Applications of Robot Training: Clearing, Cleaning, Surveillance

Ulrich Nehmzow
Department of Computer Science
University of Manchester
Manchester M13 9PL
u.nehmzow@cs.man.ac.uk*

Robot training, as opposed to robot programming or autonomous learning in robots, provides a workable solution if robots are to be used in a variety of frequently changing tasks, or in frequently changing environments. Whilst re-programming the robot is too costly for such situations, experiments presented here show that robot training can successfully be utilised to achieve rapid acquisition and re-acquisition of sensor-motor couplings in robots.

Using a neural network controller, the Manchester mobile robot FortyTwo *was trained to perform a number of different tasks, such as obstacle avoidance (random exploration and surveillance), wall following, object pushing, clearing, cleaning and simple route learning. Initially receiving feedback from a human operator by means of joystick control, the robot acquired these competences in five to ten minutes, and performed them autonomously afterwards.*

Keywords: Mobile robots, neural network control, robot training, machine learning, industrial applications of robotics, service robots, human computer interaction.

1 Introduction

For robotic tasks that are to be performed repeatedly and in structured environments, fixed installations (both hardware and software) for robot control are viable. Many industrial applications fall into this category, for example mass assembly tasks, or high-volume transportation tasks. In these cases, fixed hardware installations (robot assembly lines, conveyor belts, *etc.*) and the development of fixed, one-off control code are warranted. As robot hardware technology advances, sophisticated robots become available at constantly decreasing cost and even small workshops and service companies become interested in robotics applications ([Schmidt 95, Spektrum 95]), the development of control software becomes the governing factor in cost-benefit analysis. For low-volume tasks programming and re-programming the robot is not viable.

In this paper experiments with the Manchester mobile robot *FortyTwo* (see figure 1) are presented, in which the robot is *trained* to perform a variety of different tasks. Simple sensor-motor competences such as obstacle avoidance, random exploration or wall following, as well as more complex ones such as clearing objects out of the way, area-covering motion in corridors ("cleaning") and learning simple routes are achieved by this method, without the need to alter the robot control code. During the training phase the robot is controlled by a human operator, and uses the received feedback signals to train an associative memory (an artificial neural network). After a few tens of learning steps, taking five to ten minutes in real time, the robot performs the required task autonomously. If task, robot morphology or environment change *re-training*, not *re-programming* is used to regain the necessary skills.

1.1 Related work

The method of providing external feedback to the learning process is as yet relatively rarely used in robotics. Shepanski and Macy ([Shepanski & Macy 87]) use an operator-taught multilayer perceptron to achieve vehicle-following behaviour in a *simulated* motorway situation. The network learns to keep the simulated vehicle at an acceptable distance to the preceding vehicle after about one thousand learning steps. Colombetti and Dorigo ([Colombetti & Dorigo 93]) present a a classifier system with a genetic algorithm that enables a mobile robot to achieve phototaxis. *Autonomouse*, the robot used, began to show good light seeking behaviour after about 60 minutes of training time.

*Paper presented at International Workshop on Advanced Robotics and Intelligent Machines, Salford, UK, 5.-6.4.95

Unsupervised learning has also been used in robot control. Tasks such as obstacle avoidance and contour following ([Nehmzow 95]), box pushing ([Mahadevan & Connell 91, Nehmzow 95]), leg coordination ([Maes & Brooks 90]) or phototaxis ([Kaelbling 92, Colombetti & Dorigo 93, Nehmzow & McGonigle 94]) have successfully been implemented on mobile robots.

Teach-by-guiding ([Critchlow 85, Schmidt 95]) is still a common method of programming industrial robots. This method is different to work based on artificial neural networks, in that it does not show the generalisation properties of networks, but merely stores chains of locations to be visited in sequence.

2 Robotic Platform

The experiments presented here were conducted using a Nomad 200 mobile robot called *FortyTwo*, as shown in figure 1. This is a hexagonal mobile robot of 75 cm height, weighing 60 kg and capable of carrying a payload of 23 kg.

Figure 1: *FortyTwo*, the Nomad 200 mobile robot.

The robot is equipped with sixteen ultrasonic range finding sensors (range up to 10.6 m), sixteen infrared sensors (range up to 60 cm), twenty tactile sensors, a monochrome CCD camera (492 x 512 pixels, focal length 12.5 mm), a fluxgate compass and on-board odometry. In figure 1 seven of the sixteen sonar sensors are visible at the top of the robot. The infrared sensors are located in the middle of the robot, and 2x10 tactile sensors are incorporated in the two rubber bumpers visible at the bottom of the robot.

