Monograph:
Iglesias, R., Kyriacou, T, Nehmzow, U et al. (1 more author) (2004) Task Identification and Characterisation in Mobile Robots. Research Report. ACSE Research Report 867 . Department of Automatic Control and Systems Engineering, University of Sheffield.

Task Identification and Characterisation in Mobile Robots

R Iglesias[#], T Kyriacou[#], U Nehmzow[#], S A Billings

#Dept Computer Science, University of Essex

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

Research Report No. 867

July 2004

Task Identification and Characterisation in Mobile Robotics

Roberto Iglesias[1], Theocharis Kyriacou[2], Ulrich Nehmzow[2] and Steve Billings[3]
[1] Dept. of Electronics and Computer Science, University of Santiago de Compostela. Spain
[2]Dept. of Computer Science, University of Essex, Colchester CO4 3SQ
[3]Dept. of Automatic Control/Syst. Eng., University of Sheffield, Sheffield S1 3JD

Abstract

The RobotMODIC project run by the Universities of Essex and Sheffield investigates ways of *quantitative* description and accurate, transparent modelling of robot-environment interaction. Transparent, analysable models allow the evaluation of robot controllers with regard to issues such as stability, relevance of individual sensor perceptions for the control task at hand, and quantitative comparison of different approaches to robot control.

In this paper we specifically discuss the problem of task identification. Using the example of a simple wall-following program, we present transparent polynomial models that achieve similar behaviour to the original controller. ARMAX and NARMAX modelling methods have been used to obtain these models.

1. Introduction

1.1 Motivation

To date the main efforts in the mobile robotics field have been oriented towards the development of different controllers aimed to obtain specific behaviours in a mobile robot. These controllers are usually combined through different control architectures, in order to let the robot solve more complex tasks. Many of these control programs have been developed through an empirical trial-an-error process of iterative refinement, until the robot's behaviour resembles the desired one to the desired degree of accuracy. It is also common to find different control approaches to obtain the same behaviour in the robot.

Although the development of all these controllers is interesting and necessary, there is a very important problem related with the required comparison between the different solutions achieved with all of them. Normally, in all the published articles it is possible to find a *qualitative* analysis of the behaviour achieved by the robot, but not an evaluation from a mathematical point of view. Contrasting different behaviours or even exchanging controllers between different research groups is something extremely useful, but too complex because usually the code of the controller is not portable between different robot platforms. Moreover, if we manage to put in our robot the controller developed by another research group, it also could happen that slight differences between the robots or the environments where they move may cause the controller not to work properly. The problem in this case is that normally we don't have a clear idea why this happens.

There is a strong tendency to work in the field of mobile robots from an engineering point of view, but we mustn't forget that the object of all experimental science is the knowledge and representation of the physical world around us (Eykhoff, 1981). As it is said in (Nehmzow, 2001a), research on *quantitative* descriptions of mobile robot behaviour is still in its infancy. Mobile robotics is still an empirical discipline that uses existence proofs extensively. The first step towards a science of mobile robotics would be the development of quantitative, rather than qualitative descriptions of mobile robotics behaviour.

1.2 Robot-Environment Interaction

A mobile robot operating in, and interacting with the environment, essentially performs a complex function that is governed by sensory perception, actuator performance and environmental factors (Nehmzow, 2001b). In this paper we present the results we have obtained and the procedure followed in order to model this function. In general, *task identification and characterisation* represents a procedure which lets us model the behaviour of a robot starting from its sensor perception. The work described in this paper is part of the RobotMODIC project at the Universities of Essex and Sheffield (Nehmzow et al., 2004). This project aims to *identify* both a mobile robot's motor response to perceptual stimuli (task identification henceforth), and the perceptual properties of the robot's environment (environment identification).

Task identification and characterisation would let us understand better some important properties of the robot's behaviour, analyse the stability of the controller or the sensibility to the sensor noise.

The usual way to analyse behaviour stability is to simply move the robot to different situations and environments, but there is usually no mathematical proof. On the other hand, sometimes what the robot is doing depends just on what it is seeing through two or three specific sensors, so not all the sensor information has the same importance. Nevertheless when methodologies like reinforcement learning and artificial neural networks are applied, we don't have a transparent relationship between what the robot is doing and what it is seeing through each one of its sensors.

A deeper study through mathematical models of the different solutions achieved for the same behaviour would let us understand better the weakest aspects of the different controllers. Understanding the reason why our robot behaves properly in some situations, and why in others its behaviour is totally different and unexpected, would help us to develop more robust behaviours and solve new problems in the future, instead of solving several times the same kind of behaviours, just changing the methodology used, and without a clear idea of why our new solution may be better or worse than previous ones. As it is said in (Eykhoff, 1981), trying to optimise a control system without identifying it seems a dangerous solution.

