Staged Learning for a Continually Operating Robot

Gary R. Ireland and Ulrich Nehmzow
Department of Computer Science
University of Manchester
Manchester M13 9PL
irelandg@cs.man.ac.uk, ulrich@cs.man.ac.uk

5.4.2001

Abstract

There are applications for autonomous mobile robots in which continuous, unsupervised operation over long periods of time is desirable. Examples are surveillance, monitoring and cleaning tasks. Since real-world environments are typically changeable, and since a modification of such environments to ease robot applications is costly, control systems requiring pre-installed knowledge are disadvantageous. Rather, control systems with the ability to learn are preferable. *Continuous* operation over long periods of time poses particular learning and operation problems, which are addressed in this paper.

This paper presents a learning procedure of staged competence acquisition, where a complex functionality is decomposed into simpler competences, and acquired in stages at appropriate times during robot operation. To control this staging, each competence has a unique sensitisation function, based upon the operational time of the robot.

In our experiments, the staged-learning procedure was applied to a self-recharging robot which was required to wander and recharge when its batteries were low. Experimental results indicate that the proposed approach is suitable for controlling robots operating continuously over long periods of time.

1 Motivation

There are applications for autonomous mobile robots in which continuous, unsupervised operation over long periods of time (several days without interruption) is desirable, for instance surveillance, monitoring or cleaning tasks.

For reasons of cost and efficiency it is not desirable to have to modify an environment to allow robot operation. Nor is it desirable to use pre-installed knowledge, where this can be avoided, for the following reasons:

1. Perceptual discrepancy. Robot sensors do not necessarily respond well to the same environmental features a human sees well. A human operator may not be supplying the robot with the most useful data.
2. Inflexibility. Pre-installed competences may not be applicable to situations which were not foreseen.
3. Temporal invariance. Non-learning behaviours may not cope well with environmental changes over time.
4. Spatially invariance. Pre-installed knowledge often enables the robot to operate only in known environments.

Proc. TIMR 01 - Towards Intelligent Mobile Robots, Manchester 2001.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report number UMCS-01-4-1. http://www.cs.man.ac.uk/cstechrep/titles01.html

5. Brittleness. Inaccurate modelling can produce incorrect operation.

6. Cost. Pre-installing models is time-consuming and costly.

7. Technical complexity. Modelling typically requires skilled operators.

For these reasons we argue that it is preferable to let the robot *acquire* competences in interaction with its environment. Learning will allow the robot to exploit salient features of the environment, minimising the need for modifications to the environment, and to adapt to changes in the environment.

Machine learning has regularly been applied to robot control (see [6] for a survey), but, to our knowledge, *always* in "condensed" learning scenarios. By "condensed" learning scenarios we mean learning situations in which almost all stimuli presented to the robot are relevant with respect to the competence being learned. Such condensed learning scenarios are suitable for acquiring competences under controlled conditions, but are likely to fail in real-world scenarios of *continuous* long-term operation, where many perceived stimuli are "irrelevant".

We believe that staged competence acquisition is an approach which allows the robot to overcome these problems by controlling *when* a competence is learned, and at what rate (see [7] for a further discussion).

2 Approach: Staged Competence Acquisition

2.1 Definition

We refer to "staged competence acquisition" as a learning process in which the overall competence or behaviour of a robot is decomposed into smaller sub-competences, which are acquired in turn. Starting to learn very simple competences, the robot uses these to acquire more complicated competences later in its operation, and so on. Some inspiration for this is drawn from biological systems (e.g. [11]). Examples are hand-eye coordination which is the basis of all visuomotor competences, or the ability to balance on two legs, which is the basis for the ability to walk.

2.2 Experimental Scenario

We decided to investigate this approach using an autonomous mobile robot, whose task it was to explore its environment in ever-widening movements, and to re-connect to a charging station from ever-increasing distances. We used a modified Nomad Scout robot, which was equipped with charging contacts that allowed autonomous connection to a wall-mounted charging station.

