Applied Soft Computing 27 (2015) 575–589

Contents lists available at ScienceDirect

Applied Soft Computing

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

Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for Anti-lock Braking Systems

Radu-Emil Precup[a,*], Marius-Csaba Sabau[a], Emil M. Petriu[b]

[a] Politehnica University of Timisoara, Department of Automation and Applied Informatics, Bd. V. Parvan 2, RO-300223 Timisoara, Romania
[b] University of Ottawa, School of Electrical Engineering and Computer Science, 800 King Edward, Ottawa, ON K1N 6N5, Canada

ARTICLE INFO

Article history:
Received 13 December 2013
Received in revised form 27 June 2014
Accepted 1 July 2014
Available online 14 July 2014

Keywords:
Anti-lock Braking Systems
Operating point selection algorithm
Particle Swarm Optimization
Simulated Annealing
Takagi-Sugeno-Kang fuzzy models
Real-time experimental results

ABSTRACT

This paper suggests a synergy of fuzzy logic and nature-inspired optimization in terms of the nature-inspired optimal tuning of the input membership functions of a class of Takagi-Sugeno-Kang (TSK) fuzzy models dedicated to Anti-lock Braking Systems (ABSs). A set of TSK fuzzy models is proposed by a novel fuzzy modeling approach for ABSs. The fuzzy modeling approach starts with the derivation of a set of local state-space models of the nonlinear ABS process by the linearization of the first-principle process model at ten operating points. The TSK fuzzy model structure and the initial TSK fuzzy models are obtained by the modal equivalence principle in terms of placing the local state-space models in the rule consequents of the TSK fuzzy models. An operating point selection algorithm to guide modeling is proposed, formulated on the basis of ranking the operating points according to their importance factors, and inserted in the third step of the fuzzy modeling approach. The optimization problems are defined such that to minimize the objective functions expressed as the average of squared modeling errors over the time horizon, and the variables of these functions are a part of the parameters of the input membership functions. Two representative nature-inspired algorithms, namely a Simulated Annealing (SA) algorithm and a Particle Swarm Optimization (PSO) algorithm, are implemented to solve the optimization problems and to obtain optimal TSK fuzzy models. The validation and the comparison of SA and PSO and of the new TSK fuzzy models are carried out for an ABS laboratory equipment. The real-time experimental results highlight that the optimized TSK fuzzy models are simple and consistent with both training data and validation data and that these models outperform the initial TSK fuzzy models.

Introduction

The development of accurate models for nonlinear processes is challenging as they contribute to high control system performance in many applications. The nonlinearity of the processes specific to Anti-lock Braking Systems (ABSs) and the importance of the ABS control systems in the safety subsystems of modern cars justify the development of fuzzy models in the framework of the model-based design of fuzzy control systems.

A systematic way to obtain accurate fuzzy models is the optimization in terms of optimization problems with variables represented by the parameters of fuzzy models. Nature-inspired algorithms are involved in solving the optimization problems, and they ensure the optimal tuning of fuzzy models.

Building upon our previous fuzzy modeling results given in Refs. [1–5], the main contribution of this paper with respect to the state-of-the-art is an approach to fuzzy modeling of ABSs using nature-inspired optimization algorithms. Our approach starts with the derivation of an initial discrete-time TSK fuzzy model of the process specific to ABS by the modal equivalence principle. This fuzzy model is characterized by a set of local linearized state-space models of the nonlinear ABS process obtained by the linearization of the first-principle process model at a set of operating points, and these state-space models are placed next in the rule consequents. Optimization problems targeting the minimization of objective functions expressed as the average of squared modeling errors over the time horizon are defined, and the vector variables in these problems are a part of the parameters of the input m.f.s. The optimization problems are solved by the implementation of two representative nature-inspired algorithms, namely an SA algorithm and a PSO algorithm that finally give novel optimal TSK fuzzy models for ABSs. Concluding, this paper proposes a nature-inspired-based approach dedicated to the computation of optimal TSK fuzzy models.

* Corresponding author. Tel.: +40 256 40 32 29/30/26; fax: +40 256 40 3214.
E-mail addresses: radu.precup@upt.ro, radu.precup@aut.upt.ro (R.-E. Precup), maryusz_10_wky@yahoo.com (M.-C. Sabau), petriu@uottawa.ca (E.M. Petriu).

http://dx.doi.org/10.1016/j.asoc.2014.07.004
1568-4946/

This paper addresses fuzzy modeling. The fuzzy models will next be used in the model-based design of stable and optimal fuzzy control systems. Attractive nonlinear process models including fuzzy ones that can be treated by our approach are presented in Refs. [6–11]. The Tensor Product (TP) model transformation [12–14] can also be considered in combination with TSK fuzzy models because, as mentioned in Ref. [15], the product decision operator-based TSK fuzzy models and the function of the TP model are generally the same. The main difference is that the TSK fuzzy model originally can be represented as a combination of locally linearized Linear Time-Invariant (LTI) models, where the locality is expressed by the shapes of the input m.f.s. However in case of TP model the weighting functions (which correspond to the m.f.s in the fuzzy models) may not have locality, they spread in the whole interval of interest, so the LTI components of the model cannot readily be assigned to a definite operation point.

The paper is organized as follows. The next section gives the background. The section following the Background section presents the fuzzy modeling approach with focus on ABS process models. The optimization problem is defined and the new operating point selection algorithm is formulated. Nature-inspired optimization algorithms section gives details on the two nature-inspired algorithms, namely PSO and SA, involved in the computation of the optimal TSK fuzzy models produced by the fuzzy modeling approach. Experimental results section offers the experimental validation of the proposed TSK fuzzy models for a laboratory ABS systems and offers a discussion of the results. Although the laboratory process is nonlinear and time-invariant, it models real-world automotive applications which are characterized by parametric disturbances that justify the computation of TSK fuzzy models. Conclusion section points out the concluding remarks.

Background

Some recent approaches to the fuzzy modeling of ABSs are pointed out as follows. A Takagi-Sugeno-Kang (TSK) fuzzy model of deceleration based on analyzing the braking process and dynamic model of vehicle and wheel is suggested in Ref. [16]. A quarter vehicle braking model for ABS control with four-degrees of freedom subjected to irregular excitation from a road surface is proposed in Ref. [17]. Two discrete-time fuzzy models of ABS processes based on the modal equivalence principle are offered in Ref. [18]. Several fuzzy models are integrated in intelligent ABS controllers based on fuzzy control [19], sliding mode control [19,20], neural networks [21], neuro-fuzzy [20,22] and sliding mode fuzzy control [18,19]. The importance of fuzzy models for ABS control in automotive systems is discussed in Refs. [15,23].

The combination of fuzzy models and nature-inspired optimization algorithms belongs to the actual trends in soft computing dealing with the synergy where all individual contributing technologies are seamlessly structured together, the designers build on their unique strengths and compensate for existing limitations. Attractive points of view on the synergy delivered by the key technologies of soft computing are treated in the literature. Such approaches are exemplified as follows. The synergy between classical and soft computing techniques is discussed in Ref. [24] with focus on time series prediction. A pseudo-Gaussian basis function neural network with regression weights is proposed in Ref. [24] and produced by a method dedicated to the extraction of valuable process information from input–output data. The synergistic use of soft computing technologies for fault detection in gas turbine engines is treated in Ref. [25]. The fault detection method proposed in Ref. [25] employs statistics, signal processing, and soft computing. An approach that merges the set calculus and the fuzzy sets to deal with a description of geometry of patterns belonging to a certain category is given in Ref. [26] and expressed as a two-stage description of granular classifiers. The global optimization is used in terms of Particle Swarm Optimization (PSO) to carry out the calibration of the connections of the interrelationships between the nodes of cognitive maps in Ref. [27] in the framework of granular computing. A variable structure systems approach is combined in Ref. [28] with the Levenberg–Marquardt algorithm to produce an optimization approach applied to the training of fuzzy inference systems and tested on a two degrees of freedom direct drive SCARA robotic manipulator. A cooperative coevolutionary algorithm is combined with a basic mixture of experts model in Ref. [29] and applied to classification problems showing a better exploration of the weight space. A framework for granular computing neural-fuzzy modeling structures based on neutrosophic logic is suggested in Ref. [30] and applied on real-world industrial data. A hybrid optimization approach that combines PSO and genetic algorithms is proposed in Ref. [31] aiming the use of fuzzy logic to integrate the advantages of both optimization algorithms and to carry out the parameter tuning. Three hybrid metaheuristic algorithms that improve other two hybrid differential evolution metaheuristic algorithms are analyzed in Ref. [32]. The Lamarckian learning and the Baldwinian learning are merged in Ref. [33] by analyzing their characteristics in the process of migration among subpopulations resulting in the hybridization of differential evolution and Hooke–Jeeves local search. The cultural algorithms framework is extended in Ref. [34] by enhancing it with social networks that use seeds of knowledge at the belief space and applied to mechanical design optimization problems. The parameters of fuzzy systems are tuned by neural networks in Ref. [35] to provide an industrial quality control system for sound speakers and combined with the fractal dimension used as using, as a measure of the complexity of the sound signal. Several soft computing techniques are combined with optimization approaches in Refs. [36,37] and applied to supply chain management problems.

A representative part of the state-of-the-art on the nature-inspired optimal tuning of fuzzy models is discussed as follows. The optimization of the parameters in the rule antecedents and consequents of TSK fuzzy models by Simulated Annealing (SA) algorithms is proposed in Ref. [38] and applied to time series prediction. The design of type-2 fuzzy logic systems by means of interval type-2 fuzzy logic systems is discussed in Ref. [39] targeting the reduction of the computations needed to get the best footprint of uncertainty. The SA-based tuning of the parameters of input membership functions (m.f.s) in fuzzy control systems is applied to TSK fuzzy models of several processes such as general nonlinear systems [40], coupled tanks [41], chaotic time series [42], ABS [1], servo systems [2], magnetic levitation systems [3,4], and to fuzzy controllers as well [43,44]. PSO, Gravitational Search and Charged System Search algorithms suggest in Refs. [44–46] the optimal tuning of TSK fuzzy controllers with a reduced parametric sensitivity. PSO algorithms are involved in Ref. [5,47] in TSK fuzzy model identification. The nature-inspired optimal tuning of fuzzy controllers for networked control systems is discussed in Ref. [48]. The input m.f.s of TSK fuzzy models are tuned in Ref. [49] by a combination of Island Model Parallel Genetic Algorithm and space search memetic algorithm. The cross-entropy method [50], the genetic algorithms [51], the chaotic ant swarm optimizer [52], the Bacterial Foraging Optimization algorithms [53], the migration algorithms [54,55] and the PSO-based optimization of linguistic fuzzy models [56] prove good performance in fuzzy model optimization. Illustrative extensions to type-2 fuzzy systems are applied in Refs. [57–59] to delta parallel robots, autonomous wheeled mobile robots and machine learning data sets.

The evolving TSK fuzzy systems can also be considered to belong to the class of nature-inspired optimized fuzzy models because they are characterized by the continuous online rule base learning

[60]. Compared to the above analyzed TSK fuzzy models with fixed structures, the evolving TSK fuzzy models are different as they use several algorithms for recursive parameter estimation, eventually associated with online clustering [60–64].

The new functionalities of the nature-inspired-based approach proposed in this paper and dedicated to the computation of optimal TSK fuzzy models are:

- Two nature-inspired optimization algorithms that use SA and PSO are offered to ensure the optimal tuning of TSK fuzzy models for ABSs. 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.
- A new generation of optimal TSK fuzzy models is given. The TSK fuzzy model structure consists of three input variables, five triangular input m.f.s of the linguistic terms (LTs) of each input variable, different numbers of rules with third-order discrete-time state-space models in the rule consequents, the SUM and PROD operators in the inference engine, and the weighted average method in the defuzzification module.
- An operating point selection algorithm is proposed. This algorithm affects the structure of TSK fuzzy models because it gives the operating points and, therefore, it sets the number of rules as each rule is attached to a certain operating point in terms of the modal equivalence principle.
- An analysis of the comparison between the two nature-inspired optimization algorithms is conducted in a case study related to the fuzzy modeling of an ABS laboratory equipment. A set of experimental results is included to support this analysis. The comparison shows that the optimized TSK fuzzy models ensure the performance enhancement versus the initial TSK fuzzy models on both training and validation (testing) data in a computationally undemanding manner.