A mathematical description of a robot's behaviour, for example in the form of a (polynomial) model, facilitates communication between research groups. Control code can more easily be shared, analysed for stability and sensitivity to certain sensor signals. The sharing of models makes the design of mobile robot controllers more transparent, as the control strategy is clearly expressed in closed form. Furthermore, unlike opaque models (such as for instance artificial neural networks or fuzzy controllers), transparent models have obvious benefits when reasons for any failures need to be established.

2. System Identification

2.1 Introduction

In this section we discuss the relevant aspects to the problem of task identification and characterisation, and the different modelling options we have used.

Let us assume we have a sensor-based controller that we want to identify and model. This controller is able to process the sensor information and decide the commands that the robot must carry out at every point in time in order to solve a specific problem or carry out a particular task. Our aim is to replace the original controller with a mathematical representation in such a way that, starting from the sensor information, the model should be able to make the robot behave in the same way as the original controller did.

It is convenient to characterise system identification by three elements: a set of data, D, a set of models M, and a criterion C.

In our case, the set of data is information collected from the robot when it is moving in a specific environment, solving a particular task. The set of data consists on a sequence of input-output pairs $\{u(t_k), y(t_k)\}$, where $u(t_k)$ are the sensory perceptions and $y(t_k)$ the motor responses of the robot, which have been logged during the robot's operation at discrete sampling points t_k.

The identification problem consists of finding a model in the set M that explains the data set D. By the set M of models we can understand a class of functions used to obtain the mathematical expression we are looking for. In the model that we select there will be a set of parameters whose value has to be determined according to the information stored in D. The adjustment of this parameters will be carried out according to a specific criterion C, normally called *loss function* (Eykhoff, 1981).

2.2 Model Validation

Once the model is obtained it has to be validated. System identification is an iterative procedure which includes the selection of model structure and criteria for the parameter estimation and finally validation. There is no guarantee that a suitable model can be obtained with the chosen model structure and the parameter estimated. There are many validation methods (Box and Jenkins, 1970). In our case, to avoid the problem of overfitting and to check if the models we obtained are suitable for our purposes, we applied two different options.

First, we divided the experimental data in two different sets: a *training data set*, which is used for parameter estimation, and a *validation data set*, which is used for validation. As we will see clearly in the next subsection, the adjustment of the parameters of the models will be carried out trying to reduce a specific error (evaluated through what we previously called loss function, see equation 2) over the training data set. Once the parameters are determined, this error will be evaluated again, but in this case over the validation data set, trying to see if it is similar to the one obtained with the training data.

Second, we have validated the obtained models by actually controlling our mobile robot using the models, and comparing the behaviours obtained. The purpose of this second validation procedure was to establish that the obtained model was not only numerically close to the original, but also resulted in similar robot behaviour under real-world conditions.

In the next two subsections we'll describe briefly each one of the two modelling options we have used in this work.

2.3 ARMAX modelling

One of the parameterised model structures we have used is the ARMAX model (Auto-Regressive Moving Average model with eXogenous inputs) (Eykhoff, 1981). Through the ARMAX structure we'll consider that the task we are trying to identify can be described by equation 1.

$$y(t) = \sum_{k=1}^{P} a_k y(t-k) + \sum_{m=0}^{Q} b_m u(t-m) + \sum_{r=0}^{R} c_r e(t-r), \quad (1)$$

where $y(t)$, $u(t)$, and $e(t)$, are the sampled output, input (which can be multidimensional), and unobservable noise sequences respectively.

According to equation 1, the current output of the system $y(t)$, depends on the P previous output values $y(t-1)$,..,$y(t-P)$, and also on the current and the Q past values of an extra external input signal: $u(t)$, $u(t-1)$,...,$u(t-Q)$. P and Q values are known as input and output orders.

As previously mentioned in section 2.1, we have a set of data D which we try to model using the polynomial expression given in equation 1. The data used for modelling consists of sensory perception and motor responses obtained at consecutive discrete points in time. Half of this data are used as training data set, while the remaining half is used as validation set.

Through an iterative process, the values of the parameters $a(t)$,..,$b(t)$, are estimated trying to minimise the value of a loss function. This loss function is defined as the squared difference between the output of the system in each sampling time, $y(t)$, and the output, $\hat{y}(t)$, predicted by the model for the same point in time (equation 2).

$$loss\_function = \sum_{t=max\{P,Q\}}^{N} [y(t) - \hat{y}(t)]^2, \quad (2)$$

where N is the number of points in the data set. If correct parameter estimates are to be obtained, the noise sequence $e(t)$, which is almost always unobservable, must be estimated and accommodated within the model.

