Monograph:
Inglesias, R., Nehmzow, U. and Billings, S.A. (2008) Model Identification and Model Analysis in Robot Training. Research Report. ACSE Research Report 966 . Department of Automatic Control and Systems Engineering

---

Model Identification and Model Analysis in Robot Training

R Inglesias [*], U Nehmzow[#], S A Billings

***Electronics and Computer Science, University of Santiago de Compostela, Spain**
#Dept Computing and Electronic Systems, University of Essex

Department of Automatic Control and Systems Engineering
The University of Sheffield, Sheffield, S1 3JD, UK

Research Report No. 966

January 2008

Model Identification and Model Analysis in Robot Training

R. Iglesias[2], U. Nehmzow[1] and S.A. Billings[3]

[1]Computing and Electronic Systems, University of Essex, UK.
[2]Electronics & Computer Science, Univ. of Santiago de Compostela, Spain.
[3]Automatic Control and Systems Engineering, University of Sheffield, UK.

Abstract

Robot training is a fast and efficient method of obtaining robot control code. Many current machine learning paradigms used for this purpose, however, result in opaque models that are difficult, if not impossible to analyse, which is an impediment in safety-critical applications or application scenarios where humans and robots occupy the same workspace.

In experiments with a *Magellan Pro* mobile robot we demonstrate that it is possible to obtain *transparent* models of sensor-motor couplings that are amenable to subsequent analysis, and how such analysis can be used to refine and tune the models *post hoc*.

1 Introduction: Robot Training

Sensor-motor couplings form the back-bone of most mobile robot control tasks, and often need to be implemented fast, efficiently, and reliably. Machine-learning techniques, such as artificial neural networks are commonly used to obtain the desired sensor-motor competences. However, although these methods speed up the development of a reactive controller significantly, most of them produce opaque models that cannot be used to investigate and "understand" the characteristics of the robot's behaviour further.

In [Nehmzow et al., 2006] we presented a novel procedure to program a robot controller, based on system identification techniques. Instead of refining an initial approximation of the desired control code through a process of iterative refinement by trial an error, the robot training procedure we proposed *identifies* the motion of a manually, "perfectly" driven robot, and subsequently uses the result of the identification process to achieve autonomous robot operation. Through the use of a system identification approach the behaviour of the robot is modelled through a polynomial representation that is easily and accurately transferable to any robot platform with similar sensor configuration [Kyriacou et al., 2005]. Moreover, this polynomial representation can be analysed to understand the main aspects involved in robot behaviour: we can for instance identify the most relevant hardware components of the robot (e.g. sensors) [Iglesias et al., 2005, Nehmzow et al., 2006], or predict the robot's response to particular inputs [Kyriacou et al., 2006].

The robot-training process we proposed works in two stages: first, the robot is driven under manual control demonstrating the target behaviour. While the

robot is being manually moved, sensor readings and the robot actions are logged. In a second stage, system identification techniques like ARMAX [Eykhoff, 1981] or NARMAX [Chen and Billings, 1989] are applied to model the relationship between sensor readings, i.e. perception and actuator signals, i.e. action. These ARMAX and NARMAX models are transparent (i.e. expressed as a mathematical equation) and can therefore be formally analysed, as well as used in place of "traditional" robot control code.

In this paper we focus our attention on how the mathematical analysis of NARMAX models can be used to understand the robot's control actions, to formulate hypotheses, and to correct or improve the robot's behaviour. One main objective behind this approach is to avoid trial-and-error refinement of robot code. Instead, we seek to obtain a reliable design process, where program design decisions are based on the mathematical analysis of the model which describes the robot's behaviour. We demonstrate this procedure for different robot-behaviours.

2 The NARMAX Modelling Procedure

To obtain the desired sensor-motor couplings, we used the nonlinear system identification of Narmax (nonlinear, auto regressive moving average models with exogenous inputs). Due to space limits we can only provide a brief description of the Narmax modelling strategy, nevertheless this approach is discussed in detail in [Chen and Billings, 1989], and examples of robotics applications are given in our previous publications [Analytical and Cognitive Robotics Group, 2007].

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:

$$
\begin{aligned}
y(n) \;=\; & f(u_1(n), u_1(n-1), u_1(n-2), \cdots, u_1(n-N_u), \\
& u_1(n)^2, u_1(n-1)^2, u_1(n-2)^2, \cdots, u_1(n-N_u)^2, \\
& \cdots, \\
& u_1(n)^l, u_1(n-1)^l, u_1(n-2)^l, \cdots, u_1(n-N_u)^l, \\
& u_2(n), u_2(n-1), u_2(n-2), \cdots, u_2(n-N_u), \\
& u_2(n)^2, u_2(n-1)^2, u_2(n-2)^2, \cdots, u_2(n-N_u)^2, \\
& \cdots, \\
& u_2(n)^l, u_2(n-1)^l, u_2(n-2)^l, \cdots, u_2(n-N_u)^l, \\
& \cdots, \\
& \cdots, \\
& u_d(n), u_d(n-1), u_d(n-2), \cdots, u_d(n-N_u), \\
& u_d(n)^2, u_d(n-1)^2, u_d(n-2)^2, \cdots, u_d(n-N_u)^2,
\end{aligned}
$$

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

were $y(n)$ and $\boldsymbol{u}(n)$ 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. $f()$ 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.

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: basically the model parameters are determined trying to minimise the difference (mean-squared error) between the model predicted output and the actual one. The remaining data set is called the *validation data set* and it is used to validate the model.

The structure of the NARMAX polynomial is determined by the inputs $\boldsymbol{u}$, the output y, the input and output orders N_u and N_y respectively and the degree l of the polynomial. The problem is that the number of initial terms of the NARMAX model polynomial can be very large depending on these variables, but not all of these terms are significant contributors to the computation of the output. In order to remove the non relevant terms, the Error Reduction Ratio (ERR) [Korenberg et al., 1988] 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.

