

Neural Fuzzy Inference Network Based on Differential Evolution for Nonlinear System Control

Hsueh-Yi Lin¹ and Cheng-Jian Lin^{1,2*}

¹Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan

²College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan

(Received April 28, 2021; accepted November 12, 2021)

Keywords: temperature-sensing control, differential evolution, symbiotic evolution, neural fuzzy networks, Taguchi method

We propose a neural fuzzy inference network (NFIN) based on a symbiotic Taguchi-based modified differential evolution (STMDE) algorithm for solving nonlinear control problems. The proposed STMDE algorithm not only uses the Taguchi method in its search for the best individual but also employs adjustable parameter control to tune the scaling factor, which can prevent a solution from being trapped at local optima and reinforce the search ability. Moreover, symbiotic evolution (SE) is applied to improve the structure of individual compositions. Unlike the traditional differential evolution (DE) algorithm, SE regards each individual in a population as being the partial solution to a problem instead of the full solution. Compared with traditional DE, the proposed STMDE algorithm reduces the error by 7.95, 4.51, 5.22, and 51.34% in terms of regulation performance, noise rejection ability, robustness to parameter variation of the controlled system, and controller tracking capability, respectively. In addition, our experimental results also indicate that the proposed STMDE algorithm exhibits superior performance to other algorithms used for solving nonlinear temperature-sensing control problems.

1. Introduction

Recently, neural fuzzy inference networks (NFINs) have been commonly used in different fields, such as mobile robotic control,⁽¹⁾ image processing,⁽²⁾ and fabric pilling classification.⁽³⁾ The advantage of an NFIN is that it incorporates the reasoning characteristic of a fuzzy inference system with learning ability. NFINs can be divided into two types: Mamdani-type NFINs (M-NFINs)^(4,5) and Takagi–Sugeno–Kang (TSK)-type NFINs (T-NFINs).^(6–8) The fuzzy reasoning in an M-NFIN involves the minimum fuzzy implication rule. However, in a T-NFIN, the consequence of each fuzzy rule is a linear function including a combination of input variables. Many studies^(7,8) have shown that a T-NFIN achieves superior performance to an M-NFIN in terms of accuracy and network size. In general, the function involving a combination of input variables in a T-NFIN is linear. The traditional T-NFIN does not fully capitalize on the

*Corresponding author: e-mail: cjlin@ncut.edu.tw
<https://doi.org/10.18494/SAM3549>

mapping capabilities in some nonlinear problems. Thus, in this study, we adopted an available network structure and embedded a functional link neural network (FLNN)^(9,10) into the consequent part of each fuzzy rule, yielding a model called a functional-based NFIN (F-NFIN). In the F-NFIN, the consequent part of each fuzzy rule comprises the input functional expansion.

Learning algorithms have been used for tuning the network parameters of NFINs. The backpropagation (BP) learning algorithm^(7, 8) is widely used for tuning the parameters of NFINs. However, the BP algorithm is based on gradient descent and is easily trapped at local minima. An evolutionary algorithm has recently been designed for optimizing the parameters of NFINs. Yolmeh *et al.*⁽¹¹⁾ predicted the effect of annatto dye on *Salmonella enteritidis* in mayonnaise using a genetic algorithm (GA)-based neural network and an adaptive neuro-fuzzy inference system (ANFIS). Kuo *et al.*⁽¹²⁾ tuned membership functions and weights in NFINs by using immune algorithms. In addition, Shihabudheen *et al.*⁽¹³⁾ proposed an ANFIS based on particle swarm optimization to reduce computational complexity and improve generalization. Hou *et al.*⁽¹⁴⁾ combined the differential evolution (DE) algorithm with the BP algorithm to optimize an NFIN for forecasting network traffic. The aforementioned evolutionary methods are highly efficient in their search of the global space; however, problems pertaining to local minima and premature convergence persist. Therefore, in this study, we focused on improving the traditional DE algorithm to ensure that it can be used to obtain optimal network parameters.

In this study, we present an NFIN with a symbiotic Taguchi-based modified differential evolution (STMDE) algorithm. First, the proposed STMDE algorithm adopts an effective method to find the best individual and to adjust the scaling factor in the DE algorithm. In addition, the Taguchi method,⁽¹⁵⁾ which was designed for conducting simple value analyses through orthogonal arrays, is used to achieve a better evolutionary direction. The power of the Taguchi method originates from its use of fewer experimental sets to obtain more crucial information. Moreover, the STMDE algorithm uses symbiotic evolution (SE) to reinforce the traditional DE structure. SE is different from the traditional DE algorithm in that it assumes that each individual in the group only represents a part of the solution, rather than a complete solution to the problem.

2. Structure of NFIN

As discussed in this section, the proposed NFIN model uses an FLNN to form the consequent part of the fuzzy rules. Figure 1 shows the structure of the proposed NFIN model. The j th fuzzy rule of the NFIN is represented as follows:

The j th fuzzy rule: IF x_1 is A_{1j} and x_2 is A_{2j} , ..., and x_i is A_{ij} , ..., and x_N is A_{Nj}

$$\text{THEN } y_j = \sum_{k=1}^B w_{kj} \varphi_k, \quad (1)$$

where x_i represents the input variables, A_{ij} represents the membership functions, φ_k represents the basis trigonometric functions, w_{kj} represents the local output weights, N is the number of inputs, and B is the number of basis functions.

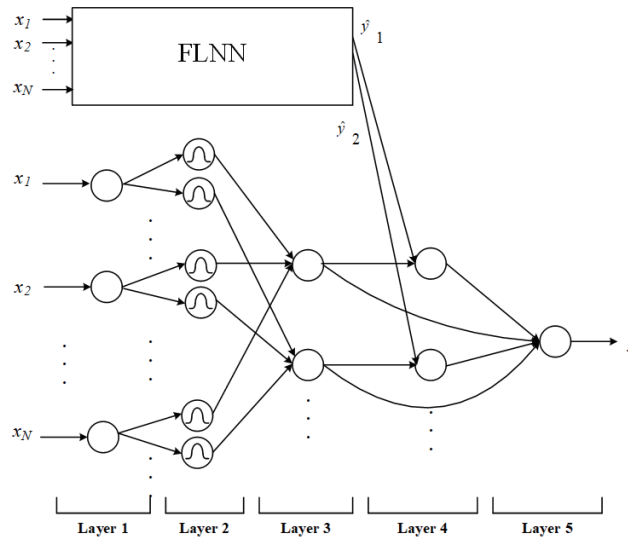


Fig. 1. Structure of proposed NFNN model.

