Published paper

Akanyeti, O., Kyriacou, T., Nehmzow, U., Iglesias, R., Billings, S.A. (2007) *Visual task identification and characterisation using polynomial models*, Robotics and Autonomous Systems, 55 (9), pp. 711-719

http://dx.doi.org/10.1016/j.robot.2007.05.016

Visual Task Identification and Characterisation Using Polynomial Models

O. Akanyeti[1], T. Kyriacou[1], U. Nehmzow[1], R. Iglesias[2] and S.A. Billings[3]

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

Abstract

Developing robust and reliable control code for autonomous mobile robots is difficult, because the interaction between a physical robot and the environment is highly complex, subject to noise and variation, and therefore partly unpredictable. This means that to date it is not possible to predict robot behaviour based on theoretical models. Instead, current methods to develop robot control code still require a substantial trial-and-error component to the software design process.

This paper proposes a method of dealing with these issues by a) establishing task-achieving sensor-motor couplings through robot training, and b) representing these couplings through transparent mathematical functions that can be used to form hypotheses and theoretical analyses of robot behaviour.

We demonstrate the viability of this approach by teaching a mobile robot to track a moving football and subsequently modelling this task using the NARMAX system identification technique.

1. Introduction

The behaviour of a robot (for example the trajectory of a mobile robot) is influenced by three components: i) the robot's hardware, ii) the program it is executing and iii) the environment it is operating in. Because this is a complex and often non-linear system, in order to program a robot task to achieve a desired behaviour, one usually has to resort to empirical trial-and-error processes. Such iterative-refinement methods are costly, time-consuming and error prone [Iglesias et al., 2005].

One of the aims of the RobotMODIC project at the universities of Essex and Sheffield is to establish a scientific, theory-based, design methodology for robot control program development. As a first step towards this aim we "identify" a robot's behaviour, using system identification techniques such as ARMAX (Auto-Regressive Moving Average models with eXogenous inputs) [Eykhoff, 1974] and NARMAX (Nonlinear ARMAX) [Chen and Billings, 1989,Billings and Chen, 1998]. These produce linear or nonlinear polynomial functions that model the relationship between user-defined input and output, both pertaining to the robot's behaviour. The representation of the task as a transparent, analysable model enables us to investigate the various factors that affect robot behaviour for the task at hand. For instance, we can identify input-output relationships such as the sensitivity of a robot's behaviour to particular sensors [Roberto Iglesias and Billings, 2005], or make predictions of behaviour when a particular input is presented to the robot. This, we believe, is a step towards the development of a theory of robot-environment interaction that will enable a more focused and methodical design of robot controllers.

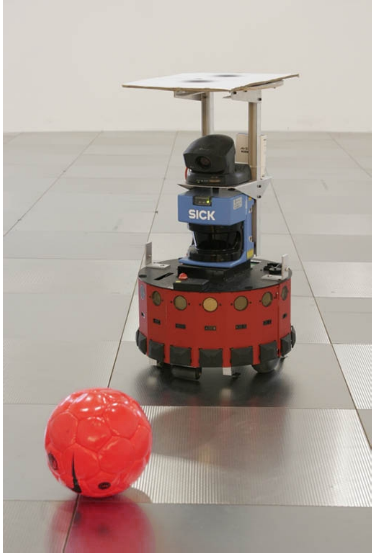

Fig. 1. The Magellan Pro mobile robot *Radix* and the orange–coloured ball used in the experiments described in this paper.

1.1. *Identification of visual sensor-motor competences*

In previous work [Iglesias et al., 2004,Iglesias et al., 2005] and [Nehmzow et al., 2005] we presented and analysed models for mobile robot tasks that used the laser and sonar rangefinder sensors of our Magellan Pro mobile robot as input modalities. Tasks included wall-following, obstacle avoidance, door traversal and route-learning.

In this paper we show how the same modelling methodology can be applied for tasks using the vision sensor (video camera) on our robot. The task that we model here is one where the robot follows a bright orange ball in an open space of more or less uniform grey colour. We use the NARMAX system identification methodology (described briefly in section 2.2) in order to model this task.

2. Experimental procedure and methods

2.1. *Experimental procedure*

The experiments described in this paper were conducted in the 100 square metre circular robotics arena of the University of Essex, using a Magellan Pro mobile robot called *Radix* (figure 1). The robot is equipped with 16 sonar, 16 infra-red and 16 tactile sensors, all uniformly distributed. A SICK laser range finder is also present. This range sensor scans the front semi-circle of the robot ($[0°, 180°]$) with a radial resolution of 1° and a distance resolution of less than 1 centimetre. The robot also incorporates a colour video camera on a pan/tilt mount. In the work presented here only the video camera was used.

2.1.1. *Acquisition of estimation and testing data*

In order to collect training data for the estimation of the task model a human driver manually drove the robot using a joystick to set both linear and angular velocities, guiding the robot to follow a moving orange ball (see figure 1). The human driver had no visual contact with the robot itself and used only the robot's camera images to steer the robot towards the ball. The robot's camera was tilted to its lower extreme so that the robot's field of view covered the area closest to the robot.

