Robotics and Autonomous Systems 56 (2008) 1061–1067

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Model identification and model analysis in robot training

R. Iglesias [b], U. Nehmzow [a,*], S.A. Billings [c]

[a] Intelligent Systems Research Centre, University of Ulster, UK
[b] Electronics & Computer Science, University of Santiago de Compostela, Spain
[c] Automatic Control and Systems Engineering, University of Sheffield, UK

ARTICLE INFO

Article history:
Available online 20 September 2008

Keywords:
Mobile robotics
Robot training
System identification
Narmax
Robot programming

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 [1] 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 and 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 [2]. 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) [3,1], or predict the robot's response to particular inputs [4].

* Corresponding author. Tel.: +44 28 7137 5059; fax: +44 2871 357 470.
E-mail address: u.nehmzow@ulster.ac.uk (U. Nehmzow).

0921-8890/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2008.09.003

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 [5] or NARMAX [6] 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 [6], and examples of robotics applications are given in our previous publications [7].

The NARMAX modelling approach is a parameter estimation methodology for identifying the important model terms and associated parameters of unknown nonlinear dynamic systems.

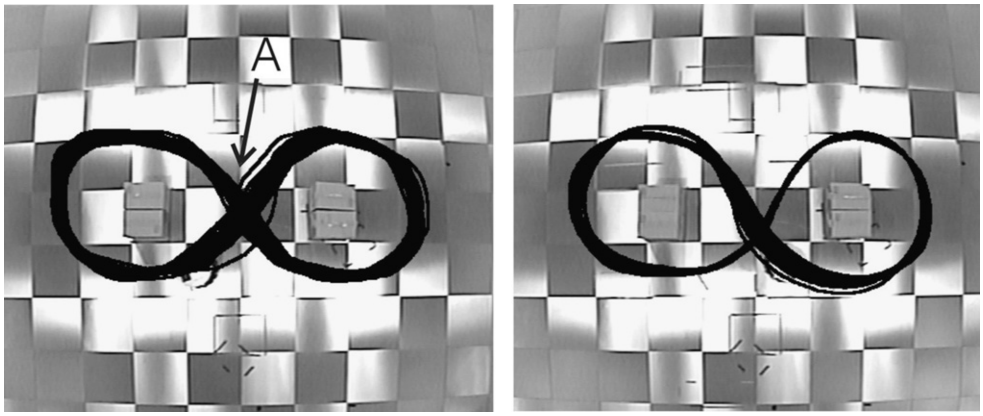

Fig. 1. Left: Robot trajectory under manual control, used to obtain training data. Right: Trajectory taken under control of the obtained model given in Table 1.

For multiple input, single output noiseless systems this model takes the form:

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

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, and d is the input dimension. $f()$ is a nonlinear 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 nonrelevant terms, the Error Reduction Ratio (ERR) [8] 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.

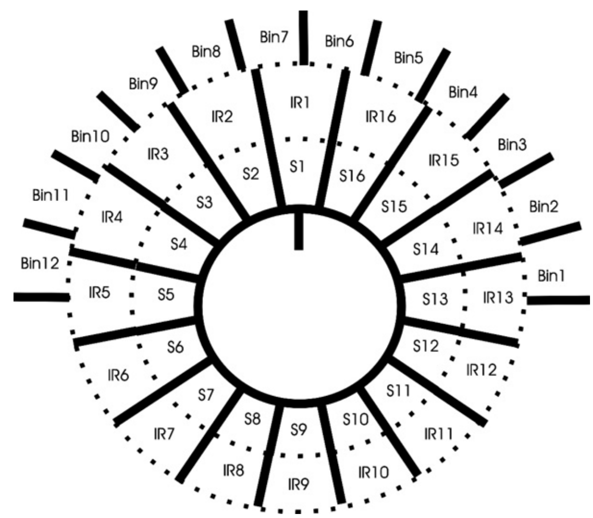

Fig. 2. Location of each sonar and infrared sensor in the Magellan Pro Robot we used in our experiments. The laser sensors have been averaged in twelve sectors of 15 degrees each (laser bins).

3. Route learning by demonstration

We applied our robot training strategy to program a reactive route following controller (Fig. 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 Fig. 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 (Fig. 1, left). During this stage sensor perceptions (Fig. 2), position, transitional and rotational velocities were recorded every 250 ms.

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 (Fig. 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 (Fig. 1, right).