ISSN 1361 - 6161

THE UNIVERSITY of MANCHESTER

Computer Science
University of Manchester

Faithful Simulation of Autonomous Mobile Robots by Means of Model Acquisition

Ten-min Lee, Ulrich Nehmzow and Roger Hubbold

Department of Computer Science

University of Manchester

Technical Report Series

UMCS-97-12-1

Faithful Simulation of Autonomous Mobile Robots by Means of Model Acquisition*

Ten-min Lee, Ulrich Nehmzow and Roger Hubbold
Department of Computer Science,
University of Manchester
Oxford Road,
Manchester M13 9PL,
England.
`leet, nehmzow, hubbold@cs.man.ac.uk`

December 1997

*Reviewed by Andreas Bühlmeier

Abstract

A novel method of mobile robot simulation is presented using an artificial neural network; a physical mobile robot is used to acquire a model of a specific environment, which can later be used to simulate robot environment interaction. Laboratory experiments are presented that demonstrate the acquisition of individual sensor models through robot-environment interaction, comparing the performance of a conventional simulated environment with our novel approach. It is shown that trajectory prediction is considerably more accurate, using acquired models.

1 Introduction

Faithful numerical modelling (simulation) of autonomous mobile robots offers a number of desirable advantages. Firstly, prototyping time and cost is dramatically reduced. Secondly, because of the deterministic nature of a numerical model of a robot, experiments can be repeated and replicated, facilitating independent confirmation of results. Most existing mobile robot simulators use simplified models of sensory perception and action; typically, computer graphics techniques are used to represent a simulated environment and the robot while deterministic and generic equations are used to model sensory perception and actuation [3, 6]. The limitations of simplified robot simulators are well understood [2, 10]. A number of attempts have been made to create faithful mobile robot simulators [1, 2, 4, 5, 8, 9]. All of these approaches have in common that sensory perception is modelled by simplified mathematical equations generating the simulated data. In c

1.1 What is being modelled?

The purpose of mobile robot simulation is to model the interaction of a robot with its environment. It is, therefore, "important to remember that it is not our world that is being simulated, but the robot's" [4]. This is one reason why it is desirable to model the world as the robot actually sees it. A second reason is that there are perceptual discrepancies between the human's observation of the world and the robot's. "The human designer inevitably has a very different perception of the world compared with the robot's view" [7]. Therefore, we contend that faithful robot simulation should be based on the robot's perception of its environment, rather than the human's perception of the robot's environment. Consequently, we have developed a mobile robot simulation environment in which the model of the robot's environment is acquired through robot learning, rather than being pre-installed by the human operator.

The paper is organised as follows. Preliminary experiments regarding the modelling of sensory perception only are presented in section 3. Experiments regarding the simulation of robot-environment interaction are presented in section 4. Section 5 presents a summary and draws conclusions.

2 Experimental platform

For the experiments reported in this paper, a Nomad 200 autonomous mobile robot was used. The Manchester *Forty-Two* is shown in Figure 1.

The robot is equipped with 16 sonar sensors, 16 infrared sensors, 20 tactile sensors and proprioceptive sensors. Nomadic Technologies supplied a simulator of the Nomad 200 robot (see Figure 2), which was used for comparison with the network simulator developed in our research.

The Nomadic Technologies simulator provides a graphical environment, allowing users to run the simulated robot in a user-defined world. All objects modelled in the simulator have the same perceptual qualities; in other words, all objects appear perceptually identical to the simulated robot. As the following experiments will show, in reality different objects in the world have different perceptual qualities, so that using the Nomad simulator will result in discrepancies between simulated and actual perception.

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.

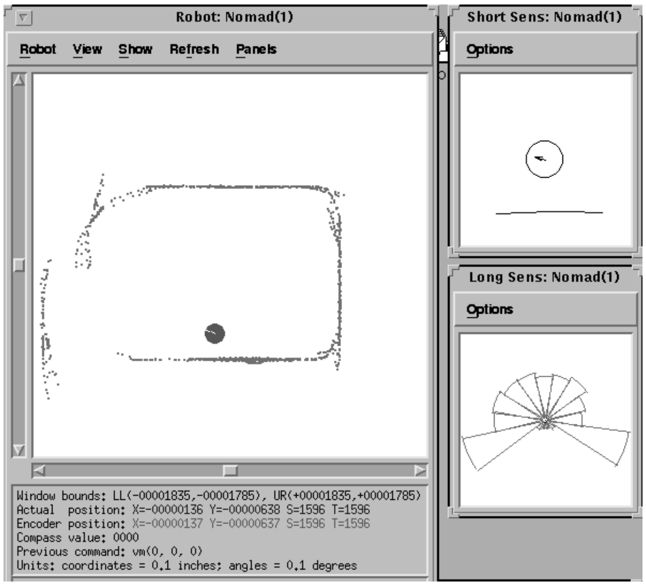

Figure 2: The Nomad simulator.

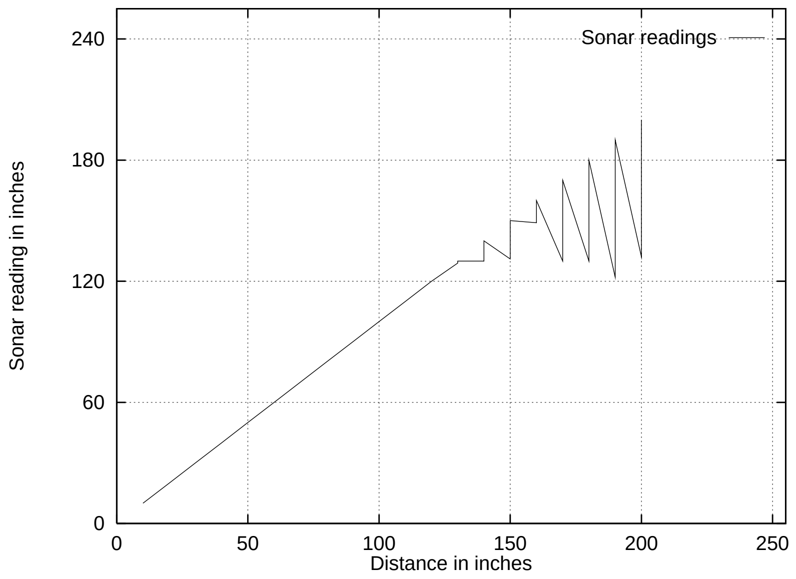

Figure 3: Sonar range finder response versus distance.

