Engineering Applications of Artificial Intelligence 29 (2014) 70–78

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Automatic extraction of the fuzzy control system by a hierarchical genetic algorithm

Jérôme Mendes *, Rui Araújo, Tiago Matias, Ricardo Seco, Carlos Belchior

Institute of Systems and Robotics (ISR-UC), and Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal

A R T I C L E I N F O

Article history:
Received 2 November 2012
Received in revised form
31 October 2013
Accepted 25 December 2013
Available online 24 January 2014

Keywords:
Fuzzy control
Hierarchical genetic algorithm
Activated sludge process
Fuzzy knowledge base

A B S T R A C T

The paper proposes a new method to automatically extract all fuzzy parameters of a Fuzzy Logic Controller (FLC) in order to control nonlinear industrial processes. The main objective of this paper is the extraction of a FLC from data extracted from a given process while it is being manually controlled. The learning of the FLC is performed by a hierarchical genetic algorithm (HGA), from a set of process-controlled input/output data. The algorithm is composed by a five level structure, being the first level responsible for the selection of an adequate set of input variables. The second level considers the encoding of the membership functions. The individual rules are defined on the third level. The set of rules are obtained on the fourth level, and finally, the fifth level selects the elements of the previous levels, as well as, the *t*-norm operator, inference engine and defuzzifier methods which constitute the FLC. To optimize the proposed method, the HGA's initial populations are obtained by an initialization algorithm. This algorithm has the main goal of providing a good initial solution for membership functions and rule based populations, enhancing the GA's tuning. Moreover, the HGA is applied to control the dissolved oxygen in an activated sludge reactor within a wastewater treatment plant. The results are presented, showing that the proposed method extracted all the parameters of the fuzzy controller, successfully controlling a nonlinear plant.

1. Introduction

Fuzzy control systems (FCSs) have been used for a wide variety of industrial systems and consumer products, attracting the attention of many researchers. Fuzzy logic controllers (FLCs) are rule-based systems which are useful in the context of complex ill-defined processes, especially those which can be controlled by a skilled human operator without any mathematical knowledge of the process's underlying dynamics (Herrera et al., 1995). FLCs are based on a set of fuzzy control rules that make use of people's common sense and experience. However, there still exist many difficulties in designing fuzzy systems to solve certain complex nonlinear problems.

In general, it is not easy to determine the most suitable fuzzy rules and membership functions to control the output of a plant, when the only available knowledge concerning the process is the empirical information transmitted by a human operator. Thus, a major challenge in current fuzzy control research is translating human empirical knowledge into FLCs. A possible candidate to meet this challenge is the application of the genetic algorithm (GA) approach to data extracted from a given process while it is being manually controlled.

GA's have been successfully applied to a wide variety of applications over the years. In particular, these algorithms have been applied in many automatic control problems, such as the development and tuning of FLCs. For that matter, they have been previously employed to select adequate sets of membership functions and fuzzy rules. Sharkawy (2010) developed a self-tuning PID control scheme with an application to antilock braking systems (ABSs) via combinations of fuzzy and genetic algorithms. Ali and Ramaswamy (2009) present an optimal fuzzy logic control algorithm for vibration mitigation of buildings using magneto-rheological (MR) dampers. A micro-genetic algorithm (m-GA) and a particle swarm optimization (PSO) are used to optimize the FLC parameters. Alam and Tokhi (2008) proposed a GA-based hybrid fuzzy logic control strategy for input tracking and vibration reduction at the end point of a single-link flexible manipulator. For that matter, a GA is used to extract and optimize the rule base of the fuzzy logic controller. In Homayouni et al. (2009), a genetic fuzzy logic control methodology is used to develop two production control architectures: genetic distributed fuzzy (GDF) and genetic supervisory fuzzy (GSF) controllers. The GA is used to tune the input variable membership functions for the GSF and GDF controllers. Coban and Can (2010) designed a trajectory tracking genetic fuzzy logic controller for research reactors. Membership function boundaries and fuzzy control rule action weights were optimally determined by GAs.