These new contributions and functionalities are important and advantageous with respect to the state-of-the-art on other optimization approaches and algorithms for TSK fuzzy model tuning because:

- Our approach ensures a relatively simple and transparent way to ensure the additional fine tuning of the initial TSK fuzzy models obtained by the modal equivalence principle. A cost-effective optimization approach is thus offered.
- The knowledge on the process models is incorporated by the local crisp models in the rule consequents.
- The algorithms lead to TSK fuzzy models with improved performance for an illustrative class of nonlinear systems represented by ABSs. The performance of the initial and optimized models is tested by real-time experiments on the laboratory equipment.
- Our approach can be easily extended such that to be combined with other nature-inspired optimization algorithms to offer high performance data-based models in terms of our point of view on the synergy of fuzzy logic and nature-inspired optimization.
- The operating point selection is an important problem for the fuzzy modeling approach suggested in this paper. A straightforward approach to select the most important operating points is to compare all possible combinations by using a predefined evaluation criterion proposed in Ref. [65] to select the most relevant inputs for ANFIS learning. Using the importance factor of a certain input defined in Ref. [66] in the framework of input selection based on neural networks, this paper proposes an operating point selection algorithm which ranks the operating points according to their importance factors. This algorithm is useful to guide modeling, and it is integrated in the third step of our fuzzy modeling approach.

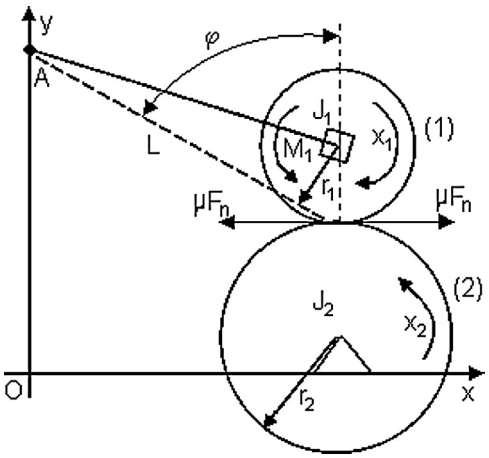

Fig. 1. Block diagram of process specific to ABS laboratory equipment.

Fuzzy modeling approach

As shown in Refs. [67,68], the ABS laboratory equipment consists of two wheels schematically illustrated in Fig. 1. The drive wheel (2) drives the upper wheel (1) by friction contact. Pulse-Width Modulation (PWM) controlled DC motors are employed to power the drive wheel (2) and to provide the resistant/break torque for the driven wheel (1).

When the driven wheel (1) reaches the predefined speed threshold, the braking procedure is initiated using a classical disk break system. The lever mechanism that transforms the rotation of the breaking motor into a displacement of the pressure pads on the brake disk introduces a variable dead time for each different reference input, i.e., for each desired wheel slip.

The continuous-time nonlinear state-space model of the ABS laboratory equipment [68] is derived on the basis of the first-principle process model [1,68]

$$\begin{aligned} J_1\dot{x}_1 &= F_n r_1 \mu(\lambda) - d_1 x_1 - M_{10} - M_1, \\ J_2\dot{x}_2 &= -F_n r_2 \mu(\lambda) - d_2 x_2 - M_{20}, \end{aligned} \tag{1}$$

where λ is the longitudinal slip (the wheel slip), J_1 and J_2 are the inertia moments of wheels, x_1 and x_2 are the angular velocities, d_1 and d_2 are the friction coefficients in wheels' axes, M_{10} and M_{20} are the static friction torques that oppose the normal rotation, M_1 is the brake torque, r_1 and r_2 are the radii of wheels, F_n is the normal force that the upper wheel pushes upon the lower wheel, $\mu(\lambda)$ is the friction coefficient, and $\dot{x}_1$ and $\dot{x}_2$ are the wheels' angular accelerations.

The identification by experiment-based measurements leads to the following parameter values given in Ref. [1]:

$$\begin{aligned} &r_1 = r_2 = 0.99\,\text{m},\ F_n = 58.214\,\text{N},\ J_1 = 7.53 \cdot 10^{-3}\,\text{kg m}^2, \\ &\quad J_2 = 25.6 \cdot 10^{-3}\,\text{kg m}^2, \\ &d_1 = 1.1874 \cdot 10^{-4}\,\text{kg m}^2/\text{s},\ d_2 = 2.1468 \cdot 10^{-4}\,\text{kg m}^2/\text{s}, \\ &\quad M_{10} = 0.0032\,\text{N m},\ M_{20} = 0.0925\,\text{N m}. \end{aligned} \tag{2}$$

The wheel slip and the nonlinear term $S(\lambda)$ are expressed as

$$\begin{aligned} \lambda &= \frac{r_2 x_2 - r_1 x_1}{r_2 x_2}, \quad x_2 \neq 0, \\ S(\lambda) &= \frac{\mu(\lambda)}{L[\sin\varphi - \mu(\lambda)\cos\varphi]}, \quad \tan\varphi \neq \mu(\lambda), \end{aligned} \tag{3}$$

where $L = 0.37$ m is the arm's length which fixes the upper wheel, and $\varphi = 65.61^\circ$ is the angle between the normal direction in wheels' contact point and the direction of L. The typical slip-friction characteristics $\mu(\lambda)$ show that the friction coefficient μ depends nonlinearly on the longitudinal slip λ, and these characteristics depend on the road surface in real-world ABSs.

The nonlinear state-space equations of the ABS process are [1,68]

$$\begin{aligned}\dot{x}_1 &= S(\lambda)(c_{11}x_1 + c_{12}) + c_{13}x_1 + c_{14} + (c_{15}S(\lambda) + c_{16})s_1M_1,\\ \dot{x}_2 &= S(\lambda)(c_{21}x_1 + c_{22}) + c_{23}x_2 + c_{24} + c_{25}S(\lambda)s_1M_1,\\ \dot{M}_1 &= c_{31}(b(u) - M_1),\end{aligned} \tag{4}$$

where u is the control signal applied to the actuator, i.e., the DC motor which drives the upper wheel, and the actuator's nonlinear model is highlighted in the third equation. The expressions of the parameters in Eq. (4) are [1,68]

$$\begin{aligned}&c_{11} = \frac{r_1d_1}{J_1}, c_{12} = \frac{(M_{10} + M_g)r_1}{J_1}, c_{13} = \frac{-d_1}{J_1}/, c_{14} = \frac{-M_{10}}{J_1},\\ &c_{15} = \frac{r_1}{J_1}, c_{16} = \frac{-1}{J_1}, c_{21} = \frac{-r_2d_1}{J_2}, c_{22} = \frac{-(M_{10} + M_g)r_2}{J_2},\\ &c_{23} = \frac{-d_2}{J_2}, c_{24} = \frac{-M_{20}}{J_2}, c_{25} = \frac{-r_2}{J_2},\end{aligned} \tag{5}$$

and a set of recommendations to compute these parameters is given in Ref. [69].

The variable λ is considered as a controlled output of the ABS process model, and the notation $y = \lambda$ can be used with this regard. The state variable x_1 is substituted from Eq. (3) in the model (4) leading to the continuous-time state-space equations of the ABS process [1,68]

$$\dot{\mathbf{x}} = \begin{bmatrix} z_1(\lambda, x_2) & 0 & z_3(\lambda, x_2)\\ 0 & z_{40}(\lambda) & z_5(\lambda)\\ 0 & 0 & -c_{31}\end{bmatrix}\mathbf{x} + \begin{bmatrix}0\\0\\c_{31}\end{bmatrix}b(u) + \mathbf{d}(\lambda, x_2), \tag{6}$$

where the state vector $\mathbf{x}$, that is also used as the input vector to the TSK fuzzy model, referred to also as the scheduling vector, is

$$\mathbf{x} = [\ \lambda \quad x_2 \quad M_1\ ]^T, \tag{7}$$

the superscript T indicates the matrix transposition, $z_1(\lambda, x_2)$, $z_3(\lambda, x_2)$, $z_{40}(\lambda)$ and $z_5(\lambda)$ are nonlinear functions, and $\mathbf{d}(\lambda, x_2)$ is the disturbance input vector.

The TSK fuzzy models are derived on the basis of the state-space equations (6) of the process specific to ABS. Our fuzzy modeling approach consists of the following steps:

Step 1. The structure of the TSK fuzzy models is set. Setting the TSK fuzzy model structure concerns the number of operating points, which is equal to the number of rules n_R of the TSK fuzzy models, the number of input LTs of the input variables λ, x_2 and M_1, the shapes of the m.f.s of the input LTs, the operators in the inference engine, and the method for defuzzification. We recommend the SUM and PROD operators in the inference engine, and the weighted average method in the defuzzification module of the TSK fuzzy models.

Step 2. The continuous-time state-space model of the process, represented by Eq. (6) and the first equation in Eq. (3), is linearized at the n_R important operating points. This results in n_R linearized continuous-time local process models. The local process models are placed in the rule consequents of the continuous-time TSK fuzzy models, and they related to the modal values of the input m.f.s, that represent exactly the coordinates of the operating points in terms of the modal equivalence principle [70]. The complete rule base of the continuous-time TSK fuzzy model is

$$R^i: \text{IF } \lambda(t) \text{ IS } T^i_{\lambda,l} \text{ AND } x_2(t) \text{ IS } T^i_{x_2,p} \text{ AND } M_1(t) \text{ IS } T^i_{M_1,q}$$
$$\text{THEN} \begin{cases}\dot{\mathbf{x}}(t) = \mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_iu(t)\\ y_m(t) = \mathbf{C}_i\mathbf{x}(t)\end{cases}, \quad i = 1\ldots n_R, \tag{8}$$

where t is the continuous time variable, $y_m(t)$ is the continuous-time TSK fuzzy model output, the notations $T^i_{v,l}$, $T^i_{v,p}$ and $T^i_{v,q}$ are used for the LTs $T_{\lambda,l}$, $T_{x_2,p}$ and $T_{M_1,q}$ of the input variables $\lambda(t)$, $x_2(t)$ and $M_1(t)$, respectively, referred to also as scheduling variables. We recommend the number of LTs and the ranges of indices $l = 1\ldots5$, $p \in \{1, 2\}$ and $q = 1$, and $v \in \{\lambda, x_2, M_1\}$. The matrices $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{C}_i$ in the rule consequents have appropriate dimensions.

Step 3. The sampling period T_s is set. The n_R models in the consequents of the TSK fuzzy model (8) are next discretized accepting the presence of the zero-order hold. The complete rule base of the discrete-time TSK fuzzy model is

$$R^i: \text{IF } \lambda_k \text{ IS } T^i_{\lambda,l} \text{ AND } x_{2,k} \text{ IS } T^i_{x_2,p} \text{ AND } M_{1,k} \text{ IS } T^i_{M_1,q}$$
$$\text{THEN} \begin{cases}\mathbf{x}_{k+1} = \mathbf{A}_{d,i}x_k + \mathbf{B}_{d,i}u_k\\ y_{k,m} = \mathbf{C}_{d,i}\mathbf{x}_k\end{cases}, \quad i = 1\ldots n_R, \tag{9}$$

where the subscript k indicates the discrete time variable, and $y_{k,m}$ is the discrete-time TSK fuzzy model output. The matrices $\mathbf{A}_{d,i}$, $\mathbf{B}_{d,i}$ and $\mathbf{C}_{d,i}$ in the rule consequents have appropriate dimensions, and their expressions will be exemplified in the sequel.

