MOBILE ROBOT SIMULATION BY MEANS OF ACQUIRED NEURAL NETWORK MODELS*

Ten-min Lee, Ulrich Nehmzow and Roger Hubbold
Department of Computer Science,
University of Manchester
Oxford Road, Manchester M13 9PL, U.K.
leet, ulrich, roger@cs.man.ac.uk

Abstract

This paper presents experiments with a Nomad 200 mobile robot, acquiring a sensor model of a specific environment and using this model to predict robot-environment interaction. Data obtained by operating the real robot in the real target environment is used to train a set of 16 artificial neural networks which can later be used to model robot-environment interaction and predict the behaviour of the real robot in off-line simulation. A number of experimental results are presented, demonstrating that this approach can be used to model sensory perception of a mobile robot, as well as to model the behaviour of a specific robot in its target environment.

1 INTRODUCTION

The advantages of numerical modelling of robot-environment interaction are well known and widely appreciated in the literature [2, 3, 4, 11]. Compared with conducting experiments with real robots, simulation is fast, cheap and, perhaps most importantly, facilitates repeated experiments under identical conditions: a property desirable if the influence of certain parameters upon a robot's behaviour is to be determined. However, the disadvantages of numerical models are also known and have so far prevented simulation from having significant impact upon mobile robotics research [2]. Typically, the numerical models of robot-environment interaction that are used for simulation are so simplified that their predictions do not match experimental observations well [5, 7, 12, 13], so much so that the outcome of a simulation can only be used for prediction of a real robot's behaviour in the simplest of cases.

In this paper, we present experiments in which the interaction of a Nomad 200 mobile robot with a target environment is used to train a neural network model of that robot's interaction with that environment. In our approach, instead of storing the interaction as a look-up table [10], we are using a neural network training method to generalise the relation between robot's environmental perceptions and its locations [9]. Compared with using look-up tables, a neural network approach does not require curve-fitting techniques to generate data between samples. This neural network model is later used for simulation of the robot's behaviour in the target environment.

In an initial training phase, real data collected from the target environment is used to train a set of 16 different artificial neural networks to compose the robot's sensory perceptions of its environment. Each network models the readings obtained from one particular sonar sensor. Once acquired, these 16 models of the 16 sonar sensors of the Nomad robot are used to predict perceptions at arbitrary places within the robot's environment and, furthermore, to predict the robot's corresponding behaviour.

Figure 1: The Manchester *Forty-Two*. The robot's sonar and infrared sensors are mounted on the turret. In these experiments, the turret was kept at a constant orientation to provide a "compass sense". Exploration was carried out using the separate translational and rotational motors located in the base of the robot.

2 MODELLING THE ROBOT'S ENVIRONMENT

2.1 Structure of the Model

A Nomad 200 mobile robot (see figure 1) has, amongst other sensors, 16 sonar sensors which are used for the experiments presented here. These sensors are evenly spaced along the perimeter of the hexagonal robot and obtain range readings between 50 cm and 6.5 metres. Owing to the physical nature of sonar sensors these sensors are subject to specular reflection and cross-talk which makes the prediction of sensory perceptions particularly difficult using simplified numerical models. The purpose of our experiments was to model the 16 sonar sensors by means of real data obtained from the Nomad interacting with a target environment. To achieve this, 16 multi-layer perceptrons were used: one to model each sonar sensor. In neural network theory, it is well known that a two-hidden-layer structure can learn arbitrary curves [1, 8]. The total number of hidden units we have tried ranges from 10 to 200. As a compromise between learning time and accuracy, each network has 50 hidden units in two layers, two input units and one output unit (see figure 2). The two input units encode the robot's current position

*12th European Simulation Multiconference, Manchester 1998.

