5 lucrări ştiinţifice în extenso

1. R.-E. Precup and H. Hellendoorn, A survey on industrial applications of fuzzy control, **Computers in Industry** (Elsevier Science), vol. 62, no. 3, pp. 213-226, 2011, impact factor (IF) = 1.529, IF according to 2019 Journal Citation Reports (JCR) released by Clarivate Analytics in 2020 = 3.954, **Highly Cited Paper according to Clarivate Analytics Web of Science** as of September/October 2019 (http://www.aut.upt.ro/~rprecup/CiI_2011_Highly_Cited_Paper.png).

2. R.-E. Precup, R.-C. David and E. M. Petriu, Grey Wolf Optimizer Algorithm-Based Tuning of Fuzzy Control Systems with Reduced Parametric Sensitivity, **IEEE Transactions on Industrial Electronics**, vol. 64, no. 1, pp. 527-534, 2017, impact factor (IF) = 7.050, IF according to 2019 Journal Citation Reports (JCR) released by Clarivate Analytics in 2020 = 7.515, **Highly Cited Paper according to Clarivate Analytics Web of Science** as of September/October 2019 (http://www.aut.upt.ro/~rprecup/TIE_2017_Highly_Cited_Paper.png).

3. R.-E. Precup, M.-B. Rădac, R.-C. Roman and E. M. Petriu, Model-Free Sliding Mode Control of Nonlinear Systems: Algorithms and Experiments, **Information Sciences** (Elsevier Science), vol. 381, pp. 176-192, 2017, impact factor (IF) = 4.305, IF according to 2019 Journal Citation Reports (JCR) released by Clarivate Analytics in 2020 = 5.910, **Highly Cited Paper according to Clarivate Analytics Web of Science** as of May/June 2018 (http://www.aut.upt.ro/~rprecup/InfSci_2017_Highly_Cited_Paper.jpg).

4. R.-E. Precup, R.-C. David, E. M. Petriu, M.-B. Rădac, St. Preitl and J. Fodor, Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems, **Knowledge-Based Systems** (Elsevier Science), vol. 38, pp. 74-84, 2013, impact factor (IF) = 3.058, IF according to 2019 Journal Citation Reports (JCR) released by Clarivate Analytics in 2020 = 5.921, **Highly Cited Paper according to Clarivate Analytics Web of Science** as of November/December 2015 (http://www.aut.upt.ro/~rprecup/KBS_2013_Highly_Cited_Paper.jpg).

5. R.-E. Precup, P. Angelov, B. S. J. Costa and M. Sayed-Mouchaweh, An overview on fault diagnosis and nature-inspired optimal control of industrial process applications, **Computers in Industry** (Elsevier Science), vol. 74, pp. 75-94, 2015, impact factor (IF) = 1.685, IF according to 2019 Journal Citation Reports (JCR) released by Clarivate Analytics in 2020 = 3.954, **Hot Paper according to Clarivate Analytics Web of Science** as of November/December 2015 (http://www.aut.upt.ro/~rprecup/CiI_2015_Hot_Paper.jpg).

Computers in Industry 62 (2011) 213–226

Contents lists available at ScienceDirect

Computers in Industry

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

Survey paper

A survey on industrial applications of fuzzy control

Radu-Emil Precup [a,*], Hans Hellendoorn [b]

[a] "Politehnica" University of Timisoara, Faculty of Automation and Computers, Department of Automation and Applied Informatics, Bd. V. Parvan 2, 300223 Timisoara, Romania
[b] Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 25 July 2008
Received in revised form 24 August 2010
Accepted 1 October 2010
Available online 28 October 2010

Keywords:
Adaptive fuzzy control
Mamdani fuzzy controllers
Predictive control
Stable design
Takagi-Sugeno fuzzy controllers

ABSTRACT

Fuzzy control has long been applied to industry with several important theoretical results and successful results. Originally introduced as model-free control design approach, model-based fuzzy control has gained widespread significance in the past decade. This paper presents a survey on recent developments of analysis and design of fuzzy control systems focused on industrial applications reported after 2000.

Contents

1. Introduction

Classical engineering approaches to the characterization of real-world problems are based on essentially qualitative and quantitative technique based on more or less accurate mathematical modelling. In such approaches expressions like "medium temperature", "big humidity", "small pressure", "very big speed",

* Corresponding author. Tel.: +40 256 5032 29/26/30/40; fax: +40 256 403214.
E-mail address: radu.precup@aut.upt.ro (R.-E. Precup).

0166-3615/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2010.10.001

related to the variables specific to the behaviour of a controlled process (CP), are subject to relatively difficult interpretations from the quantitative point of view. This interpretation is difficult because classical automation handles variables/information, processed with well-specified numerical values. Therefore the elaboration of the control strategy and its implementation in the control equipment requires an as accurate as possible quantitative modelling of the CP. Advanced control strategies may require the permanent reassessment of the models and of the values of the parameters characterizing the (parametric) models. Process control based on fuzzy set theory or fuzzy logic) – referred to as fuzzy control or fuzzy logic control – is more pragmatically with this regard because use is made of the linguistic characterization of the quality of CP dynamics and of the adaptation of this characterization as function of the concrete conditions of CP operation.

Zadeh set the basics of fuzzy set theory by a paper that seemed to be just mathematical entertainment about four decades ago [1]. The boom in computer science opened in the seventies the first prospects for applications of the meanwhile built theory in various fields ranging from control engineering, qualitative modelling, pattern recognition, signal processing, information processing, machine intelligence, decision making, management, finance, medicine, and so on. In particular, fuzzy control, as one of the earliest branches and applications of fuzzy sets and systems, has become one of the most successful applications. Fuzzy control has proven to be a successful control approach to many complex nonlinear systems or even nonanalytic ones. It has been suggested as an alternative approach to conventional control techniques in many situations. This paper will be focused on industrial applications, and the analysis is dedicated to the period after 2000.

The first fuzzy control application belongs to Mamdani and Assilian [2,3], where control of a small steam engine is considered. The reference applications of fuzzy control, associated by experiments, deal with a warm water plant [4] and with a small scale heat exchanger [5]. Afterwards, during the eighties in Japan, USA, and later, in Europe, a so-called fuzzy boom took place in the field of fuzzy control applications to several domains beginning with electrical household industry and consumer electronics up to other industries like mechanical and robotic systems, power plants and systems, telecommunications, transportation systems, automotive systems, chemical processes and nuclear reactors. This boom was caused partly by the spectacular development of electronic technology and computer systems that enables:

- the manufacturing of circuits with very high speed of information processing, dedicated (by construction and usage) to a certain purpose including fuzzy information processing and resulting in embedded systems,
- the development of computer-aided design programs, which allow the control system designer to use efficiently a large amount of information concerning the CP and the control equipment.

The industrial applications of fuzzy control reported until now emphasize two important aspects related to this control strategy:

- In some situations (for example, the control of processes with functional nonlinearities which subjected to difficult mathematical modelling and the control of ill-defined processes), fuzzy control can be viewed as a viable alternative to classical, crisp control (conventional control),
- Compared to conventional control, fuzzy control can be strongly based and focused on the experience of a human operator, and a fuzzy controller can model more accurately this experience (in linguistic manner) versus a conventional controller.

The main features of fuzzy control can be organised as follows:

- Fuzzy control employs the so-called fuzzy controllers (FCs) or fuzzy logic controllers ensuring a nonlinear input–output static map that can be influenced/modified based on designer's option.
- Fuzzy control can process several variables from the CP, hence it can be considered as belonging to the class of multi-input–multi-output (MIMO) systems with interactions. Therefore the FC can be considered as a multi-input controller (eventually, a multi-output one, too), similar to linear or nonlinear state-feedback controllers.
- FCs do not possess dynamics, but the applications and performance of FCs and fuzzy control systems (FCSs) can be enlarged significantly by inserting dynamics (i.e., derivative and/or integral components) to fuzzy controller structures resulting in the so-called fuzzy controllers with dynamics.
- FCs are flexible with regard to the modification of the transfer features (by input–output static maps). Thus the possibility to develop a large variety of adaptive control system structures is offered.

The control approach based on human experience is acting in FCs by expressing the control requirements and elaborating the control signal in terms of the natural IF–THEN rules which belong to the set of rules

$$\begin{array}{l}\cdots\\ \text{IF (antecedent) THEN (consequent)},\\ \cdots\end{array} \quad (1)$$

where the antecedent (premise) refers to the found out situation concerning the CP dynamics (compared usually with the desired/imposed dynamics), and the consequent (conclusion) refers to the measures which should be taken – under the form of the control signal u – in order to fulfil the desired dynamics. The set of rules (1) makes up the rule base of the FC.

Research results obtained in studying the behaviour of the human expert emphasize that the expert has a specific strongly nonlinear behaviour accompanied by anticipative, derivative, integral and predictive effects and by adaptation to the concrete operating conditions. Colouring the linguistic characterization of CP dynamics (and, accordingly, of fuzzy mathematical characterization) based on experience and translating it to the control signal elaboration and the analysis of CP dynamics will be characterized by parameters that enable the modification of FC features. From this point of view the FCSs can be regarded as belonging to the general framework of intelligent control systems.

The block diagram of principle (considered as classical in the literature) of an FCS is presented in Fig. 1. The FCS is considered as a single input system with respect to the reference input r and as a

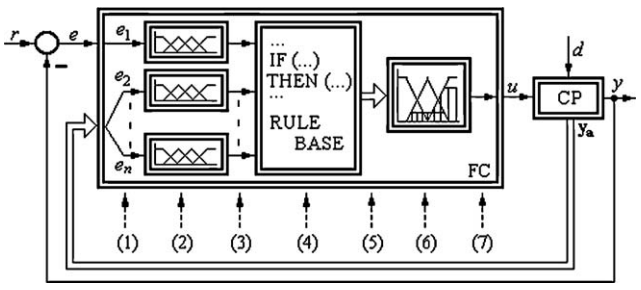

Fig. 1. Basic fuzzy control system structure.

single output system with respect to the controlled output y. The second input fed to the CP/FCS is the disturbance input d.

Fig. 1 highlights also the operation principle of an FC in its classical version, characterizing Mamdani FCs, with the following variables and modules:

(1) the crisp inputs,
(2) the fuzzification module,
(3) the fuzzified inputs,
(4) the inference module,
(5) the fuzzy conclusions,
(6) the defuzzification module,
(7) the crisp output.

The essential, already mentioned, particular feature of FCSs concerns the multiple interactions regarded from the process to the controller by the auxiliary variables $\mathbf{y_a}$, gathered in the input vector $\mathbf{e}'$

$$\mathbf{e}' = [\, e \quad \mathbf{y}_a^T \,] = [\, e_1 \quad e_2 \quad \dots \quad e_n \,]^T. \tag{2}$$

These variables are direct or indirect inputs to the fuzzy controller. No matter how many inputs to the FC are, the FC should possess at least one input variable e_1 that corresponds to the control error e

$$e_1 = e = r - y. \tag{3}$$

According to Fig. 1, the operation principle of a Mamdani FC involves the sequence of operations (a), (b) and (c):

(a) The crisp input information – the measured variables, the reference input (the set point), the control error – is converted into fuzzy representation. This operation is called fuzzification of crisp information.
(b) The fuzzified information is processed using the rule base, composed of the fuzzy IF-THEN rules referred to as fuzzy control rules of type (1) that must be well defined in order to control the given process. The principles to evaluate and process the rule base represent the inference mechanism/engine and the result is the "fuzzy" form of the control signal u, the fuzzy control signal.
(c) The fuzzy control signal must be converted into a crisp formulation, with well-specified physical nature, directly understandable and usable by the actuator in order to be capable of controlling the process. This operation is known under the name of defuzzification.

The three operations described briefly here characterize the three modules in the structure of an FC (Fig. 1), the fuzzification module (2), the inference module (4) and the defuzzification module (6). All three modules are assisted adequate databases.

In the majority of applications an FC is used for direct feedback control or on the low level in hierarchical control system structures. However, it can be used on the supervisory level, for example in adaptive control system structures. Nowadays fuzzy control is no longer only used to directly express the knowledge on the CP or, in other words, to do model-free fuzzy control. An FC can be calculated from a fuzzy model obtained in terms of system identification techniques, and thus it can be regarded in the framework of model-based fuzzy control. Most often used are:

- Mamdani fuzzy controllers, referred to also as linguistic FCs, with either fuzzy consequents that represent type-I fuzzy systems according to the classifications given in [6] or singleton consequents belonging to the type-II fuzzy systems. These FCs are usually used as direct closed-loop controllers.
- Takagi-Sugeno (T-S) fuzzy controllers, known also as type-III fuzzy systems especially when affine consequents are employed, and typically used as supervisory controllers.

Several excellent books and tutorial articles on fuzzy control are well-acknowledged [7–24]. Several survey and position papers highlight specific topics in fuzzy control, make characterizations and present points of view. A good survey on fuzzy modelling for control is done in [25]. The stability analysis methods for type-II fuzzy control systems are analyzed in detail in [6]. A very good survey on neuro-fuzzy rule generation in a rather general setting of soft computation is given in [26]. The fusion of computational intelligent methodologies, including fuzzy logic and sliding mode control, is thoroughly discussed in [27]. Conclusions of great wisdom regarding the perspectives of fuzzy control systems are pointed out in [28]. An excellent survey on analysis and design methods of model based fuzzy control systems is given in [29].

This paper is focused on industrial applications of fuzzy control with application fields that include but are not limited to manufacturing, robotics, automotive and process industry, and control of servo systems and actuators as well. A large part of these applications can be viewed in the framework of mechatronic systems. The authors are aware of the fact that the publications on the topic of fuzzy logic control are so huge that an exhaustive list is impossible. Selected papers are given in the end of this paper. Many excellent works are unfortunately missed. In addition, this survey paper is not able to cover all categories of industrial applications of fuzzy logic control in detail.

The paper addresses the following topics. Industrial applications of control systems with Mamdani fuzzy controllers are discussed in Section 2. Next, Section 3 is focused on control systems with Takagi-Sugeno fuzzy controllers. The stable design of model-based fuzzy control systems and aspects concerning the tensor product (TP) model transformation are considered. Applications of adaptive and predictive fuzzy control dealing with supervision and optimization, i.e., multi-level fuzzy control systems, are presented in Section 4. Section 5 gives concluding remarks, perspectives and challenges of fuzzy control.

2. Control systems with Mamdani fuzzy controllers

The design of FCSs with Mamdani FCs is usually performed by heuristic means incorporating human skills and experience, and it is often carried out by a model-free approach. The immediate shortcoming resulted from the model-free design and FC tuning concerns the lack of general-purpose design methods. Although the performance indices of such control systems are generally satisfactory, a major problem is the analysis of the structural properties possessed by the FCSs including stability, controllability, parametric sensitivity and robustness [19,22–24,28]. In addition, the design of such control systems suffers from the lack of systematic approaches. Therefore much research attention has been devoted to the stability analysis. Actual trends make use of Lyapunov's approach [30], Krasovskii's approach [31], the describing function method [19], Krasovskii-LaSalle's invariant set theorem [32], the small gain theorem [33], algebraic approaches [34] including the vector norms approach [35,36], applicable to Mamdani FCs but also with minor modifications to T-S FCs. The common idea of all these approaches is to regard the FC as a nonlinear controller with Lurie-Postnikov nonlinearity, the CP with crisp model (linear or not) and embed the stability problem of FCSs into the stability theory of conventional nonlinear systems.

Several applications of FCSs with Mamdani FCs are reported in manufacturing. They include control of industrial weigh belt feeders [37], the realization of specific controllers [38,39], control of machining processes [40–43], laser tracking systems [44], plastic injection molding [45] and vibration suppression [46]. The manufacturing area is related to robotics. Mamdani FCs concern control of both manipulators and mobile robots [47–60].

The automotive industry is one special successful area of Mamdani FCs. Problems and practical issues related to suspension control are discussed in [61]. The control of hybrid electric vehicles is treated in [62] and the complexity of all related control strategies is emphasized in [63]. The control of anti-lock braking systems is analyzed in [64,65].

Process industries include Mamdani fuzzy control. The applications reported in this context tackle the control of furnaces [66,67], filtration processes [68], air conditioning [69,70], heat exchangers [71] or forging machines [72].

Control systems should exhibit generally very good steady-state, dynamic performance and robustness as well. Hence they require high quality servo systems that ensure both stabilization and tracking. The same problem is in case of complex control systems where the actuators can be viewed as local control systems with high needs as the performance is concerned. Servo systems are widely used in mechatronics applications characterized by tight coupling of different implementation techniques including hydraulics, mechanics, electro-mechanics, electronics and software [73–75]. Applications of these servo systems can be found in electro-hydraulic systems, actuators in robots or automotive systems, etc., where the CPs can be well characterized in simplified forms by benchmark systems [76–83]. One of the actual trends in control systems for mechatronics is that newer generations shall always be smaller, cheaper and/or provide additional functionality [75]. One difficult and challenging task coming from this is to devise cost-effective solutions that guarantee improved performance of these systems. Fuzzy control has recently been applied to a variety of servo systems and actuators in mechatronics [84–93].

Aircraft, missile autopilot and helicopter control represent also areas where fuzzy control is applied ensure performance improvements. The results outlined in these areas [94–102] can be connected well to those dedicated to servo systems.

2.1. PI-, PD- and PID-fuzzy control

PI, PD and PID controllers are still the most widely used in industrial control loops worldwide because they have simple structures, can be designed easily and offer good control system performance at acceptable cost [103]. The CS performance indices provided by these PID controllers depend not only on the tuning parameters, but also on the necessary implementation of additional functionalities including anti-windup, feedforward action, and set point filtering [104]. However, PI, PD and PID controllers might not ensure satisfactory control system performance if the mathematical model of the CP is highly nonlinear, subjected to parameter variance, and/or uncertainties. On the other hand, conventional fuzzy control is known for its ability to cope with nonlinearities and uncertainties. Introduction of dynamic fuzzy controller structures with the aim of control system performance improvement leads to PI-, PD- or PID-fuzzy controllers [105–108]. Several Mamdani PI-fuzzy controllers (PI-FCs), PD-fuzzy controllers (PD-FCs) and PID-fuzzy controllers (PID-FCs) [109–111] as well as Takagi-Sugeno PI-FCs, PD-FCs and PID-FCs [17,90,100,111] are developed.

PI-FCs, PD-FCs, and PID-FCs can be designed and tuned using two approaches:

- the first is based on the fact that under some well-stated conditions, the approximate equivalence between linear and fuzzy controllers is generally acknowledged [37,112–114],
- the second relies on the consideration of these fuzzy controllers (FCs) as nonlinear PD, PI, or PID controllers with variable gains [115–118].

The first approach is considered as the direct action type of PI-FCs, PD-FCs and PID-FCs [110] since the inference module calculates the control signal (action) directly to control a system. The second approach is viewed as gain scheduling [119,120].

Industrial implementations of PI-, PD- or PID-fuzzy controllers involve both approaches although the gain scheduling was first accepted from industry [121]. Applications of PI-FCs, PD-FCs and PID-FCs were classified and pointed out at the beginning of this section. Other applications are reported in [118,122–128]. Several topics of interest regarding PI-FCs, PD-FCs or PID-FCs, which are well identified in [29], concern the industrial applications FC tuning [117], optimal FCs by inserting genetic algorithms [122,124,129–134] or neural networks [29,135–137], and robust FCs [29,93,138–140].

2.2. Sliding mode fuzzy control

It is well acknowledged that sliding mode control exhibits robustness properties [141]. So a natural direction is to embed this property in fuzzy control. This will lead to the alleviation of the negative effects due to the chattering phenomenon specific to sliding mode control systems and the combination between the two techniques, sliding mode and fuzzy control, leads to complementing the advantages of both ones.

Usual approaches to sliding mode control are:

- The sliding mode controller handles linguistic information modelled by means of fuzzy processing with the elimination of the chattering phenomenon by the creation of fuzzy boundary layers [142–145].
- Supervisory sliding mode controller is inserted to fuzzy controller structures leading to the guarantee of stability and improvement of robustness [146–150].

These approaches ensure the convenient treatment of FCS stability analysis and design in the framework of the well developed methods dedicated to sliding mode control. Symmetrical FCs (as to their definition in an input–output matrix) can be regarded as sliding mode controllers with multiple sliding lines.

2.3. 2-DOF fuzzy control

Since the main tasks in control, the achievement of high performance in set-point tracking and the regulation in the presence of disturbance inputs are difficult to be accomplished by means of PI and PID controllers, one typical approach is to design two-degree-of-freedom (2-DOF) controllers which have advantages over the one-degree-of-freedom ones [151–153]. But, the main drawback of 2-DOF controllers is that although they ensure the regulation, the reduction of overshoot is paid by slower set-point responses because the 2-DOF structures can be reduced to feedforward controllers with set-point weighting.

The control performance enhancement with respect to the modifications of set-point and of load disturbance inputs ensured by the FCs in connection with the overcome of the above mentioned drawback of 2-DOF controllers leads to the idea of 2-DOF fuzzy controllers [154–158]. Very good control system performance with respect to the set-point and disturbance input

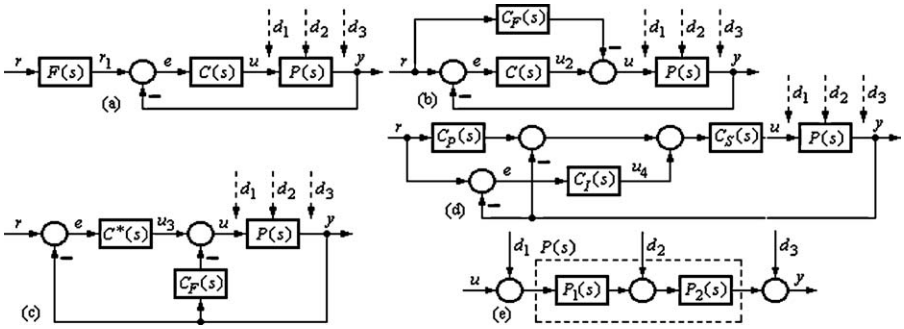

Fig. 2. Set-point filter 2-DOF control system structure (a), feedforward 2-DOF control system structure (b), feedback 2-DOF control system structure (c), component-separated 2-DOF PI control system structure (d), definition of load disturbance input scenarios (e).

can be obtained if the process with the transfer function of the controlled process $P(s)$ is included to the generic 2-DOF control system structures presented in Fig. 2, referred to as the set-point filter structure (Fig. 2(a)), the feedforward structure (Fig. 2(b)), the feedback structure (Fig. 2(c)) and the component-separated structure exemplified for the 2-DOF PI controller (Fig. 2(d)), where: r – the set-point, y – controlled output, $e = r - y$ or $e = r_1 - y$ – the control error, u – the control signal, r_1, u_2, u_2 and u_4 – the outputs of the blocks $F(s)$, $C(s)$ (in Fig. 2(b)), $C^*(s)$ and $C_I(s)$, respectively, and d_1, d_2 and d_3 – several load disturbance inputs defined in Fig. 2(e).

Some of the controller blocks in Fig. 2 can be fuzzified in order to improve the control system performance [158]. Similar structures can be formulated under the form of state-feedback control systems to be treated in the following sections.

3. Control systems with Takagi-Sugeno fuzzy controllers

T-S fuzzy models represent fuzzy dynamic models or fuzzy systems [25,28,159,160]. This brings a twofold advantage. First, any model-based technique (including a nonlinear one) can be applied to the fuzzy dynamic models. Second, the controller itself can be considered as a fuzzy system. Since the fuzzy model of the nonlinear process is usually based on a set of local linear models which are smoothly merged by the fuzzy model structure, a natural and straightforward approach is to design one local controller for each local model of the process. This idea is known as parallel distributed compensation [18]. In parallel distributed compensation (PDC) the structure of the FC model matches the structure of the fuzzy model of the CP.

The identification of T-S fuzzy models is of great importance generally for T-S FC designs and strictly necessary for PDC. Many good results and applications with this respect are reported in [16,18,29,161–166] where use is made of two approaches. The first one is to linearize the nonlinear model of the process in the vicinity of important operating points assuming that the model of the process is known for example in its first principle form. The second one, applied when the model of the process is unknown, is to make use of the data generated (analytically or experimentally) from the original nonlinear process. The second approach consists of two steps, the structure identification and the parameter estimation, and it employs various techniques to solve the optimization problem with the aim in fitting the models to the pairs of input–output data.

The industrial applications of T-S FCs were presented in the previous section in close connection to those of Mamdani FCs. However, due to the model-based design, most references offer results concerning the stabilization of T-S fuzzy models.

3.1. Stable design of model-based fuzzy control systems

The following continuous Takago-Sugeno fuzzy model of the process is considered in the state-space form

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \sum_{i=1}^{r_B} h_i(\mathbf{z}(t))(\mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_i\mathbf{u}(t)),\\ \mathbf{y}(t) &= \sum_{i=1}^{r_B} h_i(\mathbf{z}(t))\mathbf{C}_i\mathbf{x}(t),\end{aligned} \tag{4}$$

where r_B is the number of rules, $\mathbf{z}(t)$ is the vector of measurable variables of the process, $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the control signal (input) vector, $\mathbf{y}(t)$ is the output vector, and

$$h_i(\mathbf{z}(t)) \geq 0, \quad i = \overline{1, r_B}, \tag{5}$$

are the degrees of fulfilment of the rules satisfying the convex sum property

$$\sum_{i=1}^{r_B} h_i(\mathbf{z}(t)) = 1. \tag{6}$$

The local linear models of the process

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_i\mathbf{u}(t),\\ \mathbf{y}(t) &= \mathbf{C}_i\mathbf{x}(t), \quad i = \overline{1, r_B},\end{aligned} \tag{7}$$

are supposed to be observable and controllable. In discrete T-S fuzzy models, $\dot{\mathbf{x}}(t)$ is replaced by $\mathbf{x}(t+1)$ in the models (4) and (7).

The PDC controller for the system (4) is

$$\mathbf{u}(t) = -\sum_{i=1}^{r_B} h_i(\mathbf{z}(t))\mathbf{F}_i\mathbf{x}(t). \tag{8}$$

The goal of the control design problem is to obtain the gain matrices $\mathbf{F}_i$, $i = \overline{1, r_B}$, of the nonlinear state-feedback control law (8) such that the closed-loop system modelled by Eqs. (2) and (8) is stable and eventually robust. Many design problems derive the least conservative conditions [28] that fulfil the condition

$$\sum_{i=1}^{r_B}\sum_{j=1}^{r_B} h_i(\mathbf{z}(t))h_j(\mathbf{z}(t))\Gamma_{ij} < 0, \quad \Gamma_{ij} = \Gamma_{ij}^T. \tag{9}$$

The first results consider the quadratic Lyapunov function candidate [167]

$$V(\mathbf{x}) = \mathbf{x}^T(t)\mathbf{P}\mathbf{x}(t), \quad \mathbf{P} = \mathbf{P}^T > 0. \tag{10}$$

The calculation of the derivative of the function defined in (10) along the trajectories of the FCS characterized by Eqs. (2) and (8)

leads to the result

$$\Gamma_{ij} = (\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j)^T\mathbf{P} + \mathbf{P}(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j) < 0, \quad i, j = \overline{1, r_B}, \tag{11}$$

for continuous T-S models or

$$\Gamma_{ij} = (\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j)^T\mathbf{P}(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j) - \mathbf{P} < 0, \quad i, j = \overline{1, r_B}. \tag{12}$$

for discrete T-S fuzzy models:

The main approach to solve the system of Eqs. (9) and (11) makes use of linear matrix inequality (LMI)-based techniques [168]. Popular approaches employ quadratic, piecewise quadratic, non-quadratic, parameter-dependent, polynomial and fuzzy Lyapunov functions [169–182], and they show the constant effort to reduce the conservativeness of the stability conditions. Although the LMIs are computationally solvable they require numerical algorithms embedded in well acknowledged software tools.

The LMIs can be applied to other control system structures with T-S FCs and models. They include the cascade control systems [183] eventually with fuzzy observers [184] which are validated by experiments and/or simulations.

3.2. Tensor product model transformation and fuzzy control systems

One of the current trends in fuzzy control is to derive less conservative conditions to prove the stability and the performance of FCSs [19,28,185]. The fuzzy partitions are the combinations of the products of rather simple arguments expressed as membership functions. In real-world applications one particular case concerns fuzzy modelling of nonlinear systems under the form of TP fuzzy systems. The expression of TP fuzzy systems can be understood in terms of operations on multi-dimensional arrays [185].

The TP model transformation is capable of transforming a dynamic system model, given over a bounded domain, into the TP model form, including polytopic or T-S fuzzy model forms. The TP model transformation may be defined as one numerical method capable of transforming the linear parameter-varying (LPV) dynamic models into parameter-varying weighted combination of parameter independent (constant) system models under the form of linear time-invariant (LTI) systems. This transformation of LPV models is uniform in both theoretical and algorithmic execution and it considers different optimization constraints. The main advantage of TP model transformation in modifying the given LPV models to varying convex combinations of LTI models is that the LMI-based control design frameworks can be applied immediately to the resulting affine models in order to get a tractable and improved performance of the FCSs.

The widely applied transfer function of the product decision operator-based T-S fuzzy models and the function of the TP model is the same from the analytical point of view in widely general cases. The main philosophical difference between them is that the T-S fuzzy model originally means a fuzzy combination of locally linearized LTI models, where the locality is expressed by the shape of the antecedent fuzzy sets, for instance, by triangular fuzzy sets where the location of the fuzzy set is readily determinable. However in case of TP model the weighting functions (which correspond to the membership functions in the fuzzy models) may not have locality, they spread in the whole interval of interest, so as the LTI components of the model cannot readily be assigned to a definite operation point. They are mostly vertexes of a polytopic structure. In conclusion, the T-S fuzzy model originally is a fuzzy combination of linearized operation points (LTI systems are close to local models), while the TP model is originally a polytopic structure (LTI systems are the vertex models of a convex hull of the model, they may be relatively far from any linearized operation points). In other words, an LTI system affects a fuzzy local area in case of T-S models. In the case of TP models an LTI system affects the whole operation domain, but according to the weighting functions. As a matter of fact, in today systems these two original ideas are combined in both T-S and TP models, therefore the difference is not important.

The TP model transformation generates two kinds of polytopic models. Initially, it reconstructs the high order singular value decomposition (HOSVD)-based canonical form of the LPV models. The major outcome of the recently developed HOSVD comes from its ability in decomposing a given N-dimensional tensor into a full orthonormal system in a special ordering of higher order singular values.

Regarding the variety of well acknowledged and implemented identification techniques, it is difficult to derive the uniform representation of the designed LPV model forms and the forms resulted from the identification. Hence the TP model transformation might be a possible solution for that situation, and an immediate link between the model transformation and the LMIs should be determined.

A key advantage of the TP model transformation is that it allows the modification of the parameter varying convex combination according to the designer's option. The type of the convex combination considerably influences the further LMI design and resulting control performance. Therefore the design can be based on the manipulation of the convex hull beside the manipulation of the LMIs.

Based on the core theory of the TP model transformation that is coming from the singular value decomposition (SVD) methods [186] the TP model transformation is capable of reducing the complexity of TP structured functions like T-S fuzzy models or B-spline models and so on. The multilinear generalizations of the SVD and the investigation on how the tensor symmetries affect the decomposition are discussed in [187]. The HOSVD has been developed since the existing framework of vector and matrix algebra; it appeared to be insufficient as increasing number of signal processing problems involved the manipulation of quantities of which the elements are addressed by more than two indices, i.e., higher-order tensors. Use is made of higher-order tensors to describe the transformations in the same way as the matrices describe linear transformations between vector spaces.

Making use of the TP model transformation, different optimization and convexity constraints can be considered and the transformations can be executed as well without any analytical interactions within less time. Thus, the transformation replaces the usual analytical conversions.

Accepting an N-dimensional bounded parameter vector $\mathbf{p}(t)$ and considering the LPV model

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t), \end{aligned} \tag{13}$$

the system matrix

$$\mathbf{S}(\mathbf{p}(t)) = \begin{pmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{pmatrix} \in \Re^{O\times I} \tag{14}$$

is a parameter-varying object. The convex state-space TP model describes the LPV state-space model for any parameter vector $\mathbf{p}(t)$ as the convex combination of LTI system matrices

$$\begin{pmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{pmatrix} = S\otimes_{n=1}^{N} \mathbf{w}_n(\mathbf{p}_n(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{15}$$

where the row matrix $\mathbf{w}_n(\mathbf{p}_n(t))$ contains one bounded variable and its continuous weighting functions, and N indicates the tensor's dimension. The (finite element) TP model defined in (15) is convex

only if the weighting functions fulfil the condition

$$w_{n,i}(\mathbf{p}_n(t)) \in [0,1], \quad \forall n, i, \mathbf{p}_n(t),$$
$$\sum_{i=1}^{I_n} w_{n,i}(\mathbf{p}_n(t)) = 1, \quad \forall n, \mathbf{p}_n(t). \tag{16}$$

The LMI-based controller design methods can immediately be applied after the transformation of the LPV model (14) given in HOSVD-based canonical form to the TP model form expressed in (15) and (16). In other words, the TP model transformation is to be used and executed before utilizing the LMI design, i.e., when the LMI design is started the global weighting functions are already defined.

A short presentation of the applications of TP model transformation, well connected to T-S FCSs, is presented in [188] and accompanied by a temperature control application. An attractive control design method accompanied by application is given [189,190] to stabilize parameter varying nonlinear state-space models. It is based on two numerical steps. In the first step the TP model transformation is executed. In the second step LMIs are solved under the PDC framework. The first step consists in transforming a given model into a TP, so that the design techniques of the PDC framework can be employed. The operations associated to the second step produce a controller according to various control specifications. The advantages of this method are twofold. First, the controller can be derived automatically, regardless of analytic derivations. Second, the identified model can be defined either by analytical equations or by other soft computing techniques.

A popular TP model application deals with controlling the TORA system [191] where a nonlinear controller has been designed making use of the TP model transformation and a LMI-based controller design technique. The results show that both numerical methods, the TP model transformation and the LMIs, can be accomplished numerically without analytical derivations, leading to fast controller designs.

A case study regarding the TP model transformation behaviour in real-world applications is discussed in [192] with focus on the single pendulum gantry system. A generalization of the double fuzzy summation results to multiple summations with a TP structure is emphasized in [185]. This is meant to replace the well accepted common structure in many fuzzy models. A simulated application concerning the inverted pendulum system is included.

A Matlab toolbox for TP model transformation, the TP Tool, is implemented and described in [193]. The toolbox is applied to several benchmark systems and to the real-time control of the liquid levels in a three tank system [194].

An excellent application of the TP model transformation deals with offering a control solution for the aeroelastic wing section problem that was considered as unsolved previously [195,196]. This is the first convincing well detailed example of applying TP model transformation with PDC design framework. It shows the observer design as well.

4. Adaptive fuzzy control, supervision and optimization

There are many formations for the FC in FCSs similarly to the different control schemes focused on PI controllers presented in Fig. 2. An adaptive FC has one extra component, a supervisory module, as shown in Fig. 3. The supervisory module has

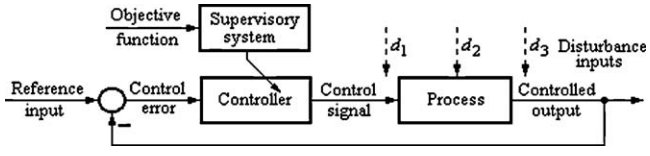

Fig. 3. An adaptive fuzzy controller with a supervisory system.

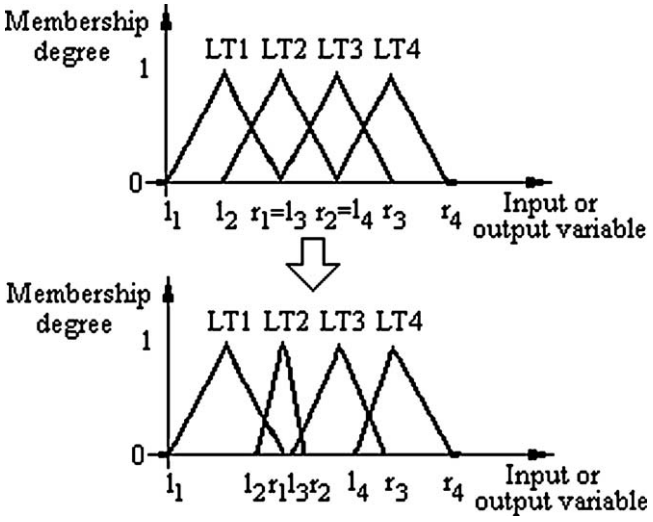

Fig. 4. Changing the side slopes of a group of membership functions. The modal values (centres) stay at the same position.

understanding of the process and of the controller, and has access to all input and usually also to all output signals. The supervisory module can modify several components of the fuzzy controller like the size of the membership functions of the fuzzy sets, the position of the membership functions, the rule weights and/or the link values. These four items will be discussed below. A predictive FC will be also described. That controller does not change the parameters of the underlying FC, but it chooses every time the best control signal based on a performance measure expressed in terms of an objective function. The goal is to minimize the objective function.

4.1. Adaptation of the size of the membership functions

The supervisor can change the size of the membership functions of the fuzzy sets corresponding to the linguistic terms of the FC, e.g., increase or decrease the support (width) of an individual membership functions or change the width to the left-hand or right-hand side, like in Fig. 4, applicable to both input and output linguistic variables of the fuzzy controller. The triangular membership functions of the linguistic terms LT1, Lt2, LT3 and LT4 were chosen in Fig. 4, with the parameters $l_i, r_i, i = \overline{1,4}$. Likewise, the spread σ of Gaussian fuzzy sets can be adapted.

Actually, the adaptation means that the supervisor can change the partitioning of the membership functions on the universe of discourse. An example of this is a controller for the cruise control of a car [197]. With cruise control the driver fixes the speed of the car. If the car goes uphill or downhill the cruise system controls the throttle in such a way that the car keeps its velocity, if the driver touches the break pedal or the "coach button" the cruise system will stop working until the "resume button" is touched. By then, the car will accelerate until it returns to the desired speed. Depending on the load of the car and the weather conditions this can lead to too fast or too slow acceleration and overshoot. The supervisor should "learn" the load of the car and adapt the cruise control to the current situation.

In [197] the system is described as in Fig. 5. The error in the velocity is calculated from the vehicle and the reference speed, the same holds for the first and second derivative of the speed. The FC calculates the throttle opening, passes it on to the actuator which applies it to the vehicle. If the throttle opening is too slow or too fast, i.e., the car accelerates too slow or to fast, the supervisor changes the membership functions of the fuzzy sets corresponding

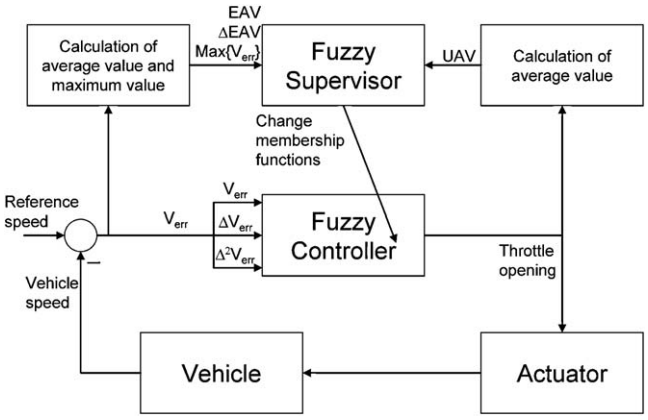

Fig. 5. Logical scheme of an adaptive fuzzy system for cruise control. EAV is the error average value, UAV is the average value of the throttle opening, V_{err} is the maximum value of the error [197].

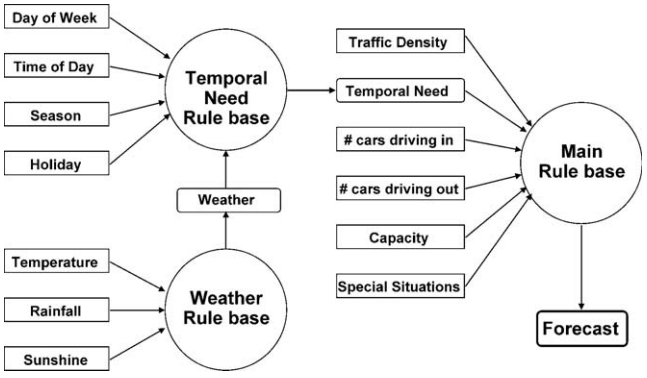

Fig. 7. Structure of fuzzy parking system.

to the linguistic terms of the fuzzy controller. The supervisor has four inputs: the error average value (EAV) and its first derivative (ΔEAV), the maximum value of the error (V_{err}) and the average value of the throttle opening (UAV). The error average value and its derivative give an indication how long it takes to bring the car back to its desired speed. If the car accelerates too slow this means, for example, that the car is fully loaded and the membership functions in the controller should be adapted to this situation.

4.2. Adaptation of the position of the membership functions

Another method to tune an FC or a fuzzy decision system is by repositioning the membership functions of the fuzzy sets corresponding to the linguistic terms of the fuzzy controller. Usually this is done in combination with a data clustering method. In a high dimensional system the data points are normally not evenly distributed over the whole data space, but occur in groups. If one uses a clustering algorithm to identify the data clusters then one can position the membership functions of the fuzzy sets in such a way that they correspond exactly to the data clusters. This enables one to work with a minimal number of fuzzy sets whose membership functions are well positioned to deal with the data. Fig. 6 illustrates this information processing, where v_1 and v_2 are the inputs, μ_{k1} and μ_{k2} are the membership functions of the input linguistic terms, $k = \overline{1,5}$. Overviews on data-driven clustering methods for adaptive fuzzy control are given in [198,199].

An example of this method is a fuzzy system for forecasting the number of empty places in a number of parking garages downtown [200]. The system has a large number of inputs like weather data, time information and traffic information as shown in Fig. 7. The weather is supposed to influence the number of people that go downtown for shopping or to visit the theatre, hence the three weather inputs (temperature, rainfall and sunshine) influence the number of cars downtown. Time information concerns the time of the day, the day of the week, the season and special days like holidays or large events. Together with the weather information they will give a forecast of the number of cars that will go downtown in the next hours. Traffic information concerns the traffic density, and therefore the time people need to arrive at the parking garage, and the time people need to enter the parking garage, as many of them have narrow passages and long waiting times when other people are manoeuvring through the garage. When one considers data points over a longer period of time for each parking garage separately, different patterns of data will lead to a different positioning of the membership functions and therefore to different fuzzy rule bases. In six different parking garages in Düsseldorf, Germany, the initial prediction quality of the described system was around 80%. After repositioning and resizing the membership functions of the fuzzy sets with a neural network, the prediction quality changed as described in Table 1.

Table 1
Parking garages and the prediction quality of the adapted fuzzy system.

Parking garage	Prediction quality July	Prediction quality August
PG1	0.9548	0.9239
PG2	0.9147	0.9086
PG3	0.9298	0.9272
PG4	0.9415	0.9051
PG5	0.9147	0.8849
PG6	0.9415	0.9374

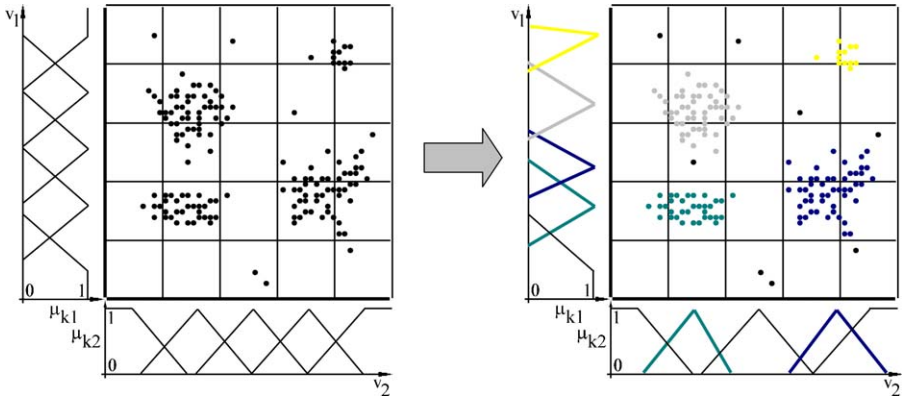

Fig. 6. Repositioning fuzzy sets such that they fit to clusters of input data.

4.3. Adaptation of the rule base

A third adaptation method option is to change the rule weights depending on the contribution of each rule to the performance of the fuzzy control system. Each rule has a rule weight r_i, usually these r_i's are initially 1. If a control action was successful the rule weights of the involved rules in this action are increased. If an action was unsuccessful the weights of the involved rules are decreased. Rules that after some time have rule weight 0 are not involved anymore and will be removed.

This is an easy and straightforward method. Of course the quality of this method highly depends on the performance measure and the selection of the training data. One should always check which rules were removed, in some cases these rules concern exception cases and should be reintroduced by hand. This method is used in identifying and controlling a large paper mill [201], and in traffic modelling and control [202].

4.4. Adaptation of the link values

A fourth adaptation method is to change the link values. This is derived from neural networks and is actually not natural for fuzzy systems. It means the following. Consider a rule base with two inputs X_1 and X_2 and one output Y, and three fuzzy sets P, Z, and N, on each input domain. One can translate the fuzzy system to a neural network as described in Fig. 8. If one takes, for example, the fifth node in the Inference column, this corresponds to the rule "if X_1 is N and X_2 is P then Y is Z". According to [203,204], the logarithms and exponentials are needed in this scheme to cope with several neural network properties.

Now one can feed the neural network with input and output data pairs and train the network. Consider, for example, the rule "if X_1 is Z and X_2 is P then Y is P", corresponding to the lowest node in the Inference column of Fig. 8. After training the network this may have become "if (0.5 X_1 is Z) and (0.3 X_2 is P) then (0.8 Y is P)", because the neural network has adapted the link values. This rule is hard to interpret but will probably describe the situation exactly.

This method is, for example, used in a system with many sensors in a car that had to diagnose the kind of road and traffic conditions the car was driving in [204]. There was an initial rule base describing the logical relation between the sensors and the driving situation. Test data were generated by driving the car with a video camera several days in all kinds of situations. The test data were categorized to several driving situations. After translating the rule base to a neural network all rule weights, the positions of the membership functions and the link weights were adapted which resulted in a much improved fuzzy rule base. The customer demanded a rule base that could be checked by hand afterwards instead of a black box neural network. The combination of a neural network learning system and a readable fuzzy rule base is perfect for automotive industry.

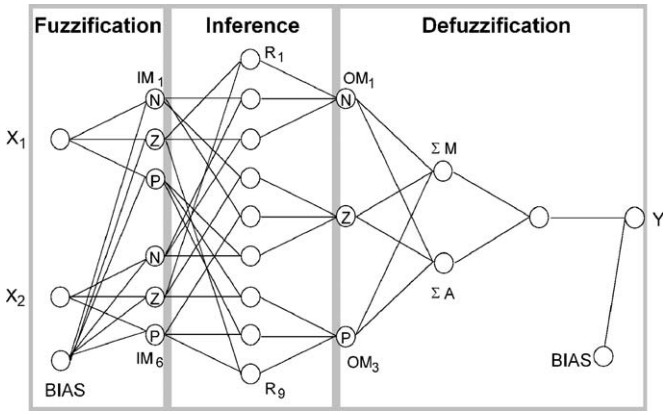

Fig. 8. A neural network translation from a fuzzy rule base.

The neuro-fuzzy control systems can identify fuzzy control rules and tune membership functions of the fuzzy controller making use of the learning algorithms specific to neural networks due to the well accepted functional equivalence between certain classes of fuzzy systems and certain architectures of neural networks [161]. Neuro-fuzzy control is in fact fuzzy control that ensures enhanced control system performance due to the learning capabilities and parallel processing brought by the neural networks. ANFIS [205] is the most popular approach with this regard. Industrial applications of adaptive fuzzy control can be found in batch processes [206–210], robotics [58,137,211–213], aircrafts [214–217] or servo systems and electrical drives [218–221]. An attractive application concerning the development of an intelligent distributed and supervised control system for high-volume production systems is suggested in [222]. A neural-fuzzy-based force model for controlling band sawing process in the framework of an intelligent adaptive control and monitoring system is given in [223]. An adaptive control solution for a ventilating and air-conditioning (HVAC) system is proposed in [224].

Topics of interest in adaptive fuzzy control include robust adaptive control, the combination with sliding mode control and the inclusion of derivative-free optimization techniques to minimize the objective function that specifies the performance measure of the FCS. The derivative-free optimization techniques are needed in industrial applications due to the complicated expression of the objective function with several possible local minima and to the specific constraints associated to the optimization problem. Several fuzzy rule interpolation techniques can be used in real-time applications which have sparse or incomplete rule bases [225–227].

4.5. Fuzzy model-based predictive control

One illustrative industrial application of fuzzy model-based predictive control is presented in [228]. A predictive controller is suggested to modify the parameters of a T-S FC using the prediction of the future process output. Use is made of the fact that if you have a fuzzy model, you can test assumed future situations by putting data into the model. It is possible to compare the outcomes of different control inputs and take the best to proceed with. The results presented in [228] involve a fuzzy model of a chemical plant under the form of the following six fuzzy rules:

$$R_j: \text{IF } C_A(k) \text{ IS } A_j \text{ THEN } C_A(k+1) = -a_{1j}C_A(k) - a_{2j}C_A(k-1) + b_{1j}q_c(k-T_{D_m}) + r_j,\ j=\overline{1,6}, \quad (17)$$

where C_A is the measured product concentration and T_{D_m} is the temperature. It is possible to rewrite these under the classical form

$$\begin{aligned} \mathbf{x}_m(k+1) &= \bar{\mathbf{A}}_m\mathbf{x}_m(k) + \bar{\mathbf{B}}_m u(k-T_{D_m}) + \bar{\mathbf{R}}_m, \\ y_m(k) &= \bar{\mathbf{C}}_m\mathbf{x}_m(k). \end{aligned} \quad (18)$$

The system output is calculated as follows for H steps ahead:

$$y_m(k+H) = \bar{\mathbf{C}}_m[\bar{\mathbf{A}}_m^H\mathbf{x}_m(k) + (\bar{\mathbf{A}}_m^H - I)(\bar{\mathbf{A}}_m - I)^{-1}(\bar{\mathbf{B}}_m u(k) + \bar{\mathbf{R}}_m)], \quad (19)$$

and it compares the output with the output of a reference model. It then calculates which parameters force the output to reach the reference trajectory in the best way and uses these parameters in the next control step. Other industrial applications of fuzzy model-based predictive control are reported in [229–236].

5. Conclusions

The paper addresses a brief survey on industrial applications of fuzzy control. The following classification of the control systems has been proposed with this regard:

- control systems with Mamdani fuzzy controllers,
- control systems with Takagi-Sugeno fuzzy controllers,
- adaptive and predictive control systems.

Although the literature makes a distinction between model-based and model-free fuzzy control, the model-based design of fuzzy control outlined in [237] is needed. The authors' opinion is that all fuzzy control applications should be tackled in the model-based design manner. This is the way that enables systematic analyses of the structural properties of the FCSs such as stability, controllability, parametric sensitivity and robustness. Furthermore, this is the only solution to guarantee the desired/imposed control system performance indices in several operating regimes, and it represents one of the perspectives of fuzzy control.

A lot of industrial applications of fuzzy control are known and reported today. This paper has highlighted just part of them. It contains both mathematics and concrete applications thus emphasizing the concrete connection between the industrial applications of fuzzy control and the necessity of understanding the basics of operating principle and mathematical characterizations of fuzzy controllers. The presentation of rather real-time experiments instead of digital simulation results is another perspective of fuzzy control. In this context the popularity of fuzzy control will increase only if future applications will exhibit significantly better performance compared to the non-fuzzy ones.

There are several challenges which deserve more study when fuzzy control is regarded from the point of view of its applicability:

- the design of rather general fuzzy controllers for well defined classes of systems instead of the particular controllers dedicated to certain narrow applications,
- the use of iterative tuning and learning techniques that start with initial fuzzy control systems and ensure next the performance enhancement making use of the variables measured from those closed-loop systems during their real-time operation [32,91],
- the identification of Takagi-Sugeno fuzzy models with trade-off to transparency, approximation accuracy and controller design possibility,
- the connection between the parameter settings and tuning of the fuzzy controller, and the imposed control system performance in terms of performance indices like overshoot and settling time,
- the alleviation of the conservatism and sufficient conditions-like character in the stable design of fuzzy control systems in case of Lyapunov's approach in connection with LMIs,
- the need for low-cost fuzzy controllers from the points of view of design in tuning transparency as well as implementation costs,
- the need for smooth control signals to be elaborated by all fuzzy controllers,
- the use of the additional parameterization offered by type-2 fuzzy logic in handling the uncertainties specific to industrial processes [238–244].

These challenges will attract the researchers and practitioners. The immediate results will be reflected in more industrial applications of fuzzy control illustrated in conference and journal publications. The 244 reference positions cited here present a sample of the results reported in the literature, and they can be viewed as a guarantee that future successful applications will be constructed.

Acknowledgements

The support from the CNMP and CNCSIS of Romania is kindly acknowledged. The research was supported by the ETOCOM project (TÁMOP-4.2.2-08/1/KMR-2008-0007) through the Hungarian National Development Agency in the framework of Social Renewal Operative Programme supported by EU and co-financed by the European Social Fund. The authors thank Prof. Robert Babuška for setting the structure of the paper, participating in its early stage and giving the idea to extend it up to a journal paper. The authors would also like to thank the colleagues and friends Prof. Stefan Preitl, Prof. Peter Baranyi, Prof. Igor Škrjanc, Assoc. Prof. Sašo Blažič and Dr. Marius L. Tomescu for their fruitful co-operation, contribution and discussions.