3 Modelling sonar perception

Robot simulation requires the modelling of both robot perception and robot action. In order to determine whether the modelling of perception or the modelling of action is the crucial component, we conducted a series of experiments attempting to replicate sonar perception through a numerical model. In the first experiment, the basic characteristics of the sonar sensors of *Forty-Two* were determined. The robot was placed facing a cardboard box (21 x 22 x 24 inches) and the readings from the front facing sonar sensor were recorded. The distance between the robot and the box was increased in steps of 10 inches up to a maximum distance of 200 inches. Range readings obtained are shown in Figure 3.

As can be seen from that Figure, the sonar sensors show a linear characteristic up to a distance of about 120 inches; beyond that distance readings become unreliable. In a second experiment regarding the sonar characteristics we investigated the performance of sonar sensors with respect to detecting obstacles made of different materials. The robot was placed facing a wall at a fixed distance, and then turned smoothly at a range of 1 degree per second until the turning angle, with respect to the wall, was greater than 75 degrees (see Figure 4). Range readings obtained from the sonar sensor are shown in Figure 5.

As can be seen from Figure 5, the range readings obtained from the sonar sensors are dependent upon the surface structure of the object detected. Whilst smooth cardboard walls cannot be detected beyond an incident angle of more than 40 degrees, rough brick walls reliably return sonar pulses even beyond an angle of 75 degrees. The Nomad simulator has a maximum incident angle of 60 degrees, which can be specified in a setup file. The results indicate that the Nomad simulator simulates one specific type of wall. This presents a fundamental restriction in using the simulator for real world applications, as typical environments consist of more than one type of object. By attempting to *acquire* a model rather than using a *pre-installed* model, we contend that those restrictions can be overcome. Secondly, we believe that specific effects, stemming from unique perceptions occurring only at specific locations in the environment, can also be modelled through our acquired model whereas it is impossible for the Nomad simulator to take those effects into account.

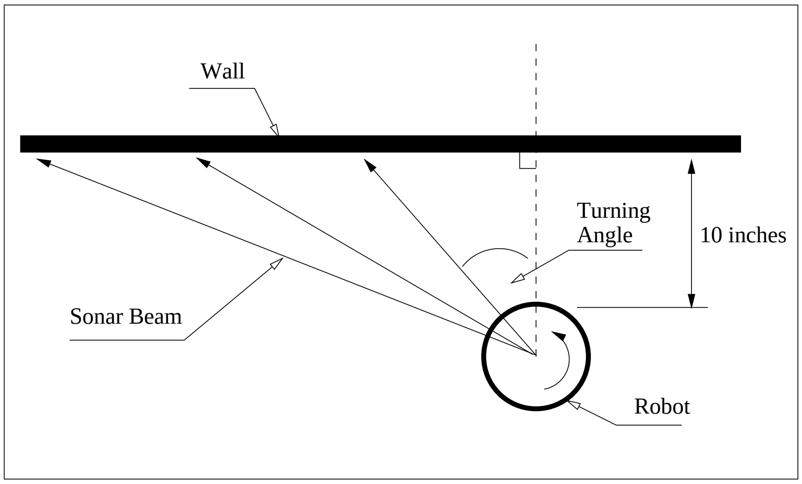

Figure 4: Experimental setup to determine sonar range finder responses to walls of different surface structure.

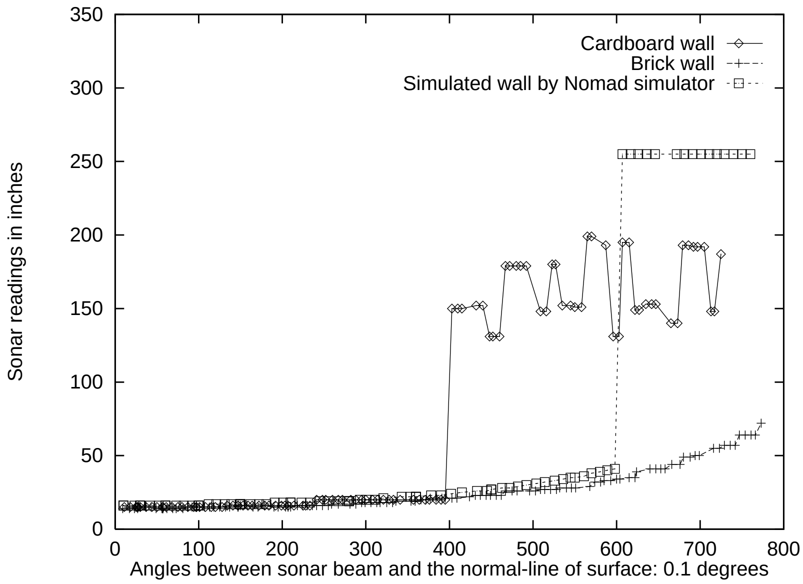

Figure 5: Sonar range finder responses to walls of different surface structure. On smooth walls specular reflection begins at incident angles above 40 degrees, rough walls are detected even at incident angles beyond 75 degrees. The Nomad simulator has a preset cutoff angle of 60 degrees.