The operation functions in each layer are as follows.

Layer 1: This layer has no calculations, and the input variable directly transmits its value to the next layer:

$$o_i^{(1)} = x_i . \tag{2}$$

Here, $o_i^{(1)}$ is the output of a node in the first layer.

Layer 2: The membership degree is calculated as follows:

$$o_{ij}^{(2)} = \exp \left(- \frac{[o_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2} \right) , \tag{3}$$

where m_{ij} and σ_{ij} represent the mean and variance of the membership function, respectively.

Layer 3: The node represents the premise part of a fuzzy rule and uses the product operator to compute the premise matching degree of the fuzzy rules. The output of each node is described as follows:

$$o_j^{(3)} = \prod_i o_{ij}^{(2)} . \tag{4}$$

Here, $\prod_i o_{ij}^{(2)}$ represents the excitation of the corresponding fuzzy rule.

Layer 4: Nodes in this layer are the consequent nodes. The inputs of the consequent nodes are the outputs from layer 3 with the nonlinear combinations of inputs from an FLNN added.

$$o_j^{(4)} = o_j^{(3)} \cdot \sum_{k=1}^B w_{kj} \phi_k \tag{5}$$

Here, w_{kj} is the weighting value of the FLNN and φ_k represents the functional expansion.⁽⁹⁾

Layer 5: Each node matches a single output in this layer to perform a defuzzification operation.

$$y = o^{(5)} = \frac{\sum_{j=1}^F o_j^{(3)}}{\sum_{j=1}^F o_j^{(3)}} = \frac{\sum_{j=1}^F o_j^{(3)} (\sum_{k=1}^B w_{kj} \varphi_k)}{\sum_{j=1}^F o_j^{(3)}} = \frac{\sum_{j=1}^F o_j^{(3)} \hat{y}_j}{\sum_{j=1}^F o_j^{(3)}} \tag{6}$$

Here, y denotes the output of the NFIN model and F is the number of fuzzy rules.

3. Proposed STMDE

SE assumes that each individual in the group only represents a part of the solution, rather than a complete solution to the problem, and it combines several individuals to solve complex solutions. That is, the partial solution of each individual is combined with the other solutions in the population to generate an effective overall solution. Figure 2 illustrates the structure of the SE.

The STMDE algorithm consists of four parts: initialization, fitness evaluation, parameter learning, and a solution-aging mechanism. During the coding step, the membership functions are processed by the fuzzy rules of an NFIN for individuals. Before the learning process begins, the sub-individual values are assigned for the initialization step. The fitness evaluation step assigns a suitable value to each fuzzy system, which is a combination of several fuzzy rules that are selected from each subpopulation. Then, the best individual is updated and recoded on the basis of the fitness value. The parameter-learning step executes STMDE for all subpopulations.

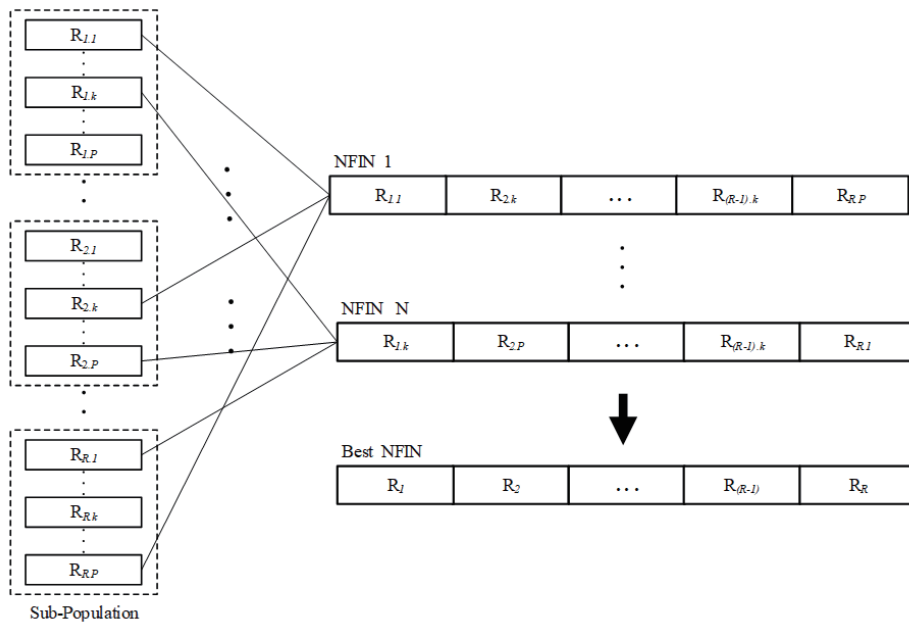


Fig. 2. Structure of SE.

This process is repeated until a given stopping condition is satisfied. A flow chart of the proposed STMDE algorithm is presented in Fig. 3, which depicts the entire learning process.

3.1 Initialization phase

The initialization phase is divided into two steps: coding and the creation of an initial population. The coding step involves the fuzzy rules representing suitable sub-individuals for SE, whereas the population values are assigned before the evolution process begins.

A. Coding step

The first step of STMDE is the coding of each individual (rule) into an NFIN. Figure 4 presents the parameter coding of the NFIN.

B. Creating initial population

Before using the STMDE algorithm, every individual $x_{i,g}$ is randomly generated in the interval $[0, 1]$, where $i = 1, 2, \dots, PS$.