References

[1] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.
[2] E.H. Mamdani, Applications of fuzzy algorithms for control of a simple dynamic plant, in: Proceedings of the IEE 121, vol. 12, 1974, pp. 1585–1588.
[3] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man–Machine Studies 7 (1) (1975) 1–13.
[4] W.J.M. Kickert, H.R. Van Nauta Lemke, Application of a fuzzy logic controller in a warm water plant, Automatica 12 (4) (1976) 301–308.
[5] J.J. Ostergaard, Fuzzy logic control of a heat exchanger process, in: M.M. Gupta, G.N. Saridis, B.R. Gaines (Eds.), Fuzzy Automata and Decision Processes, North-Holland, Amsterdam, 1977, pp. 285–320.
[6] M. Sugeno, On stability of fuzzy systems expressed by fuzzy rules with singleton consequents, IEEE Transactions on Fuzzy Systems 7 (2) (1999) 201–224.
[7] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller—Part I, IEEE Transactions on Systems, Man, and Cybernetics 20 (2) (1990) 404–418.
[8] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller—Part II, IEEE Transactions on Systems, Man, and Cybernetics 20 (2) (1990) 419–435.
[9] D. Driankov, H. Hellendoorn, M. Reinfrank, An Introduction to Fuzzy Control, Springer-Verlag, Berlin, Heidelberg, New York, 1993.
[10] W. Pedrycz, Fuzzy Control and Fuzzy Systems, Research Studies Press, Taunton, 1993.
[11] R.R. Yager, D.P. Filev, Essentials of Fuzzy Modeling and Control, John Wiley & Sons, New York, 1994.
[12] C.W. De Silva, Intelligent Control: Fuzzy Logic Applications, CRC Press, New York, 1995.
[13] J.M. Mendel, Fuzzy logic systems for engineering: a tutorial, in: Proceedings of the IEEE 83, vol. 2, 1995, pp. 345–377.
[14] R. Palm, D. Driankov, H. Hellendoorn, Model Based Fuzzy Control, Springer-Verlag, New York, 1996.
[15] L.X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, London, 1997.
[16] R. Babuška, Fuzzy Modeling for Control, Kluwer Academic Publishers, Boston, MA, 1998.
[17] K. Passino, S. Yurkovich, Fuzzy Control, Addison-Wesley, Reading, MA, 1998.
[18] K. Tanaka, H.O. Wang, Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach, John Wiley & Sons, New York, 2001.
[19] K. Michels, F. Klawonn, R. Kruse, A. Nürnberger, Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers, Springer-Verlag, Berlin, Heidelberg, New York, 2006.
[20] Z. Kovačić, S. Bogdan, Fuzzy Controller Design: Theory and Applications, CRC Press, Boca Raton, FL, 2006.
[21] H. Zhang, D. Liu, Fuzzy Modeling and Fuzzy Control, Birkhäuser, Boston, MA, 2006.
[22] J. Jantzen, Foundations of Fuzzy Control, John Wiley & Sons, Chichester, 2007.
[23] J. Kluska, Analytical Methods in Fuzzy Modeling and Control, Springer-Verlag, Berlin, Heidelberg, 2009.
[24] G. Feng, Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach, CRC Press, Boca Raton, FL, 2010.
[25] R. Babuška, H.B. Verbruggen, An overview on fuzzy modelling for control, Control Engineering Practice 4 (11) (1996) 1593–1606.
[26] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Transactions on Neural Networks 11 (3) (2000) 748–768.
[27] O. Kaynak, K. Erbatur, M. Ertugrul, The fusion of computationally intelligent methodologies and sliding-mode control—a survey, IEEE Transactions on Industrial Electronics 48 (1) (2001) 4–17.
[28] A. Sala, T.M. Guerra, R. Babuška, Perspectives of fuzzy systems and control, Fuzzy Sets and Systems 156 (3) (2005) 432–444.
[29] G. Feng, A survey on analysis and design of model-based fuzzy control systems, IEEE Transactions on Fuzzy Systems 14 (5) (2006) 676–697.
[30] M. Sugeno, T. Taniguchi, On improvement of stability conditions for continuous Mamdani-like fuzzy systems, IEEE Transactions on Systems, Man, and Cybernetics—Part B 34 (1) (2004) 120–131.
[31] E. Tian, C. Peng, Delay-dependent stability analysis and synthesis of uncertain T-S fuzzy systems with time-varying delay, Fuzzy Sets and Systems 157 (4) (2006) 544–559.

[32] R.-E. Precup, S. Preitl, I.J. Rudas, M.L. Tomescu, J.K. Tar, Design and experiments for a class of fuzzy controlled servo systems, IEEE/ASME Transactions on Mechatronics 13 (1) (2008) 22–35.
[33] B.M. Mohan, A. Sinha, Analytical structure and stability analysis of a fuzzy PID controller, Applied Soft Computing 8 (1) (2008) 749–758.
[34] J.M. Andújar, J.M. Bravo, A. Peregrín, Analysis and synthesis of multivariable fuzzy systems using interval arithmetic, Fuzzy Sets and Systems 148 (3) (2004) 337–353.
[35] P. Borne, M. Benrejeb, On the representation and the stability study of large scale systems, International Journal of Computers, Communications & Control 3 (Suppl. S) (2008) 55–66.
[36] A. Sakly, B. Zahra, M. Benrejeb, Stability analysis of continuous PI-like fuzzy control systems based on vector norms approach, in: Proceedings of Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA-EUSFLAT 2009), Lisbon, Portugal, (2009), pp. 1655–1660.
[37] Y. Zhao, E.G. Collins Jr., Fuzzy PI control design for an industrial weigh belt feeder, IEEE Transactions on Fuzzy Systems 11 (3) (2003) 311–319.
[38] A. Dvorák, H. Habiballa, V. Novák, V. Pavliska, The concept of LFLC 2000—its specificity, realization and power of applications, Computers in Industry 51 (3) (2003) 269–280.
[39] S. Galichet, L. Foulloy, Integrating expert knowledge into industrial control structures, Computers in Industry 52 (3) (2003) 235–251.
[40] R.E. Haber, J.R. Alique, A. Alique, J. Hernández, R. Uribe-Etxebarria, Embedded fuzzy-control system for machining processes: results of a case study, Computers in Industry 50 (3) (2003) 353–366.
[41] A.K. Nandi, J.P. Davim, A study of drilling performances with minimum quantity of lubricant using fuzzy logic rules, Mechatronics 19 (2) (2009) 218–232.
[42] R.E. Haber, R. Haber-Haber, A. Jiménez, R. Galán, An optimal fuzzy control system in a network environment based on simulated annealing. An application to a drilling process, Applied Soft Computing 9 (3) (2009) 889–895.
[43] R.E. Haber, R.M. del Toro, A. Gajate, Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process, Information Sciences 180 (14) (2010) 2777–2792.
[44] B. Ying, H. Zhuang, Z.S. Roth, Fuzzy logic control to suppress noises and coupling effects in a laser tracking system, IEEE Transactions on Control Systems Technology 13 (1) (2005) 113–121.
[45] M.-Y. Chen, H.-W. Tzeng, Y.-C. Chen, S.-C. Chen, The application of fuzzy theory for the control of weld line positions in injection-molded part, ISA Transactions 47 (1) (2008) 119–126.
[46] M. Marinaki, Y. Marinakis, G.E. Stavroulakis, Fuzzy control optimized by PSO for vibration suppression of beams, Control Engineering Practice 18 (6) (2010) 618–629.
[47] P. Dassanayake, K. Watanabe, K. Kiguchi, K. Izumi, Robot manipulator task control with obstacle avoidance using fuzzy behavior-based strategy, Journal of Intelligent and Fuzzy Systems 10 (3–4) (2001) 139–158.
[48] I. Baturone, F.J. Moreno-Velo, S. Sanchez-Solano, A. Ollero, Automatic design of fuzzy controllers for car-like autonomous robots, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 447–465.
[49] J. Xiao, J.Z. Xiao, N. Xi, R.L. Tummala, R. Mukherjee, Fuzzy controller for wall-climbing microrobots, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 466–480.
[50] S.X. Yang, H. Li, M.Q.H. Meng, P.X. Liu, An embedded fuzzy controller for a behavior-based mobile robot with guaranteed performance, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 436–446.
[51] P. Saeedi, P.D. Lawrence, D.G. Lowe, P. Jacobsen, D. Kusalovic, K. Ardron, P.H. Sorensen, An autonomous excavator with vision-based track-slippage control, IEEE Transactions on Control Systems Technology 13 (1) (2005) 67–84.
[52] Y.X. Su, C.H. Zheng, B.Y. Duan, Fuzzy learning tracking of a parallel cable manipulator for the square kilometre array, Mechatronics 15 (6) (2005) 731–746.
[53] A. Green, J.Z. Sasiadek, Heuristic design of a fuzzy controller for a flexible robot, IEEE Transactions on Control Systems Technology 14 (2) (2006) 293–300.
[54] Z. Sakr, E.M. Petriu, Hexapod robot locomotion using a fuzzy controller, in: Proceedings of IEEE International Workshop on Robotic and Sensors Environments (ROSE 2007), Ottawa, ON, Canada, (2007), pp. 63–67.
[55] F.M. Raimondi, M. Melluso, Fuzzy motion control strategy for cooperation of multiple automated vehicles with passengers comfort, Automatica 44 (11) (2008) 2804–2816.
[56] M.S. Alam, M.O. Tokhi, Hybrid fuzzy logic control with genetic optimisation for a single-link flexible manipulator, Engineering Applications of Artificial Intelligence 21 (6) (2008) 858–873.
[57] I. Harmati, K. Skrzypczyk, Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework, Robotics and Autonomous Systems 57 (1) (2009) 75–86.
[58] P.R. Vundavilli, D.K. Pratihar, Dynamically balanced optimal gaits of a ditch-crossing biped robot, Robotics and Autonomous Systems 58 (4) (2010) 349–361.
[59] S.G. Tzafestas, K.M. Deliparaschos, G.P. Moustris, Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: design of system on a chip, Robotics and Autonomous Systems 58 (8) (2010) 1017–1027.
[60] J. Lin, R.-J. Lian, Stability indices for a self-organizing fuzzy controlled robot: a case study, Engineering Applications of Artificial Intelligence 23 (6) (2010) 1019–1034.
[61] R. Caponetto, O. Diamante, G. Fargione, A. Risitano, D. Tringali, A soft computing approach to fuzzy sky-hook control of semiactive suspension, IEEE Transactions on Control Systems Technology 11 (6) (2003) 786–798.
[62] N.J. Schouten, M.A. Salman, N.A. Kheir, Fuzzy logic control for parallel hybrid vehicles, IEEE Transactions on Control Systems Technology 10 (3) (2002) 460–468.
[63] F.R. Salmasi, Control strategies for hybrid electric vehicles: evolution, classification, comparison, and future trends, IEEE Transactions on Vehicular Technology 56 (5) (2007) 2393–2404.
[64] A. Mirzaei, M. Moallem, B. Mirzaeian, B. Fahimi, Design of an optimal fuzzy controller for antilock braking systems, in: Proceedings of 2005 IEEE Conference on Vehicle Power and Propulsion, Chicago, IL, USA, (2005), pp. 823–828.
[65] Z. Zhao, Z. Yu, Z. Sun, Research on fuzzy road surface identification and logic control for anti-lock braking system, in: Proceedings of International Conference on Vehicular Electronics and Safety (ICVES 2006), Shanghai, China, (2006), pp. 380–387.
[66] U.-C. Moon, K.Y. Lee, Hybrid algorithm with fuzzy system and conventional PI control for the temperature control of TV glass furnace, IEEE Transactions on Control Systems Technology 11 (4) (2003) 548–554.
[67] M. Ramírez, R. Haber, V. Peña, I. Rodríguez, Fuzzy control of a multiple hearth furnace, Computers in Industry 54 (1) (2004) 105–113.
[68] M. Onat, M. Dogruel, Fuzzy plus integral control of the effluent turbidity in direct filtration, IEEE Transactions on Control Systems Technology 12 (1) (2004) 65–74.
[69] M. Eftekhari, L. Marjanovic, P. Angelov, Design and performance of a rule-based controller in a naturally ventilated room, Computers in Industry 52 (3) (2003) 299–326.
[70] J.N. Lygouras, V.S. Kodogiannis, T. Pachidis, K.N. Tarchanidis, C.S. Koukourlis, Variable structure TITO fuzzy-logic controller implementation for a solar air-conditioning system, Applied Energy 85 (4) (2008) 190–203.
[71] A. Maidi, M. Diaf, J.-P. Corriou, Optimal linear PI fuzzy controller design of a heat exchanger, Chemical Engineering and Processing: Process Intensification 47 (5) (2008) 938–945.
[72] Y.-H. Lee, R. Kopp, Application of fuzzy control for a hydraulic forging machine, Fuzzy Sets and Systems 118 (1) (2001) 99–108.
[73] J. Van Amerongen, P.C. Breedveld, Modelling of physical systems for the design and control of mechatronic systems, Annual Reviews in Control 27 (1) (2003) 87–117.
[74] R. Isermann, Mechatronic Systems: Fundamentals, Springer-Verlag, Berlin, Heidelberg, New York, 2003.
[75] W. Dieterle, Mechatronic systems: automotive applications and modern design methodologies, Annual Reviews in Control 29 (2) (2005) 273–277.
[76] K.-J. Åström, T. Hägglund, Benchmark systems for PID control, in: Preprints of IFAC PID'00 Workshop, Terrassa, Spain, (2000), pp. 181–182.
[77] K.-M. Lee, D. Zhou, A real-time optical sensor for simultaneous measurement of 3-DOF motions, IEEE/ASME Transactions on Mechatronics 9 (3) (2004) 499–507.
[78] L. Horváth, I.J. Rudas, Modelling and Solving Methods for Engineers, Elsevier, Academic Press, Burlington, MA, 2004.
[79] J. Vaščák, Fuzzy cognitive maps in path planning, Acta Technica Jaurinensis, Series Intelligentia Computatorica 1 (3) (2008) 467–479.
[80] D. Hládek, J. Vaščák, P. Sinčák, Multi-robot control system for pursuit-evasion problem, Journal of Electrical Engineering 60 (3) (2009) 143–148.
[81] L. Monostori, T. Kis, J. Váncza, B. Kádár, G.F. Erdős, Real-time, cooperative enterprises for customised mass production, International Journal of Computer Integrated Manufacturing 22 (1) (2009) 55–68.
[82] A. Ridluan, M. Manic, A. Tokuhiro, EBaLM-THP—a neural network thermohydraulic prediction model of advanced nuclear system components, Nuclear Engineering and Design 239 (2) (2009) 308–319.
[83] L. Monostori, B.C. Csáji, B. Kádár, A. Pfeiffer, E. Ilie-Zudor, Z. Kemény, M. Szathmári, Towards adaptive and digital manufacturing, Annual Reviews in Control 34 (1) (2010) 118–128.
[84] L. Reznik, Fuzzy Controllers, Newnes-Butterworth-Heinemann, Oxford, 1997.
[85] M.S. Ahmed, U.L. Bhatti, F.M. Al-Sunni, M. El-Shafei, Design of a fuzzy servo-controller, Fuzzy Sets and Systems 124 (2) (2001) 23–247.
[86] S.-K. Cho, H.-H. Lee, A fuzzy-logic antiswing controller for three-dimensional overhead cranes, ISA Transactions 41 (2) (2002) 235–243.
[87] R.-E. Precup, S. Preitl, G. Faur, PI predictive fuzzy controllers for electrical drive speed control: methods and software for stable development, Computers in Industry 52 (3) (2003) 253–270.
[88] H. Schulte, H. Hahn, Fuzzy state feedback gain scheduling control of servo-pneumatic actuators, Control Engineering Practice 12 (5) (2004) 639–650.
[89] X. Gao, Z.-J. Feng, Design study of an adaptive fuzzy-PD controller for pneumatic servo system, Control Engineering Practice 13 (1) (2005) 55–65.
[90] R.-E. Precup, S. Preitl, PI-fuzzy controllers for integral plants to ensure robust stability, Information Sciences 177 (20) (2007) 4410–4429.
[91] R.-E. Precup, S. Preitl, J.K. Tar, M.L. Tomescu, M. Takács, P. Korondi, P. Baranyi, Fuzzy control system performance enhancement by iterative learning control, IEEE Transactions on Industrial Electronics 55 (9) (2008) 3461–3475.
[92] M. Kalyoncu, M. Haydim, Mathematical modelling and fuzzy logic based position control of an electrohydraulic servosystem with internal leakage, Mechatronics 19 (6) (2009) 847–858.
[93] K.K. Ahn, D.Q. Truong, Online tuning fuzzy PID controller using robust extended Kalman filter, Journal of Process Control 19 (6) (2009) 1011–1023.
[94] H.R. Berenji, S. Saraf, P.-W. Chang, S.R. Swanson, Pitch control of the space shuttle training aircraft, IEEE Transactions on Control Systems Technology 9 (3) (2001) 542–551.
[95] S.-F. Wu, C.J.H. Engelen, Q.-P. Chu, R. Babuška, J.A. Mulder, G. Ortega, Fuzzy logic based attitude control of the spacecraft X-38 along a nominal re-entry trajectory, Control Engineering Practice 9 (7) (2001) 699–707.
[96] S.-F. Wu, C.J.H. Engelen, R. Babuška, Q.-P. Chu, J.A. Mulder, Fuzzy logic based full-envelope autonomous flight control for an atmospheric re-entry spacecraft, Control Engineering Practice 11 (1) (2003) 11–25.

[97] S.Y. Lee, H.S. Cho, A fuzzy controller for an electro-hydraulic fin actuator using phase plane method, Control Engineering Practice 11 (6) (2003) 697–708.
[98] B. Kadmiry, D. Driankov, A fuzzy flight controller combining linguistic and model-based fuzzy control, Fuzzy Sets and Systems 146 (3) (2004) 313–347.
[99] A.V. Lovato, E. Araujo, J.D.S. da Silva, Fuzzy decision in airplane speed control, in: Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2006), Vancouver, BC, Canada, (2006), pp. 1578–1583.
[100] E.N. Sanchez, H.M. Becerra, C.M. Velez, Combining fuzzy, PID and regulation control for an autonomous mini-helicopter, Information Sciences 177 (10) (2007) 1999–2022.
[101] C.-H. Cheng, S.-L. Shu, P.-J. Cheng, Attitude control of a satellite using fuzzy controllers, Expert Systems with Applications 36 (3) (2009) 6613–6620.
[102] P.J. Costa Branco, J.A. Dente, Design and experimental evaluation of a fuzzy logic pressure controller for the Airbus 310/320 braking control system, Engineering Applications of Artificial Intelligence 23 (6) (2010) 989–999.
[103] K.-J. Åström, T. Hägglund, The future of PID control, Control Engineering Practice 9 (11) (2001) 1163–1175.
[104] K.-J. Åström, T. Hägglund, PID Controllers Theory: Design and Tuning, Instrument Society of America, Research Triangle Park, NC, 1995.
[105] F. Zheng, Q.-G. Wang, T.H. Lee, X. Huang, Robust PI controller design for nonlinear systems via fuzzy modeling approach, IEEE Transactions on Systems, Man, and Cybernetics—Part A 31 (6) (2001) 666–675.
[106] M. Güzelkaya, İ. Eksin, I. Yeşil, Self-tuning of PID-type fuzzy logic controllers coefficients via relative rate observer, Engineering Applications of Artificial Intelligence 16 (3) (2003) 227–236.
[107] H. Ying, Deriving analytical input–output relationship for fuzzy controllers using arbitrary input fuzzy sets and Zadeh fuzzy AND operator, IEEE Transactions on Fuzzy Systems 14 (5) (2006) 654–662.
[108] A. Mannan, H.A. Talebi, A fuzzy Lyapunov-based control strategy for a macro-micro manipulator: experimental results, IEEE Transactions on Control Systems Technology 15 (2) (2007) 375–383.
[109] F.L. Lewis, K. Liu, Towards a paradigm for fuzzy logic control, Automatica 32 (2) (1996) 167–181.
[110] G.K.I. Mann, B.G. Hu, R.G. Gosine, Analysis of direct action fuzzy PID controller structures, IEEE Transactions on Systems, Man, and Cybernetics—Part B 29 (3) (1999) 371–388.
[111] N.B. Hui, V. Mahendar, D.K. Pratihar, Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches, Fuzzy Sets and Systems 177 (16) (2006) 2171–2204.
[112] B.S. Moon, Equivalence between fuzzy logic controllers and PI controllers for single input systems, Fuzzy Sets and Systems 69 (2) (1995) 105–113.
[113] S. Galichet, L. Foulloy, Fuzzy controllers: synthesis and equivalences, IEEE Transactions on Fuzzy Systems 3 (2) (1995) 140–148.
[114] Y. Ding, H. Ying, S. Shao, Typical Takagi-Sugeno PI and PID fuzzy controllers: analytical structures and stability analysis, Information Sciences 151 (2003) 245–262.
[115] W.Z. Qiao, M. Mizumoto, PID type fuzzy controller and parameter adaptive method, Fuzzy Sets and Systems 78 (1) (1996) 23–35.
[116] C.-L. Chen, S.-N. Wang, C.-T. Hsieh, F.-Y. Chang, Theoretical analysis of crisp-type fuzzy logic controllers using various t-norm sum-gravity inference methods, IEEE Transactions on Fuzzy Systems 6 (1) (1998) 122–136.
[117] G.K.I. Mann, B.G. Hu, R.G. Gosine, A systematic study of fuzzy PID controllers—function based evaluation approach, IEEE Transactions on Fuzzy Systems 9 (5) (2001) 699–712.
[118] J. Zheng, S. Zhao, S. Wei, Application of self-tuning fuzzy PID controller for a SRM direct drive volume control hydraulic press, Control Engineering Practice 17 (12) (2009) 1398–1404.
[119] A. Haj-Ali, H. Ying, Structural analysis of fuzzy controllers with nonlinear input fuzzy sets in relation to nonlinear PID control with variable gains, Automatica 40 (9) (2004) 1551–1559.
[120] Z.Y. Zhao, M. Tomizuka, S. Isaka, Fuzzy gain-scheduling of PID controllers, IEEE Transactions on Systems, Man, and Cybernetics 23 (5) (1993) 1392–1398.
[121] P. Sarma, Multivariable gain-scheduled fuzzy logic control of an exothermic reactor, Engineering Applications of Artificial Intelligence 14 (4) (2001) 457–471.
[122] K.S. Tang, K.F. Man, G. Chen, S. Kwong, An optimal fuzzy PID controller, IEEE Transactions on Industrial Electronics 48 (4) (2001) 757–765.
[123] G.M. Khoury, M. Saad, H.Y. Kanaan, C. Asmar, Fuzzy PID control of a five DOF robot arm, Journal of Intelligent and Robotic Systems 40 (3) (2004) 299–320.
[124] H.X. Li, L. Zhang, K.Y. Cai, G.R. Chen, An improved robust fuzzy-PID controller with optimal fuzzy reasoning, IEEE Transactions on Systems, Man, and Cybernetics—Part B 35 (6) (2005) 1283–1294.
[125] A. Aceves-Lopez, J. Aguilar-Martin, Simplified version of Mamdani's fuzzy controller: the natural logic controller, IEEE Transactions on Fuzzy Systems 14 (1) (2006) 16–30.
[126] A. Bouafia, F. Krim, J.-P. Gaubert, Design and implementation of high performance direct power control of three-phase PWM rectifier, via fuzzy and PI controller for output voltage regulation, Energy Conversion and Management 50 (1) (2009) 6–13.
[127] C. Wakabayashi, M. Embiruçu, C. Fontes, R. Kalid, Fuzzy control of a nylon polymerization semi-batch reactor, Fuzzy Sets and Systems 160 (4) (2009) 537–553.
[128] H.M. Genc, E. Yesil, I. Eksin, M. Guzelkaya, O.A. Tekin, A rule base modification scheme in fuzzy controllers for time-delay systems, Expert Systems with Applications 36 (4) (2009) 8476–8486.
[129] P.P. Angelov, Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems, Springer-Verlag, Heidelberg, New York, 2002.
[130] M. Mahfouf, M.F. Abbod, D.A. Linkens, Online elicitation of Mamdani-type fuzzy rules via TSK-based generalized predictive control, IEEE Transactions on Systems, Man, and Cybernetics—Part B 33 (3) (2003) 465–475.
[131] K. Belarbi, F. Titel, W. Bourebia, K. Benmahammed, Design of Mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm, Engineering Applications of Artificial Intelligence 18 (7) (2005) 875–880.
[132] G.L.O. Serra, C.P. Bottura, Multiobjective evolution based fuzzy PI controller design for nonlinear systems, Engineering Applications of Artificial Intelligence 19 (2) (2006) 157–167.
[133] S.M. Homayouni, T.S. Hong, N. Ismailm, Development of genetic fuzzy logic controllers for complex production systems, Computers & Industrial Engineering 57 (4) (2009) 1247–1257.
[134] R.L. Navale, R.M. Nelson, Use of genetic algorithms to develop an adaptive fuzzy logic controller for a cooling coil, Energy and Buildings 42 (5) (2010) 708–716.
[135] S.-M. Kim, W.-Y. Han, Induction motor servo drive using robust PID-like neuro-fuzzy controller, Control Engineering Practice 14 (5) (2006) 481–487.
[136] H.-C. Lu, J.-C. Chang, M.-F. Yeh, Design and analysis of direct-action CMAC PID controller, Neurocomputing 70 (16–18) (2007) 2615–2625.
[137] S. Alavandar, M.J. Nigam, New hybrid adaptive neuro-fuzzy algorithms for manipulator control with uncertainties—comparative study, ISA Transactions 48 (4) (2009) 497–502.
[138] P. Stewart, D.A. Stone, P.J. Fleming, Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop, Engineering Applications of Artificial Intelligence 17 (3) (2004) 275–284.
[139] F.M. Raimondi, M. Melluso, A new fuzzy robust dynamic controller for autonomous vehicles with nonholonomic constraints, Robotics and Autonomous Systems 52 (2–3) (2005) 115–131.
[140] Z. Sun, R. Xing, C. Zhao, W. Huang, Fuzzy auto-tuning PID control of multiple joint robot driven by ultrasonic motors, Ultrasonics 46 (4) (2007) 303–312.
[141] V.I. Utkin, Sliding Modes in Control and Optimization, Springer-Verlag, Berlin, 1992
[142] R. Palm, Robust control by fuzzy sliding mode, Automatica 30 (9) (1994) 1429–1437.
[143] S. Glower, J. Munighan, Designing fuzzy controllers from a variable structures standpoint, IEEE Transactions on Fuzzy Systems 5 (1) (1997) 138–144.
[144] Q.P. Ha, Q.H. Nguyen, D.C. Rye, H.F. Durrant-Whyte, Fuzzy sliding mode controllers with applications, IEEE Transactions on Industrial Electronics 48 (1) (2001) 38–46.
[145] H. Lee, E. Kim, H.-J. Kang, M. Park, A new sliding-mode control with fuzzy boundary layer, Fuzzy Sets and Systems 120 (1) (2001) 135–143.
[146] F.-J. Lin, R.-J. Wai, Adaptive and fuzzy neural network sliding-mode controllers for motor-quick-return servomechanism, Mechatronics 13 (5) (2003) 477–506.
[147] R.-J. Wai, C.-M. Lin, C.-F. Hsu, Adaptive fuzzy sliding-mode control for electrical servo drive, Fuzzy Sets and Systems 143 (2) (2004) 295–310.
[148] C.-L. Hwang, H.-M. Wu, C.-L. Shih, Fuzzy sliding-mode underactuated control for autonomous dynamic balance of an electrical bicycle, IEEE Transactions on Control Systems Technology 17 (3) (2009) 658–670.
[149] C.-Y. Chen, T.-H.S. Li, Y.-C. Yeh, EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots, Information Sciences 179 (1–2) (2009) 180–195.
[150] M.-K. Chang, An adaptive self-organizing fuzzy sliding mode controller for a 2-DOF rehabilitation robot actuated by pneumatic muscle actuators, Control Engineering Practice 18 (1) (2010) 13–22.
[151] G. Prashanti, M. Chidambaram, Set-point weighted PID controllers for unstable systems, Journal of The Franklin Institute 337 (2–3) (2000) 201–215.
[152] M. Araki, H. Taguchi, Two-degree-of-freedom PID controllers, International Journal of Control, Automation and Systems 1 (4) (2003) 401–411.
[153] A. Visioli, A new design for a PID plus feedforward controller, Journal of Process Control 14 (4) (2004) 457–463.
[154] C.M. Liaw, S.Y. Cheng, Fuzzy two-degrees-of-freedom speed controller for motor drives, IEEE Transactions on Industrial Electronics 42 (2) (1995) 209–216.
[155] A. Visioli, Fuzzy logic based set-point weight tuning of PID controllers, IEEE Transactions on Systems, Man, and Cybernetics—Part A 29 (6) (1999) 587–592.
[156] R.K. Barai, K. Nonami, Optimal two-degree-of-freedom fuzzy control for locomotion control of a hydraulically actuated hexapod robot, Information Sciences 177 (8) (2007) 1892–1915.
[157] R.-E. Precup, S. Preitl, P. Korondi, Fuzzy controllers with maximum sensitivity for servosystems, IEEE Transactions on Industrial Electronics 54 (3) (2007) 1298–1310.
[158] R.-E. Precup, S. Preitl, E.M. Petriu, J.K. Tar, M.L. Tomescu, C. Pozna, Generic two-degree-of-freedom linear and fuzzy controllers for integral processes, Journal of the Franklin Institute 346 (10) (2009) 980–1003.
[159] S.G. Cao, N.W. Rees, G. Feng, Analysis and design of fuzzy control systems using dynamic fuzzy global models, Fuzzy Sets and Systems 75 (1) (1995) 47–62.
[160] L.T. Kóczy, Fuzzy If-Then rule models and their transformation into one another, IEEE Transactions on Systems, Man, and Cybernetics—Part A 26 (5) (1996) 621–637.
[161] J.-S.R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Upper Saddle River, NJ, 1997.
[162] T.A. Johansen, R. Shorten, R. Murray-Smith, On the interpretation and identification of dynamic Takagi-Sugeno models, IEEE Transactions on Fuzzy Systems 8 (3) (2000) 297–313.
[163] J. Abonyi, Fuzzy Model Identification for Control, Birkhäuser, Boston, 2003.
[164] S. Mollov, R. Babuška, J. Abonyi, H.B. Verbruggen, Effective optimization for fuzzy model predictive control, IEEE Transactions on Fuzzy Systems 12 (5) (2004) 661–675.

[165] S. Blažič, I. Škrjanc, S. Gerkšič, G. Dolanc, S. Strmčnik, M.B. Hadjiski, A. Stathaki, Online fuzzy identification for an intelligent controller based on a simple platform, Engineering Applications of Artificial Intelligence 22 (4–5) (2009) 628–638.
[166] H.K. Ahn, H.P.H. Anh, Inverse double NARX fuzzy modeling for system identification, IEEE/ASME Transactions on Mechatronics 15 (1) (2010) 136–148.
[167] K. Tanaka, M. Sugeno, Stability analysis and design of fuzzy control systems, Fuzzy Sets and Systems 12 (2) (1992) 135–156.
[168] S.P. Boyd, L. El Ghaoui, E. Ferron, Linear Matrix Inequalities in Systems and Control Theory, SIAM, Philadelphia, PA, 1994.
[169] G. Feng, Stability analysis of discrete time fuzzy dynamic systems based on piecewise Lyapunov functions, IEEE Transactions on Fuzzy Systems 12 (1) (2004) 22–28.
[170] T.M. Guerra, L. Vermeiren, LMI-based relaxed non-quadratic stabilization conditions for nonlinear systems in Takagi-Sugeno's form, Automatica 40 (5) (2004) 823–829.
[171] A. Kruszewski, T.M. Guerra, New approaches for the stabilization of discrete Takagi-Sugeno fuzzy models, in: Proceedings of 44th IEEE Decision and Control and 2005 European Conference, Seville, Spain, (2005), pp. 3255–3260.
[172] B.J. Rhee, S. Won, A new fuzzy Lyapunov function approach for a Takagi-Sugeno fuzzy control system design, Fuzzy Sets and Systems 157 (9) (2006) 1211–1228.
[173] I. Abdelmalek, N. Golea, A non-quadratic fuzzy stabilization and tracking approach to a two-link robot manipulator control, in: Proceedings of 6th International Conference on Intelligent Systems Design and Applications, Jinan, Shandong, China, (2006), pp. 109–114.
[174] C. Lin, Q.-G. Wang, T.H. Lee, Y. He, Fuzzy weighting-dependent approach to H_∞ filter design for time-delay fuzzy systems, IEEE Transactions on Signal Processing 55 (6) (2007) 2746–2751.
[175] A. Kruszewski, A. Sala, T.M. Guerra, C. Arino, Sufficient and asymptotic necessary conditions for the stabilization of Takagi-Sugeno model, in: Preprints of 3rd IFAC Workshop on Advanced Fuzzy and Neural Control, Valenciennes, France, (2007), p. 6, CD-ROM, paper index MO4-4.
[176] T. Zhang, G. Feng, J. Lu, W. Xiang, Robust constrained fuzzy affine model predictive control with application to a fluidized bed combustion plant, IEEE Transactions on Control Systems Technology 16 (5) (2008) 1047–1056.
[177] L.A. Mozelli, R.M. Palhares, F.O. Souza, E.M.A.M. Mendes, Reducing conservativeness in recent stability conditions of TS fuzzy systems, Automatica 45 (6) (2009) 1580–1583.
[178] H. Gao, Y. Zhao, T. Chen, H_∞ fuzzy control of nonlinear systems under unreliable communication links, IEEE Transactions on Fuzzy Systems 17 (2) (2009) 265–278.
[179] A. Poursamad, A.H. Davaie-Markazi, Robust adaptive fuzzy control of unknown chaotic systems, Applied Soft Computing 9 (3) (2009) 970–976.
[180] H.K. Lam, M. Narimani, Stability analysis and performance design for fuzzy-model-based control system under imperfect premise matching, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 949–961.
[181] K. Tanaka, H. Yoshida, H. Ohtake, H.O. Wang, A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 911–922.
[182] B.-C. Ding, Development of stability research on Takagi-Sugeno fuzzy control systems and approximation of the necessary and sufficient conditions, Fuzzy Information and Engineering 1 (4) (2009) 367–383.
[183] Z. Lendek, R. Babuška, B. De Schutter, Stability of cascaded fuzzy systems and observers, IEEE Transactions on Fuzzy Systems 17 (3) (2009) 641–653.
[184] Z. Lendek, J. Lauber, T.M. Guerra, R. Babuška, B. De Schutter, Adaptive observers for TS fuzzy systems with unknown polynomial inputs, Fuzzy Sets and Systems 161 (15) (2010) 2043–2065.
[185] C. Arino, A. Sala, Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure, Engineering Applications of Artificial Intelligence 20 (8) (2007) 1036–1046.
[186] Y. Yam, P. Baranyi, C.T. Yang, Reduction of fuzzy rule base via singular value decomposition, IEEE Transactions on Fuzzy Systems 7 (2) (1999) 120–132.
[187] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1253–1278.
[188] R.-E. Precup, S. Preitl, I.-B. Ursache, P.A. Clep, P. Baranyi, J.K. Tar, On the combination of tensor product and fuzzy models, in: Proceedings of 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, vol. 2, (2008), pp. 48–53.
[189] P. Baranyi, D. Tikk, Y. Yam, R.J. Patton, From differential equations to PDC controller design via numerical transformation, Computers in Industry 51 (3) (2003) 281–297.
[190] P. Baranyi, TP model transformation as a way to LMI based controller design, IEEE Transactions on Industrial Electronics 51 (2) (2004) 387–400.
[191] Z. Petres, P. Baranyi, P. Korondi, H. Hashimoto, Trajectory tracking by TP model transformation: case study of a benchmark problem, IEEE Transactions on Industrial Electronics 54 (3) (2007) 1654–1663.
[192] F. Kolonić, A. Poljugan, Experimental control design by TP model transformation, in: Proceedings of 2006 International Conference on Mechatronics (ICM 2006), Budapest, Hungary, (2006), pp. 666–671.
[193] S. Nagy, Z. Petres, P. Baranyi, TP Tool—A Matlab toolbox for TP model transformation, in: Proceedings of 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2007), Budapest, Hungary, (2007), pp. 483–495.
[194] R.-E. Precup, L.-T. Dioanca, E.M. Petriu, M.-B. Rădac, S. Preitl, C.-A. Dragoş, Tensor product-based real-time control of the liquid levels in a three tank system, in: Proceedings of 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2010), Montreal, Canada, (2010), pp. 768–773.
[195] P. Baranyi, Tensor-product model-based control of two-dimensional aeroelastic system, Journal of Guidance, Control, and Dynamics 29 (2) (2006) 391–400.
[196] P. Baranyi, Output feedback control of 2-D aeroelastic system, Journal of Guidance, Control, and Dynamics 29 (3) (2006) 762–767.
[197] S. Boverie, B. Demaya, R. Ketata, A. Titli, Performance evaluation of fuzzy controller, in: Proceedings of IFAC Symposium on Intelligent Components and Instruments for Control Applications (SICICA'92), Malaga, Spain, (1992), pp. 105–110.
[198] M. Setnes, Complexity reduction in fuzzy systems, Ph.D. thesis, Delft University of Technology, The Netherlands, 2001.
[199] J.A. Roubos, M. Setnes, J. Abonyi, Learning fuzzy classification rules from labeled data, Information Sciences 150 (1–2) (2003) 77–93.
[200] H. Hellendoorn, R. Baudrexl, Fuzzy-neural traffic control and forecasting, in: Proceedings of FUZZ-IEEE/IFES International Conference, Yokohama, Japan, (1995), pp. 1511–1518.
[201] H. Hellendoorn, Fuzzy logic for paper and pulp management, in: H.B. Verbruggen, R. Babuška (Eds.), Fuzzy Logic Control Advances in Applications, vol. 23, World Scientific Series in Robotics and Intelligent Systems, 1999, pp. 173–191.
[202] H. Hellendoorn, Fuzzy logic for traffic management and control, in: H.B. Verbruggen, R. Babuška (Eds.), Fuzzy Logic Control Advances in Applications, vol. 23, World Scientific Series in Robotics and Intelligent Systems, 1999, pp. 259–273.
[203] W. Hauptmann, K. Heesche, A prototype of an integrated fuzzy-neuro system, in: Proceedings of 2nd European Congress on Fuzzy and Intelligent Technologies (EUFIT'94), Aachen, Germany, (1994), pp. 1101–1104.
[204] W. Hauptmann, K. Heesche, A neural net topology for bidirectional fuzzy-neuro transformation, in: Proceedings of FUZZ-IEEE/IFES International Conference, Yokohama, Japan, (1995), pp. 2187–2194.
[205] J.-S.R. Jang, ANFIS: adaptive network-based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (3) (1993) 665–685.
[206] C.W. Frey, H.B. Kuntze, A neuro-fuzzy supervisory control system for industrial batch processes, IEEE Transactions on Fuzzy Systems 9 (4) (2001) 570–577.
[207] J. Zhang, Modeling and optimal control of batch processes using recurrent neuro-fuzzy networks, IEEE Transactions on Fuzzy Systems 13 (4) (2005) 417–426.
[208] M. Huang, J. Wan, Y. Ma, Y. Wang, W. Li, X. Sun, Control rules of aeration in a submerged biofilm wastewater treatment process using fuzzy neural networks, Expert Systems with Applications 36 (7) (2009) 10428–10437.
[209] J.M. Escaño, C. Bordons, C. Vilas, M.R. García, A.A. Alonso, Neurofuzzy model based predictive control for thermal batch processes, Journal of Process Control 19 (9) (2009) 1566–1575.
[210] A. Das, J. Maiti, R.N. Banerjee, Process control strategies for a steel making furnace using ANN with Bayesian regularization and ANFIS, Expert Systems with Applications 37 (20) (2010) 1075–1085.
[211] J.S. Wang, C.S.G. Lee, Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle, IEEE Transactions on Robotics and Automation 19 (2) (2003) 283–295.
[212] K. Kiguchi, T. Tanaka, T. Fukuda, Neuro-fuzzy control of a robotic exoskeleton with EMG signals, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 481–490.
[213] P.A. Phan, T.J. Gale, Direct adaptive fuzzy control with a self-structuring algorithm, Fuzzy Sets and Systems 159 (8) (2008) 871–899.
[214] C.M. Lin, C.F. Hsu, Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings, IEEE Transactions on Fuzzy Systems 12 (5) (2004) 733–742.
[215] Z.L. Liu, Reinforcement adaptive fuzzy control of wing rock phenomena, IEE Proceedings—Control Theory and Applications 152 (6) (2005) 615–620.
[216] D.M. Liu, G. Naadimuthu, E.S. Lee, Trajectory tracking in aircraft landing operations management using the adaptive neural fuzzy inference system, Computers & Mathematics with Applications 56 (5) (2008) 1322–1327.
[217] S. Kurnaz, O. Cetin, O. Kaynak, Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles, Expert Systems with Applications 37 (2) (2010) 1229–1234.
[218] P. Melin, O. Castillo, Intelligent control of a stepping motor drive using an adaptive neuro-fuzzy inference system, Information Sciences 170 (2–4) (2005) 133–151.
[219] Y.-S. Kung, M.-H. Tsai, FPGA-based speed control IC for PMSM drive with adaptive fuzzy control, IEEE Transactions on Power Electronics 22 (6) (2007) 2476–2486.
[220] D. Bellomo, D. Naso, R. Babuška, Adaptive fuzzy control of a non-linear servo-drive: theory and experimental results, Engineering Applications of Artificial Intelligence 21 (6) (2008) 846–857.
[221] S.V. Ustun, M. Demirtas, Modeling and control of V/f controlled induction motor using genetic-ANFIS algorithm, Energy Conversion and Management 50 (2) (2009) 786–791.
[222] K. Tamani, R. Boukezzoula, G. Habchi, Intelligent distributed and supervised flow control methodology for production systems, Engineering Applications of Artificial Intelligence 22 (7) (2009) 1104–1116.
[223] İ. Asiltürk, A. Ünüvar, Intelligent adaptive control and monitoring of band sawing using a neural-fuzzy system, Journal of Materials Processing Technology 209 (5) (2009) 2302–2313.
[224] S. Soyguder, H. Alli, Fuzzy adaptive control for the actuators position control and modeling of an expert system, Expert Systems with Applications 27 (3) (2010) 2072–2080.

[225] P. Baranyi, L.T. Kóczy, T.D. Gedeon, A generalized concept for fuzzy rule interpolation, IEEE Transactions on Fuzzy Systems 12 (6) (2004) 820–837.
[226] Z.C. Johanyák, S. Kovács, Fuzzy rule interpolation based on polar cuts, in: B. Reusch (Ed.), Computational Intelligence, Theory and Applications, Springer-Verlag, Berlin, Heidelberg, New York, 2006, pp. 499–511.
[227] Z.C. Johanyák, D. Tikk, S. Kovács, K.K. Wong, Fuzzy rule interpolation Matlab toolbox—FRI toolbox, in: Proceedings of 15th International Conference on Fuzzy Systems (FUZZ-IEEE'06), Vancouver, Canada, (2006), pp. 1427–1433.
[228] S. Blažič, I. Škrjanc, Design and stability analysis of fuzzy model-based predictive control—a case study, Journal of Intelligent and Robotic Systems 49 (3) (2007) 279–292.
[229] J. Abonyi, L. Nagy, F. Szeifert, Fuzzy model-based predictive control by instantaneous linearization, Fuzzy Sets and Systems 120 (1) (2001) 109–122.
[230] S. Blažič, I. Škrjanc, D. Matko, Globally stable direct fuzzy model reference adaptive control, Fuzzy Sets and Systems 139 (1) (2003) 3–33.
[231] H. Sarimveis, G. Bafas, Fuzzy model predictive control of non-linear processes using genetic algorithms, Fuzzy Sets and Systems 139 (1) (2003) 59–80.
[232] A. Flores, D. Saez, J. Araya, M. Berenguel, A. Cipriano, A fuzzy predictive control of a solar power plant, IEEE Transactions on Fuzzy Systems 13 (1) (2005) 58–68.
[233] K. Belarbi, F. Megri, A stable model-based fuzzy predictive control based on fuzzy dynamic programming, IEEE Transactions on Fuzzy Systems 15 (4) (2007) 746–754.
[234] J.-C. de Barros, A.L. Dexter, On-line identification of computationally undemanding evolving fuzzy models, Fuzzy Sets and Systems 158 (18) (2007) 1997–2012.
[235] S. Ertugrul, Predictive modeling of human operators using parametric and neuro-fuzzy models by means of computer-based identification experiment, Engineering Applications of Artificial Intelligence 21 (2) (2008) 259–268.
[236] A. Núñez, D. Sáez, S. Oblak, I. Škrjanc, Fuzzy-model-based hybrid predictive control, ISA Transactions 48 (1) (2009) 24–31.
[237] Z. Preitl, Model based design methods for speed control applications, Ph.D. thesis, "Politehnica" University of Timisoara, Timisoara, Romania, 2008.
[238] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper-Saddle River, NJ, 2001.
[239] O. Castillo, P. Melin, Type-2 Fuzzy Logic Theory and Applications, Springer-Verlag, Berlin, Heidelberg, New York, 2008.
[240] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 524–539.
[241] D. Wu, W.W. Tan, A simplified type-2 fuzzy logic controller for real-time control, ISA Transactions 45 (4) (2006) 503–516.
[242] R. Sepúlveda, O. Castillo, P. Melin, A. Rodríguez-Díaz, O. Montiel, Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic, Information Sciences 177 (10) (2007) 2023–2048.
[243] M.-Y. Hsiao, T.-H.S. Li, J.-Z. Lee, C.-H. Chao, S.-H. Tsai, Design of interval type-2 fuzzy sliding-mode controller, Information Sciences 178 (6) (2008) 1696–1716.
[244] C.-F. Juang, C.-H. Hsu, Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control, IEEE Transactions on Industrial Electronics 56 (10) (2009) 3931–3940.

Radu-Emil Precup was born in Lugoj, Romania, in 1963. He received the Dipl.Ing. (Hons.) degree in Automation and Computers from "Traian Vuia" Polytechnic Institute of Timisoara, Timisoara, Romania, in 1987, the Diploma in Mathematics from the West University of Timisoara, in 1993, and the Ph.D. degree in Automatic Systems from the "Politehnica" University of Timisoara (PUT), Timisoara, in 1996. From 1987 to 1991, he was with Infoservice S.A., Timisoara. He is currently with the PUT, where he became a Professor in the Department of Automation and Applied Informatics in 2000, and he is currently a Doctoral Supervisor of Automation and Systems Engineering. He is also an Honorary Professor and a Member of the Doctoral School of Applied Informatics with the Óbuda University (previously named Budapest Tech Polytechnical Institution), Budapest, Hungary. He is currently the Deputy Director of the Research Centre in Automation and Computers with the PUT, the Director of the Automation and Applied Informatics Division, and the Head of the Students Information and Counselling Office (OICS) with the Faculty of Automation and Computers, PUT. From 1999 to 2009, he held research and teaching positions with the Université de Savoie, Chambéry and Annecy, France, Budapest Tech Polytechnical Institution, Budapest, Hungary, Vienna University of Technology, Vienna, Austria, and Budapest University of Technology and Economics, Budapest, Hungary. He has been an Editor-in-Chief of the International Journal of Artificial Intelligence since 2008 and he is also on the editorial board of several other prestigious journals. He is the author or coauthor of more than 120 papers published in various scientific journals, refereed conference proceedings, and contributions to books. His research interests include mainly development and analysis of intelligent control systems, theory and applications of soft computing, computer-aided design of control systems, modelling, and optimization. Prof. Precup is a Senior Member, IEEE, a member of the Subcommittee on Computational Intelligence as part of the IEEE Industrial Electronics Society, the Task Force on Educational aspects of standards of Computational Intelligence as part of the Technical Committee on Standards in the IEEE Computational Intelligence Society, the Virtual Reality Task Force of the Intelligent Systems Applications Technical Committee of the IEEE Computational Intelligence Society, the International Federation of Automatic Control (IFAC) Technical Committee on Computational Intelligence in Control (previously named Cognition and Control), the European Society for Fuzzy Logic and Technology (EUSFLAT), the Hungarian Fuzzy Association, and the Romanian Society of Control Engineering and Technical Informatics. He was the recipient of the "Grigore Moisil" Prize from the Romanian Academy in 2005 for his contribution on fuzzy control.

Hans Hellendoorn was born in 1964. He studied computer science at the Delft University of Technology (TU Delft). Next, he did Ph.D.-research in Delft and at the National Aerospace Laboratory in Amsterdam resulting, in 1990, in a Ph.D. thesis "Reasoning with Fuzzy Logic". From 1990 to 1997, he worked as department head in a research group on fuzzy control at Siemens R&D in Munich, Germany, with many practical applications, from power plants to cars and from chips to telecommunication networks. From 1997 to 2000, he was department head for training simulation at Siemens in the Netherlands. From 2001 to 2009, he was responsible for innovation management. In 1999 he was appointed as a Part Time Professor for Control Theory at the Faculty for Electrical Engineering. Since 2004, he is a Professor at the Delft Center for Systems and Control (DCSC), TU Delft. He is the co-author of three scientific books and the author and co-author of more than 140 scientific publications on computational intelligence and control. He has supervised more than 60 M.Sc. students and 15 Ph.D. students in Germany, The Netherlands and France. His research interests include multi-agent control of large-scale hybrid systems, such as urban or national traffic networks, electricity networks, water networks and logistic networks, road traffic control, such as traffic light control, intelligent speed systems and dynamic traffic management, and control and monitoring of harbor cranes. Prof. Hellendoorn was the president of the European Chapter of the International Fuzzy Systems Association (IFSA) and the coordinator of the Working Group on Fuzzy Sets (EUROFUSE) of the Association of European Operational Research Societies (EURO). He is a member of the board of many scientific and social foundations.

Grey Wolf Optimizer Algorithm-Based Tuning of Fuzzy Control Systems With Reduced Parametric Sensitivity

Radu-Emil Precup, *Senior Member, IEEE*, Radu-Codrut David, and Emil M. Petriu, *Fellow, IEEE*

***Abstract*—This paper proposes an innovative tuning approach for fuzzy control systems (CSs) with a reduced parametric sensitivity using the Grey Wolf Optimizer (GWO) algorithm. The CSs consist of servo system processes controlled by Takagi–Sugeno–Kang proportional-integral fuzzy controllers (TSK PI-FCs). The process models have second-order dynamics with an integral component, variable parameters, a saturation, and dead-zone static nonlinearity. The sensitivity analysis employs output sensitivity functions of the sensitivity models defined with respect to the parametric variations of the processes. The GWO algorithm is used in solving the optimization problems, where the objective functions include the output sensitivity functions. GWO's motivation is based on its low-computational cost. The tuning approach is validated in an experimental case study of a position control for a laboratory nonlinear servo system, and TSK PI-FCs with a reduced process small time constant sensitivity are offered.**

***Index Terms*—Experimental results, fuzzy control systems (CSs), Grey Wolf Optimizer (GWO), parametric sensitivity, servo systems.**

I. Introduction

SOFT computing techniques have recently proved to be effective solutions for resolving optimization problems (OPs) in various domains. Such techniques, with focus on modeling and control applications, include fuzzy logic [1]–[4], neural networks [5]–[7], probabilistic reasoning [8]–[10], knowledge-based systems [11], [12], heuristic learning and search algorithms [13], [14], and the combinations that lead to hybrid techniques [15]–[17]. Optimal tuning of fuzzy controllers is amongst those challenging OPs as they involve nonconvex or nondifferentiable objective functions (o.f.s), as consequent of controllers' structures and nonlinearities, of processes' complexity found in industrial applications, and of specific performance specifications that can lead to multiobjective OPs. The optimal control of servo systems is a representative OP, as it requires the minimization of o.f.s expressed as integral- or sum-type quadratic performance indices with the vector variables mapped onto the controller tuning parameters.

Fuzzy control has proven to be an efficient control solution in servo systems and mechatronics applications. Nevertheless, the combinations with various nature-inspired optimization algorithms (NIOAs), referred to also as metaheuristics, have a significant impact on the performance of fuzzy control systems (CSs). Some recent applications of NIOAs to the parameter tuning of fuzzy controllers for servo systems will be briefly discussed as follows. Industrial application overviews are given in [18]–[20]. A proportional-integral-derivative (PID) fuzzy controller is tuned in [21] using a particle swarm optimization (PSO)-based approach and tested experimentally on an electrical direct current (dc) drive benchmark. A hybrid PSO and pattern search optimized PI-fuzzy controller is proposed in [22], applied to the automatic generation control of multiarea power systems and validated by simulation. The gravitational search algorithm (GSA) is applied in several versions in [23] and [24] to the optimal tuning of PI- and PID-fuzzy controllers for dc servo systems and load frequency control in power systems and tested by digitally simulated and experimental results. Charged system search algorithms are suggested in [25] and applied to the optimal tuning of PI-fuzzy controllers for dc servo systems.

The Grey Wolf Optimizer (GWO) algorithm [26] has been developed by mimicking gray wolf social hierarchy and hunting habits. The social hierarchy is simulated by categorizing the population of search agents into four types of individuals, i.e., alpha, beta, delta, and omega, based on their fitness. The search process is modeled with the aim of mimicking the hunting behavior of gray wolfs making use of three stages, i.e., searching, encircling, and attacking the prey. The first two stages are dedicated to exploration and the last one covers the exploitation. The reduced number of search parameters is an important advantage of GWO algorithms reflected in various applications: blackout risk prevention in smart grids [27], training multilayer perceptrons [28], optimal reactive power dispatch [29],

Manuscript received January 12, 2016; revised May 16, 2016 and July 6, 2016; accepted August 3, 2016. Date of publication September 8, 2016; date of current version December 9, 2016. This work was supported in part by grants from the partnerships in priority areas—PN II program of the Romanian Ministry of National Education and Scientific Research (MENCS)—the Executive Agency for Higher Education, Research, Development and Innovation Funding (UEFISCDI), under Project PN-II-PT-PCCA-2013-4-0544 and Project PN-II-PT-PCCA-2013-4-0070, in part by a grant from the Romanian National Authority for Scientific Research (CNCS)—UEFISCDI under Project PN-II-RU-TE-2014-4-0207, and in part by a grant from the NSERC of Canada.

R.-E. Precup is with the Department of Automation and Applied Informatics, Politehnica University of Timisoara, Timisoara 300223, Romania. He is also with the School of Engineering, Edith Cowan University, Joondalup, WA 6027, Australia (e-mail: radu.precup@upt.ro).

R.-C. David is with the Department of Automation and Applied Informatics, Politehnica University of Timisoara, Timisoara 300223, Romania (e-mail: davidradu@gmail.com).

E. M. Petriu is with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: petriu@eecs.uottawa.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2016.2607698

popular benchmarks specific to optimization [30], hyperspectral band selection [31], and optimal tuning of PID-fuzzy controllers [32].

Using the previous results on the optimal tuning of Takagi–Sugeno–Kang PI-fuzzy controllers (TSK PI-FCs) by means of NIOAs [20], [23], [25], this paper has a twofold contribution. First, it introduces an easily understandable GWO algorithm. Second, it applies the GWO algorithm to the optimal tuning of TSK PI-FCs. The presentation is focused on servo system processes that are modeled by second-order dynamics with an integral component, variable parameters, a saturation, and dead-zone static nonlinearity.

Out of the objectives set for this paper, the first is the introduction of a reduced sensitivity design approach of TSK PI-FCs with respect to the small time constant of the process, by avoiding the use of a simplified and idealized linear/linearized model of the process. With the aim of achieving this, a sensitivity analysis with respect to process parametric variations is required in order to handle the unwanted parametric variations in a proper manner and express the output sensitivity functions as well. The simplified process models used in the design and tuning of fuzzy CSs involve the derivation of sensitivity models inserted in the o.f.s. Solving the OPs in an analytical way is rather complicated as it involves unmanageable computational costs; therefore, our GWO-based tuning approach is employed to deal with these complications.

The second objective is the use of GWO algorithm to solve the previously mentioned OPs. The simplicity and transparency of GWO proved in this paper ensure its generality and potential in solving various OPs [32]–[37] with additional economical or operational constraints [38]–[42].

The third objective is the application of the extended symmetrical optimum (ESO) method [43], [44] and the modal equivalence principle [45], with the aim of reducing the number of TSK PI-FC parameters, thus simplifying variables and o.f. expression.

The paper treats the following topics: Section II describes the OP, the process, and fuzzy controller models. The new GWO algorithm and the tuning approach for TSK PI-FCs dedicated to the servo system processes modeled by second-order dynamics with an integral component, variable parameters, a saturation, and dead-zone static nonlinearity are presented in Section III. Section IV treats the case study that deals with the GWO-based tuning of a TSK PI-FC for the angular position control of a nonlinear dc servo system. Experimental results and a comparison with other NIOAs are included. The conclusions are presented in Section V.

II. Optimization Problem and Models

The fuzzy CS structure is given in Fig. 1, where FC is the fuzzy controller, *P* is the nonlinear process, *F* is the set-point filter, *r* is the set point (the reference input), r_1 is the filtered reference input, y is the controlled output, u is the control signal, and $e = r_1 - y$ is the control error. Disturbance inputs are not considered in Fig. 1 because the integral component in the controller ensures the disturbance rejection.

