Invited Paper for Symposium on *Practical Applications of Neural Networks in Control*, The Institute of Measurement and Control, London, 28th September 1994. Published in **Measurement and Control, Vol. 28, No. 2, March 1995.**

Flexible Control of Mobile Robots through Autonomous Competence Acquisition[1]

Ulrich Nehmzow

Department of Computer Science
University of Manchester
Manchester M13 9PL
Fax: 061 - 275 6236
u.nehmzow@cs.man.ac.uk

Abstract

A self-organising controller for mobile robots is presented that allows robots to *acquire* sensor-motor competences autonomously, and thus adapt to changes in the environment, the task, or the robot itself, without external supervision and feedback.

Using artificial neural networks, the robot acquires these fundamental sensor-motor competences rapidly, in a couple of learning steps, taking a few tens of seconds.

Results of experiments with a number of mobile robots are presented, both by us and by other research groups, using the same controller architecture.

1 Introduction

As mobile robot technology advances and sophisticated mobile robots become widely available, at comparably low cost — the Nomad 200 robot described in section 4.1 of this paper being one example — the question of how to control such robots *intelligently* has become more interesting, both from a research and an industrial point of view. *Intelligent control* here means that the robot has the ability to generate its own sensor-motor couplings (i.e. determine autonomously its motor response to sensory stimuli), adapt to changes in world, task or its own morphology without external intervention, and to acquire and maintain task-achieving behaviour in the presence of noise, variation and change, which are always inherent in the real world.

This paper presents experiments conducted with a number of mobile robots which demonstrate that fast acquisition of sensor-motor competences in robots is possible, and that this allows robots to deal with unforeseen circumstances efficiently.

The robot controller presented here — common to all robots used in the experiments — uses an artificial neural network in the feedforward path of the control loop, mapping incoming sensor signals to motor actions and altering that mapping if changes in the world, the task, or the robot itself occur.

1.1 Control Theory and Adaptive Control

Provided a mathematical model of the plant that is to be controlled is known, and accurate measures of performance are available, control theory can be applied. *Adaptive control* relies on the same assumptions, but allows individual parameters within the known model to be adapted, fine-tuned during operation ([Åström 89, Barto 89]). The underlying assumption for (adaptive) control to be applicable is that a model of the plant is known a priori.

Control theory is thus suited for tasks like maintaining speed, acceleration, orientation, or force, but is badly suited for abstract task definitions such as "don't get stuck", or for applications where models of the plant are not known:

> "Optimal control methods apply if there are models of the plant and performance measures that are sufficiently accurate and tractable. However, in less structured situations it may be possible to improve plant performance over time by means of on-line learning methods performing what we call reinforcement learning" ([Barto 89]).

This is precisely what we propose to do to achieve intelligent control of mobile robots.

1.2 Artificial Intelligence in Control

Whilst the objective of control theory in robot control was to keep certain parameters within specified bounds, early Artificial Intelligence (AI) work in the area of mobile robot control concentrated on the application of AI planning techniques to robot control. Typically, the task was to select an appropriate action from a set of available actions, in order to arrive at a goal state specified by the user. "Means Ends Analysis" ([Fikes *et al.* 72]) was one method, used in controlling "Shakey", one of the first mobile robots programmed using AI techniques. In the "right" type of environment it allowed Shakey to select an action that led towards the desired goal state — in the "wrong" type of environment, however, the actions would repeat in loops.

A typical "AI"-type controller uses a *model* of the real world as its core component. Perception (i.e. sensory stimulus) directs a planning process centered around the internal world model, and leads to the selection of an action, which is subsequently executed. The *Sense-Think-Act* cycle then repeats (see, for instance, [Albus 91, Meystel 88]).

This "vertical decomposition" of the control task, in which sensor signals enter the controller, percolate through perceptional, modelling, planning and executional modules to initiate motor action has received criticism, and alternative models of controller design have been proposed ([Brooks 85, Malcolm *et al.* 89]. Their dominant features are the absence of internal world models and symbolic representation, minimalist use of computational resources, and tight coupling between sensory input and effector output. Examples for *behaviour-based* robot controllers are the MIT robots ([Brooks 85, Brooks 86, Brooks *et al.* 88]), or the Edinburgh robots ([Malcolm & Smithers 88, Nehmzow 92], [Nehmzow *et al.* 93]); see as well [Arkin 87, Payton 86].