The robot was driven in this manner for 1 hour. During this time a coarse-coded robot camera image and the robot's translational and rotational velocities were logged every 250 ms. The camera image was coarse-coded to a minimal 8x6 pixel image by averaging neighbourhoods of 20x20 pixels in the original 160x120 pixel image. Figure 2 shows an example of a camera image and its coarse-coded version. Coarse-coding of the camera image was done to minimise hard disk access and memory requirements during the robot's operation and to reduce the dimensionality of the input to the NARMAX model.

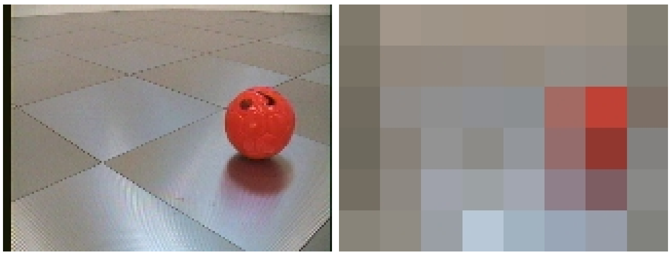

Fig. 2. An example of a robot camera image (left) and its coarse-coded version (right). Note that the coarse-coded image is shown enlarged by a factor of 20.

After the collection of the model estimation and validation data a polynomial model was obtained, expressing the rotational velocity of the robot as a function of the colour of each of the pixels in coarse-coded camera image. During the experiments the linear velocity of the robot was kept constant.

A brief explanation of the model estimation procedure used is given in the following section.

2.2. *The NARMAX modelling methodology*

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

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

where $y(n)$ 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, d is the dimension of the input vector and l is the degree of the polynomial. $f()$ is a non-linear function and here taken to be a polynomial multi-resolution expansion its arguments. Expansions such as multi-resolution wavelets or Bernstein coefficients can be used as an alternative to the polynomial expansions considered in this study.

The first step towards modelling a particular system using a NARMAX model structure is to select appropriate inputs $\boldsymbol{u}(n)$ and the output $y(n)$. The general rule in choosing suitable inputs and outputs is that there must be a causal relationship between the input signals and the output response.

After the choice of suitable inputs and outputs, the NARMAX methodology breaks the modelling problem into the following steps:

(i) Polynomial model structure detection: During this step we determine the linear and non linear combinations of inputs (for instance quotients or products) in the polynomial to search to begin with.

(ii) Model parameter estimation: Then we estimate the coefficients of each term found in the polynomial.

(iii) Model validation: And in model validation, we measure the prediction error of the obtained model.

The last two steps are performed iteratively (until the model estimation error is minimised) using two sets of collected data: (a) the *estimation* and (b) the *validation* data set. Usually a single set that is collected in one long session is split in half and used for this purpose.

The model estimation methodology described above forms an estimation toolkit that allows us to build a concise mathematical description of the input-output system under investigation. We are constructing these models in order to learn the underlying rules from the data. This is like theoretical or analytical modelling but we let the data inform us regarding what terms and effects are dominant etc. So the models are constructed term by term.

In analytical modelling we put the most important term in first then the next etc based on our knowledge. NARMAX methodology does the same but it learns them from the data, it does not have any prejudices, and it automatically accommodates assumptions and the interactions between these which have to be taken into account in analytical modelling.

ARMAX and NARMAX procedures are now well established and have been used in many modelling domains [Billings and Chen, 1998]. A more detailed discussion of how structure detection, parameter estimation and model validation are done is presented in [Korenberg et al., 1988,Billings and Voon, 1986].

3. Experimental results

We used the Narmax system identification procedure to estimate the robot's rotational velocity as a function of 144 inputs (the red, green and blue values of each of the 8x6 pixels of the coarse-coded camera image), using the training data obtained during the ball-following experiment. The model was chosen to be of first degree and no regression was used in the inputs and output (i.e. $l = 1$, $N_u = 0$, $N_y = 0$) resulting in a linear ARMAX polynomial structure. The resulting model contained 53 terms:

$$\begin{aligned}
\omega(n) = & \\
& +0.1626308495 \\
& +0.0028080424 * u_8(n) \\
& -0.0016263169 * u_{14}(n) \\
& -0.0025145629 * u_{15}(n) \\
& \ldots \\
& +0.0061225193 * u_{129}(n) \\
& -0.0051800999 * u_{136}(n) \\
& +0.0012762243 * u_{144}(n)
\end{aligned}$$

where $\omega(n)$ is the rotational velocity of the robot in rad/s at time instant n. Positive ω indicates that the robot turns left and negative ω indicates that the robot turns right. Integers u_1 to u_{48} are the red image components, u_{49} to u_{96} the green and u_{97} to u_{144} the blue channels of the coarse-coded image pixels (starting from the top left of the image and reading from left to right each image row).

A graphical representation of the model parameters is given in figure 3. This figure shows the contribution of each coarse-coded image pixel to the rotational velocity $\omega(t_n)$. This contribution is obviously dependent on the colour value of the pixel at time t_n. Note that this graphical representation of the model parameters is possible here because the model is linear.