3 Route Learning by Demonstration

We applied our robot training strategy to program a reactive route following controller (figure 1). Although this route looks quite simple, it is actually quite difficult to learn due to the lack of landmarks in the environment. The sensor readings when the robot is in the middle of the route (labelled A in figure 1) are very similar but half of the time the robot has to turn right, while the other half it has to turn left. In order to learn this route a *Magellan Pro* Robot was first steered for 1 hour along the desired route by a human operator (figure 1,left). During this stage sensor perceptions (figure 2), position, transitional and rotational velocities were recorded every 250 ms.

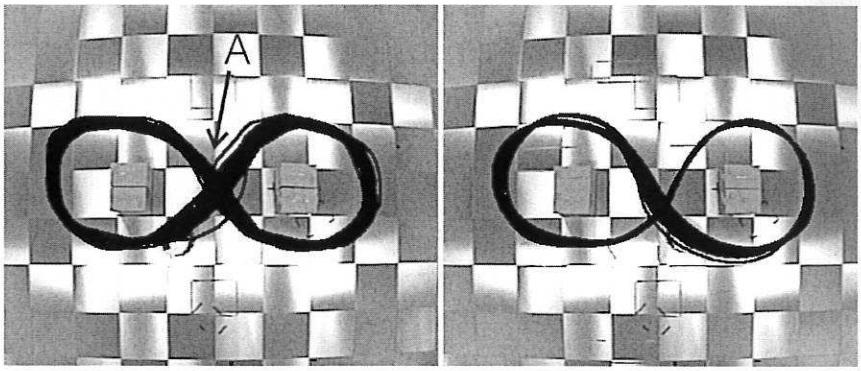

Figure 1: LEFT: ROBOT TRAJECTORY UNDER MANUAL CONTROL, USED TO OBTAIN TRAINING DATA. RIGHT: TRAJECTORY TAKEN UNDER CONTROL OF THE OBTAINED MODEL GIVEN IN TABLE 1.

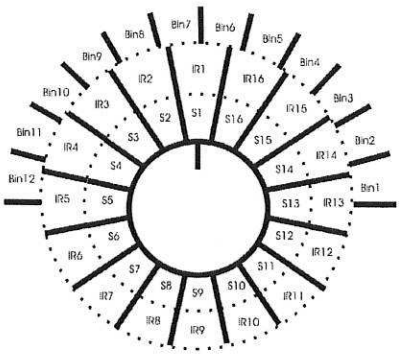

Figure 2: LOCATION OF EACH SONAR AND INFRARED SENSOR IN THE MAGELLAN PRO ROBOT WE USED IN OUR EXPERIMENTS. THE LASER SENSORS HAVE BEEN AVERAGED IN TWELVE SECTORS OF 15 DEGREES EACH (LASER BINS).

Having logged speeds and perceptions, we identified the robot's movement using the NARMAX process, taking all sonar and laser measurements as inputs to the modelling process (figure 3). Laser ranges were averaged in twelve sectors of 15 degrees each (laser bins), resulting in a twelve-dimensional vector of laser-distances. Both laser bins and the 16 sonar sensor values were inverted and normalised, so that large readings indicate close-by objects. The resulting NARMAX model is shown in table 1. The model was then used to control the robot directly (figure 1, right).

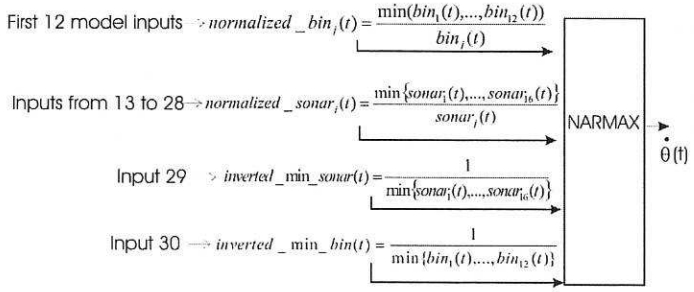

Figure 3: INPUTS USED TO MODEL ROBOT'S BEHAVIOUR IN THE ROUTE SHOWN IN FIGURE 1.

4 Behaviour Refinement through off-line Model Analysis

Robot-environment interaction is strongly influenced by the environment in which the robot operates, and it is a common occurrence in mobile robotics that control code ceases to function correctly once the environment changes. To investigate the susceptibility of our Narmax model to such changes, we modified the environment (by introducing or removing boxes, ladders *etc.*) until the model was no longer able to control the robot correctly.

Instead of trying to "guess" what was wrong, we decided to *analyse* the models to see the reasons behind the undesired behaviour, carrying out two different analyses: i) we computed the sensitivity of our control model with respect to each one of the inputs to see which inputs are the most relevant for the robot-behaviour, and ii) we used the partial derivatives of our model with respect to the identified most relevant inputs to analyse the stability of the model.