So, on the one hand there are symbol-based planning programs that can determine series of actions leading towards specified goal situations, provided the internal model of environment, agent, and agent-world interaction are adequate and no unforeseen changes have occurred; on the other hand there are behaviour-based controllers that are far less brittle, better able to cope with the noise and variation inherent in the real world, but which, as yet, lack the facility to specify higher level goals. There have been attempts to combine both approaches ([Simmons 94, Arkin & MacKenzie 94]); we believe, however, that goal directed and seemingly intelligent behaviour need not necessarily be defined using symbolic representations, and have therefore concentrated on behaviour-based controllers, using as little symbolic representation as possible, no symbolic world model, and a tight coupling between sensors and effectors instead.

1.3 Artificial Neural Networks in Control

Artificial neural networks (also often, and more appropriately, referred to as connectionist computing) receive their name because of the large number of independent computational units ("artificial neurons") working in parallel to associate incoming data with network output. They have the ability to learn, either in a supervised or in an unsupervised way, and are able to extract salient features in the input data ("generalisation"). Because of their high degree of parallelism, they are less sensitive to the failure of individual units than serial computing architectures ("graceful degradation"). Hertz, Krogh and Palmer provide an excellent introduction to the field ([Hertz *et al.* 91]).

Within the area of traditional control, artificial neural networks have mostly been used for system identification and parameter identification ([Barto 89]). Given the underlying model of plant (agent) and environment, they have therefore been used to identify the characteristic parameters of the model.

If, however, underlying models are unknown, *reinforcement learning techniques* can be applied to determine an adequate mapping between sensor signals and actuator response. The term reinforcement learning applies to all learning techniques that improve plant performance by learning through trial and error, using performance feedback only (i.e. feedback evaluating behaviour, but not indicating correct behaviour ([Sutton 91]).

Reinforcement learning techniques have been used in robot control, for instance to acquire a box-pushing behaviour (Mahadevan and Connell's work using Q-learning and statistical clustering ([Mahadevan & Connell 91])), or phototaxis ([Kaelbling 92]). Millan uses connectionist reinforcement learning to associate sonar, infrared and odometry readings of a Nomad 200 robot with motor actions which take the robot towards a specified goal location ([Millan 94]).

Besides having the ability to learn, artificial neural networks allow minimalist, distributed representation, and exhibit so-called "graceful degradation" when individual computing units fail — properties that are highly desirable in robot control. Experiments on leg coordination in walking robots show that even without using the learning capabilities artificial neural networks perform relevant functions in robots reliably ([Cruse *et al.* 93, Chiel *et al.* 92]).

2 Controller Design Principles

In developing a controller for mobile robots that enables them to exhibit and maintain task-achieving behaviour in the real world, we have followed these design principles:

2.1 Experiment, not Simulation!

Computer-based simulation can only simulate what is known already. A model of the agent to be simulated, incorporating the agent's essential characteristics as well as the characteristic properties underlying agent-environment interaction, has to be known to allow faithful simulation. In our opinion, robot-environment interaction is as yet impossible to simulate faithfully. Contrary to some researchers' opinion (for example Kaelbling, [Kaelbling 92]) we believe that conducting experiments with robots can even be *easier* than using computer simulations: the risk of missing relevant aspects of robot-environment interaction is eliminated when actually observing that interaction, whilst it is considerable if the control strategy investigated is based on (possibly unwarranted) assumptions. Furthermore, the chances of "accidentally" finding unexpected solutions while experimenting are far greater when conducting experiments with real robots, as can be seen from the dead-end example described later in this paper (section 4.2). Finally, using computer simulations always bears the risk of "cheating", not in the sense of deliberate deception, but in the sense of accidental omission ([Brooks 85]).