ARMAX modelling has been successfully applied for system identification in a large variety of different problems. The linearity in the parameters to be calculated makes its estimation easy; the linear structure allows us to obtain a predictor with a simple closed expression.

2.4 NARMAX modelling

We have also used another model structure in order to solve the problem of task identification. This new representation structure, known as NARMAX (Nonlinear Auto-Regressive Moving Average model with eXogenous inputs) (Chen and Billings, 1989, Korenberg et al., 1988), is more suitable than the previous one for the modelling of nonlinear systems. In fact, this new strategy can be very suitable for task identification in robotics, given that sometimes the behaviour of the robot tend to be of high dimensionality and often with discontinuous control laws.

The polynomial NARMAX model is linear in the parameters, thus allowing the application of readily available parameter estimation techniques, and without imposing any restrictions in the nature of the inputs. For the modelling experiments reported here, we have adapted and simplified the general model proposed in (Chen and Billings, 1989), the model we use takes the form given in equation 3.

$$y(t) = \sum_{k=1}^{P} a_k y(t-k) + F^l(u(t), ..., y(t-Q)) - 1, \quad (3)$$

where $F^l(u(t), ..., u(t-Q))$ is a polynomial of degree l where the input signals are combined. Q is the input order or maximum input lag, and P is the output order. Assuming the input signal is multidimensional, $u(t) = (u_1(t), ..., u_n(t))$, the expression of F^l can be written as:

$$F^l(u(t), ..., u(t-Q)) =$$

$$= \sum_{e_1=0}^{l} \cdots \sum_{e_{n(Q+1)}=0}^{l} \prod_{i=1}^{n} \prod_{k=0}^{Q} a_{e_1 e_2, ..., e_{n(Q+1)}} u_i^{e_{(i-1)n+k+1}}(t-k),$$

where $a_{e_1 e_2, ..., e_{n(Q+1)}}$ are coefficients whose values have to be estimated.

NARMAX system identification involves two main stages: (i) *structure selection* (by means of which the most useful terms to be included in the model are selected), and (ii) *parameter estimation.*

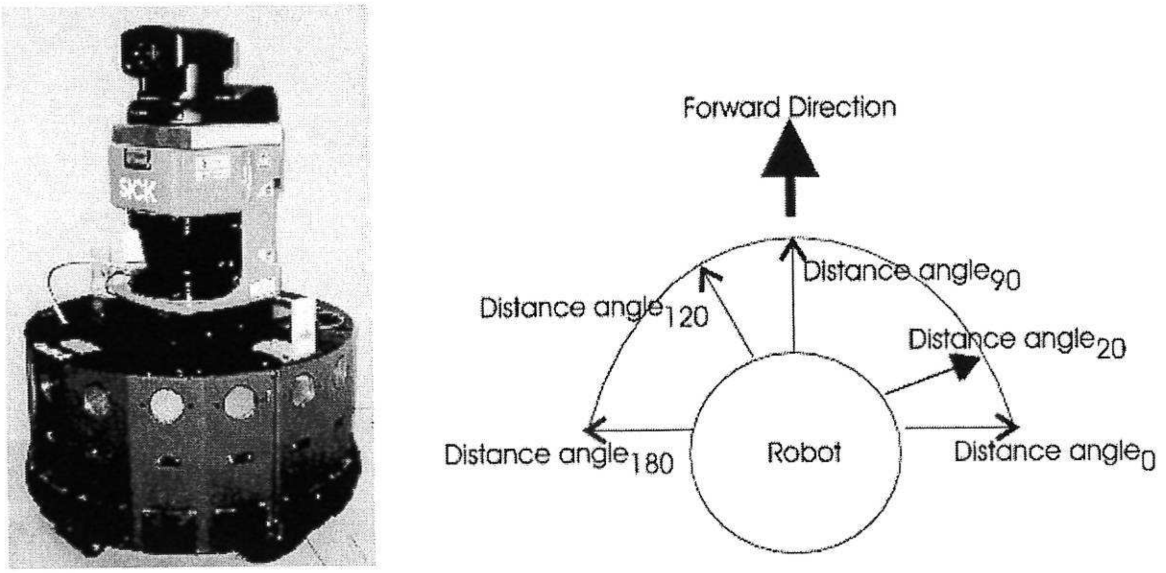
In our case we have used the mutual information (Abarbanel, 1996) between the output and the inputs as criterion to select those terms that are more relevant for the problem. Once this is done, a parameter estimation is carried out trying to minimise the same error we used in ARMAX modelling (equation 2). In general, the steps we followed in order to obtain our NARMAX models are the following:

