

# Route Training in Mobile Robotics through System Identification

Roberto Iglesias, Theocharis Kyriacou, Ulrich Nehmzow, and Steve Billings

**Abstract**—Fundamental sensor-motor couplings form the backbone of most mobile robot control tasks, and often need to be implemented fast, efficiently and nevertheless reliably. Machine learning techniques are therefore often used to obtain the desired sensor-motor competences.

In this paper we present an alternative to established machine learning methods such as artificial neural networks, that is very fast, easy to implement, and has the distinct advantage that it generates *transparent*, analysable sensor-motor couplings: system identification through nonlinear polynomial mapping.

This work, which is part of the RobotMODIC project at the universities of Essex and Sheffield, aims to develop a theoretical understanding of the interaction between the robot and its environment. One of the purposes of this research is to enable the principled design of robot control programs.

As a first step towards this aim we model the behaviour of the robot, as this emerges from its interaction with the environment, with the NARMAX modelling method (Nonlinear, Auto-Regressive, Moving Average models with eXogenous inputs). This method produces explicit polynomial functions that can be subsequently analysed using established mathematical methods.

In this paper we demonstrate the fidelity of the obtained NARMAX models in the challenging task of robot route learning; we present a set of experiments in which a Magellan Pro mobile robot was taught to follow four different routes, always using the same mechanism to obtain the required control law.

**Keywords**—Mobile robotics, system identification, non-linear modelling, NARMAX.

## I. INTRODUCTION

### A. Motivation

THE production of robot control code is often a difficult and complex task for the roboticist. Apart from relatively simple behaviours, for example, obstacle avoidance or wall-following, tasks such as route learning involve several days or weeks of refining code by trial and error until the robot achieves the desired behaviour. This method of producing code is time consuming and therefore inefficient.

Robot learning techniques make this problem a lot easier. In these cases a set of sensor-motor examples (the training set) for the particular task are used to change the parameters of a task model iteratively, until the model performs within the bounds of an acceptable error. The extent of the training (or model refinement) is usually determined by comparing the

Roberto Iglesias is with the Department of Electronics and Computer Science, University of Santiago de Compostela, Spain.

Theocharis Kyriacou and Ulrich Nehmzow are with Department of Computer Science, University of Essex, UK.

Steve Billings is with Department of Automatic Control and Systems Engineering, University of Sheffield, UK.

performance of the model to that of a validation set of sensor-motor data.

Using artificial neural networks (ANN) is one established approach applied in robot learning. This method speeds up the development of a reactive robot controller significantly; however, it also produces an opaque model that cannot be used to investigate the characteristics of the robot's behaviour further. Such investigation may include stability analyses, sensitivity of the behaviour to particular sensors or environmental features, sensor redundancy *etc.* Furthermore, the structure of a neural network model does not allow a straightforward comparison between models of similar tasks. This is very often necessary in robotic science.

The aim of the RobotMODIC project therefore is to develop a theoretical foundation for robotics. We believe that developing an understanding of robot behaviour will eventually lead to faster, more focused approaches to problem solving.

In this paper we present how we model route learning tasks using the NARMAX (Nonlinear, Autoregressive Moving Average models with eXogenous inputs) system identification process. This process yields explicit polynomial function to represent the task in hand. It is fast and requires no "guess work" as far as the structure or size of the polynomial function that would be suitable for modelling the robot's task. In addition, there is no need to do any pre-processing of the input space to assist the modelling process. The most important feature of this method, however, is that we can analyse the models obtained using established mathematical methods in order to understand robot-environment interaction. Examples of such analyses on previously obtained models and a further discussion of the advantages of the NARMAX modelling procedure for robot tasks such as wall-following and door traversal are presented in [1], [2].

The purpose of this paper is to demonstrate how well the NARMAX models can represent route learning tasks.

