“PROGRAMMING” BY TEACHING: NEURAL NETWORK CONTROL IN THE MANCHESTER MOBILE ROBOT

Paul Martin and Ulrich Nehmzow*
Department of Computer Science
Manchester University
Manchester M13 9PL
United Kingdom
pmartin@cs.man.ac.uk *u.nehmzow@cs.man.ac.uk*

Abstract

The paper presents experiments conducted with the Manchester mobile robot *FortyTwo* on the application of reinforcement learning to robot control. An artificial neural network, forming the core component of the controller presented here, associates incoming sensor signals with corresponding motor actions. Teaching the network is achieved by initially operating the robot under human control (i.e. the robot's behaviour is the result of training, not programming). The observed actions are used to train the robot's associative memory, and after a very short training time *FortyTwo* becomes able to perform the required task autonomously. Experiments are presented in which data from sonar, infrared or vision sensors is associated with motor actions; and a suitable vision preprocessing system is discussed.

Keywords: Mobile robots, neural networks, machine learning, vision, image processing, sensor fusion.

1 INTRODUCTION

Provided mathematical models of a plant that is to be controlled are available, classical control methods or adaptive control can successfully be applied (see, for instance, [Åström 89]). Speed control, orientiation control and force control are but three examples that come to mind.
If, on the other hand, robots are to be used in unstructured environments, if models of robot and environment are only partially or not at all available, reinforcement learning techniques provide a viable, if not necessary alternative ([Barto 89, Sutton 91]. *FortyTwo*, the Manchester mobile robot, has successfully demonstrated that such techniques enable robots to learn from reinforcement, successfully and in real time ([Nehmzow 94, Nehmzow 95]).

The experiments presented in this paper have taken our previous approach to robot control ([Nehmzow 92]) — *autonomous* competence acquisition by means of connectionist computing — further: feedback for the reinforcement learning process is provided by the user, allowing the robot to acquire task achieving competences through observation of a human operator.
This approach is advantageous for industrial applications: as the robot can very quickly be trained to perform the desired task, the same robot can be used for a variety of different tasks without the need for (expensive) reprogramming, thus reducing setup costs. Contrary to reinforcement learning architectures enabling autonomous competence acquisition (for example [Mahadevan & Connell 91, Kaelbling 91, Nehmzow 92]) learning here is supervised by a human operator, an aspect that can be relevant regarding safety aspects of industrial applications.

*Published at Intern. Conf. Intelligent Autonomous Vehicles 95, Helsinki. Names appear in alphabetical order, with both being principal authors.

Figure 1: The Manchester Mobile Robot, *FortyTwo*

2 THE ROBOT

FortyTwo (see figure 1), a Nomad 200 mobile robot, is fully autonomous and operates independently from any external components, such as power supplies or computers. The robot is 75 cm high, weighs 60 kg, and uses a distributed controller hardware with a 486 PC as main controller. The robot has 16 long-range ultrasonic sensors (range up to 10.6 m), 16 short-range infrared sensors (range up to 60 cm), 20 tactile sensors, wheel encoders and a visual sensor (monochrome camera, 492 x 512 pixels, focal length 12.5 mm).

3 THE CONTROL SYSTEM

The central element of the controller used in the experiments presented here is an associative memory (see figure 2), implemented by means of an artificial neural network. It associates incoming (possibly preprocessed) sensor signals (input vector $\mathbf{i}$) with the robot's motor actions(outputs o_j):

$$o_j = \mathbf{w_j}\mathbf{i}, \qquad (1)$$

with $\mathbf{w_j}$ being the weight vector of output unit j. Training this network is achieved through a supervised learning process, in which the operator's commands (issued by joystick control) are used to generate the necessary teaching signals τ (see equation 2).

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

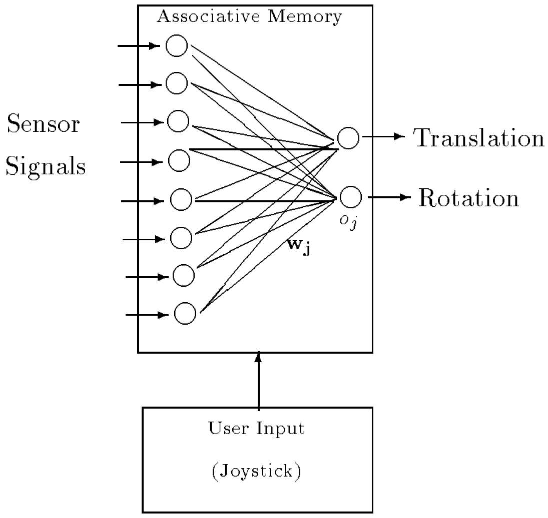

Figure 2: The Controller Architecture

$\eta(t)$ is the learning rate (or "gain"), specifying the degree of change applied at each learning step (η was initially set to 0.3 in our experiments, and reduced by 1% each learning step).