The recommended value of the sampling period is $T_s = 0.01$ s, which accounts for the process dynamics [68]. The disturbance input vector is neglected in Eqs. (8) and (9) for the sake of simplicity and for dealing with models that are close to real-world applications.

An operating point selection algorithm is proposed as follows in order to select the important n_R operating points from the set of all possible operating points $\{A_j | j = 1\ldots n\}$, with n – the number of possible operating points. The algorithm consists of the following sub-steps that guide this third step of the modeling approach:

- *Sub-step 3.1*. The algorithm is initialized in terms of setting its parameters δ, $0 < \delta < 1$ – the importance threshold, and τ, $0 < \tau < 1$ – the significance threshold.
- *Sub-step 3.2*. The operating point A_j, $j = 1\ldots n$, is introduced in the rule consequent of the initial TSK fuzzy model, the outputs $y_{j,k}$ of the initial TSK fuzzy model at the discrete time moment k, $k = 1\ldots N$, are recorded. The change range R_{A_j} that corresponds to the operating point A_j, $j = 1\ldots n$, is calculated:

$$R_{A_j} = \max_{k=1}^{N} y_{j,k} - \min_{k=1}^{N} y_{j,k}, \tag{10}$$

and the importance factor I_{A_j} of the operating point A_j, $j = 1\ldots n$, is next calculated:

$$I_{A_j} = \frac{R_{A_j}}{\max\limits_{j=1}^{n} R_{A_j}}. \tag{11}$$

As shown in Ref. [66] but in the context of the input variables of fuzzy models, the most important operating point is characterized by $I_{A_j} = 1$. Eqs. (11) and (17) show that a large change range R_{A_j} and a large importance factor I_{A_j} indicate a big influence of the corresponding operating point A_j, $j = 1\ldots n$. Contrarily, small values of R_{A_j} and I_{A_j} point out a relatively unimportant operating point A_j, $j = 1\ldots n$.

- *Sub-step 3.3*. The importance of all operating points is ranked according to the values of the importance factors I_{A_j}, $j = 1...n$.
- *Sub-step 3.4*. All operating points that fulfill the condition

$$I_{A_j} < \delta \tag{12}$$

are removed.

The condition (12) indicates that the corresponding operating point is unimportant; therefore, this operating point can be removed. The sub-step 3.4 produces the set of remaining n_r operating points that are selected out of the initial n operating points, $n_r < n$.

- *Sub-step 3.5*. The closely related operating points are recognized in order to conduct the independent operating point testing by the calculation of the correlation functions, $Corr(A_i, A_j)$, between the selected operating points A_i and A_j, $i, j = 1...n_r$:

$$Corr(A_i, A_j) = \frac{1/N\sum_{k=1}^{N}[(y_{i,k} - \bar{y}_i)(y_{j,k} - \bar{y}_j)]}{\sqrt{\phi_{y_i}\phi_{y_j}}}, \quad 0 \le Corr(A_i, A_j) \le 1, \tag{13}$$

where $\bar{y}_i$ and $\bar{y}_j$ are the means of the TSK fuzzy model output vectors $\mathbf{y}_i = [\, y_{i,1} \quad y_{i,2} \quad \dots \quad y_{i,N} \,]^T$ and $\mathbf{y}_j = [\, y_{j,1} \quad y_{j,2} \quad \dots \quad y_{j,N} \,]^T$, $i, j = 1 \dots n_r$, respectively, and $\phi_{\mathbf{y}_i}$ and $\phi_{\mathbf{y}_j}$ are the variances of the vectors $\mathbf{y}_i$ and $\mathbf{y}_j$, $i, j = 1 \dots n_r$, respectively. If the following condition is fulfilled:

$$Corr(A_i, A_j) > \tau, \tag{14}$$

then the operating point A_i is closely related with the operating point A_j.

The condition (14) is involved in keeping the independent operating points among the n_r selected operating points. The condition (14) also allows to remove one of the two operating points A_i or A_j. Therefore, the sub-step 3.5 results in the set of remaining n_R independent operating points out of the n_r selected operating points, $n_R < n_r$.

Step 4. The optimization problem that leads to optimal TSK fuzzy models is defined as

$$\begin{aligned} &\boldsymbol{\rho}^* = arg\min_{\boldsymbol{\rho}\in D} J(\boldsymbol{\rho}), \\ &J(\boldsymbol{\rho}) = \frac{1}{N}\sum_{k=1}^{N}(y_k(\boldsymbol{\rho}) - y_{k,m}(\boldsymbol{\rho}))^2 = \frac{1}{N}\sum_{k=1}^{N}(e_{k,m}(\boldsymbol{\rho}))^2, \end{aligned} \tag{15}$$

where $\boldsymbol{\rho}$ is the parameter vector of the TSK fuzzy model and also the vector variable of the objective function $J(\boldsymbol{\rho})$, $\boldsymbol{\rho}^*$ is the optimal parameter vector of the TSK fuzzy model and the solution to Eq. (15), $y_k(\boldsymbol{\rho})$ is the process output at kth sampling interval, $y_{k,m}(\boldsymbol{\rho})$ is the fuzzy model output, the modeling error is

$$e_{k,m}(\boldsymbol{\rho}) = y_k(\boldsymbol{\rho}) - y_{k,m}(\boldsymbol{\rho}), \tag{16}$$

D is the feasible domain of $\boldsymbol{\rho}$, and N is the length of the discrete time horizon. The elements of the vector $\boldsymbol{\rho}$ are a part of the parameters of the input m.f. parameters.

Step 5. The nature-inspired optimization algorithms are applied to solve the optimization problem (15) that leads to the vector $\boldsymbol{\rho}^*$. The elements of the vector $\boldsymbol{\rho}^*$ are the optimal input m.f. parameters, and they belong to the optimal TSK fuzzy model.

Nature-inspired optimization algorithms

SA and PSO are the two nature-inspired optimization algorithms employed in the numerical solving of the optimization problem defined in Eq. (15). These algorithms are presented briefly as follows.

Simulated Annealing algorithm

The operating mechanism of SA algorithms uses a probabilistic framework to accept the solution on the basis of the analogy with the temperature decrease in metallurgy [71,72]. Considering the initial solution represented by the vector $\varphi \in \mathbf{R}^q$, with the fitness value $g(\varphi)$ of the fitness function g, $g: \mathbf{R}^q \to \mathbf{R}$, the new probable solution is represented by the vector $\psi \in \mathbf{R}^q$ which is chosen such that to belong to the vicinity of φ. As proposed in Ref. [44] and applied to the optimal tuning of TSK fuzzy controllers and in Refs. [2,4] to the optimal tuning of TSK fuzzy models for servo systems and magnetic levitation systems, in order to make the SA algorithm more computationally efficient we define two additional iteration indices, namely the success rate s_r and 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 fitness function changes for a preset number of times at the same temperature level. The rejection rate r_r is an alternative index to assess the convergence of the algorithm, and it is reset only when small values of the fitness function are found and not when the temperature cools.

Our SA algorithm is adapted from the SA algorithms given in Refs. [1–4,43,44] and applied to optimal control and modeling. The steps of our SA algorithm are

Step A. The initial solution is generated, i.e., the following operations are conducted:

- A random initial solution φ is generated and its fitness value $g(\varphi)$ is calculated.
- The minimum temperature $\theta_{\min}$ is set.
- Using the notation μ for the current iteration index, the following parameters are initialized: the maximum number of iterations $\mu_{\min}$, the maximum accepted success rate $s_{r\max}$, the maximum accepted rejection rate $r_{r\max}$, and the minimum accepted value of the fitness function $g_{\min}$.
- The initial temperature θ_0 is set. θ_0 is the value of the temperature θ_μ for $\mu = 0$.
- The initial rejection rate r_r is set to $r_r = 0$.

Step B. The initial value of the iteration index μ is set to $\mu = 0$ and the initial success rate s_r is set to $s_r = 0$.

Step C. A new probable solution ψ is generated in the vicinity of φ by disturbing φ, and its fitness value $g(\psi)$ is calculated.

Step D. The decision on accepting or not the new solution is made. With this regard, the change of fitness is computed as the difference $\Delta g_{\psi\varphi}$

$$\Delta g_{\psi\varphi} = g(\psi) - g(\varphi). \tag{17}$$

If $\Delta g_{\psi\varphi} \le 0$, $\varphi = \psi$ is accepted as the new vector solution. Otherwise, the random parameter q_μ, $0 \le q_\mu \le 1$, is set, and the probability p_ψ of ψ to be the next solution is calculated

$$p_\psi = \exp\left(\frac{-\Delta g_{\psi\varphi}}{\theta_\mu}\right). \tag{18}$$

If $p_\psi > q_\mu$, $\varphi = \psi$ is the new solution.

Step E. If the new solution is accepted, the new solution and g are updated, μ is incremented and r_r is reset to $r_r = 0$. Otherwise, r_r is incremented. If r_r has reached its maximum value $r_{r\max}$, the algorithm jumps to step I; otherwise, the algorithm continues with the next step.

Step F. s_r is incremented. If s_r has reached its maximum value $s_{r\max}$, the algorithm continues with the next step; otherwise, μ is incremented. If μ has reached its maximum value $\mu_{\max}$, the

algorithm continues with the next step; otherwise, the algorithm jumps to step C.

Step G. The temperature is decreased in terms of the temperature decrement rule, referred to also as the cooling schedule, which gives the next temperature $\theta_{\mu+1}$

$$\theta_{\mu+1} = \alpha_{cs}\theta_{\mu}, \tag{19}$$

where α_{cs} = const, $\alpha_{cs} < 1$.

Step H. If $\theta_{\mu} > \theta_{\min}$ or $g(\varphi) > g_{\min}$, the algorithm jumps to step C. Otherwise, the algorithm continues with the next step.

Step I. The obtained vector solution φ is validated by checking the following inequality-type constraint which guarantees that the TSK fuzzy model with the obtained model parameters $\boldsymbol{\rho} = \varphi$ ensures the convergence of the objective function:

$$|y_N(\boldsymbol{\rho}) - y_{N,m}(\boldsymbol{\rho})| \le \varepsilon_y |y_N(\boldsymbol{\rho}) - y_1(\boldsymbol{\rho})|, \tag{20}$$

where we recommend $\varepsilon_y = 0.001$ for a 2% settling time. The validation condition (20) and this recommendation are inspired from the optimal tuning of TSK fuzzy controllers by PSO, Gravitational Search and Charged System Search algorithms [44–46].

Step J. The algorithm is stopped, and the last validated vector solution φ is the solution to the optimization problem (15).

This SA algorithm is mapped onto the optimization problem (15) by means of the following relations:

- the relations between the parameter vectors φ and ψ in the SA algorithm and the parameter vector $\boldsymbol{\rho}$ in the optimization problem:

$$\psi = \boldsymbol{\rho}, \quad \varphi = \boldsymbol{\rho}, \tag{21}$$

- the relation between the fitness function g in the SA algorithm and the objective function J in the optimization problem:

$$g(\psi) = J(\boldsymbol{\rho}), \quad g(\varphi) = J(\boldsymbol{\rho}). \tag{22}$$

With this regard, q is the dimension of the search space.