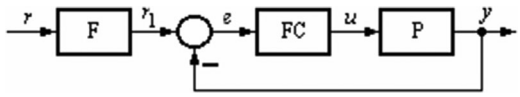

Fig. 1. Set-point filter fuzzy control system structure.

Fig. 1 illustrates a set-point filter CS structure in the framework of two-degree-of-freedom CS structures. The parameters of both FC and *F* blocks can be tuned.

The state-space model of *P* specific to the nonlinear servo systems mentioned in the previous sections is

$$m(t) = \begin{cases} -1, & \text{if } u(t) \le -u_b \\ [u(t)+u_c]/(u_b-u_c), & \text{if } -u_b < u(t) < -u_c \\ 0, & \text{if } -u_c \le |u(t)| \le u_a \\ [u(t)-u_a]/(u_b-u_a), & \text{if } u_a < u(t) < u_b \\ 1, & \text{if } u(t) \ge u_b \end{cases}$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1/T_\Sigma \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ k_P/T_\Sigma \end{bmatrix} m(t)$$

$$y(t) = [1\ 0][x_1(t)\ x_2(t)]^T \quad (1)$$

where $t \ge 0$ is the continuous time, $k_P > 0$ is the process gain, $T_\Sigma > 0$ is the small time constant, the control signal $u(t)$ applied to the dc motor is a pulse width modulated duty cycle, $x_1(t) = \alpha(t)$ is the angular position, $x_2(t) = \omega(t)$ is the angular speed, and the superscript T indicates matrix transposition. The variable $m(t)$ is the output of the saturation and dead-zone static nonlinearity, which is modeled by the first equation in (1) with the parameters u_a, u_b, and u_c, $0 < u_a < u_b$, $0 < u_c < u_b$.

The state-space model (1) includes the actuator and measurement instrumentation dynamics. The nonlinearity in (1) is not symmetric, and this offers a more general model compared with that investigated in [23].

The nonlinearity in (1) is neglected in the simplified model of the process expressed as the transfer function $P(s)$

$$P(s) = k_{EP}/[s(1+T_\Sigma s)] \quad (2)$$

where k_{EP} is the equivalent process gain

$$k_{EP} = \begin{cases} k_P/(u_b-u_c), & \text{if } -u_b < u(t) < -u_c \\ k_P/(u_b-u_a), & \text{if } u_a < u(t) < u_b. \end{cases} \quad (3)$$

It is convenient to use the transfer function $P(s)$ in the linear and fuzzy controller design and tuning in two cases out of the five cases concerning the nonlinearity in (1). As shown in [43] and [44], PI controllers can cope with the process modeled in (2) in terms of Fig. 1, with a PI controller instead of FC. The transfer function of the PI controller is

$$C(s) = k_c(1+sT_i)/s = k_C[1+1/(sT_i)], k_C = k_cT_i \quad (4)$$

where $k_C > 0$ (or $k_c > 0$) is the controller gain and $T_i > 0$ is the integral time constant.

Considering that the process parameters k_P and T_Σ are variable and the other ones are constant, the process parameter vector is

$$\boldsymbol{\alpha} = [\alpha_1\ \alpha_2]^T = [k_P\ T_\Sigma]^T. \quad (5)$$

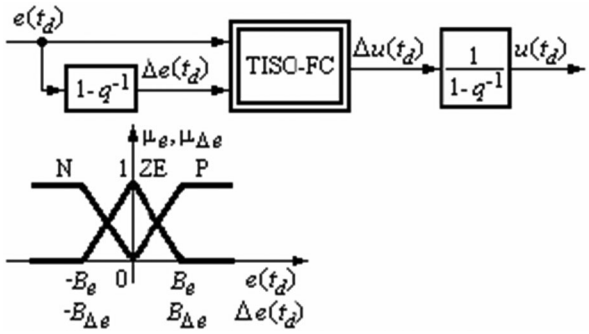

Fig. 2. Structure and input membership functions of TSK PI-FC.

The definitions of the state sensitivity functions $\lambda_\upsilon, \upsilon = 1, ..., n$, and of the output sensitivity function σ are

$$\lambda_\upsilon = \left[\frac{\partial x_\upsilon}{\partial \alpha_j}\right]_0, \ \sigma = \left[\frac{\partial y}{\partial \alpha_j}\right]_0, \ \upsilon = 1, ..., n, j \in \{1, 2\} \quad (6)$$

where the subscript 0 indicates the nominal value of the process parameter $\alpha_j, j \in \{1, 2\}$, and n is the number of state variables of the fuzzy CS. The state sensitivity models of the fuzzy CS with respect to α_j are derived using (6), with $n = 4$ for TSK-PI-FCs.

Using the notation $\boldsymbol{\rho}$ for the controller parameter vector, the OP that ensures the sensitivity reduction with respect to the modifications of the process parameter α_j is defined as

$$\boldsymbol{\rho}^* = \arg\min_{\boldsymbol{\rho} \in D_{\boldsymbol{\rho}}} J(\boldsymbol{\rho}), J(\boldsymbol{\rho}) = \sum_{t_d=0}^{\infty} \{e^2(t_d, \boldsymbol{\rho}) + \gamma^2 [\sigma(t_d, \boldsymbol{\rho})]^2\} \quad (7)$$

where γ is the weighting parameter, $\boldsymbol{\rho}^*$ is the optimal controller parameter vector (the optimal value of $\boldsymbol{\rho}$), $D_{\boldsymbol{\rho}}$ is the feasible domain of $\boldsymbol{\rho}$, $J(\boldsymbol{\rho})$ is the o.f., and $t_d \in \boldsymbol{Z}, t_d \geq 0$, is the discrete-time argument.

The ESO method [43], [44] is applied to tune the PI controller parameters in (4), and it guarantees a tradeoff to the CS performance specifications (maximum values of CS performance indices, i.e., percent overshoot, settling time, and rise time) of the linear CS using only one design parameter β within the largest recommended domain $1 < \beta \leq 20$. The PI tuning conditions specific to the ESO method are

$$k_c = 1/(\beta\sqrt{\beta} k_{\mathrm{EP}} T_\Sigma^2), \ T_i = \beta T_\Sigma, \ k_C = 1/(\sqrt{\beta} k_{\mathrm{EP}} T_\Sigma) \quad (8)$$

and the transfer function of the set-point filter that exhibits the CS performance improvement by the cancellation of a closed-loop CS zero is

$$F(s) = 1/(1 + \beta T_\Sigma s). \quad (9)$$

The TSK PI-FCs are designed and tuned using the knowledge from the PI controller structure in order to ensure an additional CS performance improvement. The structure and the input membership functions of a cost-effective TSK PI-FC are presented in Fig. 2, where q^{-1} is the backward shift operator, TISO-FC is the two-inputs single-output fuzzy controller block that produces a nonlinear input–output static map, $\Delta e(t_d)$ is the increment of control error, and $\Delta u(t_d)$ is the increment of control signal.

The two increments result by discretizing the continuous-time PI controller by Tustin's method, which leads to the recurrent equation of the incremental discrete-time PI controller

$$\Delta u(t_d) = K_P[\Delta e(t_d) + \mu e(t_d)] \quad (10)$$

and its parameters K_P and μ

$$K_P = k_c(T_i - T_s/2), \mu = 2T_s/(2T_i - T_s) \quad (11)$$

where $T_s > 0$ is the sampling period.

As shown in [25], the TISO-FC block employs the weighted average method for defuzzification, and the SUM and PROD operators in the inference engine. The rule base is

$$\begin{aligned} &\mathrm{IF}(e(t_d)\,\mathrm{IS}\,N\,\mathrm{AND}\,\Delta e(t_d)\,\mathrm{IS}\,N)\mathrm{OR}\,(e(t_d)\,\mathrm{IS}\,P \\ &\mathrm{AND}\,\Delta e(t_d)\,\mathrm{IS}\,P)\,\mathrm{THEN}\,\Delta u(t_d) = \eta K_P[\Delta e(t_d) + \mu e(t_d)] \\ &F(e(t_d)\,\mathrm{IS\,ZE})\mathrm{OR}\,(e(t_d)\,\mathrm{IS}\,N\mathrm{AND}\,\Delta e(t_d)\,\mathrm{IS\,ZE})\mathrm{OR} \\ &\quad (e(t_d)\mathrm{IS}\,N\mathrm{AND}\,\Delta e(t_d)\,\mathrm{IS}\,P)\mathrm{OR}\,(e(t_d)\mathrm{IS}\,P \\ &\quad \mathrm{AND}\,\Delta e(t_d)\mathrm{IS\,ZE})\,\mathrm{OR}\,(e(t_d)\mathrm{IS}\,P\,\mathrm{AND}\,\Delta e(t_d)\mathrm{IS}\,P) \\ &\quad \mathrm{THEN}\,\Delta u(t_d) = K_P[\Delta e(t_d) + \mu e(t_d)]. \end{aligned} \quad (12)$$

The parameter η, with the largest domain $0 < \eta < 1$, is introduced to mitigate the fuzzy CS overshoot, which occurs if the two TISO-FC inputs have the same signs. This controller structure and the rule base given in (12) indicate that our cost-effective TSK PI-FC behaves as a bumpless interpolator between two linear PI controllers.

The modal equivalence principle [45] is applied to this TSK PI-FC resulting in the tuning equation

$$B_{\Delta e} = \mu B_e \quad (13)$$

which along with (8), (10), and (11) give the parameter vector of TSK PI-FC

$$\boldsymbol{\rho} = [\beta \ B_e \ \eta]^T. \quad (14)$$

The parameter vector expressed in (14), associated with the OP defined in (7), will be obtained in the next section by the GWO-based tuning approach.

Accepting that u is changing at the discrete-sampling intervals and the presence of the zero-order hold, the derivation of the output sensitivity function $\sigma(t_d)$ requires the state sensitivity model of the fuzzy CS. With this regard, the state variables x_3 and x_4 of the TSK PI-FC are defined in terms of [23]

$$x_3(t_d) = u(t_d - 1), x_4(t_d) = e(t_d - 1) \quad (15)$$

and the state variable of F, x_5, is its output.

For a constant fuzzy CS input, namely $r(t) = \mathrm{const}$, assuming $r_1(t) = \mathrm{const}$ for simplicity, the discrete-time state sensitivity model of the fuzzy CS with respect to T_Σ is obtained by applying (6) to the discrete-time state-space model of the fuzzy CS with the state variables $x_1, x_2, \ldots, x_5$. The nonlinear input–output static map of TISO-FC is involved in the expression of $\sigma(t_d)$.

III. GWO-Based Tuning Approach

The standard operating mechanism of GWO algorithms [26] starts with the initialization of the agents that comprise the pack. A total number of N agents (i.e., gray wolves) are used, and each agent has a position vector $\boldsymbol{X}_i$ associated

$$\boldsymbol{X}_i = \left[\, x_i^1 \ \ldots \ x_i^f \ \ldots \ x_i^q \,\right]^T, i = 1, \ldots, N \tag{16}$$

where x_i^f is the position of the i^{th} agent in f^{th} dimension, $f = 1, ..., q$.

The GWO algorithm's search process continues with the exploration stage represented by the search for the prey. During this stage, the position of the alpha α, beta β, and delta δ agents dictates the search pattern by diverging from other agents and converging on the prey, represented here by the optimal solution.

The exploitation stage is represented by the attack on the prey. The top three agents constrain the other agents, viz., the omegas ω, to update their positions according to theirs. The following notations are used in the sequel for the top three agent position vectors, i.e., the first three best solutions obtained so far (or the alpha, beta, and delta solutions):

$$\boldsymbol{X}_l(k) = \left[\, x_l^1(k) \ \ldots \ x_l^f(k) \ \ldots \ x_l^q(k) \,\right]^T, \ l \in \{\alpha, \beta, \delta\} \tag{17}$$

where k is the current iteration index, $k = 1, ..., k_{\max}$, $k_{\max}$ is the maximum number of iterations, and the vector solutions $\boldsymbol{X}_\alpha(k)$, $\boldsymbol{X}_\beta(k)$, and $\boldsymbol{X}_\delta(k)$ fulfill the conditions

$$\begin{aligned} J(\boldsymbol{X}_\alpha(k)) &= \min_{i=1\ldots N} \{J(\boldsymbol{X}_i(k)), \boldsymbol{X}_i(k) \in D_{\boldsymbol{\rho}}\} \\ J(\boldsymbol{X}_\beta(k)) &= \min_{i=1\ldots N} \{J(\boldsymbol{X}_i(k)), \boldsymbol{X}_i(k) \in D_{\boldsymbol{\rho}} \backslash \{\boldsymbol{X}_\alpha(k)\}\} \\ J(\boldsymbol{X}_\delta(k)) &= \min_{i=1\ldots N} \{J(\boldsymbol{X}_i(k)), \boldsymbol{X}_i(k) \\ &\qquad \in D_{\boldsymbol{\rho}} \backslash \{\boldsymbol{X}_\alpha(k), \boldsymbol{X}_\beta(k)\}\}. \end{aligned} \tag{18}$$

The conditions (18) also result in

$$J(\boldsymbol{X}_\alpha(k)) < J(\boldsymbol{X}_\beta(k)) < J(\boldsymbol{X}_\delta(k)). \tag{19}$$

A set of search coefficient vectors, $\boldsymbol{A}_l(k)$ and $\boldsymbol{C}_l(k)$, are next defined

$$\begin{aligned} \boldsymbol{A}_l(k) &= [\, a_l^1(k) \ ... \ a_l^f(k) \ ... \ a_l^q(k) \,]^T \\ \boldsymbol{C}_l(k) &= [\, c_l^1(k) \ ... \ c_l^f(k) \ ... \ c_l^q(k) \,]^T, l \in \{\alpha, \beta, \delta\} \end{aligned} \tag{20}$$

with the components

$$a_l^f(k) = a^f(k)(2r^f - 1), \ c_l^f(k) = 2r^f, \ l \in \{\alpha, \beta, \delta\} \tag{21}$$

where r^f are uniformly distributed random numbers within $0 \le r^f \le 1$, $f = 1, ..., q$, and the coefficients $a^f(k)$ are linearly decreased from 2 to 0 during the search process

$$a^f(k) = 2[1 - (k-1)/(k_{\max} - 1)], f = 1, ..., q. \tag{22}$$

The approximate distances between the current solution and alpha, beta, and delta solutions, i.e., $d_\alpha^i(k)$, $d_\beta^i(k)$, and $d_\delta^i(k)$, respectively, are computed using the formula

$$d_l^i(k) = |c_l^f(k) x_l^f(k) - x_i^f(k)|, \ i = 1, ..., N, \ l \in \{\alpha, \beta, \delta\}. \tag{23}$$

Using the notations $\boldsymbol{X}^l(k)$, for the updated alpha, beta, and delta solutions, $l \in \{\alpha, \beta, \delta\}$, respectively,

$$\boldsymbol{X}^l(k) = [\, x^{l1}(k) \ \ldots \ x^{lf}(k) \ \ldots \ x^{lq}(k) \,]^T, \ l \in \{\alpha, \beta, \delta\} \tag{24}$$

the components of these solutions are obtained as

$$\begin{aligned} x^{lf}(k) = \ & x_l^f(k) - a_l^f(k) d_l^i(k), \ f = 1, ..., q, \\ & i = 1, ..., N, \ l \in \{\alpha, \beta, \delta\} \end{aligned} \tag{25}$$

and the updated vector solution $\boldsymbol{X}_i(k+1)$ is obtained as the arithmetic mean of these solutions

$$\boldsymbol{X}_i(k+1) = (\boldsymbol{X}^\alpha(k) + \boldsymbol{X}^\beta(k) + \boldsymbol{X}^\delta(k))/3, \ i = 1, ..., N. \tag{26}$$

The GWO algorithm that solves the OP defined in (7) consists of the following steps:

Step 1: The initial random gray wolf population, represented by N agents' positions in the q–dimensional search space, is generated. The iteration index is initialized to $k = 0$ and the maximum number of iterations is set to $k_{\max}$.

Step 2: The performance of each population member is evaluated by simulations and/or experiments conducted on the fuzzy CSs. The evaluation leads to the o.f. value in terms of mapping the GWO algorithm onto the OP using

$$\boldsymbol{X}_i(k) = \boldsymbol{\rho}, i = 1, ..., N. \tag{27}$$

Step 3: The first three best solutions obtained so far, i.e., $\boldsymbol{X}_\alpha(k)$, $\boldsymbol{X}_\beta(k)$, and $\boldsymbol{X}_\delta(k)$, are identified using (18).

Step 4: The search coefficient vectors are calculated using (20), (21), and (22).

Step 5: The agents are moved to their new positions by computing $\boldsymbol{X}_i(k+1)$ in terms of (23)–(26).

Step 6: The updated vector solution $\boldsymbol{X}_i(k+1) \in D_{\boldsymbol{\rho}}$ is validated by checking the steady-state condition for the fuzzy CS with TSK PI-FC tuning parameters $\boldsymbol{\rho} = \boldsymbol{X}_i(k+1)$ so far

$$|y(t_{df}) - r(t_{df})| \le \varepsilon_y |r(t_{df}) - r(0)| \tag{28}$$

where t_{df} is the final time moment. Theoretically, $t_{df} \to \infty$ as indicated in (7), but t_{df} takes practically a finite value to capture the transients in the fuzzy CS responses. Since a quasi-continuous digital PI controller is used in the design of the TSK PI-FC, the parameter $\varepsilon_y > 0$ is chosen using the expression of the controlled output in continuous time

$$\begin{aligned} y(t) = y(0) &+ T_\Sigma[1 - \exp(-t/T_\Sigma)] x_2(0) \\ &+ k_P \int_0^t [1 - \exp(\tau - t)/T_\Sigma] m(u(\tau)) d\tau. \end{aligned} \tag{29}$$

Supposing for the fuzzy CS shown in Fig. 1 that the initial control signal is within $-u_c \le |u(0)| \le u_a$, the steady-state value of the controlled output will be

$$y(t_{df}) = \lim_{t \to \infty} y(t) = y(0) + T_\Sigma x_2(0) \tag{30}$$

which is generally nonzero for nonzero initial conditions, and indicates that the fuzzy CS is not globally asymptotically

stable because of the dead-zone static nonlinearity. Therefore, using (28) and (30), the condition to set the value of ε_y is

$$\varepsilon_y > |y(0) + T_\Sigma x_2(0) - r(t_{df})|/|r(t_{df}) - r(0)|. \quad (31)$$

Stability conditions can be used instead of the steady-state condition (28). Rigorous proofs related to stability analyses are given, for example, in [46] and [47].

Step 7: The iteration index k is incremented and the algorithm continues with step 2 until $k_{\max}$ is reached.

Step 8: The algorithm is stopped and the final solution obtained so far is saved as

$$\boldsymbol{\rho}^* = \arg \min_{i=1,\ldots,N} J(\boldsymbol{X}_i(k_{\max})). \quad (32)$$

Aspects concerning the convergence of GWO algorithms are presented in [26] and [48].

The ESO method and the modal equivalence principle lead to the o.f. $J(\boldsymbol{\rho})$ with just three variables ($q = 3$) that belong to the controller parameter vector

$$\boldsymbol{\rho} = [\,\rho_1\ \rho_2\ \rho_3\,]^T = [\,\beta\ B_e\ \eta\,]^T. \quad (33)$$

Concluding, the novel tuning approach dedicated to TSK PI-FCs is formulated in terms of the following design steps:

Step A: The ESO method is applied to tune the parameters of the linear PI controllers, T_s is set and the discrete-time PI controllers modeled in (10) are obtained. The state sensitivity models of fuzzy CS are derived.

Step B: The weighting parameter γ in (7) is set to meet the performance specifications of the fuzzy CS. The parameter t_{df} is set according to step 6 of the GWO algorithm, and $D_{\boldsymbol{\rho}}$ is set to include all constraints imposed to the elements of $\boldsymbol{\rho}$.

Step C: The GWO algorithm is applied to solve the OP defined in (7) that gives the optimal value of the parameter vector $\boldsymbol{\rho}^*$ and the optimal parameters β^*, ${B_e}^*$, and η^*

$$\boldsymbol{\rho}^* = [\,\beta^*\ {B_e}^*\ \eta^*\,] = [\,{\rho_1}^*\ {\rho_2}^*\ {\rho_3}^*\,]^T. \quad (34)$$

Step D: The optimal value of the parameter $B_{\Delta e}$ results from the particular form of (13)

$${B_{\Delta e}}^* = 2T_s {B_e}^*/(2\beta^* T_{\Sigma 0} - T_s). \quad (35)$$

Step E: The transfer function of the set-point filter is obtained from the particular form of (9)

$$F(s) = 1/(1 + \beta^* T_{\Sigma 0} s). \quad (36)$$

IV. Validation and Results

The tuning approach proposed and given in the previous section is validated as follows by the design and tuning of a TSK PI-FC for the angular position of the experimental setup [23], [25], built around the Inteco dc servo system laboratory equipment [49]. The nominal values of the parameters of the process models given in (1) and (2) have been obtained by least squares identification algorithms as $u_a = 0.15$, $u_b = 1$, $u_c = 0.15$, $k_{P0} = k_{\mathrm{EP}0} = 140$, and $T_{\Sigma 0} = 0.92$ s. Since a reduced process small time constant sensitivity is targeted, T_Σ is variable, therefore $\alpha_1 = T_\Sigma$ in (6).

The steps A to E of the tuning approach have been applied. Step A starts with setting the sampling period to $T_s = 0.01$ s such that to have and fulfill the conditions of quasi-continuous digital control.

The weighting parameter in (7) has been set in step B in order to get a ratio of $\{0.1, 1, 10\}$, of the initial values of the two terms in the sum in (7). This has resulted in the weighting parameter values $\gamma^2 \in \{0.17187,\ 1.7187,\ 17.187\}$. The feasible domain of $\boldsymbol{\rho}$ used as search space has been set to

$$D_{\boldsymbol{\rho}} = \{3 \le \beta \le 17\} \times \{20 \le B_e \le 40\} \times \{0.55 \le \eta \le 0.75\}. \quad (37)$$

The experiment-based evaluation of the o.f. $J(\boldsymbol{\rho})$ is carried out in step C on a time horizon of 20 s in terms of the dynamic regimes of the CS with respect to the step-type modification of the reference input r to actually measure the values of $J(\boldsymbol{\rho})$. A part of the results obtained for the $r = 40rad$ step-type modification of r will be exemplified as follows.

The parameters of the GWO algorithm implemented in step C have been set to achieve a good tradeoff to convergence and use of allocated resources (agents, iterations) as $N = 20$ and $k_{\max} = 100$. The parameter ε_y in (28) has been set such that to fulfill the condition (31), namely $\varepsilon_y = 1.001$ for zero initial conditions.

The performance of the fuzzy CS tuned by means of our GWO algorithm has been compared with two other NIOAs, i.e., PSO and GSA, also applied in [23] and [25], and used here in step C of the tuning approach instead of the GWO algorithm. Since these NIOAs include random parameters, the top five obtained values for each weighting parameter values have been considered, and the information extracted from the real-time experiments will be presented as follows in terms of averaged measured values.

For a fair comparison of the NIOAs, both PSO and GSA have been implemented to work with the parameters $N = 20$ and $k_{\max} = 100$. In addition, the other parameters of PSO have been set to weighting parameters values $c_1 = 0.3$ and $c_2 = 0.9$, and a linear decrease of the inertia weight parameter within the domain determined by $w_{\max} = 0.9$ and $w_{\min} = 0.4$. The other parameters of GSA have been set to an exponential decrease law of the gravitational constant with the initial value $g(0) = 100$, the exponential parameter $\zeta = 8.5$, and the denominator parameter in the expression of the force $\varepsilon = 10^{-4}$. These parameter values lead to a good compromise to algorithms' exploration and exploitation capabilities.

The optimal values of the controller tuning parameters obtained with our GWO algorithm are synthesized in Table I along with similar values obtained by PSO and GSA. Table I shows that the GWO algorithm leads to similar optimal values of the o.f. compared with the results of the other two NIOAs for several values of the weighting parameter γ, and this demonstrates that the GWO algorithm is a viable alternative to solve (7). As in the case of other metaheuristics, GWO gives close to optimal values, which can be accepted as solution for the fuzzy CSs with a reduced parametric sensitivity.

Two performance indices have been measured to compare the performance of the three NIOAs. The first performance index is the convergence speed c_s, defined as the number of evaluations

TABLE I
WEIGHTING PARAMETER AND CONTROLLER PARAMETERS FOR THE NIOA-BASED MINIMIZATION OF J

	γ^2	$B_e{}^*$	$B_{\Delta e}{}^*$	η^*	β^*	k_c^*	T_i^*	J
GWO	0	40	0.138621	0.75	3.14193	0.0043801	2.89057	390460
	0.17187	40	0.145191	0.668808	3	0.0044825	2.76	628132
	1.7187	40	0.140516	0.75	3.09962	0.0044099	2.85165	2861740
	17.187	20.3574	0.013021	1	17	0.0018831	15.64	22794100
PSO	0	40	0.145191	0.75	3	0.004483	2.76	392076
	0.17187	40	0.145191	0.75	3	0.004483	2.76	641826
	1.7187	39.8689	0.144715	0.75	3	0.004483	2.76	1007420
	17.187	20	0.012792	0.25	17	0.001883	15.64	22809200
GSA	0	40	0.138541	0.75	3.14374	0.004379	2.8922	390459
	0.17187	39.3697	0.142008	0.75	3.01888	0.004469	2.7774	618429
	1.7187	36.4119	0.129304	0.75	3.0663	0.004434	2.821	2861380
	17.187	20	0.01281	0.287	16.9763	0.001884	15.618	22794600

TABLE II
WEIGHTING PARAMETER, INDICES c_s AND a_r FOR THREE NIOAS

	γ^2	GWO	PSO	GSA
c_s	0	1155	1952	1538
	0.17187	1138	1880	1417
	1.7187	1180	1321	1051
	17.187	1087	937	1491
a_r	0	0.8763	0.2231	0.1769
	0.17187	0.9159	0.0076	0.8221
	1.7187	0.1241	0.3292	0.1727
	17.187	0.2884	0.0069	0.1306

of $J(\boldsymbol{\rho})$ until getting $\boldsymbol{\rho}^*$. Table II highlights the values of c_s obtained by experiments for the three NIOAs.

The average value of the o.f. $J(\boldsymbol{\rho})$ obtained by running a certain NIOA, with the notation $\mathrm{Avg}(J_{\min})$, is computed as

$$\mathrm{Avg}(\mathrm{J_{min}}) = (1/\mathrm{N_{best}}) \sum_{\mathrm{j}=1}^{\mathrm{N_{best}}} \mathrm{J_{min}^{(j)}} \tag{38}$$

where $J_{\min}$ is the value of $J(\boldsymbol{\rho})$ obtained by running a certain NIOA, N_{best} is the number of best values (i.e., the smallest values) obtained for $J(\boldsymbol{\rho})$, and the superscript $j, j = 1, ..., N_{\text{best}}$ indicates the value of $J(\boldsymbol{\rho})$ obtained by one of the best N_{best} runs; therefore, ${J_{\min}}^{(j)}$ is the value of $J(\boldsymbol{\rho})$ obtained by the run $j, j = 1, ..., N_{\text{best}}$, of a certain NIOA. The value $N_{\text{best}} = 5$ has been set in this paper.

The second performance index is the accuracy rate a_r, which is the percent standard deviation of $J(\boldsymbol{\rho})$ obtained by running a certain NIOA and divided to $\mathrm{Avg}(\mathrm{J_{min}})$

$$\begin{aligned} a_r &= \mathrm{StDev}^{\%}(J_{\min}) = 100\mathrm{StDev}(J_{\min})/\mathrm{Avg}(J_{\min}) \\ \mathrm{StDev}(J_{\min}) &= \sqrt{\frac{1}{N_{\text{best}}-1} \sum_{j=1}^{N_{\text{best}}} ({J_{\min}}^{(j)} - \mathrm{Avg}(J_{\min}))^2}. \end{aligned} \tag{39}$$

Another set of experimental results is summarized in Table II as the values of the performance index a_r. The results given in Table II show that no algorithm has a clear advantage as the values of the two performance indices are rather close. However, GWO proves to be the overall best one from the point of view of c_s, and PSO proves to be the overall best one as far as a_r is concerned.

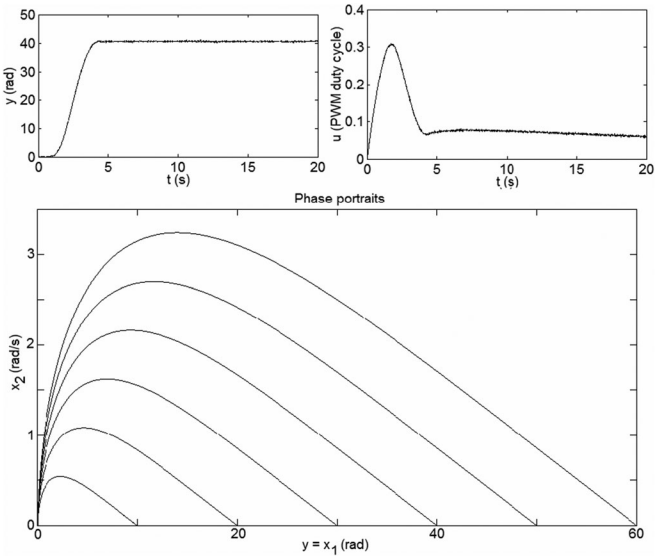

Fig. 3. Real-time experimental results of the fuzzy control system for the nominal process parameters: angular position and control signal versus time, and phase portraits (for reference input of 10, 20, 30, 40, 50, and 60 rad).

Fig. 3 offers a sample of experimental results for the fuzzy CS with the TSK PI-FC parameters given in the second row of Table I considering the nominal process, i.e., these parameters have been obtained by the GWO-based tuning for $\gamma^2 = 0.17187$. The average o.f. measured for the fuzzy CS is $J^{(1)} = 629462.33$.

The sensitivity reduction and the robustness ensured by the proposed fuzzy CS have been tested by several experiments conducted for different disturbed values (i.e., parametric disturbances) of the process parameter T_Σ using the $r = 40$ rad, step-type modification of r. For example, such a test has considered less than 10% increase of T_Σ, namely from $T_{\Sigma 0} = 0.92$ s to $T_\Sigma = 1s$. The responses are similar to those presented in Fig. 3, and the average measured o.f. is $J^{(2)} = 629578.67$. Using other TSK-PI-FC parameters, namely those given in the fourth row of Table I, which are different to those optimally tuned for $\gamma^2 = 0.17187$, the average measured o.f. for the nominal process is $J^{(3)} = 760873.58$ and the average measured o.f. for the disturbed process is $J^{(4)} = 8272554.21$. The ratios of the o.f.s are $J^{(4)}/J^{(3)} \cdot 100 = 108.76\%$ for the nonoptimal TSK-PI-FC and $J^{(2)}/J^{(1)} \cdot 100 = 100.02\%$ for the optimally tuned TSK-PI-FC, which clearly indicate the sensitivity reduction. The fuzzy CS responses are similar in these four cases, and this highlights the robustness of the proposed approach.

The results given in Fig. 3 illustrate the nonlinearity of the servo system and the friction in the bearings of the experimental setup as the proportional component of the controller is not capable of triggering an aggressive control signal. Consequently, the control signal varies in the vicinity of the dead zone. In addition, the behavior close to origin illustrates, as pointed out in relation with (30), that the closed-loop system is not globally asymptotically stable.

V. Conclusion

This paper proposed a GWO-based approach to the optimal tuning of fuzzy controllers with a reduced process small time constant sensitivity. The simplicity and robustness of our GWO algorithm made it a seamless candidate for solving the OPs comprised in the tuning approach.

Reducing the number of input variables of the o.f.s will lead to an improved cost-effective design and implementation of fuzzy controllers, with beneficial effects in various servo system and mechatronics applications [50]–[57].

Future research will deal with the adaptation of the proposed GWO algorithm formulation and fuzzy controller tuning approach to other OPs, including the optimal control of industrial applications.

References

[1] T. Vollmer, M. Manic, and O. Linda, "Autonomic intelligent cyber-sensor to support industrial control network awareness," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1647–1658, May 2014.

[2] D. Wijayasekara, O. Linda, M. Manic, and C. Rieger, "Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1829–1840, Aug. 2014.

[3] E. Tóth-Laufer, M. Takács, and I. J. Rudas, "Fuzzy logic-based risk assessment framework to evaluate physiological parameters," *Acta Polytech. Hungarica*, vol. 12, no. 2, pp. 159–178, Mar. 2015.

[4] Y. Sun, J. Yu, Z. Chen, and X. Xing, "Event-triggered filtering for nonlinear networked discrete-time systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 11, pp. 7163–7170, Nov. 2015.

[5] M. Qasim and V. Khadkikar, "Application of artificial neural networks for shunt active power filter control," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1765–1774, Aug. 2014.

[6] B. M. Wilamowski, C. Cecati, J. Kolbusz, P. Rozycki, and P. Siano, "A novel RBF training algorithm for short-term electric load forecasting and comparative studies," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6519–6529, Oct. 2015.

[7] L. Cheng, W. Liu, Z.-G. Hou, J. Yu, and M. Tan, "Neural network based nonlinear model predictive control for piezoelectric actuators," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7717–7727, Dec. 2015.

[8] J. Velagić, D. Delimustafić, and D. Osmanković, "Mobile robot navigation system based on probabilistic road map (PRM) with halton sampling of configuration space," in *Proc. 23rd IEEE Int. Symp. Ind. Electron.*, Istanbul, Turkey, 2014, pp. 1227–1232.

[9] Y. Jiang *et al.*, "System reliability calculation based on the run-time analysis of ladder program," *IEEE Trans. Ind. Electron.*, 2014, to be published, doi: 10.1109/ TIE.2014.2316222.

[10] X. Shen *et al.*, "OppCode: Correlated opportunistic coding for energy-efficient flooding in wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1631–1642, Dec. 2015.

[11] Z.-W. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[12] Y. Deng, Y.-Y. Kong, F. Bao, and Q.-H. Dai, "Sparse coding-inspired optimal trading system for HFT industry," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 467–475, Apr. 2015.

[13] G. Han, H. Zeng, M. Di Natale, X. Liu, and W.-H. Dou, "Experimental evaluation and selection of data consistency mechanisms for hard real-time applications on multicore platforms," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 903–918, May 2014.

[14] H. Gao, S. Kwong, B.-J. Fan, and R. Wang, "A hybrid particle-swarm tabu search algorithm for solving job shop scheduling problems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2044–2054, Nov. 2014.

[15] J. Seshadrinath, B. Singh, and B. K. Panigrahi, "Vibration analysis based interturn fault diagnosis in induction machines," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 340–350, Feb. 2014.

[16] A. Soualhi, H. Razik, G. Clerc, and D. D. Dinh, "Prognosis of bearing failures using hidden markov models and the adaptive neuro-fuzzy inference system," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2864–2874, Jun. 2014.

[17] H.-Y. Chung, C.-C. Hou, and Y.-S. Chen, "Indoor intelligent mobile robot localization using fuzzy compensation and Kalman filter to fuse the data of gyroscope and magnetometer," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6436–6447, Oct. 2015.

[18] O. Castillo, R., Martinez-Marroquin, P. Melin, P. F. Valdez, and J. Soria, "Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot," *Inf. Sci.*, vol. 192, pp. 19–38, Jun. 2012.

[19] O. Castillo and P. Melin, "A review on interval type-2 fuzzy logic applications in intelligent control," *Inf. Sci.*, vol. 279, pp. 615–631, Sep. 2014.

[20] R.-E. Precup, P. Angelov, B. S. J. Costa, and M. Sayed-Mouchaweh, "An overview on fault diagnosis and nature-inspired optimal control of industrial process applications," *Comput. Ind.*, vol. 74, pp. 75–94, Dec. 2015.

[21] S. Bouallègue, J. Haggège, M. Ayadi, and M. Benrejeb, "PID-type fuzzy logic controller tuning based on particle swarm optimization," *Eng. Appl. Artif. Intell.*, vol. 25, no. 3, pp. 484–493, Apr. 2012.

[22] R. K. Sahu, S. Panda, and G. T. Chandra Sekhar, "A novel hybrid PSO-PS optimized fuzzy PI controller for AGC in multi area interconnected power systems," *Int. J. Elect. Power Energy Syst.*, vol. 64, pp. 880–893, Jan. 2015.

[23] R.-C. David, R.-E. Precup, E. M. Petriu, M.-B. Radac, and S. Preitl, "Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity," *Inf. Sci.*, vol. 247, pp. 154–173, Oct. 2013.

[24] H. N. Azadani and R. Torkzadeh, "Design of GA optimized fuzzy logic-based PID controller for the two area non-reheat thermal power system," in *Proc. 13th Iranian. Conf. Fuzzy Syst.*, Qazvin, Iran, 2013, pp. 1–6.

[25] R.-E. Precup, R.-C. David, E M. Petriu, S. Preitl, and M.-B. Radac, "Novel adaptive charged system search algorithm for optimal tuning of fuzzy controllers," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1168–1175, Mar. 2014.

[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[27] B. Mahdad and K. Srairi, "Blackout risk prevention in a smart grid based flexible optimal strategy using grey wolf-pattern search algorithms," *Energy Convers. Manage.*, vol. 98, pp. 411–429, Jul. 2015.

[28] S. Mirjalili, "How effective is the grey wolf optimizer in training multi-layer perceptrons," *Appl. Intell.*, vol. 43, no. 1, pp. 150–161, Jul. 2015.

[29] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, "Using the grey wolf optimizer for solving optimal reactive power dispatch problem," *Appl. Soft Comput.*, vol. 32, pp. 286–292, Jul. 2015.

[30] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1257–1263, Jul. 2015.

[31] S. A. Medjahed, T. A. Saadi, A. Benyettou, and M. Ouali, "Gray wolf optimizer for hyperspectral band selection," *Appl. Soft Comput.*, vol. 40, pp. 178–186, Mar. 2016.

[32] A. Noshadi, J. Shi, W. S. Lee, P. Shi, and A. Kalam, "Optimal PID-type fuzzy logic controller for a multi-input multi-output active magnetic bearing system," *Neural Comput. Appl.*, vol. 27, no. 7, pp. 2031–2046, Oct. 2016.

[33] X. Yu, M. Ö. Efe, and O Kaynak, "A general backpropagation algorithm for feedforward neural networks learning," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 251–254, Jan. 2002.

[34] J. Vaščák, "Adaptation of fuzzy cognitive maps by migration algorithms," *Kybernetes*, vol. 41, nos. 3/4, pp. 429–443, Mar. 2012.

[35] E. Sariyildiz and K. Ohnishi, "Stability and robustness of disturbance-observer-based motion control systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 414–422, Jan. 2015.

[36] J. Qiu, S. X. Ding, H. Gao, and S. Yin, "Fuzzy-model-based reliable static output feedback H∞ control of nonlinear hyperbolic PDE systems," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 2, pp. 388–400, Jul. 2015.

[37] E. Osaba, E. Onieva, F. Dia, R. Carballedo, P. Lopez, and A. Perallos, "A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems," *Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 46–56, Oct. 2015.

[38] A. Gajate, R. E. Haber, P. I. Vega, and J. R. Alique, "A transductive neuro-fuzzy controller: Application to a drilling process," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1158–1167, Jul. 2010.

[39] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, and M.-B. Radac, "Gravitational search algorithms in fuzzy control systems tuning," in *Proc. 18th Int. Fed. Automat. Control World Congr.*, Milan, Italy, 2011, pp. 13624–13629.

[40] M. Bošnak, D. Matko, and S. Blažič, "Quadrocopter hovering using position-estimation information from inertial sensors and a high-delay video system," *J. Intell. Robot. Syst.*, vol. 67, no. 1, pp. 43–60, Jul. 2012.

[41] P. Angelov and R. Yager, "Density-based averaging—A new operator for data fusion," *Inf. Sci.*, vol. 222, pp. 163–174, Feb. 2013.
[42] H. Li, J. Wang, and P. Shi, "Output-feedback based sliding mode control for fuzzy systems with actuator saturation," *IEEE Trans. Fuzzy Syst.*, to be published, doi: 10.1109/TFUZZ.2015.2513085.
[43] S. Preitl and R.-E. Precup, "On the algorithmic design of a class of control systems based on providing the symmetry of open-loop bode plots," *Sci. Bull. UPT, Trans. Automat. Control Comput. Sci.*, vol. 41(55), no. 2, pp. 47–55, Dec. 1996.
[44] S. Preitl and R.-E. Precup, "An extension of tuning relations after symmetrical optimum method for PI and PID controllers," *Automatica*, vol. 35, no. 10, pp. 1731–1736, Oct. 1999.
[45] S. Galichet and L. Foulloy, "Fuzzy controllers: Synthesis and equivalences," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 140–148, May 1995.
[46] R.-E. Precup and S. Preitl, "Popov-type stability analysis method for fuzzy control systems," in *Proc. 5th Eur. Congr. Intell. Technol. Soft Comput.*, Aachen, Germany, 1997, vol. 2, pp. 1306–1310.
[47] J. Kluska, "Absolute stability of continuous fuzzy control systems," in *Stability Issues in Fuzzy Control*, vol. 44, J. Aracil and F. Gordillo F., Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 15–45.
[48] R.-E. Precup, R.-C. David, E. M. Petriu, A.-I. Szedlak-Stinean, and C.-A. Bojan-Dragos, "Grey wolf optimizer-based approach to the tuning of PI-fuzzy controllers with a reduced process parametric sensitivity," in *Proc. 4th IFAC Int. Conf. Intell. Control Autom. Sci.*, Reims, France, 2016, pp. 55–60.
[49] Modular Servo System, User's Manual, Inteco Ltd., Krakow, Poland, 2007.
[50] M. Ö. Efe, C. Unsal, O. Kaynak, and X. Yu, "Variable structure control of a class of uncertain systems," *Automatica*, vol. 40, no. 1, pp. 59–64, Jan. 2004.
[51] F. G. Filip, "Decision support and control for large-scale complex systems," *Annu. Rev. Control*, vol. 32, no. 1, pp. 61–70, Apr. 2008.
[52] S. Formentin, A. Karimi, and S. M. Savaresi, "Optimal input design for direct data-driven tuning of model-reference controllers," *Automatica*, vol. 49, no. 6, pp. 1874–1882, Jun. 2013.
[53] R. C. Luo and C. C. Lai, "Multisensor fusion-based concurrent environment mapping and moving object detection for intelligent service robotics," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4043–4051, Aug. 2014.
[54] D. Yazdani, S. Sadeghi-Ivrigh, D. Yazdani, A. Sepas-Moghaddam, and M. R. Meybodi, "Fish swarm search algorithm: A new algorithm for global optimization," *Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 17–45, Oct. 2015.
[55] W. Qiao, P. J. Zhang, and M.-Y. Chow, "Condition monitoring, diagnosis, prognosis, and health management for wind energy conversion systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6533–6535, Oct. 2015.
[56] H.-Y. Xu and R. Vilanova, "PI and fuzzy control for P-removal in wastewater treatment plant," *Int. J. Comput. Commun. Control*, vol. 10, no. 6, pp. 936–951, Dec. 2015.
[57] F. Bechet, K. Ogawa, E. Sariyildiz, and K. Ohnishi, "Electrohydraulic transmission system for minimally invasive robotics," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7643–7654, Dec. 2015.

Radu-Emil Precup (M'03–SM'07) received the Dipl.Ing. (with honors) degree in automation and computers from the "Traian Vuia" Polytechnic Institute of Timisoara, Timisoara, Romania, the Dipl. degree in mathematics from the West University of Timisoara, Timisoara, and the Ph.D. degree in automatic systems from the Politehnica University of Timisoara (UPT), Timisoara, in 1987, 1993, and 1996, respectively.

He is currently with UPT, where he became a Professor in the Department of Automation and Applied Informatics in 2000. He is also an Adjunct Professor in the School of Engineering, Edith Cowan University, Joondalup, Australia, and an Honorary Professor at Óbuda University, Budapest, Hungary. He is the author or coauthor of more than 300 papers. His current research interests include intelligent control systems and data-driven control.

Prof. Precup is a Member of the Subcommittee on Computational Intelligence of the IEEE Industrial Electronics Society.

Radu-Codrut David received the Dipl.Ing. degree in systems and computer engineering, the M.Sc. degree in automatic systems, and the Ph.D. degree in systems engineering from the Politehnica University of Timisoara (UPT), Timisoara, Romania, in 2006, 2008, and 2015, respectively.

He is currently a Postdoctoral Researcher with UPT. He is the coauthor of more than 30 papers published in scientific journals, refereed conference proceedings, and as contributions to books. His research interests include control structures and algorithms with focus on fuzzy control and nature-inspired optimization.

Dr. David is a Member of the Technical Committee on Soft Computing of the IEEE Systems, Man, and Cybernetics Society.

Emil M. Petriu (M'86–SM'88–F'01) received the Dipl.Eng. and Dr. Eng. degrees in electrical engineering from the Polytechnic Institute of Timisoara, Timisoara, Romania, in 1969 and 1978, respectively.

He is a University Research Chair Professor in the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. His research interests include multisensor systems, soft computing, biology-inspired robot sensing, and human–computer symbiosis.

Prof. Petriu is a Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He received the 2003 IEEE Donald G. Fink Prize Paper Award.

Information Sciences 381 (2017) 176–192

Contents lists available at ScienceDirect

Information Sciences

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

Model-free sliding mode control of nonlinear systems: Algorithms and experiments

Radu-Emil Precup[a,b,*], Mircea-Bogdan Radac[a], Raul-Cristian Roman[a], Emil M. Petriu[c]

[a] Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania
[b] School of Engineering, Edith Cowan University, 270 Joondalup Dr., Joondalup, WA 6027, Australia
[c] School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward, Ottawa, ON K1N 6N5, Canada

ARTICLE INFO

Article history:
Received 4 March 2016
Revised 19 October 2016
Accepted 26 November 2016
Available online 28 November 2016

Keywords:
Model-free iPI control system structure
Model-free sliding mode control system structure
Twin rotor aerodynamic system
Real-time experiments

ABSTRACT

This paper proposes two model-free sliding mode control system (MFSMCS) structures. The new structures are compared with a model-free intelligent proportional-integral (iPI) control system structure. Two simple design approaches for the MFSMCS structures are suggested. The control system structures and the design approaches are validated by a set of real-time experimental results on a nonlinear laboratory twin rotor aerodynamic system (TRAS). The MFSMCS structures are considered in the framework of a Multi Input-Multi Output TRAS control system, where the azimuth and pitch positions are controlled using separate Single Input-Single Output control system structures for each control channel (azimuth and pitch). The experimental validation is carried out by two scenarios that illustrate and allow the assessment of the MFSMCS structures performance and the comparison versus a model-free iPI control system structure as well.

1. Introduction

The Model-Free Control (MFC) technique [13,14,15], which is also referred to in the literature as model-free tuning, is a data-driven technique that uses a local linear approximation of the process model, which is valid for a small time window and a fast estimator is employed to update this approximation. The main advantages of MFC are: it does not require the process model in the controller tuning (this is specific to data-driven control techniques) and few experiments are conducted on the real-world control system structure in the tuning process. The MFC technique is implemented as MFC algorithms that usually contain proportional (P), proportional-integral (PI), proportional-derivative (PD), or proportional-integral-derivative (PID) controllers and additional terms to compensate for the mismatch of the online estimation, and altogether form the so called intelligent P/PI/PD/PID (iP/iPI/iPD/iPID) controllers. A shortcoming of the initial MFC algorithm [13] is that it has not been formulated with a set of reset conditions that guarantee stability, but several conditions have been proposed in [52,57,65] to mitigate this shortcoming.

The MFC algorithms are successfully applied to many processes and a short analysis of the state-of-the-art is presented as follows. The problems and challenges of MFC in the framework of data-driven control are analyzed versus model-based

* Corresponding author at: Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania. Fax: +40 256403214.
E-mail addresses: radu.precup@aut.upt.ro, radu.precup@upt.ro (R.-E. Precup), mircea.radac@upt.ro (M.-B. Radac), raul-cristian.roman@student.upt.ro (R.-C. Roman), petriu@uottawa.ca (E.M. Petriu).

http://dx.doi.org/10.1016/j.ins.2016.11.026
0020-0255/

control in [18]. A comparison of an MFC algorithm without stability proof and a classical PID controller is performed in [17] and validated experimentally on a shape memory alloy spring-based actuator. An event-driven MFC is applied to a simulated model of a quadrotor in [66], compared with a backstepping control and with a sliding mode control, and the stability of the control systems is proven. An MFC algorithm is designed in [19] to optimally control the oscillations of a single body heaving energy converter, and the stability is ensured through a conventional lead compensator. An MFC algorithm is suggested in [22] to control an experimental greenhouse with an application to a fault accommodation, and the results are compared with Boolean controllers. An MFC algorithm is analyzed in [4] and extended using time-varying parameters. An MFC algorithm with guaranteed stability is discussed in [52] and compared with a model-free adaptive control algorithm, both data-driven techniques are experimental validated on a twin rotor aerodynamic system (TRAS). An MFC algorithm is optimally tuned in [57] using a linear quadratic regulator in order to guarantee the stability. Other attractive MFC applications validated by simulation results are reported in [21], where an MFC algorithm without stability proof is applied to the water level control behavior of hydroelectric power plants, in [29], where an MFC algorithm without stability proven is applied to an instrumented car, and in [54], where an MFC algorithm regulates the glycemia in type-1 diabetes.

Sliding mode control is a relatively easily understandable nonlinear control technique with the advantage of robustness against parameter variations and disturbances. Some recent sliding mode control structures and applications are next discussed. The robust control using sliding mode is conducted in [48] targeting a class of under-actuated systems with mismatched uncertainties. A second-order sliding mode observer for a pendulum system with Coulomb friction is proposed in [9]. Sliding mode control and PD controllers for a permanent magnet synchronous motor are given in [25]. An adaptive seeding sliding mode control is suggested in [69] and applied to design adaptive cruise system for off-road vehicles. Sliding mode controllers that stabilize TRAS are discussed in [7,59]. The sliding delayed sliding mode control for nonlinear systems is suggested in [10]. A switched/time-based adaptation strategy for second-order sliding mode control algorithms is proposed in [35] and tested on a single-link manipulator with flexible joint and negligible damping. The adaptive sliding mode control problem of a nonlinear Markovian jump systems is treated in [24]. The input–output finite-time stabilization for a class of nonlinear system like Chua circuits via sliding mode control is investigated in [60]. The integral sliding mode control for stochastic Markovian jump systems with time-varying delay is proposed in [27]. A sliding mode fuzzy controller is designed in [5] in order to satisfy performance constrains imposed to continuous-time Takagi–Sugeno fuzzy models with a bilinear consequent part. Attractive discrete-time sliding mode control applications are reported in [26,30,32,64].

The combination of the advantages of data-driven control and sliding mode control has resulted in the merge of data-driven and sliding mode control techniques. Some mixed data-driven and sliding mode control techniques are next discussed. Sliding mode control is combined with a model-free iPI controller in [45] and applied to real-time servo system equipment; the real-time experimental results are compared with a model-free iPI controller. A mixed sliding mode control-model-free iPD controller is proposed in [65] and applied to a quadrotor system; the simulation results are compared with a PD controller and a model-free iPD controller. Sliding mode control mixed with model-free adaptive control is suggested in [67] and tested in the simulated behavior of a robotic exoskeleton. An iterative learning control scheme is designed via continuous sliding mode control in [6] and validated experimentally on a rotary plant. Sliding mode control is designed by adaptive dynamic programming in [11] and tested in a simulated class of partially unknown systems with input disturbances. An adaptive sliding mode inverse control strategy of continuous-time nonlinear dynamic systems is analyzed and digitally simulated in [68].

The application considered in this paper is a TRAS, which is a laboratory equipment designed for control system experiments. The TRAS is challenging as in some aspects it behaves like a real helicopter. From the point of view of the control, the TRAS system presents a mathematical model with strong interconnected nonlinearities. Several data-driven techniques have been applied recently to TRAS, and the most relevant ones are emphasized as follows. The real-time implementation of neuron-adaptive observers is proposed in [36]. An MFC algorithm is compared in [52] with model-free adaptive controllers, and both techniques are validated by experiments. MFC is optimized in [57] in a linear quadratic regulator framework. Two versions of model-free adaptive control algorithms (the compact form dynamic linearization and the partial form dynamic linearization) are analyzed and validated experimentally in [55]. Two implementations of model-free adaptive control are suggested in [56]; the experimental results are compared with a data-driven virtual reference feedback tuning technique. The model-free adaptive control is combined with virtual reference feedback tuning to automatically obtain the controller parameters in [58] and with iterative feedback tuning in [50,51]. The data-driven algorithms dedicated to TRAS are compared in [52,55,56,57,58] using a sum-type performance index.

Building upon our recent model-free sliding mode control system (MFSMCS) structure and design approach [45] applied to a nonlinear servo system equipment and validated experimentally and next to a quadrotor simulated model [65], this paper proposes two MFSMCS structures and design approaches. Both structures and design approaches are based on Lyapunov's stability theory. The first structure includes a simplified filter compared to [45], namely a derivative plus low-pass filter to estimate the first-order derivative of the controlled output. The estimated derivative is included in a first-order nonlinear dynamic system as a local approximation of the process model and next introduced in the control laws. The first structure and design approach make also use of the equivalent control method specific to sliding mode control. Both approaches are applied to TRAS and validated by real-time experimental results.

This paper suggests four new contributions with respect to the state-of-the-art:

- Two novel MFSMCS structures and their design approaches are offered.

- Two sliding mode reaching and existence conditions are formulated in order to guarantee the control system stability.
- The MFSMCS structures are compared systematically with an MFC structure built around an iPI controller by measuring the real-time control system performance in terms of sum-type performance indices. The parameters of the three model-free controllers are optimally tuned by the minimization of these performance indices viewed as objective functions.
- The MFSMCS structures are validated by experiments on TRAS, i.e., on a process that exhibits stronger nonlinearities than those usually used to test MFC.

Our MFSMCS structures are developed on the basis of an iPI controller for which the systematic design does not guarantee that the tracking error converges asymptotically to zero. As shown in [14], the stability of the tracking error involves the asymptotic convergence to zero only if the system is in a steady-state regime. That is the reason why this paper suggests an augmented control signal that is inserted in the control system structure and next designed by sliding mode control. Our design approaches are model-free in tuning because, as mentioned in [18,45], data-driven control is characterized by the controller designed in terms of directly using on-line or off-line input–output data of the controlled system or knowledge from the data processing but not any explicit information from the process model. The data-driven approaches given in this paper actually employ the sliding mode control of the model represented by the tracking error dynamics obtained by MFC.

The new contributions of this paper are important and advantageous with respect to the current literature in the field analyzed above because of the following reasons:

- Simple design approaches are given.
- The stability of the tracking error dynamics specific to MFC is guaranteed by sliding mode control that also influences the behavior of the tracking error dynamics.
- The control system structures resulted after design and tuning are robust against parameter variations and disturbances as a benefit of sliding mode control.

This paper is structured as follows: a short overview on model-free iPI control is given in the next section. Section 3 focuses on the MFCMCS structures and design approaches. Section 4 considers the case study of the design of model-free controllers for the position control of the TRAS. Experimental results are included in order to validate the new MFSMCS structures and to carry out the comparison with MFC. The conclusions are highlighted in Section 5.