2.2 Limit Predefinition

Inevitably, as designers we are influenced by *our* perception of the world. This is not necessarily the best starting point for specifying a robot's reaction to its sensory stimuli (which are of a totally different nature to ours). In order to overcome this fundamental problem, we aim to use as little predefinition as possible in controller design, instead, we use *self-organising* mechanisms within the controller that allow the robot to develop its own interpretation of sensor signals and its response to them.

2.3 Rapid Acquisition of Fundamental Competences

In order to facilitate flexible responses to changes in the environment, the task or the robot itself (e.g. sensor failure), we use *learning* robot controllers. Their task is to acquire sensor-motor couplings for the fundamental robot actions, such as obstacle avoidance, contour following or landmark-directed motion. Because these competences are *fundamental* to the robot's operation, they have to be acquired very rapidly — our robots acquire them in a couple of learning steps, taking under a minute in real time.

3 The Competence-Acquiring Robot Controller

The goal of the experiments described in this paper was to achieve flexibility in unforeseen situations in mobile robot control. This, we believe, requires the robot's ability to learn from experience. We therefore chose to build a controller with such a learning capability, so that the robot could determine for itself an effective wiring between sensors and actuators. These associations between sensors (input stimuli) and actuators (responses) are stored in an artificial neural network, a Pattern Associator (see below).

Figure 1 shows the general structure of the controller used in all experiments discussed in this paper. The controller consists of fixed and plastic components, fixed components being the instinct-rules, the robot morphology and various parameters within the controller; the plastic component is the Pattern Associator.

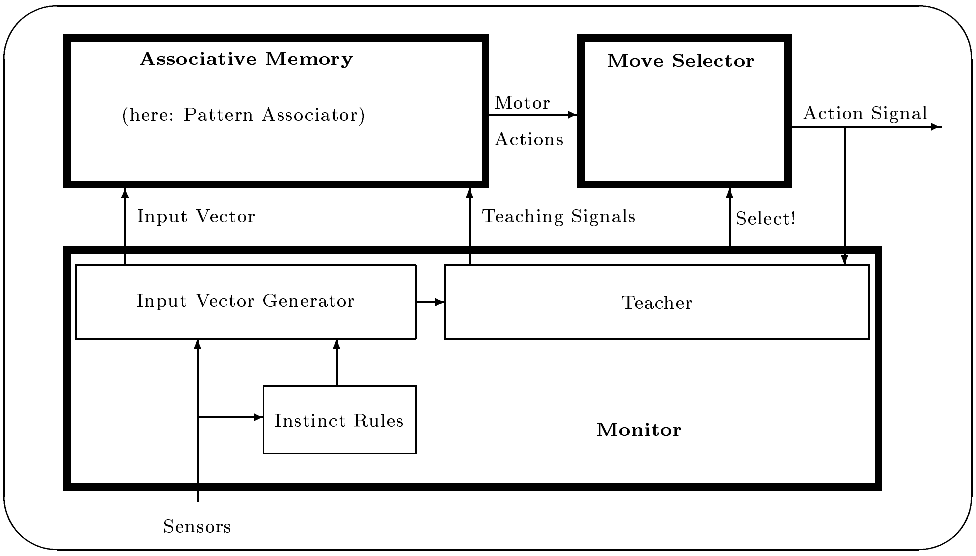

Figure 1: Computational structure of the self-organising controller.

3.1 Pattern Associator

The Pattern Associator ([Kohonen 84, McClelland & Rumelhart 88]) is an artificial neural network that is easy to implement, low in computational cost and fast in learning. It consists of two layers of computational units: the input and the output layer (see figure 2).

The fact that it learns very quickly is bought at the price that it can only learn functions that are linearly separable ([Minsky& Papert 88]). In our experiments, this limitation has *not* adversely affected the robots' performance: all the robot competences reported here can be mapped onto linearly separable functions.