The last new solution found will be the vector solution $\boldsymbol{\rho}^*$ to the optimization problem (15):

$$\boldsymbol{\rho}^* = \varphi. \tag{23}$$

Particle Swarm Optimization algorithm

The operating mechanism of PSO algorithms uses swarm particles which are characterized by two vectors, namely the particle position vector $\mathbf{X}_i$ and the particle velocity vector $\mathbf{V}_i$

$$\mathbf{X}_i = [\, x_i^1 \quad \dots \quad x_i^d \quad \dots \quad x_i^q \,]^T, \mathbf{V}_i = [\, v_i^1 \quad \dots \quad v_i^d \quad \dots \quad v_i^q \,]^T, \quad i = 1 \dots N_p, \tag{24}$$

where i, $i = 1 \dots N_p$, is the index of the current particle in the swarm, and N_p 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, $i = 1 \dots N_p$, and $\mathbf{P}_{g,Best}$ be the best swarm position vector:

$$\mathbf{P}_{i,Best} = [\, p_i^1 \quad \dots \quad p_i^d \quad \dots \quad p_i^q \,]^T,$$
$$\mathbf{P}_{g,Best} = [\, p_g^1 \quad \dots \quad p_g^d \quad \dots \quad p_g^q \,]^T, \; i = 1 \dots N_p. \tag{25}$$

The next particle velocity $v_i^d(\mu+1)$ and the next particle position $x_i^d(\mu+1)$ are obtained by the state-space equations [73,74]:

$$v_i^d(\mu+1) = w(\mu)v_i^d(\mu) + c_1 r_1 [p_g^d(\mu) - x_i^d(\mu)] + c_2 r_2 [p_i^d(\mu) - x_i^d(\mu)],$$
$$x_i^d(\mu+1) = x_i^d(\mu) + v_i^d(\mu+1), \quad d = 1 \dots q, \; i = 1 \dots N_p, \tag{26}$$

where r_1 and r_2 are uniformly distributed random variables, $0 \le r_1 \le 1$, $0 \le r_2 \le 1$, w is the inertia weight, and c_1 and c_2 are the weighting factors of the stochastic accelerations pulling the particles toward their final positions. The parameter $w(\mu)$ 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. The expression of $w(\mu)$ is

$$w(\mu) = w_{\max} - \mu(w_{\max} - w_{\min})/\mu_{\max}. \tag{27}$$

As mentioned in Ref. [44], if $w_{\max}$ is too small, particles may not explore sufficiently beyond local solutions. The minimum value of $w(\mu)$ in Eq. (27) is $w_{\min}$. Setting 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 c_1 and c_2 will result in abrupt movements toward or overshooting the target regions.

Our PSO algorithm is adapted from the PSO algorithm given in Ref. [44] and applied to the optimal tuning of TSK fuzzy controllers targeting a reduced parametric sensitivity. The steps of our PSO algorithm are

Step A. The q-dimensional search space is initialized by setting its boundaries. The maximum number of iterations $\mu_{\max}$ and the parameters in Eq. (26) are initialized.

Step B. The initial population of particles is generated by conducting the following operations:

- The particle position vector is initialized as a uniformly distributed random vector $\mathbf{X}_i$, $i = 1 \dots N_p$, that belongs to the search space, and the particle velocity vector $\mathbf{V}_i$, $i = 1 \dots N_p$, is initalized accounting for the boundaries of the search space.
- The best particle position vector $\mathbf{P}_{i,Best} = \mathbf{X}_i$, $i = 1 \dots N_p$, and the best swarm position vector $\mathbf{P}_{g,Best}$ are calculated.
- If $g(\mathbf{P}_{i,Best}) < g(\mathbf{P}_{g,Best})$, $i = 1 \dots N_p$, the best swarm position vector is updated in terms of $\mathbf{P}_{g,Best} = \mathbf{P}_{i,Best}$.

Step C. The particles' fitness is evaluated using the following relations that map the PSO algorithm onto the optimization problem (15):

- the relation between the agents' position vector $\mathbf{X}_i$ in the PSO algorithm and the parameter vector $\boldsymbol{\rho}$ in the optimization problem:

$$\mathbf{X}_i = \rho, \; i = 1 \dots N_p, \tag{28}$$

- the relation between the fitness function g in the PSO algorithm and the objective function J in the optimization problem:

$$g(\mathbf{X}_i) = J(\rho), \quad i = 1 \dots N_p. \tag{29}$$

Step D. The local best and the global best of the population are calculated in terms of

- If $g(\mathbf{X}_i) < g(\mathbf{P}_{i,Best})$, $i = 1 \dots N_p$, the best particle position vector $\mathbf{P}_{i,Best}$ is updated to $\mathbf{P}_{i,Best} = \mathbf{X}_i$.
- If $g(\mathbf{P}_{i,Best}) < g(\mathbf{P}_{g,Best})$, $i = 1 \dots N_p$, the best swarm position vector $\mathbf{P}_{g,Best}$ is updated to $\mathbf{P}_{g,Best} = \mathbf{P}_{i,Best}$.

Step E. The population is updated using (26).

Step F. The obtained vector solution $\mathbf{P}_{g,Best}$ is validated by checking the inequality-type constraint (20) which guarantees that the TSK fuzzy model with the obtained model parameters $\rho = \mathbf{P}_{g,Best}$ ensures the convergence of the objective function. This step is similar to the step I in the SA algorithm.

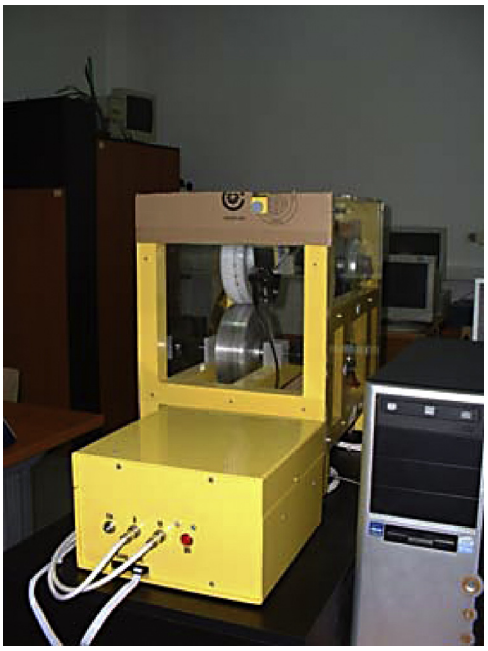

Fig. 2. ABS laboratory equipment experimental setup.

Step G. If $\mu < \mu_{\max}$, the algorithm jumps to step B. Otherwise, the algorithm is stopped, and the best found solution will be the vector solution $\boldsymbol{\rho}^*$ to the optimization problem (15):

$$\rho^* = \mathbf{P}_{g,Best}. \tag{30}$$

Experimental results

The fuzzy modeling approach presented in the section following the Background section has been applied in combination with the two nature-inspired optimization algorithms presented in the previous section in order to obtain the TSK fuzzy models of the ABS laboratory equipment. A part of the results and of the implementation details is presented as follows.

The results presented in this section have been obtained on the ABS laboratory equipment experimental setup shown in Fig. 2. The ABS laboratory equipment experimental setup belongs to the Intelligent Control Systems Laboratory of the Politehnica University of Timisoara, Romania.

Table 1
Definition of operating points of TSK fuzzy models.

Model number	Operating points	n_R
1	$A_9(\lambda = 1, x_2 = 50, M_1 = 10)$, $A_{10}(\lambda = 1, x_2 = 150, M_1 = 10)$	4
2	$A_1(\lambda = 0.1, x_2 = 50, M_1 = 10)$, $A_2(\lambda = 0.1, x_2 = 150, M_1 = 10)$, $A_5(\lambda = 0.4, x_2 = 50, M_1 = 10)$, $A_6(\lambda = 0.4, x_2 = 150, M_1 = 10)$, $A_9(\lambda = 1, x_2 = 50, M_1 = 10)$, $A_{10}(\lambda = 1, x_2 = 150, M_1 = 10)$	6
3	$A_1(\lambda = 0.1, x_2 = 50, M_1 = 10)$, $A_2(\lambda = 0.1, x_2 = 150, M_1 = 10)$, $A_2(\lambda = 0.2, x_2 = 50, M_1 = 10)$, $A_4(\lambda = 0.2, x_2 = 150, M_1 = 10)$, $A_5(\lambda = 0.4, x_2 = 50, M_1 = 10)$, $A_6(\lambda = 0.4, x_2 = 150, M_1 = 10)$, $A_7(\lambda = 0.8, x_2 = 50, M_1 = 10)$, $A_8(\lambda = 0.8, x_2 = 150, M_1 = 10)$, $A_9(\lambda = 1, x_2 = 50, M_1 = 10)$, $A_{10}(\lambda = 1, x_2 = 150, M_1 = 10)$	10

Setting the sampling period $T_s = 0.01$ s, the control signal u considered as the input of the ABS has been generated as two weighted sums (one for training and one for validation) of pseudo-random binary signals in order to cover different ranges of magnitudes and to benefit from the advantages of pseudo-random binary signals used as input signals in system identification (spectral characteristics, persistent excitation, noise reduction). The evolutions of the control signal versus time are presented in Fig. 3, which gives the input data for training, and in Fig. 4, which gives the input data for validation (testing).

The input signals illustrated in Figs. 3 and 4 have been applied to the ABS laboratory equipment in order to generate

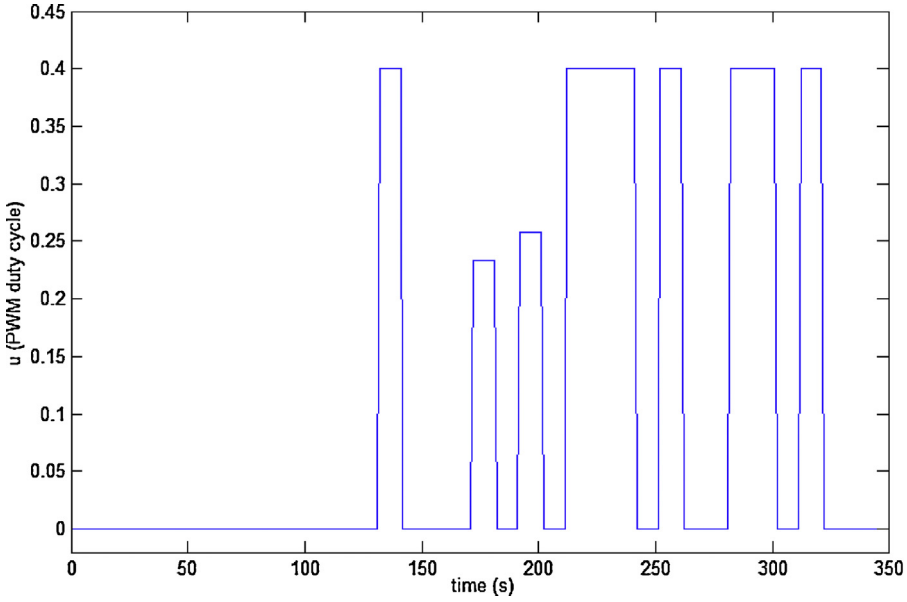

Fig. 3. Control signal versus time used in the training data.