As a result of the training process, meaningful associations between the robot's sensory input and its actuator responses develop. The generalisation property of artificial neural networks is exploited here: the developing links generate meaningful response patterns of the robot even in situations not encountered before.

4 SONAR AND INFRARED SENSORS FOR ROBOT CONTROL

4.1 *Experimental Setup*

The experiments were conducted in a room of 5 m x 15 m, containing solitary, stationary objects (cardboard boxes). The walls were white (i.e. reflecting infrared light well; but there were also cluttered areas containing dark chairs and tables). The input vector used for the experiments is shown in figure 3, it contained range information from the front facing sonar sensors, thresholded at 150 cm, and range information from the front-facing infrared sensors, thresholded such that any triggering of an infrared sensor would result in a "1" signal.

The input vector was normalised.

5 values	5 values	6 values	6 values
Left facing Sonars	Right facing Sonars	Left facing IRs	Right facing IRs

Figure 3: The Input Vector

4.2 *Obstacle Avoidance*

Initially driving the robot with the joystick, *FortyTwo* was trained to avoid convex obstacles and cul-de-sacs. The robot learned this in less than twenty learning steps (one step being one application of equation 2), taking a few minutes in real time. The resulting obstacle avoidance motion is smooth, and incorporates translational and rotational movements at varying speeds, according to the strength of association between sensory perception and corresponding motor action (see also [Nehmzow 94]).

4.3 *Wall Following*

In the same manner as before, the robot was trained to stay within a distance of about 50 cm to the wall. As the only relevant part of the input vector for this task is the infrared one, the robot moved closer towards dark objects (little infrared reflection), and further away from light objects. Again, learning was achieved within twenty learning steps, and the acquired motion was smooth. The robot can be trained to follow left hand walls, right hand walls, or both.

4.4 *Box Pushing and Clearing*

The robot can also be trained to move towards objects visible to the infrared sensors. A box placed in front of the robot is an example for this, and the resulting behaviour is that of box pushing. The robot was taught to turn in such a way that an infrared stimulus straight ahead was maintained — when the box slipped sideways, the robot followed it.

In our experiments cardboard boxes of about 50 cm height were used. These boxes were detectable by the (low mounted) infrared sensors, but not by the (high mounted) sonar sensors. Consequently, the robot could be taught to abandon boxes once it got close to a wall: the infrared part of the input vector became associated with object-following behaviour, the sonar part of the input vector with obstacle avoidance behaviour. This resulted in a behaviour in which the robot pushed boxes, but abandoned them once near a wall ("clearing behaviour").

4.5 *Cleaning, and Route Learning*

Again using the same training technique, the robot can be taught to move between parallel walls (e.g. corridors) in a meandering manner, which can be used for cleaning or inspection applications.

Simple routes can equally be learned, associating "sensory landmarks" in the environment with motor actions. For example, *FortyTwo* was taught to follow the walls of the laboratory until encountering an open door, then to go through the door, and return back into the laboratory, resuming the circular trip.

In all these experiments a low number of learning steps is required. For complicated routes, this may mean some 50 learning steps (about 15 minutes in real time), but typically the number is about a couple of tens.

5 USING COMPUTER VISION FOR ROBOT CONTROL

5.1 *Video Data Preprocessing*

Having conducted the experiments described before, we were interested to see whether the controller shown in figure 2 could be used to control the robot using the vision sensor.

The information content of visual data, regarding fundamental tasks like obstacle avoidance, is far less accessible than that of, for instance, sonar range data. Whilst the latter provides direct indications as to the presence and distance of objects, the latter merely supplies a distribution of grey level values over the visual field of the camera.