The robot employed three behaviours, two of which were learnt through staged competence acquisition:

1. Connect to charger. This was a reflex-like fixed behaviour.

2. Short range navigation back to the immediate proximity of the charger. This was a learned behaviour that determined direction and distance to the charger from distances up to 50 cm from the charger.

3. Middle range navigation back to the area where short range navigation is able to direct the robot to the charger. This was a learned behaviour that determined direction and distance to the charger from distances up to 3 m from the charger.

This experimental setup is shown in figure 1.

2.3 Major Operational Phases

Initially, the robot has no sensor-motor knowledge whatsoever, and is switched on while connected to the charging station. The robot then begins to leave the station and re-enter it in a fan-like pattern of ever increasing size, up to distances of about 50 cm. During this first phase short range navigation is learned. This phase lasts for about two minutes.

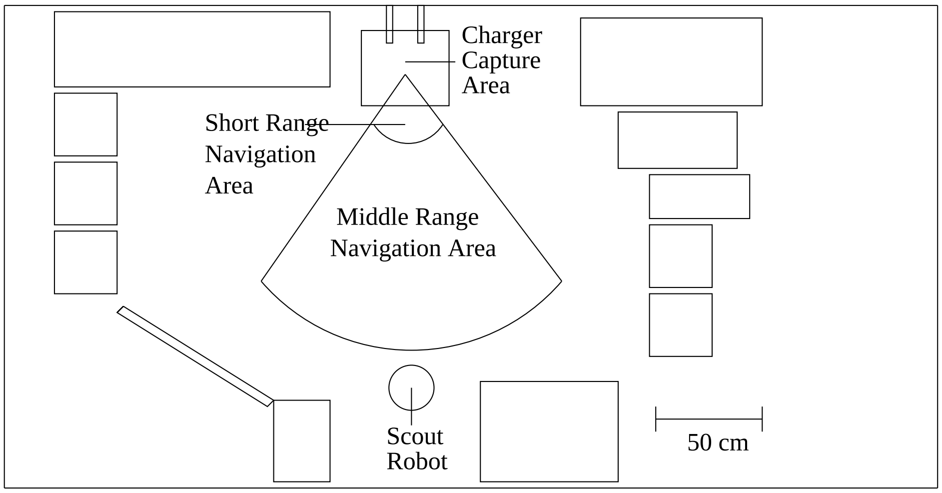

Figure 1: Areas of operation

At the next stage of learning, the robot moves in and out of the charging station at yet larger distances. At this stage, middle scale navigation is acquired. This phase lasts about twenty minutes. Both learning phases are shown in figure 2.

At the end of these two learning phases, the robot assumes a random walk behaviour until its battery level drops below a pre-defined threshold, whereupon middle range and short range navigation are used to re-connect to the charger.

2.4 The Competences in Detail

2.4.1 Principle of Short Range Navigation

The short range navigation mechanism implemented on our robot takes inspiration from the honey bee: honeybees undertake orientation flights, at the start of these flights, the bees hover in front of the hive before setting out for long range navigation, memorising the perceptual appearance of the hive ([1]).

The short range navigation competence is functionally quite similar to this hovering phase. It learns to associate environmental features with a vector to the charger. A fan shaped training pattern was used to move the robot during the learning phase (see figure 2). The robot simply backs away from the charger taking sonar readings at certain positions and used neural networks to associate those readings with odometry based distance from, and fluxgate compass based heading to the charger.