Inspection of figure 3 (and especially the *red* channel bar graph, which displays how the model will react to the near-red ball colour) reveals that the robot will tend to turn to the left (negative rotational velocity) when a red object like the ball appears to the left of the camera image, and *vice versa*. Interestingly, by looking at the green channel bar graph we can also postulate that the robot will tend to turn away from a green-coloured object. This hypothesis is tested in section 3.4.

3.1. *Testing the model*

In order to test and validate the model systematically, we performed a dynamic and a static test. In the dynamic test the ball was moved continuously in the field of view of the robot while the model controlled the robot. During this test the translational velocity of the robot was clamped to 0.15 m/s. The test was run for approximately 5 minutes. During this time the rotational velocity of the robot and the full resolution images recorded by its camera were logged every 250 ms.

Figure 4 shows the average rotational velocity of the robot corresponding to the location of the ball in the (coarse-coded) image during the test run.

To quantify the response of the robot in relation to the location of the ball in the camera image, we computed the Spearman rank correlation coefficient between the angle-

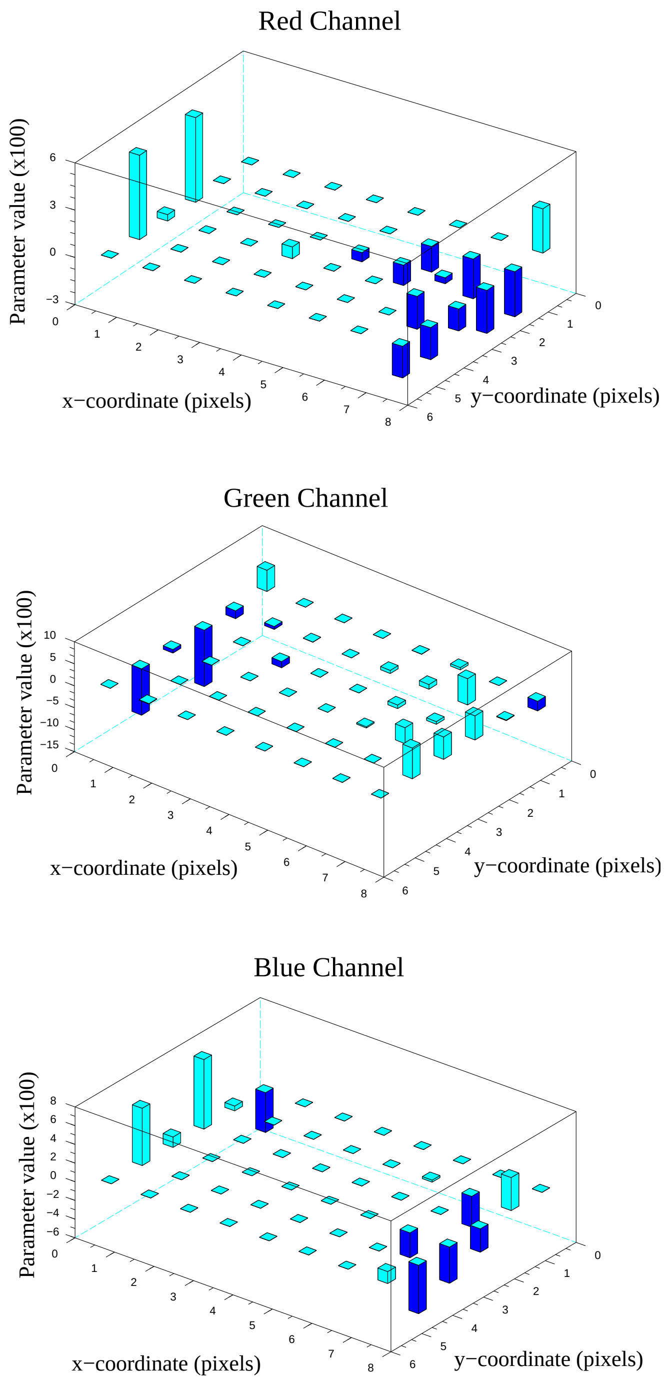

Fig. 3. The parameter values (Z axis, scaled by 100) of the model terms corresponding to the red, green and blue (top, middle and bottom respectively) channels of the coarse–coded camera image pixels. The XY plane of each graph corresponds to the coordinates of the coarse-coded image (i.e. location (0, 0) is the top-left corner of the image).

to-the-ball (from the robot's perspective) [1] and the robot's rotational velocity for the entire test run. This was found to be 0.63 (sig., $p<0.05$). This result demonstrates that there is significant correlation between the location of the ball and the rotational velocity of the robot.