2. Model-free iPI control

As shown in [13,15], the general form of the first order local process model subsequently used in the design of the iPI controller is

$$\dot{y}(t) = F(t) + \alpha u(t), \tag{1}$$

where t is the continuous variable, $F(t)$ is a function that incorporates the effects of unmodeled dynamics and disturbances, this function is estimated using the information from the control signal $u(t)$ and the controlled output $y(t)$, and $\alpha > 0$ is a design parameter, which is chosen by the user such that $\dot{y}(t)$ and $\alpha u(t)$ should have the same order of magnitude. The tracking error $e(t)$ is

$$e(t) = y(t) - y^*(t), \tag{2}$$

where $y^*(t)$ is the desired reference trajectory, that describes the behavior imposed to the control system, the control law of the iPI controller is

$$u(t) = \frac{1}{\alpha}\left(-\hat{F}(t) + \dot{y}^*(t) - K_P e(t) - K_I \int_0^t e(\tau)d\tau\right), \tag{3}$$

where $\hat{F}(t)$ is the estimate of $F(t)$, K_P and K_I are the proportional and the integral gains of the PI controller with the following transfer function:

$$C(s) = K_P + \frac{K_I}{s}. \tag{4}$$

For the practical implementation of the derivates of the controlled output $y(t)$ in (1) and the desired reference trajectory $y^*(t)$ in (3), we propose to use a first order derivative plus low-pass filter with the transfer function

$$H_{Lp1}(s) = \frac{K_{Lp1}s}{1 + T_{Lp1}s}, \tag{5}$$

where K_{Lp1} is the filter gain and T_{Lp1} is the filter time constant. The filter parameters should be chosen as a compromise to noise reduction and the delay it induces. This filter generates the estimate of $\dot{y}(t)$, with the notation $\hat{\dot{y}}(t)$, which leads to the modified expression of (1)

$$\hat{F}(t) = \hat{\dot{y}}(t) - \alpha u(t). \tag{6}$$

The filter with the transfer function given in (5) also generates $\dot{y}^*(t)$ in (3), which represents the filtered derivative of the desired reference trajectory $y^*(t)$.

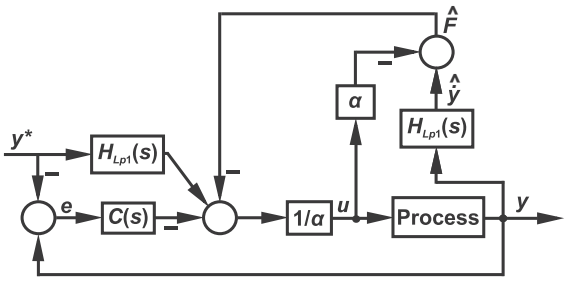

Fig. 1. Model-free control structure with iPI controller.

The structure of the MFC structure with an iPI controller is presented in Fig. 1, where the Lp1 block represents the first order derivative plus low-pass filter with the transfer function given in (5).

The estimation error of $F(t)$, namely $e_{est}(t)$ whose value is considered negligible is defined as

$$e_{est}(t) = \dot{y}(t) - \hat{\dot{y}}(t) = F(t) - \hat{F}(t). \tag{7}$$

Substituting (6) in (3), adding and subtracting the derivate of the controlled output $\dot{y}(t)$, the dynamics of the control system structure is described by

$$\hat{\dot{y}}(t) - \dot{y}(t) + \dot{y}(t) - \dot{y}^*(t) + K_P e(t) + K_I \int_0^t e(\tau)d\tau = 0. \tag{8}$$

Using (7) and (2) in (8), the control system structure given in Fig. 1 is characterized by the following tracking error dynamics:

$$\dot{e}(t) + K_P e(t) + K_I \int_0^t e(\tau)d\tau = e_{est}(t). \tag{9}$$

3. Model-free sliding mode control system structures and design approaches

Starting with the control law given in (3), which is in fact an iPI controller and adding an augmented control signal $u_{aug}(t)$, the control law of the MFSMCS structure is

$$u(t) = \frac{1}{\alpha}\left(-\hat{F}(t) + \dot{y}^*(t) - K_P e(t) - K_I \int_0^t e(\tau)d\tau\right) + u_{aug}(t), \tag{10}$$

and the closed-loop control system structure is described by

$$\dot{e}(t) + K_P e(t) + K_I \int_0^t e(\tau)d\tau = e_{est}(t) + \alpha u_{aug}(t). \tag{11}$$

For estimating the first-order derivative of the controlled output the MFSMCS structures uses a derivative plus low-pass filter with the transfer function $H_{Lp1}(s)$ expressed in (5).

To combine the MFC algorithm with sliding mode control, the state variables $x_1(t)$ and $x_2(t)$ are introduced in terms of

$$\begin{aligned} x_1(t) &= \int_0^t e(\tau)d\tau, \\ x_2(t) &= e(t), \end{aligned} \tag{12}$$

and the state-space equations of the closed-loop control system related to (11) are

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= -K_I x_1(t) - K_P x_2(t) + \alpha u_{aug}(t) + e_{est}(t), \end{aligned} \tag{13}$$

where $e_{est}(t)$ is the disturbance input.

The switching variable $\sigma(t)$ used in designing the sliding mode control law $u_{aug}(t)$ is

$$\sigma(t) = x_1(t) + T x_2(t), \tag{14}$$

where $T > 0$ is the design parameter that prescribes the desired behavior of the control system on the sliding mode surface. The Lyapunov function candidate introduced in order to guarantee the stability of the closed-loop system related to (11) is

$$V(t) = \frac{1}{2}\sigma^2(t), \tag{15}$$

and using the Lyapunov's stability theory the condition $\dot{V}(t) < 0$ is transformed into the sliding mode reaching and existence condition

$$\sigma(t)\dot{\sigma}(t) < 0, \tag{16}$$

which is employed to derive the control law $u_{aug}(t)$ as function of $x_1(t)$ and $x_2(t)$.

The variable $\dot{\sigma}(t)$ is the derivative of the switching variable, expressed as

$$\dot{\sigma}(t) = \dot{x}_1(t) + T\dot{x}_2(t). \tag{17}$$

Introducing (13) in (17), $\dot{\sigma}(t)$ becomes

$$\dot{\sigma}(t) = -K_I T x_1(t) + (1 - K_P T)x_2(t) + \alpha T u_{aug}(t) + T e_{est}(t). \tag{18}$$

Since the estimation error $e_{est}(t)$ is unknown, its range will be estimated. The value of $e_{est}(t)$ is assumed to be bounded

$$|e_{est}(t)| \le e_{est\,\max}, \tag{19}$$

where $e_{est\,\max}$ is the upper bound of $|e_{est}(t)|$, and its value is known. The parameter $e_{est\,\max}$ plays the role of a design parameter.

Two MFSMCS structures are considered as follows. The first MFSMCS structure was first proposed in [45] and applied to model-free iPI controllers but with a different filter (this time with a first order derivative plus low-pass filter) and next extended in [65] to model-free iPD controllers.

3.1. The first MFSMCS structure

The first MFSMCS structure and its design approach are described in this sub-section. The abbreviation MFSMCS1 will be used for this control system structure.

The augmented control signal $u_{aug}(t)$ consists of two control signals

$$u_{aug}(t) = u_{eq}(t) + u_{cor}(t), \tag{20}$$

where $u_{eq}(t)$ and $u_{cor}(t)$ are the equivalent control signal and correction control signal, respectively. The equivalent control signal $u_{eq}(t)$ is obtained from the ideal sliding mode condition $\sigma(t)=0$, which leads to

$$\dot{\sigma}(t) = 0. \tag{21}$$

Substituting (20) in (18), where $u_{aug}(t)=u_{eq}(t)$ and solving for $u_{eq}(t)$, the expression of $u_{eq}(t)$ becomes

$$u_{eq}(t) = \frac{K_I T x_1(t) - (1 - K_P T)x_2(t) - T e_{est}(t)}{\alpha T}. \tag{22}$$

Since $e_{est}(t)$ is unknown in (22), its value will be replaced with $e_{est\,\max}$, hence (22) is rewritten as

$$u_{eq}(t) = \frac{K_I T x_1(t) - (1 - K_P T)x_2(t) - T e_{est\,\max}}{\alpha T}. \tag{23}$$

To satisfy the sliding mode reaching and existence condition given in (16) and to alleviate the chattering effects a boundary layer approach is next applied. The proposed expression of the correction signal $u_{cor}(t)$ is

$$u_{cor}(t) = -\frac{\eta}{\alpha T}\mathrm{sat}(\sigma(t), \varepsilon) = -\frac{\eta}{\alpha T}\begin{cases} -1 & \text{if } \sigma(t) < -\varepsilon, \\ \dfrac{\sigma(t)}{\varepsilon} & \text{if } |\sigma(t)| \le \varepsilon, \\ 1 & \text{if } \sigma(t) > \varepsilon, \end{cases} \tag{24}$$

where $\eta > 0$ and $\varepsilon > 0$ are the convergence factor and the boundary layer thickness, respectively.

Substituting (23) and (24) in (20) and next $u_{aug}(t)$ in (18), the expression of $\dot{V}(t)$ becomes

$$\dot{V}(t) = \sigma(t)\dot{\sigma}(t) = -\sigma(t)\eta\,\mathrm{sat}(\sigma(t), \varepsilon) + \sigma(t)T[e_{est}(t) - e_{est\,\max}]. \tag{25}$$

Theorem 1. *Let the sliding mode control law defined in (10) for the process model specified in (1) with the disturbance estimator described in (6) be given together with the closed-loop control system dynamics presented in (13), the switching variable defined in (14) and the Lyapunov function candidate expressed in (15). Let us assume the bounded estimation error according (19). Then the MFSMCS1 structure is stable if*

$$u_{aug}(t) = \frac{K_I T x_1(t) - (1 - K_P T)x_2(t) - T e_{est\,\max}}{\alpha T} - \frac{\eta}{\alpha T}\mathrm{sat}(\sigma(t), \varepsilon). \tag{26}$$

Proof. Two cases that depend on the possible values of $\sigma(t)$ are next considered in order to ensure the fulfillment of the sliding mode reaching and existence condition (16).

Case 1. $|\sigma(t)| \leq \varepsilon$ (the state vector belongs to the boundary layer). Substituting (24) in (25), the sliding mode reaching and existence condition (16) becomes

$$\sigma(t)\dot{\sigma}(t) = -\frac{\sigma^2(t)\eta}{\varepsilon} + \sigma(t)T[e_{est}(t) - e_{est\,\max}]. \tag{27}$$

To ensure the negative right-hand term (27), the following condition is sufficient:

$$\frac{\sigma^2(t)\eta}{\varepsilon} > |\sigma(t)|T|e_{est}(t) - e_{est\,\max}|. \tag{28}$$

Since

$$|e_{est}(t) - e_{est\,\max}| \leq 2e_{est\,\max}, \tag{29}$$

a sufficient condition to fulfill (16) is

$$\frac{|\sigma(t)|\eta}{\varepsilon} > 2Te_{est\,\max}. \tag{30}$$

Case 2. $|\sigma(t)| > \varepsilon$ (the state vector is out of the boundary layer). Substituting (24) in (25), the sliding mode reaching and existence condition (16) becomes

$$\sigma(t)\dot{\sigma}(t) = -\sigma^2(t)\eta + \sigma(t)T[e_{est}(t) - e_{est\,\max}]. \tag{31}$$

To ensure the negative right-hand term in (31), the following condition is sufficient:

$$\sigma^2(t)\eta > |\sigma(t)|T|e_{est}(t) - e_{est\,\max}|. \tag{32}$$

Using (29), a sufficient condition to fulfill (16) is

$$|\sigma(t)|\eta > 2Te_{est\,\max}. \tag{33}$$

The switching variable dynamics in sliding mode is characterized by this expression derived from (25):

$$\dot{\sigma}(t) = -\eta\,\mathrm{sat}(\sigma(t), \varepsilon) + T[e_{est}(t) - e_{est\,\max}]. \tag{34}$$

The steady-state switching variable σ_∞ fulfills $\dot{\sigma}(t) = 0$ and its boundary layer is reached if

$$\mathrm{sat}(\sigma_\infty, \varepsilon) = \frac{\sigma_\infty}{\varepsilon}, \tag{35}$$

so the steady state expression of (34) is

$$0 = -\frac{\eta\sigma_\infty}{\varepsilon} + T[e_{est\,\infty} - e_{est\,\max}], \tag{36}$$

where $e_{est\,\infty}$ is the steady-state estimation error. Eq. (36) yields the steady-state switching variable

$$\sigma_\infty = -\frac{T\varepsilon[e_{est\infty} - e_{est\,\max}]}{\eta}. \tag{37}$$

Eq. (37) indicates, as (34), that large values of $\eta > 0$, small values of $\varepsilon > 0$ and accurate derivative estimators are recommended to reduce the undesirable chattering effects.

Substituting $u_{eq}(t)$ from (23) and $u_{cor}(t)$ from (24) in (20), the expression of $u_{aug}(t)$ results in terms of (26). Concluding, the two cases reflected in (30) and (33), the subsequent processing of the switching variable dynamics that leads to (24) and the steady-state estimation error given in (37). According to (26) it is guaranteed that $\dot{V}(t) < 0$, leading to a stable MFSMCS2 structure and thus concluding the proof.

Rewriting $u_{aug}(t)$ in terms of (12), the final expression of the control law specific to the MFSMCS1 structure is

$$u(t) = \frac{1}{\alpha}\left(-\hat{F}(t) + \dot{y}^*(t) - \frac{e(t)}{T} - e_{est\,\max} - \frac{\eta}{T}\mathrm{sat}(\sigma(t), \varepsilon)\right), \tag{38}$$

and it is illustrated in Fig. 2.

The design approach of the MFSMCS1 structure consists of the following steps:

Step 1.1. Set the design parameter $\alpha > 0$ such that $\dot{y}(t)$ and $\alpha u(t)$ should have the same order of magnitude.

Step 1.2. Choose the parameters of the first order derivative plus low-pass filter with the transfer function $H_{Lp1}(s)$ given in (5) such that to ensure a tradeoff to noise reduction and delay induced by the filter. This filter is also used to obtain accurate derivative estimates $\hat{y}(t)$ characterized by small estimation errors and derivatives smoothing.

Step 1.3. Estimate a small value for the design parameter $e_{est\max}$.

Step 1.4. Set the design parameter $T > 0$ to prescribe the desired behavior of the control system on the sliding manifold.

Step 1.5. Set the parameters $\eta > 0$ and $\varepsilon > 0$ using Eqs. (33) and (36).

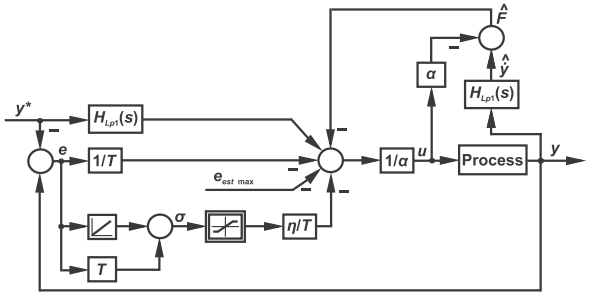

Fig. 2. MFSMCS1 structure.

3.2. The second MFSMCS structure

The second MFSMCS structure and its design approach are described in this sub-section. The abbreviation MFSMCS2 will be used for this control system structure.

Theorem 2. *Let the sliding mode control law defined in (10) for the process model specified in (1) with the disturbance estimator described in (6) be given together with the closed-loop control system dynamics presented in (13), the switching variable defined in (14) and the Lyapunov function candidate expressed in (15). Let us assume the bounded estimation error according (19). Then the MFSMCS2 structure is stable if*

$$u_{aug}(t) = -\left(\frac{\psi + |K_I T x_1(t) + (K_P T - 1)x_2(t)| + T e_{est\ \max}}{\alpha T} + \delta\right)\mathrm{sgn}(\sigma(t)), \tag{39}$$

where $\psi > 0$ and $\delta > 0$ are user-selectable design variables.

Proof. As in the first approach, two cases are considered as follows in order to ensure the fulfillment of the sliding mode reaching and existence condition (16).

Case 1. $\sigma(t) < 0$. The condition (16) yields $\dot{\sigma}(t) > 0$, but we impose the condition

$$\dot{\sigma}(t) > \psi, \tag{40}$$

where the design parameter $\psi > 0$ guarantees $\dot{V}(t) < 0$ if bounded uncertainties and unstructured system dynamics are accounted for. Using (18) in (20), the following condition for the augmented control signal $u_{aug}(t)$ is derived:

$$u_{aug}(t) > \frac{\psi + K_I T x_1(t) + (K_P T - 1)x_2(t) - T e_{est}(t)}{\alpha T}. \tag{41}$$

Using (19) in (41), a sufficient condition to fulfill (16) is

$$u_{aug}(t) > \frac{\psi + |K_I T x_1(t) + (K_P T - 1)x_2(t)| + T e_{est\ \max}}{\alpha T}. \tag{42}$$

Case 2. $\sigma(t) > 0$. The condition (16) yields $\dot{\sigma}(t) < 0$, but we impose the condition

$$\dot{\sigma}(t) < -\psi \tag{43}$$

to guarantee $\dot{V}(t) < 0$ if parametric disturbances occur and to ensure faster reaching phase. Using (18) in (43), the following condition for the augmented control signal $u_{aug}(t)$ is derived:

$$u_{aug}(t) < \frac{-\psi + K_I T x_1(t) + (K_P T - 1)x_2(t) - T e_{est}(t)}{\alpha T}. \tag{44}$$

Using next (19) in (44), a sufficient condition to fulfill (16) is

$$u_{aug}(t) < -\frac{\psi + |K_I T x_1(t) + (K_P T - 1)x_2(t)| + T e_{est\ \max}}{\alpha T}. \tag{45}$$

Concluding, the two cases reflected in (42) and (45) can be combined in the augmented control signal $u_{aug}(t)$ in (39) that guarantees the sliding mode reaching and existence condition (16), where $\delta > 0$ should be large enough to suppress all bounded uncertainties and unstructured system dynamics. According to (39) it is guaranteed that $\dot{V}(t) < 0$, leading to a stable MFSMCS2 structure and thus concluding the proof.

The dynamics of the switching variable in sliding mode is characterized by the following relationship derived from (18) and (45):

$$\begin{aligned}\dot{\sigma}(t) = {} & -K_I T x_1(t) + (1 - K_P T)x_2(t) - (\psi + |K_I T x_1(t) + (K_P T - 1)x_2(t)| \\ & + T e_{est\ \max} + \alpha T \delta)\mathrm{sgn}(\sigma(t)) + T e_{est}(t).\end{aligned} \tag{46}$$

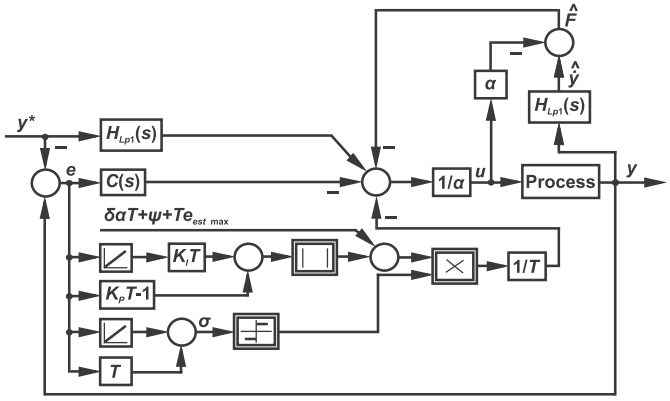

Fig. 3. MFSMCS2 structure.

The steady-state switching variable σ_∞ fulfills $\dot{\sigma}_\infty = 0$. Therefore, the steady-state expression of Eq. (46) is

$$K_I T x_1(t) + (K_P T - 1)x_2(t) + (\psi + |K_I T x_1(t) + (K_P T - 1)x_2(t)| + Te_{est\ \max} + \alpha T\delta)\mathrm{sgn}(\sigma(t)) = Te_{est\,\infty}, \tag{47}$$

where $e_{est\,\infty}$ is the steady-state estimation error. Eq. (47) points out that since $\sigma_\infty = 0$ is desired, large values of $T > 0$ and accurate derivative estimators are recommended.

The substitution of $u_{aug}(t)$ from (45) in (10) in terms of the notations (12) leads to the expression of the control law specific to the MFSMCS2 structure

$$u(t) = \frac{1}{\alpha}\left[-\hat{F}(t) + \dot{y}^*(t) - K_P e(t) - K_I \int_0^t e(\tau)d\tau - \frac{1}{T}\left(\delta\alpha T + \psi + Te_{est\ \max} + |K_I T \int_0^t e(\tau)d\tau + (K_P T - 1)e(t)|\right)\mathrm{sgn}(\sigma(t))\right], \tag{48}$$

and it is pointed out in Fig. 3.

The design approach of the MFSMCS2 structure consists of the following steps:

Step 2.1. Set the design parameter $\alpha > 0$ such that the terms $\dot{y}(t)$ and $\alpha\ u(t)$ have the same order of magnitude.

Step 2.2. Choose the parameters of the first order derivative plus low-pass filter with the transfer function $H_{Lp1}(s)$ given in (5) as in the step 1.2, such that to ensure a tradeoff to noise reduction and delay induced by the filter. This filter is also used to obtain accurate derivative estimates $\hat{y}(t)$ characterized by small estimation errors and derivatives smoothing.

Step 2.3. Estimate a small value for the design parameter $e_{est\max}$.

Step 2.4. Fine tune the parameters with small values of K_P and K_I.

Step 2.5. Set the design parameter $T > 0$ to prescribe the desired behavior of the control system on the sliding manifold and to account for the recommendations related to Eq. (47).

Step 2.6. Set small values for the parameters $\psi > 0$ and $\delta > 0$ using the experience of the control systems designer.

4. Experimental case study

This section is dedicated to the real-time experimental validation of the control structures and design approaches proposed in the previous section by the azimuth and pitch positions control of a nonlinear laboratory TRAS. Our two MFSMCS structures belong to the MIMO TRAS control system, where the azimuth and pitch positions are controlled using separate Single Input–Single Output (SISO) control system structures for each control channel, namely for azimuth and pitch. The validation is achieved by two experimental scenarios that illustrate the MFSMCS structures performance. The comparison versus a model-free iPI control system structure is included as well and the control system performance is expressed by measuring the real-time control system responses and comparing the values of two sum-type performance indices. In addition, the parameters of the three model-free controllers are tuned by the minimization of these performance indices, which are considered as objective functions in terms of two separate optimization problems, one for the optimal tuning of the azimuth controllers and the other one for the optimal tuning of the pitch controllers.

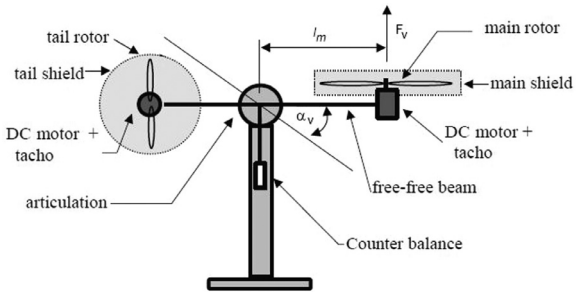

Fig. 4. The schematics of the aero-dynamical model of TRAS [62].

4.1. Experimental setup

The nonlinear state-space model that describes the MIMO TRAS process is [62]

$$\begin{aligned}
&\dot{\Omega}_h = [l_t F_h(\omega_h)\cos\alpha_v - \Omega_h k_h + u_2 k_{vh}]/J_h,\\
&\dot{\Omega}_v = [l_m F_v(\omega_v) - \Omega_v k_v - gC\alpha_v + k_{hv}u_1]/J_v,\\
&\dot{\alpha}_h = \Omega_h,\\
&\dot{\alpha}_v = \Omega_v,\\
&\dot{\omega}_h = (u_1 - k_{Hh}^{-1}(\omega_h))/I_h,\\
&\dot{\omega}_v = (u_2 - k_{Hv}^{-1}(\omega_v))/I_v,\\
&y_1 = \alpha_h,\\
&y_2 = \alpha_v,
\end{aligned} \tag{49}$$

where: u_1(%)—the first control signal, i.e., the Pulse Width Modulation (PWM) duty cycle of the azimuth direct current (DC) motor, u_2(%)—the second control signal, i.e., the PWM duty cycle of the pitch DC motor, α_h(rad)$=y_1$—the first process output, i.e., the azimuth (horizontal) position of the beam that supports the main and the tail rotor, α_v(rad)$=y_2$—the second process output, i.e., the pitch (vertical) position of the beam [50,51,52,55,56,57,58], Ω_h (rad)—the angular (pitch) velocity of the TRAS, Ω_v (rad)—the angular (azimuth) velocity of the TRAS, I_h (kg m^2)—the moment of inertia of the azimuth (tail) rotor, I_v (kg m^2)—the moment of inertia of the pitch (main) rotor, J_v (kg m^2)—the sum of moments of inertia relative to the azimuth (horizontal) axis, J_h (kg m^2)—the sum of moments of inertia relative to the pitch (vertical) axis, $F_h(\omega_h)$(N) and $F_v(\omega_v)$(N)—the nonlinear characteristics that determine the dependence of the propeller trust on DC motor rotational speed (trust characteristics), ω_h (rad/s)—the rotational speed of the azimuth (tail) rotor, ω_v (rad/s)—the rotational speed of the pitch (main) rotor, k_{hv} (N m)—the coefficient of the moment of inertia from azimuth motor to pitch, k_{vh} (N m)—the coefficient of the moment of inertia from pitch motor to azimuth, l_t(m)—the length of the azimuth (tail) part of the beam, l_m(m)—the length of the pitch (main) part of the beam, k_h(N ms)—the horizontal angular momentum, k_v(N ms)—the vertical angular momentum, C (kg m)—the sum between the half of the mass of the counter-weight beam multiplied with the length of the counter-weight beam with the product between the mass of the counter-weight and the distance between the counter-weight and the joint, and g (m/s^2)—the gravitational acceleration.

The variables k_{Hh} and k_{Hv} in (49) depend on ω_h and ω_v, namely $\omega_h = k_{Hh}(u_1)$ (rad/s) and $\omega_v = k_{Hv}(u_2)$ (rad/s), which in turn are used in determining the trust characteristics $F_h(\omega_h)$(N) and $F_v(\omega_v)$(N). The nonlinear state-space model (49) that describes the MIMO TRAS process is only an approximation of the real time equipment. This model has been included to give an overview on the TRAS equipment. That is the reason why the four trust characteristics of the propellers, $\omega_h = k_{Hh}(u_1)$ (rad/s), $\omega_v = k_{Hv}(u_2)$ (rad/s), $F_h(\omega_h)$(N) and $F_v(\omega_v)$(N), which have been measured in [62] to be used by means of their polynomial approximations, and the values of the parameters related to (49) are not included here.

The schematics of the aero-dynamical model of TRAS is given in Fig. 4.

The typical control objective for TRAS is to ensure the regulation and tracking for the vertical and the horizontal motion, i.e., to control the azimuth and the pitch positions. This paper considers a MIMO control system that is decomposed into two SISO control systems, namely the azimuth control loop and the pitch control loop. The experimental results will be presented for both SISO control systems.

The TRAS was chosen because it is a laboratory equipment that has strong nonlinearities and it represents a challenge to control it with MFSMCS1, MFSMCS2 and MFC structures.

4.2. Experimental scenarios

Two experimental scenarios are defined in this sub-section. Each experimental scenario tests three control structures: a model-free structure with iPI controller (MFC) and the two MFSMCS structures (MFSMCS1 and MFSMCS2) introduced in the previous section. The purpose of the experimental scenario is to point out that the MFSMCS1 or the MFSMCS2 structures outperform the MFC structure.

The desired reference trajectories in the first scenario are set to

$$\begin{aligned} y_1^*(k) &= 0.12\sin(0.12k) \text{ if } k = 0...9000, \\ y_2^*(k) &= 0.15\sin(0.15k) \text{ if } k = 0...9000, \end{aligned} \tag{50}$$

where $y_1^*(k)$ and $y_2^*(k)$ are the azimuth and pitch desired reference trajectories, respectively, and k is the discrete time index because all three controllers are actually implemented as quasi-continuous digital controllers. With this regard the sampling period has been set to 0.01s in order to meet the requirements of quasi-continuous digital control of this process.

The first scenario aims to assess how the MFC, MFSMCS1 and MFSMCS2 structures behave if the reference trajectory is a tracking signal represented by a sine signal.

The desired reference trajectories in the second scenario are set to

$$\begin{aligned} y_1^*(k) &= \begin{cases} 0.003k \text{ if } k = 0...3350, \\ 0.15 \text{ if } k = 3351...6500, \\ -0.1 \text{ if } k = 6501...9000, \end{cases} \\ y_2^*(k) &= \begin{cases} 0.003k \text{ if } k = 0...3350, \\ 0.15 \text{ if } k = 3351...6500, \\ -0.1 \text{ if } k = 6501...9000. \end{cases} \end{aligned} \tag{51}$$

The second scenario aims to assess how the MFC, MFSMCS1 and MFSMCS2 structures behave if the reference trajectory is a ramp signal combined with step type signals.

The following performance index will be used to measure the performance of the MFC, MFSMCS1 and MFSMCS2 structures:

$$\begin{aligned} J_\varepsilon^a(\chi^a) &= \sum_{k=1}^{N} (y_1^*(k, \chi^a) - y_1(k, \chi^a))^2, \\ J_\varepsilon^p(\chi^p) &= \sum_{k=1}^{N} (y_2^*(k, \chi^p) - y_2(k, \chi^p))^2, \end{aligned} \tag{52}$$

where the N is the length of the time horizon, $N=9000$, and the superscripts a and p indicate the azimuth and pitch control, respectively, but other objective functions that could account for fuzzy models can be used as well [16,31,43,44,46,70]. The parameter vectors $\boldsymbol{\chi}^a$ and $\boldsymbol{\chi}^p$ contain the parameters of the controllers in the MFC, MFSMCS1 and MFSMCS2 structures:

$$\chi^a = [K_P^a \;\; K_I^a], \chi^p = [K_P^p \;\; K_I^p] \tag{53}$$

for the model-free iPI controllers,

$$\chi^a = [\varepsilon^a \;\; T^a \;\; e_{est\,\max}^a \;\; \eta^a], \chi^p = [\varepsilon^p \;\; T^p \;\; e_{est\,\max}^p \;\; \eta^p] \tag{54}$$

for the controllers in the MFSMCS1 structure, and

$$\chi^a = [K_P^a \;\; K_I^a \;\; \psi^a \;\; T^a \;\; e_{est\,\max}^a \;\; \delta^a], \chi^p = [K_P^p \;\; K_I^p \;\; \psi^p \;\; T^p \;\; e_{est\,\max}^p \;\; \delta^p] \tag{55}$$

for the controllers in the MFSMCS2 structure.

4.3. Real-time experimental results in the first experimental scenario

The steps of the design approaches presented in Sections 3.1 and 3.2 are applied as follows to the TRAS. The parameter α in (1) has been set in step 1.1 and step 2.1 such that to have the same value in all experiments and for each control system structure, $\alpha = 90$ in order not to alter the behavior of the controller to later have a later fair comparison of all experimental scenarios.

The parameters of the first order derivative plus low-pass filter with the transfer function $H_{Lp1}(s)$ in (5) have been set in step 1.2 and step 2.2 to $K_{Lp1}=0.85$ and $T_{Lp1}=0.15$s in all experiments and for each control system structures such that for each experiment all the signals $y^*(t)$ and $y(t)$ shall be filtered through a first order derivative plus low-pass filter having the same parameters in order to have a later fair comparison of all experimental scenarios.

The reference trajectory used in all experiments is given in (50).

The controller parameters in all three model-free control system structures have been optimally tuned as solutions to the following optimization problems:

$$\begin{aligned} \chi^{a*} &= \arg \min_{\chi^a} J_\varepsilon^a(\chi^a), \\ \chi^{p*} &= \arg \min_{\chi^p} J_\varepsilon^p(\chi^p), \end{aligned} \tag{56}$$

where the objective functions are expressed in (52), $\boldsymbol{\chi}^{a*}$ and $\boldsymbol{\chi}^{p*}$ are the optimal parameter vectors for azimuth and pitch control, respectively, and the inequality-type constraints that appear in the design approaches are accounted for. A meta-heuristics Gravitational Search Algorithm (GSA) optimizer, with the parameters given in [8,38] has been applied to solve the optimization problems specified in (56), but other classical or modern optimization algorithms can be used as well [2,3,12,23,28,37,49,63].

A brief description of the GSA is given as follows. The GSA consists of the following steps:

Step 1. The initial populations of agents is generated, i.e., the number of agents N_{GSA} is set and the agents' position vector $\mathbf{V}_i$ is initialized randomly

$$\mathbf{V}_i = [v_i^1 \quad ... \quad v_i^d \quad ... \quad v_i^q]^T, \ i = 1...N_{GSA}, \tag{57}$$

where v_i^d is the position of ith agent in dth dimension, $d = 1...q$, and q is the dimension of the search space that includes the vector variables $\boldsymbol{\chi}^a$ or $\boldsymbol{\chi}^p$ of the objective functions, i.e., the parameter vectors of the controllers.

Using the notation μ for the iteration index in GSA, the maximum number of iterations is set to $\mu_{\max}$.

The GSA is mapped onto the optimization problems defined in (56) by two relationships. First, the relationship between the fitness function f and fitness value $f_i(\mu)$ of ith agent at the iteration index μ in the GSA on the one hand, and the objective functions defined in (52) on the other hand is

$$f_i(\mu) = J_\varepsilon^a(\chi^a), \ f_i(\mu) = J_\varepsilon^a(\chi^a), \ i = 1...N_{GSA}. \tag{58}$$

Second, the relationship between the agents' position vector $\mathbf{V}_i$ in the GSA and the parameter vectors of the controllers $\boldsymbol{\chi}^a$ or $\boldsymbol{\chi}^p$ is

$$\mathbf{V}_i = \chi^a, \mathbf{V}_i = \chi^p, i = 1...N_{GSA}. \tag{59}$$

Step 2. The agents' fitness is evaluated using (52), (58) and (59) and simulations conducted on the control system.

Step 3. The population of agents is updated. The depreciation of the gravitational constant with the advance of GSA's iterations is

$$g(\mu) = g_0 e^{-\alpha_{GSA}\mu/\mu_{\max}}, \tag{60}$$

where $g(\mu)$ is the value of the gravitational constant at the current iteration index, g_0 is the initial gravitational constant, and $\alpha_{GSA} > 0$ is a parameter that is set to ensure GSA's convergence and influence the accuracy of the search process. The expressions of the gravitational and inertial masses are [53]

$$\begin{aligned} &n_i(\mu) = [f_i(\mu) - w(\mu)]/[b(\mu) - w(\mu)], \\ &m_i(\mu) = n_i(\mu)/\sum_{j=1}^{N_a} n_j(\mu), \\ &m_{Ai} = m_{Pi} = m_{Ii} = m_i, \end{aligned} \tag{61}$$

where the terms $b(\mu)$ (corresponding to the best agent) and $w(\mu)$ (corresponding to the worst agent) are computed as follows for the optimization problems that target the minimization of the objective functions:

$$\begin{aligned} b(\mu) &= \min_{j=1...N_{GSA}} f_j(\mu), \\ w(\mu) &= \max_{j=1...N_{GSA}} f_j(\mu). \end{aligned} \tag{62}$$

Step 4. The agents' accelerations $a_i^d(\mu)$ are computed

$$a_i^d(\mu) = (1/m_{Ii}(\mu)) \sum_{j=1, j\neq i}^{N_{GSA}} \rho_j F_{ij}^d(\mu), \tag{63}$$

where $m_{Ii}(\mu)$ is the inertia mass related to ith agent, ρ_j, $0 \le \rho_j \le 1$, are randomly generated numbers, and $F_{ij}^d(\mu)$ is the force acting on ith agent from jth agent

$$F_{ij}^d(\mu) = g(\mu) m_{Pi}(\mu) m_{Aj}(\mu) [v_j^d(\mu) - v_i^d(\mu)]/[||\mathbf{V}_i(\mu) - \mathbf{V}_j(\mu)|| + \varepsilon_{GSA} v_j^d(\mu)], \tag{64}$$

$m_{Pi}(\mu)$ is the passive gravitational mass related to ith agent, $m_{Aj}(\mu)$ is the active gravitational mass related to jth agent, $||\mathbf{V}_i(\mu) - \mathbf{V}_j(\mu)||$ is the Euclidian distance between ith and jth agents, and $\varepsilon_{GSA} > 0$ is a small constant.

The agents' speeds $\vartheta_i^d(\mu+1)$ and positions $v_i^d(\mu+1)$ are updated in terms of [53]

$$\begin{aligned} \vartheta_i^d(\mu+1) &= \rho_i \vartheta_i^d(\mu) + a_i^d(\mu), \\ v_i^d(\mu+1) &= v_i^d(\mu) + \vartheta_i^d(\mu+1), \end{aligned} \tag{65}$$

where ρ_i, $0 \le \rho_i \le 1$, are uniform random variables.

Step 5. The obtained vector solution $\mathbf{V}_i(\mu+1)$ is validated if the objective functions are bounded, i.e.,

$$f_i(\mu) < \varepsilon_f, \tag{66}$$

where the parameter ε_f, $\varepsilon_f > 0$, is set be the designer in order to restrict the feasible domain for $\mathbf{V}_i(\mu+1)$ to guarantee the convergence of the objective functions and the stability of the fuzzy control system as well. The recommended value of ε_f is $\varepsilon_f = 10^5$.

Table 1
Average and variance values of the objective functions for SISO azimuth position control in the first scenario.

	MFC	MFSMCS1	MFSMCS2
Average of J_ε^a	40.78	44.53	25.28
Variance of J_ε^a	310.2827	297.0851	1.9931

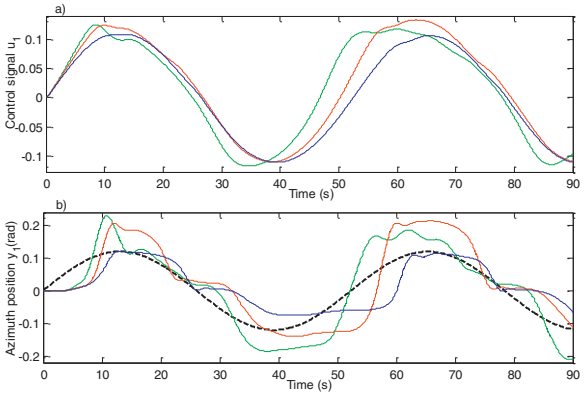

Fig. 5. Experimental results for SISO azimuth position control in the first scenario: a) the control signal u_1 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line), b) the controlled output y_1 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line) and the reference trajectory (black dashed line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Step 6. μ is incremented and the algorithm continues with the step 2 until the maximum number of iterations is reached, i.e., $\mu = \mu_{\max}$, and the final solution is

$$\chi^{a*} = \min_{i=1\ldots N_{GSA}} \mathbf{V}_i(\mu_{\max}),\ \chi^{p*} = \min_{i=1\ldots N_{GSA}} \mathbf{V}_i(\mu_{\max}). \tag{67}$$

The parameters of GSA were set as in [40,41,42] to $N_{GSA}=50$, $\mu_{\max}=100$, $g_0=100$, $\alpha_{GSA}=8.5$ and $\varepsilon_{GSA}=0.0001$. These settings ensure a good convergence of the GSA.

The application of GSA has resulted in the following parameters of the controllers that carry out the azimuth position control:

- the parameters of the model-free iPI controller: $\chi^{a*}=[0.1 \quad 0.001]$,
- the parameters of the controller in the MFSMCS1 structure in steps 1.3, 1.4 and 1.5 of the design approach: $\chi^{a*}=[10 \quad 35.5 \quad 0.0007 \quad 0.1]$,
- the parameters of the controller in the MFSMCS1 structure in steps 2.3, 2.4, 2.5 and 2.6 of the design approach: $\chi^{a*}=[0.001 \quad 0.0001 \quad 0.05 \quad 355 \quad 0.000001 \quad 0.0000012]$.

To reduce the effect of the random disturbances on the measured control system performance, the objective functions of the MFC, MFSMCS1 and MFSMCS2 structures have been averaged after 10 trials. The results for the SISO azimuth position control are synthesized in Table 1, which presents the average and the variance of J_ε^a. The real-time experimental results are also presented as the control system responses expressed as control signals and controlled outputs in Fig. 5.

The parameters of the controllers that carry out the pitch position control, obtained by GSA, are:

- the parameters of the model-free iPI controller: $\chi^{p*}=[0.25 \quad 0.001]$,
- the parameters of the controller in the MFSMCS1 structure in steps 1.3, 1.4 and 1.5 of the design approach: $\chi^{p*}=[10 \quad 4.3 \quad 0.0007 \quad 0.1]$,
- the parameters of the controller in the MFSMCS1 structure in steps 2.3, 2.4, 2.5 and 2.6 of the design approach: $\chi^{p*}=[0.00085 \quad 0.0001 \quad 0.05 \quad 350 \quad 0.0001 \quad 0.0002]$.

The results are presented by averaging the measurements after 10 experiments conducted on the control system structures. The results of the average and variance values of the objective functions J_ε^p for the SISO pitch position control are presented in Table 2. The evolution of the control signals and controlled outputs is illustrated in Fig. 6.

4.4. Real-time experimental results in the second experimental scenario

The same first two steps of the design approaches as in the first experimental scenario have been applied in this scenario as well, GSA has been employed as in Section 4.3 in solving the optimization problem (56), but the reference trajectory used in all experiments is given in (51).

Table 2
Average and variance values of the objective functions for SISO pitch position control in the first scenario.

	MFC	MFSMCS1	MFSMCS2
Average of J_ε^p	13.52	12.90	10.31
Variance of J_ε^p	0.6950	0.2904	0.2341

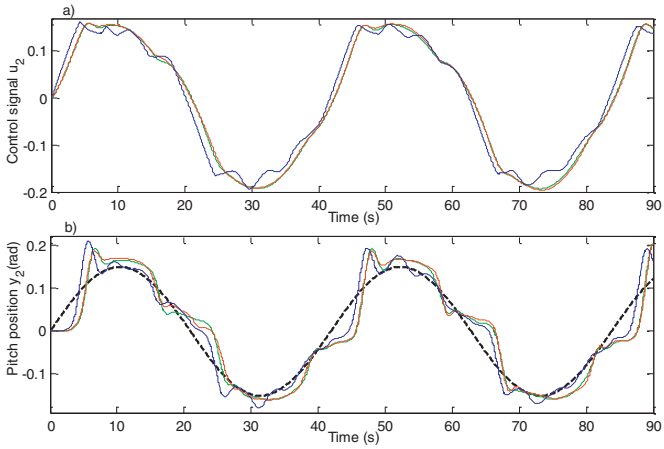

Fig. 6. Experimental results for SISO pitch position control in the first scenario: a) the control signal u_2 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line), b) the controlled output y_2 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line) and the reference trajectory (black dashed line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Average and variance values of the objective functions for SISO azimuth position control in the second scenario.

	MFC	MFSMCS1	MFSMCS2
Average of J_ε^a	18.40	18.12	53.30
Variance of J_ε^a	1.9539	8.2650	56.6316

The application of GSA has led to the following parameter vectors of the controllers that carry out the azimuth position control:

- the parameters of the model-free iPI controller: $\chi^{a*}=[0.15 \;\; 0.001]$,
- the parameters of the controller in the MFSMCS1 structure in steps 1.3, 1.4 and 1.5 of the design approach: $\chi^{a*}=[20 \;\; 5.5 \;\; 0.0007 \;\; 0.1]$,
- the parameters of the controller in the MFSMCS1 structure in steps 2.3, 2.4, 2.5 and 2.6 of the design approach: $\chi^{a*}=[0.001 \;\; 0.0001 \;\; 0.05 \;\; 355 \;\; 0.0003 \;\; 0.000095]$.

To avoid the effects of random disturbances, the objective functions of the MFC, MFSMCS1 and MFSMCS2 structures have been averaged 10 times. The results concerning the objective functions for SISO azimuth position control are presented in Table 3. The control signals and controlled outputs versus time are shown in Fig. 7.

GSA has been also applied to tine the parameters of the controllers that carry out the pitch position control, and the obtained parameter vectors are

- the parameters of the model-free iPI controller: $\chi^{p*}=[0.35 \;\; 0.001]$,
- the parameters of the controller in the MFSMCS1 structure in steps 1.3, 1.4 and 1.5 of the design approach: $\chi^{p*}=[10 \;\; 2.5 \;\; 0.0007 \;\; 0.1]$,
- the parameters of the controller in the MFSMCS1 structure in steps 2.3, 2.4, 2.5 and 2.6 of the design approach: $\chi^{p*}=[0.09 \;\; 0.01 \;\; 0.05 \;\; 350 \;\; 0.0001 \;\; 0.0001]$.

The effects of random disturbances on the results have been mitigated by conducting 10 experiments/trials for each control system structure and the results are presented as averaged values. The results related to the average and variance values of the objective functions J_ε^p for the SISO pitch position control are presented in Table 4. The real-time experimental results are also given in Fig. 8 as control system responses in terms of control signals and controlled outputs versus time.

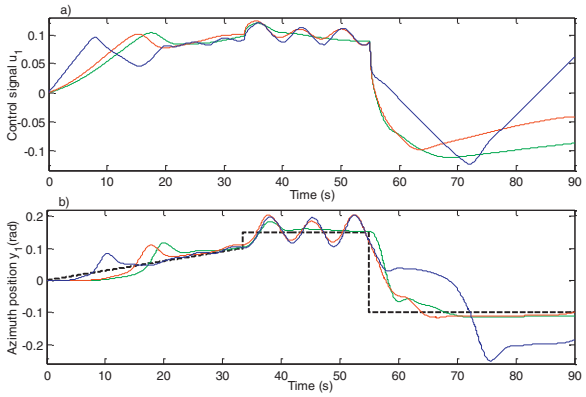

Fig. 7. Experimental results for SISO azimuth position control in the second scenario: a) the control signal u_1 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line), b) the controlled output y_1 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line) and the reference trajectory (black dashed line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Average and variance values of the objective functions for SISO pitch position control in the second scenario.

	MFC	MFSMCS1	MFSMCS2
Average of J_ε^p	12.13	11.59	13.52
Variance of J_ε^p	0.1031	0.3098	0.3658

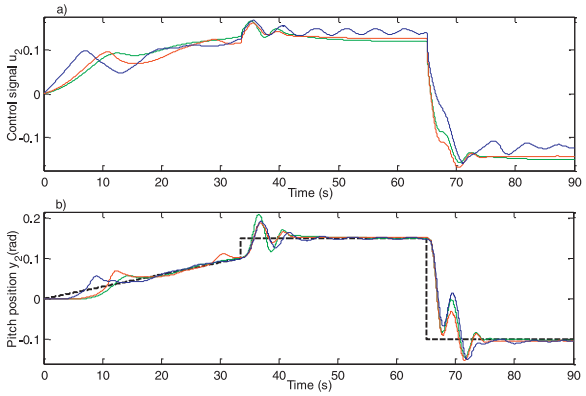

Fig. 8. Experimental results for SISO pitch position control in the second scenario: a) the control signal u_2 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line), b) the controlled output y_2 of MFC (green line), MFSMCS1 (red line) and MFSMCS2 (blue line) and the reference trajectory (black dashed line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.5. Discussion

Table 1 and Fig. 5 related to the first experimental scenario show that for SISO azimuth position control the best results have been obtained by the MFSMCS2 structure followed by the MFC structure and the MFSMCS1 structure, i.e., $J_{\varepsilon\ MFSMCS2}^a < J_{\varepsilon\ MFC}^a < J_{\varepsilon\ MFSMCS1}^a$. The reason for this ranking is represented by the following relationships between the average measured objective functions: $1.61J_{\varepsilon\ MFSMCS2}^a \approx J_{\varepsilon\ MFC}^a$ and $1.76J_{\varepsilon\ MFSMCS2}^a \approx J_{\varepsilon\ MFSMCS1}^a$. So the MFSMCS2 structure outperforms the basic MFC structure, mentioning that both MFSMCS1 and MFSMCS2 structures are built starting with the MFC algorithm.

Table 2 and Fig. 6 show that for SISO pitch position control the best performance has been achieved by the MFSMCS2 structure, then the MFSMCS1 structure and finally the MFC structure. The relationships between the average measured objective functions obtained after the real-time experiments are $J_{\varepsilon\ MFSMCS2}^p < J_{\varepsilon\ MFSMCS1}^p < J_{\varepsilon\ MFC}^p$, $1.25J_{\varepsilon\ MFSMCS2}^p \approx J_{\varepsilon\ MFSMCS1}^p$ and $1.31J_{\varepsilon\ MFSMCS2}^p \approx J_{\varepsilon\ MFC}^p$. So the MFSMCS2 structure outperforms the basic MFC structure, mentioning that both MFSMCS1 and MFSMCS2 structures are built starting with the MFC algorithm.

Table 3 and Fig. 7 related to the second experimental scenario applied to azimuth position control show that the MFSMCS1 structure and the MFC structure exhibit similar performance and they also perform approximate 2.9 times better than the MFSMCS2 structure. The relationships between the average measured objective functions are $1.01J_{\varepsilon\ MFSMCS1}^a \approx$

$J^a_{\varepsilon\ MFC}$, $2.94J^a_{\varepsilon\ MFSMCS1} \approx J^a_{\varepsilon\ MFSMCS2}$ and $J^a_{\varepsilon\ MFSMCS1} < J^a_{\varepsilon\ MFC} < J^a_{\varepsilon\ MFSMCS2}$. So the MFSMCS1 structure outperforms the basic MFC structure, mentioning that both MFSMCS1 and MFSMCS2 structures are built starting with the MFC algorithm.

Table 4 and Fig. 8 show that the best performance for SISO pitch position control has been achieved by the MFSMCS1 structure, and similar performance has been obtained by the MFC structure and next followed by the MFSMCS2 structure. The relationships between the average measured objective functions obtained after the real-time experiments are $J^p_{\varepsilon\ MFSMCS1} < J^p_{\varepsilon\ MFC} < J^p_{\varepsilon\ MFSMCS2}$, $1.04J^p_{\varepsilon\ MFSMCS1} \approx J^p_{\varepsilon\ MFC}$ and $1.16J^p_{\varepsilon\ MFSMCS1} \approx J^p_{\varepsilon\ MFSMCS2}$. So the MFSMCS1 structure outperforms the basic MFC structure, mentioning that both MFSMCS1 and MFSMCS2 structures are built starting with the MFC algorithm.

The above comparison will be different for other nonlinear processes. Such examples of representative processes are given in [1,20,33,34,39,47,61]. In addition, the performance improvement exhibited by our MFSMCS structures is not spectacular as these structures do not perform the direct model-free tuning of sliding mode controllers. As shown in Section 3, sliding mode control is applied to the model represented by the tracking error dynamics obtained by MFC and not to the process itself.

5. Conclusions

This paper has proposed two model-free control system structures that include sliding mode control. The two sliding mode control system structures control the tracking error dynamics obtained by model-free control in order to benefit from the specific advantages, i.e., robustness against parameter variations and disturbances.

Our control system structures and design approaches have been validated by real-time experimental results related to the position control of laboratory equipment represented by the twin rotor aerodynamic system. The model-free controllers have been implemented as azimuth and pitch position controllers in two SISO control loops. The performance comparison with a model-free intelligent PI control system structures proves the performance improvement and effectiveness of the new model-free control system structures.

It is not necessary to tune optimally the parameters of the three model-free controllers but this is a fair way to carry out the systematic performance comparison of the model-free control system structures presented in this paper. Different results will be obtained if other performance indices viewed as objective functions are used.

Future research will be focused on the discrete-time formulation of the control system structures, on the extension to MIMO model-free sliding mode control and on the direct model-free tuning of sliding mode controllers.

Acknowledgments

This work was supported by grants of the Romanian National Authority for Scientific Research and Innovation—the Executive Agency for Higher Education, Research, Development and Innovation Funding (CNCS—UEFISCDI), project numbers PN-II-RU-TE-2014-4-0207, PN-II-PT-PCCA-2013-4-0544 and PN-II-PT-PCCA-2013-4-0070, and from the NSERC of Canada.

References

[1] R.A. Aliev, W. Pedrycz, V. Kreinovich, O.H. Huseynov, The general theory of decisions, Inf. Sci. 327 (2016) 125–148.
[2] P. Angelov, R. Yager, Density-based averaging—a new operator for data fusion, Inf. Sci. 222 (2013) 163–174.
[3] S. Blažič, D. Matko, I. Škrjanc, Adaptive law with a new leakage term, IET Control Theory Appl. 4 (9) (2010) 1533–1542.
[4] F.J. Carrillo, F. Rotella, Some contributions to estimation for mode-free control, in: Proceedings of 17th IFAC Symposium on System Identification, Beijing, China, 2015, pp. 150–155.
[5] W.-J. Chang, F.-L. Hsu, Sliding mode control for Takagi–Sugeno fuzzy systems with bilinear consequent part subject to multiple constrains, Inf. Sci. 327 (2016) 258–271.
[6] W. Chen, Y.-Q. Chen, C.-P. Yeh, Robust iterative learning control via continuous sliding-mode technique with validation on an SRV02 rotary plant, Mechatronics 22 (5) (2012) 588–593.
[7] M.L. Corradini, A. Cristofaro, G. Orlando, Stabilization of discrete-time linear systems with saturating actuators using sliding modes: application to a twin-rotor system, in: Proceedings 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 2011, pp. 8237–8242.
[8] R.-C. David, R.-E. Precup, E.M. Petriu, M.-B. Radac, S. Preitl, Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity, Inf. Sci. 247 (2013) 154–173.
[9] J. Davila, L. Fridman, A. Levant, Second-order sliding-mode observer for mechanical systems, IEEE Trans. Autom. Control 50 (11) (2005) 1785–1789.
[10] D. Efimov, A. Polyakov, L. Fridman, W. Perruquetti, Delayed sliding mode control, Automatica 64 (2016) 37–43.
[11] Q. Fan, G. Yang, Adaptive actor-critic design-based integral sliding-mode control for partially unknown nonlinear systems with input disturbances, IEEE Trans. Neural Networks Learn. Syst. 27 (1) (2016) 165–177.
[12] F.G. Filip, Decision support and control for large-scale complex systems, Annu. Rev. Control 32 (1) (2008) 61–70.
[13] M. Fliess, C. Join, Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control? in: Proceedings of 15th IFAC Symposium on System Identification, Saint-Malo, France, 2009, pp. 1531–1550.
[14] M. Fliess, C. Join, Model-free control, Int. J. Control 86 (12) (2013) 2228–2252.
[15] M. Fliess, C. Join, M. Mboup, H. Sira-Ramirez, Vers une commande multivariable sans modele, in: Proc. Conférence Internationale Francophone d'Automatique, Bordeaux, France, 2006, pp. 1–7.
[16] S. Formentin, A. Karimi, S.M. Savaresi, Optimal input design for direct data-driven tuning of model-reference controllers, Automatica 49 (6) (2013) 1874–1882.
[17] P.-A. Gedouin, E. Delaleau, J.-M. Bourgeot, C. Join, S.A. Chirani, S. Calloch, Experimental comparison of classical PID and model-free control: position control of a shape memory alloy active spring, Control Eng. Pract. 19 (5) (2011) 433–441.
[18] Z.-S. Hou, Z. Wang, From model-based control to data-driven control: survey, classification and perspective, Inf. Sci. 235 (2013) 3–35.
[19] M.A. Jama, H. Noura, A. Wahyudie, A. Assi, Enhancing the performance of heaving wave energy converters using model-free control approach, Renewable Energy 83 (2015) 931–941.