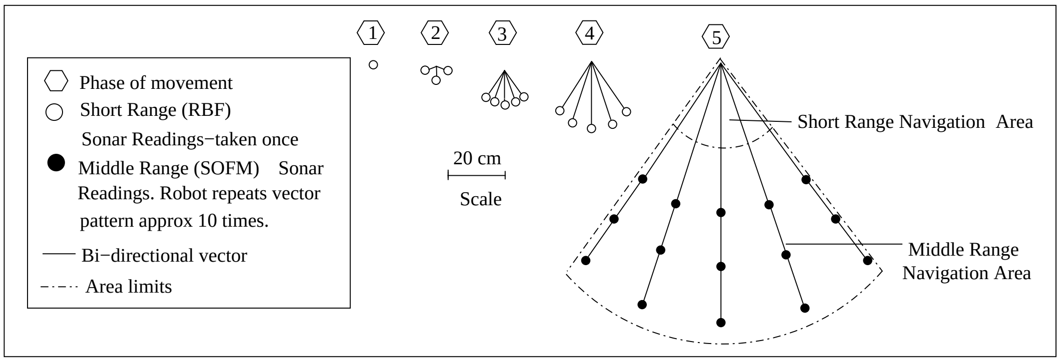

Figure 2: Robot movement while learning short range and middle range navigation

2.4.2 Implementing Short Range Navigation with a Radial Basis Function Network

The Short Range Navigation competence required a network which trained rapidly and could map a continuous function across the area directly in front of the charger.

We use a Radial Basis Function (RBF) Network for this purpose ([9]). The net had sixteen input units (one for each sonar sensor of the robot), 14 units in the radial basis layer, and two output units indicating distance and angle to the charger.

The weights of the hidden layer were determined by simply setting them to be equal to the first 14 sensory perceptions obtained during the initial fan-like movement in and out of the charger. The output units were trained using Hebbian learning.

2.4.3 Implementing Middle Range Navigation with a Self Organising Feature Map

Encouraged by previous experience in implementing middle scale navigation ([5], [10] and [4]), we chose to implement the middle range navigation competence using a self-organising feature map (SOFM).

Using a similar approach as the one described in [10], we used a network of 8x8 nodes that took an input of 18 values (16 sonar values, distance to the charger and direction to the charger). When used for data retrieval, the input vector became the sonar reading, with zeros instead of distance and direction data. The those two values are then read from the weights of the winning node (see [6, pg 121]).

2.5 Integrating the Competences

The learning rate for both the RBF and the SOFM network need to reflect the robot's current operational phase: the RBF learning rate needs to be high early on in operation and then decrease continuously, while the SOFM learning rate needs to be small initially, to increase and then again decrease subsequently. We have implemented these requirements by controlling the learning rate of both networks using sensitisation functions that take the robot's operational time t as its only parameter. These functions are shown in figure 3.

2.5.1 The Sensitisation Function for the Short Range Navigation Competence

When first switched on the robot acquires the short range navigation competence. This requires the learning rate to be near 1.0 at the start of operation, to stay high for the first two minutes and then to fall rapidly below the activation threshold. An exponential function (equation 1) with negative slope was used to produce this, the sensitisation curve λ_r is given in (figure 3) and equation 1.

$$\lambda_r = e^{-t^2/c}. \tag{1}$$

In equation 1, t is the operational time of the robot and c is a constant ($c = 500$). Learning in the RBF was inhibited if λ_r fell below 0.3.

2.5.2 The Activation Function for the Middle Range Navigation Competence

Unlike the RBF, whose learning is completely inhibited below a value of $\lambda_r = 0.3$, the SOFM learning stayed active throughout the entire operation of the robot. The SOFM learning rate λ_s, however, was modulated by equation 2 (see also figure 3).

$$\lambda_s = e^{-(((k-t)^2)/d)} \tag{2}$$

Again, t is the operational time of the robot, k and d are constants ($k = 154$ and d=70) and λ_s is the learning rate for the SOFM.

As can be seen from figure 3, the SOFM learning rate is initially very low. It peaks after about 2 minutes operational time, and then decreases again.