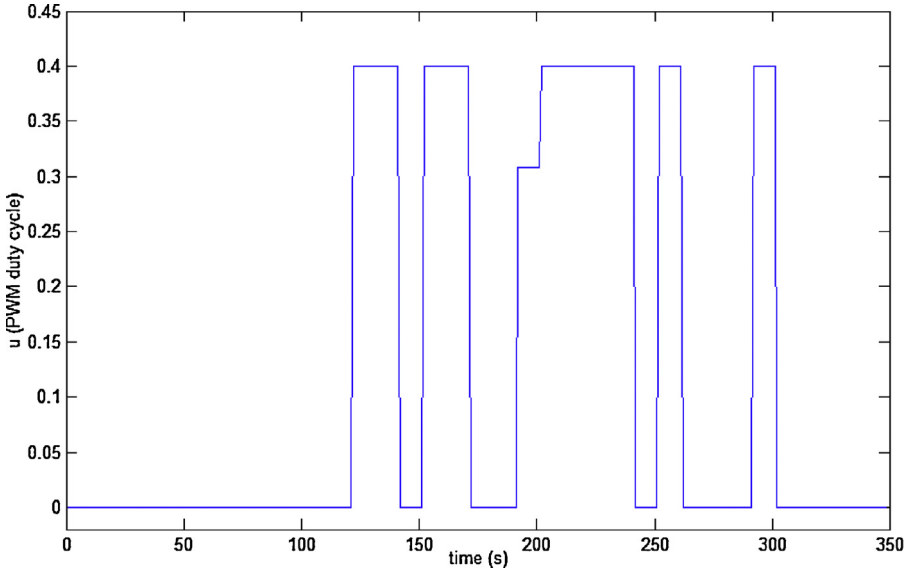

Fig. 4. Control signal versus time used in the validation (testing) data.

the input–output data points. Fig. 3 outlines a total number of $N = 35{,}000$ data points which represent the training data, and Fig. 4 outlines a total number of $N = 35{,}000$ data points which represent the validation data.

assigned to the input variables and defined as follows. The first input variable, λ, uses five LTs, with the notation, $LT_{\lambda,l}$, $l = 1 \ldots 5$, characterized by the triangular m.f.s

$$\mu_{LT_{\lambda,l}} : [0, 1.1] \to [0, 1],$$

$$\mu_{LT_{\lambda,l}}(\lambda) = \begin{cases} 0, & \lambda < a_{\lambda,l}, \\ 1 + \dfrac{(\lambda - b_{\lambda,l})}{(b_{\lambda,l} - a_{\lambda,l})}, & a_{\lambda,l} \le \lambda < b_{\lambda,l}, \\ 1 - \dfrac{(\lambda - b_{\lambda,l})}{(c_{\lambda,l} - b_{\lambda,l})}, & b_{\lambda,l} \le \lambda < c_{\lambda,l}, \\ 0, & \lambda \ge c_{\lambda,l}, \end{cases} \quad a_{\lambda,l} < b_{\lambda,l} < c_{\lambda,l}, \quad l = 1 \ldots 5. \tag{31}$$

The five steps of the fuzzy modeling approach have been applied to derive the three optimal TSK fuzzy models specified in Table 1.

The step 1 of the fuzzy modeling approach starts with setting the largest domains of variation of the state variables in all ABS operating regimes, $0 \le \lambda \le 1.1$, $0 \le x_2 \le 180$ and $0 \le M_1 \le 11$. The fuzzification modules in the TSK fuzzy models employ the LTs

The parameters $a_{\lambda,l}, j = 1 \ldots 5$, and $c_{\lambda,l}, j = 1 \ldots 5$, are variable, they are the feet of the triangular m.f.s defined in Eq. (31), and they belong to the vector variable $\boldsymbol{\rho}$ of the objective function in Eq. (15). The parameters $b_{\lambda,j}$, $j = 1 \ldots 5$, which stand for the modal values of the m.f.s, are fixed, and their values are [29] $b_{\lambda,1} = 0.1$, $b_{\lambda,2} = 0.2$, $b_{\lambda,3} = 0.4$, $b_{\lambda,4} = 0.8$, and $b_{\lambda,5} = 1$.

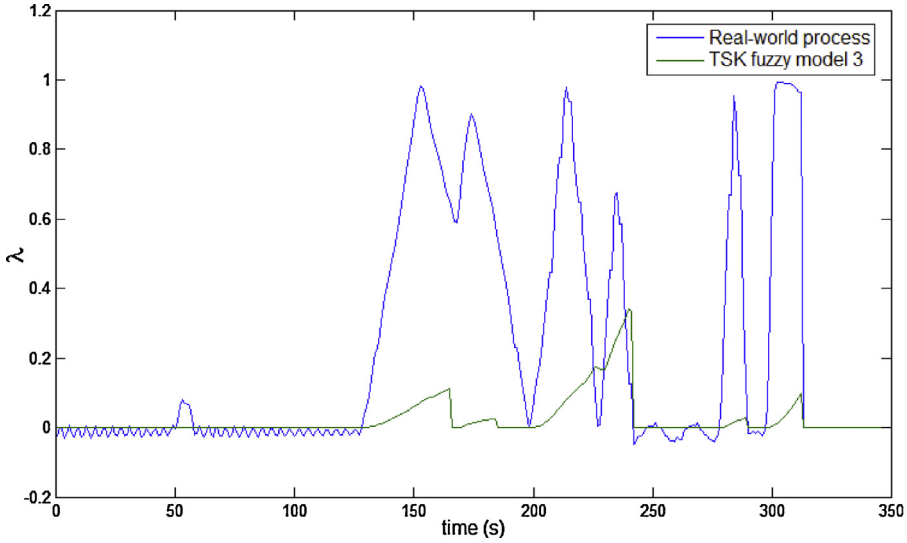

Fig. 5. Real-time experimental results: wheel slip λ versus time for the initial TSK fuzzy model 3 and for the real-world process considering the training data set.

The second input variable, x_2, uses the two LTs $LT_{x_2,p}$, $p \in \{1, 2\}$, with the triangular m.f.s

$$\mu_{LT_{x_2,p}} : [0, 180] \to [0, 1],$$

$$\mu_{LT_{x_2,p}}(x_2) = \begin{cases} 0, & x_2 < a_{x_2,p}, \\ 1 + \dfrac{(x_2 - b_{x_2,p})}{(b_{x_2,p} - a_{x_2,p})}, & a_{x_2,p} \le x_2 < b_{x_2,p}, \\ 1 - \dfrac{(x_2 - b_{x_2,p})}{(c_{x_2,p} - b_{x_2,p})}, & b_{x_2,p} \le x_2 < c_{x_2,p}, \\ 0, & x_2 \ge c_{x_2,p}, \end{cases} \quad a_{x_2,p} < b_{x_2,p} < c_{x_2,p}, \quad p \in \{1, 2\}. \tag{32}$$

The parameters $a_{x_2,p}$, $p \in \{1, 2\}$, and $c_{x_2,p}$, $p \in \{1, 2\}$, are variable, they are the feet of the triangular m.f.s defined in Eq. (32), and they belong to $\boldsymbol{\rho}$. The parameters $b_{x_2,j}$, $j = 1...2$, are fixed, and their values are [1,5] $b_{x_2,1} = 50$ and $b_{x_2,2} = 150$.

The third input variable, M_1, uses the single LT $LT_{M_1,1}$, with the triangular m.f.

$$\mu_{LT_{M_1,1}} : [0, 11] \to [0, 1],$$

$$\mu_{LT_{M_1,1}}(M_1) = \begin{cases} 0, & M_1 < a_{M_1,1}, \\ 1 + \dfrac{(M_1 - b_{M_1,1})}{(b_{M_1,1} - a_{x_2,p})}, & a_{M_1,1} \le M_1 < b_{M_1,1}, \\ 1 - \dfrac{(M_1 - b_{M_1,1})}{(c_{M_1,1} - b_{x_2,p})}, & b_{M_1,1} \le M_1 < c_{M_1,1}, \\ 0, & M_1 \ge c_{M_1,1}, \end{cases} \quad a_{M_1,1} < b_{M_1,1} < c_{M_1,1}. \tag{33}$$

The parameters $a_{M_1,1}$ and $c_{M_1,1}$ are variable, they are the feet of the triangular m.f. defined in Eq. (33), and they belong to $\boldsymbol{\rho}$. The parameter $b_{M_1,1}$ is fixed and its value is set according to Refs. [1,5], $b_{M_1,1} = 10$.

The state-space model matrices in the consequents of the rules R^1 and R^{10} of the discrete-time TSK fuzzy model 3, i.e., $n_R = 10$, derived in the step 3 of the fuzzy modeling approach, are

$$\begin{aligned} \mathbf{A}_{d,1} &= \begin{bmatrix} 0.9441 & -0.0025 & 0.0194 \\ -1.0335 & 1.0012 & -0.0601 \\ 0 & 0 & 0.8157 \end{bmatrix}, \mathbf{B}_{d,1} = \begin{bmatrix} 0.0021 \\ -0.0059 \\ 0.1843 \end{bmatrix}, \mathbf{C}_{d,1} = [\,1 \quad 0 \quad 0\,], \\ \mathbf{A}_{d,10} &= \begin{bmatrix} 1.0041 & -0.0003 & 0.0069 \\ -0.3192 & 1 & -0.0519 \\ 0 & 0 & 0.8157 \end{bmatrix}, \mathbf{B}_{d,10} = \begin{bmatrix} 0.0007 \\ -0.0054 \\ 0.1843 \end{bmatrix}, \mathbf{C}_{d,10} = [\,1 \quad 0 \quad 0\,]. \end{aligned} \tag{34}$$

Summing up the details concerning the input m.f.s presented in Eqs. (31)–(33), the parameter vector of the TSK fuzzy model 3 (the vector variable of the objective function) is

$$\rho = [\, a_{\lambda,1} \quad c_{\lambda,1} \quad a_{\lambda,2} \quad c_{\lambda,2} \quad a_{\lambda,3} \quad c_{\lambda,3} \quad a_{\lambda,4} \quad c_{\lambda,4} \quad a_{\lambda,5} \quad c_{\lambda,5} \quad a_{x_2,1} \quad c_{x_2,1} \quad a_{x_2,2} \quad c_{x_2,2} \quad a_{M_1,1} \quad c_{M_1,1} \,]^T, \tag{35}$$

the initial value of this parameter vector, which corresponds to the initial TSK fuzzy model 3, is

$$\rho = [\, 0 \quad 0.2 \quad 0.1 \quad 0.4 \quad 0.2 \quad 0.8 \quad 0.4 \quad 1 \quad 0.8 \quad 1.1 \quad 0 \quad 150 \quad 50 \quad 180 \quad 0 \quad 11 \,]^T. \tag{36}$$

The feasible domain in Eq. (15) is

$$\begin{aligned} D = {} & [-0.01, 0.01] \times [0.15, 0.25] \times [0.05, 0.15] \times [0.35, 0.45] \times [0.15, 0.25] \times [0.75, 0.85] \times [0.35, 0.45] \times [0.95, 1.05] \\ & \times [0.75, 0.85] \times [1.05, 1.15] \times [-0.1, 0.1] \times [130.9, 170.1] \times [40, 60] \times [170, 190] \times [-0.5, 0.5] \times [10.5, 11.5] \in \mathbf{R}^{16}. \end{aligned} \tag{37}$$

Eq. (37) highlights the $q = 16$-dimensional search space.

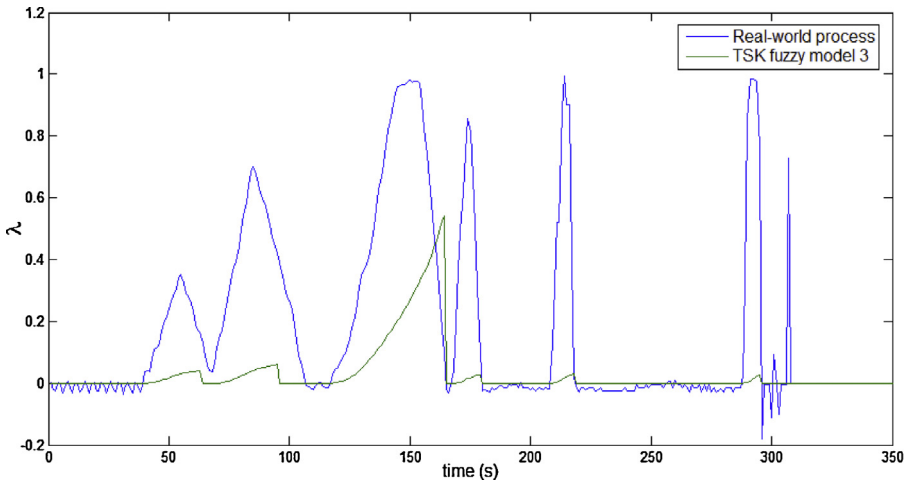

Fig. 6. Real-time experimental results: wheel slip λ versus time for the initial TSK fuzzy model 3 and for the real-world process considering the validation data set.