The cited methods only optimize membership function parameters considering the other components of the fuzzy system fixed, such as implication, aggregation and defuzzifier methods. Other common limitation is the selection of the correct set of input variables. The variable selection process is usually manual and not accompanied with the accurate selection of the right time delays, probably leading to low-accuracy results. A variable with the

* Corresponding author.
E-mail addresses: jermendes@isr.uc.pt (J. Mendes), rui@isr.uc.pt (R. Araújo), tmatias@isr.uc.pt (T. Matias), cabelchior@isr.uc.pt (C. Belchior).

0952-1976/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.12.012

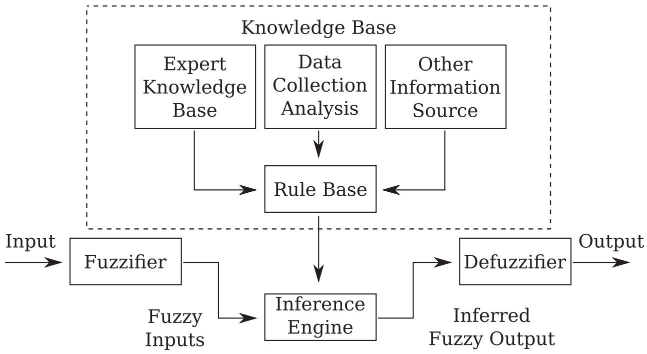

Fig. 1. Basic configuration of the fuzzy logic systems.

correct delay may contain more information about the output, than one which does not consider any delay (Souza et al., 2010).

Delgado et al. (2001) introduced a hierarchical genetic algorithm (HGA) to optimize the parameters of Takagi–Sugeno fuzzy systems from available input/output data by means of a coevolutionary genetic algorithm and a neuro-based technique. Moreover, as an improvement of this methodology, Delgado et al. (2009) added a methodology for pre-selection of variables using an auxiliary criterion. However, the variable and delay selection are not jointly performed with the learning of the fuzzy model, which precludes the global optimization of the prediction setting.

The method proposed in this paper is based on the application of the HGA suggested by Delgado et al. (2009), although, applied to controller design. The main advances and differences contemplated in this work are the improvement of the whole hierarchical structure, automatically extracting the control fuzzy rules. First, a new hierarchical level, responsible for the selection of an adequate set of input variables and respective time delays is added; then the T–S fuzzy model approach (for an identification problem) was replaced by a standard fuzzy control system, i.e., the consequent part of the rules is represented by a membership function, rather than the T–S model proposed by Delgado et al. (2009); finally, an initialization algorithm based on Andersen and Tsoi (1997) is integrated to initialize the populations of the HGA, improving the convergence time.

Moreover, in order to validate and demonstrate the performance and effectiveness of the proposed algorithm, the control of the dissolved oxygen in an activated sludge reactor within a wastewater treatment plant (WWTP) is studied. WWTPs are large and complex non-linear systems subject to large disturbances in influent flow rate and pollutant load, together with uncertainties concerning the composition of the incoming wastewater (Belchior et al., 2012). In Belchior et al. (2012), a standard fuzzy controller is manually developed and applied to control the dissolved oxygen in an activated sludge reactor within a WWTP. Using this standard fuzzy controller, a learning dataset was obtained and then the proposed HGA is off-line applied with the aim of achieving a fuzzy controller with a response similar to the manually designed one. Once the fuzzy controller parameters are determined, the controller is applied to the same plant mentioned before, considering different control references, and comparing the results with the ones previously obtained using the standard FLC. The results show that the proposed method is able to extract all parameters of the fuzzy controller, enabling the successful control of the nonlinear plant.

The paper is organized as follows. Section 2 introduces the fuzzy logic system. The proposed HGA is described in Section 3. The HGA application and respective results are presented and analyzed in Section 4. Finally, remarks and conclusions are made in Section 5.