In order to overcome this fundamental problem of image processing, the information coming from the CCD camera was preprocessed before being used as input to the artificial neural network. These preprocessing stages, some of which have counterparts in biological vision systems (for example edge detection ([Marr & Hildreth 80]) and orientational selectivity ([Campbell & Robson 68])), are described in the following sections.

The complete vision preprocessing system is shown in figure 4.

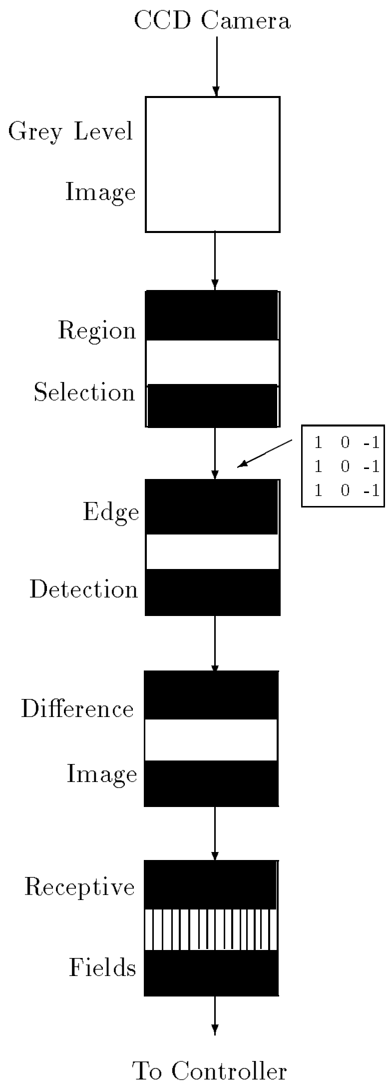

Figure 4: The vision preprocessing system.

Narrowing the Field of View

In the first stage, the amount of data is reduced by limiting the field of view (see, for example, [Dickmanns & Graefe 88] for an application of this to vehicle control).
As it is assumed here that the robot will operate in environments containing tall objects, rather than small objects on the floor or objects suspended from the ceiling — a justified assumption for environments containing walls, doors, furniture and people. This allows the field of view to be limited to the central 20% of the image, discarding the top and bottom 40% of the data supplied by the CCD camera. The resulting image size is 320 x 50 pixels. Obviously, for some applications (like the detection of descending stairs) other windows may have to be chosen.

Detecting Vertical Edges

To control robot motion, texture and shade of objects are less useful than position and size of objects. Therefore, in a second stage of visual data preprocessing texture and shade information are discarded, and vertical edges in the image only are retained. This is achieved by applying an edge detecting operator as shown in figure 4.
Based on the assumption that every object that is to be detected by the vision system has some vertical edges (boxes, people, tables, chairs, doors and walls all fulfill this assumption), not only texture and shade information are eliminated, but also information about horizontal edges, which further reduces the amount of data to be processed.

Differentiation

For applications of computer vision to robot control *distance information* is the most useful. The *optical flow*, i.e. the movement of pixels resulting from robot motion, can be used to compute distance to an object provided the *correspondence problem* (identifying the pixel belonging to the same object in subsequent frames) is solved (see, for instance, [Marr 82, p.212ff]). This, however, is difficult, particularly if the environment the robot is operating in is (partially) unknown.
As a vision preprocessing system, generating sensor information for the artificial neural network used in the controller, however, optical flow can be used *without* having solved the correspondence problem. By computing the difference between subsequent frames (whilst the robot is in forward motion) changes in the image will correspond with distance to objects (far away objects will generate little optical flow, whilst nearby objects will generate a large one). This information is sufficient for the network to acquire useful sensor-motor couplings.
Therefore, a differentiation is performed in the following manner: If the difference between pixels in subsequent frames exceeds a threshold of eight, the respective pixel is set to '1', otherwise set to '0' (the threshold value of eight was arrived at by experimentation). This provides sufficient immunity against background noise whilst still detecting changes between subsequent frames reliably.