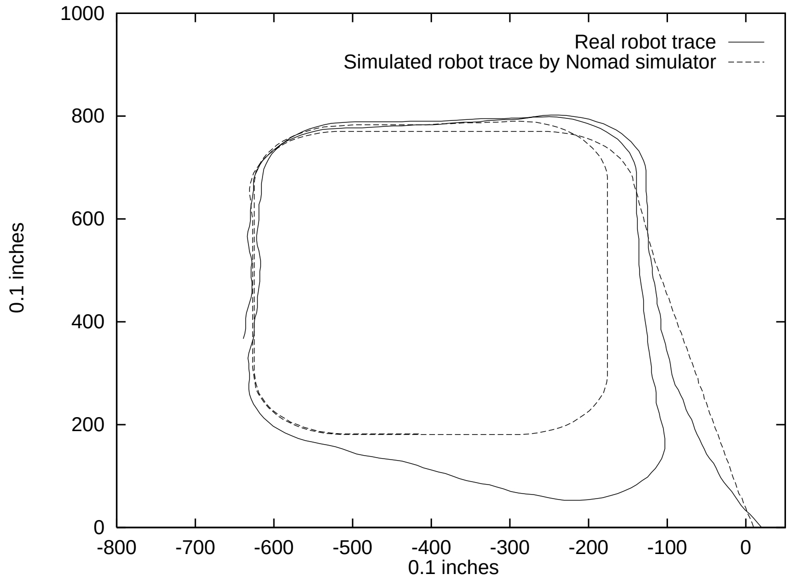

Figure 6: Trajectories followed by a simulated and a real robot, being controlled by the same control program. The wall at the bottom has a smoother surface than the other three walls, resulting in a discrepancy between the trajectory taken by the simulated robot and that by the real robot.

3.1 The cause of discrepancies between real and simulated robot

In order to address the question of building a numerical model of a mobile robot, we first investigated the source of discrepancies between real and simulated robot behaviour. Fundamentally, there are two possible sources for those discrepancies. First the modelling of robot *perception* and, second, the modelling of robot *action*. Figure 6 shows the trajectories of a simulated and the real robot in a square environment, in which the robot was executing a wall-following program.

The simulated environment consisted of four identical walls, whereas the real environment consisted of three identical walls with a fourth wall having a smoother surface. As can be seen from Figure 6, the simulated robot performs a perfect wall-following behaviour, whereas the real robot showed a marked deviation towards the smooth wall. We then conducted two further experiments to determine the cause of this discrepancy. In the first experiment the simulated robot was driven by real robot sonar readings. The trajectory taken by the robot in this experiment is shown in Figure 7.

In a second experiment, the real robot was driven by simulated sonar data. The trajectory taken by the robot in this second experiment is shown in Figure 8. It shows that the real robot is moving along a trajectory, which is very similar to that of the simulated robot shown in Figure 6. The conclusions we draw from these two experiments are that the cause of discrepancy between simulated and real robot behaviour is in the modelling of the robot's perceptions, and not in the modelling of the robot's actuators. It is therefore the modelling of the robot's perceptions that we concentrated our research on.

3.2 Acquiring models of mobile robot perception using artificial neural networks

For the reasons mentioned earlier – different surface properties of real world objects and sensor idiosyncrasies – we decided to enable the robot to acquire a model of the environment, rather than pre-installing this model. Figure 9 shows the fundamental approach we took.

The robot's position in Cartesian space is used as an input to an artificial neural network which associates that position in Cartesian space with a range reading obtained from a sonar sensor. Using

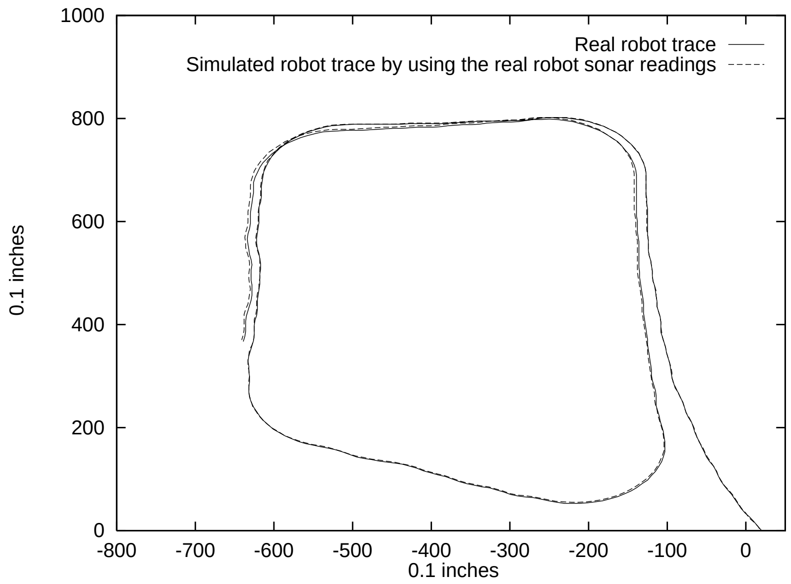

Figure 7: The trajectory of the simulated robot resembles that of the real robot if the simulated robot's controller is supplied with sonar readings obtained from the real robot.As can be seen from Figure 7, the trajectory taken is identical to the one taken by the real robot in the real environment.

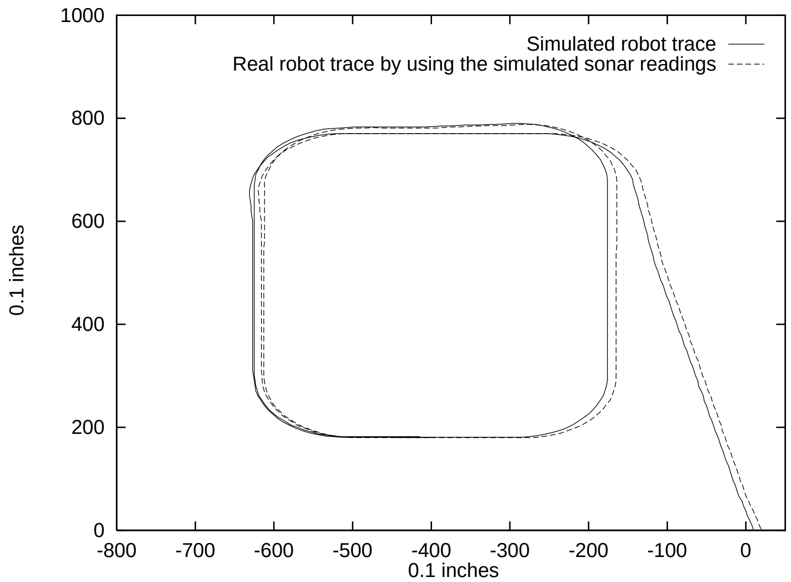

Figure 8: Again, the trajectories of the real and the simulated robot resemble one another if the real robot's controller is supplied with simulated sonar range finder data.

