Robotics and Autonomous Systems 58 (2010) 265–275

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Towards modelling complex robot training tasks through system identification

U. Nehmzow [a,*], O. Akanyeti [b], S.A. Billings [c]

[a] School of Computing and Intelligent Systems, University of Ulster, UK
[b] Department of Computer Science, University of Essex, UK
[c] Department of Automatic Control and Systems Engineering, University of Sheffield, UK

ARTICLE INFO

Article history:
Available online 5 December 2009

Keywords:
Robot programming
Robot training
System identification
NARMAX applications to robotics

ABSTRACT

Previous research has shown that sensor–motor tasks in mobile robotics applications can be modelled *automatically*, using NARMAX system identification, where the sensory perception of the robot is mapped to the desired motor commands using non-linear polynomial functions, resulting in a tight coupling between sensing and acting — the robot responds *directly* to the sensor stimuli without having internal states or memory.

However, competences such as for instance sequences of actions, where actions depend on each other, require memory and thus a representation of state. In these cases a simple direct link between sensory perception and the motor commands may not be enough to accomplish the desired tasks. The contribution of this paper to knowledge is to show how fundamental, simple NARMAX models of behaviour can be used in a bootstrapping process to generate *complex* behaviours that were so far beyond reach.

We argue that as the complexity of the task increases, it is important to estimate the current state of the robot and integrate this information into the system identification process. To achieve this we propose a novel method which relates distinctive locations in the environment to the state of the robot, using an unsupervised clustering algorithm. Once we estimate the current state of the robot accurately, we combine the state information with the perception of the robot through a bootstrapping method to generate more complex robot tasks: We obtain a polynomial model which models the complex task as a function of predefined low level sensor–motor controllers and raw sensory data.

The proposed method has been used to teach *Scitos G5* mobile robots a number of *complex* tasks, such as advanced obstacle avoidance, or complex route learning.

1. Introduction

Fundamentally, the behaviour of a robot is a result of the interaction of three factors: (i) the robot's hardware, (ii) the robot's controller, and (iii) the environment the robot is operating in. The robot acquires information from the environment through its sensors, which provides the input signals to the controller. The controller computes the desired motor commands and the robot performs these commands in the environment to achieve the desired task [1].

Given that sensing and the actions of a robot are coupled dynamically, given the sensitivity of robot sensor's to slight changes in the environment, the robot–environment interaction exhibits complex, non-linear, often chaotic and usually unpredictable characteristics [2,3]. Because of this, the task of robot programming – designing a control program to achieve a desired behaviour – is difficult. Unlike other engineering disciplines, there is no formal, theory-based design methodology which the robot programmer can follow to program a robot to achieve a desired task.

Nevertheless, we have previously shown that the robot programming process *can* be automated: sensor–motor competences in mobile robotics applications can be modelled automatically and algorithmically, using robot training and system identification methods. The stages of our method are summarized below:

(i) Acquisition of a training data set. First the programmer demonstrates the desired behaviour to the robot via driving it manually [4,5] or direct human demonstration [6,7]. During this run, sensory perception and the desired velocity commands of the robot are logged. There is a considerable corpus of robotics research on robot training, for example training by verbal instructions [8], using expectations [9] or imitation of a human trainer [10]. This work on robot training is relevant to the experiments presented here, in that the same method of acquiring training data is used, but the focus of our

* 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
doi:10.1016/j.robot.2009.11.002

experiments is not on training, but on automatically obtaining low level behaviours and, again automatically, combining these into more complex behaviours, without the need to have dedicated robot programming skills.

(ii) Preprocessing of input signals. Having thus obtained the raw training data, we preprocess the input signals to reduce the dimensionality of the input space [11] and also to identify the important sensory readings, which are highly correlated with the desired motor commands [12].

(iii) Model estimation. We then model the relationship between the encoded sensory perception and the actions of the robot using ARMAX (Auto-Regressive Moving Average models with eXogenous inputs) [13,14] and NARMAX (Non-linear ARMAX) [15,16] system identification methods. These techniques are supervised parameter estimation methodologies for identifying both the important model terms and the parameters of unknown non-linear dynamic systems. They produce linear or non-linear polynomial functions to model the input–output relationship. A single model is usually enough to identify the whole relationship successfully.

(iv) Model validation and optimization. Once the sensor-based controllers are obtained, they are used to drive the robot in the target environment to validate their performances. Also at this stage, it is sensible to carry out sensitivity analysis [17,18] in order to estimate the influence of individual sensor readings upon the robot's global behaviour [19,6]. This would help us to determine which parameters in the model contribute the most to output variability and which parameters are insignificant and can be eliminated from the final model, leading to more parsimonious models.