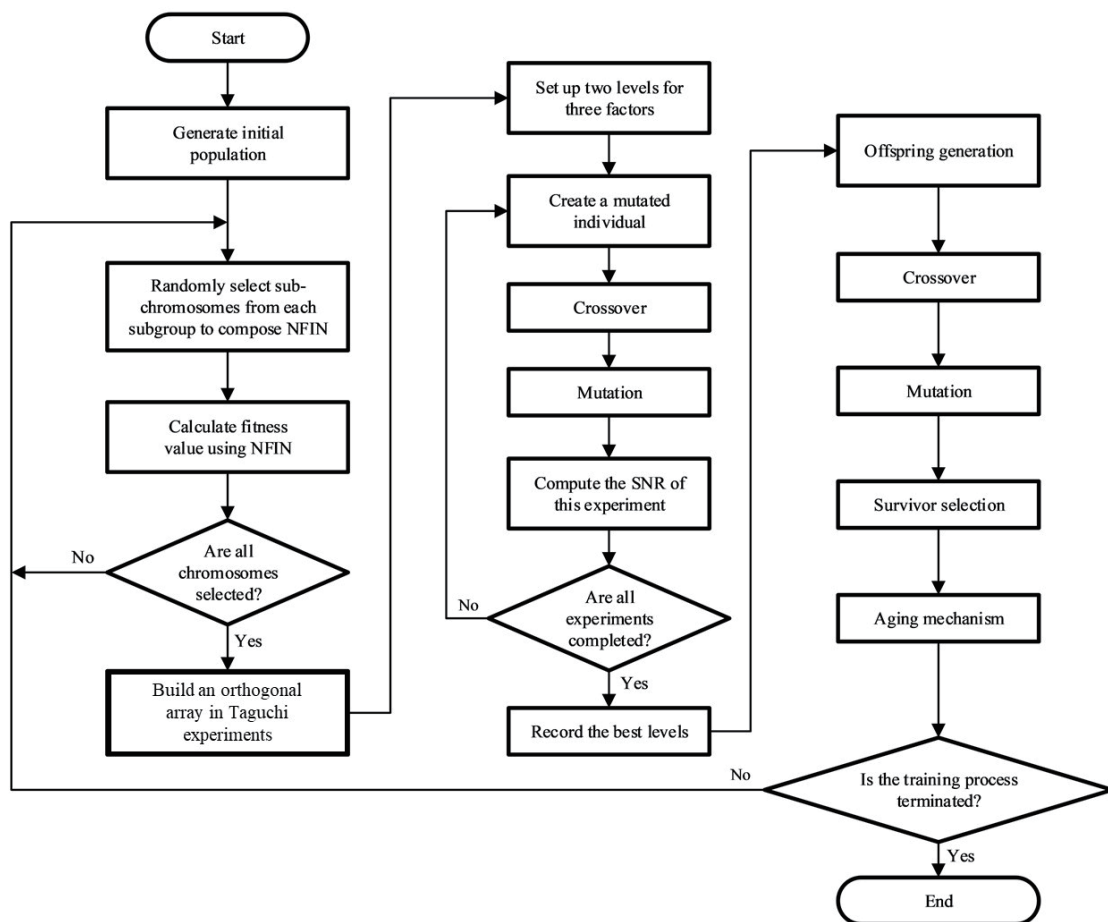


Fig. 3. Flow chart of proposed STMDE algorithm.

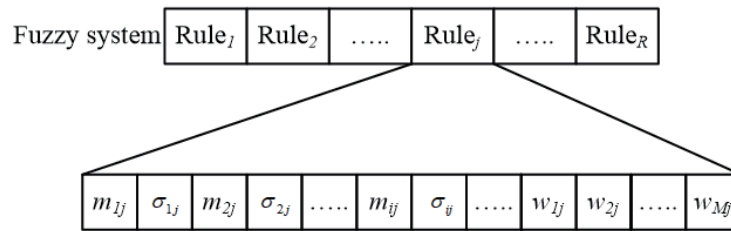


Fig. 4. Parameter coding of NFIN model.

3.2 Evaluation phase

The fitness value (FV) of an NFIN is calculated using the suitable combinations of all selected rules.

Step 1: Randomly select *F* fuzzy rules (individuals) from each subgroup with size *PS* to form an NFIN.

Step 2: Through Step 1, evaluate every NFIN to obtain an FV. In this study, the FV is described as follows:

$$FV = \frac{1}{1 + \sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} (y_k - y_k^d)^2}} \quad (7)$$

Here, y_k is the network output of the *k*th data item, y_k^d is the target output of the *k*th data item, and N_t denotes the number of training data.

Step 3: Compare the FV of each current NFIN with the FV of the best NFIN. If the FV of the current NFIN is higher than that of the best NFIN, then replace the best NFIN with the current NFIN.

Step 4: Repeat Steps 1 to 3 until each rule of the NFIN has been selected a sufficient number of times.

3.3 Parameter-learning phase

In the parameter-learning phase, the proposed STMDE algorithm conducts parameter learning to update all individuals. Then, the composed NFIN is evaluated to reveal each subpopulation, which allows each individual (rule) to develop itself. Figure 5 shows the individual structure in STMDE.

The STMDE algorithm is divided into two major phases: parent choice and reproduction phases.

3.3.1 Parent choice phase

To achieve a better evolutionary direction, the Taguchi method is commonly used to find the best parent combination, which is one of the popular optimization methods in robotic design

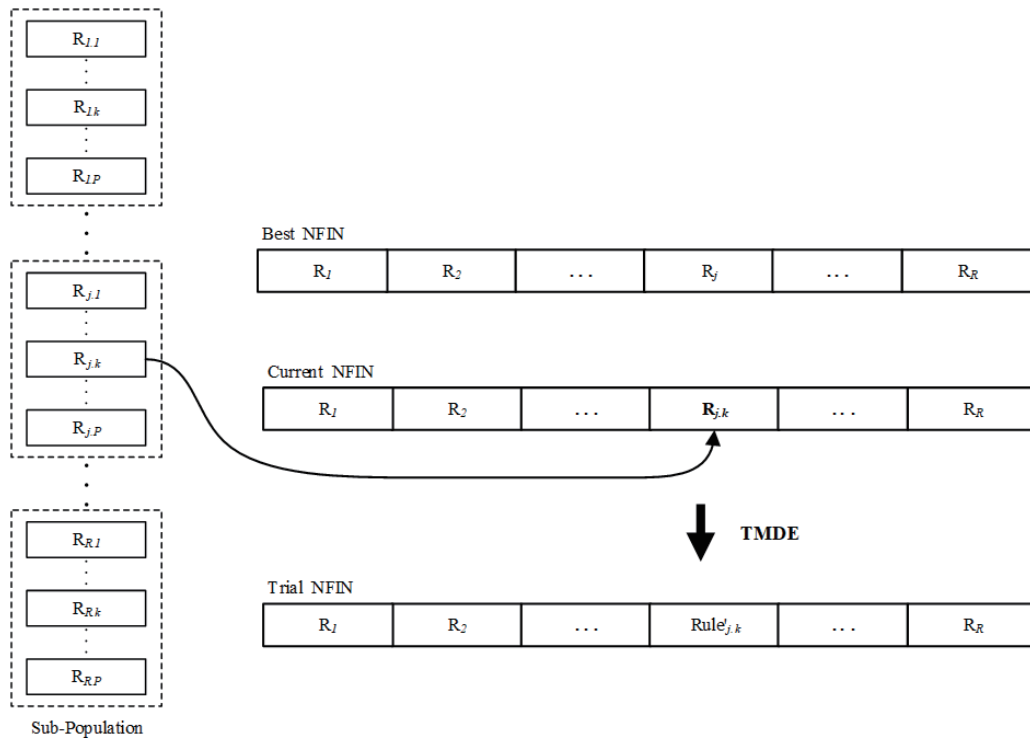


Fig. 5. Individual structure in STMDE.