The first set of experimental results is presented in Figs. 5 and 6 as the outputs of the real-world process (the ABS laboratory equipment) and of the initial TSK fuzzy model 3 before the application of the nature-inspired algorithms (i.e., with the parameter vector $\boldsymbol{\rho}$ given in Eq. (36)), for the training and validation data set, respectively. Figs. 5 and 6 illustrate the large difference between the TSK fuzzy model 3 and the real-world process, and they justify the need for the optimal tuning of the TSK fuzzy model.

Some details concerning the application of the two nature-inspired algorithms described in the previous section and involved in the step 5 of the fuzzy modeling approach will be presented as follows. The two nature-inspired algorithms, namely SA and PSO, aim the calculation of the vector solution $\boldsymbol{\rho}^*$ to the optimization problem (15).

The maximum success and rejection rates in the SA algorithm were set to $r_{\text{rmax}} = 100$ and $s_{\text{rmax}} = 50$. The SA algorithm was stopped after $\mu_{\max} = 80$ iterations, with the final temperature $\theta_{80} = 0.093246$. The initial temperature was set to $\theta_0 = 1$, and the minimum temperature was set to $\theta_{\min} = 10^{-8}$. The value of the parameter in the temperature decrement rule was set to $\alpha_{cs} = 0.8$. The minimum accepted value of the fitness function (i.e., of the objective function) was set to $g_{\min} = 10^{-4}$.

The evolution of the objective function defined in Eq. (15) versus the iteration number in the SA algorithm is presented in Fig. 7 for the TSK fuzzy model 3 considering the validation data set. Fig. 7 shows that the solution to Eq. (15) obtained by our SA algorithm offers a strong decrease of the objective function. Fig. 7 also shows that the minimum of the objective function is not reached, but the further decrease of the objective for the given TSK fuzzy model structure can be achieved by increasing the number of iterations.

The final solution obtained by the SA algorithm is

$$\rho^* = [\,0.0086 \quad 0.1614 \quad 0.09901 \quad 0.4067 \quad 0.2295 \quad 0.4019 \quad 0.9578 \quad 0.847 \quad 1.12 \quad 0.02202 \quad 169 \quad 59.85 \quad 180.5 \quad -0.3227 \quad 11.04\,]^T. \tag{38}$$

The second set of experimental results is presented in Figs. 8 and 9 as the outputs of the real-world process and of the TSK fuzzy model 3 after the application of the SA algorithm (i.e., with the parameter vector $\boldsymbol{\rho}^*$ given in Eq. (38)), for the training and validation data set, respectively. Figs. 8 and 9 clearly show that the performance of the TSK fuzzy model 3 is strongly improved by the application of our SA algorithm from the point of view the modeling errors. The modeling errors are alleviated in both training and validation data sets.

The values of the parameters of the PSO algorithm were set to $N_p = 20$, $\mu_{\max} = 80$ (equal to $\mu_{\max}$ considered in the SA algorithm in order to ensure the fair comparison with the results obtained by the SA algorithm), $c_1 = c_2 = 0.3$, $w_{\min} = 0.4$ and $w_{\max} = 0.9$. These parameter values ensure a good tradeoff to PSO algorithm's convergence speed and accuracy as recommended in Ref. [44].

The evolution of the objective function defined in Eq. (15) versus the iteration number in the PSO algorithm is presented in Fig. 10 for the TSK fuzzy model 3 considering the validation data set. Fig. 10 shows, similar to the case of the SA algorithm, that the solution to Eq. (15) obtained by our PSO algorithm offers a strong decrease of the objective function. In addition, Fig. 10 points out that the value of the objective function is smaller compared to the value of the objective function obtained by the SA algorithm (presented in Fig. 7) and that the further decrease of the objective can be achieved by increasing the number of iterations.

The final solution obtained by the PSO algorithm is

$$\rho^* = \left[\,0.0025 \quad 0.2381 \quad 0.1289 \quad 0.35 \quad 0.25 \quad 0.8245 \quad 0.4016 \quad 0.9543 \quad 0.8475 \quad 1.0577 \quad 0.089 \quad 169.54 \quad 48.21 \quad 170.29 \quad -0.4517 \quad 11.0068\,\right]^T. \tag{39}$$

The third set of experimental results is presented in Figs. 11 and 12. For comparison reasons, Figs. 11 and 12 illustrate the outputs of the TSK fuzzy model 3 after the application of the SA algorithm and of the TSK fuzzy model 3 after the application of the PSO algorithm (i.e., with the parameter vector $\boldsymbol{\rho}^*$ given in Eq. (39)) for the training and validation data set, respectively. Figs. 11 and 12 highlight:

- the performance improvement of the TSK fuzzy model 3 tuned by the PSO algorithm with respect to the initial TSK fuzzy model (versus Figs. 5 and 6),
- the behaviors of the TSK fuzzy model 3 tuned by two nature-inspired algorithms with quite different operating principles are close,

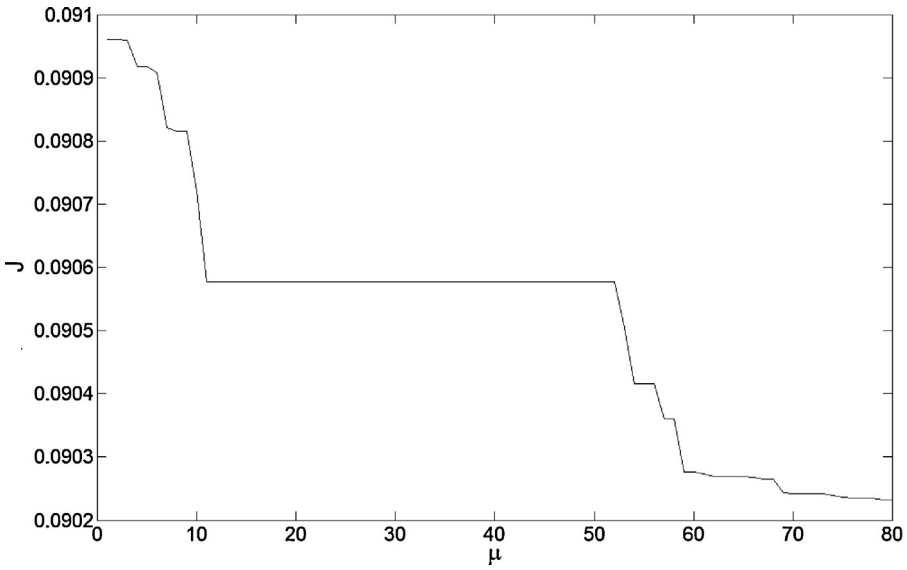

Fig. 7. Objective function J versus the iteration number λ of the SA algorithm for the TSK fuzzy model 3 considering the validation data set.

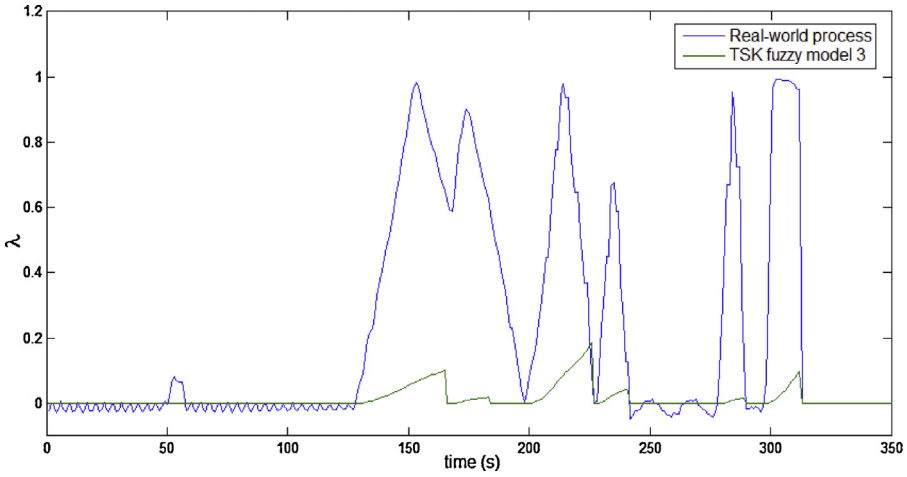

Fig. 8. Real-time experimental results: wheel slip λ versus time for the TSK fuzzy model 3 after optimization by the SA algorithm and for the real-world process considering the training data set.

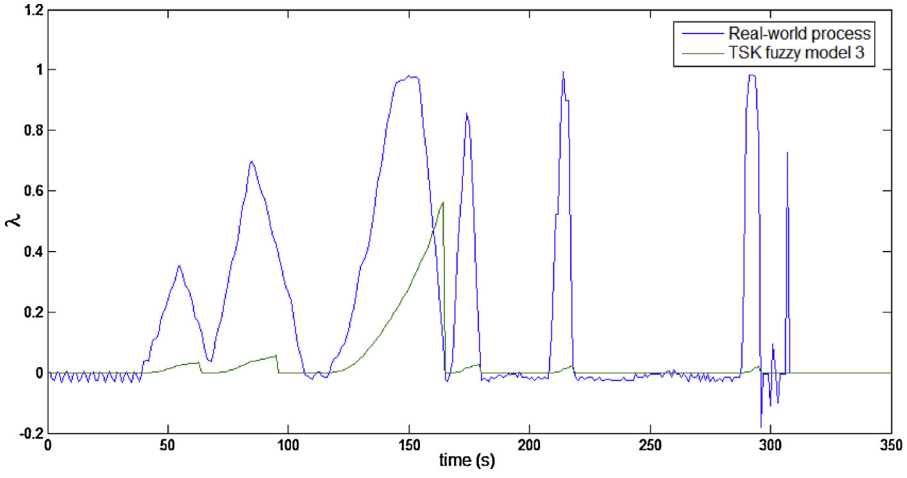

Fig. 9. Real-time experimental results: wheel slip λ versus time for the TSK fuzzy model 3 after optimization by the SA algorithm and for the real-world process considering the validation data set.

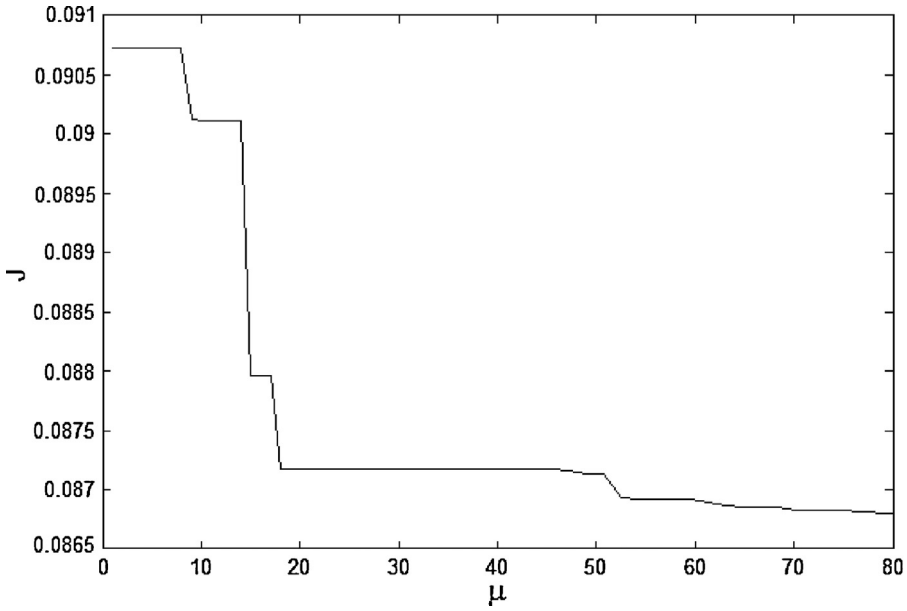

Fig. 10. Objective function J versus iteration number μ of the PSO algorithm for the TSK fuzzy model 3 considering the validation data set.