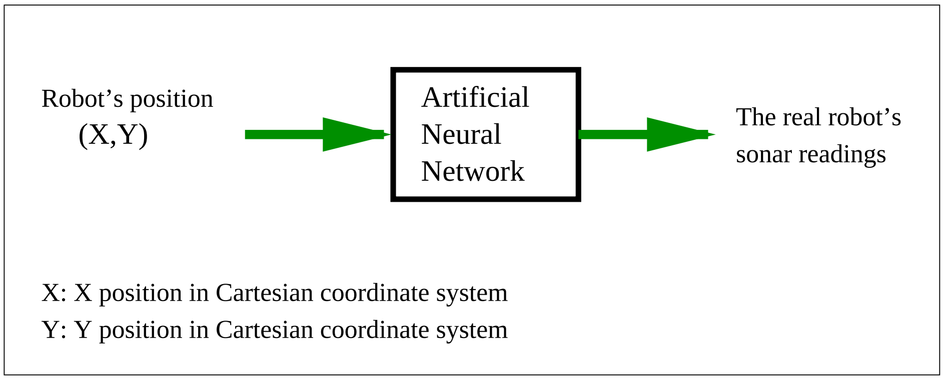

Figure 9: Association of robot position with sensory perception by means of an artificial neural network.

this approach, the complexity of the real environment can be simulated more faithfully than had a pre-installed model been used. To train the artificial neural network, the position of the robot in Cartesian space obtained from the robot's wheel encoders is used as input data. The range reading of a particular sonar sensor obtained at that position in space is used as the target output of the network. All sixteen sonar sensor of *Forty-Two* were modelled in this fashion. The network is shown in Figure 10.

Sixteen neural networks, as shown in Figure 10, were used to model the perceptions of the sixteen sonar sensors of *Forty-Two*. The inputs to each of the networks were the x and y position of the robot as obtained from the robot's odometry system. The output of each network was the sonar range reading obtained from the sonar sensor being modelled.

4 Experiments

4.1 Modelling a single wall

In the first experiment, using the artificial neural network shown in Figure 10, the robot's task was to model a single wall as shown in Figure 11.

The wall was made of rough-surfaced brick but had a closed door on it with a smooth surface. The robot's turret was fixed throughout the experiment and the sonar sensor facing the wall at an angle of 45 degrees was being modelled. At the origin (top left of Figure 11), the robot's odometry system was reset to zero. As the robot was moving along the teaching path indicated in Figure 12, the sonar range readings obtained from the sonar facing the wall at 45 degrees were recorded. 908 data points were obtained along the teaching path indicated by the solid line shown in Figure 12.

The perceptions of the modelled sonar are shown in Figure 13.

Having obtained the teaching data in the way described, the neural network was trained for 2500 iterations using all 908 sets of data of the teaching set. Figure 13 shows the response of the trained network to the teaching data. Figure 14 shows the decrease of network error with increasing number of training steps.

The training error was defined as the average difference between actually obtained network output and desired network output in the teaching set. Figure 15 shows the network's response to the testing data, in comparison with the actually obtained real sonar reading during the test run (the test run is shown in Figure 12). As can be seen from Figure 15 the acquired model of the sonar sensor facing the wall corresponds well with the actually obtained perception whilst driving the robot along the wall. Comparing teaching route and testing route (Figure 12), one can see that the neural network model has generalised the robot's perception. Only very few points in Cartesian space are identical during the teaching route and the testing route. Nevertheless, even at locations previously unvisited, the neural

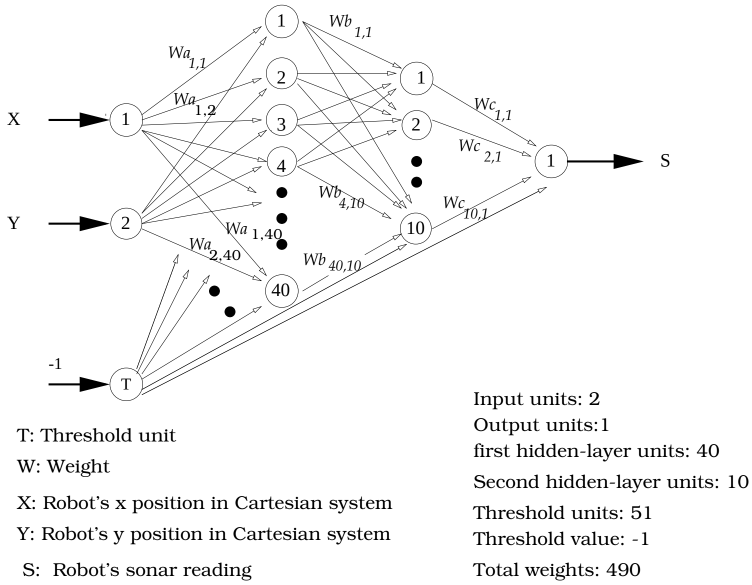

Figure 10: The two-hidden-layer neural network used to simulate *Forty Two's* sonar sensors.

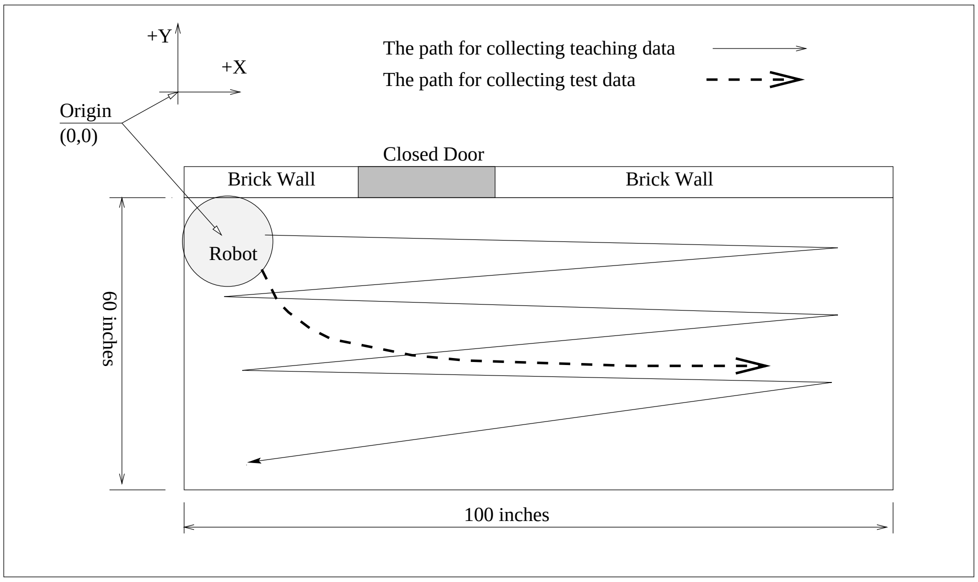

Figure 11: Experimental setup for collecting sonar range sensor readings in the one-wall environment. The door shown has a smoother surface than the brick walls to the left and right of the door.

