numerical values between '0' (total darkness) and '255' (very bright). The sensor are adjusted such that '128' is returned for average lighting conditions in our laboratory. The six values of all front-facing and side-facing infrared sensors form the first part of the input vector, the five values coming from light sensors the second part.

As these sensors respond to different environmental stimuli, and furthermore return numerical values of different orders of magnitude, both components of the input vector have to be balanced to avoid a bias towards one or the other component (i.e. *sensor fusion* has to be performed). Here, this is accomplished by first normalising the six-element long vector containing infrared sensor information, then normalising the five-element long vector containing information from the light sensors, and then normalising the eleven-element long overall input vector.

## 2.3 Representation of Network Output Space

The artificial neural network used as an associative memory is a *Pattern Associator* ([Rumelhart & McClelland 86, Kohonen 88]). It has eleven input units and three output units. A unique compound motor action is assigned to each of the three output units: swift left turn (i.e. left motor turning backwards, right motor turning forwards), swift right turn and forward motion. The action associated with the most excited output cell is the action performed by the robot at any one time. As input vectors change, due to changing stimuli from the environment, different excitation patterns will develop in the output layer of the

[1] This can be done measuring the robot's orientation with respect to a distant light source, using the light sensors.

associative memory, resulting in different motor actions performed by the robot.

## 2.4 Teaching Procedure

### 2.4.1 The Associative Memory

The central element of the controller shown in figure 2, the Pattern Associator (see figure 5), is a two-layer artificial neural network that is taught by *supervised* learning.

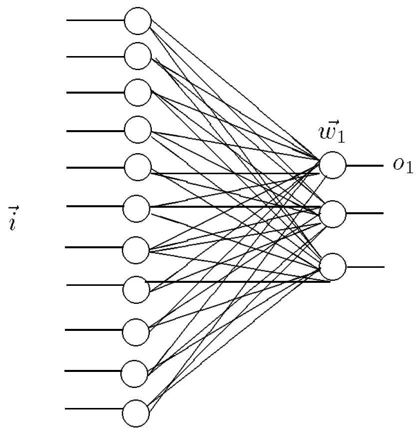


Figure 5: Schematic diagram of Pattern Associator

Compared to other network paradigms, for example multi-layer Perceptrons, this type of network has the advantage to learn extremely fast (one-shot learning in most cases) — an advantage that is vital to applications in robotics as certain fundamental functions (such as obstacle avoidance) need to be acquired rapidly, because the robot's safe operation depends on them. However, Pattern Associators can only learn linearly separable functions, i.e. functions like "exclusive OR" cannot be learned perfectly. For those functions the network settles in the minimum energy state, resulting in the robot always making *one* mistake first. So far, however, we have not encountered real-world applications that require the acquisition of input-output mappings that are not linearly separable, so that this shortcoming has not been a problem in our particular case.

The output $o_j$ of each output unit $j$ of the network is computed as the dot product of input vector $\vec{i}$ and weight vector $\vec{w_j}$:

$$o_j(t) = \vec{i}(t) \cdot \vec{w_j}(t) \qquad (1)$$

### 2.4.2 Teaching the Network

Training a Pattern Associator is done in a supervised way, i.e. a teaching signal, signifying the desired output vector $\vec{\tau}$, given a particular input vector $\vec{i}$ is provided from outside. In work reported elsewhere ([Nehmzow 91, Nehmzow *et al.* 93]) such teaching signals were generated internally by the controller, using so-called *instinct rules*; in the experiments described here, however, they are provided externally by the experimenter: by covering or uncovering the upward facing light sensor of the robot positive or negative feedback respectively can be provided. As long as the upward facing light sensor of the robot is covered, the robot's actions are assumed to be correct, as soon as negative reinforcement is given by uncovering the upward facing light sensor, the robot cycles through its repertoire of three motor actions until positive reinforcement is given again, signifying that the correct motor action for a particular sensory stimulus is performed.

Equation 2 shows the update rule used in the Pattern Associator, indicating how weight vector $\vec{w_j}$ of output unit $j$ is changed in each learning step, given the desired output $\tau_j$ of unit $j$ and the actual output $o_j$ of unit $j$. $\eta$ is the learning rate, a parameter that influences the speed of learning. Its value is typically chosen between 0.2 and 0.7.

$$\vec{w_j}(t+1) = \vec{w_j}(t) + \eta \cdot (\tau_j(t) - o_j(t)) \cdot \vec{i}(t) \qquad (2)$$

# 3 Experiments

The robot uses both infrared (i.e. obstacle detecting) sensors and light (i.e. direction detecting) sensors, and has the choice of three distinct motor actions (swift left turn, swift right turn and forward movement). Virtually all combinations of associations between sensor input and motor output can be taught.[2] To demonstrate this, we have taught the robot a number of competences, emphasizing those competences that are typically required for robot operation in office-like environments. These comprise fundamental behaviours such as avoiding obstacles (to ensure safe operation), following contours, complex routes, and orientation (i.e. competences related to navigational tasks), or moving objects (material handling). These experiments are described below. For all experiments the very same controller was used, no programming, nor adjustment of parameters, nor any change other than erasing the artificial neural network before teaching a new task were performed.