The robot is driven by three AC motors, for translation, steering and turret rotation respectively. The maximum translational speed of the robot is 50 cms^{-1}, the maximum steering and turret velocities are $60°s^{-1}$ and $90°s^{-1}$ respectively.

The main controller is a 486 PC, slave controllers use Motorola 68HC11 microprocessors to control sensors and actuators.

The robot is equipped with a speech synthesizer, which in our experiments is used to enhance the man-machine interface by giving acoustic information to the human operator during the teaching phase.

3 Controller Architecture

The central component in the controller used in the experiments reported here is an associative memory, implemented through an artificial neural network, a *Pattern Associator* ([Kohonen 88, McClelland & Rumelhart 88]). The controller is shown in figure 2.

Inputs to the associative memory consist of preprocessed sensor signals. In the experiments presented here sensor signals from the robot's sonar and infrared sensors have been used, and preprocessing was limited to a simple thresholding operation, generating a '1' input for all sonar range signals of less than 150 cm distance (see figure 3). Experiments using visual input data are reported elsewhere ([Martin & Nehmzow 95]), in these experiments thresholding, edge detection and differentiation were used in the preprocessing stages.

Output signals of the associative memory are computed according to equation 1, they are analogue signals:

$$o_j = \vec{w_j}\vec{\imath}. \tag{1}$$

The two analogue output nodes of the associative memory drive the steering and translation motor of the robot, respectively (in the experiments presented here steering and turret rotation are locked, so that the front of the robot is always facing the direction of travel). This generates continous, smooth steering and translational velocities, depending on the strength of the association between current sensory stimulus and motor response (see also [Nehmzow 94]): the robot moves fast in situations which have been trained frequently and therefore have strong associations between sensing and

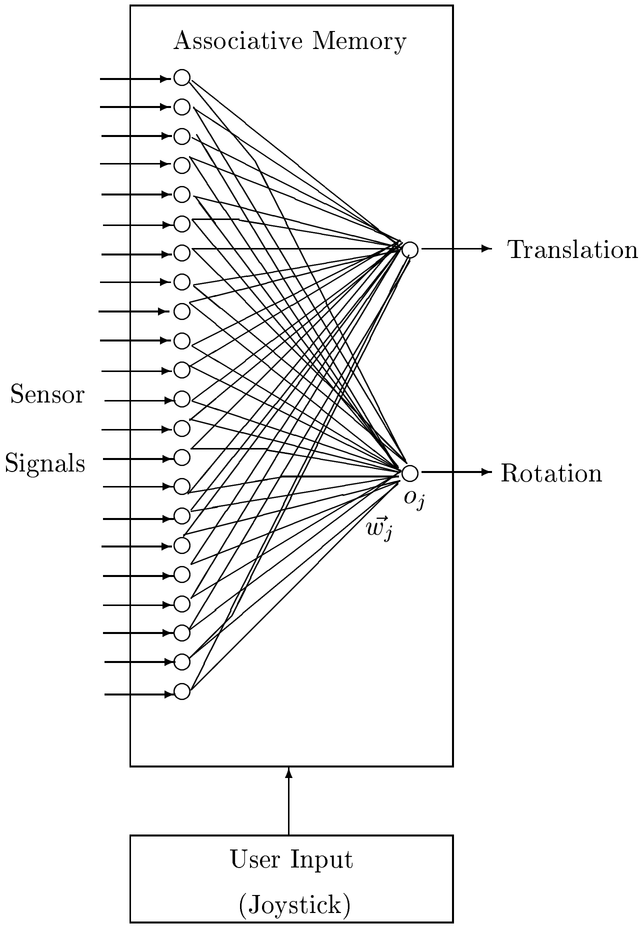

Figure 2: The Controller Architecture

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 used.

action ("familiar" situations), and slowly in "unfamiliar" situations. If sensor signals are received which have never been encountered during the training phase, i.e. situations in which the network weights between sensory input and motor action output are zero, the robot does not move at all.

3.1 Training the network

Initially, there are no sensor-motor associations stored in the associative memory (i.e. all network weights are set to zero), and the robot is driven by the operator by means of the joystick. Information about the current environment (the input vector $\vec{\imath}$ to the controller) and the desired motor action in this situation (yielding the desired output τ_j for each of the two output units of the network) is therefore available to the robot controller in the training phase.