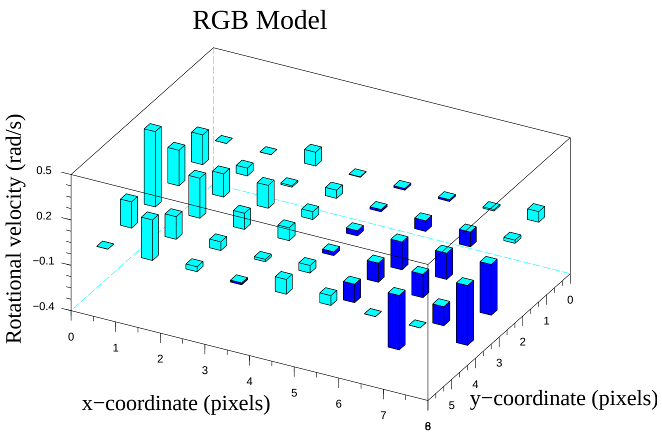

Fig. 4. The average rotational velocity (in rad/s) of the robot as a function of the location of the ball in the coarse–coded image during the dynamic testing of the model.

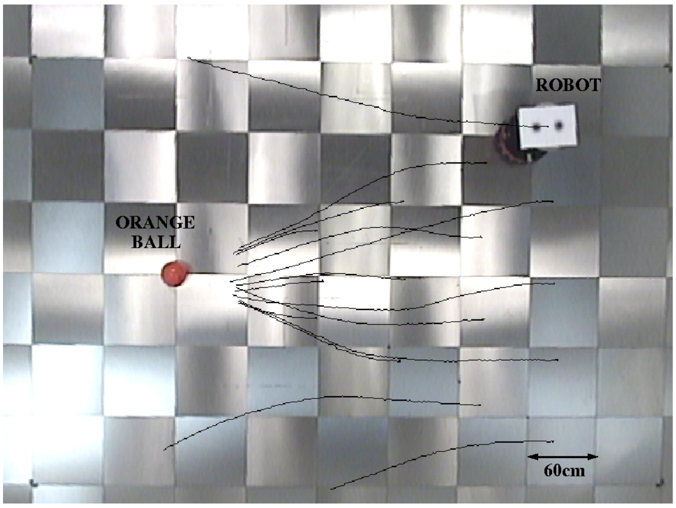

Fig. 5. The behaviour of the robot when being controlled by the model. The robot was started from 15 different locations relative to the ball.

In the static test of the model the robot was placed at different starting locations relative to the ball (always ensuring that the ball was in the robot's field of view at startup) before executing the model code. The translational velocity of the robot was again clamped to a constant 0.15 m/s during every test run. Figure 5 shows the behaviour of the robot from 15 different starting locations.

Figure 5 shows that the robot fails occasionally to home on the ball when starting from extreme side positions. We attributed this to the following reasons:

(i) The RGB colour space that was used to encode the input image is not very reliable when in cases where the illumination intensity varies (or similarly the video sensor automatically adjusts its gain to maintain constant image brightness).
(ii) There are only a few samples in the model estima-

[1] To find the angle-to-the-ball the images recorded were processed off-line by a hand-coded ball tracking program.

tion/validation data set where the ball appears in the top left and top right of the robot camera image.

(iii) The apparent ball size in the coarse-coded camera image is smaller than one pixel when the robot is furthest from the ball. This produces a weak input to the model, which results in a low output (i.e. rotational velocity).

3.2. *Improved input encoding*

To investigate the validity of these assumptions we obtained a second model, using the same estimation data, but representing the coarse-coded image using the *chromaticity* colour space which is less illumination dependent. First we normalised the RGB space

$$C_r = \frac{R}{R+G+B} \tag{1}$$

$$C_g = \frac{G}{R+G+B} \tag{2}$$

$$C_b = \frac{B}{R+G+B} \tag{3}$$

where C_r, C_g and C_b are the red chromaticity, green chromaticity and blue chromaticity components respectively, and R, G and B are the red, green and blue values respectively of the colour to be described. Then we divided red chromaticity and green chromaticity components by the blue chromaticity component in order to reduce the dimensionality of input space.

$$\hat{C_r} = \frac{R}{B} \tag{4}$$

$$\hat{C_g} = \frac{G}{B} \tag{5}$$

For this model we therefore used 96 integer inputs u_i ($0 \leq u_i \leq 255$). Again the model was chosen to be of first degree and no regression was used in the inputs and output (i.e. $l = 1$, $N_u = 0$, $N_y = 0$). The second model contained 51 terms:

$$\begin{aligned} \omega(n) = & \\ & -0.4437781579 \\ & +0.0956916362 * u_1(n) \\ & +0.4417766711 * u_2(n) \\ & +0.1789131625 * u_3(n) \\ & \ldots \\ & -0.4311080588 * u_{81}(n) \\ & +0.5639699608 * u_{88}(n) \\ & -0.4189265756 * u_{89}(n) \end{aligned}$$

where again $\omega(n)$ is the rotational velocity of the robot (in rad/s) at time instant n, and u_1 to u_{48} the red and u_{49} to u_{96} the green green chromaticity components of the coarse-coded image pixels.