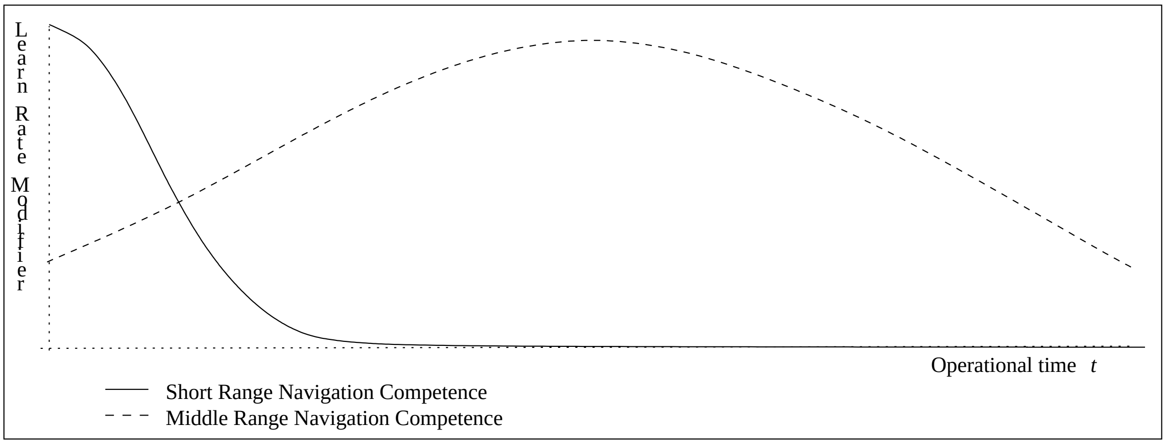

Figure 3: The sensitisation curves of the short range navigation (λ_r) and middle range navigation competences (λ_s) competences

2.5.3 Fusion of the RBF and SOFM Generated Data

When the robot approaches the charger, both the SOFM and RBF will generate homing data. However, due to the position of the robot, both, one or neither of these data sets will prove adequate for homing (see figure 4). In order to arbitrate between the values and decide which are valid, some measure of confidence is required, which we obtained from the networks themselves in the following manner.

In the RBF net, the output of each unit in the RBF layer is proportional to the Euclidean distance of that unit's weight vector and the currently perceived input vector. In other words: if any RBF unit's output is high, it means a familiar input vector is presented to the network (i.e. the "confidence" value is high). We therefore used the integrated activation of *all* RBF units as a confidence value in the network's prediction of distance and direction to the charger.

In the winner-takes-all SOFM net, the confidence value we selected is the Euclidean distance between the winning node and the input vector.

To make both confidence values numerically comparable, the SOFM confidence value is modified with the same operations as those applied to an RBF node. A further complication arises because the RBF value is a summation of the distance of the input vector from all nodes, not simply the nearest as in the SOFM. Analysis of the results of some simple experiments suggested that, in order to make the values comparable, the RBF value should be halved. This modification of both confidence values allows the results to be compared directly.

Several tests were conducted. Fourteen of the confidence values from these tests are displayed in figure 4.

When used during the robot homing, the confidence values generated by the nets are examined. Values below 0.03 are ignored because the homing data will be too inaccurate to enable the robot to home. If only one of the values is above 0.03 its data alone is used. If both RBF and SOFM values are acceptable and one confidence value is a factor of 2 greater, then the data from that net alone is used. Otherwise the data from both nets are used proportionally, see equation 3 and 4.

$$rot_{fused} = \frac{(rot_r \cdot c_r)}{(c_r + c_s)} + \frac{(rot_s \cdot c_s)}{(c_r + c_s)}, \tag{3}$$

with rot_{fused} being the final assumed direction to the charger, rot_r and rot_s being the assumed direction to the charger as indicated by the RBF net and the SOFM alone, and c_r and c_s the confidence values of the RBF net and the SOFM respectively.

$$dist_{fused} = \frac{(dist_r \cdot c_r)}{(c_r + c_s)} + \frac{(dist_s \cdot c_s)}{(c_r + c_s)}, \tag{4}$$