processes. In an evolutionary algorithm, the Taguchi method can effectively lead the individual to evolve in a better direction. In our proposed algorithm, this experimental method is used to obtain an optimal parent combination in the learning algorithm. We describe how to obtain the best parent combination using the Taguchi method. The orthogonal array (OA) in the Taguchi method provides many reduced variances for the experiment with the optimal control parameter setting. An OA provides a set of well-balanced (minimum number of) experiments and helps in analyzing the factor efficiency and predicting the optimal parameter combination. A two-level OA is used in the proposed algorithm. The general symbol for a standard two-level OA is

$$L_p(q^s), \tag{8}$$

where p is the number of experiments, q is the level of each factor, and s is the number of affecting factors.

A. Experimental factor determination and OA creation

For the i th individual in the g th generation $x_{i,g}$, where $i = 1, 2, \dots, PS$, three individuals ($x_{r_1,g}$, $x_{r_2,g}$, and $x_{r_3,g}$) are randomly selected from the population, where r_1, r_2 , and $r_3 \in \{1, 2, \dots, PS\}$ and $i \neq r_1 \neq r_2 \neq r_3$. Therefore, a mutated individual $u_{i,g}$ is generated as follows:

$$u_{i,g} = x_{r_1,g} + C \cdot (x_{r_2,g} - x_{r_3,g}), \tag{9}$$

where C is a scaling factor. To optimize the search direction and capability, a new method for mutating individuals is proposed and a mutated individual is redefined as follows:

$$u_{i,g} = x_{r_1,g} + (1 - C) \cdot (x_{r_2,g} - x_{r_3,g}) + C \cdot (x_{best,g} - x_{r_1,g}), \quad (10)$$

where C is the control rate of the evolved population and is equal to g/G , G is the number of generations, g is the current generation, and $x_{best,g}$ is the best individual of the g th generation. Because the three parameters (r_1 , r_2 , and r_3) of the evolutionary direction are closely related, these parameters are adopted as the affecting factors in the Taguchi method. Table 1 presents an $L_4(2^3)$ two-level OA in the proposed method. In the OA, the column “Experiment #” lists the experiments, where there are four experiments in this case. In the $L_4(2^3)$ OA, an experiment determines the influence of three different independent factors (r_1 , r_2 , and r_3) with each factor having two levels. The column “Performance value” lists the values of ε used to evaluate each experimental result in terms of the signal-to-noise ratio (SNR).

The SNR is widely used as a quality evaluation metric in communication engineering. Table 2 presents the homologous table of the factors and their corresponding levels. Here, r_{f_i} denotes the parent index, which is between 1 and PS .

In the proposed algorithm, the tournament method is used to select the best individual. In this process, three individuals are randomly selected and the one with the best FV is selected. Then, this process is repeated three times until six different individuals from the population are obtained.

Next, the experiment is initiated in accordance with the created OA. Each individual requires four separate experiments, and the offspring individuals v_i for each individual are generated by executing some evolutionary steps (mutated individual creation, crossover mutation). In the following paragraph, we discuss the experimental analysis.

B. Experimental analysis

Although the SNR in the Taguchi method is used for quality evaluation, it involves three types of transformations: larger-the-better, nominal-is-best, and smaller-the-better. In this study, larger-the-better is used and calculated with the following equation:

Table 1
Two-level OA.

Experiment #	Factors			Performance value (SNR)
	r_1	r_2	r_3	
No. 1	1	1	1	ε_1
No. 2	1	2	2	ε_2
No. 3	2	1	2	ε_3
No. 4	2	2	1	ε_4

Table 2
Homologous table of factors and their corresponding levels.

Level #	Factors		
	r_1	r_2	r_3
Level 1	$r_{1,1}$	$r_{2,1}$	$r_{3,1}$
Level 2	$r_{1,2}$	$r_{2,2}$	$r_{3,2}$

$$\varepsilon = -10 \cdot \log_{10} \left(\frac{1}{N_t} \sum_{k=1}^{N_t} \frac{1}{\left(\frac{1}{1 + (y_k - y_k^d)^2} \right)} \right). \quad (11)$$

Here, y_k denotes the network output of the k th data item, y_k^d denotes the target output of the k th data item, and N_t denotes the number of training data. The affecting factors are described as follows:

$$AF_{f,l} = \sum \varepsilon_i, \quad \forall \text{level } l \text{ at factor } f, \quad (12)$$

where i is the number of experiments, f is the factor name, and l is the number of levels. Therefore, on the basis of the effects of the various factors, the level of the best performance of each factor can be determined. Table 3 presents the performance results obtained under the assumption that all experiments have been completed. The best level (BL) in this generation is described as follows:

$$BL_f = \begin{cases} 1, & \text{if } AF_{f,1} \geq AF_{f,2} \\ 2, & \text{otherwise} \end{cases}. \quad (13)$$

For example, Table 3 shows that the value of $AF_{1,2}$ is higher than that of $AF_{1,1}$ and that the BL of factor 1 is equal to level 2. Therefore, the best combination of the parents ($r_{1,2}$, $r_{2,1}$, and $r_{3,1}$) can be obtained.

3.3.2 Reproduction phase

A. Offspring generation

According to Eq. (8), the TMDE algorithm adopts a differential method to produce a mutated individual $u_{i,g}$. To supplement the differential search operation, a discrete recombination operation called a crossover operation is used. The mutated individual $u_{i,g}$ in the crossover

Table 3
Two-level OA.