1. Structure selection (mutual information is used for this)

2. Parameter estimation

3. If the error value calculated over the training data set is being reduced go to step 4, otherwise stop

4. Removal of the irrelevant terms according to the last values estimated for their coefficients. Go to step 2.



A) B)

Figure 1: THE MAGELLAN PRO MOBILE ROBOT USED IN THE EXPERIMENTS DESCRIBED IN THIS PAPER (A) IS EQUIPPED WITH 16 ULTRASOUND SENSORS AND A SICK LASER SCANNER. THIS LASER SCANNER HAS AN OPENING ANGLE OF 180°, AND IT IS ABLE TO DELIVER 180 DISTANCE MEASUREMENTS, ONE PER DEGREE (B).

3. Experimental Results

3.1 Modelling Approach and Experimental Procedure

In this section we discuss the application of the previously described ARMAX and NARMAX modelling approaches for task characterisation and identification. A detailed discussion is beyond the scope of this paper — instead we demonstrate the method using basic examples obtained through real-world robotics experimentation at Essex.

The robot we use is the autonomous mobile robot Radix, a Magellan Pro robot shown in figure 1. As we can see in this figure, this robot has 16 ultrasound sensors, and is furthermore equipped with a SICK laser scanner.

Initially, we let Radix operate under the control of a controller developed to let the robot follow walls on its right hand side in a closed environment. While the robot was moving, the sensory perceptions, as well as the translational and angular velocity of the robot were logged every 250 milliseconds.

In order to solve the problem of task identification in this case, we applied the ARMAX and NARMAX strategies trying to obtain two different kinds of model, one to model the link between sensory perception and angular velocity, and the other to model the link between sensory perception and translational velocity of the robot.

3.2 The Wall-Following Controller Modelled

To clarify our modelling approach by way of presenting an example, we begin by discussing the original wall-following controller used (this section). We continue by presenting the identification (models) of this task (section 3.3.1) and conclude by discussing the validation of these models (section 3.3.2).

The controller we have developed determines which are the motor responses of the robot suitable to keep it following the wall on its right (figure 2). In particular it determines the angular and linear velocities starting from a distance, *ld*, which is calculated from the information provided by the laser scanner:

$$ld = \frac{\sum_{i=0}^{90} distance\_angle_i i}{\sum_{i=0}^{90} i}, \tag{4}$$

where $distance\_angle_i$, with $i \in [0, 90°]$, are the right side distances delivered by the SICK laser scanner. We can see that these 90 distances are weighted in such a way that the information coming from the front of the robot is more important than the distances detected on the right side (0 degrees).

Once the *ld* distance is calculated, the angular and linear velocities $\dot{\theta}$ and v are set to the following values:

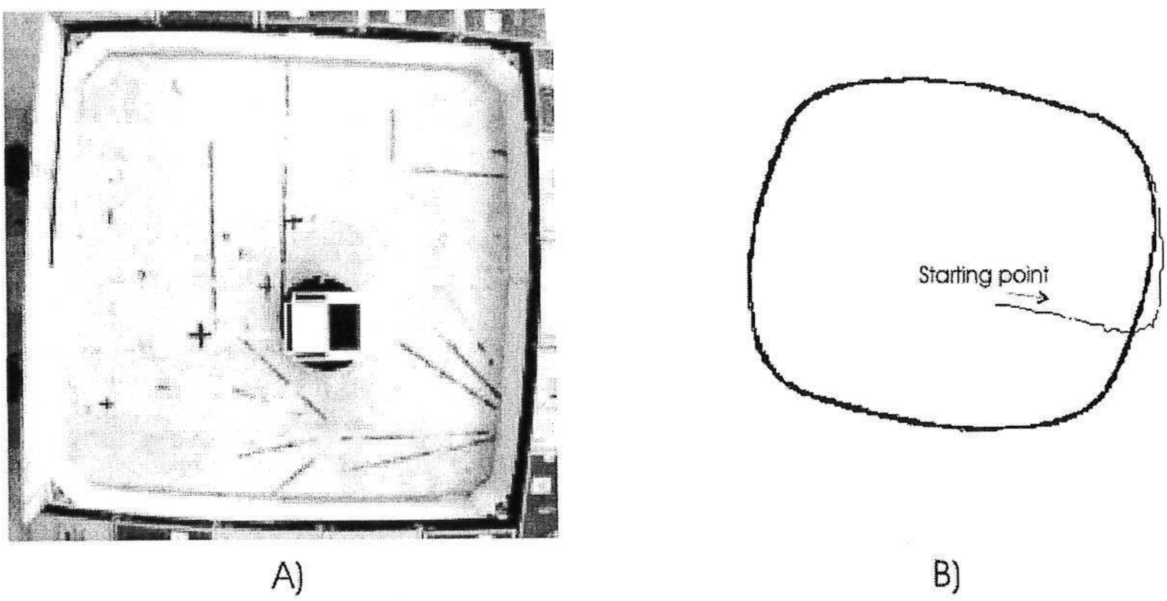


Figure 2: (A) ENVIRONMENT WHERE THE EXPERIMENTS WERE CONDUCTED. THE ROBOT IS VISIBLE CLOSE TO THE CENTRE OF THE IMAGE. (B) ROBOT'S TRAJECTORY WHEN THE WALL FOLLOWING CONTROLLER IS BEING USED.

$$V(t) = max\{min\{V_{max}[1 - (\frac{d_{max} - d}{d_{max}})^2], V_{max}\}, 0\},$$

$$\dot{\theta}(t) = max\{-\dot{\theta}_{max}, min\{\dot{\theta}_{max}(\frac{d - d_{avg}}{d_{min} - d_{avg}}), \dot{\theta}_{max}\}\},$$

where $V_{max} = 0.15m/s$ is the maximum desired speed, $d_{max} = 1.2m$ and $d_{min} = 0.5m$, and, finally, $d_{avg} = 1m$.

3.3 *Experimental Results*

3.3.1 *Models Obtained*

After moving the robot using this controller, we have modelled the angular and linear velocities using different ARMAX and NARMAX strategies.

For modelling the angular velocity, both strategies (ARMAX and NARMAX) were able to obtain accurate models when the input signal $u(t)$ used the same distance criterion as the original controller (equation 4). The models obtained in both cases are shown in table 1.

To obtain the model in this case, we have actually used information about the original control program, that is, we used exactly the same sensor information that was used in the original wall follower. Furthermore, we preprocessed the sensor information in the same way as was done in the original controller. For many applications, however, such information may not be available — it will sometimes be the case that only trajectory and sensor information is available to model the control program governing the robot's behaviour.

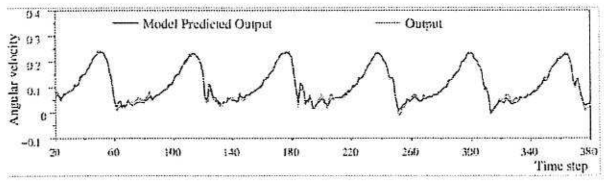


Figure 3: NARMAX MODELLING OF THE ANGULAR VELOCITY AS A FUNCTION OF THE LASER SENSOR PERCEPTION (SEE TABLE 2)

To show the usefulness of task identification, we therefore obtained different models of the angular velocity, where different sensory inputs were used. Specifically, we obtained and validated 4 different models of the angular velocity. Two of them used the 16 sonar sensor measurements as input signal at every point in time, while the other two used the laser sensor in a way different to that used in the control program.

The ARMAX strategy was successfully employed in the two models where all the ultrasound measurements were used as inputs.

Regarding the models where the values coming from the laser scanner make up the input information, both ARMAX and NARMAX strategies were used (table 2). In this case, the values delivered by the SICK laser scanner were grouped in order to obtain a 6 dimensional vec-

NARMAX	$\dot{\theta}(t) = +1.3089801 * ld(t-1) - 1.3270617 * ld(t) * ld(t-1)$
ARMAX	$\dot{\theta}(t) = +0.5699629 * \dot{\theta}(t-1) + 0.4298233 * \dot{\theta}(t-2) - 0.9185964 * ld(t) + 0.4559368 * ld(t-1)$ $+0.4644224 * ld(t-2)$

Table 1: ARMAX AND NARMAX MODELS OF THE ANGULAR VELOCITY $\dot{\theta}$, WHEN THE SAME DISTANCE CRITERION AS IN THE ORIGINAL CONTROLLER (4) WAS USED AS INPUT. IN THE ARMAX MODEL WE USED INPUT AND OUTPUT ORDERS OF 2 AND 1, RESPECTIVELY. FOR THE NARMAX MODEL INPUT ORDER=1, OUTPUT ORDER=0, AND DEGREE=1.

tor of distances $\vec{d} = (d[1], ..., d[6])$. Each distance was calculated in the following way:

$$d[i] = \frac{\sum_{j=(i-1)15+1}^{j=(i-1)15+15} distance\_angle_j}{15}, \quad i = 1, ..., 6, \tag{5}$$

where $distance\_angle_j$ are the distances on the right side delivered by the SICK laser scanner.