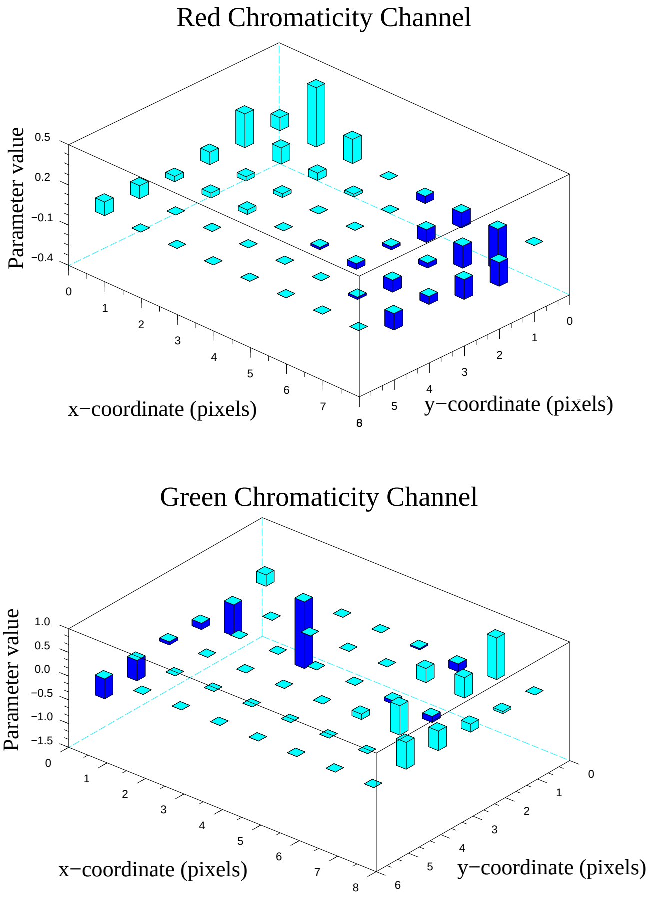

Fig. 6. The parameter values (Z axis) of the model terms corresponding to the red (top) and green (bottom) chromaticity components of the coarse-coded camera image pixels. The XY plane of each graph corresponds to the coordinates of the coarse-coded image.

Figure 6 shows the parameters of the model in two bar graphs (one for the red and one for the green chromaticity terms in the model).

Figure 6 shows an improvement with respect to model 1 (figure 3). The red chromaticity bar graph (which characterises the response to the image of the near-red orange ball) shows clear trends of increasing rotational velocity as the ball appears to the extreme left or right of the robot's image and it is nearly 0 when the ball appears in the centre of the image. The contribution of the red chromaticity to the rotational velocity also increases as the ball appears to the top of the image (when the ball is furthest from the robot).

3.3. *Evaluation of model 2*

Again the model was tested in the same way as model 1. The Spearman rank correlation coefficient between the angle-to-the-ball and the robot's rotational velocity im-

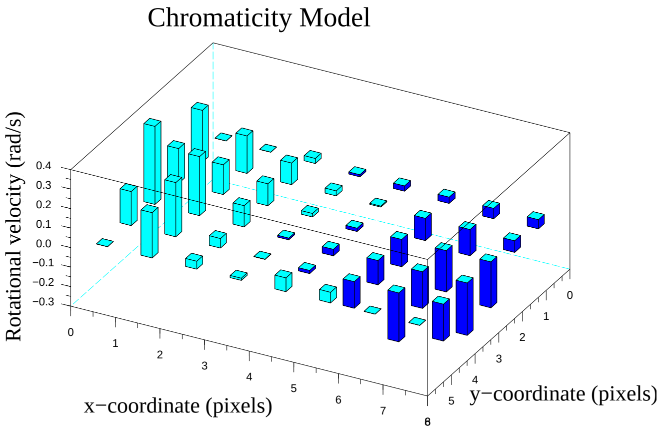

Fig. 7. The rotational velocity (in rad/s) of the robot as a function of the location of the ball colour in the coarse–coded image as this is predicted by the chromaticity-input model 2.

proved to 0.75 (sig., p<0.05) confirming that the chromaticity model is better.

The corresponding graph showing the average rotational velocity of the robot for each location of the ball in the (coarse-coded) image during the test is shown in figure 7.

Figure 9 shows the behaviour of the robot with the chromaticity-input model during the static test. As seen in figure 9 in this case the robot was successfully able to home on the ball in all 15 runs, even in cases where model 1 had failed.

To look for a significant difference between the trajectories of model 1 and model 2 in the static test (figures 5 and 9) we adapted the directness measure used by [Webb and Reeve, 2003] for cricket phototaxis. This was calculated as follows:

- From each trajectory, the x, y coordinates of the robot were extracted.
- Each successive pair of coordinates was used to define a vector with

$$distance_i = \sqrt{(x_{i+1} - x_i)2 + (y_{i+1} - y_i)2} \qquad (6)$$

$$heading_i = arctan(\frac{y_i}{x_i}) - arctan(\frac{y_i - y_{i+1}}{x_i - x_{i+1}}) \qquad (7)$$