The function of input and output units is as follows: input units simply pass the received input signals $\vec{i}$ on to all output nodes, the output o_j of output unit j is determined by

$$o_j = \sum_{k=1}^{M} w_{jk} i_k = \vec{w_j} \cdot \vec{i} \tag{1}$$

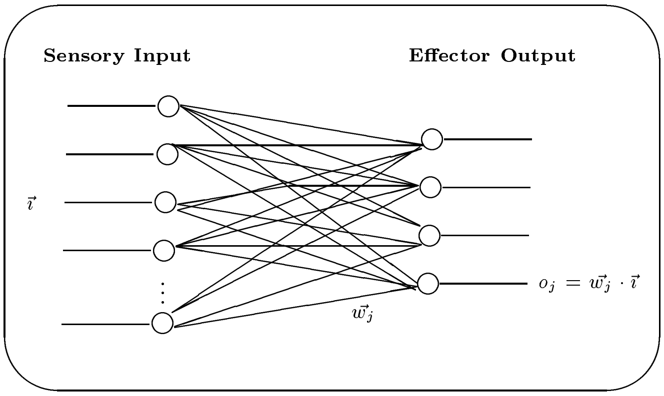

Figure 2: Association between sensor stimuli and motor actions.

where $\vec{w_j}$ is the individual weight vector of output unit j and M the number of input units. If the desired output, the target vector $\vec{\tau}$, is known, updating the weight vectors is done according to the following rule:

$$\Delta \vec{w_j} = \eta(\tau_j - o_j)\vec{\imath} \tag{2}$$

$$\vec{w_j}(t+1) = \vec{w_j}(t) + \Delta \vec{w_j}. \tag{3}$$

η is the so-called gain or learning rate, a parameter that determines how big the changes to the weights are, and therefore how quickly the net learns. A big η (for example 0.8) will result in a network that adjusts very quickly to changes, but which will also be "neurotic" (it will forget all it has learned and learn something new as soon as a couple of freak signals occur). A small η (for example 0.1), on the other hand, will result in a "lethargic" network that takes a long time before it learns a function.

How the target vector $\vec{\tau}$ was determined in the experiments described in this paper is explained in the following section.

3.2 Instinct-Rules

As the Pattern Associator is trained by 'supervised learning', a means of evaluating the robot's behaviour and determining teaching signals for the Pattern Associator had to be found. We use fixed rules for this purpose which we call *instinct-rules*. They are similar, but not identical to *instincts* as defined by [Webster 81]: here, an instinct is defined as a "complex and specific response on the part of an organism to environmental stimuli that is largely hereditary and unalterable though the pattern through which it is expressed may be modified by learning, that does not involve reason, and that has as its goal the removal of a somatic tension or excitation". This describes behaviour and is therefore different to the instinct-rules used in the experiments described here, as instinct-rules are not behaviour, but constants (sensor states) that guide the learning of behaviour. The goal of instinct and instinct-rules, however, is the same: the removal of a somatic tension (in the case of the robot such a 'somatic tension' is an external sensor stimulus, or, in some experiments, the lack of it).

Each instinct-rule has a dedicated sensor in order that it can be established whether it is violated or not. This sensor can be a physical, external sensor (for example a whisker), or an internal sensor (for example a clock that is reset every time some external sensor stimulus is received).

3.3 Input and Output

Current and previous sensor signals constitute the input signals to the Pattern Associator. Information about violated instinct-rules could similarly be used; but this has not been done in the experiments described in this paper.

The output of the network denotes motor actions of the robot. Examples for such motor actions are swift left turn (i.e. right motor moving forward while left motor is moving backward — in other words a left turn on the spot), swift right turn, forward and backward movement. In later experiments ([Nehmzow 94]) such "compound" motor actions are no longer assigned to the output nodes of the network. Instead, just two output nodes drive one motor each of a differential drive system, thus behaving like artificial motor neurons.

The idea behind this controller setup is that effective associations between sensor signals and motor actions arise over time through the robot's interaction with the environment.

