Accepted for publication in H. Cruse, H. Ritter and J. Dean, Proc. Workshop "Prerational Intelligence in Robotics", Bielefeld 16.-20.5.1994

Acquisition of Smooth, Continuous Obstacle Avoidance in Mobile Robots

Ulrich Nehmzow

Department of Computer Science
University of Manchester
GB - Manchester M13 9PL
Tel: +4461 - 275 6169
Fax: +4461 - 275 6236
u.nehmzow@cs.man.ac.uk

Abstract

To enable robots to cope with unforeseen situations, as well as the noise and variation inherent in the real world, we believe it is essential to incorporate plasticity into robot controllers, and use as little predefined knowledge as possible.

We present first results of experiments in autonomous competence acquisition with "FortyTwo", a mobile robot equipped with tactile, sonar, infrared and visual sensors, used at the Department of Computer Science at Manchester. In these experiments, associations between incoming sensor signals and corresponding motor responses are stored in an artificial neural network, the learning process is independent from human interaction and, due to the type of network used, fast.

Contrary to our earlier work, in which active output nodes of the network triggered *compound* motor actions, FortyTwo uses just two output nodes which function similar to motor neurons in animate systems: each node directly drives one of two drive motors. This means that a continous spectrum of motor actions can be acquired, resulting in a smooth and continous obstacle-avoiding motion of the robot.

1 Introduction

This paper presents experiments on autonomous competence acquisition, conducted with the Manchester mobile robot "FortyTwo", a fully autonomous robot, equipped with on-board controllers, actuators, and sonar, infrared, tactile and visual sensors.

The questions it addresses are: How can flexibility in situations unforeseen by the designer be achieved? How can robots autonomously adapt to changing circumstances in world, task, or their own morphology? How can they improve their performance, based on past experiences?

The experiments described in this paper use an adaptable mapping between incoming sensor signals and outgoing actuator responses to approach these questions: an artificial neural network is used to store sensor-motor couplings and is trained in real time, using the information that is obtained through the robot's interaction with the environment. The learning is rapid, tasks such as obstacle avoidance or contour following are acquired in a couple of learning steps, within seconds.

In order to achieve as flexible a control as possible, we believe it is essential to reduce the amount of predefined knowledge used by the robot. In this paper we present an extension to earlier work on autonomous competence acquisition ([Nehmzow *et al.* 89, Nehmzow 92]) that reduces the amount of necessary predefined knowledge even further by using artificial motor neurons that drive motors directly.

1.1 Background: Reinforcement Learning

Reinforcement Learning — the learning mechanism used here — covers techniques of "learning by trial and error through performance feedback, i.e. from feedback that evaluates the behaviour, ... but does not indicate correct behaviour" ([Sutton 91]). As it uses a measure of overall performance only, it is relatively easy to implement and has been used in robot control to achieve phototaxis ([Kaelbling 91, Colombetti & Dorigo 93]), boxpushing ([Mahadevan & Connell 91]), obstacle avoidance ([Nehmzow *et al.* 89]) or walking ([Maes & Brooks 90]). One problem in reinforcement learning is that if the search space (e.g. the space of possible actions) is large, learning may take a long time to "take off", simply because very little positive reinforcement is received in the early stages of the learning process. Kaelbling describes this vividly by pointing out that her light-seeking robot needed to be pushed towards the light manually in the early stages of learning, in order to subsequently acquire the competence of phototaxis ([Kaelbling 90]). Similarly, Mahadevan and Connell report that their best reinforcement learning algorithm used for box-finding (Q-learning with statistical clustering) received an average reward of about 0.16 after 2000 learning steps, whilst a hand-coded box-finding agent was able to achieve an average award of about 0.25.

It should be obvious that the benefits of reinforcement learning in robots — the ability to adapt to changing environments, to cope with changes in the agent, such as sensor failure, or to acquire new competences, etc. — can be lost if too long a learning phase is required, particularly for tasks fundamental to the robot's operation, such as obstacle avoidance. Careful design of input and output space (for example by using quantised input data —[Mahadevan & Connell 91] — or a restricted set of motor actions — [Kaelbling 91, Maes & Brooks 90]) is a way to alleviate this problem. In addition to these two methods we have chosen a "Pattern Associator" as core component of the controller — an artificial neural network that can learn only a restricted set of functions, but these extremely fast (one or two orders of magnitude faster than other learning schemes, such as for instance the backpropagation network). This enables our robots to acquire fundamental sensor-motor couplings rapidly, requiring a couple of learning steps only.