### B. The NARMAX Modelling Procedure

The NARMAX modelling approach is a parameter estimation methodology for identifying the important model terms and associated parameters of unknown nonlinear dynamic systems. For multiple input, single output noiseless systems this model takes the form:

$$y(n) = f(u_1(n), u_1(n-1), u_1(n-2), \dots, u_1(n-N_u), u_1(n)^2, u_1(n-1)^2, u_1(n-2)^2, \dots, u_1(n-N_u)^2, \dots,$$

$$\begin{aligned}
& u_1(n)^l, u_1(n-1)^l, u_1(n-2)^l, \dots, u_1(n-N_u)^l, \\
& u_2(n), u_2(n-1), u_2(n-2), \dots, u_2(n-N_u), \\
& u_2(n)^2, u_2(n-1)^2, u_2(n-2)^2, \dots, u_2(n-N_u)^2, \\
& \dots, \\
& u_2(n)^l, u_2(n-1)^l, u_2(n-2)^l, \dots, u_2(n-N_u)^l, \\
& \dots, \\
& \dots, \\
& u_d(n), u_d(n-1), u_d(n-2), \dots, u_d(n-N_u), \\
& u_d(n)^2, u_d(n-1)^2, u_d(n-2)^2, \dots, u_d(n-N_u)^2, \\
& \dots, \\
& u_d(n)^l, u_d(n-1)^l, u_d(n-2)^l, \dots, u_d(n-N_u)^l, \\
& y(n-1), y(n-2), \dots, y(n-N_y), \\
& y(n-1)^2, y(n-2)^2, \dots, y(n-N_y)^2, \\
& \dots, \\
& y(n-1)^l, y(n-2)^l, \dots, y(n-N_y)^l
\end{aligned}$$

were  $y(n)$  and  $\mathbf{u}(n)$  are the sampled output and input signals at time  $n$  respectively,  $N_y$  and  $N_u$  are the regression orders of the output and input respectively and  $d$  is the input dimension.  $f(\cdot)$  is a non-linear function, this is typically taken to be a polynomial or wavelet multi-resolution expansion of the arguments. The degree  $l$  of the polynomial is the highest sum of powers in any of its terms.

Noise is always present in physical systems and can be accommodated as part of the NARMAX model. In the present study we have initially assumed that the effects of the noise are small and can be neglected. In later studies the effects of any noise will be accommodated by fitting noise models as part of the identification procedure to ensure that unbiased models are obtained.

The NARMAX methodology breaks the modelling problem into the following steps:

- 1) Structure detection,
- 2) Parameter estimation,
- 3) Model validation,
- 4) Prediction and
- 5) Analysis.

These steps form an estimation toolkit that allows the user to build a concise mathematical description of the system. These procedures are now well established and have been used in many modelling domains [3].

A detailed procedure of how structure detection, parameter estimation and model validation is done is presented in [4]. A brief explanation of these steps is given below.

Any data set that we intend to model is first split in two sets (usually of equal size). We call the first the *estimation data set* and it is used to determine the model structure and parameters. The remaining data set is called the *validation data set* and it is used to validate the model.

The initial structure of the NARMAX polynomial is determined by the inputs  $\mathbf{u}$ , the output  $y$ , the input and output orders  $N_u$  and  $N_y$  respectively and the degree  $l$  of the polynomial.

The general rule in choosing suitable inputs for the model is that at least some of them should be causing the output. The number of initial terms of the NARMAX model polynomial can be very large depending on these variables, but not all of them are significant contributors to the computation of the output. In fact most terms, as explained below, are removed from the model equation in the process of refining the model. The final structure of the estimated NARMAX model will indicate any insignificant or redundant inputs.

Before any removal of model terms an equivalent *auxiliary model* is computed from the original NARMAX model. The model terms of the auxiliary model are orthogonal. This allows the computation of their associated parameters to be done in a sequential manner, which is computationally very efficient.

The calculation of the auxiliary model parameters and the refinement of the model's structure is an iterative process. Each iteration involves three steps:

- 1) Estimation of model parameters (using the estimation data set),
- 2) Model validation (using the validation data set) and
- 3) Removal of non-contributing terms.

After the model validation step, if there is no significant error between the model-predicted output and the actual output, non-contributing model terms are removed in order to reduce the size of the polynomial as much as possible. This is primarily done to increase the speed of computation of the model output during its use and also to avoid overfitting the model to the training data.