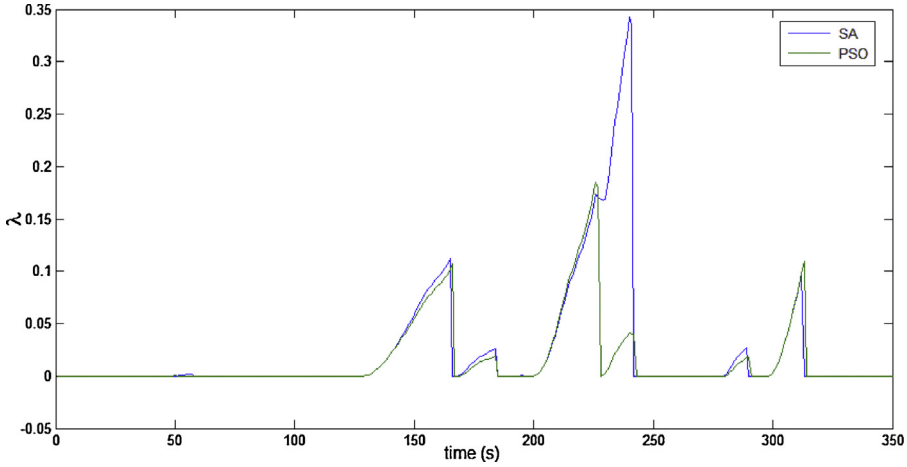

Fig. 11. Real-time experimental results: wheel slip λ versus time for the TSK fuzzy model 3 after optimization by the SA algorithm and after optimization by the PSO algorithm considering the training data set.

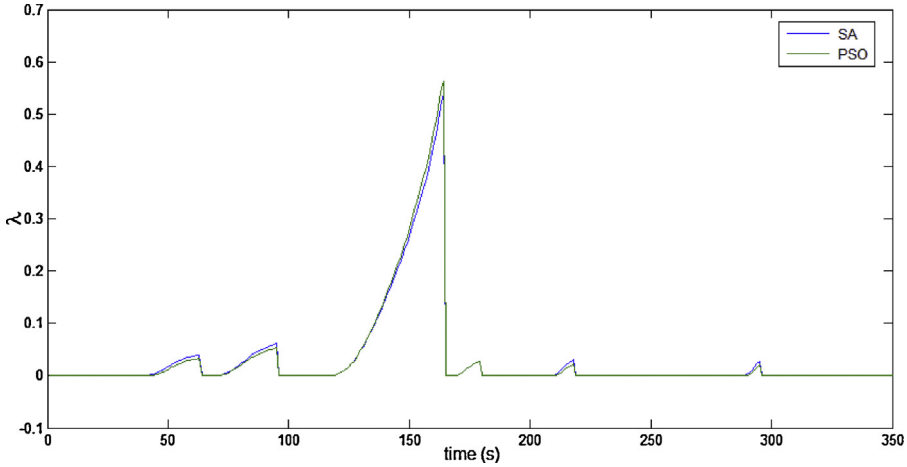

Fig. 12. Real-time experimental results: wheel slip λ versus time for the TSK fuzzy model 3 after optimization by the SA algorithm and after optimization by the PSO algorithm considering the validation data set.

Table 2
Results for TSK fuzzy models 1–3 on the training data and on the validation data.

Model number	Nature-inspired algorithm	RMSE training	RMSE validation
1	SA	0.39833	0.41376
1	PSO	0.38368	0.40785
1	GSA	0.36929	0.42375
2	SA	0.33685	0.36879
2	PSO	0.32122	0.34193
2	GSA	0.30917	0.35526
3	SA	0.28532	0.30038
3	PSO	0.27481	0.29462
3	GSA	0.26450	0.30610

- the TSK fuzzy model 3 tuned by the PSO algorithm exhibits a better behavior (i.e., closer to the real-world process response) compared to the TSK fuzzy model 3 tuned by the SA algorithm.

The algorithms and the model performance have been compared with the TSK fuzzy model 1 and with the TSK fuzzy model 2. The comparative analysis of the models is carried out in terms of the root mean square error (RMSE) between the model output (the wheel slip) of the TSK fuzzy models and of the real-world process. The RMSE is viewed as a global performance index defined as

$$RMSE = \sqrt{J(\rho^*)} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k(\rho^*) - y_{k,m}(\rho^*))^2} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(e_{k,m}(\rho^*))^2}. \quad (40)$$

The RMSE has been computed and measured for the training data and for the validation (testing) data.

The results obtained by the three TSK fuzzy models tuned by the SA algorithm and by the PSO algorithm are summarized in Table 2 for the training data and for the validation data. In addition, the results of the proposed method are compared with another algorithm reviewed in Section 2, namely a Gravitational Search Algorithm (GSA) is employed in the numerical solving of the optimization problem defined in Eq. (15). The values of the parameters of the GSA were set to: number of agents $N_p = 20$ (equal to N_p considered in the PSO algorithm for the fair comparison of the results), maximum number of iterations $\mu_{max} = 80$ (equal to μ_{max} considered in both the SA algorithm and the PSO algorithm for the fair comparison of the results), depreciation law of the gravitational constant $g(k)$ with the advance of the GSA's iterations expressed as $g(\mu) = g_0 \exp(-\zeta\mu/\mu_{max})$, with the initial value of the gravitational constant $g_0 = 100$ and the parameter $\zeta = 20$, and the parameter ε in the denominator of the force the force acting between to agents set to $\varepsilon = 0.0001$. These parameter values ensure a good tradeoff to GSA's convergence speed and accuracy as recommended in Ref. [44]. The results obtained by the application of the GSA are also included in Table 2.

Table 2 and Figs. 5–12 show that the best performance on the training data is ensured by the TSK fuzzy model 3 tuned by the GSA. The best performance on both the training data and the validation data is ensured by the TSK fuzzy model 3 tuned by the PSO algorithm. The TSK fuzzy model 3 offers the lowest value of RMSE and of J. Table 2 shows the performance enhancement exhibited by the final TSK fuzzy models tuned by our fuzzy modeling approach compared to the initial TSK fuzzy models on both the training data and the validation data for the same process inputs.

The results presented in Table 2 and in Figs. 5–12 also show that the performance of the proposed TSK fuzzy models are consistent with both training data and validation data. The further performance improvement can be ensured by increasing the number of iterations, by increasing the number of operating points, by considering other shapes of input m.f.s as those applied in Ref. [4] for another process and by inserting other nature-inspired optimization algorithms in our fuzzy modeling approach.

Conclusions

This paper has suggested a set of TSK fuzzy models dedicated to ABSs. The models are derived by the synergy of fuzzy logic and nature-inspired optimization in terms of the nature-inspired optimal tuning of the input membership functions of TSK fuzzy models in the framework of appropriately defined optimization problems which target the minimization of objective functions expressed as the average of squared modeling errors over the time horizon. Our synergy is formulated as a novel fuzzy modeling approach based on the combination of the modal equivalences principle and on the optimal tuning of a part of the input m.f. parameters in terms of two nature-inspired algorithms The fuzzy modeling approach is assisted by a new operating point selection algorithm to guide modeling.

The main advantages of our approach are the simplicity of TSK fuzzy models, their consistency with both training data and testing data, and the simplicity and transparency of the fuzzy modeling approach. These advantages should be also emphasized in other suggestive applications.

Our fuzzy modeling approach is different and advantageous with respect to the popular sector nonlinearity approach [75]. The difference concerns the fact that the bounds of the domains of the scheduling (variables) are not accounted for, but the important operating points are taken into consideration. Therefore, our approach offers better highlighting of system's behavior in the important operating regimes.

The TSK fuzzy models are validated against experimental data which correspond to an ABS laboratory equipment as a representative nonlinear dynamic system. The comparison of experimental results shows the performance improvement ensured by the proposed TSK fuzzy models after the application of the nature-inspired algorithms. The comparison can continue accounting for other state-of-the-art algorithms for evolving fuzzy systems [60–63] and for other nature-inspired algorithms accounting for several constraints [31–35,76,77].

Future research will deal with the extension of the fuzzy modeling approach to the modeling of Multi Input-Multi Output nonlinear dynamic systems. The experiment-based estimation of the gradients of the objective functions will be included in order to ensure the online identification of closed-loop fuzzy control systems in order to next carry out the model-based design of stable and optimal fuzzy control systems.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, by a grant in the framework of the Partnerships in priority areas – PN II program of the Romanian National Authority for Scientific Research ANCS, CNDI – UEFISCDI, project number PN-II-PT-PCCA-2011-3.2-0732, and by a grant from the NSERC of Canada. The cooperation between the Óbuda University, Budapest, Hungary, the University of Ljubljana, Slovenia, and the Politehnica University of Timisoara, Romania, in the framework of the Hungarian-Romanian and Slovenian-Romanian Intergovernmental Science & Technology Cooperation Programs is acknowledged. The strong support of M.Sc. Ramona-Bianca Grad,

Dr. Mircea-Bogdan Radac and Dr. Claudia-Adina Dragos offered in conducting the experiments is also acknowledged.

References

[1] R.-C. David, R.-B. Grad, R.-E. Precup, M.-B. Radac, C.-A. Dragos, E.M. Petriu, An approach to fuzzy modeling of anti-lock braking systems, in: V. Snášel, P. Krömer, M. Köppen, G. Schaefer (Eds.), Soft Computing in Industrial Applications, Advances in Intelligent Systems and Computing, vol. 223, Springer-Verlag, Cham, Heidelberg, New York, Dordrecht, London, 2014, pp. 83–93.

[2] R.-E. Precup, M.-B. Radac, E.M. Petriu, C.-A. Dragos, S. Preitl, Simulated annealing approach to fuzzy modeling of servo systems, in: Proc. 2013 IEEE International Conference on Cybernetics, Lausanne, Switzerland, 2013, pp. 267–272.

[3] R.-C. David, C.-A. Dragos, R.-G. Bulzan, R.-E. Precup, E.M. Petriu, M.-B. Radac, An approach to fuzzy modeling of magnetic levitation systems, Int. J. Artif. Intell. 9 (2012) 1–18.

[4] C.-A. Dragos, R.-E. Precup, R.-C. David, S. Preitl, A.-I. Stinean, E.M. Petriu, Simulated annealing-based optimization of fuzzy models for magnetic levitation systems, in: Proc. 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, Edmonton, AB, Canada, 2013, pp. 286–291.

[5] R.-E. Precup, M.-C. Sabau, C.-A. Dragos, M.-B. Radac, L.-O. Fedorovici, E.M. Petriu, Particle swarm optimization of fuzzy models for anti-lock braking systems, in: Proc. IEEE Conference on Evolving and Adaptive Intelligent Systems, Linz, Austria, paper index 05, 2014, 6 pp.

[6] I. Škrjanc, S. Blažič, O.E. Agamennoni, Interval fuzzy model identification using l_∞-norm, IEEE Trans. Fuzzy Syst. 13 (2005) 561–568.

[7] Z.C. Johanyák, S. Kovács, Distance based similarity measures of fuzzy sets, in: Proc. 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herl'any, Slovakia, 2005, pp. 265–276.

[8] S. Oblak, I. Škrjanc, S. Blažič, Fault detection for nonlinear systems with uncertain parameters based on the interval fuzzy model, Eng. Appl. Artif. Intell. 20 (2007) 503–510.

[9] D. Tikk, Z.C. Johanyák, S. Kovács, K.W. Wong, Fuzzy rule interpolation and extrapolation techniques: Criteria and evaluation guidelines, J. Adv. Comput. Intell. Intell. Inform. 15 (2011) 254–263.

[10] P. Angelov, R. Yager, Density-based averaging – a new operator for data fusion, Inf. Syst. 222 (2013) 163–174.