3.4 Mechanism

As said earlier, the Pattern Associator requires a teaching signal to develop meaningful associations between its input and its output. This teaching signal is provided by the *monitor*, a "critic" that uses the instinct rules to assess the robot's performance and teach the network accordingly: as soon as any of the instinct-rules become violated (i.e. as soon as a specified sensor status is no longer maintained), an input signal is generated by the *input vector generator*, sent to the associative memory and the output of the network is computed (fig. 1). The *move selector* determines which output node carries the highest output value and which motor action this output node stands for. That motor action is then performed for a fixed period of time (the actual length of time depends on the speed of the robot). If the violated instinct-rule becomes satisfied within this period of time, the association between original input signal and output signal within the Pattern Associator is taken to be correct and is confirmed to the network (this is done by the monitor). If, on the other hand, the instinct-rule remains violated, a signal is given from the monitor to the move selector to activate the motor action that is associated with the second strongest output node. This action is then performed for a slightly longer period of time than the first one to compensate the action taken earlier; if this motor action leads to satisfaction of the violated instinct-rule the network will be taught to associate the initial sensor state with this type of motor action; if not, the move selector will activate the next strongest output node. This process continues until a successful move is found. Because the monitor is part of the robot controller, the process of sensor-motor competence acquisition is completely independent from an operator's supervision: the robot acquires its competences autonomously.

Figure 2 shows the general structure of the Pattern Associator used. The actual input to the network may vary from experiment to experiment, the output nodes denote motor actions.

The result of this process is that effective associations between input stimuli and output signals (motor actions) arise. The instinct-rules and sensors we used are described in more detail in [Nehmzow 92], the following sections of this paper briefly describe the experiments and contain references to the relevant literature.

4 Experiments

The controller described in general in section 3 has been implemented on a number of different mobile robots by different research groups. The consistent finding is that it enables mobile robots to acquire fundamental sensor-motor couplings very rapidly, using no more than twenty learning steps (and typically far less than that), taking a few tens of seconds in real time. Due to the ability to autonomously re-map sensory input to actuator output the robots have the ability to maintain task-achieving competences even when changes in the world, the task, or the robot itself occur.

4.1 Robots

The first experiments were conducted with two custom-made, simple mobile robots, *Alder* and *Cairngorm* (see figure 3), equipped with tactile sensors only, and using low-power industrial controllers, such as the ARC 52 (based on the Intel 8052 microprocessor) or the Flight 68k (based on the Motorola 68000 microprocessor). See [Nehmzow *et al.* 89, Nehmzow 92] for more details.

They were followed by experiments with an *R2* mobile robot (see figure 4), a commercially available robot built predominantly for research purposes. This robot is equipped with eight infrared range-finding sensors, as well as tactile sensor pads and a colour sensor in its gripper. In addition to this, we built

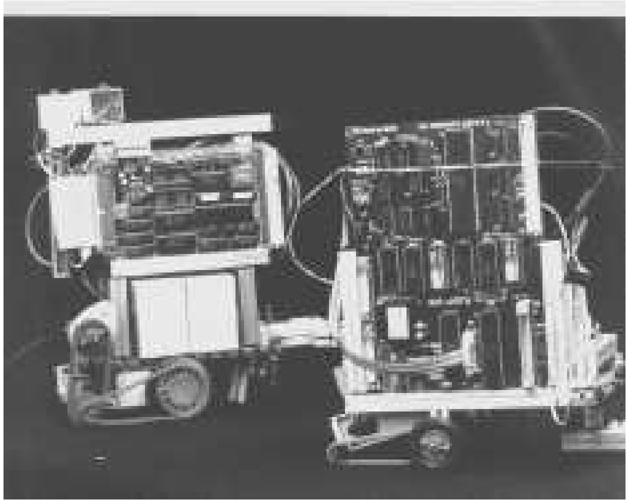

Figure 3: *Alder* (left) and *Cairngorm*

light-sensitive sensors for the robot, which could be used in tasks that involved orientation and direction-finding ([Nehmzow & McGonigle 93]).

At Manchester, a *Nomad 200* mobile robot is used for our experiments (see figure 5). *FortyTwo*, the Manchester robot, uses sixteen sonar sensors for long-range distance measurement; sixteen infrared sensors for short-range measurement. The robot furthermore has a visual sensor (CCD camera) and twenty bumper sensors, both of which were not used in experiments reported here.