To determine the contribution of a model term to the output the Error Reduction Ratio (ERR) [4] is computed for each term. The ERR of a term is the percentage reduction in the total mean-squared error (i.e. the difference between model-predicted and true system output) as a result of including (in the model equation) the term under consideration. The bigger the ERR is, the more significant the term. Model terms with ERR under a certain threshold (usually around 0.05%) are removed from the model polynomial in the last step of each iteration during the refinement process.

In the following iteration, if the error is higher as a result of the last removal of model terms then these are re-inserted back into the model equation and the model is considered as final. Finally, the NARMAX model parameters are computed from the auxiliary model.

## II. EXPERIMENTAL PROCEDURE

The robot used in our experiments is the Magellan Pro autonomous mobile robot *Radix* (figure 1). The robot is equipped with 16 sonar, 16 infra-red and 16 tactile sensors distributed uniformly around its circumference. A SICK laser range finder is also present which scans the front semi-circle of the robot ( $[0^\circ, 180^\circ]$ ) with a radial resolution of  $1^\circ$  and a distance resolution of less than 1 centimetre. The robot also incorporates a video camera. In the work presented in this paper only the laser, sonar and infra-red sensors were used.

During experiments with *Radix*, the input from all its sensors (apart from the video camera), its position, orientation,

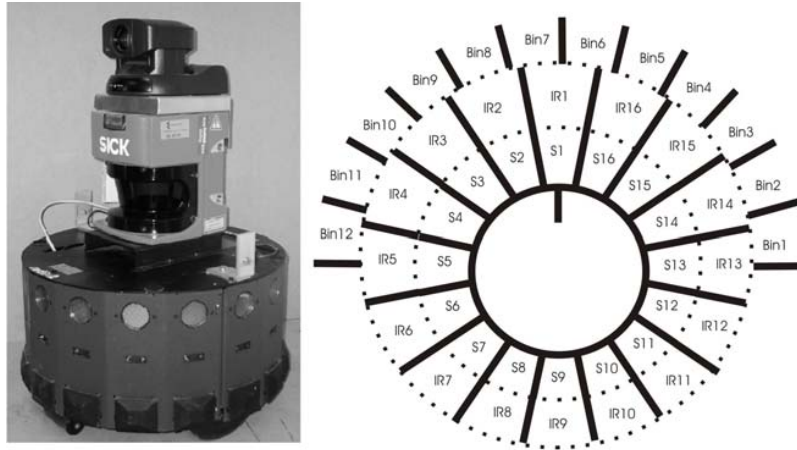


Fig. 1. Left: Radix, the Magellan Pro robot used in our experiments. Right: Location of each sonar and infrared sensor. The laser sensors have been averaged in twelve sectors of 15 degrees each (laser bins)

translational and rotational velocities are recorded every 250ms. Position and orientation of the robot are obtained by placing two point-targets on top of the robot and using an overhead video camera to track them continuously. After a logging session the sensor data from the robot and the position/orientation data from the tracking system are aligned using time as the common reference.

Experimental setups of the robot's static environment are built in a dedicated robot arena usually using carton boxes. Examples of such experimental setups are shown in figures II-A, 4, 5 and 6.

For the purposes of this paper we have conducted four separate route learning experiments. In each case we initially drove the robot manually several times through the specific route to be learned. The translational velocity of the robot was kept constant at  $0.07m/s$  while the rotational velocity was controlled by a human operator using a keyboard. During this time the robot's sensor values and rotational velocity were logged. The data collected were then used for the estimation and validation of the NARMAX model. Once the model was estimated it was put on the robot and executed in order to record a further set of data that was used to test the model's performance in each case.

In the sections that follow each of the four experiments is presented individually.

#### A. Route 1

The first route the robot had to learn is shown in figure II-A. Although this route looks quite simple, it is actually quite difficult to learn because of perceptual aliasing. I.e. the robot's sensor readings when it is in the middle of the route (labelled A in the graph) are very similar but half of the time the robot has to turn right, while the other half it has to turn left.