Experiment #	Factors			Performance value (SNR)
	r_1	r_2	r_3	
No. 1	1	1	1	-1.93
No. 2	1	2	2	-3.06
No. 3	2	1	2	-1.02
No. 4	2	2	1	-1.59
$AF_{f,1}$	-4.99	-2.95	-3.52	
$AF_{f,2}$	-2.61	-3.65	-4.08	
Best level (BL)	2 ($r_{1,2}$)	1 ($r_{2,1}$)	1 ($r_{3,1}$)	

operation mates with $x_{i,g}$ to generate the offspring $v_{i,g}$. The individual $v_{i,g}$ inherits from $x_{i,g}$ and $u_{i,g}$ and is determined using a crossover probability (CR).

$$v_{id,g} = \begin{cases} u_{id,g}, & \text{if } \text{Rand}(d) \leq CR \\ x_{id,g}, & \text{if } \text{Rand}(d) > CR \end{cases} \quad (14)$$

Here, $\text{Rand}(d) \in [0, 1]$ represents the d th evaluation by using a random number generator, the CR value is between 0 and 1, and $d = 1, 2, \dots, D$ represents the d th element of individual vectors. Figure 6 illustrates the crossover mechanism with eight-dimensional vectors.

B. Mutation

To avoid the proposed STMDE algorithm becoming trapped at local optima, a mutation scheme is adopted to maintain the population diversity and increase the search ability. We adopt a one-point operation in the proposed algorithm. If the element x_d in $x = (x_1, x_2, \dots, x_d, \dots, x_D)$ is randomly selected, the resulting offspring is $x = (x_1, x_2, \dots, x'_d, \dots, x_D)$. Therefore, the newly generated element x'_d is randomly set between 0 and 1.

C. Survivor selection

The STMDE algorithm only applies the selection process to select survivors. Therefore, the FV of the current composition of the NFIN, the FV of the trial composition of the NFIN, and the FV of the best NFIN are compared. If the FV of the current composition of the NFIN is greater than that of the best NFIN, the current composition of the NFIN replaces the best NFIN. If the FV of the trial composition of the NFIN is greater than that of the best NFIN, the trial composition of the NFIN replaces the best NFIN.

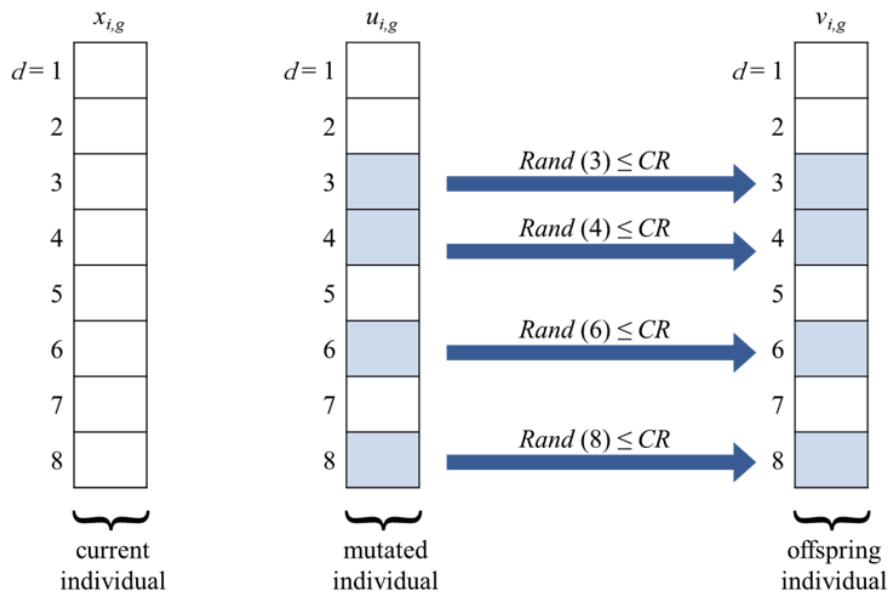


Fig. 6. (Color online) Crossover operation for $D = 8$ parameters.

3.4 Aging mechanism

All individuals are evolved using the best NFIN, and the subpopulations of the network may be similar to each other. To improve the proposed method, we propose an aging mechanism to overcome the problem of trapping at local optima. Here, a counter value (CV) is used to record the number of times the FV of the individual obtained in the current generation is lower than that in the previous generation. If the CV is larger than a threshold value ε , a neighborhood value is generated to maintain solution variation. This situation can prevent the occurrence of premature convergence. A new individual is generated in a suitable space and is in the neighborhood of the previous individual as follows.

$$x_{i,g+1} + 1 = x_{i,g} + \alpha \quad (15)$$

Here, α represents a control parameter used to determine the distance between the previous and new individuals. The value of α is between 0.001 and -0.001 . If a new individual is generated, the CV is set to zero. When all individuals are updated, the generation should be inspected to determine whether it should be continued. If the generation does not achieve the terminal value, then the execution process returns to the fitness evaluation step.

4. Water Bath Temperature Control

To determine the performance of the proposed STMDE algorithm, we conduct a control experiment. As the initial parameters considered in this experiment, the number of generations is 1000, the number of fuzzy systems (N) is 100, CR is 0.9, the mutation rate is 0.3, and ε is 3. Each problem is simulated on an Intel Pentium Core i7 3.00-4.70 GHz desktop computer and the program is developed using Visual C++ 6.0.

The target of this example is water bath temperature control according to the following expression:

$$\frac{dy(t)}{dt} = \frac{u(t)}{U} + \frac{T_0 - y(t)}{HU} \quad (16)$$

Here, $y(t)$ denotes the output temperature and is obtained using a temperature sensor, $u(t)$ represents the heat flowing into the water bath system, and H and U are constant values. The discrete-time equation of the water bath system is

$$y(k+1) = e^{-aS} y(k) + \frac{b(1 - e^{-aS})}{1 + e^{0.5y(k)-40}} u(k) + [1 - e^{-aS}] T_0, \quad (17)$$

where a and b are constant values ($a = 1.0015e^{-4}$, $b = 8.67973e^{-3}$), T_0 is room temperature (25.0 °C), S is the sampling period and set as 30, and $u(k)$ is set between 0 and 5 V.

Figure 7 shows the proposed training method of the control system. This method is divided into the learning and control parts. In the learning part, we connect switches S1 and S2 to points 1 and 2, respectively. In the learning process, $y(k) = [y_p(k+1) \ y_p(k)]$ is used as the input of the training data and $u(k)$ is used as the desired output of the training data. The general inverse modeling⁽¹⁶⁾ training method is adopted for the NFIN controller. In the control part, we connect switches S1 and S2 to points 3 and 4, respectively.