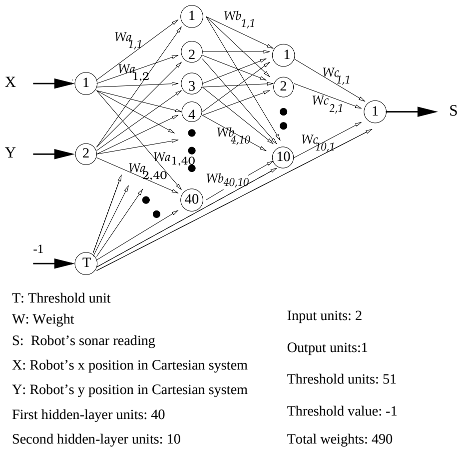

Figure 2: The two-hidden-layer neural network used to simulate *Forty-Two's* sonar sensors.

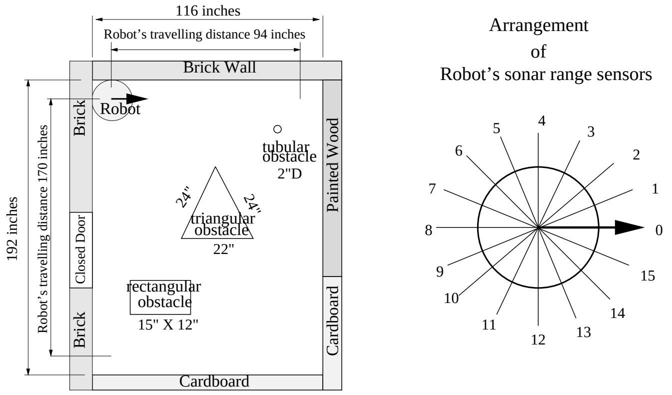

Figure 3: A rectangular room with three different-shaped obstacles (*Left*). The arrangement of robot's sonar range sensors (*Right*).

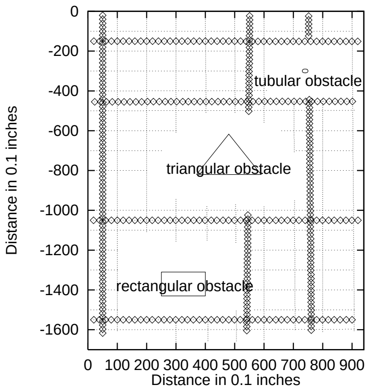

Figure 4: The dotted line shows the path used to collect training data, the "box" line shows the path for collecting validation data.

in Cartesian space as obtained from the robot's odometry system. The single output unit encodes the range reading obtained from that sonar sensor at that particular position. Each network is trained individually according to a procedure described in section 2.2 using the backpropagation algorithm and a sigmoid activation function with a steepness parameter of 3.0 at a learning rate of 0.25 [6].

2.2 Training Procedure

In the experiments presented here, the robot operated in a rectangular room of 2.9 x 4.8 metres (see figure 3), whose walls consisted of different materials such as bricks, smooth plastics, painted wood and cardboard, and which contained three different shaped obstacles. The arrangement of the robot's 16 sonar sensors with respect to its environment is shown in the right picture of figure 3. Three sets of data were obtained by driving the real robot along particular routes in that environment: training data, validation data and test data. Training data was used to train the 16 neural networks to model the perception of their respective sensor. Validation data was used to determine the optimal point at which training should be suspended. The training and validation data was obtained along two different paths indicated in figure 4. The test data, finally, was used to evaluate the robot's prediction of sensory perception along a third path, which coincided with the training path only at a small number of points. Test data was obtained along the path shown in figure 5.

2.3 Determining the Optimal Training Duration

When training an artificial neural network, it is not usually easy to determine the optimal point at which to stop training the network. Using the minimum error obtained when training the network, using the training data, does not necessarily result in a low general error when the network model is actually applied to other test data due to the effect of over-training. Following the procedure widely used by co-workers (for example, [1]) we therefore obtained a second set of data, validation data, which was used to determine the optimal point at which to stop training the network. We define the network error as the absolute value of the distance in inches between the network's predicted range reading at a particular Cartesian position and the actual range reading obtained by the robot at that position. Figures 6 show how the validation data can be used to determine the optimal point at which to stop training the network. Referring to the top picture of figure 6, for instance, one can see that a minimum error using the training data is obtained after 8,200 training iterations. However, looking at the validation data, it becomes clear that the general error of the network is lowest after 5,000 training iterations, at which point over-training of the network sets in and the error increases again. Similar behaviour can be observed in the bottom picture of figures 6.