After manual control of the robot along the route for 1 hour, we identified the robot's movement using a NARMAX process. All the sonar and laser measurements were taken

into account (figure 3 (left)). The values delivered by the laser scanner were averaged in twelve sectors of 15 degrees each (laser bins), to obtain a twelve dimensional vector of laser-distances (figure 1 (right)). These laser bins, as well as the 16 sonar sensor values, were inverted so that large readings indicate close-by objects. Finally, as we can see in figure 3 (left), the sonar and laser readings at each instant were normalised by the minimum sonar and laser reading, respectively, at that instant. These minimum values are also introduced as inputs into the model, so that all the available information is provided. The reason of scaling the inputs is because we saw that the resulting models usually were more robust to small perturbations in the environment.

The parameters of the NARMAX model we obtained are:  $N_u = 0$ ,  $N_y = 0$ ,  $N_e = 0$ ,  $degree = 2$ . The initial model had 496 terms, but after the removal of the non-contributing terms, 70 remained (table I).

A visual, qualitative inspection of figure II-A is enough to see how closely the robot's trajectory under NARMAX model control resembles the trajectory obtained when *Radix* was manually driven. In order to compare the two trajectories quantitatively as well, we analysed the space occupancy of the two trajectories in  $(x, y)$  space, using the Mann-Whitney U-test [5] to determine whether the two trajectories occupy space significantly differently or not. It turns out that the difference between the distribution of values  $(x - \bar{x})$ , when the robot is under manual control, and the distribution of the same values under NARMAX model control, is not significant ( $p > 0.05$ ).  $\bar{x}$  represents the mean of the x-positions. Similarly, there is no significant difference between the distribution of values  $(y(t) - \bar{y})$  under manual control, and that of the same values under NARMAX model control, ( $p > 0.05$ ).

#### B. Route 2

Figure 4 shows the second route learnt by the robot through system identification. This time only the laser sensor was used

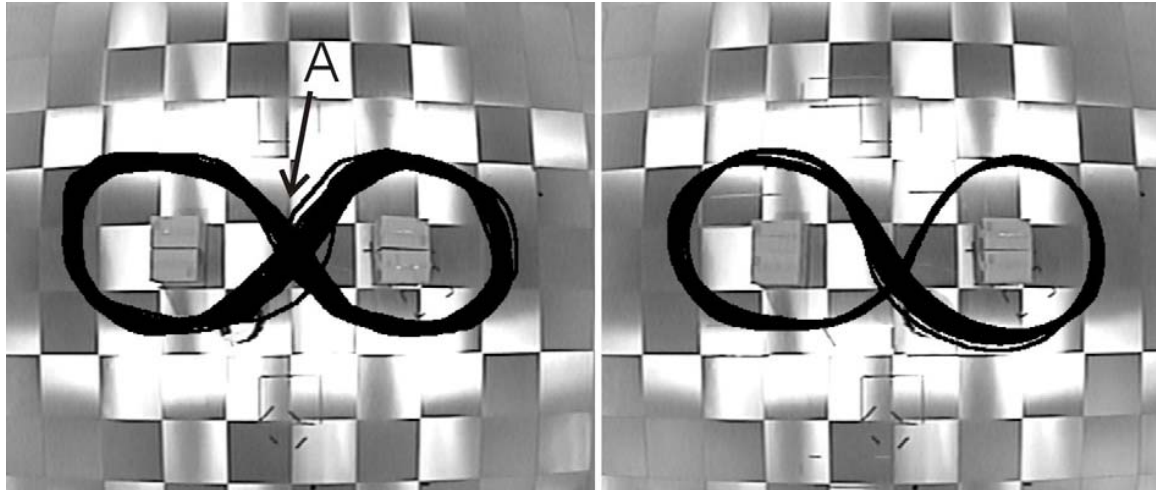


Fig. 2. Route 1. Robot trajectory under manual control , used as training data (left), and trajectory taken under model control (right). The model is shown in Table I