The input $u(k)$ is entered into the water bath system in accordance with Eq. (17) by the training method. Moreover, we select 120 training data on the basis of the input–output features to cater to the reference output. The water temperature gradually increases from $T_0 = 25.0$ °C after the random input signal is injected.

In this study, we compare the proposed STMDE and TMDE algorithms with the DE,⁽¹⁴⁾ immune algorithm (IA),⁽¹²⁾ and GA.⁽¹¹⁾ The following performance measures are used in this study: regulation performance, noise rejection ability, parameter variation of the controlled system, and controller tracking capability.

In the first simulation, the water bath system is controlled to track three temperatures.

$$y_{ref}(k) = \begin{cases} 35 \text{ }^\circ\text{C}, & \text{for } k \leq 40 \\ 55 \text{ }^\circ\text{C}, & \text{for } 40 < k \leq 80 \\ 75 \text{ }^\circ\text{C}, & \text{for } 80 < k \leq 120 \end{cases} \quad (18)$$

Figure 8(a) shows the regulation results of the STMDE learning algorithm. The regulation performances of the TMDE algorithm, DE, IA, and GA are also evaluated. Figure 8(b) shows the error results of the STMDE algorithm, TMDE algorithm, DE, IA, and GA, indicating that the STMDE algorithm exhibits the least error among the compared algorithms. The evaluation index, namely, the sum of the absolute errors (SAE), is used to further test the regulation performance. This index is defined as

$$SAE = \sum_k |y(k) - y_{ref}(k)|, \quad (19)$$

where $y(k)$ and $y_{ref}(k)$ are the model and reference temperatures of the water bath system, respectively. The SAE values of the STMDE algorithm, TMDE algorithm, DE, IA, and GA are 356.79, 361.36, 387.61, 470.84, and 452.67, respectively (Table 4, second row). The proposed STMDE algorithm exhibits a much better SAE value than the TMDE algorithm, DE, IA, and GA.

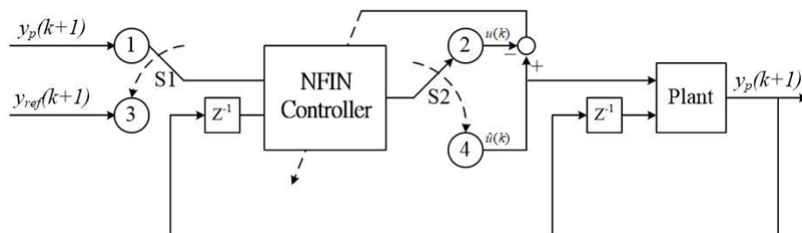


Fig. 7. Training method of the water bath system.

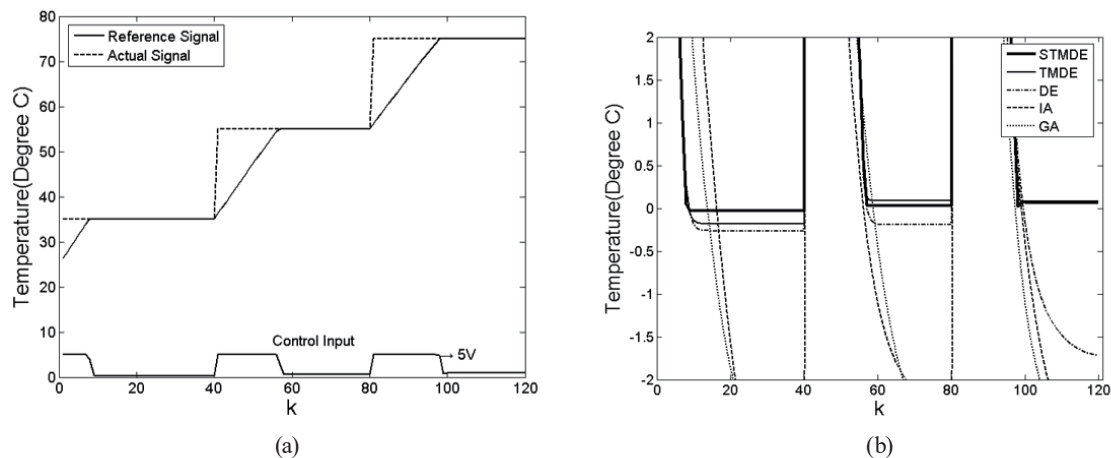


Fig. 8. (a) Regulation results of NFIN controller with the proposed STMDE learning algorithm in a water bath system. (b) Error results of STMDE algorithm, TMDE algorithm, DE, IA, and GA.

Table 4

Comparison of results of various existing algorithms.

$SAE = \sum_{k=1}^{120} y(k) - y_{ref}(k) $	STMDE	TMDE	DE ⁽¹⁴⁾	IA ⁽¹²⁾	GA ⁽¹¹⁾
Regulation performance	356.79	361.36	387.61	470.84	452.67
Influence of impulse noise	247.83	251.24	259.53	408.58	523.33
Effect of change in plant dynamics	234.86	238.27	247.82	424.63	565.59
Tracking performance	31.68	39.15	65.11	281.04	272.34

The second simulation is implemented to evaluate the anti-noise ability of the five algorithms under conditions that involve unknown impulse noise interfering with the execution processes. A noise with $-5\text{ }^{\circ}\text{C}$ is injected to the water bath system output at the 60th sample time. Consequently, a set point of $50\text{ }^{\circ}\text{C}$ is adopted in this simulation. Figures 9(a) and 9(b) show the control of the STMDE algorithm and the error results for all five algorithms under an impulse noise in the controller, respectively. The SAE values of the STMDE algorithm, TMDE algorithm, DE, IA, and GA are 247.83, 251.24, 259.53, 408.58, and 523.33, respectively (Table 4, third row). The STMDE algorithm exhibits superior performance to the other algorithms and recovers rapidly after the impulse noise.

In many industrial control processes, plant parameters usually vary unpredictably. In this study, a $0.6 \times u(k-2)$ signal is added to the water bath system input after the 60th sample time in the third simulation to evaluate the robustness of the algorithms against disturbance. The $50\text{ }^{\circ}\text{C}$ set point is also adopted. Figure 10(a) shows the robustness of the STMDE algorithm against disturbance. Figure 10(b) shows the corresponding errors of all five algorithms. The SAE values of the STMDE algorithm, TMDE algorithm, DE, IA, and GA are 356.79, 361.36, 387.61, 470.84, and 452.67, respectively (Table 4, fourth row). The results demonstrate the robustness against disturbance of the STMDE learning algorithm.