4.1 Sensitivity Analysis

To determine which model-inputs contribute the most to the output (angular velocity), we used three different tests: partial derivative analysis, Sobol's mechanism [Sobol, 1993], and mutual information.

$$
\begin{aligned}
\dot{\theta}(t) = {} & +0.08 - 0.50 * I_1 - 0.62 * I_4 + 0.46 * I_6 \\
& +0.07 * I_7 - 0.09 * I_8 + 0.14 * I_9 + 0.02 * I_{10} \\
& +0.20 * I_{12} - 0.88 * I_{13} + 0.22 * I_{15} - 0.04 * I_{17} \\
& +0.004 * I_{18} - 0.04 * I_{19} + 0.20 * I_{22} \\
& -0.02 * I_{26} + 0.11 * I_{28} - 0.43 * I_{30} \\
& +0.041 * I_1^2 + 0.02 * I_2^2 - 0.06 * I_3^2 \\
& +0.53 * I_4^2 - 0.44 * I_6^2 + 0.01 * I_9^2 \\
& -8.70 * I_{30}^2 - 0.07 * I_1 * I_2 \\
& +0.07 * I_1 * I_3 + 0.44 * I_1 * I_4 \\
& +0.40 * I_1 * I_{10} - 0.24 * I_1 * I_{11} \\
& +0.83 * I_1 * I_{13} + 0.09 * I_1 * I_{16} \\
& -0.79 * I_1 * I_{23} - 0.04 * I_1 * I_{25} \\
& +0.08 * I_1 * I_{29} + 3.58 * I_1 * I_{30} \\
& +0.36 * I_2 * I_4 - 0.73 * I_2 * I_9 \\
& -0.05 * I_2 * I_{12} + 0.04 * I_2 * I_{25} \\
& +0.63 * I_3 * I_8 - 0.28 * I_3 * I_{12} \\
& +0.11 * I_3 * I_{15} - 0.48 * I_4 * I_{10} \\
& -0.27 * I_5 * I_8 + 0.11 * I_5 * I_{13} \\
& +0.26 * I_6 * I_8 + 0.02 * I_7 * I_{17} \\
& +0.15 * I_7 * I_{18} - 0.18 * I_7 * I_{24} \\
& -0.17 * I_8 * I_{10} + 0.03 * I_8 * I_{17} \\
& -0.10 * I_{10} * I_{22} + 0.05 * I_{10} * I_{24} \\
& +0.03 * I_{12} * I_{22} + 0.06 * I_{12} * I_{23} \\
& +0.01 * I_{12} * I_{28} + 2.68 * I_{13} * I_{18} \\
& -0.30 * I_{13} * I_{23} - 1.99 * I_{14} * I_{18} \\
& +3.91 * I_{14} * I_{30} + 0.13 * I_{15} * I_{17} \\
& -1.27 * I_{15} * I_{18} - 1.85 * I_{15} * I_{30} \\
& +0.05 * I_{16} * I_{23} - 0.13 * I_{16} * I_{29} \\
& -0.23 * I_{18} * I_{22} + 0.89 * I_{18} * I_{23} \\
& +0.08 * I_{27} * I_{28} + 5.06 * I_{28} * I_{29}
\end{aligned}
$$

Table 1: NARMAX MODEL OF THE ANGULAR VELOCITY $\dot{\theta}$ FOR THE ROUTE FOLLOWING BEHAVIOUR SHOWN IN FIGURE 1. $I_1 \ldots I_{30}$, ARE THE MODEL INPUTS: THE NORMALISED LASER BINS, NORMALISED SONAR, INVERTED MINIMUM SONAR AND LASER READINGS, SHOWN IN FIGURES 2 AND 3.

4.1.1 Partial Derivative Analysis

Using Taylor's theorem [Apostol, 1974], it is possible to estimate the change in the angular velocity of the robot due to changes in input sensor readings (eq 1).

$$
\begin{aligned}
\Delta\dot{\theta} = {} & \sum_{i=1}^{n} \frac{\partial\dot{\theta}}{\partial I_i}\Delta I_i + \frac{1}{2!}\sum_{j=1}^{n}\sum_{i=1}^{n} \frac{\partial^2\dot{\theta}}{\partial I_j \partial I_i}\Delta I_j \Delta I_i + \\
& + \frac{1}{3!}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} \frac{\partial^3\dot{\theta}}{\partial I_k \partial I_j \partial I_i}\Delta I_k \Delta I_j \Delta I_i + \ldots,
\end{aligned} \tag{1}
$$

where $\dot{\theta}$ is the turning speed of the robot, n the number of input signals in the model (30 in our case) and I_i and I_j represent model inputs $1 \ldots n$ (figure 3). For the model shown in table 1, the third order derivatives are all zero, and the contribution of the 97 non-zero second order derivatives is very small, which allows us to rewrite equation 1 in the simplified form of equation 2.

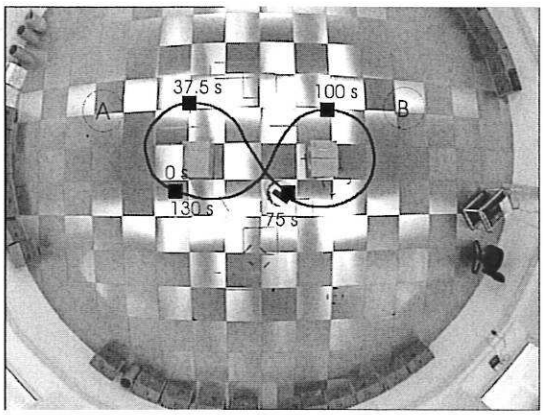

Figure 4: FIRST FULL TRAVERSAL OF THE DESIRED ROUTE, TAKING 130 SECONDS TOTAL TRAVELLING TIME.

$$\dot{\theta}_{t+1} \approx \dot{\theta}_t + \sum_{i=1}^{30} \frac{\partial \dot{\theta}}{\partial I_i} \Delta I_i \,. \tag{2}$$

Using equation 2, we can estimate the influence of each model input upon the robot's steering speed. To estimate these influence values, we computed for every model-input the difference between the angular velocity $\dot{\theta}_t$ and the predicted angular velocity when the input under consideration is removed from equation 2. Figure 5 shows the average influence of each model input for the first full traversal of the route (figure 4). The most relevant inputs turn out to be I_{13}, I_{14}, I_{15}, I_{23},I_{28}, I_{29} and I_{30}.