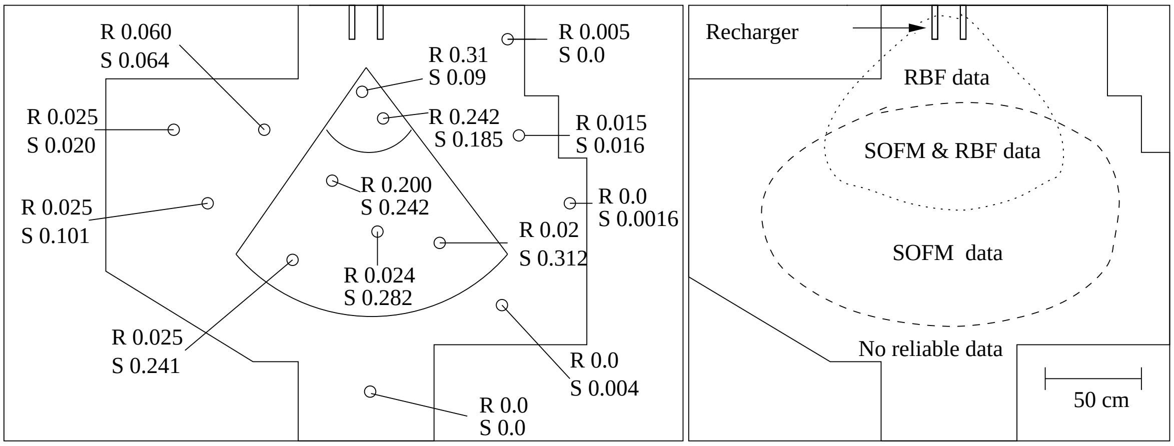

Figure 4: Left:Confidence values at 14 robot positions. "R" indicates confidence values of the RBF net, i.e. the short range navigation competence. "S" indicates confidence values of the SOFM net, i.e. the middle range navigation competence. Right: Approx limits of confidence values >0.03

with $dist_{fused}$ being the final assumed distance to the charger, $dist_r$ and $dist_s$ being the assumed distance to the charger as indicated by the RBF net and the SOFM alone, and c_r and c_s the confidence values of the RBF net and the SOFM respectively.

3 Experiments

3.1 Robot Hardware: The Nomad Scout Robot *Strange*

The Nomad Scout is a mobile robot designed for research in robotics ([8]). It has a sixteen-sided body with a sonar device in each facing. A differential drive mechanism, with built in odometry, provides movement. Basic hardware functions are provided and a serial interface for external high level control.

Modifications to the Nomad Scout robot *Strange* made it particularly suitable for this experiment:

- On Board High Level Controller: The Amubot board designed at Manchester University ([12]) has several advantages for continuously operating robots: low power usage; powerful processor; large memory (0.5 gigabyte); 7 analogue inputs; 16 digital inputs; 4 motor drivers and 8 digital outputs. In addition it mounts a Fluxgate compass, enabling orientation within the external world.
- Self-Recharge Unit: This charging unit allows the robot to recharge its batteries autonomously ([3]). It is shown in figure 5.

Figure 5: left to right: the anode contact, *Strange* contact, cathode contact, two views across the Experimental Area

A cable is attached to a copper plated board, suspending it sloping upwards from a wall hinge. On the wall beneath the board, two sprung cathodes are mounted (figure 5).

Strange was fitted with a cathode, a metal ribbon pinned to a pole on top of the Scout, and an anode, a piece of flexible copper plate attached along the front upper edge of the robot's body (figure 5).

3.2 Experimental Procedure

3.2.1 Navigation Learning and Application of the Acquired Knowledge

The same experimental area was used throughout the experiments (see figures 1 and 5). During the learning phase the robot moved using the fan shaped pattern (figure 2) whilst the sonar generated data for the neural networks. The activation function with output λ_r was used to control the activation and learn rate of short range navigation learning and λ_s was used to control the learn rate of middle range navigation learning.