It is our experience that, in order to increase the robot's flexibility in unforeseen situations, it is beneficial to reduce the amount of predefined knowledge used in the controller as much possible. This paper describes extensions to work presented earlier ([Nehmzow 92]) that do exactly this: by introducing the concept of an artificial motor neuron the output space of the artificial neural network is reduced, while the repertoire of possible robot actions is increased, now covering the complete range of physically possible actions.

2 The Manchester Mobile Robot

The Manchester mobile robot (shown in figure 1), named FortyTwo (after its serial number) is a hexagonal robot, 50 cm in diameter and 80 cm high (weight 59 kg). It is equipped with sixteen sonar range finding sensors (range 15 cm to 10.60 m), sixteen infrared proximity sensors (active range, depending on surface colour approximately 60 cm), twenty tactile bumper sensors and a monochrome camera (490 x 512 pixels) and frame grabber.

The robot is controlled by a distributed controller, the master being a 486 PC (66 MHz, 20 MIPS). Sensor modules are controlled by a 68HC11F1 microprocessor.

Turret and base rotation in FortyTwo are independent, the maximum speed of the robot is 0.5 ms^{-1}.

3 The Robot Controller

3.1 General Principle

The general design of the controller used in FortyTwo is shown in figure 2: associations between incoming sensor signals and corresponding motor actions are stored in an associative memory, implemented in this case as an artificial neural network (a Pattern Associator. See [Rumelhart & McClelland 86] for more information). Such a Pattern Associator is shown in figure 3.

The output o_j of each output unit j of the network is computed as the dot

Figure 1: The Manchester Mobile Robot, “FortyTwo”

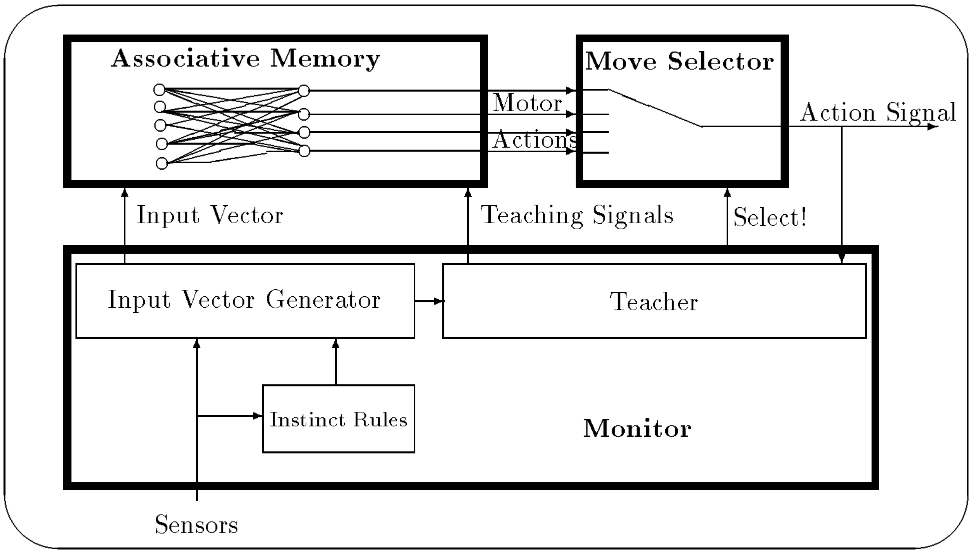

Figure 2: Architecture of Learning Controller — General Layout

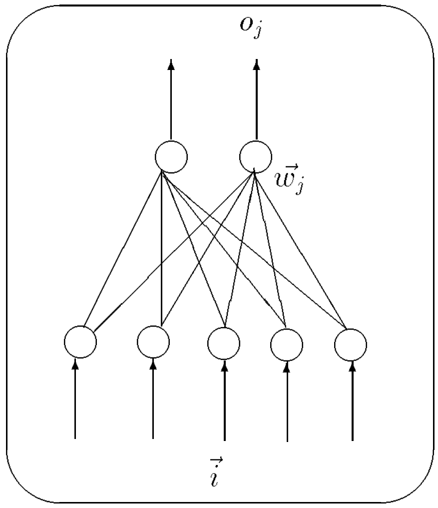

Figure 3: Pattern Associator

product of input vector $\vec{i}$ and weight vector $\vec{w_j}$:

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

Each output node of the network stands for one action pattern of the robot, for example "move left motor forward, whilst moving the right motor backward" (which results in a swift left turn).