4.1.2 Sensitivity Analysis: Sobol Indices

We also applied the mechanism proposed by I. M. Sobol [Sobol, 1993] [Saltelli, 2002] to estimate the sensitivity of the model shown in table 1 with respect to each of its inputs. This analysis determines which input parameters contribute most to the model output, and which parameters are insignificant and might therefore be eliminated from the model.

We assume that a mathematical model is described by $y = f(\vec{x})$, where $\vec{x}$ represents a vector of n independent random variables defined in a unit n-dimensional cube. We'll also assume that the joint probability density function of the input is $p(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} p_i(x_i)$. The total sensitivity of $f(\vec{x})$ with respect to each one of the different variables x_i, is calculated as the percentage of the total variance of $f(\vec{x})$ which is due to the variance in x_i.

Figure 6 shows the sensitivities computed for each model input. I_{13}, I_{14}, I_{15}, I_{18}, I_{28} and I_{29} are the most relevant inputs. This result agrees with the

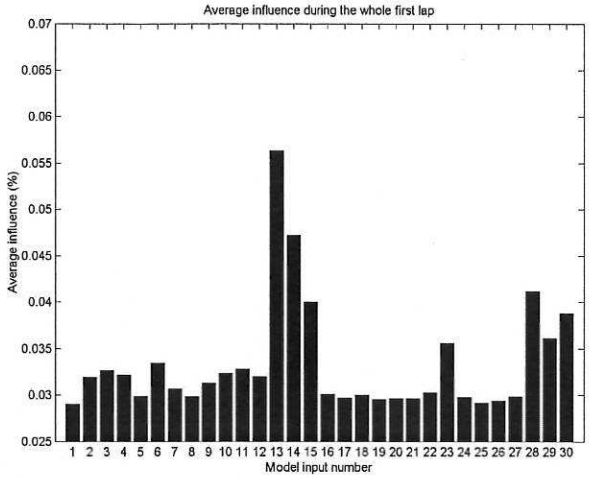

Figure 5: AVERAGE INFLUENCE (ESTIMATED THROUGH PARTIAL DERIVATIVES) OF EACH MODEL-INPUT.

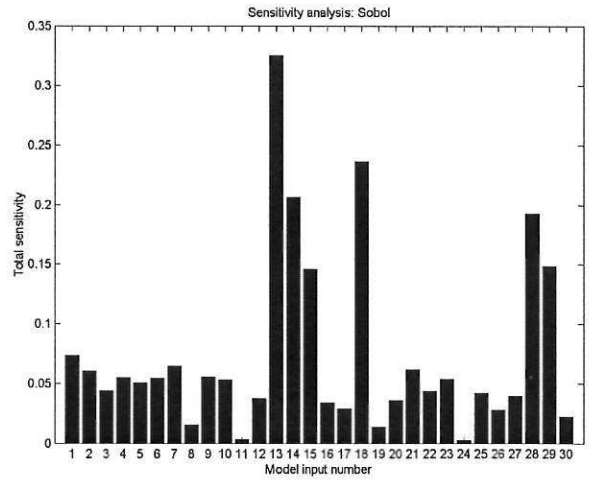

Figure 6: TOTAL SENSITIVITY OF THE MODEL WITH RESPECT TO EACH OF ITS INPUTS, CALCULATED USING SOBOL'S METHOD.

conclusions we drew before from the partial derivative analysis (compare figure 6 with figure 5).

4.1.3 Mutual Information in Sensitivity Analysis

Finally, we estimated the sensitivity of the model with respect to each input by using the mutual information I_m between model input u and model output y, given in equation 3.

$$I_m = H(u) + H(y) - H(u, y), \tag{3}$$

with $H(m) = -\sum_k p(m_k) ln\, p(m_k)$ and
$H(m, n) = -\sum_{k,l} p(m_k, n_l) ln\, p(m_k, n_l)$. $p(m_k)$ is the probability that the value of variable m falls into bin k, and $p(m_k, n_l)$ is the probability that variable m falls into bin k and variable n falls into bin l.

Using a Monte Carlo simulation, we estimate the sensitivity of the model with respect to input u_i by generating a large number of random input vectors $\vec{u}$, where only component u_i is kept constant, and computing the mutual information between input u_i and output y. As the mutual information can be interpreted as a measure of how much information y contains about u_i, it will be particularly high for components u_i that influence the model output y strongly (in other words, inputs u_i that are "important").

Figure 7 gives the results, inputs I_{13}, I_{14}, I_{15}, I_{18}, I_{28}, and I_{29} turn out to be the most relevant — a good agreement with the results obtained in figures 5 and 6.

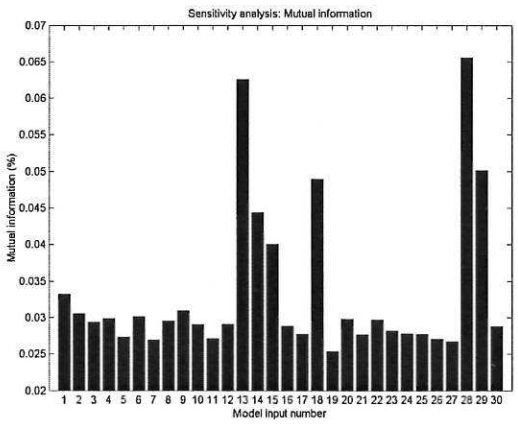

Figure 7: SENSITIVITY ANALYSIS: MUTUAL INFORMATION BETWEEN THE MODEL-OUTPUT AND MODEL-INPUTS $u_1 \ldots u_{30}$.

4.2 Stability Analysis

The last aspect we analysed is how much the angular velocity of the robot would change if one of the relevant sensors became noisy or if there were changes in the environment: we calculated partial derivatives of the angular velocity model with respect to the most relevant model inputs along the robot's trajectory.