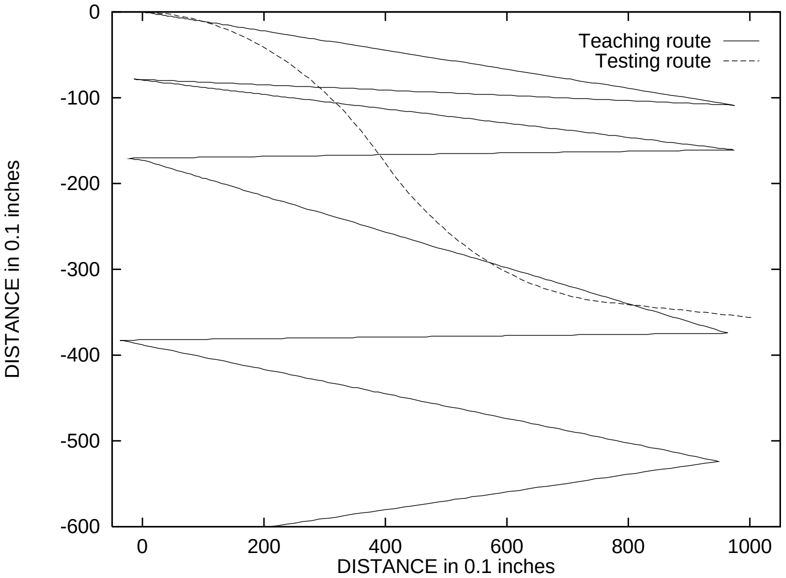

Figure 12: Robot trajectories for collecting teaching and testing data. Note that both routes overlap only in few points, meaning that for predicting the sensor readings along the testing route the network has to use generalisations obtained during training.

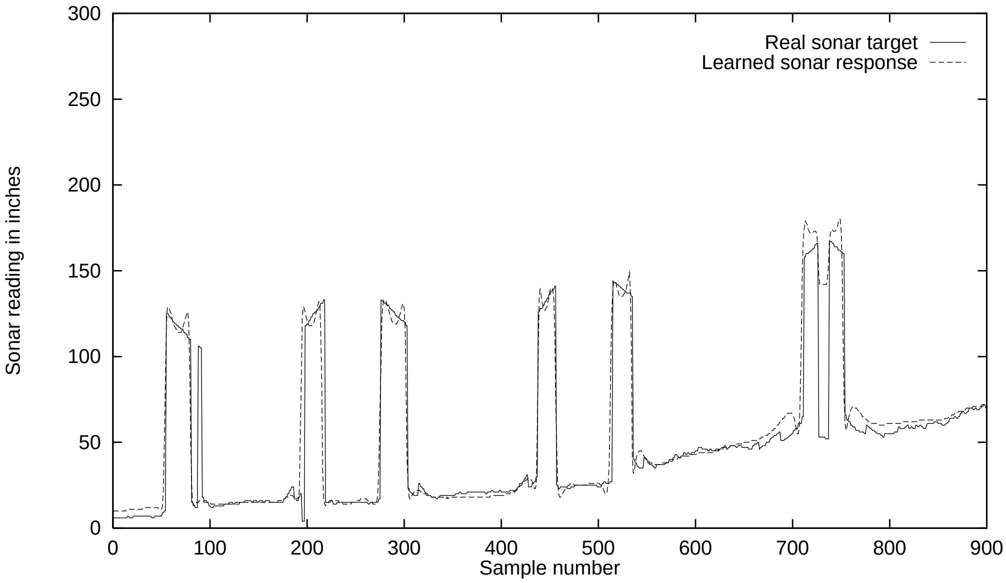

Figure 13: The response of the fully trained network to the teaching data.

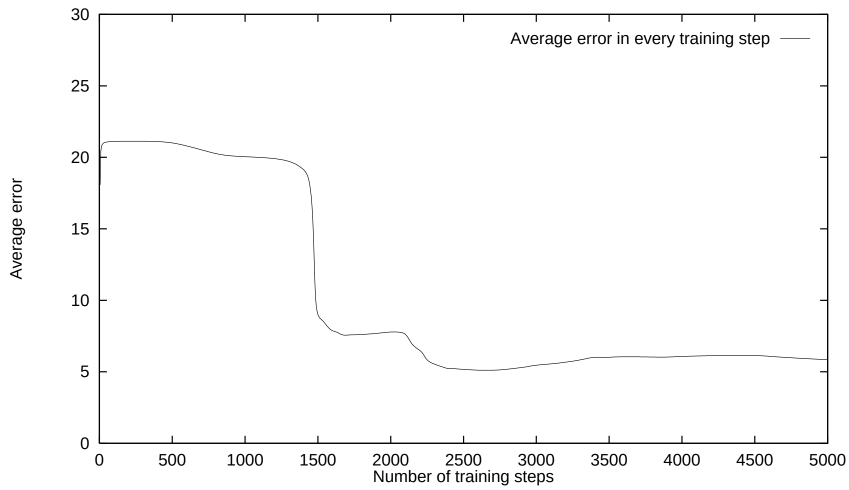

Figure 14: Average prediction error versus training time.

network model of the sonar sensor retains the characteristic features of the actual perception. Figure 15 also shows the response of the Nomad simulato

4.2 Closing the loop between perception and action

Using the model acquired in the previously presented experiment, a wall-following program was executed on *Forty-Two*. This program instructed the robot to use only sonar sensor number 2 (see Figure 16) to keep at a distance of 10 inches from the wall.