In the final simulation, ramp-reference signals are used to evaluate the tracking capability of the STMDE algorithm. The following values are defined:

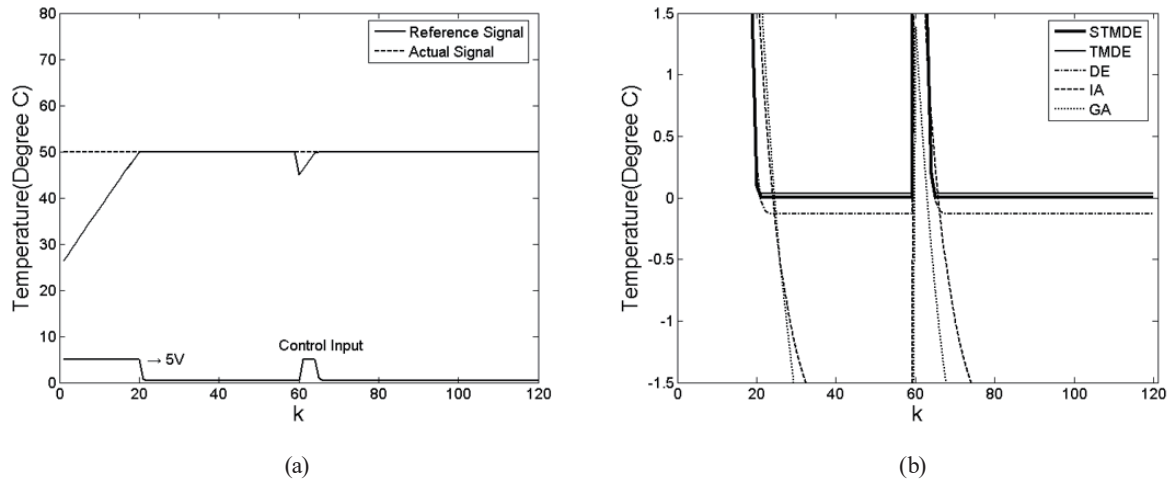


Fig. 9. (a) Results obtained for NFIN controller with STMDE learning algorithm after the impulse noise. (b) Error results of STMDE algorithm, TMDE algorithm, DE, IA, and GA.

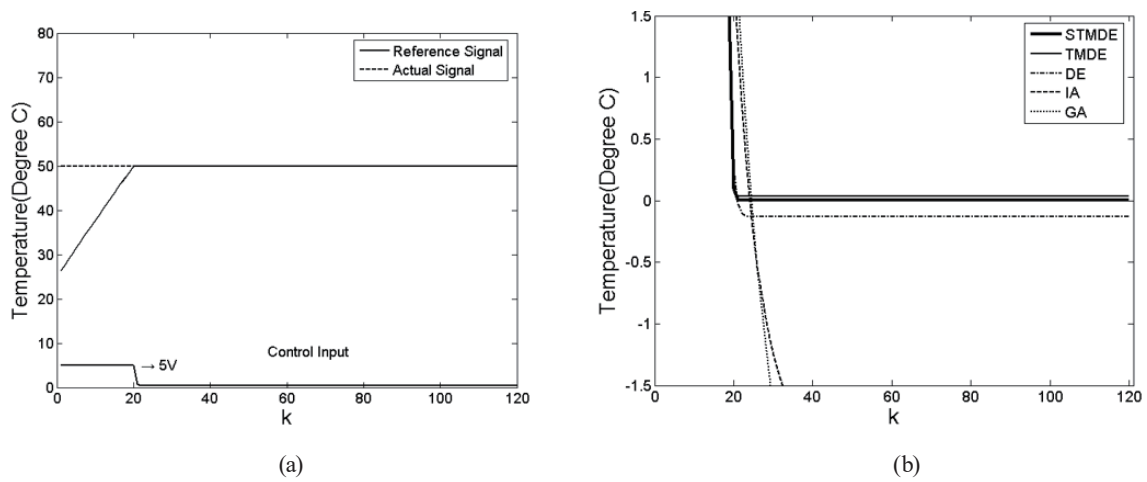


Fig. 10. (a) Results of NFIN controller with STMDE learning algorithm under a disturbance rejection. (b) Error results of STMDE algorithm, TMDE algorithm, DE, IA, and GA.

$$y_{ref}(k) = \begin{cases} 34^{\circ}\text{C}, & \text{for } k \leq 30 \\ (34 + 0.5 \times (k - 30))^{\circ}\text{C}, & \text{for } 30 < k \leq 50 \\ (44 + 0.8 \times (k - 50))^{\circ}\text{C}, & \text{for } 50 < k \leq 70 \\ (60 + 0.5 \times (k - 70))^{\circ}\text{C}, & \text{for } 70 < k \leq 90 \\ 70^{\circ}\text{C}, & \text{for } 90 < k \leq 120 \end{cases} \quad (20)$$

Figure 11(a) shows the tracking capability of the STMDE algorithm. In addition, Fig. 11(b) shows the tracking error results of all five algorithms. The SAE values of the STMDE algorithm, TMDE algorithm, DE, IA, and GA are 31.68, 39.15, 65.11, 281.04, and 272.34, respectively (Table 4, fifth row).

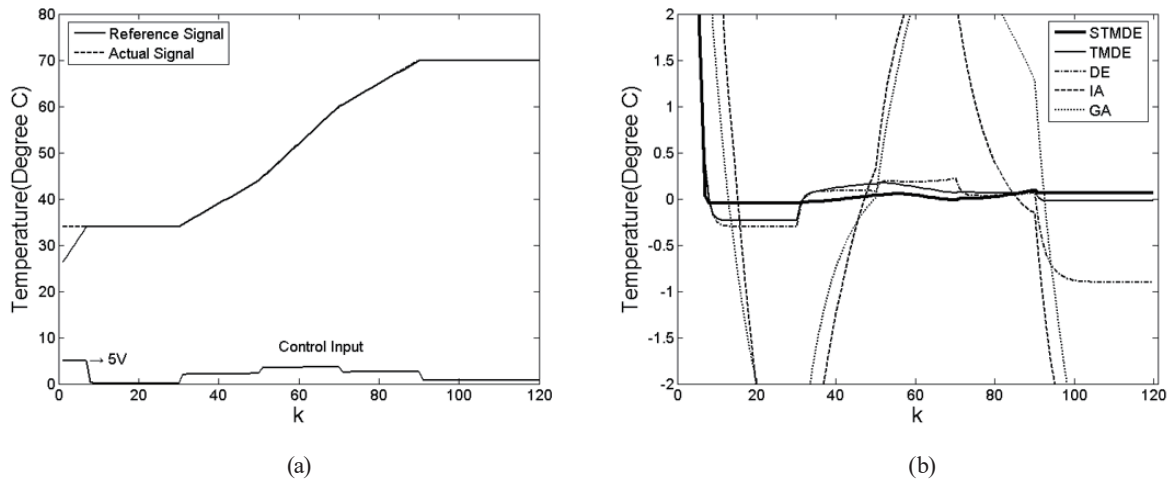


Fig. 11. (a) Tracking results of NFIN controller with the proposed STMDE learning algorithm. (b) Error results of STMDE algorithm, TMDE algorithm, DE, IA, and GA.