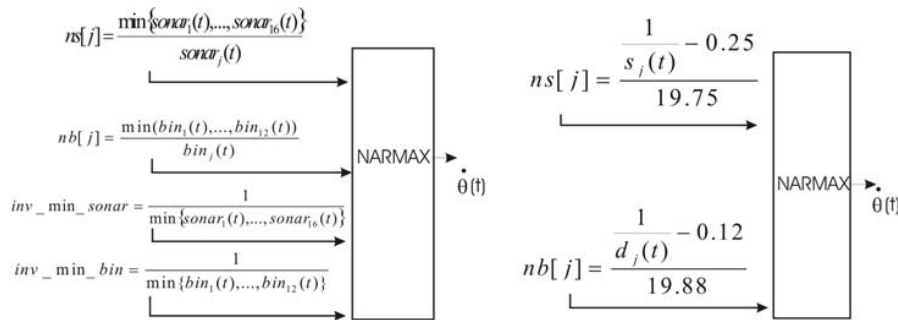


Fig. 3. Different ways of providing the laser and sonar information to the modelling strategy. The option shown on the left was used to model robot's behaviour in routes 1 and 2, while the option shown on the right was used in route 3

(pre-processed as it is shown in figure 3 (left) as this produced a better model. The characteristics of the NARMAX model obtained in this case are  $N_u = 0$ ,  $N_y = 0$ ,  $N_e = 0$ , and  $degree = 3$ . The initial model had 573 terms, but just 94 remained after the removal of non-contributing terms.

Figure 4 shows how closely the NARMAX model behaviour resembles the manual control behaviour. Statistical space occupancy tests along the  $x$  and  $y$  axes, using the Mann-Whitney U-test also confirm that there is no significant difference between the two trajectories ( $p > 0.05$ ).

C. Route 3

Figure 5 shows the third route the robot had to learn. There are many symmetries and two narrow passes the robot had to go through. To model the behaviour of the robot along this new route we used the NARMAX strategy once again. To obtain the model, the inverted and normalised laser bins were used (figure 3 (right)).

The best NARMAX model we found does not use past information at all ( $N_u = 0$ ,  $N_y = 0$ ), and the degree of the

polynomial function is two. The initial model had 91 terms, but after the removal of the non-contributing ones just 73 remained.

Figure 5 shows the trajectory of the robot during 1 hour when the NARMAX model was used to control it. Once again the model was able to properly learn the route in this environment. Apart from the qualitative assessment, this was confirmed using the Mann-Whitney U-test which indicated no significant difference in the space occupancy of the robot ( $p > 0.05$ ) between the trajectory of the robot during manual control and during NARMAX model control.

D. Route 4

Figure 6 shows a fourth route the robot had to learn. This is an episodic route, which means that the operation of the robot is naturally divided into independent episodes, each beginning with the robot at position labelled as A in figure 6 and ending in position labelled B.

To model the robot's behaviour along this episodic route the ARMAX modelling procedure was used. This is the linear

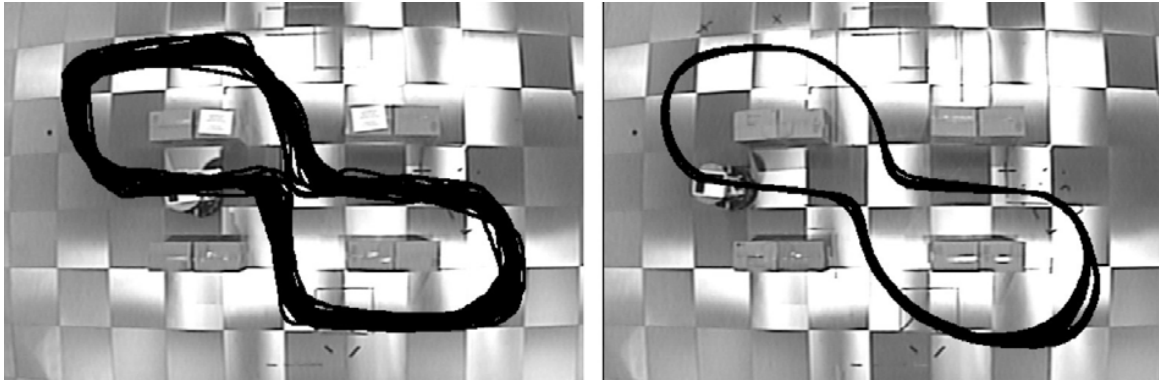


Fig. 4. Route 2. Robot trajectory under manual control (left), and, for comparison, under NARMAX model control (right)