The same vector of distances, $\vec{d}$, also proved to be very useful in the ARMAX model we obtained to predict the linear velocity (table 3).

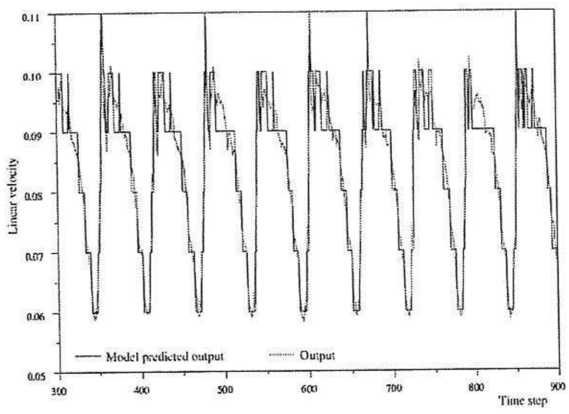


Figure 4: ARMAX MODELLING OF THE ROBOT'S TRANSLATIONAL VELOCITY AS A FUNCTION OF THE LASER SENSOR PERCEPTION (SEE ALSO TABLE 3)

Figures 3 and 4 show the actual angular and translational velocities of the robot, as well as the model-predicted velocities. To see the degree of accuracy of the models we have calculated the Pearson (r) and Spearman rank correlation coefficients between the modelled and the true velocities of the robot. Thus, in the case of the angular velocity, r varies between 0.993 (significant, $p<0.05$) in the best case (NARMAX model), and 0.872 (sig., $p<0.05$) in the worst one (ARMAX model using the 16 ultrasound sensors as input information, input order=1, output order=0). The Spearman rank correlation coefficient obtained for these two models, 0.983 (sig., $p<0.05$) and 0.817 (sig., $p<0.05$) respectively, also represent the maximimum and the minimum values for this parameter.

When the linear velocity was modelled using ARMAX, the values correponding to the Pearson and the Spearman rank correlation coefficients were 0.966 (sig., $p<0.05$), and 0.946 (sig., $p<0.05$), respectively.

Through these values is important to notice that NARMAX strategy let us achieve a better model than ARMAX, which is reasonable taking into account that NARMAX can model non linear systems. On the other hand, we could also appreciate that the correlation coefficients are higher for those models using laser as input information than for those using sonar readings instead. Thus, r=0.934 (sig., $p<0.05$) for the ARMAX model using 16 ultrasound sensors, input order=4, and output order=0, while for the ARMAX model using laser as input information, input order=1, and output order=0, r=0.968 (sig., $p<0.05$). The Spearman rank correlation for these two models is 0.896 (sig., $p<0.05$), and 0.923 (sig., $p<0.05$), respectively. There are two reasons which can justify these values: one of them is the fact that laser readings are more accurate. The other reason is that the original behaviour we have modelled decides the robot's behaviour at each instant starting from the laser readings.

3.3.2 Validation of Models

In order to validate all these models, we applied the two options mentioned in section 2.2. The value of the loss function was evaluated over the training and the testing data sets, and all these models were used to control the movement of the robot.

The trajectories observed when the models were applied to move the robot were very similar to the one obtained with the original controller (compare figures 2 and 5).

In general, we observed that the use of high input or output orders in the model tend to cause a generalisation problem (overfitting). As the number of terms of the model increases, the value of the loss function over the training data decreases. The problem is that when we apply a model with a high number of terms to control the robot, its movement tends to become unstable. Occasionally the model even makes the robot fail.

We observed some statistics of the original controller's

NARMAX	$\dot{\theta}(t) = -8.9721909 * d[6](t-1) + 12.234227 * d[6](t) - 2.1568866 * d[6](t) * d[6](t-1)$ $-3.5634503 * d[5](t-1) + 13.713349 * d[5](t-1) * d[6](t-1) - 9.6794691 * d[5](t-1) * d[6](t)$ $-1.2538074 * d[5](t-1) * d[6](t) * d[6](t-1) + 3.9611084 * d[5](t) + 6.7678404 * d[5](t) * d[6](t-1)$ $-16.980652 * d[5](t) * d[6](t) + 5.2273765 * d[5](t) * d[6](t) * d[6](t-1)$ $-0.6226611 * d[5](t) * d[5](t-1) - 10.826077 * d[5](t) * d[5](t-1) * d[6](t-1)$ $+14.435493 * d[5](t) * d[5](t-1) * d[6](t) - 2.1819942 * d[5](t) * d[5](t-1) * d[6](t) * d[6](t-1)$
ARMAX	$\dot{\theta}(t) = -0.0833922 * d[1](t) + 0.0686805 * d[2](t) + 0.0012728 * d[3](t) - 0.2209385 * d[4](t)$ $+0.2141653 * d[5](t) - 0.4676208 * d[6](t) + 0.3688172 * d[1](t-1) + 0.0356499 * d[2](t-1)$ $+0.0500439 * d[3](t-1) + 0.2122142 * d[4](t-1) - 0.1606333 * d[5](t-1) + 0.2895952 * d[6](t-1)$