Receptive Fields

The final input vector to the control system is generated by dividing the image of 320 x 50 pixels into 16 receptive fields of 20 x 50 pixels each.
The number of '1' pixels in each receptive field is determined, the whole input vector (consisting

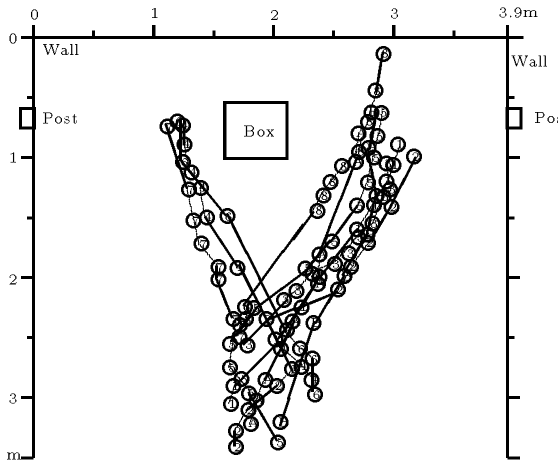

Figure 5: Obstacle Avoidance, using optical input.

of 16 integer values) is then normalised and presented to the robot controller (figure 2).

One important property of the vision preprocessing system discussed here is that it is computationally cheap and can successfully be used on a 486 PC, in real time. The maximum speed of the robot, under visual guidance, was 37 cms^{-1}. (Kosaka and Kak, for example, report mobile robotics experiments on vision and model based reasoning for landmark recognition, in which the robot travels at 15 cms^{-1} ([Kosaka & Kak 92])).

5.2 *Experimental Setup*

The following experiments were conducted in the same room as the previous experiments. The white walls of the room were covered with dark vertical strips at approximately 30 cm intervals, to allow visual detection of the otherwise invisible walls.

5.3 *Obstacle Avoidance*

In this experiment, the camera was mounted facing the direction of travel, pointing downwards at an angle of about 30 degrees. The camera was focussed at approximately 75 cm. Only preprocessed vision data (see figure 4) was used in this experiment, infrared and sonar sensors were not used.

The robot was trained to approach an obstacle (a stack of cardboard boxes) and to pass it either on the left or the right hand side (see figure 5). The robot learned to perform this task autonomously after about 10 learning steps.

Because vision input is only generated when subsequent frames differ, the robot learned to perform additional rotational movements as well as translational ones when approaching an obstacle. This would generate suitable visual stimuli for the vision preprocessing system to generate input vectors to the network. Once the object was sufficiently clearly situated on one side of the robot, *FortyTwo* would pass it using translational movements only.

Depending on the sensory stimuli received, *FortyTwo* moved at varying speeds whilst approaching and avoiding the obstacle, the average speed being about 5 cms^{-1} and the maximum speed being 37 cms^{-1} (the robot rarely travelled at this speed).

5.4 *Wall Following*

In this experiment, the camera was placed pointing about 20 degrees downwards, and about 20 degrees to the left of the center of the robot. The other camera parameters were set as before.

The robot was trained to maintain a distance of about 70 cm to a left-hand wall, using preprocessed visual data only. The robot acquired this competence after about 10 learning steps and was then able to move and remain parallel to the wall. However, if the visual input changed considerably from that observed during training (i.e. a sideways view of the wall) no successful wall following was possible. In one of the experiments the robot began to face the wall directly, and became unable to resume a parallel path to the wall (see figure 6). This was due to the fact that the robot had not been trained to avoid head-on collisions, but only to stay parallel to a left-hand wall.

Depending on the sensory stimuli received, *FortyTwo* moved at varying speeds whilst following a left hand wall. The average speed was approximately $6cms^{-1}$, the maximum speed $9cms^{-1}$.

6 SUMMARY AND CONCLUSION

Experiments are presented in which a mobile robot is *trained* by a human operator to perform tasks such as obstacle avoidance, wall following, box pushing, cleaning, clearing and route following. The core element of the controller is an associative memory, implemented by means of an artificial neural network. In this associative memory incoming sensor signals are associated with motor actions. Input from various sensors of the robot — infrared, sonar and visual — has been used in