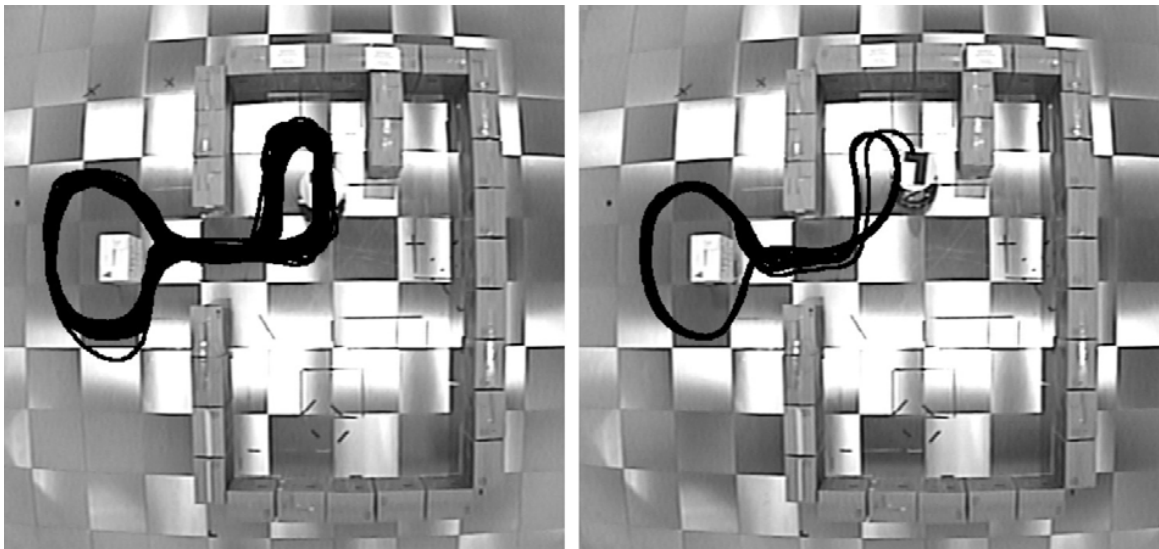


Fig. 5. Route 3. Robot trajectory under manual control (left), and, for comparison, trajectory of the robot under NARMAX model control (right)

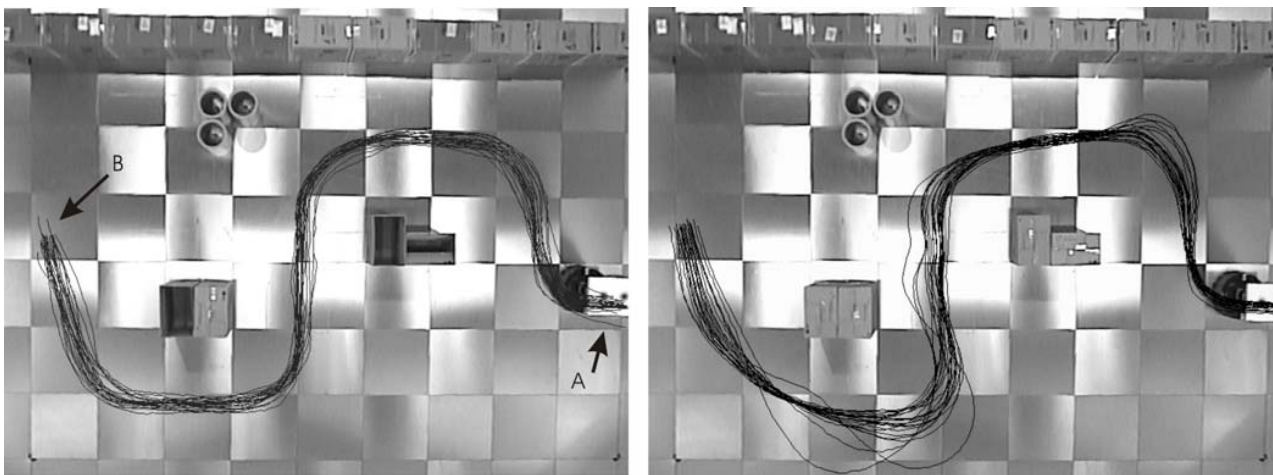


Fig. 6. Route 4. Left: Robot trajectories under manual control (22 runs, training data), this is an episodic route where each episode comprises the movement of the robot from A to B. Right: Trajectories taken under ARMAX model control (22 runs, test data)