Table 2: ARMAX AND NARMAX MODELS OF THE ANGULAR VELOCITY $\dot{\theta}$, WITH THE INFORMATION DELIVERED FROM THE LASER SCANNER USED AS INPUT. IN THE ARMAX MODEL THE INPUT AND OUTPUT ORDERS WERE SET TO 1 AND 0, RESPECTIVELY. FOR THE NARMAX MODEL INPUT ORDER=1, OUTPUT ORDER=0, AND DEGREE=1. THE MEANING OF $d[i]$ VALUES, $\forall i = 1, ..., 6$ IS SPECIFIED IN EQUATION 5.

v(t)=+0.0075999*d[1](t)+0.0271140*d[2](t)
-0.0099592*d[3](t)+0.0471389*d[4](t)
-0.0095064*d[5](t)+0.0855093*d[6](t)
-0.0221077*d[1](t-1)-0.0371495*d[2](t-1)
+0.0239458*d[3](t-1)-0.0280440*d[4](t-1)
+0.0299362*d[5](t-1)-0.0364634*d[6](t-1)

Table 3: ARMAX MODEL OF THE ROBOT'S TRANSLATIONAL VELOCITY $v(t)$, WHEN THE INFORMATION DELIVERED FROM THE LASER SCANNER WAS BEING USED AS INPUT. INPUT ORDER=1, OUTPUT ORDER=0. THE MEANING OF D[I] VALUES, $\forall i = 1, ..., 6$ IS SPECIFIED IN EQUATION (5)

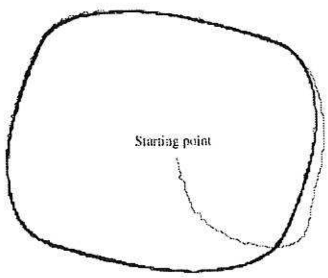


Figure 5: ROBOT'S TRAJECTORY WHEN THE MODELS CORRESPONDING TO THE TRANSLATIONAL AND ANGULAR VELOCITIES WERE USED. IN THIS CASE THE MODEL USED FOR THE LINEAR VELOCITY WAS THE ONE USING ARMAX, WHILE THE ANGULAR VELOCITY WAS MODELLED WITH NARMAX. THE VALUES DELIVERED BY THE LASER SCANNER ARE USED AS INPUT INFORMATION IN BOTH MODELS. COMPARE THIS TRAJECTORY WITH THE ORIGINAL TRAJECTORY SHOWN IN 2.

behaviour as well as those of the models being executed on the robot, for a period of about 100 minutes each. Results are shown in table 4. The trajectory obtained using the model given in the last row of table 4 is shown in figure 5.

It is interesting to note here that a robot's behaviour — in this case wall-following — can be achieved in the traditional way of encoding the control procedure, using an established language such as for example C. It can, however, also be obtained by identifying the behaviour (in the sense of system identification), and executing the identified model on the robot. The possibilities for generating robot control code and sharing it across platforms is obvious.

In the histogram of distances shown in figure 6, we can see that in almost all the different modelling situations, the robot's distance from the wall stays in the interval [0.25m,0.5m]. Even more, all the curves in the histogram show a fast fall between 0.35 and 0.5 meters. The only behaviour which seems to be different is the one obtained through the use of the ARMAX model of the angular velocity, when laser information was used as input. In this case, although the shape of the curve is similar to the one corresponding to the original controller, the robot seems to move a bit further away from the wall which is being followed.

The fact that we were able to obtain accurate models with different input signals proves that it is not necessary to know in advance the characteristics of the controller. This fact increases the usefulness of the task identification and characterisation, given that the mathematical expressions obtained can help us to understand better the relationship between what the robot sees through the different sensors, and its movement in the environment. Through the coefficients obtained in the model we can analyse which sensor information seems to be more relevant.We can also attempt to achieve the same wall following behaviour, using different sensors, e.g, replacing laser with sonar.