## 3.1 Obstacle Avoidance

It is obvious that no higher level tasks can be accomplished by the robot if safe operation is not guaranteed. Obstacle avoidance is fundamental to this. We therefore trained the robot to avoid obstacles first.

When facing open space (i.e. neither of the IR sensors detects anything) the robot was given positive reinforcement, likewise for turning left whilst facing an object on its right hand side, and for turning right when facing an object on the left hand side. After three learning steps, one for each of the aforementioned constellations, the Edinburgh R2 had acquired an obstacle avoidance behaviour. The teaching process takes under one minute.

[2]With the restriction mentioned in section 2.4.1, that the learned function has to be linearly separable.

## 3.2 Box Pushing (Object Following)

Object following displays the robot's ability to react to sensor stimuli in a dynamically changing environment; we trained the robot to push a cardboard box, as an example of an object following competence (the function of following an object moved by the robot is equivalent to that of following an object moved by an external power source). Figure 6 shows this experiment: whenever the box moves out of the center of the robot, the robot adjusts it course and steers back towards the box.

Object following behaviour is achieved in four learning steps. Positive reinforcement needs to be given twice for moving forward when facing an object straight ahead, and once each for the appropriate turns when facing objects on the left hand side or right hand side, respectively.

## 3.3 Wall Following

Route following is the most reliable method of navigation, used widely in the animate world (e.g. rodents). Following contours, such as walls, is one instance of route following, and relevant for navigational tasks of a robot.

The robot can be taught to stay within a given range of distances to objects on its side (either left or right), and to move ahead at the same time if the space in front is obstacle-free by giving positive reinforcement whenever the robot is in the 'correct' distance to the wall (approx. 20 cm in this case), and by positively reinforcing moves towards the wall or away from the wall whenever the robot is too far away or too close respectively.

Four teaching steps were necessary to acquire the wall following competence, two for the 'move forward' situation and one each for the 'turn away' and 'move closer' situation. Again, this teaching process requires less than one minute in real time.


Figure 6: Object following, here: box pushing.

Figure 7: The Edinburgh R2 in the Route Following Experiment.

### 3.4 Phototaxis

Like contour following, orientation is indispensible for successful navigation. Insects, for example bees and ants, use the position of the sun for orientation, birds are able to sense the earth's magnetic field ([Waterman 89]).

Like insects, the robot uses external light sources for orientation. It can be taught to move towards a light source (phototaxis), or away from it (negative phototaxis). This is achieved in four learning steps, taking under one minute in real time: twice giving positive reinforcement for a forward movement whilst facing the light source, once giving positive reinforcement for a right turn whilst seeing the light on the right hand side, and once similarly for the light source on the left hand side. To achieve negative phototaxis, four teaching steps are required as well, obviously positive reinforcement is then given for movement *away* from the light source.

## 4 Re-combination of Basic Behavioural Components

Using the same teaching process, the *Edinburgh R2* can be taught to acquire more than one of the above mentioned competences at the same time. This leads to more complex behaviours.

### 4.1 Phototaxis and Obstacle Avoidance

After the robot has acquired the phototaxis competence (section 3.4), it can be taught to avoid obstacles (section 3.1) as well. This requires a further two learning steps, one to associate objects on the left hand side with a right turn, and one to associate objects on the right hand side with a left turn. The ability to move towards a light source is not lost by this.

If no further teaching is used, the robot will, however, repeatedly turn towards and away from an obstacle that is placed exactly in the path towards the light source, until small fluctuations make the robot pass the object. This problem can be alleviated by further teaching (see section 4.2).

### 4.2 Obstacle Avoidance, Phototaxis and Wall Following

Using the fundamental competences of obstacle avoidance (section 3.1), phototaxis (section 3.4) and wall following (section 3.3), the robot can also be taught to learn routes through simple mazes, such as the one shown in figure 8. This is an example of an acquired navigational competence.

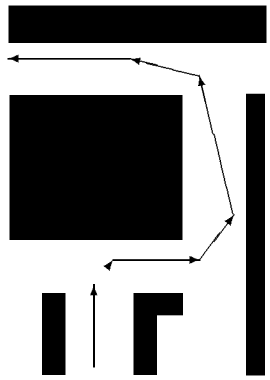

Figure 8: A three-junction maze

This maze contains three T-junctions, which means the robot has to associate the sensory stimuli perceived at each of the three junctions with either a left or a right turn, as well as those stimuli perceived in corridors (of whatever heading) with a forward movement. Whilst moving through the maze, walls must be avoided at all times, so that maze learning involves both obstacle avoidance and direction-oriented movement. Figure 7 shows the robot negotiating the maze.