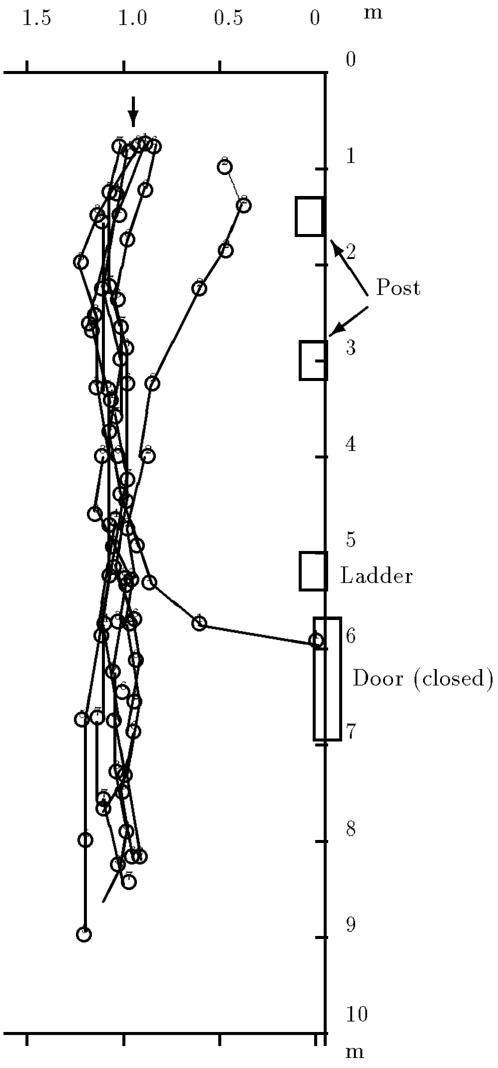

Figure 6: Wall-following, using optical input.

the experiments described, and a vision data preprocessing system has been presented that, whilst computationally cheap, processes visual data in such a way that the controller can successfully associate such stimuli with actuator responses.

FortyTwo, the Manchester mobile robot, acquires sensor-motor couplings rapidly, in normally less than twenty learning steps taking a few minutes in real time.

The controller architecture discussed here seems applicable to certain industrial applications. For small scale tasks or industrial applications which change often it is not feasible to program a robot explicitly, programming being too labour-intensive and time consuming. *Training* offers an alternative in these cases, as the same robot can rapidly be trained and retrained to perform the desired task. Training fundamental tasks proves to be fast and easy from the operator's point of view. It is conceivable that robots with a learning capability as described here can be shared between many users (robot pool, robot hire), and thus be amortised more quickly than custom-programmed ones.

References

[Åström 89] Karl Johan Åström, *Toward Intelligent Control*, IEEE Control Systems Magazine, April 1989.

[Barto 89] A. Barto, *Connectionist Learning for Control: An Overview*, Department of Computer and Information Science, University of Massachusetts at Amherst, COINS technical report 89-89.

[Campbell & Robson 68] F.W.C. Campbell and J. Robson, *Application of Fourier analysis to the visibility of gratings*, J. Physiol. (London) 197, 551-566.

[Dickmanns & Graefe 88] Ernst Dieter Dickmanns and Volker Graefe, *Dynamic monocular machine vision*, Universität der Bundeswehr München, Technical Report UniBwM/LRT/WE 13/FB/88-3.

[Kosaka & Kak 92] Akio Kosaka and Avinash Kak, *Fast Vision-Guided Mobile Robot Navigation Using Model-Based Reasoning and Prediction of Uncertainties*, CVGIP: Image Understanding, Vol 56 No. 3, pp. 271-329, 1992.

[Kaelbling 91] Leslie Kaelbling, *An Adaptable Mobile Robot*, in F. Varela and P. Bourgine, *Toward a practice of autonomous systems*, MIT Press 1992.

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

[Marr & Hildreth 80] D. Marr and E. Hildreth, *Theory of edge detection*, Proc. R. Soc. Lond. B 207, 187-217 (after [Marr 82]).

[Marr 82] David Marr, *Vision*, W.H. Freeman and Company, San Francisco 1982.

[Nehmzow 92] Ulrich Nehmzow, Experiments in Competence Acquisition for Autonomous Mobile Robots, PhD thesis, University of Edinburgh 1992.

[Nehmzow 94] U. Nehmzow, *Acquisition of Smooth, Continuous Obstacle Avoidance in Mobile Robots*, in H. Cruse, H. Ritter and J. Dean (eds.). Proc. Workshop "Prerational Intelligence in Robotics", ZIF, University of Bielefeld, 1994.

[Nehmzow 95] U. Nehmzow, *Flexible Control of Mobile Robots through Autonomous Competence Acquisition*, Trans. Institute for Measurement and Control (IMC), 3/1995.

[Sutton 91] R. Sutton, *Reinforcement Learning Architectures for Animats*, in J.A. Meyer and S. Wilson (eds), *From Animals to Animats*, MIT Press 1991.