[20] Z.C. Johanyák, Fuzzy modeling of thermoplastic composites' melt volume rate, Comput. Inf. 32 (4) (2013) 845–857.
[21] C. Join, G. Robert, M. Fliess, Model-free based water level control for hydroelectric power plants, in: Proceedings of the IFAC Conference on Control Methodologies and Technology for Energy Efficiency, Vilamoura, Portugal, 2010, pp. 134–139.
[22] F. Lafont, J.-F. Balmat, N. Pessel, M. Fliess, A model-free control strategy for an experimental greenhouse with an application to fault accommodation, Comput. Electron. Agric. 110 (2015) 139–149.
[23] X.-J. Lei, F. Wang, F.-X. Wu, A.-D. Zhang, W. Pedrycz, Protein complex identification through Markov clustering with firefly algorithm on dynamic protein–protein interaction networks, Inf. Sci. 329 (2016) 303–316.
[24] H. Li, P. Shi, D. Yao, L. Wu, Observer-based adaptive sliding mode control for nonlinear Markovian jump systems, Automatica 64 (2016) 133–142.
[25] Y. Li, J.-B. Son, J.-M. Lee, PMSM speed controller using switching algorithm of PD and sliding mode control, in: Proceedings 2009 ICCAS-SICE Conference, Fukuoka, Japan, 2009, pp. 1260–1266.
[26] M. Liu, X.-B. Cao, S.-J. Zhang, W. Yang, Sliding mode control of quantized systems against bounded disturbances, Inf. Sci. 274 (2014) 261–272.
[27] L. Ma, C. Wang, S. Ding, L. Dong, Integral sliding mode control for stochastic Markovian jump system with time-varying delay, Neurocomputing 179 (2016) 118–125.
[28] D. Martín, R. Del Toro, R. Haber, J. Dorronsoro, Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process, Int. J. Innovative Comput. Inf. Control 5 (2009) 3405–3414 10(B).
[29] L. Menhour, B. d'Andréa-Novel, M. Fliess, D. Gruyer, H. Mounier, A new model-free design for vehicle control and its validation through an advanced simulation platform, in: Proceedings of 14th European Control Conference, Linz, Austria, 2015, pp. 1–6.
[30] C. Milosavljevic, B. Perunicic-Drazenovic, B. Veselic, Discrete-time velocity servo system design using sliding mode control approach with disturbance compensation, IEEE Trans. Ind. Inf. 9 (2) (2013) 920–927.
[31] P. Moallem, B.S. Mousavi, S.S. Naghibzadeh, Fuzzy inference system optimized by genetic algorithm for robust face and pose detection, Int. J. Artif. Intell. 13 (2) (2015) 73–88.
[32] R.K. Munje, B.M. Patre, A.P. Tiwari, Discrete-time sliding mode spatial control of advanced heavy water reactor, IEEE Trans. Control Syst. Technol. 24 (1) (2016) 357–364.
[33] R. Palm, D. Driankov, Velocity potentials and fuzzy modeling of fluid streamlines for obstacle avoidance of mobile robots, in: Proceedings of 2015 IEEE International Conference on Fuzzy Systems, Istanbul, Turkey, 2015, pp. 1–8.
[34] Z.-H. Peng, D. Wang, Y. Shi, H. Wang, W. Wang, Containment control of networked autonomous underwater vehicles with model uncertainty and ocean disturbances guided by multiple leaders, Inf. Sci. 316 (2015) 163–179.
[35] A. Pisano, M. Tanelli, A. Ferrara, Switched/time-based adaptation for second-order sliding mode control, Automatica 64 (2016) 126–132.
[36] B. Pratap, S. Purwar, Real-time implementation of neuro adaptive observer-based robust backstepping controller for twin rotor control system, J. Control Autom. Electr. Syst. 25 (2) (2014) 137–150.
[37] R.-E. Precup, P. Angelov, B.S.J. Costa, M. Sayed-Mouchaweh, An overview on fault diagnosis and nature-inspired optimal control of industrial process applications, Comput. Ind. 74 (2015) 75–94.
[38] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Radac, Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy controlled servo systems, IET Control Theory Appl. 7 (1) (2013) 99–107.
[39] R.-E. Precup, C.-A. Dragos, S. Preitl, M.-B. Radac, E.M. Petriu, Novel tensor product models for automatic transmission system control, IEEE Syst. J. 6 (3) (2012) 488–498.
[40] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, A.S. Paul, Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity, in: A. Gaspar-Cunha, R. Takahashi, G. Schaefer, L. Costa (Eds.), Advances in Intelligent and Soft Computing, Eds., Springer-Verlag, Berlin, Heidelberg, 2011, pp. 141–150.
[41] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Radac, Gravitational search algorithms in fuzzy control systems tuning, in: Proceedings of 18th World Congress of the International Federation of Automatic Control (IFAC 2011), Milano, Italy, 2011, pp. 13624–13629.
[42] R.-E. Precup, R.-C. David, E.M. Petriu, M.-B. Radac, S. Preitl, J. Fodor, Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems, Knowl. Based Syst. 28 (2013) 74–84.
[43] R.-E. Precup, S. Preitl, Stability and sensitivity analysis of fuzzy control systems. Mechatronics applications, Acta Polytech. Hung. 3 (1) (2006) 61–76.
[44] R.-E. Precup, S. Preitl, M. Balas, V. Balas, Fuzzy controllers for tire slip control in anti-lock braking systems, in: Proceedings 2004 IEEE International Conference on Fuzzy Systems, 3, Budapest, Hungary, 2004, pp. 1317–1322.
[45] R.-E. Precup, M.-B. Radac, E.M. Petriu, C.-A. Dragos, S. Preitl, Model-free tuning solution for sliding mode control of servo systems, in: Proceedings Eighth Annual IEEE International Systems Conference, Ottawa, ON, Canada, 2014, pp. 30–35.
[46] R.-E. Precup, M.-B. Radac, M.L. Tomescu, E.M. Petriu, S. Preitl, Stable and convergent iterative feedback tuning of fuzzy controllers for discrete-time SISO systems, Expert Syst. Appl. 40 (1) (2013) 188–199.
[47] R.-E. Precup, M.L. Tomescu, S. Preitl, Fuzzy logic control system stability analysis based on Lyapunov's direct method, Int. J. Comput. Commun. Control 4 (4) (2009) 415–426.
[48] D. Qian, J. Yi, D. Zhao, Robust control using sliding mode for a class of under-actuated systems with mismatched uncertainties, in: Proceedings of 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 1449–1457.
[49] Q.-D. Qin, S. Cheng, Q.-Y. Zhang, L. Li, Y. Shi, Biomimicry of parasitic behavior in a coevolutionary particle swarm optimization algorithm for global optimization, Appl. Soft Comput. 32 (2015) 224–240.
[50] M.-B. Radac, R.-E. Precup, E.M. Petriu, S. Preitl, Iterative data-driven tuning of controllers for nonlinear systems with constrains, IEEE Trans. Ind. Electron. 61 (11) (2014) 6360–6368.
[51] M.-B. Radac, R.-E. Precup, S. Preitl, C.-A. Dragos, Constrained data-driven controller tuning for nonlinear systems, in: Proceedings 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 2013, pp. 3404–3409.
[52] M.-B. Radac, R.-C. Roman, R.-E. Precup, E.M. Petriu, Data-driven model-free control of twin rotor aerodynamic systems: Algorithms and experiments, in: Proceedings of 2014 IEEE Multi Systems Conference, Antibes, France, 2014, pp. 1889–1894.
[53] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[54] T.M. Ridha, C.-H. Moog, Model free control for type-1 diabetes: a fasting-phase study, in: Proceedings of Ninth IFAC Symposium on Biological and Medical Systems, Berlin, Germany, 2015, pp. 76–81.
[55] R.-C. Roman, M.-B. Radac, R.-E. Precup, Data-driven model-free adaptive control of twin rotor aerodynamic systems, in: Proceedings of 2014 IEEE Ninth International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, 2014, pp. 25–30.
[56] R.-C. Roman, M.-B. Radac, R.-E. Precup, E.M. Petriu, two data-driven control algorithms for a MIMO aerodynamic system with experimental validation, in: Proceedings of 2015 19th International Conference on System Theory, Control and Computing, Cheile Gradistei, Romania, 2015, pp. 736–741.
[57] R.-C. Roman, M.-B. Radac, R.-E. Precup, E.M. Petriu, Data-driven optimal model-free control of twin rotor aerodynamic systems, in: Proceedings of 2015 IEEE International Conference on Industrial Technology, Seville, Spain, 2015, pp. 161–166.
[58] R.-C. Roman, M.-B. Radac, R.-E. Precup, E.M. Petriu, Data-driven model-free adaptive control tuned by virtual reference feedback tuning, Acta Polyteh. Hung. 13 (1) (2016) 83–96.
[59] D.K. Saroj, I. Kar, V.K. Pandey, Sliding mode controller design for twin rotor MIMO system with a nonlinear state observer, in: Proceedings of 2013 International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing, Kottayam, India, 2013, pp. 668–673.
[60] J. Song, Y. Niu, Y. Zou, Finite-time sliding mode control synthesis under explicit output constraint, Automatica 65 (2016) 111–114.
[61] N. Tomin, A. Zhukov, D. Sidorov, V. Kurbatsky, D. Panasetsky, V. Spiryaev, Random forest based model for preventing large-scale emergencies in power systems, Int. J. Artif. Intell. 13 (1) (2015) 211–228.
[62] Two Rotor Aerodynamical System, User's Manual, Inteco Ltd., Krakow, Poland, 2007.

[63] J. Vaščák, Adaptation of fuzzy cognitive maps by migration algorithms, Kybernetes 41 (3–4) (2013) 429–443.
[64] B. Wang, B. Brogliato, V. Acary, A. Boubakir, F. Plestan, Experimental comparisons between implicit and explicit implementations of discrete-time sliding mode controllers: toward input and output chattering suppression, IEEE Trans. Control Syst. Technol. 23 (5) (2015) 2071–2075.
[65] H. Wang, X. Ye, Y. Tian, N. Christov, Attitude control of a quadrotor using model free based sliding model controller, in: Proceedings of 2015 20th International Conference on Control Systems and Science, Bucharest, Romania, 2015, pp. 149–154.
[66] J. Wang, M.-S. Geamanu, A. Cela, H. Mounier, S.-I. Niculescu, Event driven model free control of quadrotor, in: Proceedings of 2013 IEEE International Conference on Control Applications, Hyderabad, India, 2013, pp. 722–727.
[67] X. Wang, X. Li, J. Wang, X. Fang, X. Zhu, Data-driven model-free adaptive sliding mode control for the multi degree-of-freedom robotic exoskeleton, Inf. Sci. 327 (2016) 246–257.
[68] X.-S. Wang, S.-Y. Su, H. Hong, Adaptive sliding inverse control of a class of nonlinear systems preceded by unknown non-symmetrical dead-zone, in: Proceedings of 2003 IEEE International Symposium on Intelligent Control, Huston, TX, USA, 2003, pp. 16–21.
[69] H. Yu, U. Ozguner, Adaptive seeking sliding mode control, in: Proceedings of 2006 American Control Conference, Minneapolis, MN, USA, 2006, pp. 4694–4699.
[70] W. Zheng, N. Hu, Automated test sequence optimization based on the maze algorithm and ant colony algorithm, Int. J. Comput. Commun. Control 10 (4) (2015) 593–606.

Knowledge-Based Systems 38 (2013) 74–84

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems

Radu-Emil Precup [a,*], Radu-Codruţ David [a], Emil M. Petriu [b], Mircea-Bogdan Rădac [a], Stefan Preitl [a], János Fodor [c]

[a] Department of Automation and Applied Informatics, Faculty of Automation and Computers, "Politehnica", University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania
[b] School of Information Technology and Engineering, University of Ottawa, 800 King Edward, Ottawa, ON, Canada K1N 6N5
[c] Institute of Intelligent Engineering Systems, Óbuda University, Bécsi út 96/B, H-1034 Budapest, Hungary

ARTICLE INFO

Article history:
Available online 26 July 2011

Keywords:
Gravitational Search Algorithm
Parametric sensitivity
Particle Swarm Optimization
Simulated Annealing
Takagi–Sugeno PI fuzzy controllers

ABSTRACT

This paper suggests the optimal tuning of low-cost fuzzy controllers dedicated to a class of servo systems by means of three new evolutionary optimization algorithms: Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO) algorithm and Simulated Annealing (SA) algorithm. The processes in these servo systems are characterized by second-order models with an integral component and variable parameters; therefore the objective functions in the optimization problems include the output sensitivity functions of the sensitivity models defined with respect to the parametric variations of the processes. The servo systems are controlled by Takagi–Sugeno proportional-integral-fuzzy controllers (T–S PI-FCs) that consist of two inputs, triangular input membership functions, nine rules in the rule base, the SUM and PROD operators in the inference engine, and the weighted average method in the defuzzification module. The T–S PI-FCs are implemented as low-cost fuzzy controllers because of their simple structure and of the only three tuning parameters because of mapping the parameters of the linear proportional-integral (PI) controllers onto the parameters of the fuzzy ones in terms of the modal equivalence principle and of the Extended Symmetrical Optimum method. The optimization problems are solved by GSA, PSO and SA resulting in fuzzy controllers with a reduced parametric sensitivity. The comparison of the three evolutionary algorithms is carried out in the framework of a case study focused on the optimal tuning of T–S PI-FCs meant for the position control system of a servo system laboratory equipment. Reduced process gain sensitivity is ensured.

1. Introduction

The performance specifications in many control system applications can be fulfilled by fuzzy control and fuzzy models as convenient nonlinear control strategies [2,6,7,31,33,40,48]. The performance specifications can be fulfilled optimally and systematically by the definition of optimization problems where the objective functions are integral performance indices, the variables are the tuning parameters of the fuzzy controllers, and the constraints should be accounted for. The solutions to the optimization problems are the optimal tuning parameters as part of optimal fuzzy control systems.

The evolutionary algorithms prove to be successful in the optimal tuning of fuzzy control systems. Some current approaches include the Simulated Annealing (SA) algorithms in conventional and adaptive fuzzy control system structures [18,21,38], the piecewise parametric polynomial fuzzy sets in optimization [11], the cross-entropy method [17], the fuzzy bang-bang control for minimum time response [32], the genetic algorithms in sliding mode and cascade fuzzy control systems [9,34], and the differential evolution and similarity classifier results [1,30].

This paper will discuss three new evolutionary optimization algorithms dedicated to the optimal tuning of low-cost fuzzy controllers for servo systems. The three optimization algorithms use Gravitational Search Algorithms (GSAs) [43,44], Particle Swarm Optimization (PSO) algorithms [26,27] and SA algorithms [15,25]. Building upon our previous works on GSAs for the optimal tuning of Takagi–Sugeno proportional-integral (PI)-fuzzy controllers [37,39] and on PSO algorithms for the optimal tuning of PI controllers and of Takagi–Sugeno PI-fuzzy controllers (T–S PI-FCs) [12,13], the new evolutionary optimization algorithms presented in this paper aim the minimization of the objective functions expressed as discrete-time weighted sums of the control error and of the output sensitivity functions. The output sensitivity functions are taken from the sensitivity models defined with respect to the parametric

* Corresponding author.Tel.: +40 2564032 29/30/26; fax: +40 256403214.
E-mail addresses: radu.precup@aut.upt.ro (R.-E. Precup), davidradu@gmail.com (R.-C. David), petriu@site.uottawa.ca (E.M. Petriu), mircea.radac@aut.upt.ro (M.-B. Rădac), stefan.preitl@aut.upt.ro (S. Preitl), fodor@uni-obuda.hu (J. Fodor).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.07.006

variations of the process that belongs to a class of servo systems characterized by linear or linearized second-order models with an integral component. Since the variables in the optimization problems solved by the new GSA, PSO algorithm and SA algorithm are the parameters of T–S PI-FCs, our algorithms lead to optimal T–S PI-FCs.

The main new contributions of this paper with respect to the current treated in the literature are:

- Three new evolutionary optimization algorithms that use GSA, PSO and SA are offered to ensure the optimal tuning of low-cost fuzzy controllers for servo systems with a reduced parametric sensitivity. Our new algorithms are characterized by the implementation of a constraint that ensures the convergence of the objective functions, which is inserted to validate the solution for the next iteration, and by the calculation of the state-space models of the T–S PI-FCs. In order to make the SA algorithm more computationally efficient two additional iteration indices are introduced, the success rate and the rejection rate, aiming the acceleration of the cooling process and to improve the convergence of the algorithm when small values of the objective functions are found.
- A new generation of low-cost optimal T–S PI-FCs is given. The T–S PI-FCs consist of two inputs, triangular input membership functions, nine rules in the rule base, the SUM and PROD operators in the inference engine, and the weighted average method in the defuzzification module.
- An analysis of the comparison between the three evolutionary optimization algorithms is conducted accepting a case study related to the angular position control of a direct current (DC) servo system laboratory equipment. Digital and experimental results are included to support this analysis. A reduced process gain sensitivity is obtained, i.e., a reduced sensitivity with respect to the modifications of the process gain.

These new contributions are important and advantageous with respect to the state-of-the-art because:

- Our evolutionary optimization algorithms offer a reduced sensitivity with respect to the parameters involved in the sensitivity models, which enables the use of simplified process models in the design and tuning of fuzzy and of intelligent control systems [5,10,29,49,51,53].
- Our optimal T–S PI-FCs are implemented as low-cost fuzzy controllers in the framework of low-cost automation solutions [8,28,35,36,41,46] because of the relatively small number of tuning parameters used as variables in the objective functions, and of the simplicity of the unified tuning approach used for GSA, PSO and SA as well.

This paper is organized as follows. The definition of the optimization problem and the new evolutionary algorithms are presented in Section 2 and in Section 3, respectively. Section 4 is next dedicated to the case study of the tuning of T–S PI-FCs for a class of servo systems where the process is described by second-order linear or linearized models with an integral component. A unified tuning approach for low-cost T–S PI-FCs proposed that results in a new generation of low-cost optimal T–S PI-FCs. Section 5 validates the new fuzzy controllers and optimization algorithms by a case study concerning the angular position control of a DC servo system laboratory equipment and aiming a reduced process gain sensitivity. An analysis of the comparison of the three optimization algorithms, digital simulation results and real-time experimental results are included. The conclusions are pointed out in Section 6.

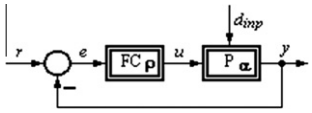

Fig. 1. Structure of the fuzzy control system.

2. Optimization problems

The fuzzy control system structure is presented in Fig. 1, where FC is the fuzzy controller, P is the process, r is the reference input, d_{inp} is the disturbance input, y is the controlled output, u is the control signal, e is the control error

$$e = r - y, \tag{1}$$

$\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_m]^T \in R^m$ is the parameter vector with the elements α_a, $a = 1 \dots m$, which are the parameters of the process, $\boldsymbol{\rho} = [\rho_1 \quad \rho_2 \quad \dots \quad \rho_q]^T \in R^q$ is the parameter vector with the elements ρ_l, $l = 1 \dots q$, which are the tuning parameters of the controller, and the superscript T indicates the matrix transposition.

The control system structure presented in Fig. 1 can be extended to tracking controllers in terms of adding an additional block. This leads to two-degree-of-freedom (2-DOF) control system structures that can ensure improved regulation and tracking performance with respect to the one-degree-of-freedom (1-DOF) control system structures. Several ways to introduce this additional block in fuzzy control systems are presented in [40] with emphasis on PI-FCs.

The process is represented by the Single Input–Single Output (SISO) discrete-time state-space model

$$\begin{aligned} &\mathbf{x}_P(t+1) = \mathbf{f}_{P,d}(\mathbf{x}_P(t), \boldsymbol{\alpha}, u(t), d_{inp}(t)),\\ &y(t) = g_{P,d}(\mathbf{x}_P(t), \boldsymbol{\alpha}, d_{inp}(t)),\\ &\mathbf{x}_P(t_0) = \mathbf{x}_{P,0}, \end{aligned} \tag{2}$$

where $t_0 \in N$ is the initial time moment, $t, t \in N$, $t \geqslant t_0$, is the discrete time argument, $\mathbf{x}_P = [x_{P,1} \quad x_{P,2} \quad \dots \quad x_{P,n}]^T \in R^n$ is the state vector of the process, $\mathbf{x}_{P,0} \in R^n$ is the initial state vector of the process, and the functions $\mathbf{f}_{P,d}: R^{n+m+2} \to R^n$ and $g_{P,d}: R^{n+m+1} \to R$ are differentiable with respect to the parameter α_a, $a = 1 \dots m$. The state-space models presented in (2) is a nonlinear model without direct feedthrough, and it can be obtained from a SISO continuous-time state-space model accepting that the inputs u and d_{inp} are changing at the discrete sampling intervals, i.e., accepting the presence of the zero-order hold (ZOH).

The fuzzy controller is characterized by the general nonlinear SISO discrete-time state-space model

$$\begin{aligned} &\mathbf{x}_C(t+1) = \mathbf{f}_{C,d}(\mathbf{x}_C(t), \boldsymbol{\rho}, e(t)),\\ &u(t) = g_{C,d}(\mathbf{x}_C(t), \boldsymbol{\rho}, e(t)),\\ &\mathbf{x}_C(t_0) = \mathbf{x}_{C,0}, \end{aligned} \tag{3}$$

where $\mathbf{x}_C = [x_{C,1} \quad x_{C,2} \quad \dots \quad x_{C,p}]^T \in R^p$ is the state vector of the controller, $\mathbf{x}_{C,0} \in R^p$ is the initial state vector of the controller, $\mathbf{f}_{C,d}: R^{p+q+1} \to R^p$ and $g_{C,d}: R^{p+q+1} \to R$, and the convergence of the integrals in the objective functions requires that the controller should have an integral component in order to ensure the zero steady-state value of the control error for several types of disturbance inputs. The functions $\mathbf{f}_{C,d}: R^{p+q+1} \to R^p$ and $g_{C,d}: R^{p+q+1} \to R$ in the model (3) should contribute to the assurance of the differentiability of state-space model of the fuzzy control system with respect to the parameter α_a, $a = 1 \dots m$.

The state vector of the process $\mathbf{x}_P$ and the state vector of the controller $\mathbf{x}_C$ are grouped in the state vector of the control system $\mathbf{x}$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_P \\ \mathbf{x}_C \end{bmatrix} = [\, x_1 \quad x_2 \quad \dots \quad x_{n+p} \,]^T \in R^{n+p},$$
$$x_b = \begin{cases} x_{P,i} & \text{if} b = \overline{1,n}, \\ x_{C,i-n} & \text{otherwise}, \end{cases} \quad b = \overline{1,n+p}. \tag{4}$$

The state-space models (2) and (3) are merged using Eqs. (1) and (4) leading to the following discrete-time state-space model of the fuzzy control system:

$$\mathbf{x}(t+1) = \begin{bmatrix} \mathbf{f}_{P,d}(\mathbf{x}_P(t), \boldsymbol{\alpha}, g_{C,d}(\mathbf{x}_C(t), \boldsymbol{\rho}, r - g_{P,d}(\mathbf{x}_P(t), \boldsymbol{\alpha}, d_{inp}(t))), d_{inp}) \\ \mathbf{f}_{C,d}(\mathbf{x}_C(t), \boldsymbol{\rho}, r - g_P(\mathbf{x}_P(t), \boldsymbol{\alpha}, d_{inp}(t))) \end{bmatrix}$$
$$= \mathbf{f}_d(\mathbf{x}(t), \boldsymbol{\alpha}, \boldsymbol{\rho}, r(t), d_{inp}(t)), \tag{5}$$

$$y(t) = g_{P,d}(\mathbf{x}(t), \boldsymbol{\alpha}, d_{inp}(t)) h_{P,d}(\mathbf{x}(t), \boldsymbol{\alpha}, d_{inp}(t)),$$
$$\mathbf{x}(t_0) = \begin{bmatrix} \mathbf{x}_{P,0} \\ \mathbf{x}_{C,0} \end{bmatrix},$$

where the functions $\mathbf{f}_d : R^{n+p+m+q+2} \to R^{n+p}$ and $h_{P,d} : R^{n+p+m+1} \to R^{n+p}$ are differentiable with respect to the process parameter α_a, $a = 1 \dots m$.

The state sensitivity functions $\lambda_b^{\alpha_a}, b = 1 \dots n+p$, and the output sensitivity function σ^{α_a} are defined as follows:

$$\lambda_b^{\alpha_a} = \left[\frac{\partial x_b}{\partial \alpha_a}\right]_{\alpha_a,0}, \quad \sigma^{\alpha_a} = \left[\frac{\partial y}{\partial \alpha_a}\right]_{\alpha_a,0}, \quad b = 1 \dots n+p, \quad a = 1 \dots m, \tag{6}$$

where the subscript 0 indicates the nominal value of the process parameter α_a, $a = 1 \dots m$, which is subjected to variations and therefore the sensitivity reduction is justified.

The following discrete-time objective functions are defined to ensure the sensitivity reduction with respect to the modifications of α_a, $a = 1 \dots m$:

$$I_{ISE}^{\alpha_a}(\boldsymbol{\rho}) = \sum_{t=0}^{\infty} \{ e^2(t) + (\gamma^{\alpha_a})^2 [\sigma^{\alpha_a}(t)]^2 \}, \quad a = 1 \dots m, \tag{7}$$

where $\gamma^{\alpha_a}, a = 1 \dots m$, are the weighting parameters, all variables in the sum depend on $\boldsymbol{\rho}$ standing for the vector variable of the objective function, and ISE points out the Integral of Squared Error. The objective functions defined in (7) are referred to as extended ISE criteria. In practical control problem solutions the sum in (7) should be truncated such that to capture all transients of the control systems during the time horizon specific to (7). The time horizon should include the moments when the objective functions reach their steady-state values. The upper limit of the sum depends on the dynamics of the particular process under consideration.

The minimization of the objective functions defined in (7) aims the sensitivity reduction, and it is expressed in terms of the optimization problems

$$\boldsymbol{\rho}^* = \arg\min_{\boldsymbol{\rho} \in D_\rho} I_{ISE}^{\alpha_a}(\boldsymbol{\rho}), \quad a = 1 \dots m, \tag{8}$$

where $\boldsymbol{\rho}^*$ is the optimal value of the vector $\boldsymbol{\rho}$, and $D_{\boldsymbol{\rho}}$ is the feasible domain of $\boldsymbol{\rho}$. Several constraints including the stability of the fuzzy control system can be imposed and expressed by means of $D_{\boldsymbol{\rho}}$.

3. Evolutionary optimization algorithms

GSA, PSO and SA are the three evolutionary optimization algorithms employed in the numerical solving of the optimization problems defined in (8). These algorithms are presented briefly in the sequel.

3.1. Gravitational Search Algorithms

The operating mechanism of GSAs makes use of agents (i.e., particles) and of Newton's law of gravity. The depreciation of the gravitational constant with the advance of the GSA's iterations is modeled by

$$g(k) = g_0 \exp(-\zeta k / k_{\max}), \tag{9}$$

where $g(k)$ is the value of the gravitational constant at the current iteration index k, g_0 is the initial value of the gravitational constant, ζ is a preset constant to ensure the GSA's convergence and to influence the search accuracy, and $k_{\max}$ is the maximum number of iterations.

Considering N agents and a q-dimensional search space the position of the ith agent is defined by the vector

$$\mathbf{X}_i = [\, x_i^1 \quad \dots \quad x_i^d \quad \dots \quad x_i^q \,]^T, \quad i = 1 \dots N, \tag{10}$$

where x_i^d is the position of the ith agent in the dth dimension, $d = 1 \dots q$. To ensure the stochastic characteristic of the GSA the total force acting on the ith agent in the dth dimension, referred to as $F_i^d(k)$, is a randomly weighted sum of all forces exerted from the other agents:

$$F_i^d(k) = \sum_{j=1, j \neq i}^{N} \rho_j F_{ij}^d(k), \tag{11}$$

where ρ_j, $0 \leqslant \rho_j \leqslant 1$, is a random generated number, and the force acting on the ith agent from the jth agent is

$$F_{ij}^d(k) = g(k) \frac{m_{Pi}(k) m_{Aj}(k)}{r_{ij}(k) + \varepsilon} [x_j^d(k) - x_i^d(k)], \tag{12}$$

where $m_{Pi}(k)$ is the active gravitational mass related to the ith agent, $m_{Aj}(k)$ is the passive gravitational mass related to the jth agent, $\varepsilon > 0$ is a relatively small constant, and $r_{ij}(k)$ is the Euclidian distance between the ith and the jth agents:

$$r_{ij}(k) = \|\mathbf{X}_i(k) - \mathbf{X}_j(k)\|. \tag{13}$$

The law of motion leads to the acceleration $a_i^d(k)$ of the ith agent at the iteration index k in the dth dimension:

$$a_i^d(k) = F_i^d(k) / m_{Ii}(k), \tag{14}$$

where $m_{Ii}(t)$ is the inertia mass related to the ith agent. The next velocity of an agent, $v_i^d(k+1)$, and the next position of an agent, $x_i^d(k+1)$, are obtained in terms of the state-space equations [44,45]:

$$v_i^d(k+1) = \rho_i v_i^d(k) + a_i^d(k),$$
$$x_i^d(k+1) = x_i^d(k) + v_i^d(k+1), \tag{15}$$

where ρ_i, $0 \leqslant \rho_i \leqslant 1$, is a uniform random variable.

The passive gravitational mass and the inertial masses are obtained by means of

$$n_i(k) = \frac{f_i(k) - w(k)}{b(k) - w(k)},$$
$$m_i(k) = \frac{n_i(k)}{\sum_{j=1}^{N} n_j(k)}, \tag{16}$$
$$m_{Ai} = m_{Ii} = m_i,$$

where $f_i(k)$ is the fitness value of the ith agent at the iteration index k, and the terms $b(k)$ (corresponding to the best agent) and $w(k)$ (corresponding to the worst agent) are

$$b(k) = \min_{j=1 \dots n} f_j(k),$$
$$w(k) = \max_{j=1 \dots n} f_j(k). \tag{17}$$

The relation between the fitness function f and the value $f_i(k)$ in the GSA on the one hand, and the objective functions defined in (7) on the other hand is

$$f_j(k) = I^{\alpha_a}_{ISE}(\boldsymbol{\rho}), \quad a = 1 \ldots m, \quad j = 1 \ldots N \tag{18}$$

and the relation between the agents' position vector $\mathbf{X}_i$ in the GSA and the parameter vector $\boldsymbol{\rho}$ of the fuzzy controller is

$$\mathbf{X}_i = \boldsymbol{\rho}, \quad i = 1 \ldots N. \tag{19}$$

The relationships (18) and (19) map the GSA onto the optimization problems defined in (8). The steps of our GSA are presented in Fig. 2.

The step 1 of the GSA concerns the generation of the initial population of agents, i.e., the initialization of the q-dimensional search space, of the number of agents N, the random initialization of the agents' position vector $\mathbf{X}_i$, and the initialization of the maximum number of iterations $k_{\max}$. The step 6 validates the obtained vector solution $\mathbf{X}_i(k)$ by checking the following inequality-type constraint which guarantees that the fuzzy control system with the obtained fuzzy controller tuning parameters $\boldsymbol{\rho} = \mathbf{X}_i(k)$ ensures the convergence of the objective functions:

$$|y(t_f) - r(t_f)| \leqslant \varepsilon_y |r(t_f) - r(t_0)|, \tag{20}$$

where t_0 is the initial time moment, t_f is the final time moment, and $\varepsilon_y = 0.001$ for a 2% settling time. Theoretically $t_f \to \infty$ as shown in (7), but t_f takes practically a finite value to capture the transients

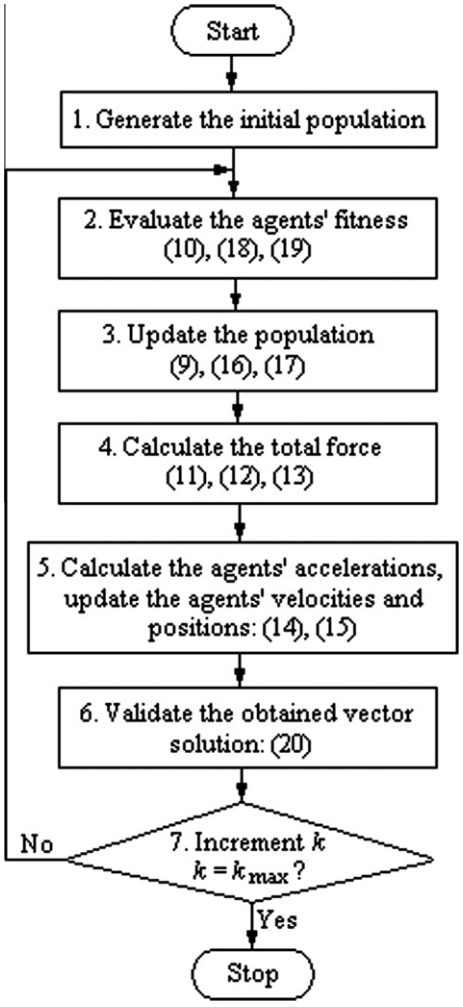

Fig. 2. Flowchart of the GSA.

in the fuzzy control systems' response. The condition (20) guarantees the stability of the fuzzy control systems and it also ensures the zero steady-state control error of the fuzzy control systems.

3.2. Particle Swarm Optimization algorithms

The operating mechanism of PSO algorithms makes use of swarm particles characterized by two vectors $\mathbf{X}_i$, referred to as particle position vector, and $\mathbf{V}_i$, referred to as particle velocity vector:

$$\begin{aligned} &\mathbf{X}_i = [\, x_i^1 \quad \ldots \quad x_i^d \quad \ldots \quad x_i^q \,]^T, \\ &\mathbf{V}_i = [\, v_i^1 \quad \ldots \quad v_i^d \quad \ldots \quad v_i^q \,]^T, \quad i = 1 \ldots N, \end{aligned} \tag{21}$$

where $i, i = 1 \ldots N$, is the index of the current particle in the swarm, and N is the number of particles in the swarm. Let $\mathbf{P}_{i,Best}$ be the best particle position vector of a specific particle with the index $i, i = 1 \ldots N$, and $\mathbf{P}_{g,Best}$ be the best swarm position vector:

$$\begin{aligned} &\mathbf{P}_{i,Best} = [\, p_i^1 \quad \ldots \quad p_i^d \quad \ldots \quad p_i^q \,]^T, \\ &\mathbf{P}_{g,Best} = [\, p_g^1 \quad \ldots \quad p_g^d \quad \ldots \quad p_g^q \,]^T, \quad i = 1 \ldots N. \end{aligned} \tag{22}$$

The next particle velocity, $v_i^d(k+1)$, and the next particle position, $x_i^d(k+1)$, are obtained in terms of the state-space equations processed from [26,27]:

$$\begin{aligned} &v_i^d(k+1) = w(k) v_i^d(k) + c_1 r_1 [p_g^d(k) - x_i^d(k)] + c_2 r_2 [p_i^d(k) - x_i^d(k)], \\ &x_i^d(k+1) = x_i^d(k) + v_i^d(k+1), \quad d = 1 \ldots q, \quad i = 1 \ldots N, \end{aligned} \tag{23}$$

where r_1 and r_2 are uniformly distributed random variables, $0 \leqslant r_1 \leqslant 1$, $0 \leqslant r_2 \leqslant 1$, w is the inertia weight, c_1 and c_2 are the weighting factors of the stochastic accelerations pulling the particles towards their final positions, and k is the current iteration index. The parameter $w(k)$ points out the effect of the previous velocity vector on the new one, and an upper limit $w_{\max}$ is imposed to all velocities to prevent the particles from moving too rapidly in the search space:

$$w(k) = w_{\max} - k[(w_{\max} - w_{\min})/k_{\max}]. \tag{24}$$

If $w_{\max}$ is too small, particles may not explore sufficiently beyond local solutions. The minimum value of $w(k)$ in (24) is $w_{\min}$.

Adopting low values of c_1 and c_2 will allow particles to roam far from the target regions before being tugged back; contrarily, too high values of the weighting factors c_1 and c_2 will result in abrupt movements towards or overshooting the target regions. The steps of our PSO algorithm are presented in Fig. 3.

The notation g, $g: R^q \to R$, is used for the fitness function in the PSO algorithm. The relation between the value of the fitness function, $g(\mathbf{X}_i)$, in the PSO algorithm on the one hand, and the objective functions defined in (7) on the other hand is similar to (18):

$$g(\mathbf{X}_i) = I^{\alpha_a}_{ISE}(\boldsymbol{\rho}), \quad a = 1 \ldots m, \quad i = 1 \ldots N \tag{25}$$

and the relation between the particle position vector $\mathbf{X}_i$ in the PSO algorithm and the parameter vector $\boldsymbol{\rho}$ of the fuzzy controller is (19). The step 1 of the PSO algorithm is related to the generation of the initial population of particles, i.e.:

- The initialization of the q-dimensional search space by setting its boundaries.
- The initialization of the maximum number of iterations $k_{\max}$.
- The initialization of the parameters in (23).
- The initialization of the particle position vector with a uniformly distributed random vector $\mathbf{X}_i$, $i = 1 \ldots N$, that belongs to the search space, and the initialization of the particle velocity vector $\mathbf{V}_i$, $i = 1 \ldots N$, accounting for the boundaries of the search space.
- The initialization of the best particle position vector $\mathbf{P}_{i,Best} = \mathbf{X}_i$, $i = 1 \ldots N$, and of the best swarm position vector $\mathbf{P}_{g,Best}$.

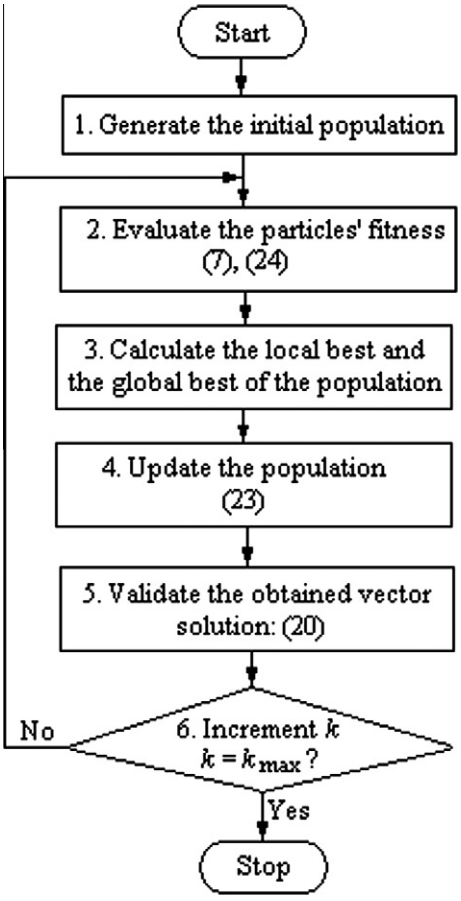

Fig. 3. Flowchart of the PSO algorithm.

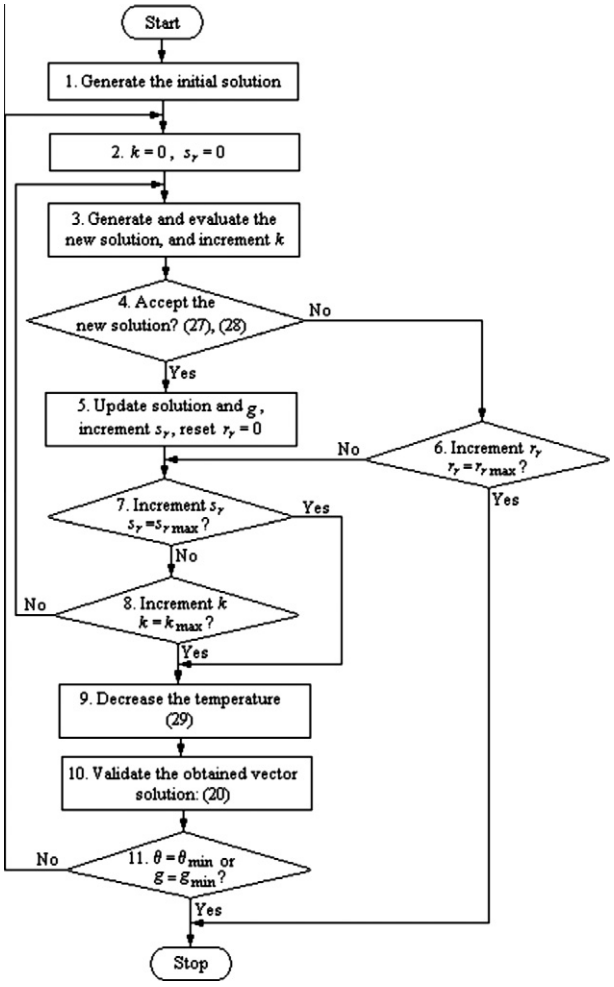

Fig. 4. Flowchart of the SA algorithm.

– If $g(\mathbf{P}_{i,Best}) < g(\mathbf{P}_{g,Best})$, $i = 1 \ldots N$, update the best swarm position vector $\mathbf{P}_{g,Best} = \mathbf{P}_{i,Best}$.

The step 3 of the PSO algorithm consists of the following calculations:

– If $g(\mathbf{X}_i) < g(\mathbf{P}_{i,Best})$, $i = 1 \ldots N$, update the best particle position, $\mathbf{P}_{i,Best} = \mathbf{X}_i$, and if $g(\mathbf{P}_{i,Best}) < g(\mathbf{P}_{g,Best})$, $i = 1 \ldots N$, update the best swarm position vector, $\mathbf{P}_{g,Best} = \mathbf{P}_{i,Best}$.

The step 5 of the PSO algorithm is similar to the step 6 of the GSA, and the best found solution will be the solution to one of the optimization problems defined in (8):

$$\boldsymbol{\rho}^* = \mathbf{P}_{g,Best}. \tag{26}$$

3.3. Simulated Annealing algorithms

The operating mechanism of SA algorithms is based on a probabilistic framework involved in the acceptance of the solution using the analogy with the temperature decrease in metallurgy [15,25]. Considering the initial solution represented by the vector $\varphi \in R^q$ with the fitness value $g(\varphi)$ of the fitness function g, $g: R^q \to R$, the new probable solution is represented by the vector $\boldsymbol{\psi} \in R^q$ that is chosen such that to belong to the vicinity of φ. The steps of our SA algorithm are presented in Fig. 4.

In order to make the SA algorithm more computationally efficient two additional iteration indices are introduced, viz. the success rate s_r and with the rejection rate r_r. The success rate s_r aims the acceleration of the cooling process by forcing a jump in temperature when the minimum value of the objective function changes for a preset number of times at the same temperature level. The rejection rate r_r is introduced as an alternative indicator for the convergence of the algorithm, and it is reset only when small values of the objective function are found and not when the temperature cools.

The step 1 of the SA algorithm is related to the generation of the initial solution, i.e.:

– The random generation of the initial solution φ and the calculation of its fitness value $g(\varphi)$.
– The setting of the minimum temperature $\theta_{\min}$.
– The initialization of the maximum number of iterations $k_{\max}$, of the maximum accepted success rate $r_{r\max}$ of the maximum accepted rejection rate $r_{r\max}$, and of the minimum accepted value of the objective function expressed as the minimum fitness value $g_{\min}$.
– The setting of the initial temperature $\theta(k)$, where k is the current iteration index.
– The initial setting of the rejection rate to $r_r = 0$.

The step 3 of the SA algorithm concerns the random generation of the new probable solution $\boldsymbol{\psi}$ in the vicinity of φ in terms of disturbing φ. The step 4 of the SA algorithm consists of the following calculations:

– Calculate the difference of the fitness values referred to as $\Delta g_{\varphi\boldsymbol{\psi}}$:

$$\Delta g_{\varphi\boldsymbol{\psi}} = g(\varphi) - g(\boldsymbol{\psi}). \tag{27}$$

– If $\Delta g_{\varphi\psi} \leqslant 0$, then accept ψ as the new solution. Otherwise, select randomly r_n, $0 \leqslant r_n \leqslant 1$, and calculate the probability of ψ being the new solution, referred to as p_ψ:

$$p_\psi = \begin{cases} 1 & \text{if } \Delta g_{\varphi\psi} \leqslant 0, \\ \exp[-\Delta g_{\varphi\psi}/\theta(k)] & \text{otherwise}, \end{cases} \tag{28}$$

and if $p_\psi > r_n$, then ψ is the new solution.

The temperature decrease in the step 9 of the SA algorithm is carried out according to [38]

$$\theta(k+1) = \alpha_{cs}\theta(k), \tag{29}$$

where $\alpha_{cs} = \text{const}, \alpha < 1, \alpha \approx 1$.

The step 10 of our SA algorithm is similar to the step 6 of the GSA and to the step 5 of the PSO algorithm, and the last new solution found will be the solution to one of the optimization problems defined in (8):

$$\rho^* = \psi. \tag{30}$$

The relations between the fitness values $g(\varphi)$ and $g(\psi)$ in the SA algorithm on the one hand, and the values of the objective function defined in (7) on the other hand are similar to (18) and (25):

$$g(\varphi) = I^{\alpha_a}_{ISE}(\rho), \quad g(\psi) = I^{\alpha_a}_{ISE}(\rho), \quad a = 1 \ldots m, \tag{31}$$

and the relation between the solutions φ and ψ in the SA algorithm and the parameter vector ρ of the fuzzy controller are similar to (19):

$$\varphi = \rho, \psi = \rho. \tag{32}$$

Our three evolutionary algorithms will be applied in the next section to obtain low-cost optimal T–S PI-FCs for a class of nonlinear servo systems such that to ensure a reduced process gain sensitivity.

4. Case study

A class of servo systems used in position control is considered, where the process is described by the nonlinear continuous-time state-space model

$$m(t) = \begin{cases} 0, & \text{if } |u(t)| \leqslant u_a, \\ k_{u,m}(u(t) - u_a sgn(u(t))), & \text{if } u_a < |u(t)| < u_b, \\ k_{u,m}(u_b - u_a) sgn(u(t)), & \text{if } |u(t)| \geqslant u_b, \end{cases}$$

$$\dot{\mathbf{x}}_P(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1/T_\Sigma \end{bmatrix} \mathbf{x}_P(t) + \begin{bmatrix} 0 \\ k_{P1}/T_\Sigma \end{bmatrix} m(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} d_{iinp}(t), \tag{33}$$

$$y(t) = [1 \quad 0]\mathbf{x}_P(t),$$

where t is the independent continuous time argument, $t \in R$, $t \geqslant 0$, the control signal u is a pulse width modulation duty cycle, m is the output of the saturation and dead zone static nonlinearity with the parameters $k_{u,m} > 0, u_a, u_b, 0 < u_a < u_b$ represented by the first equation in (33). The process structure is illustrated in Fig. 5, and it includes the actuator and measuring element dynamics. The state vector $\mathbf{x}_P(t)$ is expressed as follows in angular position applications:

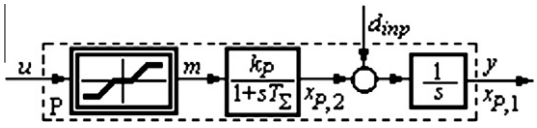

Fig. 5. Structure of the process.

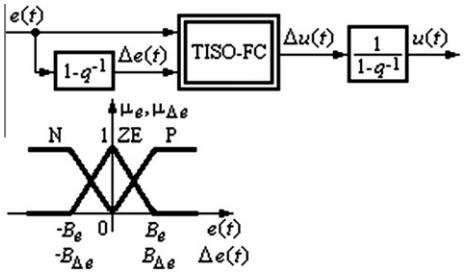

Fig. 6. Structure and input membership functions of the Takagi–Sugeno PI-fuzzy controller.

$$\mathbf{x}_P(t) = [x_{P,1}(t) = \alpha(t) \quad x_{P,2}(t) = \omega(t)]^T, \tag{34}$$

where $x_{P,1}(t) = \alpha(t)$ is the first state variable that represents the angular position, and $x_{P,2}(t) = \omega(t)$ is the second state variable that represents the angular speed.

The process has an input nonlinearity concerning the actuator, i.e., a saturation and dead zone static nonlinearity. However, this is not included in the following simplified model of the process expressed as the transfer function $P(s)$ used in the tuning of low-cost fuzzy controllers:

$$P(s) = \frac{k_P}{s(1 + T_\Sigma s)}, \tag{35}$$

where k_P is the process gain, $k_P = k_{u,m}k_{P1}$, and T_Σ is the small time constant. The models presented in (33) and (35) are used in servo systems in many applications [14,16,19,22,24,47,50].

As shown in [42], PI controllers can cope with the process modeled in (33) and (35), and the PI controllers can be tuned by the Extended Symmetrical Optimum (ESO) method to guarantee a good trade-off to the desired/ imposed control performance indices (overshoot, settling time, rise time, etc.) using a single design parameter referred to as β. The parameters of the PI controllers are next mapped onto the parameters of the T–S PI-FCs (Fig. 6) in terms of the modal equivalence principle to improve these control system performance indices.

More membership functions can be defined but they lead to complicated rule base. One solution to deal with such situations in order to design low-cost fuzzy controllers is represented by fuzzy rule interpolation [3,4,23,52].

The Two Inputs-Single Output fuzzy controller (TISO-FC) block introduced in Fig. 6 is characterized by the weighted average method in the defuzzification module, and by the SUM and PROD operators in the inference engine. The dynamics of the Takagi–Sugeno PI-FC is ensured by the two linear blocks in Fig. 6. The rule base of the TISO-FC block is formulated as the decision table presented in Table 1, and the consequents of the rules are modeled by

$$f_1(t) = K_P[\Delta e(t) + \mu e(t)], \quad f_2(t) = \eta f_1(t). \tag{37}$$

The parameters in (37) are obtained by the application of Tustin's method to discretize the linear PI controller with the transfer function

$$C(s) = k_c(1 + sT_i)/s = k_C[1 + 1/(sT_i)], \quad k_C = k_c T_i, \tag{38}$$

Table 1
Decision table of the TISO-FC block.

$\Delta e(t)$	$e(t)$		
	N	ZE	P
P	$\Delta u(t) = f_1(t)$	$\Delta u(t) = f_1(t)$	$\Delta u(t) = f_2(t)$
ZE	$\Delta u(t) = f_1(t)$	$\Delta u(t) = f_1(t)$	$\Delta u(t) = f_1(t)$
N	$\Delta u(t) = f_2(t)$	$\Delta u(t) = f_1(t)$	$\Delta u(t) = f_1(t)$

where k_c is the controller gain and T_i is the integral time constant. Tustin's method leads to

$$K_P = k_c(T_i - T_s/2), \quad \mu = 2T_s/(2T_i - T_s), \tag{39}$$

where T_s is the sampling period.

The parameter η is introduced in (37) to alleviate the overshoot of the fuzzy control system when $e(t)$ and $\Delta e(t)$ have the same signs. The modal equivalence principle results in the tuning equation

$$B_{\Delta e} = \mu B_e. \tag{40}$$

The PI tuning conditions specific to the ESO method are

$$k_c = 1/(\beta\sqrt{\beta}T_\Sigma^2 k_P), \quad T_i = \beta T_\Sigma. \tag{41}$$

The parameter vector of the controller $\boldsymbol{\rho}$ ($q = 3$) and the parameter vector of the process $\boldsymbol{\alpha}$ ($m = 2$) obtain the following particular expressions if the evolutionary algorithms described in the previous section are employed in the numerical solving of the optimization problems defined in (8):

$$\begin{aligned} &\boldsymbol{\rho} = [\,\rho_1 = \beta \quad \rho_2 = B_e \quad \rho_3 = \eta\,]^T \in R^3, \\ &\boldsymbol{\alpha} = [\,\alpha_1 = k_P \quad \alpha_2 = T_\Sigma\,]^T \in R^2. \end{aligned} \tag{42}$$

Our three evolutionary algorithms are employed to solve the following optimization problem that ensures the sensitivity reduction of the fuzzy control systems with respect to the process parameter $\alpha_1 = k_P$, i.e., a reduced process gain sensitivity:

$$\boldsymbol{\rho}^* = \arg\min_{\boldsymbol{\rho}\in Do} I_{ISE}^{k_P}(\boldsymbol{\rho}), \quad I_{ISE}^{k_P}(\boldsymbol{\rho}) = \sum_{t=0}^{\infty}\{e^2(t) + (\gamma^{k_P})^2[\sigma^{k_P}(t)]^2\}, \tag{43}$$

where γ^{k_P} is the weighting parameter, and σ^{k_P} is the output sensitivity function. The calculation of σ^{k_P} results from the state sensitivity model of the fuzzy control system derived as follows.

Accepting that the inputs u and d_{inp} are changing at the discrete sampling intervals, the following discrete-time state-space model of the process is obtained ($n = 2$):

$$m(t) = \begin{cases} 0, & \text{if } |u(t)| \leqslant u_a, \\ k_{u,m}(u(t) - u_a sgn(u(t))), & \text{if } u_a < |u(t)| < u_b, \\ k_{u,m}(u_b - u_a)sgn(u(t)), & \text{if } |u(t)| \geqslant u_b, \end{cases} \tag{44}$$

$$\begin{aligned} x_{P,1}(t+1) &= x_{P,1}(t) + T_\Sigma[1 - \exp(-T_s/T_\Sigma)]x_{P,2}(t) \\ &\quad + k_{P1}[T_s + T_\Sigma\exp(-T_s/T_\Sigma) - T_\Sigma]m(t) + T_s d_{inp}(t), \\ x_{P,2}(t+1) &= [\exp(-T_s/T_\Sigma)]x_{P,2}(t) + k_{P1}[1 - \exp(-T_s/T_\Sigma)]m(t), \\ y(t) &= x_{P,1}(t), \end{aligned}$$

where $t \in N$. The discretization of the process model given in (33) is conducted such that the models in (33) and (44) should exhibit the same response at the discrete time moments defined by the discrete sampling intervals. The choice of T_s depends on the time constant (s) of the process, and it should fulfill the conditions of quasi-continuous digital control [20].

The state variables $x_{C,1}$ and $x_{C,2}$ are next defined for the T–S PI-FC:

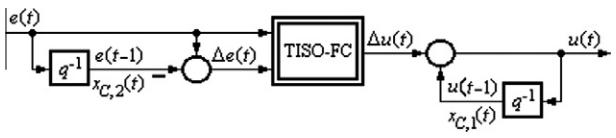

Fig. 7. Modified structure of the Takagi–Sugeno PI-fuzzy controller such that to include the state variables.

$$\begin{aligned} x_{C,1}(t) &= u(t-1), \\ x_{C,2}(t) &= e(t-1), \end{aligned} \tag{45}$$

where the two state variables ($p = 2$ according to Section 2) are highlighted in Fig. 7. Fig. 6 and (45) result in the following discrete-time state-space model of the T–S PI-FC:

$$\begin{aligned} x_{C,1}(t+1) &= x_{C,1}(t) + f_{TISO-FC}(e(t), e(t) - x_{C,2}(t)), \\ x_{C,2}(t+1) &= e(t), \\ u(t) &= x_{C,1}(t) + f_{TISO-FC}(e(t), e(t) - x_{C,2}(t)), \end{aligned} \tag{46}$$

where

$$\begin{aligned} &f_{TISO-FC} : R^2 \to R, \\ &f_{TISO-FC}(e(t), \Delta e(t)) = f_{TISO-FC}(r(t) - x_{P,1}(t), r(t) - x_{P,1}(t) - x_{C,2}(t)), \end{aligned} \tag{47}$$

is the nonlinear input–output map of the TISO-FC block, and the notation $f_{TISO-FC}(t)$ is also used to simplify the presentation.