2. Fuzzy system

This section briefly overviews the main concepts of fuzzy control systems (Wang, 1997). A fuzzy system is a knowledge-based system defined by a group of *IF–THEN* rules, which can be used to implement fuzzy controllers. The following example illustrates such a rule:

$$\text{IF the } \textit{temperature} \text{ is } \textit{cold}, \quad \text{THEN turn } \textit{down} \text{ the } \textit{speed} \text{ of the fan,} \tag{1}$$

where *temperature* and *speed* are the input and output variables, respectively. These variables are characterized by the fuzzy sets A, through a mapping defined by $\mu_A(x) = \mathbf{U} \rightarrow [0, 1]$, for which, *cold* and *down*, are referred to as linguistic terms. $\mathbf{U}$ is the universe of discourse of the variable.

Fuzzy systems are constituted by a group of four main elements: knowledge base, fuzzifier, fuzzy inference engine and defuzzifier, as shown in Fig. 1.

The knowledge base is composed by a set of N fuzzy IF–THEN rules R_j in the generic form

$$R_j : \text{IF } x_1 \text{ is } A_1^j, \text{ and } \ldots \text{ and } x_n \text{ is } A_n^j \quad \text{THEN } u \text{ is } B_j, \tag{2}$$

where $j = 1, 2, \ldots, N$; x_i $(i = 1, 2, \ldots, n)$ are the input variables of the fuzzy system, u is the output variable and A_i^j and B_j are the linguistic terms characterized by the fuzzy membership functions $\mu_{A_i^j}(x)$ and $\mu_{B_j}(u)$, respectively.

The fuzzifier is the fuzzy system element responsible for mapping the real values of the input linguistic variables, x, into the corresponding fuzzy sets described by membership functions X. In this paper, the only utilized fuzzifier is the singleton fuzzifier (Wang, 1997).

The next element, the fuzzy inference engine (FIE), uses the collection of fuzzy IF–THEN rules to map the input fuzzy set X into the rule consequent fuzzy sets B_j. The collection of fuzzy outputs of the rules are then combined into an overall inferred fuzzy output U. In this paper, to process the antecedent part of the rule, only propositions connected by the fuzzy AND operator (T-norms) are considered.

Finally, the defuzzifier is responsible for mapping a fuzzy set U into a real-valued output, u^*.

3. Hierarchical genetic algorithm

This section describes the proposed algorithm. The main goal of this work is to develop a FLC that does not require prior explicit expert knowledge. To do so, the work proposes an automatic method based on a HGA for the extraction of all fuzzy parameters of a FLC from a dataset obtained from an existing controller (human or automatic).

Some biologically inspired algorithms, such as genetic algorithms (GAs), ant colony optimization (ACO), and particle swarm optimization (PSO), have been proved efficient in optimization problems. GAs are search methods that are inspired on natural evolution, selection, and survival of the fittest in the biological world. PSO is inspired in the social behavior of living organisms such as bird flocking or fish schooling. ACO is a multiagent approach that simulates the foraging behavior of ants. All algorithms could be used to design the T–S fuzzy models. However, because GAs provide a robust search with the ability to find near optimal solutions in complex and large search spaces (Cordón et al., 2001; Herrera, 2008), GAs are a useful soft computing technique to design T–S fuzzy models. Other advantages in the use of GAs in the design of T–S fuzzy models are GAs are simple to implement, they have the possibility of using different types of solution encoding (e.g. for different parts of the model), and they are adaptive, which means that they have the ability to learn, accumulating relevant knowledge to solve optimization problems (Kasabov, 1996).

A HGA will be used instead of a GA with just one optimization level due to the complexity of the problem. It is well known that computation, search, and optimization problems become more difficult to solve when the dimensionality increases (curse of dimensionality), and, therefore, when more complex design decisions involving a large number of parameters must be made, a