Adjusting the weight vector $\vec{w}_j$ of output unit j is performed according to equation 2:

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

o_j is the current output of output unit j, and $\eta(t)$ the learning rate, determining the extent of change at each learning step. η was chosen at 0.3, and decreased by 1% every learning step. This eventually stabilises the associations stored in the associative memory of the controller. There are other ways conceivable of chosing the learning rate: an initially high, but rapidly falling rate will result in a robot learning only during the initial stages of the learning process, whilst a constant η will lead to continous learning behaviour. Thirdly, by means of a novelty detector a low η could be increased when drastic changes are detected.

4 Experiments

The experiments described here were conducted in a laboratory of 5 m x 15 m, containing tables, chairs and boxes which were detectable by the robot's sensors.

Initially, there are no sensor-motor associations stored in the associative memory, and the robot is driven by the operator by means of the joystick. Using this information for training, meaningful sensor-motor couplings are developed within the artificial neural network, and the robot improves rapidly in performing a specified task. After a few tens of learning steps the acquired information is usually sufficient to control the robot without any user intervention, the robot then performs the task autonomously, solely under network control.

4.1 Boxpushing and clearing

Boxpushing

Both infrared sensor and sonar sensor range data is part of the input vector presented to the associative memory of the controller (see figure 3), and can therefore be used by the controller to associate action with perception. *FortyTwo's* infrared sensor are mounted low, about 30 cm above ground, whilst the sonar sensors are mounted hight (ca. 60 cm above ground).

In a boxpushing experiment the robot was taught to move forward if a low object was detected straight ahead, and to turn left or right if an object was detected to the left or the right side of the robot respectively. During the training phase of about 30 learning steps associations between the robot's infrared sensors (which are relevant to this task) developed, whilst sonar data delivers contradictory information in this experiment and strong associations between the 'sonar part' of the input vector and the motor-driving outputs do not develop.

The robot acquired the ability to push a box quickly, in a few minutes of real time. After the initial training phase the robot was able to push and follow a box autonomously, staying behind the box even if the box moved sideways.

Clearing

As the infrared sensors of the robot return identical signals for *any* object placed near the robot, it is impossible for the controller to differentiate between boxes, walls, or people. For example, the robot will attempt to push walls, as much as it will push boxes.

There is, however, a way of training *FortyTwo* to abandon boxes near tall obstacles: as the high mounted sonar sensors reach beyond a low box, the robot can be trained to pursue objects as long as they appear only in the 'infrared part' of the input vector, but to abandon them as soon as tall obstacles appear in the 'sonar part' of the input vector. Training for about 30 to 50 learning steps, taking between five and ten minutes in real time, the robot can be taught to acquire such a "clearing" ability. After the training phase, the robot will pursue any object becoming visible to the infrared sensors only, and push it until tall obstacles are detected by the sonar sensors. *FortyTwo* then turns away from the object, looking for new boxes away from the tall obstacles. The resultant behaviour is one where the robot autonomously pushes cardboard boxes towards walls, and deposits them along the perimenter of the room.

4.2 Surveillance and Route Learning

Surveillance

Using the same method, and without any need to reprogram the robot, *FortyTwo* can be trained to move in random directions whilst avoiding obstacles. During the training phase of approximately 30 learning steps, the robot is instructed to move forward when no objects are detected, and to turn away from obstacles once they become visible to the robot's sensors. This develops associations between the sonar and infrared sensors and the motor-driving output units of the network such that the robot acquires a general obstacle avoidance behaviour, also for input constellations that have not been encountered during the training phase (this is a result of the generalisation capability of the artificial neural network).

The resultant obstacle avoidance behaviour is smooth and continous; fast translational movement occurs when familiar sensor states indicating free space are detected. Smooth turning action is perceived when distant obstacles are detected, rapid turning action results from nearby obstacles being detected. The smooth motion of the robot is achieved through the direct association of the (analogue) outputs of the network with motor action, and the output itself being dependent on the strength of the input signal received (nearby objects will produce stronger input signals).

Such random exploration and obstacle avoidance behaviour forms the basis of a surveillance function of the robot. Using this behaviour, *FortyTwo* will, for example, follow corridors, avoiding moving and stationary obstacles as it moves.