The resulting trajectory (see Figure 17) shows that the actual robot hit the wall at the point where the smooth door was located. Our network simulator also predicted that collision (see Figure 17), whereas the Nomad simulator was unable to predict the collision.

This result, therefore, shows that the actual trajectory taken by the real robot and the trajectory predicted by the network model agree well, whereas the generic Nomad simulator failed to predict the collision which is shown in Figure 17.

4.3 Closing the loop between perception and action: Experiment two

The first experiment (see previous section) demonstrates that the sensory model acquired by the artificial neural network and created by using the robot's own perceptions of the environment can indeed model robot-environment interaction faithfully. The purpose of the second experiment described now was to determine whether this modelling is possible for more complicated environments and for more complicated robot control programs. In this second experiment a closed room of 90 x 100 sq. inches was used. Walls were made of four different materials, the environmental layout is shown in Figure 18.

During this experiment the robot's turret was kept stationary, facing to the right in Figure 18. All 16 sonar sensor readings and the corresponding positions were recorded both during the teaching and testing path data acquisitions. To minimise the influence of odometry errors during testing data acquisition, the robot's odometry system was recalibrated after each traversal of the environment, by taking the robot back to the origin (shown at the top left of Figure 18). A typical sonar reading obtained in the testing route is shown in Figure 19.

The artificial neural network was then trained for 10,000 iterations of the entire teaching set to model the sonar sensor response obtained from the real robot during the teaching run. The network's response to the x-y co-ordinates occupied by the robot during the teaching run is shown in Figure 20. It shows that the network is able to learn the teaching pattern by 10,000 training iterations.

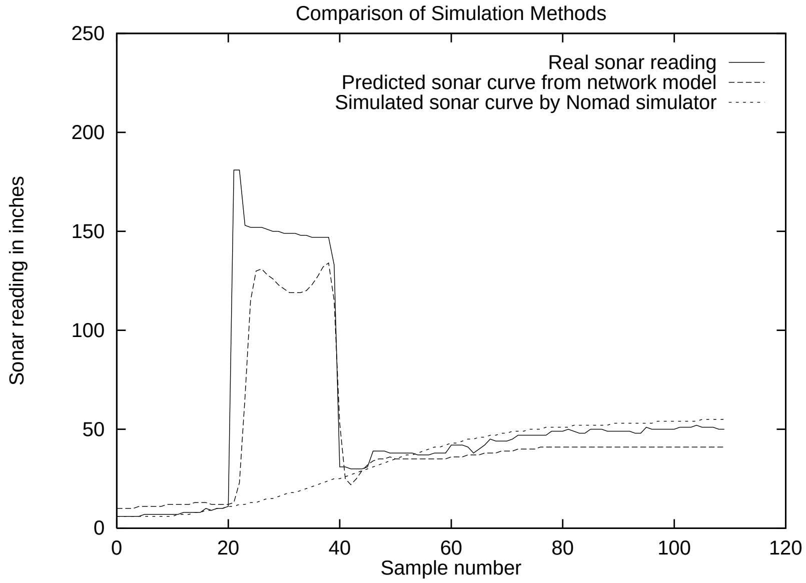

Figure 15: The network simulator's response to the test data.

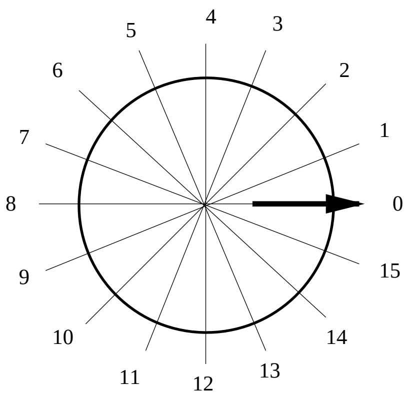

Figure 16: Arrangement of *Forty Two's* sonar range sensors.

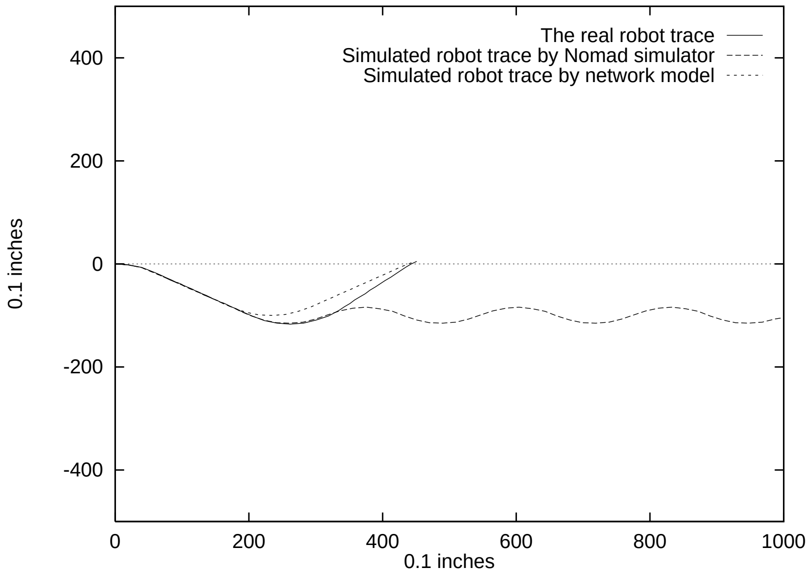

Figure 17: Assuming a homogeneous wall surface, the Nomad simulator fails to predict the collision that the real robot encounters, due to specular reflections. The network simulator *is* able to predict the collision.