After learning, the robot wandered randomly within the experimental area, pausing before each change of direction to allow the experimenter to record the robot's positional data. Flicking the Amubot boards miniature joystick allowed the robot to resume random movement.

Because the robot's batteries last tens of hours, we simulated a quicker discharge to be able to observe recharge movements more often. Once this mock battery level was below a predefined limit the robot halted and used the RBF and SOFM to generate confidence and homing vector values, displaying the output and positional data on the LCD screen. These were recorded.

If both confidence values were below 0.03, random wandering continued, repeatedly taking readings until a position was reached where confidence values greater than 0.03 were found. When reliable confidence values were obtained, the robot displayed the generated values and paused for data recording. The robot then turned to the heading and moved 30 percent of the estimated distance to the charging station.

This process was repeated until the RBF generated translation value had dropped below 50 cm, or the robots sonar detected an object closer than 36 cm ahead in the robot's path. The connect-to-charger function then assumed control, overseeing connection to the charger.

3.2.2 Connect-to-Charger Behaviour

The connect-to-charger function moved the robot straight forward 30 cm (though the robot would stop if sensors detected an obstruction at less than 20 cm). If electrical contact was made with the charger a mock recharge was triggered, otherwise recharge had failed. In the case of failure the robot first reversed 27 cm, and then large random distance away, from where it would commence random movement once more.

3.3 Experimental Results

3.3.1 Trial 1

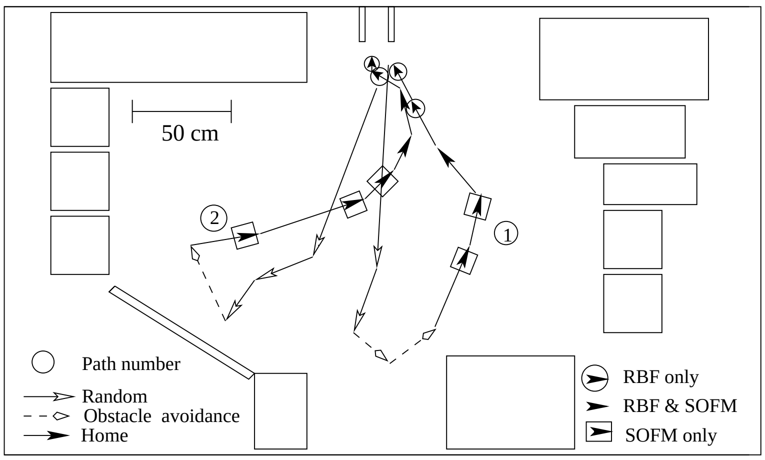

Figure 6: Robot wander and home paths from 2 Separate Trials

During the learning phase of the test, the robot experienced some odometric drift that was severe enough to force it to use emergency braking each time it approached the right side wall. This did not, however, later affect the robot returning to the charger.

On return trip 1 (see figure 6) the first return heading generated by the SOFM was approximately 30 degrees out. The next reading, also SOFM only, continued along this incorrect course. The reading that followed this was based on RBF and SOFM data. The robot assumed a heading which would take it within the short range navigation area. The following two readings were both RBF only and provided the robot with an extremely good heading (path 1 on figure 6). The robot was stopped by the emergency braking function and control was passed over to the connect-to-charger function. The robot moved into contact with the charger and mock recharging successfully commenced.

3.3.2 Trial 2

The learning phase passed uneventfully and the robot wandered until the mock battery level went low. The first two vectors generated were SOFM only and once more they were quite inaccurate, but consistent. The third reading, (also SOFM only) showed some improvement and the remaining readings, mixed and then RBF only, showed good orientation (path 2 on figure 6). The robot successfully moved into contact with the charger, using the connect-to-charger behaviour.