- And the normalised mean vector for the trajectory was then calculated as:

$$magnitude = X2 + Y2 \qquad (8)$$

$$angle = arctan(\frac{Y}{X}) \qquad (9)$$

where

$$X = \frac{\sum distance_i * cos(heading_i)}{length} \qquad (10)$$

$$Y = \frac{\sum -distance_i * sin(heading_i)}{length} \qquad (11)$$

$$length = \sum_i distance_i \qquad (12)$$

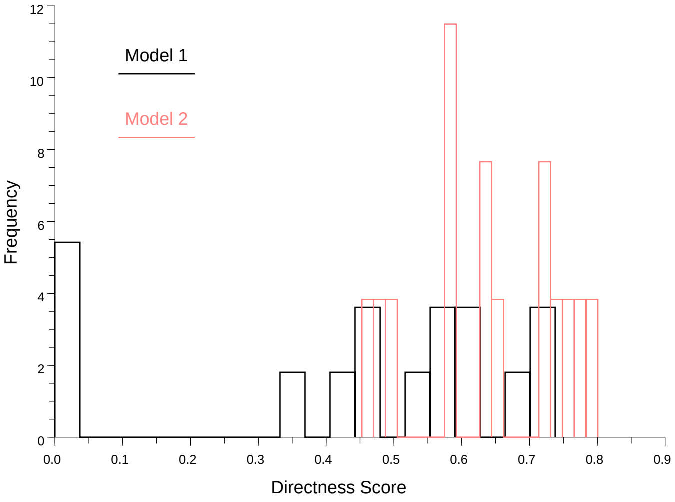

Fig. 8. The directness score D distribution of RGB-based model 1 and Chromaticity-based model 2. Model 2 is significantly more direct than model 1, with a median directness of 0.64 (confidence interval [0.57,0.74] at the 5% significance level, U-test), compared to model 1's median directness 0.54 (confidence interval [0.37,0.62]).

The *angle* of normalised mean vector indicates the average heading of the robot relative to the ball during the trajectory, and the *magnitude* is a measure of the amount of variance around that direction, such that the mean vector of *angle* = 0 and *magnitude* = 1 would indicate a completely direct path to the ball from the starting position.

- Overall directness is then scored as:

$$D = magnitude * cos(angle) * S \qquad (13)$$

which varies from 1 to 0 as the robot deviates more from the shortest path to reach the ball. This combines the *cosine* of the *angle* of the mean vector (which varies from 1 to 0 as the robot deviates from heading towards the ball), the *length* of the mean vector (which varies from 1 to 0 as the robot deviates more around the mean angle) and S measure corresponding to the success of the robot in tracking the ball along the trajectory (which is 1 if the robot is able to home on the ball and 0 if it looses the ball from its sight.)

For each model we computed 15 directness scores D, one for each trajectory. The results showed that model 2 is significantly more direct than model 1 in reaching the ball (U-test, p<0.05, figure 8).

3.4. *Hypothesis postulation and testing*

Having a transparent model like the one in figure 6 has many advantages, such as the possibility of generating testable hypotheses. For instance, by looking at the green chromaticity bar graph of figure 6, we postulated that the robot would turn away from a green coloured object. This hypothesis was tested using the static test with a green ball. The resulting trajectories of the robot from 15 different positions are shown in figure 10. It is interesting to observe that when the ball falls on the centre vertical line of the robot's view, the robot does not steer away from the

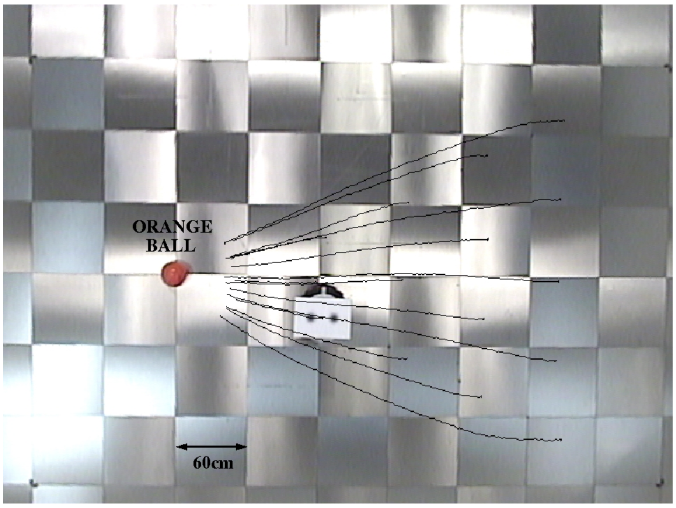

Fig. 9. The behaviour of the robot when running the chromaticity-input model. The robot was started from 15 different locations relative to the ball as in figure 5.

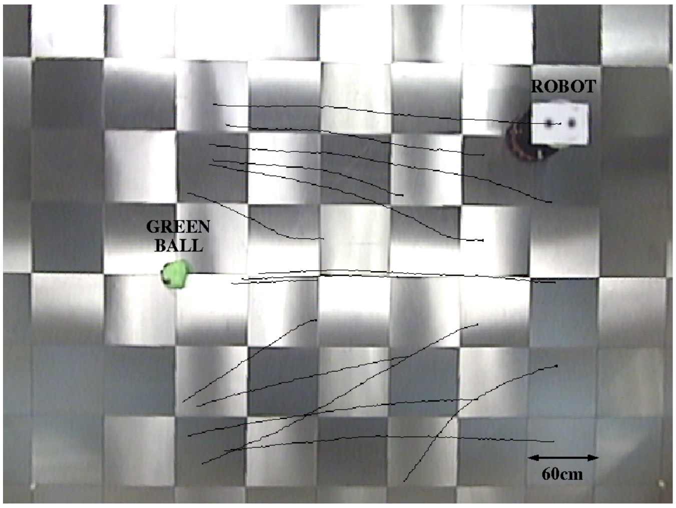

Fig. 10. The behaviour of the robot facing a green-coloured ball when running the chromaticity-input model. The robot was started from 15 different locations relative to the ball as in the previous static tests.

ball, but rather moves passively towards it. This behaviour is justified by realising that there is no rotational velocity bias when the green ball is in the middle of the input image (see green chromaticity bar graph of figure 6). In all other cases we can clearly see that the robot steers away from the green ball.