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 τ_j of unit j and the *actual* output o_j of unit j.

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

η is the learning rate, a parameter that influences the speed of learning. Its value is typically chosen between 0.2 and 0.7.

As training the Pattern Associator is supervised learning, a means of assessing the robot's performance and providing feedback for the learning process has to sought; this is accomplished using so-called "instinct-rules", defined sensor states that have to be maintained at all time (for example "tactile sensors must not be *on*"). Training takes place only when an instinct-rule is violated. When this happens, an input vector to the associative memory is generated, and the corresponding motor action performed for a certain time. If the violated instinct-rule becomes satisfied by this action, the association between the particular input vector at the time and the output unit of the successful action is strengthened, otherwise the next strongest output of the network is performed, for a slightly longer period. The robot continues to cycle through its action repertoire in this manner until a successful action is found.

3.2 Improved Controller of FortyTwo

The output space of the controller described in section 3.1 is discrete: each output node of the network stands for a particular, "compound" motor action (e.g. "Turn left motor forwards whilst turning right motor backwards"). The repertoire of possible actions is therefore limited to the number of output nodes of the network.

The major problem with this approach is that the designer has to determine the repertoire of actions that *might* be needed for the robot's operation at the design stage; this can be difficult, particularly as one purpose of using a learning controller is to achieve flexibility in *unforeseen* circumstances.

For the experiments reported here we have therefore altered the controller shown in figure 2: the network now uses two output nodes only, each output node acting like a "motor neuron" and driving one motor directly (see figure 4)[1]. This means that the robot's repertoire of possible actions is no longer limited to the number of output units of the network, but is continuous and contains translational as well as rotational moves, all at a continous range of speeds.

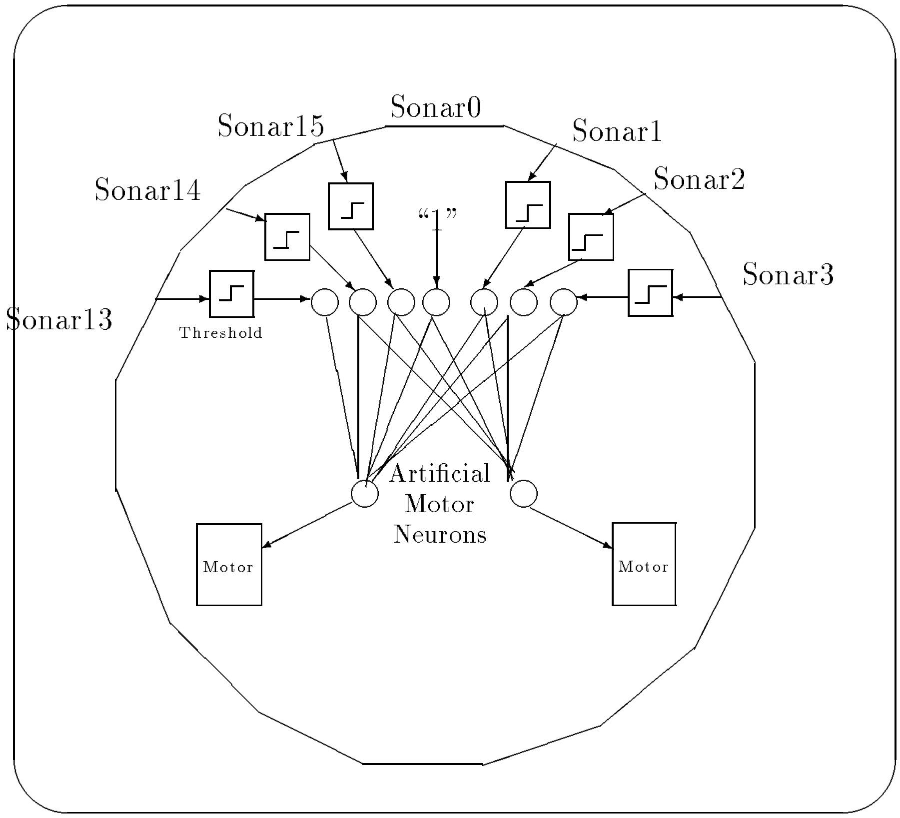

Figure 4: Neural Network Control on "FortyTwo"

[1]FortyTwo really has only one driving motor, but driving and steering motors are accessed by the controller as if the robot had two differential drive motors.