To achieve route learning in mazes, more learning steps than in the examples previously given are necessary. Firstly, this has to do with the fact that the robot has to acquire both the obstacle avoidance and wall following competences, *and* has to learn the correct turns at junctions. Secondly, *sensor fusion* becomes important: directional information (coming from the light sensors) has to be combined with object information (coming from the infrared sensors), in order to negotiate the maze successfully. In our implementation, sensor fusion occurs in the construction of the input vector (see section 2.2), and in the artificial neural network itself.

In order to learn the path shown in figure 8, the robot needs about five presentations of the whole route through the maze. This means that about thirty to fifty learning steps are required, in real time this takes about ten minutes.

The complexity of mazes learnable by the robot depends, on the one hand, on the robot's sensors and sensory stimuli perceived at different junctions. If distinct junctions give rise to identical sensory stimuli, only one and the same action at these distinct junctions can be learned. Secondly, the capacity of the network (see [Hertz *et al.* 91] for more details) influences the maximum complexity of a learnable maze. More research is necessary to address the latter point conclusively.

## 5 Summary and Conclusion

In experiments with the *Edinburgh R2* mobile robot we have shown that it is possible to teach robots fundamental competences such as obstacle avoidance, contour following, phototaxis or object following, as well as more complex combinations of these (such as phototaxis *and* obstacle avoidance, or route learning in mazes).

The teaching process used is similar to a method known as "shaping" in the field of animal learning: through external feedback, given by the experimenter, the robot acquires the effective associations between perceived sensor signals and desired motor responses.

Learning these task-achieving competences is very fast, in the experiments presented here the robot acquires fundamental competences such as obstacle avoidance or contour following in a couple of learning steps, taking under one minute of real time; more complex capabilities such as maze learning are acquired in tens of learning steps.

## Acknowledgements

The research reported here was conducted in Edinburgh and supported by the MRC, grant number G9110835 ("Cognitive Modelling and the Design of Artifically Intelligent Systems", principal investigators Dr. Brendan McGonigle and Professor Mark Lee, Computer Science, National University of Wales at Aberystwyth).

We thank Claudia Nehmzow for helpful comments on earlier versions of this paper.

## References

[Brooks 85] Rodney Brooks, *A Robust Layered Control System for a Mobile Robot*, MIT AI memo 864, 1985.

[Colombetti & Dorigo 93] M. Colombetti & M. Dorigo, *Robot Shaping: Developing Situated Agents through Learning*, International Computer Science Institute, Berkely, Technical Report no. 40, April 1993.

[Hertz *et al.* 91] John Hertz, Anders Krogh and Richard Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, 1991.

[Kaelbling 90] Leslie Kaelbling, *Learning in Embedded Systems*, Stanford Technical Report TR-90-04, June 1990.

[Kohonen 88] Teuvo Kohonen, *Self-Organization and Associative Memory*, 2nd edition, Springer Verlag 1988.

[Maes & Brooks 90] Pattie Maes and Rodney Brooks, *Learning to Coordinate Behaviours*, Proc. AAAI 1990.

[Mahadevan & Connell 91] Sridhar Mahadevan and Jonathan Connell, *Automatic Programming of Behaviour-Based Robots using Reinforcement Learning*, Proc. AAAI 1991.

[McGonigle 91] Brendan McGonigle, *Incrementing Intelligent Systems by Design*, in J.A. Meyer & S. Wilson (eds.), *From Animals to Animats*, MIT Press 1991.

[Malcolm *et al.* 89] Chris Malcolm, Tim Smithers and John Hallam, *An Emerging Paradigm in Robot Architecture*, in T. Kanade, F.C.A. Groen and L.O. Hertzberger (eds.), *Intelligent Autonomous Systems 2*, Amsterdam 1989; ISBN 90-800410-1-7.

[Nehmzow *et al.* 89] Ulrich Nehmzow, John Hallam and Tim Smithers, *Really Useful Robots*, in T. Kanade, F.C.A. Groen and L.O. Hertzberger (eds.), *Intelligent Autonomous Systems 2*, Amsterdam 1989; ISBN 90-800410-1-7.

[Nehmzow 91] Ulrich Nehmzow, *Experiments in Competence Acquisition for Autonomous Mobile Robots*, PhD thesis, Edinburgh University, 1991.

[Nehmzow *et al.* 93] Ulrich Nehmzow, Tim Smithers and Brendan McGonigle, *Increasing Behavioural Repertoire in a Mobile Robot*, in J.A. Meyer, H. Roitblat and S. Wilson (eds.), *From Animals to Animats 2*, MIT Press 1993.

[Waterman 89] Talbot H. Waterman, *Animal Navigation*, Scientific American Library, W.H. Freeman, 1989.

[Rumelhart & McClelland 86] David Rumelhart, James McClelland and the PDP Research Group, *Parallel Distributed Processing*, Vol. 1, p.61, MIT Press 1986.

[Shepanski & Macy 87] J.F. Shepanski & S.A. Macy, *Manual Training Techniques of Autonomous Systems Based on Artificial Neural Networks*, Proc. IEEE First Intern. Conf. on Neural Networks, San Diego 1987.