Figures 8 and 9 show the values of the partial derivatives of the model with respect to inputs I_{15} and I_{18} (sonar sensor 3 and 6, respectively). As we can see, the partial derivative with respect to the third ultrasound sensor is quite often high during the first lap of the robot in the environment; this means that if any obstacle is placed at the left side of the robot when it is turning to the right (region A in figure 4), the robot-behaviour might be affected.

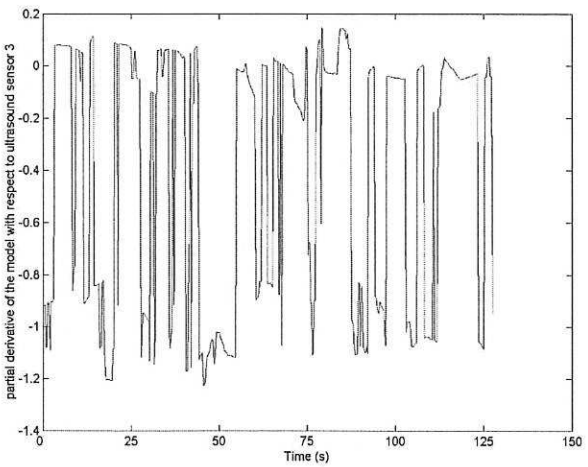

Figure 8: VALUE OF THE PARTIAL DERIVATIVE OF THE CONTROL MODEL WITH RESPECT TO INPUT 15 (SONAR 3) DURING THE FIRST LAP OF THE ROBOT IN THE ENVIRONMENT SHOWN IN FIGURE 4.

Regarding sonar 6 (figure 9), noisy readings would not be critical during most of the trajectory. If we consider figure 9 and the time it takes the robot to reach every part of the route (figure 4), we can see that the partial derivative is only high when the robot is between the two boxes in the environment. This is in good agreement with the results shown in figure 5.

If we now consider sonar 16 (model input I_{28}), if a box was placed at, say, location B (figure 4), the analysis given in figure 10 shows that that would affect the robot's steering speed considerably — there are numerous high values of $\frac{\partial \dot{\Theta}}{\partial I_{28}}$ between seconds 75 and 100.

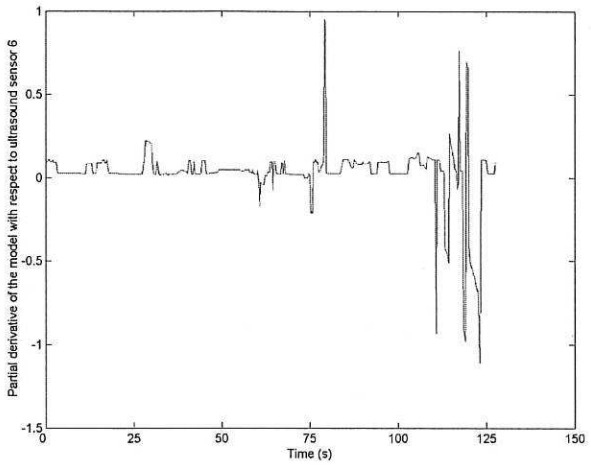

Figure 9: VALUE OF THE PARTIAL DERIVATIVE OF THE MODEL WITH RESPECT TO INPUT 18 (SONAR 6) DURING THE FIRST LAP OF THE ROBOT IN THE ENVIRONMENT SHOWN IN FIGURE 4.

5 Refining the Narmax Control Model

The analysis in section 4.2 revealed that the model is affected by fluctuations especially on sonars 3 and 16. We therefore decided to obtain a refined Narmax model, whose inputs are "myopic" sensor readings that are deliberately set to zero when they detect obstacles more than 2 m away. The model inputs I_i we used are given in equation 4

$$I_i = \begin{cases} \delta_i \cdot \frac{2.0}{bin(i)} & 0 \leq i \leq 12 \\ \delta_i \cdot \frac{2.0}{sonar(i)} & 12 \leq i \leq 28, \end{cases} \tag{4}$$

with

$$\delta_i = \begin{cases} 1 & \text{if } bin(i) < 2.0 \text{ or } sonar(i) < 2.0 \\ 0 & otherwise \end{cases}$$

We obtained a revised model of 100 terms, the robot's trajectory under control of this model is given in figure 11.

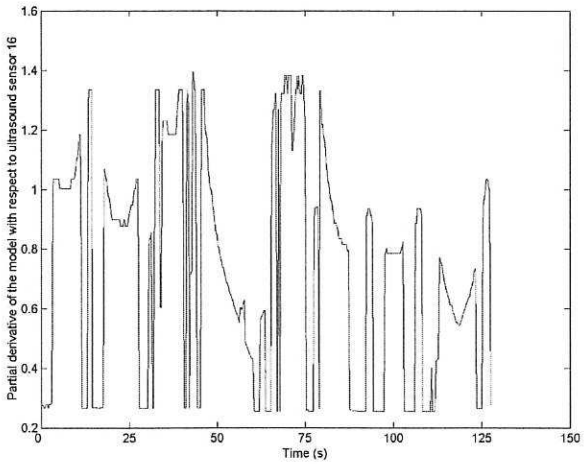

Figure 10: VALUE OF THE PARTIAL DERIVATIVE OF THE MODEL WITH RESPECT TO SONAR 16 ALONG THE FIRST LAP OF THE ROBOT IN THE ENVIRONMENT SHOWN IN FIGURE 4.