Accordingly, we have trained each neural network for each of the 16 sonar sensors until the network error, using validation data, begins to increase again. The number of training iterations required for each of the 16 sonars of the Nomad 200 is shown in table 1. Different numbers of iterations are required for modelling different sonar sensors. The reason for this is that the complexity of the sonar response depends very much upon incident angle and surface structure and different sonar sensors perceive different parts of the environment as the robot travels along a particular path. Figure 7, for instance, shows

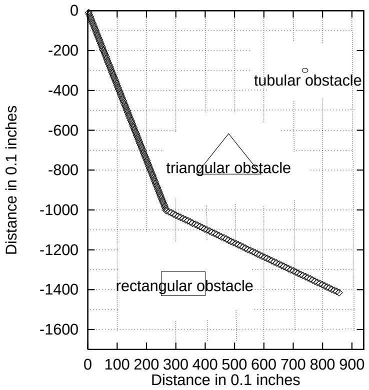

Figure 5: The dotted line shows the path used to collect training data, the "box" line shows the path for collecting test data.

the different perceptions of sonar sensor No. 4 and sonar sensor No. 15 as the robot travels along the test path indicated in figure 5.

It is clear that modelling sonar sensor No. 15 is more complicated than modelling sonar sensor No. 4. Training the network No. 15 accordingly required 10,000 learning steps, while training network No. 4 required only 1,000 training steps (see table 1). Figure 8 shows the final errors obtained along the testing route shown in figure 5 using either the testing data to determine the optimal training length or the validation data to determine the optimal training length. As can be seen from figure 8, there is only one case (sensor No. 15) in which determining the duration of training using validation data produces a worse result than using test data for that purpose, and even in that case the difference is marginal. However, in all the other cases using validation data to determine the optimal training length produces better or identical results. We have, therefore, adopted this method and use validation data to determine when to stop training the individual networks.

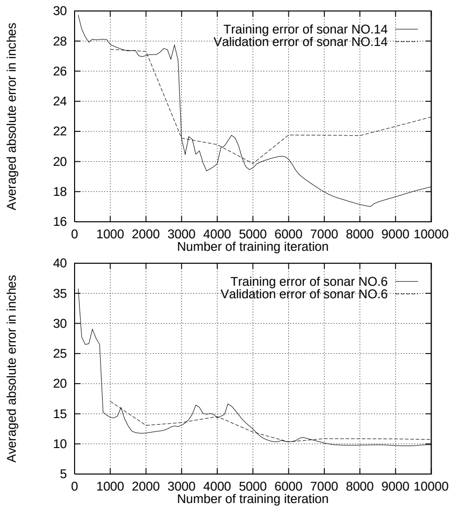

Figure 6: Error curves from the training data and the validation data of sonar No.14 (*Top*) and sonar No.6 (*Bottom*).

Sonar Number	By training	By validation
0	10,000	10,000
1	10,000	10,000
2	8,000	8,000
3	6,000	5,000
4	1,000	1,000
5	9,000	9,000
6	10,000	6,000
7	5,000	5,000
8	4,000	4,000
9	10,000	10,000
10	10,000	10,000
11	3,000	9,000
12	9,000	5,000
13	9,000	9,000
14	8,000	5,000
15	7,000	10,000

Table 1: The training iteration for the best network generalisation performance.

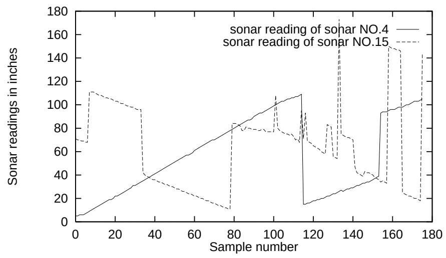

Figure 7: Comparison of sonar readings. Because of the non-uniform environment the sonar readings from sonar No.15 are more complicated than those from sonar No.4.; therefore, the best number of training steps for sonar No.4 is 1,000 but for sonar No.15 is 10,000. See Table 1.