[11] K. Lamár, J. Neszveda, Average probability of failure of aperiodically operated devices, Acta Polytech. Hung. 10 (2013) 153–167.

[12] P. Baranyi, D. Tikk, Y. Yam, R.J. Patton, From differential equations to PDC controller design via numerical transformation, Comput. Ind. 51 (2003) 281–297.

[13] P. Baranyi, TP model transformation as a way to LMI-based controller design, IEEE Trans. Ind. Electron. 51 (2004) 387–400.

[14] P. Baranyi, Y. Yam, P. Varlaki, TP Model Transformation in Polytopic Model-Based Control, Taylor & Francis, Boca Raton, FL, 2013.

[15] R.-E. Precup, H. Hellendoorn, A survey on industrial applications of fuzzy control, Comput. Ind. 62 (2011) 213–226.

[16] T. Zheng, F. Ma, K. Zhang, Estimation of reference vehicle speed based on T-S fuzzy model, in: Proc. International Conference on Advanced in Control Engineering and Information Science, Dali, China, vol. 15, 2011, pp. 188–193.

[17] W.-Y. Wang, M.-C. Chen, S.-F. Su, Hierarchical T-S fuzzy-neural control of anti-lock braking system and active suspension in a vehicle, Automatica 48 (2012) 1698–1706.

[18] R.-E. Precup, S.V. Spataru, M.-B. Radac, E.M. Petriu, S. Preitl, C.-A. Dragos, R.-C. David, Experimental results of model-based fuzzy control solutions for a laboratory antilock braking system, in: Z.S. Hippe, J.L. Kulikowski, T. Mroczek (Eds.), Human-Computer Systems Interaction: Backgrounds and Applications 2. Part 2. Advances in Intelligent and Soft Computing, vol. 99, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 223–234.

[19] P. Naderi, A. Farhadi, M. Mirsalim, T. Mohammadi, Anti-lock and anti-slip braking system, using fuzzy logic and sliding mode controllers, in: Proc. IEEE Vehicle Power and Propulsion Conference, Lille, France, 2010, 6 pp.

[20] A.V. Topalov, Y. Oniz, E. Kayacan, O. Kaynak, Neuro-fuzzy control of antilock braking system using sliding mode incremental learning algorithm, Neurocomputing 74 (2011) 1883–1893.

[21] D. Aleksendrić, Ž. Jakovljević, V. Ćirović, Intelligent control of braking process, Expert Syst. Appl. 39 (2012) 11758–11765.

[22] V. Ćirović, D. Aleksendrić, Adaptive neuro-fuzzy wheel slip control, Expert Syst. Appl. 40 (2013) 5197–5209.

[23] S. Kar, S. Das, P.K. Ghosh, Applications of neuro fuzzy systems: a brief review and future outline, Appl. Soft Comput. 15 (2014) 243–259.

[24] I. Rojas, F. Rojas, H. Pomares, L.J. Herrera, J. González, O. Valenzuela, The synergy between classical and soft-computing techniques for time series prediction, in: R. Monroy, G. Arroyo-Figueroa, L.E. Sucar, H. Sossa (Eds.), MICAI 2004: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2972, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 30–39.

[25] O. Uluyol, K. Kim, E.O. Nwadiogbu, Synergistic use of soft computing technologies for fault detection in gas turbine engines, IEEE Trans. Syst. Man Cybern. C. Appl. Rev. 36 (2006) 476–484.

[26] W. Pedrycz, B.-J. Park, S.-K. Oh, The design of granular classifiers: a study in the synergy of interval calculus and fuzzy sets in pattern recognition, Pattern Recogn. 41 (2008) 3720–3735.

[27] W. Pedrycz, The design of cognitive maps: a study in synergy of granular computing and evolutionary optimization, Expert Syst. Appl. 37 (2010) 7288–7294.

[28] M.Ö. Efe, O. Kaynak, A novel optimization procedure for training of fuzzy inference systems by combining variable structure systems technique and Levenberg–Marquardt algorithm, Fuzzy Sets Syst. 122 (2001) 153–165.

[29] M.H. Nguyen, H.A. Abbass, R.I. Mckay, A novel mixture of experts model based on cooperative coevolution, Neurocomputing 70 (2006) 155–163.

[30] A.R. Solis, G. Panoutsos, Granular computing neural-fuzzy modelling: a neutrosophic approach, Appl. Soft Comput. 13 (2013) 4010–4021.

[31] F. Valdez, P. Melin, O. Castillo, An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, Appl. Soft Comput. 11 (2011) 2625–2632.

[32] H. Yi, Q. Duan, T.W. Liao, Three improved hybrid metaheuristic algorithms for engineering design optimization, Appl. Soft Comput. 13 (2013) 2433–2444.

[33] C. Zhang, J. Chen, B. Xin, Distributed memetic differential evolution with the synergy of Lamarckian and Baldwinian learning, Appl. Soft Comput. 13 (2013) 2947–2959.

[34] M.Z. Ali, K. Alkhatib, Y. Tashtoush, Cultural algorithms: emerging social structures for the solution of complex optimization problems, Int. J. Artif. Intell. 10 (2013) 20–42.

[35] P. Melin, O. Castillo, An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory, Inf. Sci. 177 (2007) 1543–1557.

[36] M. Ko, A. Tiwari, J. Mehnen, A review of soft computing applications in supply chain management, Appl. Soft Comput. 10 (2010) 661–674.

[37] N.M. Nasab, R. Shafaei, M. Rabiee, M. Mazinani, Minimization of maximum tardiness in a no-wait two stage flexible flow shop, Int. J Artif. Intell. 8 (2012) 166–181.

[38] M. Almaraashi, R. John, S. Coupland, A. Hopgood, Time series forecasting using a TSK fuzzy system tuned with simulated annealing, in: Proc. 2010 IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 2010, 6 pp.

[39] M. Almaraashi, R. John, S. Coupland, Designing generalised type-2 fuzzy logic systems using interval type-2 fuzzy logic systems and simulated annealing, in: Proc. 2012 IEEE International Conference on Fuzzy Systems, Brisbane, QLD, Australia, 2012, 6 pp.

[40] G. Liu, W. Yang, Learning and tuning of fuzzy membership functions by simulated annealing algorithm, in: Proc. 2000 IEEE Asia-Pacific Conference on Circuits and Systems, Tianjin, China, 2000, pp. 367–370.

[41] R. Jain, N. Sivakumaran, T.K. Radhakrishnan, Design of self tuning fuzzy controllers for nonlinear systems, Expert Syst. Appl. 38 (2011) 4466–4476.

[42] T.A. Yanara, Z. Akyürek, Fuzzy model tuning using simulated annealing, Expert Syst. Appl. 38 (2011) 8159–8169.

[43] 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 Trans. Ind. Electron. 59 (2012) 3049–3061.

[44] 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. 38 (2013) 74–84.

[45] 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.

[46] 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 Syst. Appl. 41 (2014) 1168–1175.

[47] H. Wang, L. Zhao, W. Du, F. Qian, A hybrid method for identifying T-S fuzzy models, in: Proc. 2011 8th International Conference on Fuzzy Systems and Knowledge Discovery, Shanghai, China, vol. 1, 2011, pp. 11–15.

[48] I. Pan, S. Das, A. Gupta, Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay, ISA Trans. 50 (2011) 28–36.

[49] D.T. Ho, J.M. Garibaldi, An improved optimisation framework for fuzzy time-series prediction, in: Proc. 2013 IEEE International Conference on Fuzzy Systems, Hyderabad, India, 2013, 8 pp.

[50] 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, Inf. Sci. 80 (2010) 2777–2792.

[51] S. Das, I. Pan, S. Das, Performance comparison of optimal fractional order hybrid fuzzy PID controllers for handling oscillatory fractional order processes with dead time, ISA Trans. 52 (2013) 550–566.

[52] A. Chatterjee, S.P. Ghoshal, V. Mukherjee, Chaotic ant swarm optimization for fuzzy-based tuning of power system stabilizer, Int. J. Electr. Power Energy Syst. 33 (2011) 657–672.

[53] B. Shaw, A. Banerjee, S.P. Ghoshal, V. Mukherjee, Comparative seeker and bio-inspired fuzzy logic controllers for power system stabilizers, Int. J. Electr. Power Energy Syst. 33 (2011) 1728–1738.

[54] J. Vaščák, M. Pal'a, Adaptation of fuzzy cognitive maps for navigation purposes by migration algorithms, Int. J. Artif. Intell. 8 (2012) 20–37.

[55] J. Vaščák, Adaptation of fuzzy cognitive maps by migration algorithms, Kybernetes 41 (2012) 429–443.

[56] D. Chen, J. Wang, F. Zou, H. Zhang, W. Hou, Linguistic fuzzy model identification based on PSO with different length of particles, Appl. Soft Comput. 12 (2012) 3390–3400.

[57] O. Linda, M. Manic, Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot, IEEE Trans. Ind. Inform. 7 (2011) 661–670.

[58] O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Appl. Soft Comput. 12 (2012) 1267–1278.

[59] W. Pedrycz, M. Song, A genetic reduction of feature space in the design of fuzzy models, Appl. Soft Comput. 12 (2012) 2801–2816.
[60] P. Angelov, Evolving Rule based Models: A Tool for Design of Flexible Adaptive Systems, Springer-Verlag, Berlin, Heidelberg, 2002.
[61] E. Lughofer, P. Angelov, Handling drifts and shifts in on-line data streams with evolving fuzzy systems, Appl. Soft Comput. 11 (2011) 2057–2068.
[62] E. Lughofer, Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications, Springer-Verlag, Berlin, Heidelberg, 2011.
[63] L. Maciel, A. Lemos, F. Gomide, R. Ballini, Evolving fuzzy systems for pricing fixed income options, Evol. Fuzzy Syst. 3 (2012) 5–18.
[64] S. Ahmed, N. Shakev, A. Topalov, K. Shiev, O. Kaynak, Sliding mode incremental learning algorithm for interval type-2 Takagi-Sugeno-Kang fuzzy neural networks, Evol. Fuzzy Syst. 3 (2012) 179–188.
[65] J.-S.R. Jang, Input selection for ANFIS learning, in: Proc. 5th IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, vol. 2, 1996, pp. 1493–1499.
[66] D.A. Linkens, M.-Y. Chen, Input selection and partition validation for fuzzy modelling using neural network, Fuzzy Sets Syst. 107 (1999) 299–308.
[67] 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 Trans. Educ. 54 (2011) 345–355.
[68] The Laboratory Anti-lock Braking System Controlled from PC – User's Manual, INTECO Ltd., Krakow, 2007.
[69] M.-B. Radac, R.-E. Precup, S. Preitl, J.K. Tar, K.J. Burnham, Tire slip fuzzy control of a laboratory anti-lock braking system, in: Proc. European Control Conference 2009, Budapest, Hungary, 2009, pp. 940–945.
[70] S. Galichet, L. Foulloy, Fuzzy controllers: synthesis and equivalences, IEEE Trans. Fuzzy Syst. 3 (1995) 140–148.
[71] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 721–741.
[72] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi Jr., Optimization by simulated annealing, Science 220 (1983) 671–680.
[73] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
[74] J. Kennedy, R.C. Eberhart, A new optimizer using particle swarm theory, in: Proc. 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.
[75] H. Ohtake, K. Tanaka, H.O. Wang, Fuzzy modeling via sector nonlinearity concept, in: Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, BC, Canada, vol. 1, 2001, pp. 127–132.
[76] R.-E. Precup, S. Preitl, PI and PID controllers tuning for integral-type servo systems to ensure robust stability and controller robustness, Electr. Eng. 88 (2006) 149–156.
[77] H. Zhang, W. Pedrycz, D. Miao, Z. Wei, From principal curves to granular principal curves, IEEE Trans. Cybern. 44 (2014) 748–760.