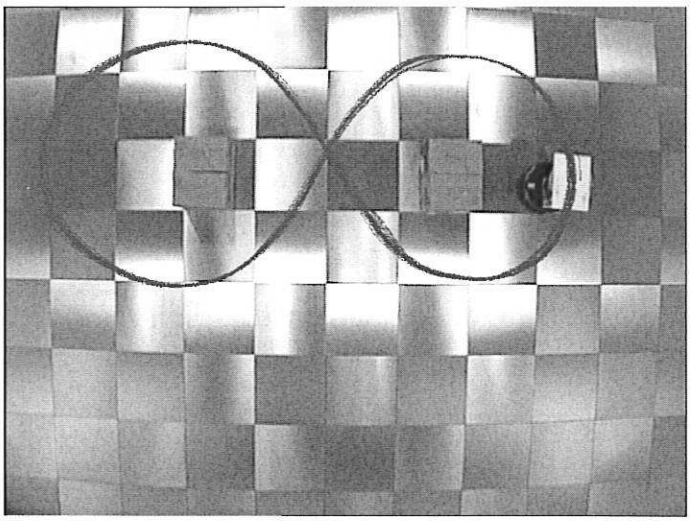

Figure 11: ROBOT TRAJECTORY UNDER THE CONTROL OF THE NEW, REFINED NARMAX MODEL (SECTION 5).

6 Sensitivity Analysis using Mutual Information

We have applied three different strategies to carry out a sensitivity analysis, and consider mutual information to be the best amongst them, due to its low computational cost and intuitive meaning: It measures the information obtained about a particular model input, given a particular model output. If the model output, y, is completely independent of a particular model input, u, then $H(u, y) = H(u) + H(y)$ and, therefore, $I_m = 0$. To demonstrate the utility of mutual information in sensitivity analysis, we will apply it to further models in this section.

6.1 Door traversal behaviour

We have first considered a model that identifies the behaviour of a manually driven robot across the door shown in figure 12. The model is given in table 2.

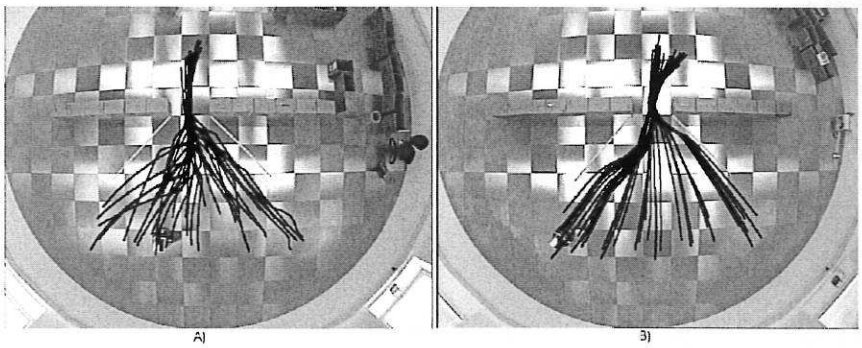

Figure 12: A) ROBOT TRAJECTORIES UNDER MANUAL CONTROL (39 RUNS, TRAINING DATA). B) TRAJECTORIES TAKEN UNDER MODEL CONTROL (41 RUNS, TEST DATA). THE WHITE LINES ON THE FLOOR WERE USED TO AID THE HUMAN OPERATOR IN SELECTING START LOCATIONS, THEY WERE INVISIBLE TO THE ROBOT.

When we apply mutual information to analyse this model, we get the results shown in figure 13. The relevant model inputs are the right side laser readings (laser bins 1,4, and 6, figure 2), and the rear ultrasound sensor (sensor 21, figure 2). These results agree with the ones published in [Iglesias et al., 2006], where a new Narmax model using only the sensor information coming from the right side of the robot was enough to identify the door traversal behaviour, and it was also able to properly control the movement of the real robot. The relevance of the rear ultrasound sensor 21, is due to a very interesting phenomena that can be observed in figure 12: In the door traversal under human control, the human operator moved the robot towards the centre of the door when the robot was still far from the opening. As the human operator gained experience, he was able to execute more efficient motions, nearer the door.

Table 2: NARMAX MODEL OF THE STEERING DIRECTION AND VELOCITY $\dot{\theta}$ OF THE ROBOT FOR THE DOOR TRAVERSAL BEHAVIOUR. THE SONAR READINGS ARE REPRESENTED AS $s_1, \cdots, s_{16}$, AND THE 12 LASER BINS ARE $d_1, \cdots, d_{12}$.

$$
\begin{aligned}
&\dot{\theta}(t) = 0.272 + 0.142 * (1/d_1(t)) - 0.470 * (1/d_3(t)) \\
&-0.070 * (1/d_4(t)) - 0.347 * (1/d_6(t)) + 0.157 * (1/d_8(t)) \\
&+0.091 * (1/d_9(t)) - 1.070 * (1/s_9(t)) - 0.115 * (1/s_{12}(t)) \\
&+0.130 * (1/d_3(t))^2 - 0.166 * (1/d_8(t))^2 + 0.183 * (1/s_9(t))^2 \\
&+0.081 * (1/(d_1(t) * d_3(t))) - 0.098 * (1/(d_1(t) * d_4(t))) \\
&-0.382 * (1/(d_1(t) * d_5(t))) - 0.204 * (1/(d_1(t) * d_6(t))) \\
&-0.049 * (1/(d_1(t) * d_8(t))) - 0.078 * (1/(d_1(t) * s_8(t))) \\
&+0.060 * (1/(d_2(t) * s_7(t))) + 0.300 * (1/(d_3(t) * d_5(t))) \\
&+0.037 * (1/(d_3(t) * s_5(t))) + 0.209 * (1/(d_3(t) * s_{12})) \\
&+1.014 * (1/(d_4(t) * d_6(t))) + 0.061 * (1/(d_4(t) * s_4(t))) \\
&+0.273 * (1/(d_4(t) * s_{12}(t))) - 0.536 * (1/(d_5(t) * d_6(t))) \\
&+0.230 * (1/(d_5(t) * d_7(t))) - 0.503 * (1/(d_6(t) * d_9(t))) \\
&+2.516 * (1/(d_6(t) * s_9(t))) - 0.067 * (1/(d_6(t) * s_{13}(t))) \\
&-0.009 * (1/(d_7(t) * s_{15}(t))) + 0.086 * (1/(d_8(t) * s_3(t))) \\
&-0.038 * (1/(d_8(t) * s_6(t))) - 0.060 * (1/(d_9(t) * s_4(t))) \\
&-0.067 * (1/(d_{10}(t) * d_{12}(t))) - 0.040 * (1/(d_{10}(t) * s_{12}(t))) \\
&+0.059 * (1/(d_{11}(t) * s_1(t))) - 0.045 * (1/(d_{12}(t) * s_7(t)))
\end{aligned}
$$