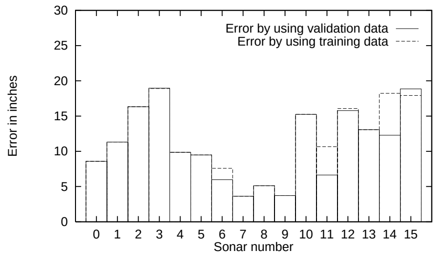

Figure 8: The testing errors from networks which are selected by using training error or selected by using validation error.

3 EXPERIMENTAL RESULTS

3.1 Modelling Perception

Our first experiments were concerned with modelling the perception of the individual sonars using the neural network model acquired in the method described earlier. In particular, we wanted to compare this acquired model with a simple sonar model (we refer to this model as the Nomad model). The latter simplified model is supplied by Nomadic, the manufacturers of the Nomad robot, and attempts to model the performance of sonar sensors facing one particular type of object with one particular set of physical properties. The Nomad model is a typical

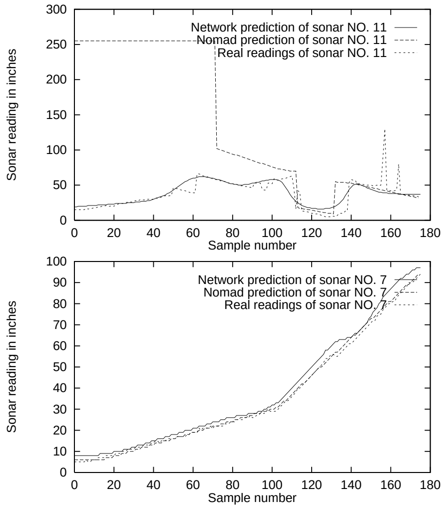

Figure 9: Comparison between network method and conventional method (Nomad simulator). The network prediction is close to the real sonar reading, while the Nomad prediction is very different. The Nomad simulator is only good in a simple case as sonar No. 7 (the bottom picture).

example of numerical models used in robotics research, taking into account fundamental physical properties of one type of surface, but not taking into account the variability across a particular target environment.

To evaluate the accuracy of the model, the data obtained along the testing route (see figure 5) was used. Figure 9 shows the predictions achieved along the test route for sonar sensors Number 11 and 7, pictures from top to bottom respectively. As can be seen from the top picture, the acquired network prediction follows the actual observed sensory readings more closely than the simplified numerical model used by the Nomad simulator. The Nomad simulator is only good in the simplest case such as the bottom picture of figure 9. The main reason for this effect is that the Nomad simulator is unable to model surfaces with different textures.

3.2 Modelling Robot-Environment Interaction

Having obtained models of the robot's sensory perception, we were interested to investigate whether these models could be used to model actual robot-environment interaction. For this purpose, we wrote a "Find Freest Space" control program, which worked in the following manner. Per time step, the robot is moved by one inch determining the direction of freest space by evaluating all 16 sonar range values and then moving the robot in the direction of the biggest reading by one inch. The program terminates after 100 time steps, that is after a journey of 100 inches. The trajectories of the real robot, the Nomad-simulated robot and the network-simulated robot are plotted to allow comparison. We used three different starting points in the target environment, obtaining three different trajectories. The trajectories taken by the real robot, the network-simulated robot and the Nomad-simulated robot are shown in figure 10. The top picture of figure 10 reveals that the Nomad simulator is unable to perceive the triangular obstacle in the middle of the room and therefore moves very close to the triangular obstacle where the robot stops, whereas the real robot's trajectory and the network-simulated trajectory coincide very well and the network's prediction is reasonably accurate. In the middle picture of figure 10, this effect is even more pronounced. Here, the Nomad-simulated robot will move towards a completely different position, compared to the real robot and the network-simulated robot. Our worst result is presented in the bottom picture of figure 10 where there are considerable differences between the trajectory predicted by the network simulator and the trajectory actually taken by the real robot. Looking at the trajectory taken by the real robot, it is clear from the erratic shape of the trajectory that effects like specular reflection have influenced the movement of the robot. This effect is particularly difficult to model and explains why the network model has difficulties in predicting the robot's trajectory accurately. It is the purpose of future research to determine precisely the reason for discrepancies such as the ones shown in the bottom of figure 10. However, our results show that normally the prediction made by the network simulator is accurate and can be used to model robot-environment interaction (see the top and middle picture of figures 10).