TABLE I

NARMAX MODEL OF THE ANGULAR VELOCITY  $\dot{\theta}$  FOR THE ROUTE FOLLOWING BEHAVIOUR SHOWN IN FIGURE II-A.  $nb$ ,  $ns$ ,  $inv\_min\_sonar$ , AND  $inv\_min\_laser$ , ARE THE NORMALISED BINS, NORMALISED SONAR, INVERTED MINIMUM SONAR AND LASER READINGS, RESPECTIVELY, SHOWN IN FIGURE 3 (LEFT)

$$\begin{aligned} \dot{\theta}(t) = & +0.08 - 0.50 * nb[1] - 0.62 * nb[4] + 0.46 * nb[6] \\ & + 0.07 * nb[7] - 0.09 * nb[8] + 0.14 * nb[9] + 0.02 * nb[10] \\ & + 0.20 * nb[12] - 0.88 * ns[1] + 0.22 * ns[3] - 0.04 * ns[5] \\ & + 0.004 * ns[6] - 0.04 * ns[7] + 0.20 * ns[10] - 0.02 * ns[14] \\ & + 0.11 * ns[16] - 0.43 * inv\_min\_bin + 0.041 * nb[1]^2 \\ & + 0.02 * nb[2]^2 - 0.06 * nb[3]^2 + 0.53 * nb[4]^2 - 0.44 * nb[6]^2 \\ & + 0.01 * nb[9]^2 - 8.70 * inv\_min\_bin^2 - 0.07 * nb[1] * nb[2] \\ & + 0.07 * nb[1] * nb[3] + 0.44 * nb[1] * nb[4] + 0.40 * nb[1] * nb[10] \\ & - 0.24 * nb[1] * nb[11] + 0.83 * nb[1] * ns[1] + 0.09 * nb[1] * ns[4] \\ & - 0.79 * nb[1] * ns[11] - 0.04 * nb[1] * ns[13] \\ & + 0.08 * nb[1] * inv\_min\_sonar + 3.58 * nb[1] * inv\_min\_bin \\ & + 0.36 * nb[2] * nb[4] - 0.73 * nb[2] * nb[9] - 0.05 * nb[2] * nb[12] \\ & + 0.04 * nb[2] * ns[13] + 0.63 * nb[3] * nb[8] - 0.28 * nb[3] * nb[12] \\ & + 0.11 * nb[3] * ns[3] - 0.48 * nb[4] * nb[10] - 0.27 * nb[5] * nb[8] \\ & + 0.11 * nb[5] * ns[1] + 0.26 * nb[6] * nb[8] + 0.02 * nb[7] * ns[5] \\ & + 0.15 * nb[7] * ns[6] - 0.18 * nb[7] * ns[12] - 0.17 * nb[8] * nb[10] \\ & + 0.04 * nb[8] * ns[5] - 0.10 * nb[10] * ns[10] + 0.05 * nb[10] * ns[12] \\ & + 0.03 * nb[12] * ns[10] + 0.06 * nb[12] * ns[11] \\ & + 0.01 * nb[12] * ns[16] + 2.68 * ns[1] * ns[6] - 0.30 * ns[1] * ns[11] \\ & - 1.99 * ns[2] * ns[6] + 3.91 * ns[2] * inv\_min\_bin \\ & + 0.13 * ns[3] * ns[5] - 1.27 * ns[3] * ns[6] \\ & - 1.85 * ns[3] * inv\_min\_bin + 0.05 * ns[4] * ns[11] \\ & - 0.13 * ns[4] * inv\_min\_sonar - 0.23 * ns[6] * ns[10] \\ & + 0.89 * ns[6] * ns[11] + 0.08 * ns[15] * ns[16] \\ & + 5.06 * ns[16] * inv\_min\_sonar \end{aligned}$$