Another hypothesis was also tested. This time we postulated that by swapping the red chromaticity inputs with the green chromaticity ones the robot would *follow* the green ball. Again this was tested using the static test with the green ball. Figure 11 confirms our expectations.

Using the directness measure described above, the behaviour of the robot facing a green-coloured ball running the new model was compared with the behaviour of the robot facing an orange-coloured ball running the model 2. The U test result showed that there is no significant difference between the two model's trajectories ($p>0.05$).

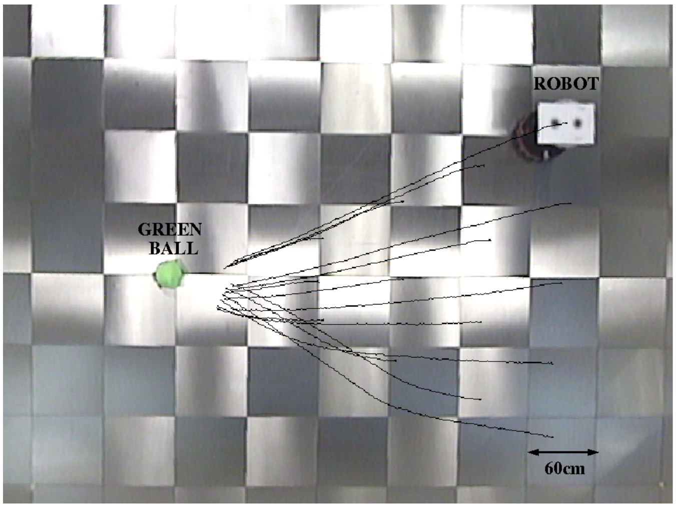

Fig. 11. The behaviour of the robot facing a green-coloured ball when running the chromaticity-input model but with the parameters of the green and red chromaticities swapped.

Following the same reasoning therefore, the task can be adapted so that the robot can follow a ball of any particular colour by normalising the chromaticity inputs by that colour. This conclusion follows directly from the analysis of our ARMAX model and obviously the fact that in a linear model all inputs are orthogonal. In other words: the system identification procedure described in this paper can be used not only to obtain robot control code, but also to achieve a *range* of different behaviours based on theoretical considerations.

4. Conclusions

We have shown how the NARMAX modelling approach can be used to identify a simple vision-based task. In our experiments, a linear model was sufficient, but our identification approach is not limited to linear models. Obtaining a task-achieving controller through system identification is very efficient despite using complex input from a vision sensor code was ready to run within a few hours.

The task investigated in this paper could have been achieved using other machine learning approaches, such as supervised artificial neural networks (MLP, RBF, LVQ, ...) or support vector machines. However, these approaches tend to be slow in learning, especially when using large input spaces and, more importantly, generate opaque models that are difficult to visualise and analyse.

In contrast, our modelling approach produces transparent mathematical functions that can be directly related to the task. This allows for predictions to be made about the behaviour of the robot executing the models without actually evaluating the output of the input space. In the example presented here the function of the two models obtained can be predicted by only looking at the parameters of the model equations. Furthermore, understanding the workings of the task by looking at the polynomial model allowed us to change the task in a predictable manner in

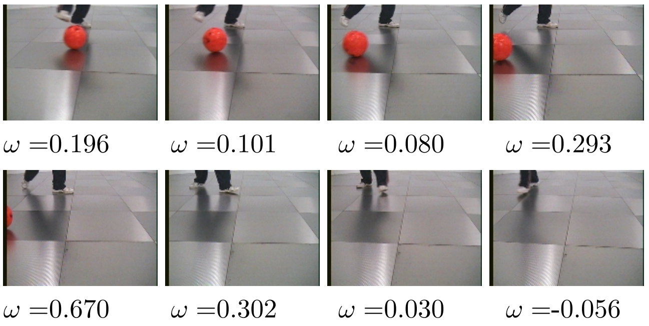

Fig. 12. Stream of eight successive images during the testing of a model with lag 8 (2 seconds) that shows the ball gradually disappearing to the left, and the corresponding rotational velocity of the robot (positive rotational velocity = "left turn").