4.2 Obstacle Avoidance

Being the most fundamental competence of a mobile robot, all our robots acquired the ability to avoid obstacles. The learning process is very fast: in the case of Alder and Cairngorm (see section 4.1) it took two learning steps, one for associating an obstacle on the left hand side with a right turn, and one for respective association for obstacles on the right hand side of the robot ([Nehmzow *et al.* 89]). The R2 acquired the competence equally quickly, using its infrared sensors, rather than tactile sensors as had been the case with Alder and Cairngorm. FortyTwo, the Manchester robot, associates readings from its sonar range-finding sensors with direct excitations of two artificial motor neurons, and learns to avoid obstacles in a smooth and continuous motion, again taking less than ten learning steps in less than a minute ([Nehmzow 94]). The instinct rules used were *Keep whiskers "off"* (in Alder's and Cairngorm's case) and *Keep sensor readings above a minimal distance* (R2 and FortyTwo).

Using our controller architecture on a Lego robot equipped with two tactile

Figure 4: The R2 mobile robot.

and two infrared sensors, Daskalakis conducted similar experiments on the acquisition of an obstacle avoidance behaviour ([Daskalakis 91]). His robot learned to avoid obstacles after six learning steps, three for avoiding obstacles on the left hand side of the robot, and another three for obstacles on the right hand side. Again using the same controller architecture, Ramakers at the VUB in Brussels used the mobile robot *Lola2* to investigate in particular the influence of the learning rate on the speed of learning ([Ramakers 93]). Using the infrared sensors of Lola2 as input signals, the robot acquired an obstacle avoidance behaviour in between two and five learning steps per side, depending on the learning rate η (see equation 3 in section 3.1).

We stated earlier that the robot's ability to cope with unforeseen situations increases through its learning capability. Two experiments we conducted with Alder and Cairngorm ([Nehmzow 92]) highlight this. In the first experiment, the robots were placed in an environment containing convex obstacles, and quickly learned to turn away from a touched whisker. When encountering a dead end, the robots performed poorly to begin with (by turning back into the dead end, rather than towards the exit), but soon (within two minutes) found the exit and had by then acquired a new behaviour: they were now turning in one and the same direction, regardless of whether the left or the right whisker was touched. This behaviour is best suited for leaving dead ends: the robots had adapted to their new environment.

In the second experiment, the robots' whiskers were physically swapped

Figure 5: The Manchester Mobile Robot, *FortyTwo*

after they had acquired the obstacle avoidance competence. In between four and six learning steps they adapted the associations between sensor signals and motor actions within the artificial neural network, regained the ability to avoid obstacles and thus "repaired" the externally induced fault[2].

4.3 Contour Following

By extending the set of instinct rules, the robots can acquire the ability to stay close to contours such as walls: in Alder's and Cairngorm's case a second

[2]On the R2 we found on one occasion that the robot had been used for weeks, successfully learning many tasks, whilst inadvertently one of the eight infrared sensors had not been working due to a loose connection.

instinct-rule was added (*Do touch something every four seconds*, [Nehmzow 92]), the second instinct rule of the R2 specified a range bin within which the robot was to stay.

Daskalakis' experiments with a Lego robot confirm our findings: his robot acquired the ability to stay close to walls equally quickly ([Daskalakis 91]).

As in the case of obstacle avoidance, we conducted experiments to demonstrate how successful the robots are in coping with unforeseen situations. If Alder or Cairngorm are turned by 180^0 after having successfully learned to follow a wall on, say, their right hand side, they re-map sensors input to motor action within three to four learning steps such that they are able to now follow a wall on their left hand side. If the robots' direction of motion is changed again, the re-learning process is even faster, because latent associations, acquired earlier, are still present within the network and only need to be strengthened to become active again.

4.4 Phototaxis and Box Pushing

In order to show how the behavioural repertoire of a robot can be expanded to more complicated tasks, changing the instinct-rules only, we conducted experiments with an *R2* mobile robot (see section 4.1).