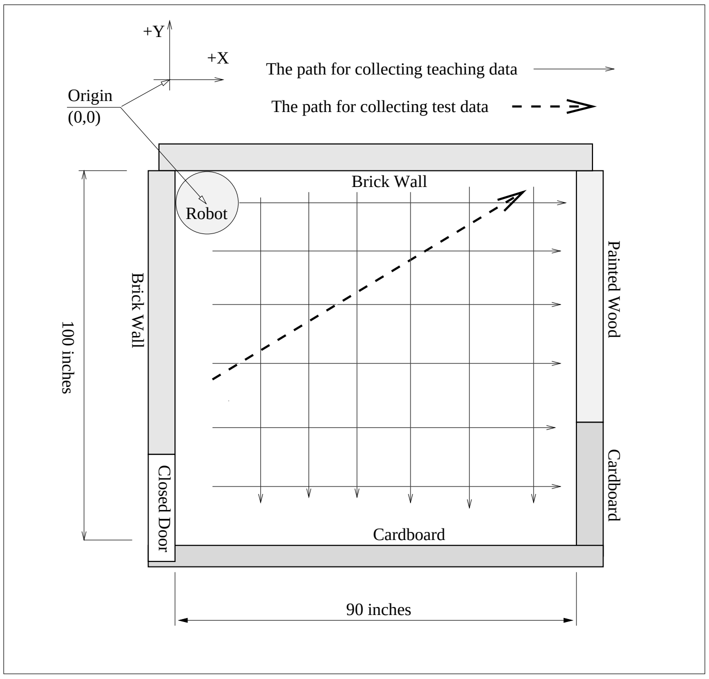

Figure 18: Experimental setup for the second experiment in robot-environment interaction. Walls have different surface structure.

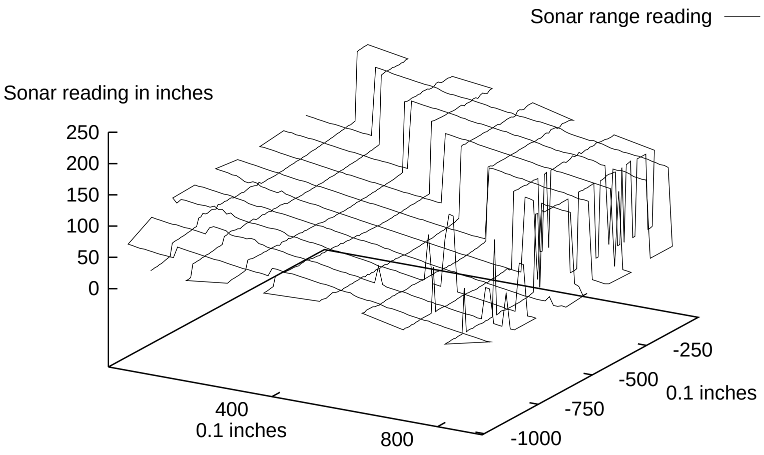

Figure 19: The sonar reading of one of the sixteen sonar sensors of *Forty Two*, obtained along the teaching route shown in Figure 18.

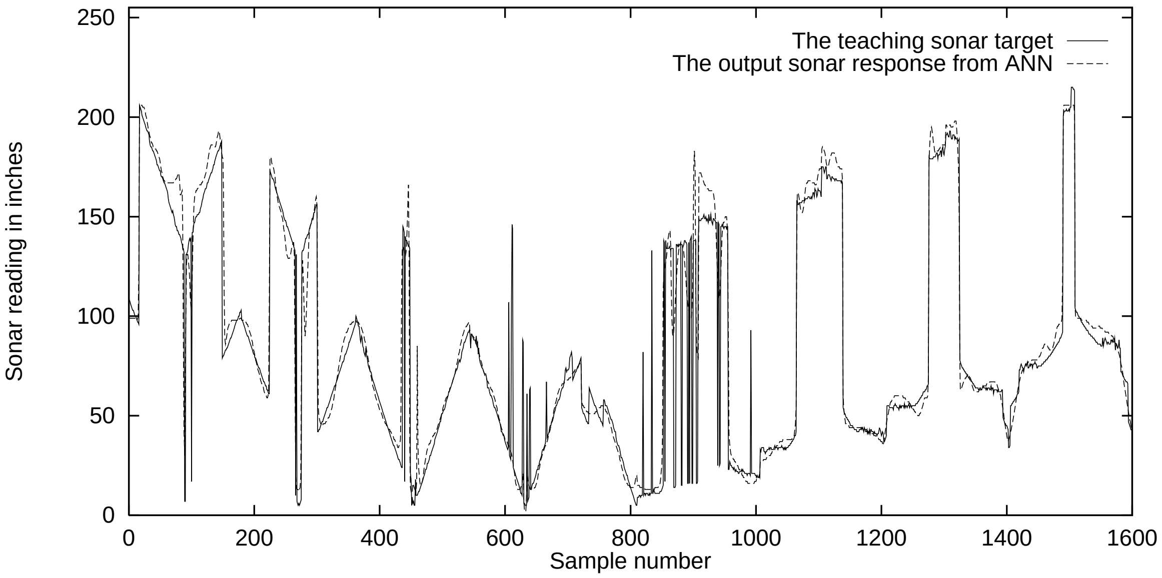

Figure 20: The network's response to the teaching data after 10000 training iterations.

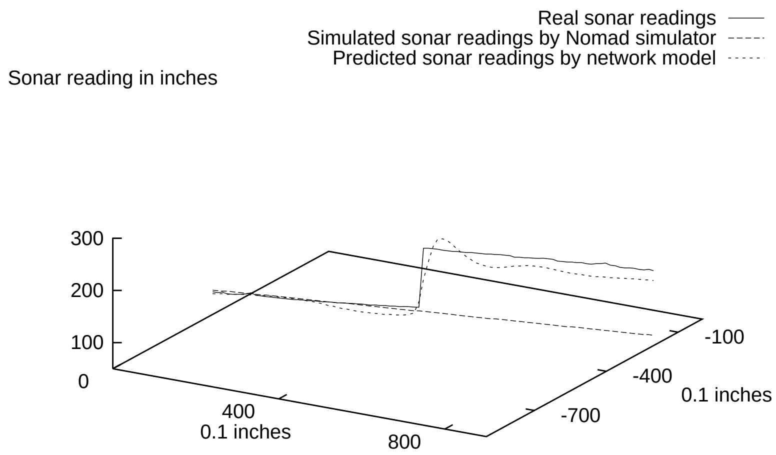

Figure 21: Network response to test data. The Nomad simulator is unable to predict the sudden increase in range reading due to specular reflection, while the network simulator predicts this correctly.