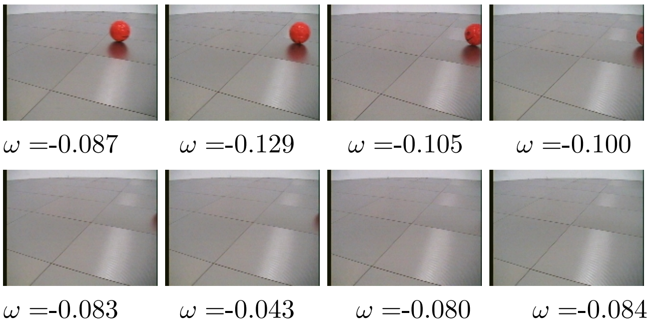

Fig. 13. Stream of eight successive images during the testing of a model with lag 8 (2 seconds) that shows the ball gradually disappearing to the right, and the corresponding rotational velocity of the robot (negative rotational velocity = "right turn").

order to achieve a different behaviour.

Another important advantage of the NARMAX modelling methodology is that it does not suffer from the "pitfall" of local minima in the calculation of the model parameters. Through a single optimal set of parameters we *always* obtain the estimation of the model. This increases the efficiency of the estimation process, by removing the need to use mechanisms such as "momentum" or "inertia" to avoid sub-optimal solutions.

The RGB colour space is not reliable for colour discrimination applications in *variable* illumination intensity environments [Land, 1983]. This has been demonstrated one more time with our experiments. The obtained RGB model was not good enough to make the robot home on the ball when the ball appears to the extreme left or right of the robot's image. Using an less illumination-dependent colour encoding solved this problem.

4.1. *Future work*

Apart from the models presented above, we have also obtained models with different lags (i.e. incorporating inputs presented to the system earlier than the current time n). Such lags in the model represent "memory", and our hypothesis was that the robot would keep turning in the "right" direction even if the ball would leave the camera's field of view.

We tested this hypothesis experimentally. Tables 12 and 13 show two streams of images obtained during the testing of one such model with lag 8 (i.e. incorporating terms with inputs presented in the last 2 seconds). Below each image is the rotational velocity of the robot after the processing of the corresponding image. These results demonstrate that clearly this "memory effect" is detectable in certain situations. In fact, throughout the entire testing session (approximately 30min) the robot showed a tendency to follow the ball after it had left the image (as the two examples of the tables show). However, statistical analysis of the entire test data did not produce conclusive evidence that the robot will *always* follow a disappearing ball correctly. We believe the reason for this lies in the fact that a linear model is not able to express relationships between different input channels or different time instants of one or more input channels. As an example, consider the ball rolling from the right to the left of the camera image. This will produce inputs, some of which will "encourage" right turning (negative ω) and some left turning (positive ω). The resulting rotational velocity will therefore be the sum of these contributions and its value will depend on the speed of the ball and the lag of the model. This may well result in the robot wrongly turning to the right if, for example, the ball appeared longer to the right during its path.

We therefore believe that in cases where task-specific information is contained in multiple input channels or multiple successive time-frames (such as the example above), a non-linear NARMAX model would produce better results. This hypothesis is subject to ongoing research at the University of Essex.

Acknowledgements

We thank the following bodies for their support in this work: The RobotMODIC project is supported by the Engineering and Physical Sciences Research Council under grant GR/S30955/01. Roberto Iglesias is supported through the research grants PGIDIT04TIC206011PR, TIC2003-09400-C04-03 and TIN2005-03844.

References

[Billings and Chen, 1998] Billings, S. and Chen, S. (1998). The determination of multivariable nonlinear models for dynamical systems. In Leonides, C., (Ed.), *Neural Network Systems, Techniques and Applications*, pages 231–278. Academic press.

[Billings and Voon, 1986] Billings, S. and Voon, W. S. F. (1986). Correlation based model validity tests for non-linear models. *International Journal of Control*, 44:235–244.

[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, 1974] Eykhoff, P. (1974). *System Identification: parameter and state estimation.* Wiley-Interscience, London.

[Iglesias et al., 2004] Iglesias, R., Kyriacou, T., Nehmzow, U., and Billings, S. (2004). Task identification and characterisation in mobile robotics. In *Proc. of TAROS 2004, (Towards Autonomous Robotic Systems)*. Springer Verlag.

[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 *Proc. of ECMR 05 - European Conference on Mobile Robots 2005.* Springer Verlag.

[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.

[Land, 1983] Land, E. H. (1983). Recent advances in retinex theory and some implications for cortical applications: Color vision and the nature of the image. In *Proc. of Nat. Acad. Sci.*, volume 80, pages 5163–5169, USA.

[Nehmzow et al., 2005] Nehmzow, U., Kyriacou, T., Iglesias, R., and Billings, S. (2005). Self-localisation through system identification. In *Proc. of ECMR 05 - European Conference on Mobile Robots 2005.* Springer Verlag.

[Roberto Iglesias and Billings, 2005]
Roberto Iglesias, Ulrich Nehmzow, T. K. and Billings, S. (2005). Modelling and characterisation of a mobile robot's operation. Santiago de Compostela, Spain.

[Webb and Reeve, 2003] Webb, B. and Reeve, R. (2003). Reafferent or reduntant: How should a robot cricket use an optomotor reflex? In *Adaptive Behaviour*, volume 11, pages 137–158.