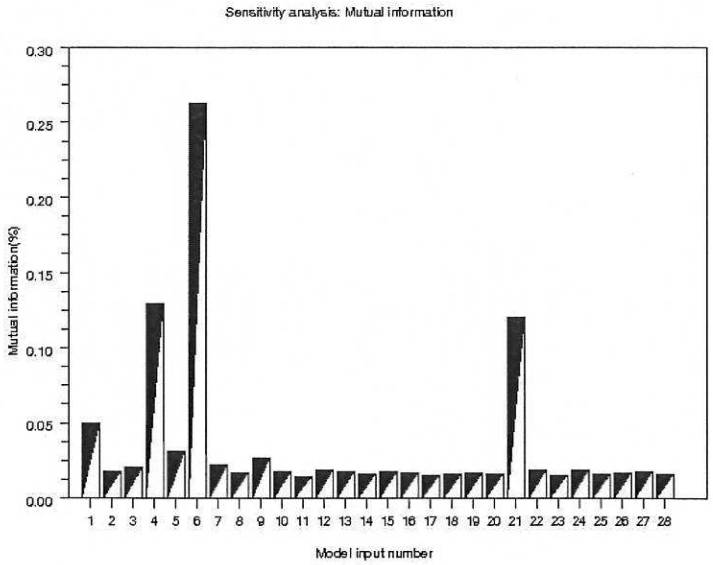

Figure 13: Mutual information between the model-output and model-inputs for the door traversal behaviour.

6.2 Wall following behaviour

The second robot's behaviour that we have analysed using mutual information was a wall following behaviour [Kyriacou et al., 2005]. In this case, instead of using the robot-training strategy described in the introduction, an artificial neural network (ANN) based controller was used to drive the robot first, its motion was then identified using the Narmax model shown in table 3. The ANN-based controller [Iglesias et al., 1998] uses a set of self-organising maps (SOM) [Kohonen, 1997] and a multilayer perceptron (MLP) neural network to process the information provided by 9 ultrasound sensors and thus determine the angular velocity the robot should attain at each instant, figure 14.

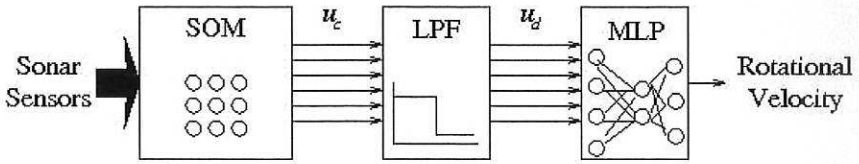

Figure 14: DIAGRAMMATIC REPRESENTATION OF THE ANN WALL-FOLLOWING PROGRAM USED TO INITIALLY CONTROL THE MOVEMENT OF THE ROBOT. THIS CONTROLLER USES THE READINGS COMING FROM THE ULTRASOUND SENSORS 1,2,3,16,15,14,13,12 AND 11 (FIGURE 2).

In order to avoid making assumptions about the relevance of specific sonar sensors, all the ultrasound measurements were taken into account in the Narmax model. Nevertheless, when we applied the mutual information analysis (figure 15), we got that the relevant sensor coincide with those sensors used by the original controller: ultrasound sensors 1,13,14,15 and 16. This is an amazing result which proves the high usefulness of the mutual information strategy.

$$
\begin{aligned}
&\omega_{model}(t) = -0.282 - 0.129 * (1/s_1(t)) \\
&-0.039 * (1/s_1(t-1)) - 0.076 * (1/s_1(t-2)) \\
&-0.017 * (1/s_3(t)) + 0.007 * (1/s_5(t)) \\
&+0.017 * (1/s_9(t)) + 0.009 * (1/s_{10}(t)) \\
&-0.007 * (1/s_{12}(t-1)) + 0.165 * (1/s_{13}(t)) \\
&-0.019 * (1/s_{13}(t-1)) + 0.079 * (1/s_{14}(t)) \\
&-0.051 * (1/s_{15}(t)) - 0.072 * (1/s_{16}(t)) \\
&+0.134 * (1/(s_1(t))^2) + 0.017 * (1/(s_1(t-1))^2) \\
&+0.096 * (1/(s_1(t-2))^2) + 0.001 * (1/(s_2(t))^2) \\
&+0.018 * (1/(s_7(t-2))^2) - 0.019 * (1/(s_{13}(t))^2) \\
&+0.056 * (1/(s_{15}(t))^2) + 0.099 * (1/s_{16}(t))^2 \\
&+0.063 * (1/(s_1(t-1) * s_{16}(t-1)) \\
&-0.071 * (1/(s_1(t-2) * s_9(t-2)) \\
&+0.039 * (1/(s_2(t) * s_{14}(t))) \\
&-0.038 * (1/(s_2(t-1) * s_6(t) \\
&+0.059 * (1/(s_3(t-1) * s_{15}(t))) \\
&+0.003 * (1/(s_{13}(t) * s_{13}(t-1))) \\
&-0.027 * (1/(s_{13}(t) * s_{14}(t)))
\end{aligned}
$$

Table 3: The NARMAX model of the ANN wall-following task, showing the rotational velocity ω as a function of the sonar sensor values s_i, $\forall i = 1, ..., 16$.