4 Experiments

4.1 Instinct-Rules and Input Vector

The instinct rules used for the experiment described here are shown in figure 5 (The numbering of the sensors is shown in figure 4).

1. Sonars 0,1,2,14 and 15 must always return range values greater than 50 cm.

2. The velocity sensor must always return values greater than 4 cms^{-1}.

Figure 5: Instinct-Rules

The network has seven input units: a constant “1” input to ensure non-zero inputs at all times, as well as the inputs from six sonar sensors located at the front of the robot (sensors 1,2,3,13, 14 and 15). The range values from these sensors are thresholded, input lines being switched to “1” if any range falls below 1 m, otherwise being “0”. This input vector is shown in figure 6.

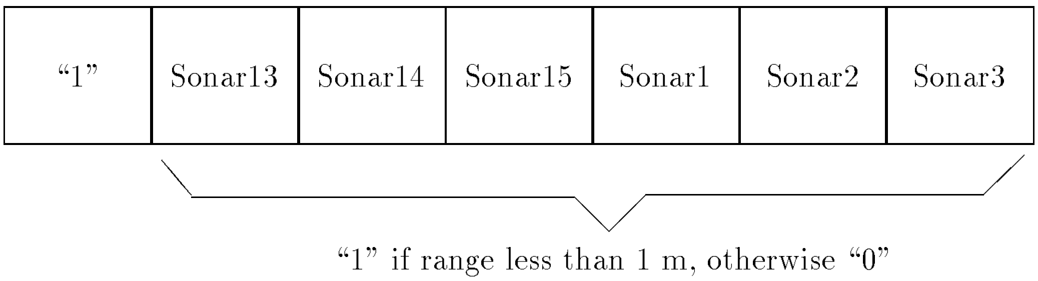

Figure 6: Input Vector Used

As can be seen from figures 5 and 6, different sensors are used in the definition of instinct-rules and input vector. This reflects their different purposes — one is used to assess robot performance, the other to serve as input to the associative memory.

4.2 Mechanism and Results

When the robot is switched on, all network weights are reset to zero, i.e. no previously acquired knowledge is used. This results in an all-zero output of the net (none of the motors will be turning as a consequence of that). Instinct-rule 2 (see figure 5) is violated, and the robot tries to find a motor action to satisfy this violated instinct-rule. By clamping the output nodes to positive-negative, negative-positive and positive-positive outputs in turn, the robot will cycle through a set of three actions (right turn, left turn and forward movement) until an action has been found that satisfies the violated instinct-rule. The network associations between the input vector present at the time of violation and the successful action will be strengthened. Very quickly meaningful associations (meaningful with respect to the instinct-rules shown in figure 5) between sensory perception and motor action develop within the network.

Very soon after this "bootstrapping" phase the robot will not need to clamp the network output. Instead, the continuous response to incoming sensor signals will result in robot actions that maintain the instinct-rules; using the instinct-rules shown in figure 5 the robot moves through its environment, smoothly avoiding obstacles.

Using the described controller and mechanism, the robot acquires the ability to move forward and avoid obstacles in ten to twenty learning steps, in a couple of minutes at the most. After that FortyTwo moves smoothly through its environment, avoiding obstacles by small corrections to the steered course as soon as they become visible in the input (i.e. closer than 1 m). The robot learns to slow down in cluttered areas, and to speed up in open spaces. Turns are gentle if obstacles are far away, and swift if they are detected at close range. The usual motion of the robot is a combination of translation and rotation, the rotational element becoming more pronounced the closer obstacles are. After the first few minutes of learning it is very rare that the acquired sensor-coupling is not sufficient for satisfying all instinct-rules at all times, normally the robot does not have to alter the acquired knowledge after these first few minutes of learning.