4 SUMMARY AND CONCLUSIONS

4.1 Summary

Numerical simulation of robot-environment interaction has a number of advantages, notably the low cost in implementation, fast execution and repeatable experimental conditions. These advantages are paid for by the low accuracy and low predictive quality of current numerical models of such robot-environment interaction. Typical approaches to numerical modelling model the physical properties of, for instance, sonar sensors and detecting one particular type of object in an environment [3, 14]. However, typical robot environments are not uniform and cannot be modelled faithfully using this approach. The result of such modelling, therefore, is that simulation results can only be transferred to real robot controls in very simple cases.

Contrary to the approaches mentioned above — installing *a priori* models of sensory perception using laws of physics — we attempted to acquire models of sensory perception of a specific robot (a Nomad 200 mobile robot) in a specific target environment. A set of 16 multi-layer perceptrons were trained to model each of the 16 sonar sensors of the robot. Using training data and validation data, we determined the optimal training lengths for each network individually and thus obtained perceptual models for each of the 16 sonar sensors of our Nomad 200 robot. These models were evaluated and compared against a numerical model based on physical properties: in our case the simulation environment supplied by Nomadic, the manufacturers of the Nomad 200 robot.

The perceptual models obtained in this manner were then used to predict the robot's trajectory in the target environment using a particular control program ("Find Freest Space") and the trajectories predicted by the net-

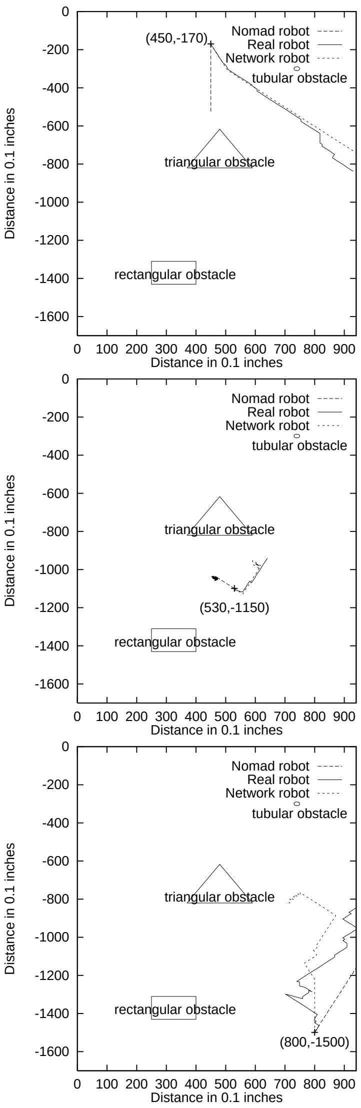

Figure 10: Robot trace comparison. The robot's start points are (*Top:* 450, -170), (*Middle:* 530, -1100), (*Bottom:* 800, -1500).

work simulator and the Nomad simulator were then compared. The results show that the network simulation is, in general, more faithful and closer to the actual trajectory observed in the real robot. However, even on the network simulator discrepancies between real robot trajectory and predicted trajectory occur. The effects of the causes of this are subject to ongoing research, but we expect these to be due to strong specular reflections and perceptual abnormalities at the edges of the experimental environment.