3.3.3 Trial 3

In trial 3 the learning phase was affected by emergency brakes cause by spurious sensor readings, with no ill effect to the robot's ability to connect with the charger later.

In total, five recharge cycles were conducted (see figure 7).The third path failed to make contact with the charger, but the connect-to-charger function correctly determined this and returned the robot to wandering (figure 7).

Only three of the paths had SOFM-only vectors. Path 1 showed very good orientation. Path 4 showed behaviour much like trial 2 and path 5 produced two vectors, the first with an extremely bad heading, approximately 70 degrees out, followed by a much more accurate one. The SOFM network produced erroneous values when the robot was quite far to the left or right of the Recharge Unit.

Path 3, on which contact failed, showed very unusual behaviour. The first reading was an RBF only value of high confidence, even though the robot was far away from the short range navigation area (figure 7). It is therefore very likely that the high confidence values were generated by perceptual aliasing. All the remaining vectors on this path were mixed, but of very low confidence.

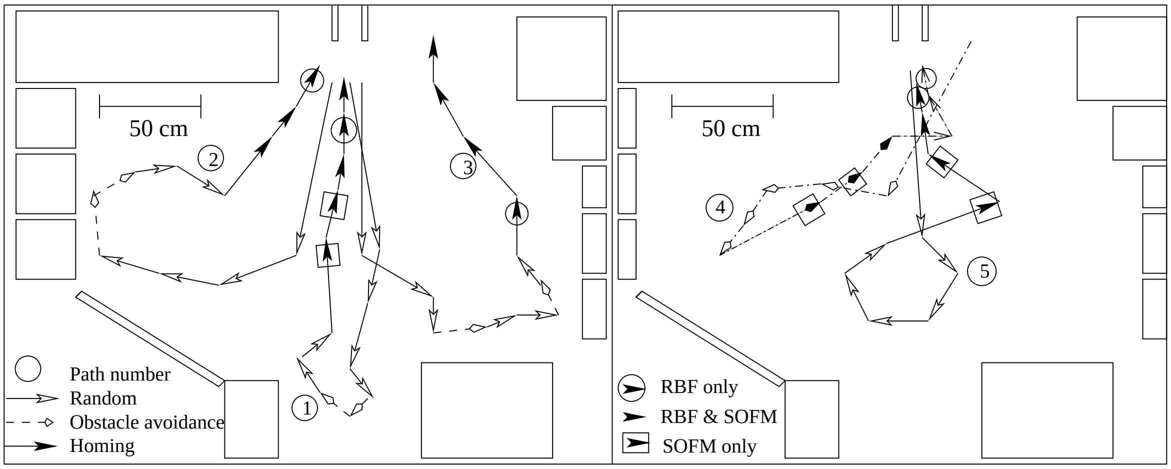

Figure 7: Robot wander and home paths from 5 path continuous trial

3.4 Interpretation

The robot trained correctly and displayed the correct functionality by wandering until the battery was low, returning to the charger, recharging and then resuming wandering. The robot accomplished this 6 out of 7 times, and when the robot did fail to connect (path 3) the robot showed the correct failure behaviour and afterwards connected correctly.

During the wandering phase the Self Organising Feature Map produced less accurate vectors than it had in the separate competence tests. However, because of the fusion of the SOFM output with the RBF the robot still performed satisfactorily. The fusion of the output from the two neural networks showed how a certain amount of redundancy in the system can affect performance positively. The redundancy is simply the overlapping of the SOFM and RBF mapping areas (figure 4), both producing useful results in this area.

The only time that there was an obvious failure of confidence level as predictor of viable homing data was the first SOFM value on path 3 (figure 7). There is no obvious perceptual aliasing at this position, maybe the configuration of the environment at this place created some incorrect sonar reading.

Although over-reliance on odometry can create problems [2], basing movement on odometry during the learning phase presented none. The robot suffered from a small degree of odometric drift during learning, this was only translational and not rotational because the fluxgate compass removed any drift error.