5 Summary and Conclusion

We believe that in order to increase the flexibility of robots, facing unforeseen changes such as changes in their environment or sensor failure, the amount of predefined knowledge used in the control strategy has to be kept as low as possible. We have therefore used controllers which *acquire* sensor-motor couplings through the robot's interaction with the environment; that this approach enables robots to successfully interact with their environment, independent from the particular robot's hardware (e.g. sensors) has been shown by both our own experiments with a variety of robots ([Nehmzow *et al.* 89, Nehmzow 92, Nehmzow 93, Nehmzow & McGonigle 94]), as well as the work of other research groups using the same controller ([Daskalakis 91, Ramakers 93, Körtge & Monnerjahn 92]).

Rather than assigning compound motor actions like "swift left turn" etc. to each output node of the associative memory storing the sensor-motor couplings — as done before — we have introduced the concept of artificial motor neurons, driving motors directly. There are a number of advantages in this:

- The range of possible robot actions is now continous, rather than discrete. As the complete range of mechanically possible robot actions is available to the controller, "guessing" of an appropriate subset of robot actions that *might* be needed later by the robot is no longer necessary. This reduces the amount of predefined knowledge needed, and increases the robot's flexibility.
- The approach is computationally cheaper, as it uses only two output units, and furthermore requires no sorting of outputs by magnitude.
- The robot's motion is smooth, the robot initiates the obstacle avoidance behaviour early and continously. This is more energy efficient and results in greater robot speed, because translation and rotation are now executed

at the same time. Furthermore, the robot's behaviour is more predictable to the human observer, which is desirable as a safety aspect.

Our first results with this new, improved learning controller show that the robot acquires the ability to move forward whilst avoiding obstacles quickly and reliably, in a learning process taking a couple of minutes, and less than twenty learning steps.

Acknowledgements

The work reported here has been suported by a grant from the Department of Computer Science, University of Manchester.

References

[Colombetti & Dorigo 93] Marco Colombetti and Mario Dorigo, *Learning to Control an Autonomous Robot by Distributed Genetic Algorithms*, in J.A. Meyer, H. Roitblat and S. Wilson (eds.), *From Animals to Animats 2*, MIT Press 1993.

[Daskalakis 91] Nikolas Daskalakis, *Learning Sensor-Action Coupling in Lego Robots*, MSc Thesis, Department of Artificial Intelligence, Edinburgh University, 1991.

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

[Kaelbling 91] Leslie Pack Kaelbling, *An Adaptable Mobile Robot*, in F. Varela and P. Bourgine (eds.), *Toward a Practice of Autonomous Systems*, MIT Press 1991.

[Körtge & Monnerjahn 92] N. Körtge and J. Monnerjahn, Forschungsberichte Informatik der Universität Bremen, No. 7/92, ISSN -0722-8996, 1992.

[Maes & Brooks 90] Pattie Maes and Rodney Brooks, *Learning to Coordinate Behaviors*, 8th National Conference on Artificial Intelligence, AAAI 1990.

[Mahadevan & Connell 91] Sridhar Mahadevan and Jonathan Connell, *Automatic Programming of Behavior-based Robots using Reinforcement Learning*, 9th National Conference on Artificial Intelligence, AAAI 1991.

[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*, Proceedings of IAS 2, ISBN 90-800410-1-7, Amsterdam 1989.

[Nehmzow 92] Ulrich Nehmzow, *Experiments in Competence Acquisition for Autonomous Mobile Robots*, Ph.D. thesis, Edinburgh University, 1992.

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

[Nehmzow & McGonigle 94] Ulrich Nehmzow and Brendan McGonigle, *Achieving Rapid Adaptations in Robots by Means of External Tuition*, Proc. "Simulation of Adaptive Behaviour 3", Brighton, August 1994.

[Ramakers 93] Wilfried Ramakers, *Investigation of a Competence Acquiring Controller for a Mobile Robot*, Licentiate thesis, Vrije Universiteit Brussel, 1993.

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

[Sutton 91] Richard S. Sutton, *Reinforcement Learning Architectures for Animats*, in Jean-Arcady Meyer and Stewart Wilson (eds.), *From Animals to Animats*, MIT Press 1991.