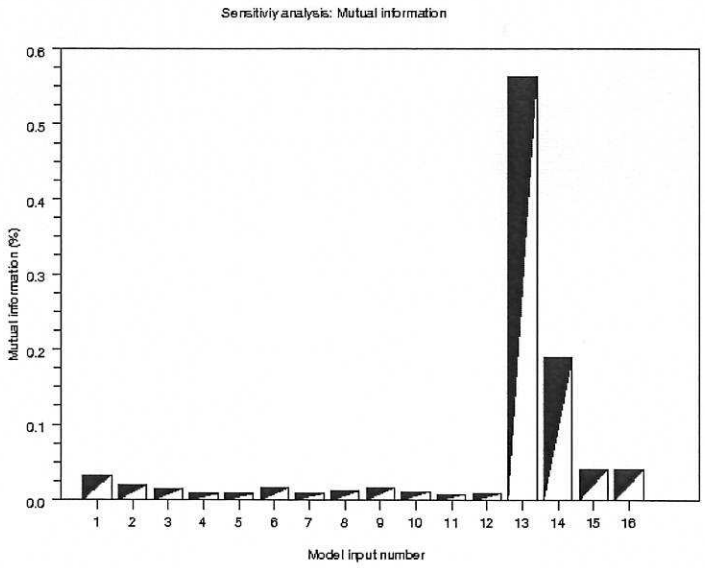

Figure 15: MUTUAL INFORMATION BETWEEN THE MODEL-OUTPUT AND MODEL-INPUTS FOR THE WALL FOLLOWING BEHAVIOUR.

7 Conclusions

Robot training using system identification is a novel method of obtaining robot control code that eliminates the need for iterative refinement of code through trial and error. To achieve sensor-motor tasks we first operate the robot under human supervision, logging sensor motor information at the same time. We then use the Narmax modelling approach to obtain a control model which identifies the coupling between sensor perception and motor responses, which is used to control the robot to move autonomously.

In this paper we show how the mathematical analysis of these models can be used to formulate hypotheses that allow the post-hoc modification of models. We demonstrate how sensitivity analysis can be used to determine the most relevant sensors in the robot's behaviour, so that a subsequent stability analysis of those relevant sensors — through partial derivatives — pinpoint those regions in the environment where sensor accuracy is crucial.

We used three different methods of sensitivity analysis, and in conclusion argue that mutual information is the most suitable indicator of sensor relevance, because it computes how much information an output conveys about a particular input — it has an actual, physically grounded meaning.

Acknowledgements

The RobotMODIC project is supported by the British Engineering and Physical Sciences Research Council under grant GR/S30955/01. Roberto Iglesias is supported through Spanish research grants PGIDIT04TIC206011PR, TIC2003-09400-C04-03 and TIN2005-03844.

We thank Theo Kyriacou and Otar Akanyeti for their help in some of these experiments, and their helpful comments on this work.

References

[Analytical and Cognitive Robotics Group, 2007] Analytical and Cognitive Robotics Group (2007). http://cswww.essex.ac.uk/staff/udfn/robotmodic/. *University of Essex.*

[Apostol, 1974] Apostol, T. (1974). *Mathematical Analysis.* Addison-Wesley Publishing Company.

[Chen and Billings, 1989] Chen, S. and Billings, S. (1989). Representations of non-linear systems: The narmax model. *International Journal of Control*, 49:1013–1032.

[Eykhoff, 1981] Eykhoff, P. (1981). *Trends and Progress in System Identification.* Pergamon Press.

[Iglesias et al., 2005] Iglesias, R., Kyriacou, T., Nehmzow, U., and Billings, S. (2005). Robot programming through a combination of manual training and system identification. In *2nd European Conference on Mobile Robots.*

[Iglesias et al., 2006] Iglesias, R., Nehmzow, U., Kyriacou, T., and Billings, S. (2006). *Current Topics in Artificial Intelligence. Lecture Notes in Artificial Intelligence*, volume 4177, chapter Training and Analysis of Mobile Robot Behaviour Through System Identification. Springer.

[Iglesias et al., 1998] Iglesias, R., Regueiro, C. V., Correa, J., Schez, E., and Barro, S. (1998). Improving wall following behaviour in a mobile robot using reinforcement learning. In *Proc. of the International ICSC Symposium on Engineering of Intelligent Systems*. ICSC Academic Press.

[Kohonen, 1997] Kohonen, T. (1997). *Self-Organizing Maps*. Springer, second edition.

[Korenberg et al., 1988] Korenberg, M., Billings, S., Liu, Y. P., and McIlroy, P. J. (1988). Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control*, 48:193–210.

[Kyriacou et al., 2006] Kyriacou, T., Akanyeti, O., Nehmzow, U., Iglesias, R., and Billings, S. (2006). Visual task identification using polynomial models. In *Proc. of Towards Autonomous Robotic Systems*.

[Kyriacou et al., 2005] Kyriacou, T., Nemhzow, U., Iglesias, R., and Billings, S. A. (2005). Task characterisation and cross-platform programming through system identification. *INT. J. ADVANCED ROBOTIC SYSTEMS*, 2:317–324.

[Nehmzow et al., 2006] Nehmzow, U., Iglesias, R., Kyriacou, T., and Billings, S. A. (2006). Robot learning through task identification. *International Journal on ROBOTICS AND AUTONOMOUS SYSTEMS*.

[Saltelli, 2002] Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145.

[Sobol, 1993] Sobol, I. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiment*, 1:407–414.