Phototaxis was acquired within less than ten learning steps, using an instinct-rule stipulating that the light sensors mounted at the front of the robot must return the highest value (i.e. face the brightest light).

Similarly, by using a instinct rule requiring the front facing infrared sensors to be "on" constantly (i.e. to return sensor values that indicate an object was situated in front of the robot), the robot acquired a box-pushing or object following competence in two learning steps, one for each of the two situations where the box is placed to the right or the left of the robot's centre respectively. The learning time to acquire a box-pushing competence is less than one minute.

4.5 Learning by External Tuition

In order to establish the complexity of sensor-motor couplings the robot could theoretically acquire, we replaced the feedback given through the instinct-rules by one given by the experimenter (i.e. external feedback). This means that correct and complex feedback is available to the learning process.

Besides the basic competences mentioned above, we could show in experiments with an R2 robot that skills such as phototaxis *and* obstacle avoidance, phototaxis *and* obstacle avoidance *and* obstacle circumnavigation (i.e. moving round an obstacle that is blocking the path to the light source) and even routes through mazes can be learned ([Nehmzow & McGonigle 94]). Equally, the robot was able to acquire an *area covering competence*, i.e. the ability to shuttle between facing walls of a corridor — a competence relevant for cleaning tasks.

These experiments have been replicated on FortyTwo: using sonar and infrared sensor signals as input, the robot has successfully been taught, through external feedback, to avoid obstacles, follow walls, push boxes, or follow people.

5 Conclusions

We have replaced the traditionally fixed coupling between sensor signals and motor actions in the feedforward path of a robot control loop by an artificial neural network. This makes it possible for the robot to determine the wiring between sensors and actuators for itself, i.e. the forward path of the control loop is modifiable through (reinforcement) learning. As a result of this, our robots have, on the one hand, become able to acquire fundamental sensor-motor skills autonomously. On the other hand, they have become able to cope with unforeseen situations, such as changes in the environment (convex/concave obstacles), sensor failure, or even more drastic changes such as dislocated sensors. It may seem far-fetched to expect robot sensors to get dislocated in the course of operation, but the wiring of sensors can be error prone if a large number of sensors is to be connected. If the robot can determine the best wiring autonomously, malfunctions due to wrong sensor connections can be avoided. [Maes & Brooks 90]:

> "For more complicated robots prewiring solutions become either too difficult or impractical. ... Additionally, it is often too difficult for the programmer to fully grasp the peculiarities of the task and the environment, so as to be able to specify what will make the robot successfully achieve the task."

Becoming less dependent on the designer's interpretation of the robot's sensor signals, being able to acquire competences rapidly, using feedback received directly through the robot's interaction with its environment, and to alter behaviour equally rapidly in the presence of noise, variation and change inherent in the real world — these are the major advantages of the self-organising controller presented here. Its strengths are in the acquisition of fundamental sensor-motor competences — representation of abstract, "symbolic" goals is not possible (and not intended). It is conceivable that the controller discussed here could be supplemented with a "symbolic" layer. However, we believe that it is equally possible to achieve higher-level intelligent robot behaviour through the interaction of independent behavioural modules, overall behaviour being emergent through this interaction (see, for instance, [Steels 91, Deneubourg *et al.* 91]). This, we believe, is a sensible next step from here.

Acknowledgements

The work with Alder and Cairngorm was conducted at the Department of Artificial Intelligence at the University of Edinburgh, the experiments with the R2 at the Department of Psychology at Edinburgh. The experiments with *FortyTwo* were conducted at Manchester University; the author thanks all institutions for their support.

References

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

[Albus 91] J. Albus, *Outline for a theory of intelligence*, IEEE Trans. Systems, Man and Cybernetics, Vol.21 No.3.

[Arkin 87] R. Arkin, *Motor Schema based navigation for a mobile robot: an approach to programming by behaviour*, Proc. Conf. Robotics and Automation, 1987.