The discrete-time state-space models of the process and of the fuzzy controller expressed in (44) and (46) are next merged as shown in Section 2, and the discrete-time state-space model of the fuzzy control system is

$$\begin{aligned} x_1(t+1) &= x_1(t) + T_\Sigma[1 - \exp(-T_s/T_\Sigma)]x_2(t) + T_s d_{inp}(t) \\ &+ \begin{cases} 0, & \text{if } |u(t)| \leqslant u_a, \\ k_P[T_s + T_\Sigma\exp(-T_s/T_\Sigma) - T_\Sigma]x_3(t) + k_P[T_s + T_\Sigma\exp(-T_s/T_\Sigma) \\ -T_\Sigma]f_{TISO-FC}(r(t) - x_1(t), r(t) - x_1(t) - x_4(t)) - k_P[T_s & \text{if } u_a < |u(t)| < u_b, \\ +T_\Sigma\exp(-T_s/T_\Sigma) - T_\Sigma]u_a sgn(u(t)), \\ k_P[T_s + T_\Sigma\exp(-T_s/T_\Sigma) - T_\Sigma](u_b - u_a)sgn(u(t)), & \text{if } |u(t)| \geqslant u_b, \end{cases} \end{aligned}$$

$$\begin{aligned} x_2(t+1) &= [\exp(-T_s/T_\Sigma)]x_2(t) \\ &+ \begin{cases} 0, & \text{if } |u(t)| \leqslant u_a, \\ k_P[1 - \exp(-T_s/T_\Sigma)]x_3(t) + k_P[1 - \exp(-T_s/T_\Sigma)] \\ \cdot f_{TISO-FC}(r(t) - x_1(t), r(t) - x_1(t) - x_4(t)) - k_P[1 & \text{if } u_a < |u(t)| < u_b, \\ -\exp(-T_s/T_\Sigma)]u_a sgn(u(t)), \\ k_P[1 - \exp(-T_s/T_\Sigma)](u_b - u_a)sgn(u(t)), & \text{if } |u(t)| \geqslant u_b, \end{cases} \end{aligned}$$

$$\begin{aligned} x_3(t+1) &= x_3(t) + f_{TISO-FC}(r(t) - x_1(t),\ r(t) - x_1(t) - x_4(t)) + T_s d_{inp}(t), \\ x_4(t+1) &= r(t) - x_1(t), \\ y(t) &= x_1(t), \end{aligned} \tag{48}$$

where the state variables x_1, x_2, x_3 and x_4 are the elements of the state vector $\mathbf{x}$ of the fuzzy control system,

$$\mathbf{x} = [\,x_1 = x_{P,1} \quad x_2 = x_{P,2} \quad x_3 = x_{C,1} \quad x_4 = x_{C,2}\,]^T \in R^4. \tag{49}$$

Accepting constant inputs of the fuzzy control system, $r(t) = \text{const}$ and $d_{inp}(t) = \text{const}$, the state sensitivity models of the fuzzy control system with respect to the parameter k_P is derived using the definition (6) in the model (48) for

$$\alpha_1 = k_P, m = 1. \tag{50}$$

The state sensitivity model of the fuzzy control system is obtained as follows by the differentiation of the right-hand terms of all equations in (48) with respect to $\alpha_1 = k_P$ using the definitions of the state and output sensitivity functions given in (6):

$$\begin{aligned} \lambda_1^{k_P}(t+1) &= \lambda_1^{k_P}(t) + T_{\Sigma,0}[1 - \exp(-T_s/T_{\Sigma,0})]\lambda_2^{k_P}(t) \\ &+ \begin{cases} 0, & \text{if } |u(t)| \leqslant u_a, \\ [T_s + T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) - T_{\Sigma,0}]x_{3,0}(t) + k_{P,0}[T_s \\ +T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) - T_{\Sigma,0}]\lambda_3^{k_P}(t) + [T_s + T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) \\ -T_{\Sigma,0}]f_{TISO-FC,0}(t) - k_{P,0}[T_s + T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) \\ -T_{\Sigma,0}]\left[\frac{\partial f_{TISO-FC,0}(t)}{\partial e(t)}\right]_0\lambda_1^{k_P}(t) - k_{P,0}[T_s + T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) & \text{if } u_a < |u(t)| < u_b, \\ -T_{\Sigma,0}]\left[\frac{\partial f_{TISO-FC,0}(t)}{\partial \Delta e(t)}\right]_0[\lambda_1^{k_P}(t) + \lambda_4^{k_P}(t)] - [T_s + T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) \\ -T_{\Sigma,0}]u_a sgn(u(t)), \\ [T_s + T_{\Sigma,0}\exp(-T_s/T_{\Sigma,0}) - T_{\Sigma,0}](u_b - u_a)sgn(u(t)), & \text{if } |u(t)| \geqslant u_b, \end{cases} \end{aligned}$$

$$\lambda_2^{k_P}(t+1)=\exp(-T_s/T_{\Sigma,0})\lambda_2^{k_P}(t)$$
$$+\begin{cases}0, & \text{if } |u(t)|\leqslant u_a,\\ [1-\exp(-T_s/T_{\Sigma,0})]x_{3,0}(t)+k_{P,0}[1-\exp(-T_s/T_{\Sigma,0})]\lambda_3^{k_P}(t) & \\ +[1-\exp(-T_s/T_{\Sigma,0})]f_{TISO-FC,0}(t)-k_{P,0}[1 & \\ -\exp(-T_s/T_{\Sigma,0})]\left[\frac{\partial f_{TISO-FC,0}(t)}{\partial e(t)}\right]_0\lambda_1^{k_P}(t)-k_{P,0}[1 & \text{if } u_a<|u(t)|<u_b,\\ -\exp(-T_s/T_{\Sigma,0})]\left[\frac{\partial f_{TISO-FC,0}(t)}{\partial \Delta e(t)}\right]_0[\lambda_1^{k_P}(t)+\lambda_4^{k_P}(t)] & \\ -[1-\exp(-T_s/T_{\Sigma,0})]u_a sgn(u(t)), & \\ [1-\exp(-T_s/T_{\Sigma,0})](u_b-u_a)sgn(u(t)), & \text{if } |u(t)|\geqslant u_b,\end{cases}$$

$$\lambda_3^{k_P}(t+1)=\lambda_3^{k_P}(t)-\left[\frac{\partial f_{TISO-FC,0}(t)}{\partial e(t)}\right]_0\lambda_1^{k_P}(t)-\left[\frac{\partial f_{TISO-FC,0}(t)}{\partial \Delta e(t)}\right]_0\lambda_4^{k_P}(t), \quad (51)$$

$$\lambda_4^{k_P}(t+1)=-\lambda_1^{k_P}(t),$$

$$\sigma^{k_P}(t)=\lambda_1^{k_P}(t),$$

where the subscript 0 indicates both the nominal value of process parameter and the nominal trajectory of the fuzzy control system, i.e., the trajectory considered for the nominal parameters of the process.

The parameters of the T–S PI-FCs are obtained in terms of the following unified tuning approach that consists of the steps A to D:

Step A. Set the sampling period T_s and derive the state sensitivity model (51) which is based on a linear tuning method and on the modal equivalence principle.

Step B. Set the weighting parameter γ^{k_P} in the objective function $I_{ISE}^{k_P}$ defined in (43) to meet the performance specifications of the fuzzy control systems. Set the upper limit of the sum in (43) that replaces infinity such that the finite time horizon includes all transients of the fuzzy control systems until $I_{ISE}^{k_P}$ reaches the steady-state values.

Step C. Apply the of the GSA, of the PSO algorithm or of the SA algorithm presented in the previous section to solve the optimization problem defined in (43) that leads to the optimal parameter vector of the fuzzy controller, $\boldsymbol{\rho}^*$:

$$\boldsymbol{\rho}^*=[\,\rho_1^*=\beta^*\quad \rho_2^*=B_e^*\quad \rho_3^*=\eta^*\,]^T. \quad (52)$$

Step D. Apply Eqs. (40) and (52) to obtain the tuning parameters of the T–S PI-FC, $B_e^*, B_{\Delta e}^*$ and η^*.

This four-step tuning approach is applied to all three evolutionary optimization algorithms considered in this paper, i.e., GSA, PSO and SA. It is sufficiently general as it can employ other optimization algorithms.

The implementation of low-cost fuzzy controllers is targeted by the application of the ESO method and of the modal equivalence principle which maps the parameters of the PI controller onto the parameters of the PI-fuzzy one. Therefore a new generation of low-cost optimal T–S PI-FCs is offered. The application of our tuning approach is exemplified as follows.

5. Results and discussion

The application of our unified tuning approach is carried out to ensure the fuzzy control of the angular position of the experimental setup built around the INTECO DC servo system laboratory equipment (Fig. 8). The setup is characterized by rated amplitude of 24 V, rated current of 3.1 A, rated torque of 15 N cm, rated speed of 3000 rpm, and weight of inertial load of 2.03 kg. The parameters of the linear dynamics of the process model in (33) are the gain $k_{P1}=121.6956$ and the small time constant $T_\Sigma=0.9198$ s. The parameters of the saturation and dead zone static nonlinearity in (33) are identified by nonlinear least squares as $k_{u,m}=1.149$, $u_a=0.13$ and $u_b=1$. Therefore the process gain is $k_P=k_{u,m}k_{P1}=139.88$.

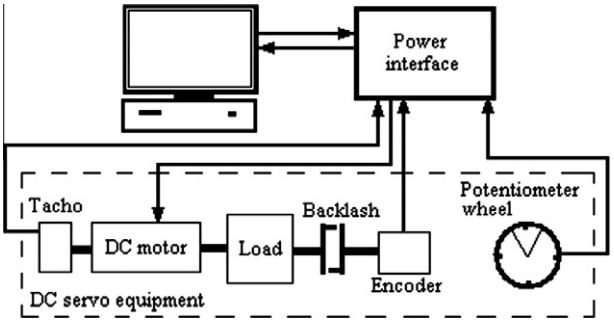

Fig. 8. Structure of the experimental setup.

The steps A, B, C and D of the tuning approach proposed in Section 4 are applied as follows. They start with setting the sampling period to $T_s=0.01$ s such that to fulfill the conditions of quasi-continuous digital control.

The transfer function of the process given in (35) is used in the application of the linear tuning method in the step A of our tuning approach which enables the reduction of the number of elements in the vector variable $\boldsymbol{\rho}$ of the optimization problems, viz. the number of tuning parameters of the T–S PI-FCs. The detailed process model (33) is used in the Matlab & Simulink-based simulations conducted in the step C of the tuning approach where the fitness function values are calculated in the framework of the GSA, of the PSO algorithm and of the SA algorithm. Our three evolutionary algorithms do not require the calculations of the partial derivatives of the nonlinear input–output map $f_{TISO-FC}$ of the TISO-FC block because they are calculated in the simulations conducted in the step C of the tuning approach. These simulations are based on the introduction of the reference filter with the transfer function [42]

$$F(s)=1/(1+\beta T_\Sigma s), \quad (54)$$

to improve the fuzzy control system performance indices. The average value of the design parameter β was taken into account, $\beta=7$. The fuzzy control system becomes thus a two-degree-of-freedom fuzzy control system structure, the filter can be complicated and its parameters can be included in the vector variable $\boldsymbol{\rho}$ of the optimization problems.

The step B of the tuning approach employs the calculation of the weighting parameter for the objective function (i.e., the fitness functions in the evolutionary algorithms) $I_{ISE}^{k_P}$ defined in (43) in order for the ratio of the initial values of the two sums in (43) to take the values {0.1,1,10}. In addition the cases with zero weighting parameters were analyzed. The upper limit of the sum in (43) was wet to 150 s in our simulations.

The step C of the tuning approach uses the digital simulation of the fuzzy control system behavior with respect to the step-type modification of the reference input r to evaluate the fitness function $I_{ISE}^{k_P}$. A part of the results obtained for $r=40$ rad and no disturbance input will be exemplified as follows.

The parameters of the three evolutionary algorithms are initialized taking into consideration the following boundaries which define the domain $D_{\boldsymbol{\rho}}$:

$$D_{\boldsymbol{\rho}}=\{\beta|3\leqslant\beta\leqslant 17\}\times\{B_e|20\leqslant B_e\leqslant 40\}\times\{\eta|0.55\leqslant\eta\leqslant 1\}. \quad (55)$$

Other stability conditions imposed to the fuzzy control system can be added to make stronger the inequality-type constraints defined

Table 2
Results for the GSA-based minimization of $I_{ISE}^{k_P}$.

$(\gamma^{k_P})^2$	$B_{\Delta e}^*$	B_e^*	η^*	β^*	$k_c^*\ T_i^*$		$I_{ISE\,\min}^{k_P}$
0	0.066726	20	1	3.26339	0.004302	3.00232	436071
1129.5	0.102233	30.3089	1	3.22792	0.004325	2.96969	480680
11295	0.073425	20.2706	1	3.0062	0.004482	2.76571	875441
112950	0.077628	21.4454	1	3.00825	0.00448	2.76759	4780930

Table 3
Results for the PSO algorithm-based minimization of $I_{ISE}^{k_P}$.

$(\gamma^{k_P})^2$	$B_{\Delta e}^*$	B_e^*	η^*	β^*	$k_c^*\ T_i^*$		$I_{ISE\,\min}^{k_P}$
0	0.096608	28.9388	1	3.26141	0.004303	3.00049	436068
1129.5	0.067616	20	1	3.22051	0.00433	2.96287	480674
11295	0.110882	30.6078	1	3.00587	0.004482	2.7654	875415
112950	0.144907	40	1	3.00587	0.004482	2.7654	4779430

Table 4
Results for the SA algorithm-based minimization of $I_{ISE}^{k_P}$.

$(\gamma^{k_P})^2$	$B_{\Delta e}^*$	B_e^*	η^*	β^*	$k_c^*\ T_i^*$		$I_{ISE\,\min}^{k_P}$
0	0.133453	39.9756	1	3.26141	0.004303	3.0005	436068
1129.5	0.126662	37.4652	1	3.22052	0.00433	2.96288	480674
11295	0.120132	33.6636	1	3.05132	0.004448	2.80721	875876
112950	0.144906	39.9999	0.999999	3.00587	0.004482	2.7654	4779430

in Eq. (55) but they are not necessary because the condition defined in (20) guarantees the stability of the fuzzy control system.

The objective function depends on three variables, β, B_e and η as shown in (42), so the parameter vector of the controller of all optimization algorithms is a $d = 3$-dimensional vector. The values of the weighting parameters in the optimization problem defined in (43) were set to $(\gamma^{k_P})^2 \in \{0, 1129.5, 11295, 112950\}$ such that to ensure a ratio of 0, 0.1, 1 and 10 of the first and second term in the sum defined in (43).

In order to ensure a good convergence of the GSA the parameters were set to $N = 20$, $k_{\max} = 200$, $\zeta = 20$, $\varepsilon = 0.0001$, and $g_0 = 100$. These values of the parameters of the GSA are derived on the basis of the study conducted in [37]. The optimal values of the controller tuning parameters obtained with our GSA are synthesized in Table 2.

In the case of the PSO algorithm the parameters were set to $N = 20$, $k_{\max} = 200$, $c_1 = c_2 = 1.2$, $w_{\min} = 0.4$ and $w_{\max} = 0.9$ such that to ensure also a good tradeoff to PSO algorithm's convergence speed and accuracy as proved in [13]. The optimal values of the controller tuning parameters obtained with our PSO algorithm are presented in Table 3.

The parameters of our SA algorithm were set to $\theta_{\min} = 10^{-8}$, $g_{\min} = 10^{-4}$, $\alpha_{cs} = 0.9$, $s_{r\max} = 50$, and $r_{r\max} = 1000$ in order to obtain a slow cooling rate and thus a good convergence; these are the values of the parameters obtained in accordance with the analysis carried out in [38]. The optimal values of the controller tuning parameters obtained with our SA algorithm are presented in Table 4.

The performance index referred to as convergence speed (c_s) is introduced to compare our evolutionary optimization algorithms. The convergence speed is represented by the number of evaluations of the objective functions until finding its minimum value. Table 5 highlights average values of this performance index calculated for the best five runs in case of each algorithm.

Tables 2–4 show that all three evolutionary algorithms lead to the same values of the tuning parameters and of the minimum values of the objective function for the same values of the weighting parameter γ^{k_P}. This result is important as it shows that although it is not guaranteed that the optimal solutions were obtained, since the solutions obtained through three different algorithms are very close, it can be accepted that our solutions are very close to the optimal ones.

Table 5
Values of convergence speed.

$(\gamma^{k_P})^2$	c_s for the GSA	c_s for the PSO algorithm	c_s for the SA algorithm
0	418.6	1964.6	8608.6
1129.5	511	1949.6	7732.8
11295	582.4	1882.4	3773.8
112950	265.2	1711.2	4020.2

The fourth column in Tables 2–4 points out that the implementation of our evolutionary optimization algorithms in combination with the ESO method results practically in the same rule consequent for all rules of the T–S PI-FCs. This result is explained by the fact that the ESO method represents in fact an advantageous constraint that reduces the number of variables of the objective functions. Dropping out this constraint will complicate the implementation of our algorithms and will increase the cost of implementation but it can result in smaller values of the objective functions with respect to those illustrated in the last column of Tables 2–4.

The results presented in Table 5 outline that the GSA is two to four times quicker compared to the PSO algorithm and fifteen to twenty times quicker than the SA algorithm.

Once the optimal tuning parameters of the T–S PI-FCs are obtained in terms of Matlab & Simulink-based digital simulations conducted with the fuzzy control system to evaluate the objective functions, they are next mapped onto the parameters of the T–S PI-FCs implemented for the real-world servo system. This requires that very accurate process models are needed. If no accurate process models are available our evolutionary optimization algorithms are not applicable because the solutions will be slightly

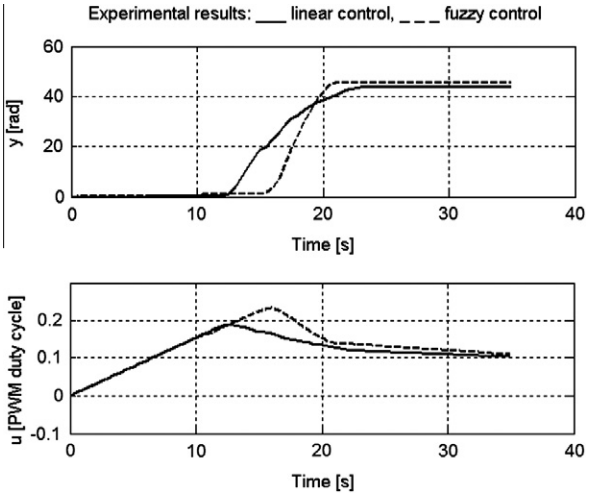

Fig. 9. Real-time experimental results: controlled output and control signal of the control system with the PI controller (dashed line) and of the control system with the T–S PI-FC (solid line) for the parameters in the second line of Tables 2–4.

different to the optimal ones. However our sensitivity reduction approach aims the alleviation of these differences. Several real-time experiments were conducted to illustrate this aspect, and a part of the real-time experimental results will be presented as follows. The presentation of the real-time experimental results is organized in terms of plotting the evolutions of the control signal u and of the controlled output y versus time, and of evaluating the objective functions for the control systems on the real-world servo system represented by our experimental setup.

The experimental results were obtained for the step-type angular position reference input of $r = 40$ rad. The experiments were conducted for the control systems with both the PI controller and the T–S PI-FCs. A sample of the real-time experimental that correspond to the control systems with the controllers and the parameters given in the second line of Tables 2–4 is presented in Fig. 9. The objective function measured for the fuzzy control system (i.e., the control system with the T–S PI-FC) is $I_{ISE}^{K_P} = 506502$.

6. Conclusions

This paper has proposed a novel evolutionary algorithm-based tuning approach of fuzzy controllers for servo systems that ensures the sensitivity reduction with respect to the parametric variations of the process models. This approach is important as a new generation of low-cost optimal T–S PI-FCs is offered; this generation is characterized by simplified process models in the controller tuning and by simple fuzzy controller structures. Therefore the sensitivity reduction is justified and proved here by the reduction of the objective functions that include the weighted output sensitivity functions.

The proposed tuning approach has proved to be effective in reference input tracking and load disturbance regulation when controlling a real-world servo system targeting the reduced process gain sensitivity. The experimental results validate our evolutionary algorithms, our tuning approach and the fuzzy controllers.

The main drawback of the approach presented in this paper is the need to calculate the sensitivity models. The convenient expressions of the state-space models of the T–S PI-FCs derived in Section 4 can be used as a support for other fuzzy controller structures and models.

Another drawback of our evolutionary algorithms is that they are supported by many simulations conducted with the fuzzy control system to calculate the objective functions, viz. to evaluate the fitness functions. Working with experiments instead of simulations is not a good choice for critical or complex industrial applications in these situations. Therefore the future research will be focused on the reduction of the number of experiments, using iterative experiment-based approaches. Another direction of future research will be focused on the modifications of the objective functions to ensure an improved robustness as well.

Acknowledgement

This work was supported by the UEFISCDI of Romania. The support from the cooperation between the "Politehnica" University of Timisoara, Romania, the Óbuda University, Budapest, Hungary, and the University of Ljubljana, Slovenia, in the framework of the Hungarian–Romanian and Slovenian–Romanian Intergovernmental Science & Technology Cooperation Programs is acknowledged. This work was partially supported by the strategic Grant POSDRU 6/1.5/S/13 (2008) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People. The authors have contributed equally to this paper.

References

[1] F. Al-Obeidat, N. Belacel, J.A. Carretero, P. Mahanti, Differential evolution for learning the classification method PROAFTN, Knowledge-Based Systems 23 (5) (2010) 418–426.
[2] X. Ban, X.Z. Gao, X. Huang, Graphical L_2-stability analysis of Takagi–Sugeno control systems in the frequency domain, Fuzzy Sets and Systems 171 (1) (2011) 93–105.
[3] P. Baranyi, L.T. Kóczy, A general and specialised solid cutting method for fuzzy rule interpolation, Journal BUSEFAL 66 (1) (1996) 13–22.
[4] P. Baranyi, Y. Yam, Singular value-based approximation with Takagi–Sugeno type fuzzy rule base, in: Proceedings of 6th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97), vol. 1, Barcelona, Spain, 1997, pp. 265–270.
[5] S. Blažič, D. Matko, I. Škrjanc, Adaptive law with a new leakage term, IET Control Theory and Applications 4 (9) (2010) 1533–1542.
[6] S. Blažič, I. Škrjanc, S. Gerkšič, G. Dolanc, S. Strmčnik, M.B. Hadjiski, A. Stathaki, Online fuzzy identification for an intelligent controller based on a simple platform, Engineering Applications of Artificial Intelligence 22 (4–5) (2009) 628–638.
[7] F.J. Cabrerizo, I.J. Pérez, E. Herrera-Viedma, Managing the consensus in group decision making in an unbalanced fuzzy linguistic context with incomplete information, Knowledge-Based Systems 23 (2) (2010) 169–181.
[8] C.-Y. Chen, T.-H.S. Li, Y.-C. Yeh, EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots, Information Sciences 179 (1–2) (2009) 180–195.
[9] P.C. Chen, C.W. Chen, W.L. Chiang, GA-based modified adaptive fuzzy sliding mode controller for nonlinear systems, Expert Systems with Applications 36 (3) (2009) 5872–5879.
[10] J.-Y. Chen, C.-C. Wong, Construction of a fuzzy controller through width adaptation, Knowledge-Based Systems 14 (7) (2001) 353–358.
[11] D.M. Dalalah, Piecewise parametric polynomial fuzzy sets, International Journal of Approximate Reasoning 50 (7) (2009) 1081–1096.
[12] R.-C. David, R.-E. Precup, S. Preitl, J.K. Tar, J. Fodor, Parametric sensitivity reduction of PI-based control systems by means of evolutionary optimization algorithms, in: Proceedings of 6th International Symposium on Applied Computational Intelligence and Informatics (SACI 2011), Timisoara, Romania, 2011, pp. 241–246.
[13] R.-C. David, M.-B. Rădac, S. Preitl, J.K. Tar, Particle swarm optimization-based design of control systems with reduced sensitivity, in: Proceedings of 5th International Symposium on Applied Computational Intelligence and Informatics (SACI '09), Timisoara, Romania, 2009, pp. 491–496.
[14] H. Fujita, J. Hakura, M. Kurematu, Intelligent human interface based on mental cloning-based software, Knowledge-Based Systems 22 (3) (2009) 216–234.
[15] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6) (1984) 721–741.
[16] S. Gorlatch, J. Müller-Iden, M. Alt, J. Dünnweber, H. Fujita, Y. Funyu, Clayworks: toward user-oriented software for collaborative modeling and simulation, Knowledge-Based Systems 32 (3) (2009) 209–215.
[17] R.E. Haber, R.M. del Toro, A. Gajate, Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process, Information Sciences 180 (4) (2010) 2777–2792.
[18] R.E. Haber, R. Haber-Haber, A. Jiménez, R. Galán, An optimal fuzzy control system in a network environment based on simulated annealing. An application to a drilling process, Applied Soft Computing 9 (3) (2009) 889–895.
[19] L. Horváth, I.J. Rudas, Modelling and Solving Methods for Engineers, Elsevier, Academic Press, Burlington, MA, 2004.

[20] R. Isermann, Digital Control Systems, second ed., Springer-Verlag, Berlin, Heidelberg, New York, 1989.
[21] R. Jain, N. Sivakumaran, T.K. Radhakrishnan, Design of self tuning fuzzy controllers for nonlinear systems, Expert Systems with Applications 38 (4) (2011) 4466–4476.
[22] Z.C. Johanyák, A brief survey and comparison on various interpolation based fuzzy reasoning methods, Acta Polytechnica Hungarica 3 (1) (2006) 91–105.
[23] Z.C. Johanyák, Student evaluation based on fuzzy rule interpolation, International Journal of Artificial Intelligence 5 (A10) (2010) 37–55.
[24] N. Kabbaj, Y. Nakkabi, A. Doncescu, Analytical and knowledge based approaches for a bioprocess supervision, Knowledge-Based Systems 23 (2) (2010) 116–124.
[25] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[26] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks (ICNN '95), Perth, Australia, 1995, pp. 1942–1948.
[27] J. Kennedy, R.C. Eberhart, A new optimizer using particle swarm theory, in: Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.
[28] J. Liu, W. Wang, F. Golnaraghi, E. Kubica, A neural fuzzy framework for system mapping applications, Knowledge-Based Systems 23 (6) (2010) 572–579.
[29] R. Lovassy, L.T. Kóczy, L. Gál, Function approximation performance of fuzzy neural networks, Acta Polytechnica Hungarica 7 (4) (2010) 25–38.
[30] P. Luukka, Similarity classifier using similarities based on modified probabilistic equivalence relations, Knowledge-Based Systems 22 (1) (2009) 57–62.
[31] J. Ma, J. Lu, G. Zhang, Decider: a fuzzy multi-criteria group decision support system, Knowledge-Based Systems 23 (1) (2010) 23–31.
[32] F. Nagi, L. Perumal, Optimization of fuzzy controller for minimum time response, Mechatronics 19 (3) (2009) 325–333.
[33] B. Mihailović, E. Pap, Asymmetric general Choquet integrals, Acta Polytechnica Hungarica 6 (1) (2009) 161–173.
[34] S.-K. Oh, H.-J. Jang, W. Pedrycz, The design of a fuzzy cascade controller for ball and beam system: a study in optimization with the use of parallel genetic algorithms, Engineering Applications of Artificial Intelligence 22 (2) (2009) 261–271.
[35] S.-K. Oh, H.-J. Jang, W. Pedrycz, Optimized fuzzy PD cascade controller: a comparative analysis and design, Simulation Modelling Practice and Theory 19 (1) (2011) 181–195.
[36] C. Pozna, R.-E. Precup, J.K. Tar, I. Škrjanc, S. Preitl, New results in modelling derived from Bayesian filtering, Knowledge-Based Systems 23 (2) (2010) 182–194.
[37] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, A.S. Paul, Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity, in: A. Gaspar-Cunha, R. Takahashi, G. Schaefer, L. Costa (Eds.), Advances in Intelligent and Soft Computing, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 141–150.
[38] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Rădac, Fuzzy control systems with reduced parametric sensitivity based on simulated annealing, IEEE Transactions on Industrial Electronics (2011) 15. doi:10.1109/TIE.2011.2130493.
[39] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Rădac, Gravitational search algorithms in fuzzy control systems tuning, in: Proceedings of 18th World Congress of the International Federation of Automatic Control (IFAC 2011), Milano, Italy, 2011, paper index 938, p. 6.
[40] R.-E. Precup, H. Hellendoorn, A survey on industrial applications of fuzzy control, Computers in Industry 62 (3) (2011) 213–226.
[41] R.-E. Precup, S. Preitl, I.J. Rudas, M.L. Tomescu, J.K. Tar, Design and experiments for a class of fuzzy controlled servo systems, IEEE/ASME Transactions on Mechatronics 13 (1) (2008) 22–35.
[42] S. Preitl, R.-E. Precup, An extension of tuning relations after Symmetrical Optimum method for PI and PID controllers, Automatica 35 (10) (1999) 1731–1736.
[43] E. Rashedi, Gravitational Search Algorithm, M.Sc. thesis, Shahid Bahonar University of Kerman, Kerman, Iran, 2007.
[44] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Information Sciences 179 (13) (2009) 2232–2248.
[45] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, BGSA: binary gravitational search algorithm, Natural Computing 9 (3) (2010) 727–745.
[46] R. Rodrigues Sumar, A.A. Rodrigues Coelho, L. dos Santos Coelho, Computational intelligence approach to PID controller design using the universal model, Information Sciences 180 (20) (2010) 3980–3991.
[47] S.-B. Roh, S.-K. Oh, W. Pedrycz, A fuzzy ensemble of parallel polynomial neural networks with information granules formed by fuzzy clustering, Knowledge-Based Systems 23 (3) (2010) 202–219.
[48] A. Sala, T.M. Guerra, R. Babuška, Perspectives of fuzzy systems and control, Fuzzy Sets and Systems 156 (3) (2005) 432–444.
[49] J.K. Tar, J.F. Bitó, L. Nádai, J.A. Tenreiro Machado, Robust fixed point transformations in adaptive control using local basin of attraction, Acta Polytechnica Hungarica 6 (1) (2009) 21–37.
[50] J. Vaščák, Navigation of mobile robots using potential fields and computational intelligence means, Acta Polytechnica Hungarica 4 (1) (2007) 63–74.
[51] J. Vaščák, L. Madarász, Adaptation of fuzzy cognitive maps-a comparison study, Acta Polytechnica Hungarica 7 (3) (2010) 109–122.
[52] Y. Yam, M.L. Wong, P. Baranyi, Interpolation with function space representation of membership functions, IEEE Transactions on Fuzzy Systems 14 (3) (2006) 398–411.
[53] S. Zhou, G. Ji, Z. Yang, W. Chen, Hybrid intelligent control scheme of a polymerization kettle for ACR production, Knowledge-Based Systems 24 (7) (2011) 1037–1047.

Computers in Industry 74 (2015) 75–94

Contents lists available at ScienceDirect

Computers in Industry

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

An overview on fault diagnosis and nature-inspired optimal control of industrial process applications

Radu-Emil Precup [a,*], Plamen Angelov [b,c], Bruno Sielly Jales Costa [d], Moamar Sayed-Mouchaweh [e]

[a] Politehnica University of Timisoara, Department of Automation and Applied Informatics, Bd. V. Parvan 2, 300223 Timisoara, Romania
[b] Lancaster University, Data Science Group, School of Computing and Communications, Lancaster LA1 4WA, United Kingdom
[c] Chair of Excellence, Carlos III University, Madrid, Spain
[d] Federal Institute of Rio Grande do Norte – IFRN, Campus Natal, Zona Norte, Natal, Brazil
[e] École des Mines de Douai, Mines-Douai, IA, 59500 Douai Cedex, France

ARTICLE INFO

Article history:
Received 31 December 2014
Received in revised form 27 February 2015
Accepted 4 March 2015
Available online 23 March 2015

Keywords:
Data-driven control
Data mining
Evolving soft computing techniques
Fault diagnosis
Nature-inspired optimization algorithms
Wind turbines

ABSTRACT

Fault detection, isolation and optimal control have long been applied to industry. These techniques have proven various successful theoretical results and industrial applications. Fault diagnosis is considered as the merge of fault detection (that indicates if there is a fault) and fault isolation (that determines where the fault is), and it has important effects on the operation of complex dynamical systems specific to modern industry applications such as industrial electronics, business management systems, energy, and public sectors. Since the resources are always limited in real-world industrial applications, the solutions to optimally use them under various constraints are of high actuality. In this context, the optimal tuning of linear and nonlinear controllers is a systematic way to meet the performance specifications expressed as optimization problems that target the minimization of integral- or sum-type objective functions, where the tuning parameters of the controllers are the vector variables of the objective functions. The nature-inspired optimization algorithms give efficient solutions to such optimization problems. This paper presents an overview on recent developments in machine learning, data mining and evolving soft computing techniques for fault diagnosis and on nature-inspired optimal control. The generic theory is discussed along with illustrative industrial process applications that include a real liquid level control application, wind turbines and a nonlinear servo system. New research challenges with strong industrial impact are highlighted.

1. Introduction

For four decades fault diagnosis, also referred to as Fault Detection and Isolation (FDI), has been a strong field of study, however, it has gained more attention in the later years. Considering the complexity of the current industrial applications, FDI is a very challenging issue nowadays. Studies have shown that the human operator is responsible for 70–90% of the accidents in industrial environments [1,2]. Therefore, FDI techniques are used to improve operational safety, preventing (or reducing) accidents and unscheduled stoppages.

The data-driven methods, or process history-based methods, are regularly used for FDI tasks, many times combined with model-based approaches. The main advantage of such methods is that they do not require much (or none at all) expertise of the operator/designer, since they are mainly based on the data collected from the process, which can be historical (off-line) or real-time (on-line). This is a very important feature, since data-driven methods can cope with the problem of data drift and other unpredicted disturbances.

The data drift is defined in [3] as a change in the learned structure that occurs over time and can lead to a drastic drop of classification accuracy. Many times a system is designed to work under determined circumstances and its behaviour can be inadequate when dealing with unpredictable changes. These variations can occur due to process faults, disturbances or even slow environmental alterations.

At this point it is important to distinguish the concepts of *adaptive* and *evolving* systems. The term "adaptive" usually refers to the conventional systems, known in control theory for working

* Corresponding author. Tel.: +40 256 403229; fax: +40 256 403214.
E-mail address: radu.precup@upt.ro (R.-E. Precup).

http://dx.doi.org/10.1016/j.compind.2015.03.001
0166-3615/

with linear parameter adaptation, e.g., the traditional Adaptive Neural Fuzzy Inference Systems (ANFIS). The term "evolving", on the other hand, is associated to the systems that are also able to perform a gradual change in its core [4], either updating (adding and removing rules) the rule base, in the case of fuzzy systems, or updating (adding and removing nodes and layers) the structure, in the case of neural networks.

The decision regarding the use of process history-based methods rely on the fact that model-based techniques are very restrictive in the sense that they require information that it is usually not available in practical applications. While quantitative model-based methods require accurate mathematical descriptors of the process, qualitative model-based techniques work with a qualitative database, usually built from knowledge from an operator or an expert system. These models are often not available and, even it they are, it is many times impracticable to obtain all information of relevant physical parameters of the system, not to mention that external parameters, such as unpredictable disturbances, model uncertainties and so on, are not considered.

Process history-based techniques, on the other hand, do not require any knowledge, either quantitative or qualitative, about the process. Instead, they use massive historical information collected from the process. This data is, then, transformed and presented as a priori information to the FDI system.

The task of FDI can be analyzed as a classification problem in both stages–detection and isolation. Fault detection is the task where it is possible to identify whether the system is working in a normal operating state or in a faulty mode. However, in this stage, vital information about the fault, such as physical location, length or intensity, are not provided to the operator [5]. The detection stage, thus, can be addressed by a general one-class classifier, able to distinguish if the current collected data samples belong to a determined class of data, e.g. "normal". Fault isolation, on the other hand, refers to determination of kind, location and time of detection of a fault, and follows the fault detection stage [6]. The challenge is to match each pattern of the symptom vector with one of the pre-assigned classes of faults, in the case of supervised approaches, and the fault-free case [7], as a multi-class classifier.

Once the fault diagnosis is guaranteed, a systematic way to meet the performance specifications of control systems in industrial applications is the optimization in terms of optimization problems with variables represented by the tuning parameters of the controllers. Nature-inspired algorithms can solve these optimization problems, and they ensure the optimal tuning of controllers in order to meet the performance specifications expressed by adequately defined objective functions and constraints. The constraints are due to technical and/or economical operating conditions of industrial process applications, and they include stability, sensitivity, robustness and fault diagnosis conditions.

The motivation for nature-inspired optimization algorithms (NIOAs) in the optimal control of industrial process applications concerns the ability of such algorithms to cope with non-convex or non-differentiable objective functions because of the process complexity, of the controllers' structures and eventually the controllers' nonlinearities, which can lead to multi-objective optimization problems. In addition, the complexity of the classical optimization algorithms is very high, and this requires enormous amount of computational work. Therefore, the NIOAs are appreciated because they are better in terms of efficiency and complexity than classical optimization algorithms.

As generally shown or with focus on certain classes of NIOAs in [8–13], these algorithms are based on biological, physical, and chemical phenomena of nature. NIOAs have the distinct ability of finding the global minimum (or maximum) of certain objective functions under specific conditions. In addition, the analytical expression of the objective functions depending on other design (or tuning) parameters may be difficult or even impossible to formulate. These are the reasons why the NIOAs applied to optimal control of industrial processes are justified and also challenging.

This paper addresses the following topics: the state-of-the-art on machine learning, data mining and evolving soft computing techniques for fault diagnosis is discussed in Section 2. The unsupervised and autonomous self-evolving fault diagnosis is treated in Section 3. An illustrative fault diagnosis application related to a real liquid level control system controlled by a multi-stage fuzzy controller using a pilot plant for industrial process control is included. The problem setting for fault diagnosis in wind turbines and a classification of the methods used with this regard is presented in Section 4. It discusses the interest, motivation and challenges related to the fault diagnosis of wind turbines. Machine learning and data mining techniques are next organized in the framework of a general scheme that achieves fault diagnosis of wind turbines. Then, the different steps of this scheme are detailed in order to emphasize the links between the methods and techniques that they use and how they answer the challenges related to the fault diagnosis of wind turbines. Section 5 discusses NIOAs in the optimal tuning of linear controllers for industrial process applications and gives an example concerning the position control of a nonlinear servo system. The nature-inspired optimal tuning of nonlinear controllers in industrial process applications is next treated in Section 6 with focus on fuzzy controllers, but neural network controllers and sliding mode controllers are also considered. The drawbacks and research challenges in fault diagnosis and nature-inspired optimal control are discussed in Section 7. The concluding remarks are highlighted in Section 8.

2. State-of-the-art on machine learning, data mining and evolving soft computing techniques for fault diagnosis

Many authors have, very recently, contributed to the fault diagnosis field of study, with extensive studies, compilations and throughout reviews. In this section, we are going to address a few important approaches to FDI using intelligent techniques, specially focusing on machine learning, data mining, clustering and evolving techniques applied to industrial problems.

Among the recent studies on the topic, the paper [14] proposes an incremental support vector data description and extreme learning machines are used to solve the problem of classification of faults when the number of classes is unknown and tend to increase over time. While the proposed Support Vector Machine (SVM) is used to quickly detect new failure modes, Extreme Learning Machine (ELM) is changed into an elastic structure whose output nodes can be added incrementally to cope with the new fault scenario. The algorithm is applied to a Diesel engine under eleven different fault conditions, however can be suitable to other mechanical equipment.

An application of ELM to real-time FDI with data pre-processing through wavelet packet transform and time-domain statistical features is suggested in [15]. The process of feature extraction is, then, performed by a kernel Principal Component Analysis (PCA) algorithm. As a case of study, a comparison between ELM and SVM on a fault detection problem is conducted, resulting in a considerable advantage for the ELM algorithm.

An automatic method for bearing FDI using the vibration signals as input is given in [16]. A one-class v-SVM is used to distinguish normal and faulty operation modes, where the model of normality is built from data extracted under normal conditions. Band-pass filters and Hilbert Transform are used to isolate the fault. An experimental study is then performed using two different data sets: real data from a laboratory test-to-failure experiment and data obtained from a fault-seeded bearing test. The results

obtained were very satisfactory both to detection and isolation stages.

An approach to FDI of monoblock centrifugal pump using SVM is presented in [17]. Again, vibration signals are used as input and the feature extraction task is performed by continuous wavelet transform. Classification accuracies of different wavelet families and levels were calculated and compared to find the best wavelet for the FDI task.

A wind turbine feature extraction FDI method, based on diagonal spectrum and clustering binary tree SVM using, once again, vibration rotating machine signals as input, is suggested in [18]. Then, a self-organizing feature map neural network is used to build a cluster binary tree of fault features and multiple fault classifiers are designed to train and test the data samples.

An FDI scheme for condition machinery of an industrial steam turbine using a fusion of SVM and ANFIS classifiers is introduced in [19]. A multi-attribute data is fused into aggregated values of a single attribute by ordered weighted averaging operators. A simulation study is then performed and indicates that the proposed technique outperforms individual SVM and ANFIS individual methods.

A comparative study between several unsupervised learning algorithms applied to FDI in industrial machines is performed in [20]. A particular weighted local and global regressive mapping algorithm is proposed and compared to the others, including regressive mappings, locality preserving projection, Isomap and PCA. Machine vibration signals are analyzed and deviations from the normal pattern are used to build a 13-dimension feature data set representing different health conditions. The results show that, not only different faults can be classified, but degree of fault severity can also be captured, especially with the proposed algorithm.

A semi-supervised weighted kernel clustering algorithm to FDI of known and unknown fault patterns in order to overcome the recurrent problem of traditional supervised learning methods such as SVM is proposed in [21]. The algorithm is based on gravitational search and is able to use historical classification information to improve diagnosis accuracy. A clustering data set composed of labelled and unlabeled fault samples is created and the model is defined based on wrong classification rate of labelled samples. New fault samples are isolated by calculating the weighted kernel distance to fault cluster centres. If the fault sample is unknown, it will be added in historical data sets and clustering results updated. The algorithm is applied to FDI of rotary bearing and compared to other traditional clustering methods, SVM and neural network.

An adaptive recursive clustering algorithm for novelty detection applied to real-time FDI of a pneumatic process is suggested in [22]. The procedure is able of identifying previously unseen operation modes of the process, updating the knowledge basis and clusters, evolving and learning the new features of the monitored data.

An early fault detection method based on thermal parameters and a fault predication system for marine Diesel engines is discussed in [23]. The algorithm is developed based on data mining of abnormal thermal parameters from local safety, alarm and control systems.

An approach to adaptive FDI by detecting new operation modes of a process and incorporating that information in an evolving fuzzy classifier used for diagnosis is presented in [24]. The algorithm is based on incremental clustering to generate fuzzy rules describing each new detected state of operation, which is then learned and properly identified the next time it occurs. The adaptive classification process is performed on-line and applied to the FDI of an industrial actuator. The results show that the algorithm is very suitable for the FDI of dynamic systems, especially when there is no previous knowledge on all failure modes of the process.

The problem of design and development of data-driven multiple sensor FDI for nonlinear processes is addressed in [25]. The problem solving is based on an evolving multi-Takagi-Sugeno framework, in which each sensor output is estimated using a model derived from the available input-output data, and an algorithm is proposed. Simulation results from a continuous-flow stirred-tank reactor validate the algorithm.

A fuzzy relational sliding mode observer for estimation of magnitude of incipient faults in non-linear systems is proposed in [26]. An on-line learning FDI scheme is used to update the model and identify the fault in periodical modes. The algorithm is applied to a simulated cooling-coil subsystem of an air-conditioning plant.

A general method to deal with concept drift on one-class data-driven models is presented in [27]. The proposed algorithm is able to develop an automatically retrained and updated polygon-based model on-line. Data representing relevant faults was inserted into the data set and the proposal tested using data collected from a real hydraulic drive system. Results were significantly improved when compared to static polygon-based models.

An application of evolving fuzzy modelling to FDI divided in two steps, detection and accommodation, is given in [28]. Both detection and accommodation stages use evolving Takagi-Sugeno fuzzy models. Data from detection stage is used for accommodation in a model predictive control scheme. The evolving fuzzy modelling approach increases control performance when the system is in a faulty mode. The algorithm continuously evaluates control performance and performs on-line clustering. Two simulated faults – load process and change in heating – in a distillation column process were used to validate the proposal.

3. Unsupervised and autonomous self-evolving fault diagnosis

A few unsupervised and autonomous fuzzy rule-based approaches have very recently been proposed to fault diagnosis problems [29–31]. These approaches divide the fault diagnosis task in its two sequential stages, detection and isolation.

The detection algorithm is based on the recursive density estimation (RDE) [32], which allows building, accumulating, and self-learning a dynamically evolving information model of "normality", based on the process data for particular specific plant, and considering the normal/accident-free cases only. It works as other statistical methods, however, RDE does not require that the process parameters follow Gaussian/normal distribution (or any distribution at all) nor makes other prior assumptions.

The isolation stage is based on the concept of self-learning evolving classifiers, firstly proposed in [33]. However, this new algorithm, called *AutoClass*, is fully unsupervised, based on AnYa model and thus, unlike the traditional fuzzy rule-based (FRB) systems (e.g., Mamdani, Takagi-Sugeno), does not require the definition of membership functions. The antecedent part of the inference rule works with the concepts of data clouds [34] and relative data density, representing the exact real distribution of the data.

A data cloud can be defined as a set of data points in the n-dimensional feature space, similarly to concept of data clusters. However, a data cloud does not have boundaries or pre-defined shapes, as the clusters, and are also non parametric, following the exact distribution of the data [32]. Its model is intrinsically fuzzy, since a given data sample can belong to all the data clouds with a different membership degree.

Although AnYa-like systems enable the use of any type of consequents (Mamdani, linear, quadratic and so on), the inference rules of AutoClass use zero-order Takagi-Sugeno crisp functions, i.e., a class label $L_i = [1,K]$. The inference rules follow the default

construct of an AnYa FRB system [33]:

$$R^i: \text{IF } \mathbf{x} \sim (i) \quad \text{THEN} \quad L_i, \tag{1}$$

where R^i is ith rule, $x = [x_1 \dots x_n]^T \in R^n$ is the input data vector, $(i) \in R^i$ is ith data cloud, and $\sim$ denotes the fuzzy membership expressed linguistically as "is associated with". The inference is, then, produced with the "winner-takes-it-all" rule.

The rule base is, at each step/iteration of the algorithm, updated. Not only the parameters of each rule/cloud is changed (e.g., centre, radius, class label), but new rules can be created after the discovery of a new regular point of operation. The whole procedure is recursive, which guarantees that read data samples do not need to be stored in memory, enhancing performance and minimizing computational effort and requirements. The membership degree of x_k to the data cloud (i) can be recursively calculated as [32]

$$\gamma_k^i = \frac{1}{1 + ||x_k - \mu_k||^2 + X_k - ||\mu_k||^2}, \tag{2}$$

where μ_k and X_k are updated by

$$\mu_k = x_1, \quad \mu_k = \frac{k-1}{k}\mu_{k-1} + \frac{1}{k}x_k, \quad k = 2 \dots n, \tag{3}$$

$$X_1 = ||x_1||^2, \quad X_k = \frac{k-1}{k}X_{k-1} + \frac{1}{k}||x_k||^2, \quad k = 2 \dots n. \tag{4}$$

Due to the recursive calculations, both stages are highly suitable for on-line and real-time applications, can start with no prior knowledge about the process or off-line training, from the very first data point acquired. The AnYa-type fuzzy rules have no specific shapes or parameters for the membership functions and the approach is entirely data-driven.

As a case of study, the algorithms were tested in a real liquid level control application, using a pilot plant for industrial process control. The plant, which is developed by De Lorenzo [35], and consists of a set of commercial industrial sensors and actuators connected to two tanks through a piping system, is presented in Fig. 1.

The plant is controlled by a multi-stage fuzzy controller, developed in [36]. The fault-free behaviour of the system for the steady state regime is shown in Fig. 2. Note that all signals (u is the control action and y is the plant output) are stable/non-oscillatory for the normal state of operation of the plant, which leads to a high data density. Any considerable density drops will be automatically evaluated.

A group of four different faults, including actuator offsets and tank leakages, was generated. The faults F_2 (positive high offset), F_4 (negative high offset), F_1 (positive low offset) and F_9 (tank leakage) were sequentially activated, separated by short periods of fault-free states of operation. Fig. 3 shows the detection stage of the proposed sequence of faults. While monitoring the oscillation of the error and control signals, the algorithm is able to detect the beginning (black dashed bars) and end (grey dashed bars) of each fault without any considerable delays. In the first stage, the system is responsible only for distinguish the normal operation from a fault.

Fig. 1. Pilot plant for industrial process control.

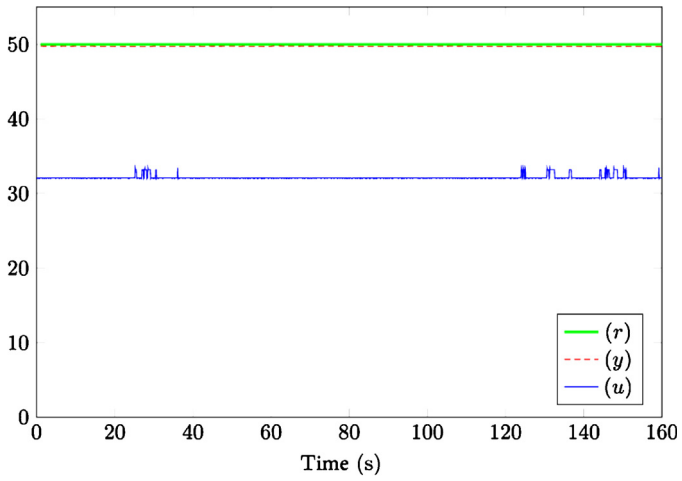

Fig. 2. Fault-free state of operation of the plant.

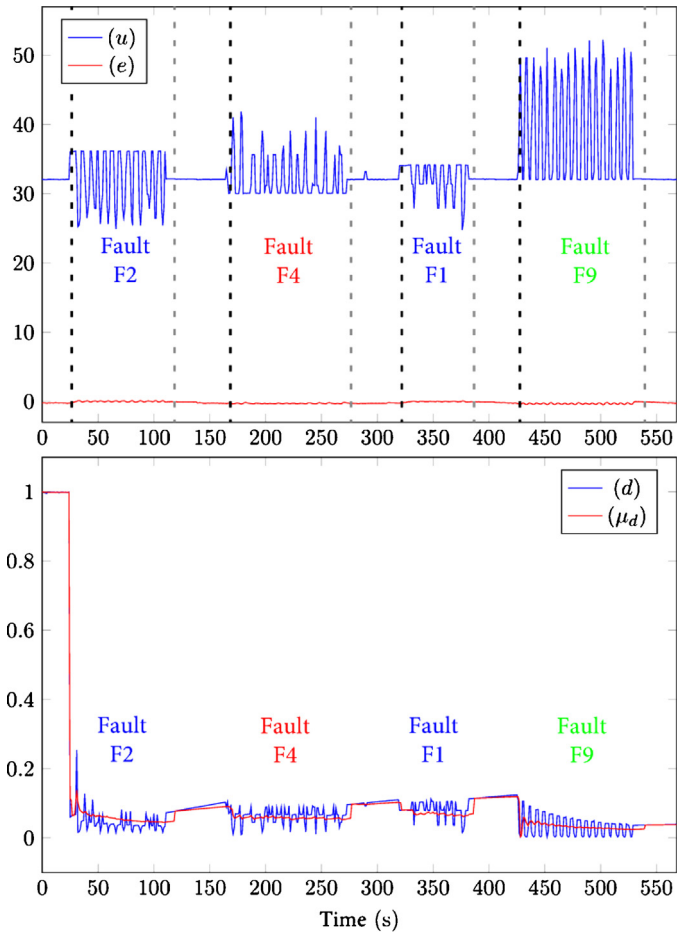

Fig. 3. Detection of a sequence of faults using RDE.

When a fault is detected (event indicated by the black dashed bars in Fig. 3), the isolation stage of the proposed algorithm is activated. The data set is spatially distributed in the n-dimension feature space. The feature input vector is composed of the most descriptive data elements from the process, which may be manually or automatically selected (through PCA, wavelet or any known feature extraction techniques). For this particular example, two features were manually selected and used as input to the isolation stage – here called Feature 1 and Feature 2 –, where the first represents the period and the second represents the amplitude of the control signal. Fig. 4 shows the 2-dimensional feature space, after the reading of all available data samples.

At the end of the on-line procedure, the fuzzy rule base, autonomously generated from the data stream (total of 5600 data samples) is

$$\begin{aligned} &R^1: IF\ \mathbf{x} \sim (1)\ \text{THEN}\ (\text{"Class 1"}) \\ &R^2: IF\ \mathbf{x} \sim (2)\ \text{THEN}\ (\text{"Class 2"}) \\ &R^3: IF\ \mathbf{x} \sim (3)\ \text{THEN}\ (\text{"Class 3"}) \end{aligned} \tag{5}$$

with

$$\begin{aligned} &(1): c_1 = [0.416 \quad 3.316]\ \text{AND}\ Z_1 = [0.251 \quad 0.756], \\ &(2): c_2 = [-0.513 \quad 2.706]\ \text{AND}\ Z_2 = [0.250 \quad 0.601], \\ &(3): c_3 = [-0.416 \quad 1.491]\ \text{AND}\ Z_3 = [0.197 \quad 0.451], \end{aligned} \tag{6}$$

where c_i is the focal point (mean) and Z_i is the zone of influence of the cloud, $i = 1 \ldots 3$.

It is possible to see in Fig. 4 that three different clouds (Class 1, Class 2 and Class 3) were created, each one representing one class of fault (F_1 and F_2 – actuator positive offset, F_4 – negative offset, F_9 – tank leakage). The normal state of operation is not represented in the chart because the isolation/classification stage is only triggered once a fault is detected. The classification algorithm is quite unique in the sense that it autonomously and in a completely unsupervised manner (automatic labelling) identifies the types of faults. Traditional models, such as neural networks, start to drift and a re-calibration is needed. This unsupervised and self-learning method does not suffer from such disadvantage because it is adapting and evolving.

It is important to highlight that all class labels are generated automatically in a sequence ("Class 1", "Class 2" and so on), as different and unknown faults are detected. The labels, thus, do not represent the actual type or location of the fault, but they are very useful to distinguish different faults. Since no training or pre-definition of faults or models are needed, the correct labelling can be done in a semi-supervised manner by the human operators, without requiring prompt/synchronized actions of the user. Moreover, in a semi-supervised approach, the operator, should be able to merge, split and rename the generated clouds/rules/classes of faults, enabling the classification of faults/operation states that cannot be represented by compact/convex data clouds.