polynomial equivalent of NARMAX. I.e. the degree of the polynomial is 1. In order to avoid making assumptions about the relevance of specific sensor signals, all sensor modalities (sonar, infra-red and laser sensors) were presented as possible inputs to the modelling procedure. The values delivered by the laser scanner were averaged in 9 sectors of 20 degree each (i.e. 9 laser bins). No other pre-processing was done on the inputs. The reason why the pre-processing is a bit different from the one used in the previous routes, is just because the person who carried out the experiment was different. This can be interpreted as a blind test which proves that the modelling process presented here does not depend on any clever pre-processing the input space. This was merely done in the previous experiments in order to assist the readability of the model.

In this experiment the regression order of the output ( $N_y$ ) was 0 and that of the inputs ( $N_u$ ) was 8. This means that the model uses a two-second history of sensor inputs.

As shown in figure 6, this model has successfully learned the route from A to B. This is again confirmed using statistical analysis.

### III. CONCLUSIONS AND FUTURE WORK

This paper introduces a novel mechanism to program a route following reactive controller through system identification. To obtain the controller, the robot is first operated under human supervision, making it follow a desired route in a specific environment. Once the training data are acquired in this way, we use the NARMAX modelling approach to obtain a model

which identifies the coupling between sensory perceptions and motor responses as a non-linear or linear polynomial, respectively. This model is then used to control the robot when it moves autonomously.

Our results confirm that there are cases where the methodology described in this paper is viable. In particular the models we obtained were able to make the robot move properly along the desired routes regardless of the significant perceptual aliasing present in most situations.

An interesting feature of this modelling methodology is the fact that it is able to determine automatically which of the inputs (or combinations of these) are significant contributors to the output. Also, past input and output values can be easily incorporated in the model with very little effort from the roboticist's side. This fact makes the design process much simpler than using other learning paradigms. In the case of artificial neural networks, for example, the nature of the problem would require the use of recurrent architectures.

Finally, it is also very important to consider that a control program obtained through the procedure we propose, is not just an "existence proof", in the sense that it shows that the robot can achieve a particular behaviour under specific environmental conditions, but it also allows the mathematical analysis of the main factors involved in the robot's operation in the environment. Thus, the sensitivity analysis of the models obtained, for example, would allow to answer questions related with the importance of every sensor with respect to the robot's global behaviour, or even to analyse the effect of small perturbations in the environment upon the robot's operation [6], [1]. The mathematical analysis of these models, and how it can be used to formulate new and testable hypotheses that might lead to a new stage in the experimentation and design of a robot controller, is part of the ongoing research at the universities of Sheffield and Essex.

### Acknowledgements

The RobotMODIC project is supported by the Engineering and Physical Sciences Research Council (grant GR/S30955/01). R. Iglesias is supported through grants PGIDIT04TIC206011PR, TIC2003-09400-C04-03 and TIN2005-03844.

### REFERENCES

- [1] Theocharis Kyriacou Ulrich Nehmzow, Roberto Iglesias and Stephen A. Billings, "Robot learning through task identification," *J. Robotics and Autonomous Systems.*, Accepted.
- [2] T. Kyriacou, U. Nehmzow, R. Iglesias, and S. Billings, "Cross-platform programming through system identification," in *Proceedings of TAROS 2005*, London, 2005.
- [3] S. Billings and S. Chen, "The determination of multivariable nonlinear models for dynamical systems," in *Neural Network Systems, Techniques and Applications*, C. Leonides, Ed., pp. 231–278. Academic press, 1998.
- [4] M. Korenberg, S. Billings, Y. P. Liu, and P. J. McIlroy, "Orthogonal parameter estimation algorithm for non-linear stochastic systems," *International Journal of Control*, vol. 48, pp. 193–210, 1988.
- [5] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, Iowa State Press, 8th edition, 1989.
- [6] I.M. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Mathematical Modelling and Computational Experiment*, vol. 1, pp. 407–414, 1993.