[Arkin & MacKenzie 94] R. Arkin and D. MacKenzie, *Temporal coordination of perceptual algorithms for mobile robot navigation*, IEEE Trans. Robotics and Automation, Vol.10, No.3, 1994.

[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.

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

[Brooks 86] Rodney Brooks, *Achieving Artificial Intelligence through Building Robots*, MIT AI Memo 899, May 1986.

[Brooks *et al.* 88] Rodney Brooks, Jonathan Connell and Peter Ning, *Herbert: A Second Generation Mobile Robot*, MIT AI Memo 1016. January 1988.

[Chiel *et al.* 92] H. Chiel, R. Beer, R. Quinn and K. Espenschied, *Robustness of a Distributed Neural Network Controller for Locomotion in a Hexapod Robot*, IEEE Robotics and Automation, Vol.8, No.3, 1992.

[Cruse *et al.* 93] H. Cruse, U. Müller-Wilm and J. Dean, *Artificial Neural Net for Controlling a Six-Legged Walking System*, in J.A.Meyer, H. Roitblat and S. Wilson (eds), *From Animals to Animats 2*, MIT Press 1993.

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

[Deneubourg *et al.* 91] J.L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain and L. Chretien, *The Dynamics of Collective Sorting. Robot-like Ants and Ant-like Robots*, in J.A. Meyer and S. Wilson (eds), *From Animals to Animats*, MIT Press, 1991.

[Fikes *et al.* 72] R.E. Fikes, P.E. Hart and N.J. Nilsson, Learning and Executing Generalized Robot Plans, AI, 3 (1972), pp. 251 - 288.

[Hertz *et al.* 91] J. Hertz, A. Krogh and R. Palmer, *Introduction to the theory of neural computation*, Addison Wesley 1991.

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

[Kohonen 84] T. Kohonen, *Self-Organization and Associative Memory*, Springer Verlag, 1984.

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

[Malcolm & Smithers 88] Chris Malcolm and Tim Smithers, *Programming Assembly Robots in Terms of Task Achieving Behavioural Modules: First Experimental Results*, Proc. Intern. Adv. Robotics Programme, Second Workshop on Manipulators, Sensors and Steps towards Mobility, Manchester 1988. Also published as DAI RP 410, Department of Artificial Intelligence, Edinburgh University.

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

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

[Meystel 88] A. Meystel, *Intelligent control in robotics*, J. of Robotics Systems, 1988.

[Millan 94] J. Millan, *Learning Efficient Reactive Behavioural Sequences from Basic Reflexes in a Goal-Directed Autonomous Robot*, in D. Cliff, P. Husbands, J.A. Meyer and S. Wilson (eds), *From Animals to Animats 3*, MIT Press 1994.

[Minsky& Papert 88] Marvin Minsky and Seymour Papert, Perceptrons, MIT Press Cambridge Mass. and London, England, 1988.

[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, PhD thesis, University of Edinburgh 1992.

[Nehmzow *et al.* 93] U. Nehmzow, T. Smithers and B. McGonigle, *Increasing Behavioural Repertoire in a Mobile Robot*, in J.A. Meyer, H. Roitblat and S. Wilson, *From Animals to Animats 2*, MIT Press 1993.

[Nehmzow & McGonigle 93] Ulrich Nehmzow and Brendan McGonigle, *Robot Navigation by Light*, European Conference on Artificial Life, Brussels 1993.

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

[Nehmzow & McGonigle 94] U. Nehmzow and B. 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.

[Payton 86] D. Payton, *An architecture for reflexive autonomous vehicle control*, Proc. Conf. Robotics and Automation, 1986.

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

[Simmons 94] R. Simmons, *Structured control for autonomous robots*, IEEE Trans. Robotics and Automation, Vol.10, No.1, 1994.

[Steels 91] L. Steels, *Toward a Theory of Emergent Functionality*, in J.A. Meyer and S. Wilson (eds), *From Animals to Animats*, MIT Press, 1991.

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

[Webster 81] *Webster's Third New International Dictionary*, Encyclopaedia Britannica Inc., Chicago 1981.