Network performance was then evaluated using the data obtained along the test run shown in Figure 18. The robot's sonar sensor No. 15 (see Figure 16) is one of the sensors that show a most pronounced difference between Nomad simulator prediction and actual observation on the real robot. Figure 21 shows the simulated and actual range readings of sonar sensor 15. As can be seen from Figure 21, the network simulator predicts a marked step function visible halfway along the path, resulting from specular reflections off a smooth surface, while the Nomad simulator is unable to predict this step function.

Again, closing the loop between perception and action, a “find-freest-space” control program was executed on the real robot and compared to the trajectories predicted by the Nomad simulator and the network simulator. The functionality of the ”find-freest-space” program is to move the robot 1 inch per time step, determining the direction of freest space by evaluating all sixteen sonar range values and moving the robot in the direction of the furthest reading by 1 inch. The program will stop either after 100 time steps or when the robot's infrared sensors detect an obstacle closer than a pre-defined threshold. The robot starts movement from a specified location within the environment (see Figure 22).

Figure 22 presents the trajectory taken by the real robot, the predicted trajectory by the Nomad simulator and the predicted trajectory by our network simulator. As can be seen from Figure 22, due to specular reflection the real robot will actually move towards a painted wall on the right hand side of the environment and collide with it. Due to its inability to model different types of surface structure, the Nomad simulator is unable to predict this collision and instead moves the simulated robot to the centre of the environment, as indeed would be suggested by a “find-freest-space” program operating in a homogeneous environment. On the other hand, our network simulator *is* able to predict the robot's collision with the wall of painted wood, as can be seen on the right hand side of Figure 22.

5 Summary and Conclusion

Faithful simulation of autonomous mobile robots is desirable for a number of reasons. First of all, it will facilitate independent and repeated replication of experiments. It will, furthermore, reduce prototyping time and therefore costs. However, current simulators make assumptions which render them unsuitable for faithful simulation of actual robot environment interaction. Most notably, they assume a uniform

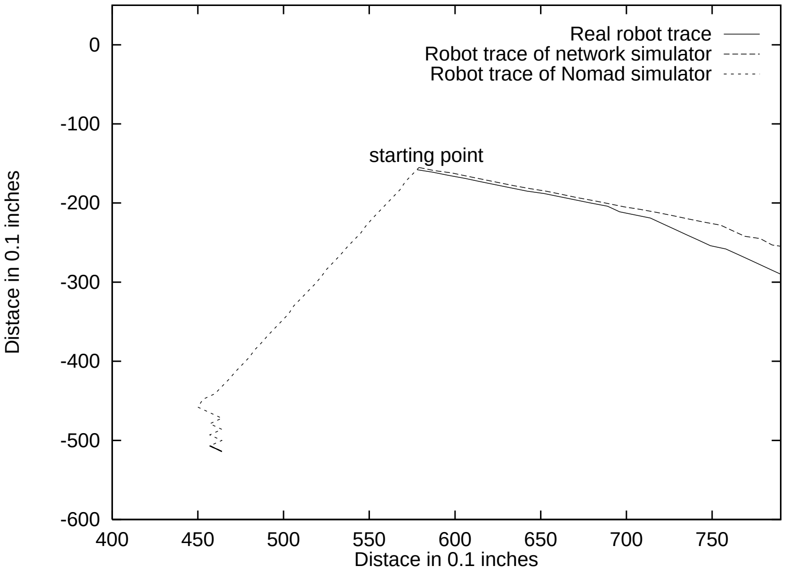

Figure 22: Simulated and real robot trajectories in response to the "find freest space" program. The geometrical centre of the robot's environment is at (450,-450).

model of objects in the world and are unable to model the environment as the robot sees it. We have, therefore, conducted experiments in the area of autonomous model acquisition facilitating the acquisition of a model of robot environment interaction based on a physical robot's perception of its environment. The experiments presented in this paper show that first the fundamental cause of discrepancies between simulated and real robot behaviour are the discrepancies in modelling the robot's *perceptions* rather than the modelling the robot's *actions*. We have, therefore, concentra

However, there are a number of research questions that have to be addressed if this sort of simulation is to be used on a larger scale. First of all, the input data for training the network is obtained from the robot's odometry sensors. Odometry is inherently unreliable due to wheel slippage effects. In our experiments we have minimised the impact of odometry errors through frequent recalibration. Odometry errors are no problem during the simulation phase as the odometry of a simulated robot is perfect and does not suffer from wheel slippage effects. We therefore believe that the use of odometry to train the artificial neural network may not be a problem even for scaled-up experiments, but intend to investigate this further.

The main advantage of our simulation architecture is that it can model environments from the robot's perspective, but it can take into account specific perceptual properties of individual objects within the environment and that it can cope with unusual sensor characteristics as, for example, the one shown in Figure 3. Further work is currently being conducted at Manchester to determine whether the proposed simulation architecture can be scaled up to environments of realistic size.

References

[1] 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.

[2] ChuXin Chen, Mohan M. Trivedi, Clint R. Bidlack. Simulation and Animation of Sensor-Driven Robots. In IEEE Transaction on Robotics and Automation, Vol. 10, NO. 5, OCT 1994.

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

[4] I. Harvey, P. Husbands, D. Cliff. Issues in Evolutionary Robotics. In Proceedings of SAB92, the Second International Conference on Simulation of Adaptive Behaviour. 1992.

[5] Nick Jakobi, Phil Husbands, Inman 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.

[6] Alonzo Kelly. Essential Kinematics for Autonomous Vehicles. Technical Report, CMU-RI-TR-94-14, Carnegie Mellon University, MAY 1994.

[7] Ulrich Nehmzow, Tom Mitchell, The Prospective student's Introduction to the Robot Learning Problem. Technical Report Series, UMCS-95-12-6, the Department of Computer Science, Manchester University, DEC 1995.

[8] 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.

[9] Mohan Manubhai Trivedi, Mongi A. Abidi, Rechard O. Eason, Ralph C. Gonzalez. Developing Robotic Systems with Multiple Sensors. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 20, NO. 6, NOV/DEC 1990.

[10] 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.