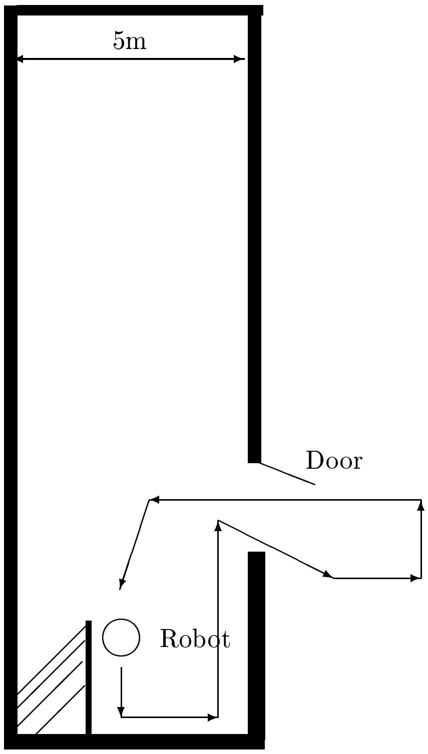

Figure 4: Route Learning.

Route Learning

As the robot effectively learns to immediately associate sensory perception with motor response, it can be taught to perform certain motor actions at particular locations, thus performing navigation. This is navigation along "perceptual landmarks", in which the perceptual properties of the environment are used to perform the desired motion at each physical location. Therefore, if two physically distinct, but perceptually identical locations are encountered, the controller is unable to steer the robot differently at the two locations. The way to solve this problem is to make the physically distinct locations perceptually distinct as well, by either enhancing the sensor capabilities of the robot, or adding distinguishing features to the perceptual landmarks.

We have trained *FortyTwo* to follow a route as shown in figure 4. After about fifteen minutes of training time the robot was able to follow the wall in the way indicated in figure 4, leave the laboratory through the door (which had about twice the width of the robot's diameter), turn and return through the door back in to the laboratory and resume the path.

4.3 Cleaning

Cleaning tasks require the robot to cover as much floorspace as possible. For large, open spaces random

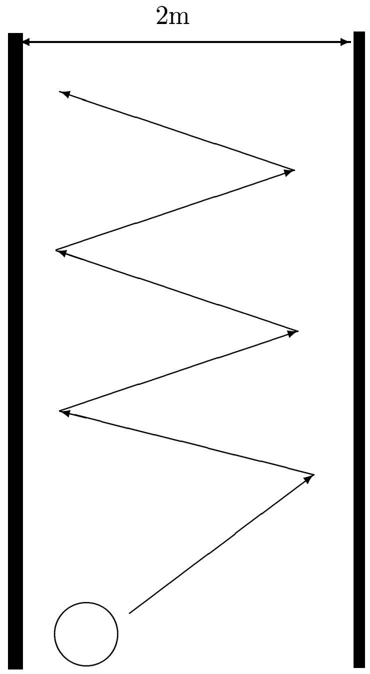

Figure 5: Cleaning operation.

exploration as described in section 4.2 might be sufficient to accomplish the task over time. However, it is possible to train *FortyTwo* to cover floorspace in a more methodical manner.

Using the same training method as before, the robot was trained to move along a corridor in the Computer Science Department in the manner shown in figure 5. Provided cleaning equipment is attached to the robot (which was not the case here) such motion is suited for cleaning floors in a methodical way.

As before, learning is fast and the robot acquires the competence in a few tens of learning steps.

5 Summary and Conclusions

5.1 Summary

With decreasing robot hardware costs the application of robots in low-volume or infrequent tasks becomes conceivable. It is obvious that as hardware costs decrease the cost of control software development becomes the governing factor in cost-benefit analysis, and will increase further in importance as hardware costs continue to fall. Frequent re-programming of robots is time-consuming (a matter of several hours for small changes of existing code, to several weeks for development of new code in a teach-in robot — [Schmidt 95]) and prone to error. For low volume and infrequent tasks other ways of achieving the desired robot behaviour have to be sought.

In experiments with the Manchester mobile robot *FortyTwo* training has been used as instruction method, rather than programming or teach-in methods: an associative memory stores associations between incoming sensor signals and desired motor responses. Training of the associative memory (which is implemented as an artificial neural network) is achieved during an initial training phase, during which the robot is controlled by a human operator. This training phase typically lasts for five to ten minutes.

Using this method, it was possible to train *FortyTwo* to perform random exploration and obstacle avoidance (surveillance), contour following, object pushing, clearing, cleaning and simple route following tasks.

5.2 Conclusions

If learning robot controllers are to be used successfully in industrial applications, the following criteria have to be fulfilled:

1. very fast competence acquisition,
2. generalisation of acquired knowledge,
3. safe operation,
4. intuitively clear man-machine interface.