The experiment investigated how a complex functionality can be decomposed into simpler functions and then trained in a staged manner using the operational time of the robot to control staging. This approach produced good results, suggesting that further experimentation with this method is warranted.

4 Conclusion and Future Work

4.1 Summary

In order to expand the use of robots beyond the factory floor there is a need for autonomous robots which can operate continuously in unmodified environments. The staged competence acquisition model, loosely based upon animal life cycles, offers a possible approach, albeit on a very small scale, to provide such a system. Complex functionality can be decomposed into simpler competences which can be learned at different stages in the robot's operation, so that at some point the complex overall functionality is acquired. The operational time of the robot was used as a variable upon which each competence based a tailored activation function. The output of the activation function was used to determine when and at what rate learning occurred.

This experiment investigated how a functionality can be acquired with competences whose staging during learning and utility overlaps. A self recharging robot explored its environment until its batteries were low, return to a charger, recharge and then resume wandering. This was achieved by learning two independent competences, short range and middle range navigation.

4.2 Conclusions

The test runs produced encouraging results. The robot trained correctly and displayed the correct functionality, suggesting that further experimentation with this method is warranted. An unexpected bonus was the demonstration of the positive effect of redundancy and how it can increase system robustness.

The immediate plans for the future are to increase the number of competences, on the one hand obstacle avoidance, on the other long range navigation capabilities.

We believe that continuous robot operation over long periods of time and learning under such conditions poses unique problems that are not apparent under "condensed" learning scenarios. The amount of "irrelevant" data (irrelevant with respect to the particular competence being learned at that point in time) will be high, and the effect of noise and contradictory sensory perceptions will be more pronounced than under condensed learning situations. Therefore, once a robot capable of continuous operation over long periods of time in a large area has been built, it will be used to investigate the problems which arise from continuous operation in detail.

References

[1] E. A. Capaldi, A. D. Smith, J. L. Osborne, S. E. Fahrbach, S. M. Farris, D. R. Reynolds, A. S. Edwards, A. Martin, G. E. Robinson, G. M. Poppy, and J. R. Riley. Ontogeny of orientation flight in the honeybee revealed by harmonic radar. *Nature (letter)*, 403(6769):537–540, 2000.

[2] T. Duckett and U. Nehmzow. Knowing your place in real world environments. In *EUROBOT'99: 3rd European Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1999.

[3] R. Henderson. *A Continuously Operating Mobile Robot*. Manchester University, 1999.

[4] O. Lemon and U. Nehmzow. The scientific status of mobile robotics: Multi-resolution mapbuilding as a case study. *Robotics and Autonomous Systems*, 24(1–2), 1998.

[5] U. Nehmzow. An episodic mapping algorithm for mobile robot self-localisation: "meaning" through self-organisation. In *International workshop on "Computation for Metaphors, Analogy and Agents", Aizu Wakamatsu*, 1998.

[6] U. Nehmzow. *Mobile Robotics: A Practical Introduction*. Springer Verlag, 1999.

[7] U. Nehmzow. Continuous operation and perpetual learning in mobile robots. In *International Workshop on Recent Advances in Mobile Robots*. DeMonfort University Leicester, 2000.

[8] Nomad Technologies. *Nomad Scout User's Manual*. Nomadic Technologies, Inc., 1999.

[9] M. J. L. Orr. *Introduction to Radial Basis Function Networks*. University of Edinburgh, 1996.

[10] C. Owen and U. Nehmzow. Map interpretation in dynamic environments. In *Proceedings of the 8th International Workshop on Advanced Motion Control*. IEEE Press 1998 (ISBN 0-7803-4484-7), 1998.

[11] J. Piaget. *The Origins of Intelligence in Children*. International Universities Press New York, 1952.

[12] S. Rackstraw. *Amulet Based Robot Controller*. Manchester University, 1997.