(v) Analytical analysis of the obtained models. The representation of the task as a transparent, analysable polynomial model simplifies the identification of the important factors that affect the robot's behaviour. For instance, the error reduction ratio gives an indication of the importance of individual model terms. Likewise, variance-based methods of sensitivity analysis [18,20,21] or entropy-based methods [22] allow the identification of important input components (e.g. sensors).

1.1. Motivation: From simple to complex tasks

The method described above has been successfully applied to generate various sensor–motor tasks, from simple behaviours, such as wall following [4] or door traversal [19], to some complicated behaviours, such as following a moving object [23] and path learning [11].

However as the complexity of task increases, representing the whole relationship between sensory perception and the desired motor responses of the robot in one single model using only raw sensory inputs would lead to large models. Training such models is extremely difficult, and obtained models often exhibit brittle performance.

The novel contribution of this paper is to show how the NARMAX system identification method can be used to model more *complex* robot training tasks, such as tasks where sensor–motor couplings change along a path, or depending on circumstance. To do so, we focus on two fundamental ideas:

(i) For complex tasks, the actions of the robot depend not only on raw sensory perception, but also on the current state of the robot. Therefore there is a need to represent the present state of the robot, and to incorporate it into the model.

(ii) As our goal is to simplify the robot programming process such that non-programmers can generate robot control code, there is still need for a simple method to generate the motor commands that take the robot from one state to another, accomplishing the desired task.

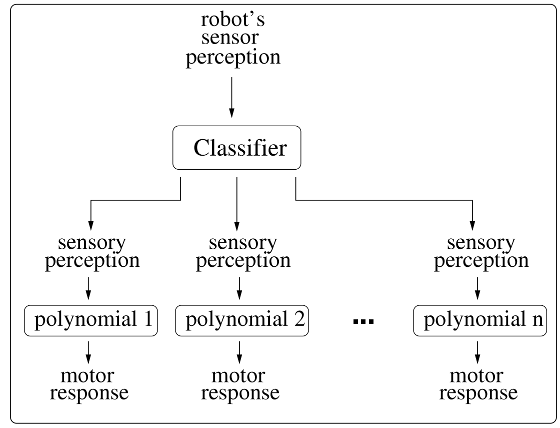

Fig. 1. The proposed method to cope with the state transition problem while generating robot control programs: a classifier divides the perception–action space of the robot into subspaces, and generates a separate model for each subspace.

In this paper, we address both issues with a general overlook. In Section 2 we focus on the state transition problem, and propose a novel method relating the state of the robot to distinctive objects seen in the environment: First, the robot learns to recognize landmarks in the environment, using standard classification techniques. Once the robot is capable of localising, using these landmarks, it obtains a different sensor–motor coupling for each recognized landmark.

After estimating which state the robot is in, the next step is to combine the state information with the perception of the robot in a general framework to generate the essential motor commands in order to accomplish the desired complex robot training tasks.

In Section 3, we therefore introduce a bootstrapping method of generating complex robot training tasks using polynomial NARMAX models. The method is based on obtaining hierarchical polynomial models which model the desired task by combining predefined low level sensor–motor controllers, raw sensory data and state inputs.

2. State estimation through unsupervised learning

In complex tasks it is often the case that the relationship between perception and the motor response varies along the robot's path. We deal with this situation, using two stages:

In the first stage the robot clusters the environment in to subspaces using standard classification techniques (SOM, K-means, etc.) based on its own sensory perception. Note that here we assume that state transitions can be observed by the robot through its sensors. Then in the second stage it obtains a model for each cluster separately using system identification techniques (Fig. 1).

With this method, state transitions are related to robot–environment interaction, which allows the robot to identify the state changes automatically, using its own perception. In this paper, the K-means algorithm is used as a classifier, described in Section 2.1.

It might be instructive at this point to refer briefly to the work done on simultaneous localisation and mapping (SLAM). SLAM focusses on precise robot localisation, using adaptive filtering techniques such as Kalman filters and Bayesian methods [24], and is a more sophisticated method of robot self-localisation than simple clustering of a robot's sensory perception. We use the clustering to divide the input space to our system identification process, not to localise *precisely*, i.e. we do not perform SLAM in the experiments presented here.

2.1. The K-means classifier

The k-means algorithm [25] is an unsupervised clustering algorithm which is used to classify a given data set into k clusters.