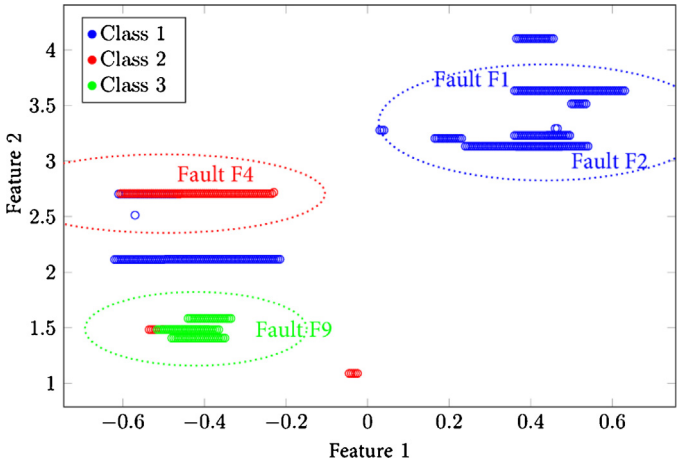

Fig. 4. Classification and automatic labelling of the faults using AutoClass.

4. Methods used for fault diagnosis in wind turbines

Wind turbines convert the kinetic energy of wind into the electrical energy. The most used wind turbines are horizontal-axis using a three-bladed rotor design with an active yaw system keeping the rotor oriented upwind [37]. The wind turbine main components (Fig. 5) are the gearbox, the generator with converter and the blade and pitch system [38]. The gearbox connects the low-speed shaft to the high-speed shaft in order to increase the rotational speed to a level required by the generator to produce electric energy. The generator converts rotational energy into electric energy. The role of the pitch system is to adjust the blade pitch by rotating it according to the pitch angle position reference provided by the controller. The latter decides the pitch angle position reference according to the wind speed in order to allow an optimum energy production.

Wind turbines operate, generally, in severe and remote environments and require frequent schedule maintenance. Reducing their operations and maintenance (O&M) costs provides the impetus for increasing the competitiveness of this clean energy source according to the traditional ones. Indeed, O&M costs may reach 25–30% of the energy generation cost [39]. One of the main sources for the O&M costs is the unscheduled maintenance due to unexpected failures. This can be costly not only for the maintenance support but also for the produced energy. Moreover, the accidents, in particular the fatal ones, of wind turbines increases year over year [40]. Therefore, a comprehensive and automated health monitoring system can reduce the O&M costs as well as the lost production time and ensure the wind turbines security and safety by detecting and isolating faults before becoming expensive, critical or catastrophic.

The operational state of a wind turbine varies from fully operational to malfunction and shutdown. Their monitoring system reports excessive wear to maintenance operators based on the use of SCADA (Supervisory Control and Data Acquisition) data. SCADA records the values of multiple operational parameters at time intervals of 10 min as well as system's potential or emerging faults expressed as status codes. The latter are event trigger that point to the actions needed to overcome the fault or to indicate affected components. However, SCADA based monitoring systems suffer from two major drawbacks. Firstly, they cannot achieve a precise and reliable localization of affected components. This is due to fault propagation from the component, source of this fault, to the other connected components

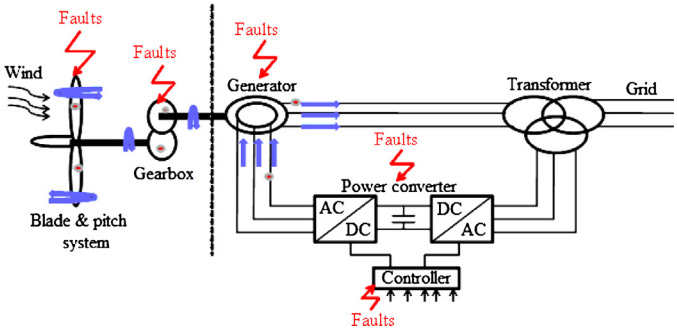

Fig. 5. Main components of a horizontal-axis wind turbine using a three-bladed rotor design.

with it. Indeed, a faulty component, as blade angle asymmetry, adversely impacts other components, as blade pitch motor, electronic control and gearbox, since they are directly in contact with this affected component. Therefore, several components of wind turbine are suspected and additional time is required to isolate the component responsible of the occurrence of this fault. This will lead to increase the time of wind turbine unavailability as well as its cost maintenance. As an example, a turbine shutdown may occur due to the blade angle asymmetry fault. Turbine unavailability due to this fault may reach 180 h resulting in production losses [41]. The second major drawback of SCADA monitoring systems is the delay between acknowledging status codes and maintenance actions which can exceed, for example, 168 h [42] for blade angle asymmetry fault. Moreover, some critical faults, as blade pitch motor, can generate or execute dangerous or prohibitive behaviours. Therefore, to ensure wind turbine security, it is crucial to detect these faults after their occurrence but prior to the execution of a dangerous behaviour in order to preserve their safety and security.

However, fault diagnosis of wind turbines is a challenging task because:

- the measurements of wind turbines are not enough reliable due to the high uncertainty of wind speed and to the turbulence around the rotor plane,
- the nonlinearity of the wind turbine dynamics,
- the occurrence of certain faults (e.g., blade pitch motor faults) in operation conditions (power optimization region) in which fault consequences are hidden,
- the actions of the control feedback which compensate the fault effects,
- the low volume of data (imbalance data) describing the faults according to the data coming from normal operation conditions, which makes the fault prediction task difficult.

There are numerous methods in the literature that are used to achieve fault diagnosis in wind turbines. They achieve the fault diagnosis by reasoning over differences between desired or expected behaviour, defined by a model, and observed behaviour provided by sensors. Generally, they can be classified according to the system size (wind turbine as a whole or one of its components) into general purpose methods treating the whole wind turbine and components-based methods focusing on one component of the wind turbine. In general purpose methods [43], the model represents the global behaviour of the wind turbine based on its general parameters, as wind speed, generated electrical power, air temperature, etc. Thresholds are defined as alarm levels in order to detect significant changes in turbine behaviour. Exceeding these alarm levels, due to the occurrence of a fault, leads to shut down the turbine and to wait for a remote restart or repair. Therefore, trend analysis of some representative signals using signal processing and data mining techniques can help to detect the fault occurrence at early stage. These methods have the advantage to be cost-effective since no need for a prior knowledge about the relationships between wind turbine components. However, they do not provide a specific or precise diagnostics about the components responsible of the fault occurrence. Component-based methods [44] achieve the fault diagnosis of one specific component of wind turbine. The failure of this component, e.g., gearbox, blade pitch system, is normally critical according to its maintenance costs or/and its frequency occurrence. These methods provide reliable and precise diagnostics. However, a depth analysis is required to determine the high sensitive parameters and features to normal and faulty behaviours of the monitored component.

In both categories, a model, representing the wind turbine normal or/and fault behaviours, is required. There are two main categories of methods to build a model: internal and external methods. The internal methods use a mathematical or/and structural model to represent the relationships between measurable variables by exploiting the physical knowledge or/and experimental data about the system dynamics. These variables represent the internal parts of the wind turbine. The response of the mathematical model is compared to the observed values of variables in order to generate indicators used as a basis for the fault diagnosis. Generally, the model is used to estimate the system state, its output or its parameters. The difference between the system and the model responses is monitored. Then, the trend analysis of this difference can be used to detect changing characteristics of the system resulting from a fault occurrence. The internal methods used to achieve the fault diagnosis of wind turbines are divided into three main categories (Fig. 6): parameter identification [45], observer- and state-based [46] and signal- or feature-based [47] approaches. These methods were applied successfully to achieve the diagnosis of faults impacting the pitch system [45,46,48], the generator [48,49], the converter [49], and the gearbox [48].

The major advantages of these methods are their ability to:

- detect both the abrupt and progressive failures via trend analysis, and
- give a precise decision or isolation of a failure.

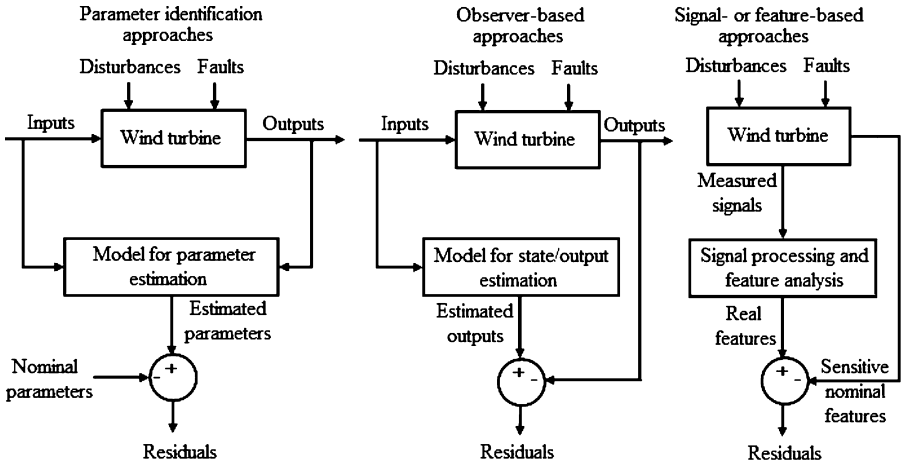

Fig. 6. Internal methods used to achieve the fault diagnosis of wind turbines.

However, they suffer from:

- the necessity to depth information about system behaviour and failures, which is hard to obtain for complex and strong non-stationary systems as wind turbines, and
- the sensitivity of the fault detection to model design errors and measurements noises.

The external methods consider the system as a black box, in other words, they do not need any mathematical model to describe the system dynamical behaviours. They use exclusively a set of measurements or/and heuristic knowledge about system dynamics to build a mapping from the measurement space into a decision space. They include expert systems and machine learning and data mining techniques. These methods are suitable for systems that are difficult to model, they are simple to implement and require short processing time. However, since the obtained models are not transparent, the obtained results are hard to be interpreted and demonstrated.

There are several machine learning and data mining methods used to achieve the fault diagnosis of wind turbines. Such methods are described and successfully applied in [50–59]. They are used through a general scheme based on the following steps (Fig. 7): data preparation, data preprocessing and labelling, data analysis, model design and model validation. These steps are developed in the following subsections.

4.1. Data preparation

The available data are collected through SCADA system of one or several wind farms. A typical SCADA system records data on more than 50 parameters. They include operation data and status data. Operational data records measurements averaged over 10 min intervals on controlled parameters as blade pitch angles and generator torque and on non-controlled parameters as wind speed, tower acceleration and gear box temperature. Status data are event trigger data recorded when a wind turbine changes its status because of potential faults expressed as status codes. The latter trigger, according to the fault severity, one of the following three alarms: information, warning and failure.

Since machine learning and data mining techniques require data about normal and failure conditions, the aim of data preparation step is to divide SCADA data into several data sets. Each data set describes the operation conditions of a wind turbine according to normal conditions or to a fault occurrence in one of its components. This can be achieved as follows. When a status code reports the occurrence of a fault in a component at time t, all SCADA operational data recorded in advance to t of n 10 min will be considered as describing the conditions of occurrence of this fault. n must be determined according to the fault dynamics. As an example, electrical faults have significant faster dynamics than the ones of mechanical faults. Therefore, n is much smaller in the case of electrical faults than the one for mechanical faults. It is worth to note that 1 second sampled snapshot files are generated automatically by SCADA system to record operational data at 1 second sampling rate 7 min before and 3 min after the occurrence of some critical faults. These snapshots can be very useful in the analysis and model design steps since they provide extensive data during the fault occurrence and development. Finally, each data set will be divided into learning and validation sets. The learning set will be used in the model design step to construct the decision space and the validation set to compute the indicators about the model performance.

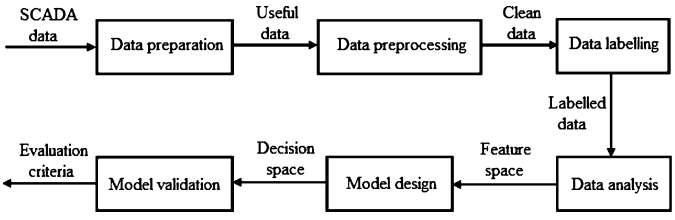

Fig. 7. General scheme used by machine learning and data mining techniques in order to achieve fault diagnosis of wind turbines.

A severe problem meeting in this step is related to the effect of imbalanced data sets. Indeed, SCADA data about normal operation conditions are much higher than the one of failure conditions [41]. The fact that the faults occurring in any wind turbine component, in particular critical ones, are not frequent entails that the available SCADA data describing their dynamical behaviour is low. This makes their prediction a very hard task. However, several approaches, as the ones studied in [41,42] and the references therein, can be used to mitigate the impact of imbalanced data sets.

4.2. Data preprocessing and labelling

The SCADA data collected from wind turbines suffers from several imperfections as noises due to sensors and their environment, sensor errors, sensor uncertainties, inconsistent data, outliers and other abnormal measurements, etc. Therefore, preprocessing techniques are applied on this data in order to obtain valid and clean data set. The latter is essential for data analysis and model design steps in order to build a discriminant feature space and model optimal generalization. The principal preprocessing techniques are [57]:

- denoising in order to remove noises and enhance the signal to noise ratio,
- validity check in order to remove outliers,
- outside the expected ranges, and inconsistent input data with other input variables or with output targets,
- data scaling in order to normalize the different input data variables,
- missing data processing in order to remove data with some unknown or missing attributes or targets (outputs),
- lag removal in order to remove the delay between wind turbine measured signals and the changes in its operation conditions.

Before using the preprocessed SCADA data set by data analysis and model design steps, it must be labelled by a fault label indicating the fault origin or source or by normal label N. Indeed, operational state of a component of a wind turbine varies from fully operational to malfunction and shutdown; these states are labelled with status codes. As an example, when the generator brushes starts to wear, the turbine remains operational until the worn-out of the brushes. In this case, the wind turbine is shut-down. The generator brushes worn-out can be seen through the deterioration of the wind turbine produced power during several consecutive days. The status codes generated by SCADA monitoring system are used to label the operation data during the period of produced power degradation. However, it is worth to note that the time occurrence of a fault cannot be precisely decided. Therefore, clustering techniques can be jointly used with status codes in order to enhance SCADA data labelling. In clustering techniques [56,59], no labels of observations are available. Therefore, these observations are partitioned into groups (clusters) of similar properties.

4.3. Data analysis

The aim of data analysis is to find the minimum number of features leading to discriminate between normal and each of the different fault behaviour of a wind turbine. Features are the abstraction of data. They are sensitive (related) to the system behaviours (conditions). They are related to physical characteristics of a system. These features are determined using two operations: feature selection and feature extraction [60]. Feature selection is the operation of seeking, from a large candidate set, distinguishing features leading to separate as well as possible the different dynamical behaviours in order to reduce the prediction error. These candidates are determined based on data (signal) processing on time-domain (statistical moments), on frequency response (i.e., Power Spectral Density), on the estimation of the coefficients of Auto Regressive model, etc. Feature extraction (dimension reduction) is the procedure of keeping a subset of features from the set of selected features by removing the redundant and irrelevant ones.

There are more than 50 parameters measured by SCADA system for wind turbines. In order to define the feature space, the parameters related to each component of wind turbines are determined. As an example, to diagnose the blade pitch angle asymmetry (abnormal deviation of the pitch angle among the blades) and implausibility (abnormal difference between the actual and desired blade angles) faults [41,54], only the parameters (original and derived ones) related to the pitch system are analyzed (e.g., blades 1, 2 and 3 pitch angles provided by each of the two redundant sensors, desired pitch angle, nacelle revolution, tower deflection, etc.). Features calculated based on the deviations between the pitch angles of blade 1, blade 2 and blade 3 are discriminant to diagnose the blade pitch angle asymmetry and implausibility [41,54,61].

4.4. Model design

This step aims at building a model as a mapping of a pattern, representing the current wind turbine operation mode or one of its components, from the feature space into the decision one. The latter is defined by a set of predefined labels indicating normal or faulty behaviours. This mapping is achieved using a classifier, a regression model or statistical indicators. A classifier is a method or algorithm which generates a decision function in order to assign a label to unlabelled incoming patterns. The classifier can be built using supervised (labelled patterns in the learning set) or unsupervised (unlabelled patterns in the learning set) learning methods. There are many supervised and unsupervised learning methods in the literature [62]. They were applied successfully to achieve the fault diagnosis of wind turbines such as k nearest neighbours [41,54], bagging [41], neural networks [41,55–57], support vector machines [58,61] and self-organizing feature maps [56]. The regression model estimates or predicts the value of an input pattern at time t based on the correlation between wind turbine input variables. Then, a fault is diagnosed when the estimation or prediction error becomes greater than a predefined threshold [57,58]. The statistical indicators are generated based on some statistical tests, as Q-stat and T^2 stat, on patterns obtained by the projection of the learning set into a new space using linear or non-linear principal components analysis [57].

The choice of one of these methods depends on several factors as the discrimination power of the feature space, the decision boundaries between the different classes representing normal and fault behaviours, simple or multiple fault scenarios, multiplicative or additive faults, abrupt or drift-like faults, etc.

4.5. Model validation

In order to evaluate the prediction accuracy of the designed model in the previous step, the following evaluation metrics are applied using the test data sets:

- accuracy:

$$Acc = (TP + TN)/(TP + TN + FP + FN), \tag{7}$$

- specificity:

$$Spe = TP/(TP + FN), \tag{8}$$

- sensitivity:

$$Sen = TN/(TP + FN), \tag{9}$$

- weighted prediction accuracy:

$$Wpa = (w_1 Acc + w_2 Spe + w_3 Sen)/(w_1 + w_2 + w_3), \tag{10}$$

where TP denotes the number of test patterns correctly assigned to normal operation conditions (true positive), TN is the number of test patterns correctly assigned to fault operation conditions (true negative), FP is the number of test patterns assigned to normal operation conditions while they represent fault operation conditions (false positive), and FN defines the number of test patterns assigned to fault operation conditions while they belong to normal operation conditions (false negative). ($TP+TN+FP+FN$) in (7) is the number of patterns in the test data sets. w_1, w_2 and w_3 in (10) are weights used to determine the importance of contribution of each evaluation criterion to the prediction accuracy of the designed model. These weights are used in order to take into account the effect of imbalanced data discussed before. The accepted prediction accuracy depends on the severity of faults and their impact on the deterioration of produced electrical power and the cost maintenance.

5. Nature-inspired optimization algorithms for the optimal tuning of linear controllers

Several NIOAs are applied to the optimal tuning of linear controllers dedicated to industrial process applications. Some of the latest industrial applications of representative NIOAs will be briefly discussed as follows in relation with the most frequently used proportional-integral (PI), proportional-derivative (PD) and proportional-integral-derivative (PID) controllers.

Particle Swarm Optimization (PSO) is a population-based NIOA initially introduced in [63] and [64]. PSO is inspired by the behaviour of entities observed in flocks of birds or schools of fishes. The movement of the population, characterized by agents, in PSO is guided by simple laws that repeat at each iteration, helping these agents, which represent candidate solutions, flow through the search domain. Each agent has assigned a multidimensional vector that is updated according to the calculated velocity, which takes into consideration the best position explored by the agent and best solution explored by the swarm.

A PSO-based optimal tuning solution for PI controllers in a self-commissioning procedure for direct drives with elastic coupling is proposed in [65]; the solution is implemented, with low computational cost, on a Digital Signal Processor (DSP) and experimentally tested in industrial processes. The PSO-based optimal tuning of PID controllers using an objective function subjected to H_∞ norm to achieve robust stability and disturbance attenuation is given in [66]; the designed H_∞ PID controller is applied to a Single Input-Single Output (SISO) flexible link manipulator. A PSO algorithm characterized by a modified velocity update equation, where an additional factor, namely the best particle of each sub-population, is added to enhance the search ability, is suggested in [67]; this PSO algorithm is applied

to the tuning of PID controllers for Multi Input-Multi Output (MIMO) processes focusing on a chemical process with two inputs and two outputs. The velocity update equation specific to PSO is modified in [68] using a Gaussian distribution; the resulted PSO algorithm is applied to the optimal tuning of multi-loop PI controllers for MIMO systems and applied to a multi-variable quadruple-tank process. A fuzzy gain scheduling control scheme for servo-pneumatic system is proposed in [69]; the local PID controllers are optimally tuned using PSO. A comprehensive review of PSO algorithms' applications in solar photovoltaic systems is conducted in [70]; optimal tuning solutions and real-world applications are included.

Gravitational Search Algorithm (GSA) [71,72] is a relatively newly introduced NIOA inspired by Newtonian physics principals of gravity and interaction between masses [73,74]. Each agent, also referred to as object, interacts with the existing population, by the law of gravity. This interaction is proportional to each agent's mass, expressed in accordance to its fitness, and inversely proportional to the distance between them. The attraction effect between the particles of the universe is also inserted by means of the gravitational constant. The variation of the gravitational constant is modelled by several decrease laws with respect to GSA's iterations.

GSA is applied in [75] to excitation control of wind turbines; the GSA optimizes the switching angles of the inverter targeting the minimization of the total harmonic distortion of the generated voltage. GSA is employed in [76] to optimally tune a damping controller in multi-machine power systems; the simultaneous design of power system stabilizers and thyristor controlled series capacitors is achieved. A modified GSA based on batching mechanism is suggested in [77] to minimize the makespan in a two-stage supply chain; a mixed integer programming model is solved by this GSA and compared with a PSO algorithm and a genetic algorithm.

Simulated Annealing (SA) is an NIOA derived from a metallurgy process which describes the way in which the metal cools and freezes into a minimum energy crystalline structure and the search for a minimum in a more general system [78]. For this process, the selected cooling schedule has a decisive role in the final properties of the substance: if a fast cooling schedule is used the resulting substance will be easily broken due to an imperfect structure, so as to avoid this scenario an appropriate cooling schedule has to be employed, for the resulting structure to be well organized and strong.

An SA algorithm is hybridized with a genetic algorithm in [79] to solve the storage container problem in ports; this hybrid NIOA minimizes the distance between the vessel berthing locations and the storage zone. An SA algorithm is applied in [80] to optimize the parameters of PID controllers for time-delay systems; a suggestive drilling process application is given. The optimal tuning of PI controllers using PSO and SA algorithms is proposed in [81]; optimal PI controllers with a reduced parametric sensitivity for servo systems are designed and experimentally tested.

As shown in [82], the cross-entropy (CE) method translates deterministic optimization problems into stochastic optimization ones and uses rare event simulation techniques. This leads to NIOAs, which have started with adaptive algorithms that estimate the probabilities of rare events in complex stochastic networks on the basis of variance minimization.

The CE-based optimization of the flexible process planning is reported in [83]; the implementation is based on a specific sample representation and probability distribution parameter. Other illustrative applications of NIOAs with focus on quantum inspired algorithms are given in [84].

The specific features of Charged System Search (CSS) algorithms in the context of NIOAs concern the random determination of the initial positions of charged particles, and the initial velocities of the charged particles set to zero [85–87]. A class of PI controllers is optimally tuned by CSS algorithms in [88] to ensure a reduced sensitivity with respect to the variations of the small time constant in nonlinear servo systems; the results of the CSS-based optimization are compared with a PSO algorithm and a GSA.

As shown in [89] and [90], Bacterial Foraging Optimization algorithms (BFOAs) involve evolutionary and chemical operators and steps. Several parameters are involved in this regard as the number of bacteria in the population, the number of chemotaxis steps, the number of swim steps, the number of reproduction steps, the number of elimination-dispersal events, the number of bacteria in the reproduction step, the probability of elimination, and the size of the step taken during each run or tumble.

The BFOA-based optimal tuning of PI controllers for switched reluctance motor drives is proposed in [91]; the PI controllers are applied and implemented in a DSP scheme dedicated to speed control. An adaptive BFOA for active noise control systems is suggested in [92]; the experimental results prove that this algorithm prevents falling into local minima. A performance analysis of PID controllers tuned by BFOA is given in [93]; the application concerns torque motor systems in the automotive industry. A review on NIOAs applied to PID controller tuning is presented in [94]; BFOA and PSO algorithms are included.

The controller tuning based on evolutionary multi-objective optimization is investigated in [95]. Applications of PID, predictive, fuzzy and state feedback controllers are considered.

The analysis presented above shows that PI, PD and PID controllers have proven very good control system performance if these controllers are tuned by NIOAs. In addition, the optimal tuning by NIOAs can compensate for the eventual less experience of the control systems designer. Other popular linear controllers, such as:

- state feedback controllers,
- two-degree-of-freedom (2-DOF) controllers,

that require more experience of the control systems designer, are also tuned by NIOAs. However, the results reported in the literature are validated mainly by digital simulation results, and there is still room for industrial applications and validations.

One of the shortcomings of NIOAs is the large number of evaluations of the objective functions, which affects the industrial applications, where the users are not allowed to conduct many experiments on the real-world processes. In this regard, in order to reduce the number of evaluations of the objective functions and the number of experiments conducted on the real-world control systems, we propose the introduction of the gradient in the update laws for the variables of the NIOAs. This will be presented as follows in connection with the experiment-based solving of a widely used state feedback control problem, namely the Linear Quadratic Regulator (LQR). In addition, the approach given in [96] is carrying out the estimation of the objective function gradients directly on the basis of measurements carried out during the control system operation, and it belongs to data-driven controller tuning techniques.

Let the process be characterized by the single input discrete-time linear time-invariant (LTI) state-space model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \tilde{\mathbf{B}}\mathbf{w}(k), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \tilde{\mathbf{C}}v(k), \end{aligned} \tag{11}$$

where $k \in \mathbf{Z}$, $k \geq 0$ is the discrete time argument, u is the control signal, $\mathbf{x} = [x_1 \dots x_n]^T \in \mathbf{R}^n$ is the state vector, n is the system order, the subscript T indicates the matrix transposition, $\mathbf{y} \in \mathbf{R}^n$ is the controlled output, and $\mathbf{w} \in \mathbf{R}^n$ and $\mathbf{v} \in \mathbf{R}^n$ are the uncorrelated process noise vector and measurement noise vector, respectively.

The LQR-type objective function is

$$I(\boldsymbol{\rho}) = \sum_{k=0}^{\infty} [\mathbf{x}^T(\boldsymbol{\rho}, k) \ \mathbf{Q} \ \mathbf{x}(\boldsymbol{\rho}, k) + \lambda u^2(\boldsymbol{\rho}, k)], \tag{12}$$

where the vector variable of the objective function $\boldsymbol{\rho} = [\rho_1 \dots \rho_n]^T$ is a parameter vector, the state vector and the control signal are parameterized by $\boldsymbol{\rho}$, and the weights are $\mathbf{Q} \geq 0$ and $\lambda > 0$, with $\mathbf{Q} = [q_{ij}]_{i,j=1\dots n}$, $q_{ij} = q_{ji}$, $i,j = 1 \dots n$.

The following optimization problem leads to the optimal parameter vector $\boldsymbol{\rho}^*$ that corresponds to the optimal state feedback gain matrix, referred to also as the gain matrix $(\boldsymbol{\rho}^*)^T$:

$$\boldsymbol{\rho}^* = \underset{\rho}{\operatorname{argmin}} I(\rho). \tag{13}$$

The solution to (7) is the state feedback control law

$$u(\boldsymbol{\rho}^*, k) = -(\boldsymbol{\rho}^*)^T x(\boldsymbol{\rho}^*, k). \tag{14}$$

Adding the reference inputs, the state feedback controller is characterized by

$$\begin{aligned} \mathbf{e}(\boldsymbol{\rho}, k) &= \mathbf{r}(k) - \mathbf{x}(\boldsymbol{\rho}, k), \\ u(\boldsymbol{\rho}, k) &= \boldsymbol{\rho}^T \mathbf{e}(\boldsymbol{\rho}, k), \end{aligned} \tag{15}$$

where $\mathbf{r} = [r_1 \dots r_n]^T$ is the reference input vector, r_i are the reference inputs that correspond to the state variables x_i, $i = 1 \dots n$, and $\mathbf{e} = [e_1 \dots e_n]^T = [r_1 - x_1 \dots r_n - x_n]^T$ is the state control error vector that consists of the state variable errors e_i, $i = 1 \dots n$. The state feedback control system structure is presented in Fig. 8(a), where P is the process and C is the controller.

Using the general stochastic framework considered in [96], the modified objective function is

$$J(\boldsymbol{\rho}) = E\{\sum_{k=0}^{N} [\mathbf{e}^T(\boldsymbol{\rho}, k) \ \mathbf{Q} \ \mathbf{e}(\boldsymbol{\rho}, k) + \lambda \ \mathbf{e}_u^2(\boldsymbol{\rho}, k)]\}, \tag{16}$$

where $E\{\}$ is the expectation with respect to the stochastic disturbances, $e_u(\boldsymbol{\rho}, k)$ is defined as the difference between the control signal and its steady-state value $u(\boldsymbol{\rho}, \infty)$

$$e_u(\boldsymbol{\rho}, k) = u(\boldsymbol{\rho}, k) - u(\boldsymbol{\rho}, \infty), \tag{17}$$

and N is the finite time horizon taken into consideration for practical evaluations of the objective function. For a sufficiently large N, the following equality holds [96]:

$$I(\boldsymbol{\rho}) \approx J(\boldsymbol{\rho}), \tag{18}$$

so the optimization problem (7) is transformed into the optimization problem

$$\boldsymbol{\rho}^* = \arg \min_{\boldsymbol{\rho}} J(\boldsymbol{\rho}). \tag{19}$$

The optimization problem (13) is experimentally solved in [96] in terms of an Iterative Feedback Tuning (IFT) algorithm. The update law of this algorithm is

$$\boldsymbol{\rho}^{i+1} = \boldsymbol{\rho}^i - \gamma^i (\mathbf{R}^i)^{-1} est\left[\frac{\partial J}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^i)\right], \ \mathbf{R}^i > 0, \tag{20}$$

where $i \in \mathbf{Z}$, $i \geq 0$, is the current iteration/experiment index, $\gamma^i > 0$ is the step size, $est[(\partial J/\partial \boldsymbol{\rho})(\boldsymbol{\rho}^i)]$ is the unbiased estimate of the gradient, and the regular matrix $\mathbf{R}^i$ can be the estimate of the Hessian matrix, the Gauss-Newton approximation of the Hessian, or the identity matrix. The steps of the algorithm are

Step 0. The step size, the initial controller parameter vector $\boldsymbol{\rho}^0$ and the weights in the objective function are set.

Step 1. An initial "normal" experiment is conducted using the control system structure shown in Fig. 8(a). The evolution of all state variables is recorded.

Step 2. n gradient experiments are conducted using the experimental setup illustrated in Fig. 8(b) to obtain the partial derivatives $(\partial x_i/\partial \rho_l)$ and $(\partial u/\partial \rho_l)$, where parameter ρ_l, $l = 1 \dots n$, is a certain controller parameter.

Step 3. A "normal" experiment is conducted once more such that the states contain realizations of noise that differ from the noise at *step 2* to ensure the unbiased estimate of the gradient.

Step 4. The estimates of the gradient of the objective functions are computed as [96]

$$\frac{\partial J}{\partial \rho_l} = 2\sum_{k=0}^{N} \left\{ \left[\sum_{\substack{i,j=1 \\ i \geq j}}^{n} \left(q_{ij} \ e_i \frac{\partial e_j}{\partial \rho_l} \right) \right] + \lambda \ e_u \frac{\partial e_u}{\partial \rho_l} \right\} = 0, \ \ l = 1 \dots n. \tag{21}$$

Step 5. The next parameter vector $\boldsymbol{\rho}^{i+1}$ is computed by the update law (20).

Step 0 is done only once. *Steps 1–5* are repeated iteratively until a stopping condition is met.

This approach is very useful in industrial applications because of two reasons. First, the estimation of the gradients is carried out during the real-world control system operation. Second, the number of iterations/experiments is reduced as suggestively illustrated in [96] for a nonlinear servo system application.

As mentioned before, this approach can be combined with NIOAs by adding the following term from (20):

$$-\gamma^i (\mathbf{R}^i)^{-1} est\left[\frac{\partial J}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^i)\right] \tag{22}$$

to the right-hand terms of the update laws for the variables of the NIOAs. Therefore, the combination of NIOAs and data-driven control is expected to merge the advantages of both controller tuning techniques. We also propose the application of NIOAs combined with data-driven control to the tuning of state feedback H_∞ controllers.

The 2-DOF controllers have been proposed because the main tasks in control, namely the achievement of high performance in set-point tracking and the regulation in the presence of disturbance inputs are difficult to be accomplished by means of PI and PID controllers [97–99]. A set of 2-DOF control system structures with focus on PI controllers is presented in Fig. 9, where $P(s)$ is the process transfer function: the set-point filter structure (Fig. 9(a)), the feedforward structure (Fig. 5(b)), the feedback structure (Fig. 9(c)), and the component-separated structure (Fig. 9(d)), where: r–the set-point, y – controlled output, $e = r - y$ or $e = r_1 - y$ – the control error, u – the control signal, r_1, u_2, u_2 and u_4 – the

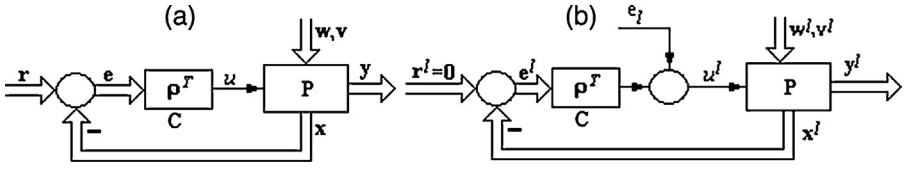

Fig. 8. State feedback control system structure (a) and experimental setup to compute partial derivatives (b).

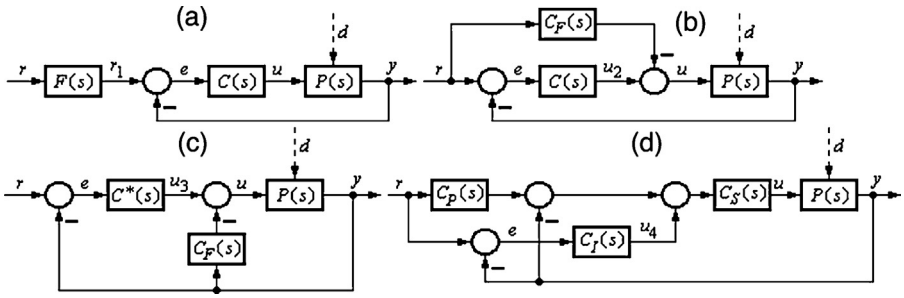

Fig. 9. 2-DOF control system structures: set-point filter 2-DOF control system structure (a), feedforward 2-DOF control system structure (b), feedback 2-DOF control system structure (c), component-separated 2-DOF PI control system structure (d).

outputs of the blocks $F(s)$, $C(s)$ (in Fig. 9 (b)), $C^*(s)$ and $C_I(s)$, respectively, and d – the disturbance input.

However, as shown in [100], the main drawback of 2-DOF controllers is that although they ensure the regulation, the reduction of overshoot is paid by slower set-point responses because the 2-DOF structures can be reduced to feedforward controllers with set-point weighting. A first solution is to introduce fuzzy control in the structure of 2-DOF linear control systems resulting in 2-DOF fuzzy controllers [101–106]. Some of the controller blocks in Fig. 9 can be fuzzified in order to improve the control system performance [105]. We propose a second solution, to tune the parameters of a part of the blocks of 2-DOF controllers by NIOAs. Since the search space of optimal control problems for 2-DOF controllers is large (i.e., more parameters to be tuned backed up by a solid experience of the control systems designer), this shortcoming will be mitigated, thus increasing the interest in industrial and applications of 2-DOF linear and fuzzy controllers tuned by NIOAs. State feedback control system structures can be considered as well.

The algorithms were tested on a case study, where the process is characterized by the following continuous-time time-invariant SISO state-space model, which defines a general class of nonlinear servo systems:

$$m(t)=\begin{cases}-1, & \text{if } u(t)\le -u_b,\\ \dfrac{u(t)+u_c}{u_b-u_c}, & \text{if } -u_b<u(t)<-u_c,\\ 0, & \text{if } -u_c\le |u(t)|\le u_a,\\ \dfrac{u(t)-u_a}{u_b-u_a}, & \text{if } u_a<u(t)<u_b,\\ 1, & \text{if } u(t)\ge u_b,\end{cases} \tag{23}$$

$$\dot{x}_P(t)=\begin{bmatrix}0 & 1\\ 0 & -1/T_\Sigma\end{bmatrix}x_P(t)+\begin{bmatrix}0\\ k_P/T_\Sigma\end{bmatrix}m(t)+\begin{bmatrix}1\\ 0\end{bmatrix}d(t),$$

$$y(t)=[1\quad 0]x_P(t),$$

where t is the continuous time argument, $t\in\mathrm{R}$, $t\ge 0$, k_P is the process gain, T_Σ is the small time constant, the control signal u is a pulse width modulated (PWM) duty cycle, and m is the output of the saturation and dead zone static nonlinearity specific to the actuator. The nonlinearity is modelled by the first equation in (23), with the parameters u_a, u_b and u_c, that are subjected to $0<u_a<u_b$, $0<u_c<u_b$. The state-space model (23) includes the actuator and measuring element dynamics. The state vector $x_P(t)$ is expressed as follows in (angular) position applications for $n=2$:

$$x_P(t)=[x_{P,1}(t)\quad x_{P,2}(t)]^T=[\alpha(t)\quad \omega(t)]^T, \tag{24}$$

where $\alpha(t)$ is the angular position and $\omega(t)$ is the angular speed.

The nonlinearity in (23) is neglected in the following simplified model of the process expressed as the transfer function $P(s)$, which is convenient as it can be used relatively easily in the controller design and tuning:

$$P(s)=\frac{k_{EP}}{s(1+T_\Sigma s)}, \tag{25}$$

where the equivalent process gain is k_{EP}:

$$k_{EP}=\begin{cases}\dfrac{k_P}{u_b-u_c}, & \text{if } -u_b<u(t)<-u_c,\\ \dfrac{k_P}{u_b-u_a}, & \text{if } u_a<u(t)<u_b.\end{cases} \tag{26}$$

The nonlinear servo system model given in (23) is important as illustrative example in many industrial applications. One such experimental setup is illustrated in Fig. 10. An optical encoder is used for the measurement of the angle and a tacho-generator for the measurement of the angular speed. The speed can also be estimated from the angle measurements. The PWM signals that are proportional with the control signal are produced by the actuator in the power interface. The main features of the experimental setup are [107]: rated amplitude of 24 V, rated current of 3.1a, rated torque of 15 N cm, rated speed of 3000 rpm, and weight of inertial load of 2.03 kg. The nominal values of the parameters of the process model given in (23) and (26), obtained by a least squares algorithm, are $u_a=0.15$, $u_b=1$, $u_c=0.15$, $k_P=k_{eP}=140$, and $T_\Sigma=0.92$ s.

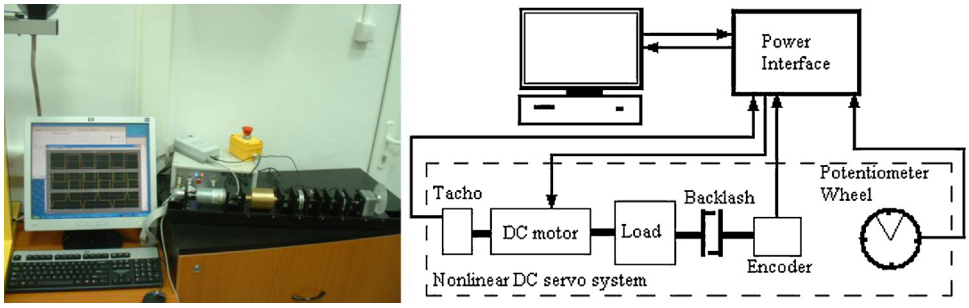

Fig. 10. Nonlinear servo system experimental setup and block diagram.

The NIOAs are mapped onto the optimization problems by the following two aspects:

- the fitness function in the NIOAs are equal to the objective functions in the optimization problems,
- the solution vectors in the NIOAs, which are, for example, the position vectors in case of PSO, GSA and CSS, are equal to the vector variables of the objective functions, which are, in fact, the parameter vectors of the controllers subjected to optimal tuning.

The PI controllers can cope with the process modelled in (25) if they are inserted in 2-DOF linear control system structures given in Fig. 9. The transfer function of the PI controller is

$$C(s) = \frac{k_c(1+sT_i)}{s} = k_C\left(1+\frac{1}{sT_i}\right), \quad k_C = k_cT_i, \tag{27}$$

where k_c is the controller gain and T_i is the integral time constant. The PI controllers can be tuned by the Extended Symmetrical Optimum (ESO) method [108] to guarantee a desired compromise to the performance specifications (i.e., maximum values of control system performance indices) imposed to the control system using a single design parameter referred to as β, with the recommended values within $1 < \beta \leq 20$. The PI tuning conditions specific to the ESO method are

$$k_c = \frac{1}{\beta\sqrt{\beta}k_{EP}T_\Sigma^2}, \quad T_i = \beta T_\Sigma, \quad k_C = \frac{1}{\sqrt{\beta}\ k_{EP}\ T_\Sigma}, \tag{28}$$

and the single design parameter β is next optimally tuned by NIOAs.

CSS algorithms are applied in [109] to ensure the optimal PI controllers with a reduced process gain sensitivity, namely a reduced sensitivity with respect to the variations of the process gain k_P in (23). The following objective function is defined in this regard:

$$I_{2e}^{kP}(\beta) = \int_0^\infty [e^2(t) + \gamma_{kP}^2\sigma_{kP}^2(t)]dt, \tag{29}$$

where $\gamma_{\kappa P}$ is the weighting parameter, $\sigma_{\kappa P}$ and is the output sensitivity function in the appropriately defined state sensitivity model. The expression of the objective function in (29), where the variables in the integral depend on β, allows for the definition of the optimization problem

$$\beta^* = \arg\min_{\beta>1} I_{2e}^{kP}(\beta), \tag{30}$$

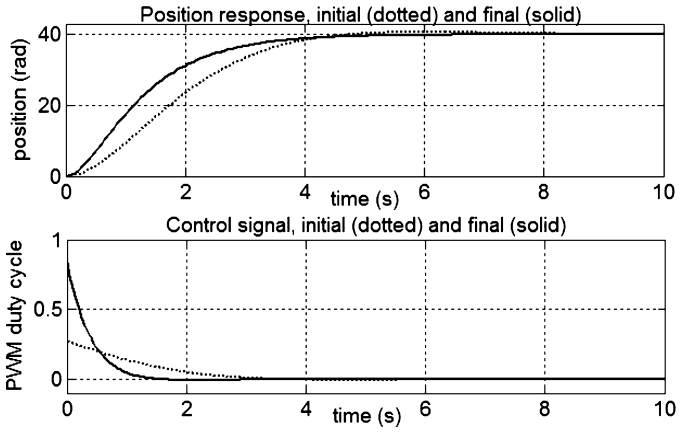

Fig. 11. Control system response with PI controller before and after tuning by CSS algorithm.

where β^* is the optimal value of the design parameter β. Solving (30) guarantees the optimal tuning of PI controllers with a reduced process gain sensitivity. Use is made in [109] of CSS algorithms to solve the optimization problem defined in (30).

A typical control system response with PI controller before and after tuning by the CSS algorithm is presented in Fig. 11. Fig. 11 shows the control system performance improvement.

The illustration of the application of the IFT algorithm is presented as follows for the same nonlinear servo system. The weights in the objective function (16) were set to

$$\mathbf{Q} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad \lambda = 400, \tag{31}$$

and the step angular position reference input of 40 rad was considered. Using the initial step size $\gamma^0 = 2 \times 10^{-8}$, the values of the consequent step sizes according to [96], and $\mathbf{R}^i = \mathbf{I}_2$, where $\mathbf{I}_2$ is the identity matrix, the initial parameters of the state feedback controller are tuned by LQR, and the final parameters of the state feedback controller are considered those obtained by the application of the IFT algorithms using only 15 iterations.

The evolution of the objective function with respect to the iteration number (i.e., during the tuning) and the time responses of the state feedback control system before and after the application of the IFT algorithm are presented in Fig. 12. Fig. 12 highlights the control system performance improvement obtained by few iterations and experiments conducted in the framework of this IFT-based data-driven control technique.

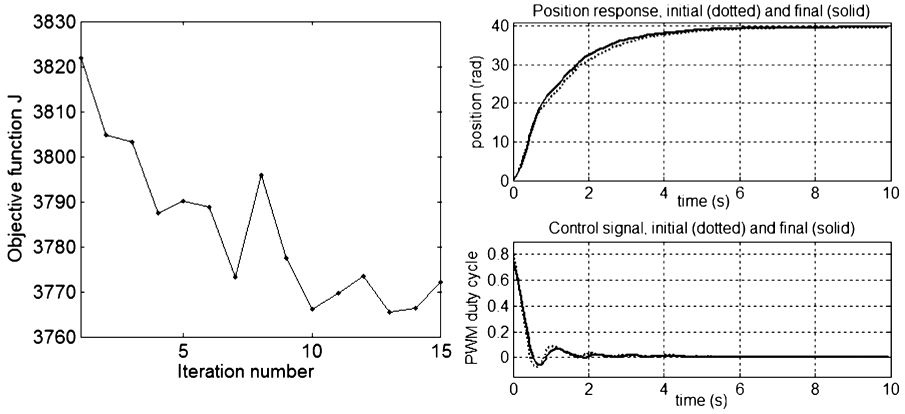

Fig. 12. Control system response with state feedback controller before and after IFT.

6. Nature-inspired optimization algorithms for the optimal tuning of nonlinear controllers

The fuzzy controllers are the most widely used and optimally tuned nonlinear controllers in industrial applications because of threes reasons pointed out in [100]:

- In some situations (for example, the control of processes with functional nonlinearities which subjected to difficult mathematical modelling and the control of ill-defined processes), fuzzy control can be viewed as a viable alternative to classical, crisp control (conventional control).
- Compared to conventional control, fuzzy control can be strongly based and focused on the experience of a human operator, and a fuzzy controller (FC) can model more accurately this experience (in linguistic manner) versus a conventional controller.
- The Takagi-Sugeno fuzzy models represent fuzzy dynamic models or fuzzy systems characterized by two advantages: (i) any model-based technique (including a nonlinear one) can be applied to the fuzzy dynamic models, (ii) the controller itself can be considered as a fuzzy system.

In this regard, the overview will be firstly focused on the nature inspired optimal tuning of FCs. The industrial applications of the NIOAs presented in the previous section will be considered as follows.

The PSO-based tuning of FCs for photovoltaic systems is proposed in [110]; the optimized FCs are designed to achieve maximum power point tracking using a current-mode boost converter with bifurcation control. A multi-objective PSO algorithm is used in [111] to get multiple Pareto optimal solutions in a multi-objective optimization problem for the identification of the membership functions and rules of FCs; the multi-objective PSO algorithm is applied to the optimal tuning of FCs for voltage regulated DC-DC power converters. An evolutionary-group-based PSO algorithm for the optimal tuning of FCs dedicated to mobile robot navigation problems is suggested in [112]; this NIOA is based on building different groups to select parents in crossover operations, particle updates and replacements, and on an adaptive velocity-mutated operation to improve the search ability. A PSO algorithm is applied in [113] to the optimal tuning of PID-FCs in terms of two self-tuning mechanisms; experimental results for DC drives are given. A PSO algorithm is applied in [110] to the tuning of an FC for the maximum power point tracking of a photovoltaic system; a current-mode boost converter with bifurcation control is included.

PSO, GSA and SA algorithms are applied in [114] to tune the parameters of PI-FCs for nonlinear servo systems. FCs with a reduced process parametric sensitivity are suggested and experimentally tested.

The GSA-based tuning of FCs is carried out in [115,116]. The industrial applications of these controllers concern servo systems.

The optimal tuning of FCs in networked control systems using an offline SA algorithm is proposed in [117]; a drilling process application is presented. SA algorithms are applied in [118] to the optimal tuning of FCs for servo systems; experimental results are included.

The CE-based optimization of FCs is suggested in [119]; the optimally tuned FCs are applied to the force regulation in a network-based drilling process control system. CE is applied in [120] to tune the scaling factors of FCs; the designed FCs carry out collision avoidance tasks for unmanned aerial systems. A survey on learning and optimization applied to FC tuning is conducted in [121]; industrial applications or CE are shown.

The CSS-based optimal tuning of FCs is conducted in [122]. FCs with a reduced parametric sensitivity applied to servo systems are proposed and tested by real-time experiments.

A BFOA selects the weights of a fuzzy model predictive control in [123]. The optimally tuned controller is applied to two nonlinear chemical processes.

As shown in [124], the type-2 FCs can outperform the conventional type-1 FCs when the control problems have a high degree of uncertainty. The structure of a type-2 fuzzy control system is presented in Fig. 13. The type-2 fuzzy control system is considered as a single input system with respect to the reference input r and as a single output system with respect to the controlled output y. The second input applied to the controlled process (CP) is the disturbance input d.

Fig. 13 illustrates the operation principle of a type-2 FC, with the following variables and modules: (1) the crisp inputs, (2) the fuzzification module, (3) the fuzzified inputs, (4) the inference module, (5) the type-2 fuzzy conclusions, (6) the type reducer module, (7) the type-1 fuzzy conclusions, (8) the defuzzification module, and (9) the crisp output.

One specific feature of fuzzy control systems concerns the multiple interactions regarded from the process to the controller by the auxiliary variables $\mathbf{y_a}$, gathered in the input vector $\mathbf{e}'$

$$\mathbf{e}' = [e \quad \mathbf{y}_a^T] = [e_1 \quad e_2 \quad \dots \quad e_n]^T. \tag{32}$$

These variables are direct or indirect inputs to the FC. No matter how many inputs to the FC are, the FC should have least one input variable e_1, which corresponds to the control error e

$$e_1 = e = r - y \tag{33}$$

According to Fig. 13, the operation principle of a type-2 FC involves the following sequence of operations:

(a) The crisp input information – the measured variables, the reference input (the set point), the control error – is converted into a fuzzy representation. This operation is called fuzzification of crisp information.
(b) The fuzzified information is processed using the rule base, composed of the fuzzy IF-THEN rules that must be well defined in order to control the given process. The principles to evaluate and process the rule base represent the inference mechanism/engine and the result is the "fuzzy" form of the control signal u, namely the fuzzy control signal.
(c) The fuzzy control signal must be converted into a crisp formulation, with well-specified physical nature, directly understandable and usable by the actuator in order to be capable to control the process. This operation is known under the name of output processing, and it consists of the type reducer and the defuzzification.

Due to the increased number of parameters in the modules of type-2 FCs compared to type-1 FCs, the tuning of type-2 FCs is more difficult. Therefore, NIOAs are employed to cope with the optimal tuning of type-2 FCs. Interesting surveys on NIOAs applied to the optimal tuning of type-2 FCs are conducted in [125,126]; the

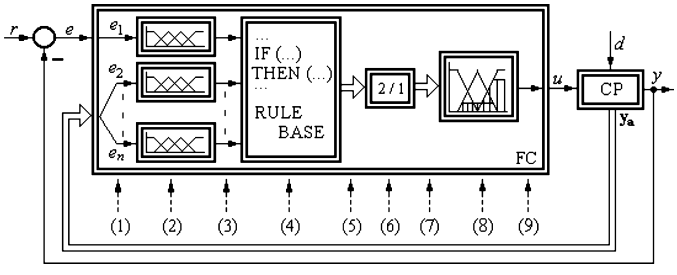

Fig. 13. Type-2 fuzzy control system structure.

effects of the comparisons of different NIOAs are emphasized. The PSO-based tuning of type-2 FCs for DC motor systems is presented in [127]; an attractive hardware implementation is suggested. Type-2 FCs applied to robotics and fault tolerant systems are reported in [128,129].

Other successful industrial applications of NIOAs concern the optimal tuning of neural network controllers and of sliding mode controllers, which are exemplified as follows. These two representative nonlinear controllers can ensure good control system performance, but they exhibit lower interpretability in comparison with FCs.

An SA algorithm is hybridized with a genetic algorithm in [130] and applied to train a feed-forward backpropagation neural network; this hybrid NIOA improves the performance of dressing conditions in grinding processes. The SA algorithm-based optimal tuning of neuro-fuzzy models for a green logistic vehicle routing problem is proposed in [131]; an adaptive neural network is implemented in combination with logistics operators.

A survey on sliding mode controllers for induction motor is conducted in [132]; several applications including the optimal tuning are emphasized. The optimal tuning of sliding mode controllers by BFOA is proposed in [133]; the performance of the sliding mode control system is compared with that of a PID control system in an automotive industry application.

The nature-inspired optimal tuning of nonlinear controllers is illustrated as follows with focus on a Takagi-Sugeno PI-fuzzy controller (T-S PI-FC) for the nonlinear servo system described in the previous section. The T-S PI-FC replaces the block $C(s)$ in the control system structure given Fig. 9(a). The T-S PI-FC is designed starting with the linear PI controller such that to ensure the further improvement of the control system performance for the nonlinear process modelled in (23). The structure and the input membership functions of a simple T-S PI-FC are presented in Fig. 14, where q^{-1} is the backward shift operator.

Fig. 14 points out the increment of control error $\Delta e(k) = e(k) - e(k-1)$ and the increment of control signal $\Delta u(k) = u(k) - u(k-1)$. These increments offer the dynamics of the T-S PI-FC and they result from discretizing the continuous-time PI controller. Tustin's method leads to the incremental form of the discrete-time PI controller:

$$\Delta u(k) = K_P[\Delta e(k) + \mu\, e(k)] \tag{34}$$

and to its parameters

$$K_P = k_c\left(T_i - \frac{T_s}{2}\right), \quad \mu = \frac{2\ T_s}{2\ T_i - T_s} \tag{35}$$

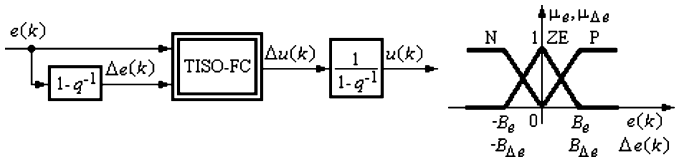

Fig. 14. Structure and input membership functions of Takagi-Sugeno PI-fuzzy controller.

obtained by the model equivalence principle as shown in [105], where T_s is the sampling period set in accordance with the requirements of quasi-continuous digital control.

The Two Inputs-Single Output fuzzy controller (TISO-FC) block presented in Fig. 14 is characterized by the weighted average method in the defuzzification module, and by the SUM and PROD operators in the inference engine. The consequents of the fuzzy control rules are modelled by means of the two terms, $f_{C1}(k)$ and $f_{C2}(k)$:

$$f_{C1}(k) = K_P[\Delta e(k) + \mu e(k)], \quad f_{C2}(k) = \eta\, f_{C1}(k). \tag{36}$$

The parameter η is introduced in (36) to reduce the overshoot of the fuzzy control system and eventually the downshoot, and these appear if $e(k)$ and $\Delta e(k)$ have the same signs. Therefore, the rule base of the TISO-FC block is formulated as

$$\begin{array}{ll} R^1: \text{IF } e(k) \text{ IS N AND } \Delta e(k) \text{ IS N} & \text{THEN } \Delta u(k) = f_{C2}(k), \\ R^2: \text{IF } e(k) \text{ IS P AND } \Delta e(k) \text{ IS P} & \text{THEN } \Delta u(k) = f_{C2}(k), \\ R^3 \ldots R^9: \text{IF } e(k) \text{ AND } \Delta e(k) \text{ DO NOT FULFIL } R^1 \text{ AND } R^2 & \text{THEN } \Delta u(k) = f_{C1}(k). \end{array} \tag{37}$$

The modal equivalence principle results in the following tuning equation, which reduces the number of tuning parameters of the T-S PI-FC:

$$B_{\Delta e} = \mu\ B_e. \tag{38}$$

The application of the ESO method and of the modal equivalence principle leads to only three tuning parameters for the T-S PI-FC. These parameters are included in the controller parameter vector $\boldsymbol{\rho}$, which is also the vector variable of the objective function:

$$\boldsymbol{\rho} = [\,\rho_1 \quad \rho_2 \quad \rho_3\,]^T = [\,\beta \quad B_e \quad \eta\,]^T. \tag{39}$$

As shown in the previous section, a reduced process gain sensitivity is obtained by using the objective function

$$I(\boldsymbol{\rho}) = \sum_{k=0}^{\infty}[e^2(k) + \gamma_{kP}^2\sigma_{kP}^2(k)^2], \tag{40}$$

where the variables in the sum depend on $\boldsymbol{\rho}$. The optimization problem is similar to (30):

$$\boldsymbol{\rho}^* = \arg\min_{\boldsymbol{\rho}\in D_\rho} I(\boldsymbol{\rho}), \tag{41}$$

where $\boldsymbol{\rho}^*$ is the optimal value of the controller parameter vector $\boldsymbol{\rho}$, and $D_{\boldsymbol{\rho}}$ is the feasible domain of $\boldsymbol{\rho}$. Solving (41) guarantees the optimal tuning of T-S PI-FCs with a reduced process gain sensitivity. GSA, PSO and SA algorithms are used in [114] to solve the optimization problem defined in (41).