4.2 Conclusions and Future Work

In our experiments, the acquired simulator using artificial neural networks generally shows better performance than a numerical model modelling physical properties of sonar sensors. However, there are a number of points that need to be looked at before this can be used for large-scale modelling of robot-environment interaction. Firstly, the input to our model is based on Cartesian coordinates obtained from the robot's odometry system. Odometry is notoriously unreliable for all but short distances. Future research will attempt to remove this dependency upon odometry and replace the Cartesian coordinates used in training our networks by other location information. Secondly, we have only modelled small environments of just under 15 square metres. Future experiments will investigate whether this method is suitable for modelling middle-scale and, ultimately, large-scale environments. Thirdly, we have only modelled sonar sensors so far. The Nomad 200 robot possesses infrared sensors as well and it is our plan to use the approach presented in this paper to model both sonar and infrared sensors.

An approach like the one presented here is only suitable for modelling the interaction of a particular robot in a particular target environment. That is, the increase in faithfulness of simulation is bought at the expense of generality. However, we believe that this is the only feasible method of using numerical models to predict robots' behaviour, because generalised numerical simulations of robot perception are usually so different from the actually observed perceptions, that their predictive power is too limited to be of any real use for mobile robotics research.

References

[1] C. M. Bishop, Neural Networks for Pattern Recognition, Published by Oxford Univ. Press, 1997, pp. 343-345.

[2] R. A. Brooks, M. J. Mataric, Real Robots, Real Learning Problems. In Chapter 8, Robot Learning, Edited by J. H. Connell and S. Mahadevan, Kluwer Academic Publishers, 1993.

[3] C. Chen, M. M. Trivedi, C. R. Bidlack. Simulation and Animation of Sensor-Driven Robots. In IEEE Transaction on Robotics and Automation, Vol. 10, No. 5, OCT 1994.

[4] H. R. Everett. Sensors for Mobile robots: Theory and Application. Published by A K Peters, Ltd., 1995.

[5] I. Harvey, P. Husbands, D. Cliff. Issues in Evolutionary Robotics. In Proc. SAB92, the 2nd Int. Conf. on Simulation of Adaptive Behaviour. 1992.

[6] J. Hertz, Anders Krogh, Richard G. Palmer. Introduction to the Theory of Neural Computation. Published by Addison-Wesley, 1991.

[7] N. Jakobi, P. Husbands, I. Harvey. Noise and The Reality Gap: The Use of Simulation in Evolutionary Robotics. In Lecture Notes in Artificial Intelligence, 1995, Vol.929 pp.704-720.

[8] A. Lapedes, R. Farber. How Neural Nets Work. In Proc. IEEE Denver Conf. on Neural Nets, 1987.

[9] T. Lee, U. Nehmzow, R. Hubbold. Faithful Simulation of Autonomous Mobile Robots by Means of Model Acquisition. Technical Report Series, UMCS-97-12-1, Computer Science Dept., Manchester Univ., ISSN 1361-6161, 1997.

[10] H. H. Lund and O. Miglino. From Simulated to Real Robots. In Proc. IEEE 3rd Int. Conf. on Evolutionary Computation. IEEE Press, 1996.

[11] M. M. Trivedi, M. A. Abidi, R. O. Eason, R. C. Gonzalez. Developing Robotic Systems with Multiple Sensors. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 20, No. 6, NOV/DEC 1990.

[12] U. Nehmzow, T. Mitchell, The Prospective student's Introduction to the Robot Learning Problem. Technical Report Series, UMCS-95-12-6, the Department of Computer Science, Manchester Univ., DEC 1995.

[13] N. Oreskes, K. Shrader-Frechette and K. Belitz, *Verification, Validation, and Confirmation of Numerical models in the Earth Sciences*, Science, Vol 263, 4 Feb 1994, pp. 641-646.

[14] K. Song, W. Tang, Environment Perception for a Mobile Robot Using Double Ultrasonic Sensors and a CCD Camera. IEEE Transactions on Industrial Electronics. Vol. 43, No. 3, JUN 1996.