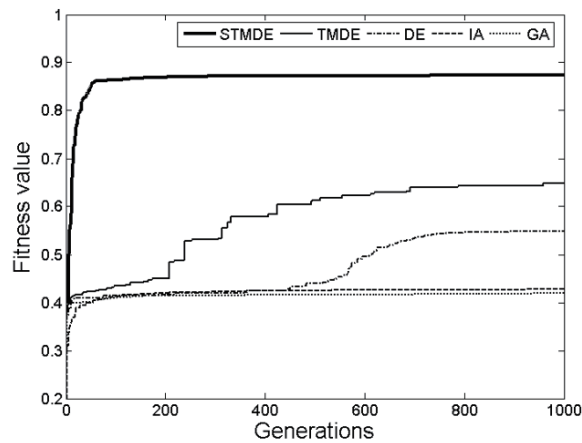


Fig. 12. Learning results of STMDE algorithm, TMDE algorithm, DE, IA, and GA.

The learning curves of the STMDE algorithm, TMDE algorithm, DE, IA, and GA are illustrated in Fig. 12, revealing the highest FV value of the proposed STMDE algorithm in the water bath system. The four performance measure results are presented in Table 4. In this table, the results show that the NFIN with the STMDE algorithm outperformed the other algorithms.

5. Conclusions

We proposed an NFIN based on an STMDE algorithm to solve nonlinear temperature control problems. In the NFIN, a nonlinear combination of inputs is adopted as the consequent part of a fuzzy rule and the STMDE algorithm is used to optimize the controller parameters. The proposed STMDE algorithm uses SE to find suitable rule combinations. Then, an aging

mechanism is applied to prevent the solution from falling into a local optimum. This STMDE algorithm improves the convergence speed and accuracy. To determine the performance of the proposed STMDE, we conducted a temperature-sensing control experiment, where the temperature of the water bath system was measured using a temperature sensor. The obtained SAE values of the proposed STMDE algorithm were 356.79, 247.83, 234.86, and 31.68 for the regulation, anti-noise ability, robustness to parameter variation of the controlled system, and tracking capability of the controller, respectively. These values were also lower than those of other algorithms. In future work, the NFIN based on the STMDE algorithm can be implemented in a system on a chip to achieve high-speed operations in real-time temperature control applications.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology of the Republic of China, Taiwan for financially supporting this research under Contract No. MOST 110-2221-E-67-031-MY2.

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- 1 T. C. Lin, C. C. Chen, and C. J. Lin: *J. Intell. Robot. Syst.* **92** (2018) 343. <https://doi.org/10.1007/s10846-017-0743-y>
- 2 H. Y. Lin, C. J. Lin, and M. L. Huang: *J. Inf. Sci. Eng.* **34** (2018) 701. [https://doi.org/10.6688/JISE.201805_34\(3\).0008](https://doi.org/10.6688/JISE.201805_34(3).0008)
- 3 M. Eldessouki and M. Hassan: *Expert Syst. Appl.* **42** (2015) 2098. <https://doi.org/10.1016/j.eswa.2014.10.013>
- 4 C. L. Lee, C. J. Lin, and H. Y. Lin: *Smart Sci.* **5** (2017) 160. <https://doi.org/10.1080/23080477.2017.1348061>
- 5 S. K. Halgamuge: *IEEE Trans Fuzzy Syst.* **6** (1998) 304. <https://doi.org/10.1109/91.669031>
- 6 T. Takagi and M. Sugeno: *IEEE Trans. Syst. Man Cybern. Syst.* **15** (1985) 116. <https://doi.org/10.1109/TSMC.1985.6313399>
- 7 J. W. Yeh and S. F. Su: *IEEE Trans Cybern.* **47** (2017) 2343. <https://doi.org/10.1109/TCYB.2016.2638861>
- 8 J. G. Wang, S. C. Tai, and C. J. Lin: *Int. J. Innov. Comput. Inf. Control.* **11** (2015) 707. <http://www.ijicic.org/ijicic-110222.pdf>
- 9 J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda: *IEEE Trans. Syst. Man Cybern.* **29** (1999) 254. <https://doi.org/10.1109/3477.752797>
- 10 E. Lotfi and A. A. Rezaee: *Soft Comput.* **22** (2018) 4613. <https://doi.org/10.1007/s00500-017-2644-1>
- 11 M. Yolmeh, M. B. H. Najafi, and F. Salehi: *Microb. Pathog.* **67–68** (2014) 36. <https://doi.org/10.1016/j.micpath.2014.02.003>
- 12 R. Kuo, Y. Tseng, and Z. Y. Chen: *J. Intell. Manuf.* **27** (2016) 1191. <https://doi.org/10.1007/s10845-014-0944-1>
- 13 K.V.Shihabudheen, M.Mahesh, and G.N.Pillai: *Expert Syst. Appl.* **92** (2018) 474. <https://doi.org/10.1016/j.eswa.2017.09.037>
- 14 Y. Hou, L. Zhao, and H. Lu: *Future Gener. Comput. Syst.* **81** (2018) 425. <https://doi.org/10.1016/j.future.2017.08.041>
- 15 G. Taguchi and S. Konishi: *Taguchi Methods Orthogonal Arrays and Linear Graphs: Tools for Quality Engineering* (American Supplier Institute, 1987) pp. 8–35.
- 16 D. Psaltis, A. Sideris, and A. Yamamura: *IEEE Contr. Syst.* **8** (1988) 17. <https://doi.org/10.1109/37.1868>