1. Reinforcement learning mechanisms can be slow in learning. For controller architectures employing Q-learning, several thousand learning steps are not uncommon ([Kaelbling 92, Sutton 91]). Some researchers abandon experiments with robots for this reason, and resort to simulation ([Kaelbling 90]. Similarly, Multilayer Perceptrons — albeit having been used in robot control related applications (see, for instance, [Shepanski & Macy 87]) — show slow convergence rate and typically require several several hundred learning steps.

The Pattern Associator, the associative memory used here, learns extremely fast; within a few learning steps meaningful associations between inputs and outputs develop. However, the network is only able to learn linearly separable functions ([Minsky & Papert 88]) and is therefore not suitable for all types of control task.

2. One property of artificial neural networks is their ability to generalise, i.e. to acquire input-output associations for input situations that have not been encountered explicitly. For large sensor spaces it is practically impossible that the robot encounters all possible sensor states during the training phase — the generalisation ability of artificial neural networks provides a solution to this problem.

3. Some properties of safe operation can be observed here: as sensor stimuli are directly associated with (analogue) motor responses, the strength of an association determines the velocity of the action. This means in practice that the robot moves fast in familiar territory, and slowly in unfamiliar territory; that the robot turns rapidly if nearby obstacles are detected, and gently in the presence of distant ones. This property is emergent, and not a design feature of the controller.

4. "Programming" through teaching provides a simple and intuitively clear man machine interface. Furthermore, as the operator is able to instruct the machine *directly*, without the need of a middle man (the programmer), the risk of ambiguities is reduced. Using artificial neural networks as discussed here furthermore enables the operator to specify solutions to the control problem even when explicit rules are unknown.

One feature of the controller discussed here is that it is the *robot* that interprets its own sensor signals, not the human programmer. This feature is considered important, as this is the most direct way of establishing sensor-motor couplings; the human operator is not necessarily the best instance to define these couplings, as his own experience of the world, through rich senses, differs greatly from that of a sensor-impoverished robot.

There remain a number of research issues that merit further investigation. In particular, tools for controller analysis are as yet lacking. Formal verification of the robot's behaviour remains a research issue. Secondly, the question of sensor data representation (sensor data preprocessing) deserves more attention; and lastly, it is unclear how complex a task can be accomplished using this method.

Currently we are concentrating on the latter question, investigating navigation and learning of complex routes, using the controller described here. The *Electronic Porter* project at Manchester is developing a mobile service robot that can perform surveillance, delivery, guidance and cleaning tasks, as well as navigation over medium distances in man-made environments.

References

[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.

[Critchlow 85] Arthur Critchlow, *Introduction to Robotics*, Macmillan 1985.

[Kaelbling 92] Leslie Kaelbling, *An Adaptable Mobile Robot*, in F. Varela and P. Bourgine (eds), *Toward a Practice of Autonomous Systems*, MIT Press, 1992.

[Kaelbling 90] Leslie Kaelbling, *Learning in Embedded Systems*, PhD thesis, Stanford University, 1990.

[Kohonen 88] Teuvo Kohonen, *Self Organization and Associative Memory*, Springer Verlag, Berlin, Heidelberg, New York, 2nd edition, 1988.

[Minsky & Papert 88] M. Minsky and S. Papert, *Perceptrons*, MIT Press, 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.

[Martin & Nehmzow 95]
Paul Martin and Ulrich Nehmzow, *"Programming" by Teaching: Neural Network Control in the Manchester Mobile Robot*, Intern. Conf. "Intelligent Autonomous Vehicles", Helsinki, June 1995.

[McClelland & Rumelhart 88] J. McClelland and D. Rumelhart, *Explorations in Parallel Distributed Processing*, MIT Press 1988.

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

[Nehmzow & McGonigle 94] Ulrich Nehmzow & Brendan McGonigle, *Achieving Rapid Adaptations in Robots by Means of External Tuition*, in D. CLiff, P. Husbands, J.A. Meyer and S. Wilson (eds.), *From Animals to Animats 3*, MIT Press 1994.

[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*, Measurement and Control, Vol. 28, March 1995.

[Shepanski & Macy 87] J.F. Shepanski and S.A. Macy, *Manual Training Techniques of Autonomous Systems Based on Artificial Neural Networks*, IEEE 1st International Conference on Neural Networks, San Diego 1987.

[Schmidt 95] Dietmar Schmidt, *Roboter als Werkzeuge für die Werkstatt*, Spektrum der Wissenschaft, March 1995, pp. 96-100.

[Spektrum 95] Spektrum der Wissenschaft, March 1995, pp. 96-115.

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