Moreover, we could even think of moving the robot with the help of a man-machine interface or a joystick, "teaching" the desired behaviour. Once this is done, we could try to obtain the corresponding models starting

Model	Minimum distance	Maximum distance
Original Controller	0.26m	0.85m
ARMAX, 16 ultrasound sensors, input order=1,output order=0	0.28m	0.59m
ARMAX, 16 ultrasound sensors, input order=4,output order=0	0.25m	0.89m
ARMAX, laser, input order=1, output order=0	0.41m	0.78m
NARMAX, laser, input order=1, degree=1, output order=0	0.25m	0.97m
NARMAX, laser for angular velocity, ARMAX laser for linear velocity	0.24m	1.0m

Table 4: DISTANCES BETWEEN THE ROBOT AND THE WALL WHICH WAS BEING FOLLOWED WHEN ITS MOVEMENT WAS CONTROLLED BY EACH ONE OF THE MODELS. SEE ALSO FIGURE 6

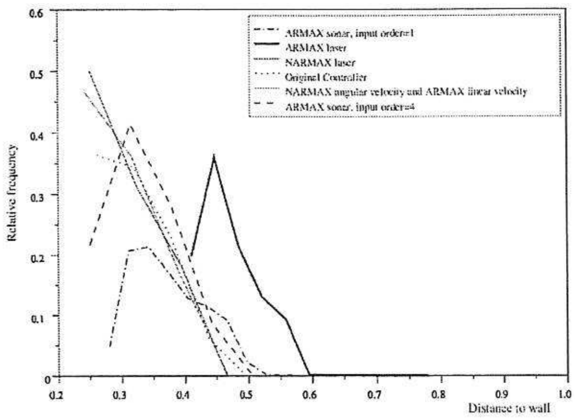


Figure 6: THE RELATIVE FREQUENCY OF THE DISTANCES BETWEEN THE ROBOT AND THE WALL WHICH WAS BEING FOLLOWED CAN BE SEEN IN THIS HISTOGRAM. IT WAS OBTAINED STARTING FROM THE ROBOT'S MOVEMENT WHEN IT WAS CONTROLLED BY EACH ONE OF THE MODELS. SEE ALSO TABLE 4.

from the data logged during the robot's operation. Finally, taking into account the conclusions obtained from the analysis of the model (trying to see, for example, which sensor information seems to be more relevant, etc) it should be easier to develop a good controller.

4. Conclusions

In this paper we have emphasised the use and relevance that task identification and characterisation could have in the mobile robotics domain. Using mathematical models to characterise the link between sensor perceptions and motor response of a mobile robot opens new doors in the research on quantitative descriptions of behaviours in mobile robotics.

Once transparent mathematical models of sensor-motor behaviour are available, they can be used to determine the stability of controllers, to identify the relevance of individual sensor signals for the overall robot operation, or even to obtain new and more concise control mechanisms.

In certain cases it may be possible to exchange task models between different robot platforms, thus essentially "programming" mobile robots not through the explicit definition of a control mechanism, but through a model derived from it. This may, in some cases, lead to platform-independent control programs, in other cases it will aid the design of control programs for novel robot platforms.

Our first experimental results confirm that there are cases where this approach is viable. The circumstances under which task identification is a viable way of controlling robots *in general* is subject to ongoing research at Essex and Sheffield.

5. Acknowledgements

The authors thank the following institutions for their support: The RobotMODIC project is supported by the Engineering and Physical Sciences Research Council under grant GR/S30955/01.

R.I. is supported by Xunta de Galicia through sabbatical fellowship DOG 20/05/2004, and through the research grant TIC2003-09400-C04-03.

References

Abarbanel, H. D. (1996). *Analysis of Observed Chaotic Data.* Springer.

Box, G. E. P. and Jenkins, G. M. (1970). *Time Series Analysis, Forecasting and Control.* Holden-Day.

Chen, S. and Billings, S. A. (1989). Representations of non-linear systems: The narmax model. *Int. J. Control*, 49:1013–1032.

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

Korenberg, M., Billings, S. A., Liu, Y. P., and McIlroy, P. J. (1988). Orthogonal parameter estimation algorithm for non-linear stochastic sytems. *Int. J. Control*, 48(1):193–210.

Nehmzow, U. (2001a). Mobile robotics: Research, applications and challenges. In *Future Trends in Robotics.* Institution of Mechanical Engineers.

Nehmzow, U. (2001b). Quantitative analysis of robot-environment interaction-on the difference bewtween simulations and the real thing. In *Eurobot.*

Nehmzow, U., Kyriacou, T., Iglesias, R., and Billings, S. (2004). Robotmodic: Modelling, identification and characterisation of mobile robots. Submitted to TAROS 2004 - Towards Autonomous Robotic Systems.