A sample of real-time experimental results obtained for the fuzzy control system with the SA-based optimized T-S PI-FC is presented in Fig. 15. The results are obtained after 50 iterations of the SA algorithms, and they illustrate good control system performance that can be improved if more iterations of the SA algorithm are conducted.

7. Research challenges in fault diagnosis and nature-inspired optimal control

Fault diagnosis in wind turbines is a challenging task. This is due principally to their complex nonlinear dynamics and to their strong non-stationary environments. Data mining and machines learning methods represent an efficient solution to answer this challenge. However, they suffer from several drawbacks:

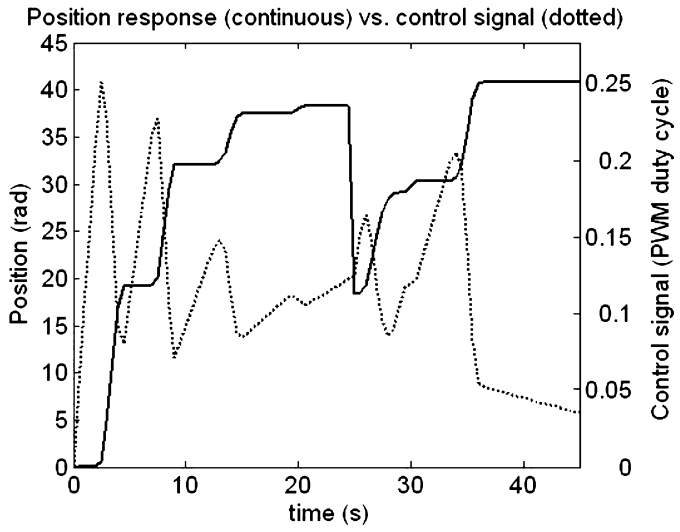

Fig. 15. Real-time experimental results for of the fuzzy control system with the SA-based optimized T-S PI-FC.

- they require a sufficient number of patterns according to each fault behaviour in order to obtain an efficient diagnosis model,
- they are usually insensitive to the occurrence of undefined or unpredicted fault or dangerous behaviour,
- since the obtained models are not transparent, the obtained results are hard to be interpreted and demonstrated,
- they are not adapted to detect drift-like faults representing the component degradation.

Therefore, it is justified to develop an advanced systematic methodology and architecture of fault diagnosis in wind turbines that should be able to:

- separate any abnormal change caused by components degradation from normal change due to environmental (e.g., weather conditions) or load (e.g., electricity network status) effects,
- describe wind turbine dynamical behaviours (normal/degraded/faulty) without the need to depth a priori knowledge, and
- provide complete and coherent explanation of the observed fault behaviour by providing two complementary forms of the global diagnosis decision; the first form is a sequence of states and actions in order to explain what happened to the system, while the second form is the fault type label in order to identify what is wrong with the system.

One solution to achieve these tasks is the use of dynamical machine learning and data mining approaches. In these approaches, the model parameters and structure are updated continuously according to the novelties and changes in either its internal dynamical states or in its environment conditions. This update enables a continuous learning of the system behaviours leading to improve or at least to maintain its performance over time.

The main drawbacks of NIOAs, that can be also considered as challenges of NIOAs applied to the tuning of fuzzy controllers in industrial applications, are twofold:

- they require a large number of evaluations of the objective functions in the optimization (that correspond to the fitness functions in the NIOAs), which requires a large number of experiments conducted on the real-world control systems, and this is critical in industry,
- the presence of random parameters in the NIOAs, which affects the results that cannot be transferred to industry because they could not be repeatable after several experiments/trials.

A convenient approach to cope with the first drawback is to introduce, as outlined in Section 5, the gradient in the update laws for the variables of the NIOAs (i.e., the tuning parameters of the controllers). This can be associated with the experiment-based estimation of the gradient in the framework of data-driven control. The main data-driven techniques are Iterative Feedback Tuning [134,135], Correlation-based Tuning [136,137], Simultaneous Perturbation Stochastic Approximation [138,139], data-driven predictive control [140,141], Model-free Control [142,143], Model-free Adaptive Control [144,145], unfalsified control [146], [147], and adaptive online Iterative Feedback Tuning [148]. The number of experiments conducted on the real-world processes needed in the implementation these techniques must be reduced, and this will lead to novel experimental schemes and control structures.

As shown in [149,150], chaos describes the complex behaviour of nonlinear deterministic systems. Due to the non-repetition of chaos, the introduction of chaos in NIOAs can carry out overall searches at higher speeds than stochastic ergodic searches that depend on probabilities. For example, the application of chaotic sequences in [150,151] instead of random sequences in PSO algorithms leads to the diversification of the population of particles and improves PSO algorithms' performance in preventing premature convergence to local minima. That is the reason why a first way to cope with the second drawback is to generate the random parameters in NIOAs by chaotic maps. The eventual stabilization of the chaotic maps can also be considered, and some recent examples of chaotic maps considered as nonlinear dynamic systems stabilized by fuzzy control are given in [152–154].

Another way to mitigate the effects of the second drawback is to modify the objective functions by adding weighted terms that result from the definition of sensitivity models with respect to the random parameters in the NIOAs. Let the general expressions of the velocity and position update laws in an NIOA be

$$\begin{aligned} v_i(k+1) &= \mathbf{f}(v_i(k), \alpha, \mathbf{a}_{\text{var},i}(k)), \\ \mathbf{x}_i(k+1) &= \mathbf{x}_i(k) + v_i(k), \end{aligned} \tag{42}$$

where k is the index of the current iteration, the subscript i, $i = 1 \dots N$, indicates ith agent in the population that consists of the total number of N agents, $v_i(k) = [\, v_i^1(k) \quad \dots \quad v_i^d(k) \quad \dots \quad v_i^q(k) \,]^T \in \mathbf{R}^q$ is the velocity vector of ith agent, $v_i^d(k)$ is the velocity of ith agent in dth dimension, $d = 1 \dots q$, q is the dimension of the search space, i.e., the number of variables of the objective function, $\mathbf{x}_i(k) = [\, x_i^1(k) \quad \dots \quad x_i^d(k) \quad \dots \quad x_i^q(k) \,]^T \in \mathbf{R}^q$ is the position vector of ith agent, $x_i^d(k)$ is the position of ith agent in dth dimension, $d = 1 \dots q$, $a_{\text{var},i}$ is the vector that includes all variables specific to the NIOA (accelerations, forces, etc.), $\boldsymbol{\alpha} = [\, \alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_m \,]^T \in R^{\mathbf{m}}$ is the random parameter vector of the NIOA, with the elements α_a, $a = 1 \dots m$, which are the random parameters in the NIOA, and the vector function the function $\mathbf{f}$ must be differentiable with respect to the random parameters α_a, $a = 1 \dots m$.

Eq. (42) can be viewed as the state-space equations of a nonlinear discrete-time dynamical system in the iteration domain, with the state vector

$$[\, (v_i(k))^T \quad (\mathbf{x}_i(k))^T \,]^T = [\, x_{1,i}(k) \quad \dots \quad x_{b,i}(k) \quad \dots \quad x_{2q,i}(k) \,] \in \mathbf{R}^{2q}. \tag{43}$$

In this context, the state sensitivity functions $\lambda_b^{\alpha_a}$, $b = 1 \dots 2q$, are defined as follows:

$$\lambda_b^{\alpha_a} = \left[\frac{\partial x_{b,i}}{\partial \alpha_a} \right]_{\alpha_a, 0}, \quad b = 1 \dots 2q, \quad a = 1 \dots m, \tag{44}$$

where the subscript 0 indicates the nominal value of the random parameter α_a, $a = 1 \ldots m$. Adding one or more properly weighted terms of type $(\lambda_b^{\alpha_a})^2$, $b = 1 \ldots 2q$, or $|\lambda_b^{\alpha_a}|$, $b = 1 \ldots 2q$, in the objective functions, will result in NIOAs with a reduced sensitivity with respect to their random parameters.

This suggested sensitivity reduction approach is more general, and it can be combined with several other objective functions that come from the field of industrial applications. Such objective functions are presented in various forms in [155–165].

All optimization problems should account for constraints imposed from the specific industrial environment. The formulation of constraints depends on the industrial environments related to the optimization problems and/or on specific systems engineering analysis aspects [166–173].

8. Conclusions

This paper has addressed a brief overview on fault diagnosis and on nature-inspired optimal control in industrial applications. Recent developments in machine learning, data mining and evolving soft computing techniques for fault diagnosis have been considered.

The theoretical results are accompanied by illustrative industrial process applications. These examples concern a real liquid level control application, wind turbines and a nonlinear servo system.

This paper has proposed the following classification of internal methods for the fault diagnosis of wind turbines:

- parameter identification,
- observer- and state-based approaches,
- signal- or feature-based approaches.

A lot of NIOAs applied to the tuning of the parameters of controllers in industrial applications are known and reported today. This paper has highlighted just part of them, the most recent ones, and only those that treat concrete industrial applications. That is the reason why genetic algorithms have not been presented, and other NIOAs have been shown: PSO, GSA, SA, CE, CSS and BFOA. The authors consider that it is necessary to understand the basics of the operating principle and of the mathematical characterizations of the fuzzy controllers that are subjected to optimal tuning.

The nature-inspired optimal tuning of controllers has been divided in linear and nonlinear controllers. A trade-off to these controllers is represented by the Tensor Product (TP) model transformation that exhibits two advantages proven in many applications [174–183]:

- Although the transfer function of the product decision operator-based Takagi-Sugeno fuzzy models and the function of the TP model is generally the same, there is an important difference. This difference comes from the fact that the Takagi-Sugeno fuzzy model means a fuzzy combination of locally linearized Linear Time-Invariant (LTI) models, where the locality is expressed by the shape of the antecedent fuzzy sets. But, in case of TP model the weighting functions (that correspond to the membership functions in the fuzzy models) may not have locality, they spread in the whole interval of interest, so as the LTI components of the model cannot readily be assigned to a definite operation point. They are mostly vertexes of a polytopic structure as emphasized in [100].
- It allows the modification of the parameter varying convex combination according to the designer's option. The type of convex combination influences the further Linear Matrix Inequality (LMI) design and, therefore, the control system performance. Hence, the design can employ the manipulation of the convex hull specific to TP model transformation beside the manipulation of the LMIs, so more flexibility is achieved.

The presentation of rather real-time experiments instead of digital simulation results is a perspective of NIOAs. In this context the popularity of NIOAs in the optimal tuning of controllers will increase only if future applications as, for example, those from [184–189], will exhibit significantly better performance compared to existing ones.

The combination of fault diagnosis and NIOAs is one of the perspectives that the authors are suggesting. On the one hand, the NIOAs can be applied in the optimal tuning of the models involved in fault diagnosis. On the other hand, the fault diagnosis methods can lead to conditions that can be incorporated as additional constraints in the design and implementation of optimal controllers. This will attract both researchers and practitioners, and will represent a guarantee for future successful industrial applications.

Acknowledgements

This work was supported by a grant from the Partnerships in priority areas – PN II programme of the Romanian National Authority for Scientific Research ANCS, CNDI – UEFISCDI, project number PN-II-PT-PCCA-2011-3.2-0732, and by grants from the Partnerships in priority areas – PN II programme of the Romanian Ministry of National Education (MEN) – the Executive Agency for Higher Education, Research, Development and Innovation Funding (UEFISCDI), project numbers PN-II-PT-PCCA-2013-4-0544 and PN-II-PT-PCCA-2013-4-0070. The authors would also like to thank the colleagues and friends Dr. Mircea-Bogdan Radac, Prof. Stefan Preitl, Dr. Radu-Codrut David, Prof. Peter Baranyi, Prof. Igor Škrjanc, Prof. Sašo Blažič, Dr. Marius L. Tomescu and Prof. Hans Hellendoorn for their excellent co-operation and inspired discussions.

References

[1] V. Venkatasubramanian, K. Rengaswamy, K. Yin, S.N. Kavuri, A review of process fault detection and diagnosis: Part I: Quantitative model-based methods, Computers & Chemical Engineering 27 (3) (2003) 293–311.
[2] P. Wang, C. Guo, Based on the coal mine's essential safety management system of safety accident cause analysis, American Journal of Environment, Energy and Power Research 1 (3) (2013) 62–68.
[3] P.B. Dongre, L.G. Malik, A review on real time data stream classification and adapting to various concept drift scenarios, in: Proceedings of 2014 IEEE International Advance Computing Conference, Gurgaon, India, (2014), pp. 533–537.
[4] P. Angelov, N. Kasabov, Evolving intelligent systems, eIS, IEEE SMC eNewsLetter 19 (1) (2006) 1–13.
[5] D.R.C. Silva, Sistema de Detecção e Isolamento de Falhas em Sistemas Dinâmicos Baseado em Identificação Paramétrica, (Fault Detection and Isolation in Dynamic Systems Based on Parametric Identification), PhD Thesis, Federal University of Rio Grande do Norte (UFRN), Brazil, 2008.
[6] S. Donders, Fault Detection and Identification for Wind Turbine Systems: A closed-loop analysis, MSc Thesis, University of Twente, The Netherlands, 2002.
[7] P.M. Frank, B. Köppen-Seliger, Fuzzy logic and neural network applications to fault diagnosis, International Journal of Approximate Reasoning 16 (1) (1997) 67–88.
[8] R. Chiong (Ed.), Nature-Inspired Algorithms for Optimisation, Springer-Verlag, New York/Berlin/Heidelberg, 2009.
[9] L. Bianchi, M. Dorigo, L.M. Gambardella, W.J. Gutjahr, A survey on metaheuristics for stochastic combinatorial optimization, Natural Computing 8 (2) (2009) 239–287.
[10] D. Karaboga, B. Akay, A survey: algorithms simulating bee swarm intelligence, Artificial Intelligence Review 31 (1–4) (2009) 61–85.
[11] A. Biswas, K.K. Mishra, S. Tiwari, A.K. Misra, Physics-inspired optimization algorithms: a survey, Journal of Optimization (2013) 1–16, Article ID 438152.
[12] X.-S. Yang, Nature-Inspired Optimization Algorithms, Elsevier, London/Waltham, MA, 2014.
[13] F. Valdez, P. Melin, O. Castillo, A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation, Expert Systems with Applications 41 (2014) 6459–6466.
[14] G. Yin, Y.-T. Zhang, Z.-N. Li, G.-Q. Ren, H.-B. Fan, Online fault diagnosis method based on incremental support vector data description and extreme learning

machine with incremental output structure, Neurocomputing 128 (2014) 224–231.
[15] P.K. Wong, Z. Yang, C.M. Vong, J. Zhong, Real-time fault diagnosis for gas turbine generator systems using extreme learning machine, Neurocomputing 128 (2014) 249–257.
[16] D. Fernández-Francos, D. Martínez-Rego, O. Fontenla-Romero, A. Alonso-Betanzos, Automatic bearing fault diagnosis based on one-class ν-SVM, Computers & Industrial Engineering 64 (1) (2013) 357–365.
[17] V. Muralidharan, V. Sugumaran, V. Indira, Fault diagnosis of monoblock centrifugal pump using SVM, Engineering Science and Technology: An International Journal 17 (3) (2014) 152–157.
[18] W. Liu, Z. Wang, J. Han, G. Wang, Wind turbine fault diagnosis method based on diagonal spectrum and clustering binary tree SVM, Renewable Energy 50 (2013) 1–6.
[19] K. Salahshoor, M. Kordestani, M.S. Khoshro, Fault detection and diagnosis of an industrial steam turbine using fusion of SVM (support vector machine) and ANFIS (adaptive neuro-fuzzy inference system) classifiers, Energy 25 (12) (2010) 5472–5482.
[20] X. Jin, F. Yuan, T.W.S. Chow, M. Zhao, Weighted local and global regressive mapping: a new manifold learning method for machine fault classification, Engineering Applications of Artificial Intelligence 30 (2014) 118–128.
[21] C. Li, J. Zhou, Semi-supervised weighted kernel clustering based on gravitational search for fault diagnosis, ISA Transactions 53 (5) (2014) 1534–1543.
[22] M. Petković, M.R. Rapaić, Z.D. Jeličić, A. Pisano, On-line adaptive clustering for process monitoring and fault detection, Expert Systems with Applications 39 (11) (2012) 10226–10235.
[23] Y. Chen, Y. Yu, Z. Peng, The application of data mining for marine diesel engine fault detection, in: Proceedings of 9th International Conference on Fuzzy Systems and Knowledge Discovery, Sichuan, China, (2012), pp. 1430–1433.
[24] A. Lemos, W. Caminhas, F. Gomide, Adaptive fault detection and diagnosis using an evolving fuzzy classifier, Information Sciences 220 (2013) 64–85.
[25] M. El-Koujok, M. Benammar, N. Meskin, M. Al-Naemi, R. Langari, Multiple sensor fault diagnosis by evolving data-driven approach, Information Sciences 259 (2014) 346–358.
[26] Y. Zhou, J. Liu, A.L. Dexter, Estimation of an incipient fault using an adaptive neurofuzzy sliding-mode observer, Energy and Buildings 77 (2014) 256–269.
[27] A. Alzghoul, M. Löfstrand, Addressing concept drift to improve system availability by updating one-class data-driven models, Evolving Systems (2014), http://dx.doi.org/10.1007/s12530-014-9107-z.
[28] D. Chivala, L.F. Mendonça, J.M.C. Sousa, J.M.G. Sá da Costa, Application of evolving fuzzy modeling to fault tolerant control, Evolving Systems 1 (4) (2010) 209–223.
[29] B.S.J. Costa, P.P. Angelov, L.A. Guedes, Real-time fault detection using recursive density estimation, Journal of Control, Automation and Electrical Systems 25 (4) (2014) 428–437.
[30] B.S.J. Costa, P.P. Angelov, L.A. Guedes, Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier, Neurocomputing 150 (2015) 289–303.
[31] B.S.J. Costa, P.P. Angelov, L.A. Guedes, A new unsupervised approach to fault detection and identification, in: Proceedings of 2014 International Joint Conference on Neural Networks, Beijing, China, (2014), pp. 1557–1564.
[32] P. Angelov, Autonomous Learning Systems: From Data to Knowledge in Real Time, John Willey and Sons, Chichester, UK, 2013.
[33] P. Angelov, X. Zhou, Evolving fuzzy-rule-based classifiers from data streams, IEEE Transactions on Fuzzy Systems 16 (6) (2008) 1462–1475.
[34] P. Angelov, R. Yager, A new type of simplified fuzzy rule-based systems, International Journal of General Systems 41 (2) (2012) 163–185.
[35] A. Marins, Bancada de processo continuo, Manual tecnico, De Lorenzo do Brasil, Brasil, 2009.
[36] B.S.J. Costa, C.G. Bezerra, L.A.H.G. de Oliveira, A multistage fuzzy controller: toolbox for industrial applications, in: Proceedings of 2012 IEEE International Conference on Industrial Technology, Athens, Greece, (2012), pp. 1142–1147.
[37] How Stuff Works, Horizontal-axis Turbine, 2006, http://static.howstuffworks.com/gif/wind-power-horizontal.gif.
[38] The Encyclopedia of Alternative Energy and Sustainable Living, Wind Turbine (2005). http://www.daviddarling.info/encyclopedia/W/AE_wind_turbine.html.
[39] D. Milborrow, Operation and maintenance costs compared and revealed, Wind Stats Newsletter 19 (3) (2006).
[40] Caithness Windfarm Information Forum, Summary of Wind Turbine Accident data to 30 June 2014, 2014.
[41] A. Kusiak, A. Verma, A data-driven approach for monitoring blade pitch faults in wind turbines, IEEE Transactions on Sustainable Energy 2 (1) (2011) 87–96.
[42] A. Verma, A. Kusiak, Fault monitoring of wind turbine generator brushes: a data mining approach, Journal of Solar Energy Engineering 134 (2012) 1–9.
[43] T.W. Verbruggen, Wind turbine operation & maintenance based on condition monitoring WT-Ω, Final report ECN-C–03-047, ECN, Petten, The Netherlands, 2003.
[44] F.P.G. Márquez, A.M. Tobias, J.M.P. Pérez, M. Papaelias, Condition monitoring of wind turbines: techniques and methods, Renewable Energy 46 (2012) 16–178.
[45] S. Simani, P. Castaldi, M. Bonfe, Hybrid model-based fault detection of wind turbine sensors, in: Proceedings of 18th IFAC World Congress, Milano, Italy, (2011), pp. 7061–7066.
[46] W. Chen, S.X. Ding, A. Haghani, A.S. Naik, A.Q. Khan, S. Yin, Observer-based FDI schemes for wind turbine benchmark, in: Proceedings of 18th IFAC World Congress, Milano, Italy, (2011), pp. 7073–7078.
[47] F. Gustafsson, Adaptive Filtering and Change Detection, John Wiley & Sons, New York, 2000.
[48] A.A. Ozdemir, P. Seiler, G.J. Balas, Wind turbine fault detection using counter-based residual thresholding, in: Proceedings of 18th IFAC World Congress, Milano, Italy, (2011), pp. 8289–8294.
[49] P.F. Odgaard, J. Stoustrup, Unknown input observer-based detection of sensor faults in a wind turbine, in: Proceedings of 2010 IEEE International Conference on Control Applications, Yokohama, Japan, (2010), pp. 310–315.
[50] W. Yang, R. Court, J. Jiang, Wind turbine condition monitoring by the approach of SCADA data analysis, Renewable Energy 53 (2013) 365–376.
[51] A. Kusiak, W. Li, The prediction and diagnosis of wind turbine faults, Renewable Energy 36 (1) (2011) 16–23.
[52] E.E. Kostandyan, J.D. Sørensen, Physics of failure as a basis for solder elements reliability assessment in wind turbines, Reliability Engineering & System Safety 108 (2012) 100–107.
[53] A. Kusiak, A. Verma, Monitoring wind farms with performance curves, IEEE Transactions on Sustainable Energy 4 (1) (2013) 192–199.
[54] H. Toubakh, M. Sayed-Mouchaweh, E. Duveilla, Advanced pattern recognition approach for fault diagnosis of wind turbines, in: Proceedings of 12th IEEE International Conference on Machine Learning and Applications, vol. 2, Miami, FL, USA, (2013), pp. 368–373.
[55] H. Toubakh, M. Sayed-Mouchaweh, Advanced data mining approach for wind turbines fault prediction, in: Proceedings of Second European Conference of the Prognostics and Health Management Society, Nantes, France, (2014), pp. 288–296.
[56] K. Kim, G. Parthasarathy, O. Uluyol, W. Foslien, S. Sheng, P. Fleming, Use of SCADA data for failure detection in wind turbines, in: Proceedings of 2011 Energy Sustainability Conference and Fuel Cell Conference, Washington DC, USA, (2011), pp. 2071–2079.
[57] M. Schlechtingen, I. Ferreira Santos, Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection, Mechanical Systems and Signal Processing 25 (5) (2011) 1849–1875.
[58] J. Zeng, D. Lu, Y. Zhao, Z. Zhang, W. Qiao, X. Gong, Wind turbine fault detection and isolation using support vector machine and a residual-based method, in: Proceedings of 2013 American Control Conference, Washington, DC, (2013), pp. 3667–13672.
[59] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[60] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, B. Wu, Intelligent Fault Diagnosis and Prognosis for Engineering Systems, John Wiley & Sons, New York, 2006.
[61] N. Laouti, N. Sheibat-Othman, S. Othman, Support vector machines for fault detection in wind turbines, in: Proceedings of 18th IFAC World Congress, Milano, Italy, (2011), pp. 7067–7072.
[62] R.O. Duda, P.E. Hart, D.E. Stork, Pattern Classification, 2nd ed., John Wiley & Sons, New York, 2001.
[63] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of 1995 IEEE International Conference on Neural Networks, Perth, Australia, (1995), pp. 1942–1948.
[64] J. Kennedy, R.C. Eberhart, A new optimizer using particle swarm theory, in: Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, (1995), pp. 39–43.
[65] M. Calvini, A. Formentini, G. Maragliano, M. Marchesoni, Self-commissioning of direct drive systems, in: Proceedings of 2012 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Sorrento, Italy, (2012), pp. 1348–1353.
[66] M. Zamani, N. Sadati, M. Karimi Ghartemani, Design of an H_∞ PID controller using particle swarm optimization, International Journal of Control, Automation and Systems 7 (2) (2009) 273–280.
[67] W.-D. Chang, C.-Y. Chen, PID controller design for MIMO processes using improved particle swarm optimization, Circuits, Systems, and Signal Processing 33 (5) (2014) 1473–1490.
[68] L. dos Santos Coelho, H.V.H. Ayala, N. Nedjah, L. de Macedo Mourelle, Multi-objective Gaussian particle swarm approach applied to multi-loop PI controller tuning of a quadruple-tank system, in: N.L Nedjah, L dos Santos Coelho, Mourelle de Macedo (Eds.), Multi-Objective Swarm Intelligent Systems, Studies in Computational Intelligence, vol. 261, Springer-Verlag, Berlin/Heidelberg, 2010, pp. 1–16.
[69] S. Kaitwanidvilai, P. Olranthichachat, Robust loop shaping-fuzzy gain scheduling control of a servo-pneumatic system using particle swarm optimization approach, Mechatronics 21 (1) (2011) 11–21.
[70] A. Khare, S. Rangnekar, A review of particle swarm optimization and its applications in solar photovoltaic system, Applied Soft Computing 13 (5) (2013) 2997–3006.
[71] E. Rashedi, Gravitational search algorithm, M.Sc. thesis, Shahid Bahonar University of Kerman, Kerman, Iran, 2007.
[72] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Information Sciences 179 (13) (2009) 2232–2248.
[73] D. Holliday, R. Resnick, J. Walker, Fundamentals of Physics, 7th ed., John Wiley & Sons, Hoboken, NJ, 2005.
[74] B. Schutz, Gravity from the Ground Up, Cambridge University Press, Cambridge, 2003.
[75] A. Chatterjee, K. Roy, D. Chatterjee, A Gravitational Search Algorithm (GSA) based Photo-Voltaic (PV) excitation control strategy for single phase operation of three phase wind-turbine coupled induction generator, Energy 74 (2014) 707–718.
[76] M. Eslami, H. Shareef, A. Mohamed, M. Khajehzadeh, Gravitational search algorithm for coordinated design of PSS and TCSC as damping controller, Journal of Central South University 19 (4) (2012) 923–932.

[77] J. Pei, X. Liu, P.M. Pardalos, W. Fan, S. Yang, L. Wang, Application of an effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain, The International Journal of Advanced Manufacturing Technology 70 (1–4) (2014) 335–348.
[78] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[79] R. Moussi, N.F. Ndiaye, A. Yassine, Hybrid Genetic Simulated Annealing Algorithm (HGSAA) to solve storage container problem in port, in: J.-S. Pan, S.-M. Chen, N.T. Nguyen (Eds.), Intelligent Information and Database Systems, Lecture Notes in Computer Science, vol. 7197, Springer-Verlag, Berlin/Heidelberg, 2012, pp. 301–310.
[80] R.E. Haber, R. Haber-Haber, R.M. del Toro, J.R. Alique, Using simulated annealing for optimal tuning of a PID controller for time-delay systems. An application to a high-performance drilling process, in: F. Sandoval, A. Prieto, J. Cabestany, M. Graña (Eds.), Computational and Ambient Intelligence Ambient Intelligence, Lecture Notes in Computer Science, vol. 4507, Springer-Verlag, Berlin/Heidelberg, 2007, pp. 1155–1162.
[81] R.-E. Precup, R.-C. David, S. Preitl, E.M. Petriu, J.K. Tar, Optimal control systems with reduced parametric sensitivity based on particle swarm optimization and simulated annealing, in: M. Köppen, G. Schaefer, A. Abraham (Eds.), Intelligent Computational Optimization in Engineering, Studies in Computational Intelligence, vol. 366, Springer-Verlag, Berlin/Heidelberg, 2011, pp. 177–207.
[82] R.Y. Rubinstein, Optimization of computer simulation models with rare events, European Journal of Operations Research 99 (1) (1997) 89–112.
[83] S. Lv, L. Qiao, A cross-entropy-based approach for the optimization of flexible process planning, The International Journal of Advanced Manufacturing Technology 68 (9–12) (2013) 2099–2110.
[84] A. Manju, M.J. Nigam, Applications of quantum inspired computational intelligence: a survey, Artificial Intelligence Review 42 (1) (2014) 79–156.
[85] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mechanica 213 (3–4) (2010) 267–289.
[86] A. Kaveh, S. Talatahari, Optimal design of truss structures via the charged system search algorithm, Structural Multidisciplinary Optimization 37 (6) (2010) 893–911.
[87] A. Kaveh, S. Talatahari, A charged system search with a fly to boundary method for discrete optimum design of truss structures, Asian Journal of Civil Engineering (Building and Housing) 11 (3) (2010) 277–293.
[88] R.-C. David, R.-E. Precup, S. Preitl, J.K. Tar, J. Fodor, Three evolutionary optimization algorithms in PI controller tuning, in: R.-E. Precup, S. Kovács, S. Preitl, E.M. Petriu (Eds.), Topics in Intelligent Engineering and Informatics, Applied Computational Intelligence in Engineering and Information Technology, vol. 1, Springer-Verlag, Berlin/Heidelberg, 2012, pp. 95–106.
[89] S. Das, A. Biswas, S. Dasgupta, A. Abraham, Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications, in: A. Abraham, A.-E. Hassanien, P. Siarry, A. Engelbrecht (Eds.), Foundations of Computational Intelligence, Studies in Computational Intelligence, vol. 203, Springer-Verlag, Berlin/Heidelberg, 2009, pp. 23–55.
[90] K.M. Passino, Bacterial foraging optimization, International Journal of Swarm Intelligence Research 1 (1) (2010) 1–16.
[91] E. Daryabeigi, B.M. Dehkordi, Smart bacterial foraging algorithm based controller for speed control of switched reluctance motor drives, International Journal of Electrical Power & Energy Systems 62 (2014) 364–373.
[92] S. Gholami-Boroujeny, M. Eshghi, Active noise control using an adaptive bacterial foraging optimization algorithm, Signal, Image and Video Processing 8 (8) (2014) 1507–1516.
[93] R.-E. Precup, A.-L. Borza, M.-B. Radac, E.M. Petriu, Performance analysis of torque motor systems with PID controllers tuned by bacterial foraging optimization algorithms, in: Proceedings of 2014 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, Ottawa, ON, Canada, (2014), pp. 141–146.
[94] S. Ghosal, R. Darbar, B. Neogi, A. Das, D.N. Tibarewala, Application of swarm intelligence computation techniques in PID controller tuning: a review, in: S.C. Satapathy, Avadhani P.S., Abraham A. (Eds.), Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012, Advances in Intelligent and Soft Computing Volume, vol. 132, Springer-Verlag, Berlin/Heidelberg, 2012, pp. 195–208.
[95] G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez, Controller tuning using evolutionary multi-objective optimisation: current trends and applications, Control Engineering Practice 28 (2014) 58–73.
[96] M.-B. Radac, R.-E. Precup, E.M. Petriu, S. Preitl, Experiment-based performance improvement of state feedback control systems for single input processes, Acta Polytechnica Hungarica 10 (1) (2013) 5–24.
[97] G. Prashanti, M. Chidambaram, Set-point weighted PID controllers for unstable systems, Journal of The Franklin Institute 337 (2–3) (2000) 201–215.
[98] M. Araki, H. Taguchi, Two-degree-of-freedom PID controllers, International Journal of Control, Automation and Systems 1 (4) (2003) 401–411.
[99] A. Visioli, A new design for a PID plus feedforward controller, Journal of Process Control 14 (4) (2004) 457–463.
[100] R.-E. Precup, H. Hellendoorn, A survey on industrial applications of fuzzy control, Computers in Industry 62 (3) (2011) 213–226.
[101] C.M. Liaw, S.Y. Cheng, Fuzzy two-degrees-of-freedom speed controller for motor drives, IEEE Transactions on Industrial Electronics 42 (2) (1995) 209–216.
[102] A. Visioli, Fuzzy logic based set-point weight tuning of PID controllers, IEEE Transactions on Systems, Man, and Cybernetics: Part A 29 (6) (1999) 587–592.
[103] R.K. Barai, K. Nonami, Optimal two-degree-of-freedom fuzzy control for locomotion control of a hydraulically actuated hexapod robot, Information Sciences 177 (8) (2007) 1892–1915.
[104] R.-E. Precup, S. Preitl, P. Korondi, Fuzzy controllers with maximum sensitivity for servosystems, IEEE Transactions on Industrial Electronics 54 (3) (2007) 1298–1310.
[105] R.-E. Precup, S. Preitl, E.M. Petriu, J.K. Tar, M.L. Tomescu, C. Pozna, Generic two-degree-of-freedom linear and fuzzy controllers for integral processes, Journal of the Franklin Institute 346 (10) (2009) 980–1003.
[106] A. Rodriguez-Martinez, R. Garduno-Ramirez, 2 DOF fuzzy gain-scheduling PI for combustion turbogenerator speed control, in: Proceedings of IFAC Conference on Advances in PID Control, Brescia, Italy, (2012), pp. 276–281.
[107] Modular Servo System, User's Manual, Inteco Ltd., Krakow, Poland, 2007.
[108] S. Preitl, R.-E. Precup, An extension of tuning relations after symmetrical optimum method for PI and PID controllers, Automatica 35 (10) (1999) 1731–1736.
[109] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Radac, Charged system search algorithms for optimal tuning of PI controllers, in: Proceedings of 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control, Würzburg, Germany, (2012), pp. 115–120.
[110] N. Khaehintung, A. Kunakorn, P. Sirisuk, A novel fuzzy logic control technique tuned by particle swarm optimization for maximum power point tracking for a photovoltaic system using a current-mode boost converter with bifurcation control, International Journal of Control, Automation and Systems 8 (2) (2010) 289–300.
[111] P. Siano, C. Citro, Designing fuzzy logic controllers for DC–DC converters using multi-objective particle swarm optimization, Electric Power Systems Research 112 (2014) 74–83.
[112] C.-F. Juang, Y.-C. Chang, Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments, IEEE Transactions on Fuzzy Systems 19 (2) (2011) 379–392.
[113] S. Bouallègue, J. Haggège, M. Ayadi, M. Benrejeb, PID-type fuzzy logic controller tuning based on particle swarm optimization, Engineering Applications of Artificial Intelligence 25 (3) (2012) 484–493.
[114] R.-E. Precup, R.-C. David, E.M. Petriu, M.-B. Radac, S. Preitl, J. Fodor, Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems, Knowledge-Based Systems 38 (2013) 74–84.
[115] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, A.S. Paul, Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity, in: A. Gaspar-Cunha, R. Takahashi, G. Schaefer, L. Costa (Eds.), Soft Computing in Industrial Applications, Advances in Intelligent and Soft Computing, vol. 96, Springer-Verlag, Berlin/Heidelberg, 2011, pp. 141–150.
[116] R.-C. David, R.-E. Precup, E.M. Petriu, M.-B. Radac, S. Preitl, Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity, Information Sciences 247 (2013) 154–173.
[117] R.E. Haber, R. Haber-Haber, A. Jiménez, R. Galán, An optimal fuzzy control system in a network environment based on simulated annealing: an application to a drilling process, Applied Soft Computing 9 (3) (2009) 889–895.
[118] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Radac, Fuzzy control systems with reduced parametric sensitivity based on simulated annealing, IEEE Transactions on Industrial Electronics 59 (8) (2012) 3049–3061.
[119] R.E. Haber, R.M. del Toro, A. Gajate, Optimal fuzzy control system using the cross-entropy method: a case study of a drilling process, Information Sciences 180 (14) (2010) 2777–2792.
[120] M.A. Olivares-Mendez, L. Mejias, P. Campoy, I. Mellado-Bataller, Cross-entropy optimization for scaling factors of a fuzzy controller: a see-and-avoid approach for unmanned aerial systems, Journal of Intelligent & Robotic Systems 69 (1–4) (2013) 189–205.
[121] R. Haber, R.M. del Toro, J. Godoy, A. Gajate, Intelligent tuning of fuzzy controllers by learning and optimization, in: F. Matía, G.N. Marichal, E. Jiménez (Eds.), Fuzzy Modeling and Control: Theory and Applications, Atlantis Computational Intelligence Systems, vol. 9, Atlantis Press/Springer-Verlag, Paris, 2013, pp. 135–158.
[122] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Radac, Novel adaptive charged system search algorithm for optimal tuning of fuzzy controllers, Expert Systems with Applications 41 (4) (2014) 1168–1175.
[123] A. Chakrabarty, S. Banerjee, S. Maity, A. Chatterjee, Fuzzy model predictive control of non-linear processes using convolution models and foraging algorithms, Measurement 46 (4) (2013) 1616–1629.
[124] O. Castillo, P. Melin, A. Alanis, O. Montiel, R. Sepulveda, Optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms, Soft Computing 15 (6) (2011) 1145–1160.
[125] O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Applied Soft Computing 12 (4) (2012) 1267–1278.
[126] O. Castillo, P. Melin, A review on interval type-2 fuzzy logic applications in intelligent control, Information Sciences 279 (2014) 615–631.
[127] Y. Maldonado, O. Castillo, P. Melin, Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications, Applied Soft Computing 13 (1) (2013) 496–508.
[128] O. Linda, M. Manic, Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot, IEEE Transactions on Industrial Informatics 7 (4) (2011) 661–670.
[129] O. Linda, M. Manic, Interval type-2 fuzzy voter design for fault tolerant systems, Information Sciences 181 (14) (2011) 2933–2950.
[130] H. Baseri, Simulated annealing based optimization of dressing conditions for increasing the grinding performance, The International Journal of Advanced Manufacturing Technology 59 (5–8) (2012) 531–538.

[131] G. Ćirović, D. Pamučar, D. Božanić, Green logistic vehicle routing problem: routing light delivery vehicles in urban areas using a neuro-fuzzy model, Expert Systems with Applications 41 (9) (2014) 4245–4258.
[132] V.M. Panchade, R.H. Chile, B.M. Patre, A survey on sliding mode control strategies for induction motors, Annual Reviews in Control 37 (2) (2013) 289–307.
[133] R.-E. Precup, A.-L. Borza, M.-B. Radac, E.M. Petriu, Bacterial foraging optimization approach to the controller tuning for automotive torque motors, in: Proceedings of IEEE 23rd International Symposium on Industrial Electronics (ISIE 2014), Istanbul, Turkey, (2014), pp. 972–977.
[134] H. Hjalmarsson, Iterative feedback tuning: an overview, International Journal of Adaptive Control and Signal Processing 16 (5) (2002) 373–395.
[135] O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, L. Triest, Iterative feedback tuning of PID parameters: comparison with classical tuning rules, Control Engineering Practice 11 (9) (2003) 1023–1033.
[136] A. Karimi, M. Butcher, R. Longchamp, Model-free precompensator tuning based on the correlation approach, IEEE Transactions on Control Systems Technology 16 (5) (2008) 1013–1020.
[137] S. Formentin, A. Karimi, S.M. Savaresi, Optimal input design for direct data-driven tuning of model-reference controllers, Automatica 49 (6) (2013) 1874–1882.
[138] J.C. Spall, J.A. Cristion, Model-free control of nonlinear stochastic systems with discrete-time measurements, IEEE Transactions on Automatic Control 43 (9) (1998) 1198–1210.
[139] M.-B. Radac, R.-E. Precup, E.M. Petriu, S. Preitl, Application of IFT and SPSA to servo system control, IEEE Transactions on Neural Networks 22 (12) (2011) 2363–2375.
[140] R. Kadali, B. Huang, A. Rossiter, A data driven subspace approach to predictive controller design, Control Engineering Practice 11 (3) (2003) 261–278.
[141] S.S. Ge, Z. Li, H. Yang, Data driven adaptive predictive control for holonomic constrained under-actuated biped robots, IEEE Transactions on Control Systems Technology 20 (3) (2012) 787–795.
[142] M. Fliess, C. Join, S. Riachy, Revisiting some practical issues in the implementation of model-free control, in: Proceedings of 18th IFAC World Congress, Milano, Italy, (2011), pp. 8589–8594.
[143] M. Fliess, C. Join, Model-free control, International Journal of Control 86 (12) (2013) 2228–2252.
[144] Z.-S. Hou, S. Jin, A novel data-driven control approach for a class of discrete-time nonlinear systems, IEEE Transactions on Control Systems Technology 19 (6) (2011) 1549–1558.
[145] Z.-S. Hou, S. Jin, Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems, IEEE Transactions on Neural Networks 22 (12) (2011) 2173–2188.
[146] M.G. Safonov, T.-C. Tsao, The unfalsified control concept and learning, IEEE Transactions on Automatic Control 42 (6) (1997) 843–847.
[147] T. Wonghong, S. Engell, Automatic controller tuning via unfalsified control, Journal of Process Control 22 (10) (2012) 2008–2025.
[148] A.J. McDaid, K.C. Aw, E. Haemmerle, S.Q. Xie, Control of IPMC actuators for microfluidics with adaptive "online" iterative feedback tuning, IEEE/ASME Transactions on Mechatronics 17 (4) (2012) 789–797.
[149] R. Caponetto, L. Fortuna, S. Fazzino, M.G. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 289–304.
[150] L. dos Santos Coelho, V.C. Mariani, A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch, Chaos, Solitons and Fractals 39 (2) (2009) 510–518.
[151] M. Pluhacek, R. Senkerik, D. Davendra, Z. Kominkova Oplatkova, I. Zelinka, On the behavior and performance of chaos driven PSO algorithm with inertia weight, Computers and Mathematics with Applications 66 (2) (2013) 122–134.
[152] R.-E. Precup, M.L. Tomescu, S. Preitl, Fuzzy logic control system stability analysis based on Lyapunov's direct method, International Journal of Computers, Communications & Control 4 (4) (2009) 415–426.
[153] I.M. Ginarsa, A. Soeprijanto, M.H. Purnomo, Controlling chaos and voltage collapse using an ANFIS-based composite controller-static var compensator in power systems, International Journal of Electrical Power & Energy Systems 46 (2013) 79–88.
[154] V. Vembarasan, P. Balasubramaniam, Chaotic synchronization of Rikitake system based on T-S fuzzy control techniques, Nonlinear Dynamics 74 (1–2) (2013) 31–44.
[155] I. Škrjanc, S. Blažič, D. Matko, Direct fuzzy model-reference adaptive control, International Journal of Intelligent Systems 17 (10) (2002) 943–963.
[156] I. Škrjanc, S. Blažič, S. Oblak, J. Richalet, An approach to predictive control of multivariable time-delayed plant: stability and design issues, ISA Transactions 43 (4) (2004) 585–595.
[157] R.-E. Precup, S. Preitl, Stability and sensitivity analysis of fuzzy control systems. Mechatronics applications, Acta Polytechnica Hungarica 3 (1) (2006) 61–76.
[158] S. Preitl, R.-E. Precup, J. Fodor, B. Bede, Iterative feedback tuning in fuzzy control systems. Theory and applications, Acta Polytechnica Hungarica 3 (3) (2006) 81–96.
[159] F.-G. Filip, K. Leiviskä, Large-scale complex systems, in: Nof S.Y (Ed.), Springer Handbook of Automation, Springer-Verlag, Berlin/Heidelberg, 2009, pp. 619–638.
[160] D. Tikk, Z.C. Johanyák, S. Kovács, K.W. Wong, Fuzzy rule interpolation and extrapolation techniques: criteria and evaluation guidelines, Journal of Advanced Computational Intelligence and Intelligent Informatics 15 (3) (2011) 254–263.
[161] R.-E. Precup, S. Preitl, M.-B. Radac, E.M. Petriu, C.-A. Dragos, J.K. Tar, Experiment-based teaching in advanced control engineering, IEEE Transactions on Education 54 (3) (2011) 345–355.
[162] J. Vaščák, Adaptation of fuzzy cognitive maps by migration algorithms, Kybernetes 41 (3–4) (2012) 429–443.
[163] L. Tang, Y. Zhao, J. Liu, An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production, IEEE Transactions on Evolutionary Computation 18 (2) (2014) 209–225.
[164] I.H. Khan, A comparative study of evolutionary algorithms, International Journal of Artificial Intelligence 12 (1) (2014) 1–17.
[165] C. Lagos, B. Crawford, R. Soto, J.-M. Rugio, E. Cabrera, F. Parades, Combining tabu search and genetic algorithms to solve the capacitated multicommodity network flow problem, Studies in Informatics and Control 23 (3) (2014) 265–277.
[166] R.-E. Precup, S. Preitl, PI and PID controllers tuning for integral-type servo systems to ensure robust stability and controller robustness, Electrical Engineering 88 (2) (2006) 149–156.
[167] J. Vaščák, K. Hirota, Integrated decision-making system for robot soccer, Journal of Advanced Computational Intelligence and Intelligent Informatics 15 (2) (2011) 156–163.
[168] L. Tang, Y. Yang, J. Liu, Modeling and solution for the coil sequencing problem in steel color-coating production, IEEE Transactions on Control Systems Technology 20 (6) (2012) 1409–1420.
[169] Z.C. Johanyák, Fuzzy modeling of thermoplastic composites' melt volume rate, Computing and Informatics 32 (4) (2013) 845–857.
[170] R. Zamoum Boushaki, B. Chetate, Y. Zamoum, Artificial neural network control of the recycle compression system, Studies in Informatics and Control 23 (1) (2014) 65–76.
[171] H.-N. Teodorescu, E.-F. Iftene, Efficiency of a combined protection method against correlation, International Journal of Computers, Communication & Control 9 (1) (2014) 79–84.
[172] J.L. Guerrero, A. Berlanga, J.M. Molina, Multiobjective local search as an initialization procedure for evolutionary approaches to polygonal approximation, International Journal of Artificial Intelligence 12 (1) (2014) 150–165.
[173] E. Osaba, F. Diaz, E. Onieva, R. Carballedo, A. Perallos, AMCPA: a population metaheuristic with adaptive crossover probability and multi-crossover mechanism for solving combinatorial optimization problems, International Journal of Artificial Intelligence 12 (2) (2014) 1–23.
[174] P. Baranyi, D. Tikk, Y. Yam, R.J. Patton, From differential equations to PDC controller design via numerical transformation, Computers in Industry 51 (3) (2003) 281–297.
[175] P. Baranyi, TP model transformation as a way to LMI based controller design, IEEE Transactions on Industrial Electronics 51 (2) (2004) 387–400.
[176] P. Baranyi, Tensor-product model-based control of two-dimensional aeroelastic system, Journal of Guidance, Control, and Dynamics 29 (2) (2006) 391–400.
[177] F. Kolonić, A. Poljugan, I. Petrović, Tensor product model transformation-based controller design for gantry crane control system – an application approach, Acta Polytechnica Hungarica 4 (3) (2006) 95–112.
[178] C. Arino, A. Sala, Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure, Engineering Applications of Artificial Intelligence 20 (8) (2007) 1036–1046.
[179] R.-E. Precup, S. Preitl, I.-B. Ursache, P.A. Clep, P. Baranyi, J.K. Tar, On the combination of tensor product and fuzzy models, in: Proceedings of 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, vol. 2, Cluj-Napoca, Romania, (2008), pp. 48–53.
[180] R.-E. Precup, L.-T. Dioanca, E.M. Petriu, M.-B. Radac, S. Preitl, C.-A. Dragos, Tensor product-based real-time control of the liquid levels in a three tank system, in: Proceedings of 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Montreal, Canada, (2010), pp. 768–773.
[181] R.-E. Precup, C.-A. Dragos, S. Preitl, M.-B. Radac, E.M. Petriu, Novel tensor product models for automatic transmission system control, IEEE Systems Journal 6 (3) (2012) 488–498.
[182] P. Galambos, P. Baranyi, Representing the model of impedance controlled robot interaction with feedback delay in polytopic LPV form: TP model transformation based approach, Acta Polytechnica Hungarica 10 (1) (2013) 139–157.
[183] P. Baranyi, B. Takarics, Aeroelastic wing section control via relaxed tensor product model transformation framework, Journal of Guidance, Control, and Dynamics 37 (5) (2014) 1671–1678.
[184] I. Dumitrache, M. Dragoicea, Intelligent techniques for cognitive mobile robots, Control Engineering and Applied Informatics 6 (2) (2004) 3–8.
[185] R.-E. Precup, S. Preitl, M. Balas, V. Balas, Fuzzy controllers for tire slip control in anti-lock braking systems, in: Proceedings of IEEE International Conference on Fuzzy Systems, vol. 3, Budapest, Hungary, (2004), pp. 1317–1322.
[186] G. Hermann, Robust convex-hull based algorithms for straightness and flatness determination, Acta Polytechnica Hungarica 4 (4) (2007) 111–120.
[187] C. Pozna, F. Troester, R.-E. Precup, J.K. Tar, S. Preitl, On the design of an obstacle avoiding trajectory: method and simulation, Mathematics and Computers in Simulation 79 (7) (2009) 2211–2226.
[188] R.-E. Precup, M.L. Tomescu, M.-B. Radac, E.M. Petriu, S. Preitl, C.-A. Dragos, Iterative performance improvement of fuzzy control systems for three tank systems, Expert Systems with Applications 39 (9) (2012) 8288–8299.
[189] R.-E. Precup, R.-C. David, E.M. Petriu, S. Preitl, M.-B. Radac, Novel adaptive gravitational search algorithm for fuzzy controlled servo systems, IEEE Transactions on Industrial Informatics 8 (4) (2012) 791–800.

Radu-Emil Precup received the Dipl.Ing. (with honours) degree in automation and computers from the "Traian Vuia" Polytechnic Institute of Timisoara, Timisoara, Romania, the Dipl. degree in mathematics from the West University of Timisoara, Timisoara, and the Ph.D. degree in automatic systems from the Politehnica University of Timisoara (PUT), Timisoara, Romania, in 1987, 1993, and 1996, respectively. He is currently with the PUT, Timisoara, Romania, where he became a Professor with the Department of Automation and Applied Informatics in 2000, and is currently a Doctoral Supervisor of Automation and Systems Engineering. He is also an Honorary Professor and a Member of the Doctoral School of Applied Informatics, Óbuda University, Budapest, Hungary. From 1999 to 2009, he held research and teaching positions with the Université de Savoie, Chambéry and Annecy, France, Budapest Tech Polytechnical Institution, Budapest, Hungary, Vienna University of Technology, Vienna, Austria, and Budapest University of Technology and Economics, Budapest, Hungary. He has been an Editor-in-Chief of the International Journal of Artificial Intelligence since 2008, and he is also on the editorial board of several other prestigious journals. He has published more than 300 papers in refereed journals, refereed conference proceedings, and contributions to books. His current research interests cover intelligent control systems, data-driven control, and nature-inspired algorithms for optimization. He is a senior member of IEEE, the vice-chair of the Virtual Reality Task Force of the Intelligent Systems Applications Technical Committee (TC) of the IEEE Computational Intelligence Society, a member of the Task Force on Autonomous Learning Systems within the Neural Networks TC of the IEEE Computational Intelligence Society, the Subcommittee on Computational Intelligence as part of the TC Committee on Control, Robotics and Mechatronics in the IEEE Industrial Electronics Society, the Task Force on Educational Aspects of Standards of Computational Intelligence as part of the TC on Standards in the IEEE Computational Intelligence Society, the International Federation of Automatic Control (IFAC) Technical Committee on Computational Intelligence in Control (previously named Cognition and Control), the Working Group WG 12.9 on Computational Intelligence of the Technical Committee TC12 on Artificial Intelligence of the International Federation for Information Processing (IFIP), the European Society for Fuzzy Logic and Technology (EUSFLAT), the Hungarian Fuzzy Association, and the Romanian Society of Control Engineering and Technical Informatics. Prof. Precup received the "Grigore Moisil" Prize from the Romanian Academy in 2005 for his contribution on fuzzy control, the Excellency Diploma of the International Conference on Automation, Quality & Testing, Robotics AQTR 2004 (THETA 14, Cluj-Napoca, Romania), Two Best Paper Awards in the Intelligent Control Area of the 2008 Conference on Human System Interaction HSI 2008, Krakow (Poland), Best Paper Award of 16th Online World Conference on Soft Computing in Industrial Applications WSC16 (Loughborough University, UK) in 2011, and the Certificate of Appreciation for the Best Paper in the Session TT07 1 Control Theory of 39th Annual Conference of the IEEE Industrial Electronics Society IECON 2013 (Vienna, Austria).

Plamen Angelov (MEng'98 Sofia Technical University; PhD'93 Bulgarian Academy of Sciences) holds a Chair of Intelligent Systems and leads the Data Science Group at Lancaster University, UK. He is a Senior Member of IEEE and INNS of which he is a member of the Board of Governors. He authored or co-authored over 200 peer reviewed publications including two research monographs, a dozen of edited books and five patents. His research interests include autonomous machine learning, knowledge extraction from data streams, evolving systems. Prof. Angelov chairs a Technical Committee and couple of Task Forces within IEEE and a number of high profile conferences. He was awarded "The Engineer Innovation and Technology 2008" Award in two categories: (i) Aerospace and Defence and (ii) The Special Award. He was also awarded Chair of Excellence Award (2015) to be hold in Carlos III University, Madrid, Spain. He is founding Editor-in-Chief of the Springer's journal on Evolving Systems and Associate Editor of several other international journals, including two IEEE Transactions.

Bruno Sielly Jales Costa received the B. Eng., M. Eng. and D. Eng. degrees in Electrical and Computer Engineering from The Federal University of Rio Grande do Norte (UFRN), Brazil, in 2008, 2009 and 2014, respectively. He is currently a Professor with The Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), Brazil. Prof. Costa's latest work is based on computational intelligence techniques applied to control and fault detection and diagnosis problems.

Moamar Sayed-Mouchaweh received his Master degree from the University of Technology of Compiegne-France in 1999. Then, he received his PhD degree from the University of Reims-France in December 2002. He was nominated as Associated Professor in Computer Science, Control and Signal processing at the University of Reims-France in the Research Centre in Sciences and Technology of the Information and the Communication (CReSTIC). In December 2008, he obtained the Habilitation to Direct Researches (HDR) in Computer science, Control and Signal processing. Since September 2011, he is working as a Full Professor in the High National Engineering School of Mines "Ecole Nationale Supérieure des Mines de Douai" at the Department of Computer Science and Automatic Control (Informatique & Automatique IA). He co-organized and chaired several special sessions, workshops and tutorials in the domain of machine learning and its applications within several well-known international conferences. He was involved in the programme committee of several international conferences and edited or co-edited several books in the domain of machine learning in non-stationary environments.