Sādhanā (2020)45:252
https://doi.org/10.1007/s12046-020-01491-2

An integrated feedforward-feedback control structure utilizing a simplified global gravitational search algorithm to control nonlinear systems

OMAR FAROUQ LUTFY

Control and Systems Engineering Department, University of Technology, Baghdad, Iraq
e-mail: 60157@uotechnology.edu.iq

MS received 15 May 2018; revised 18 April 2020; accepted 8 September 2020

Abstract. This paper presents an integrated feedforward-feedback control structure to control nonlinear dynamical systems. This intelligent control system exploits a modified recurrent wavelet neural network (MRWNN) in the feedforward (FF) and the feedback (FB) loops of the control structure. Specifically, the MRWNN is proposed to boost the approximation performance of a previously reported network by employing two amendments to the original structure. To optimize the parameters of both the FF and the FB controllers, an enhanced version of the gravitational search algorithm (GSA) is developed to improve the searching capability of the original algorithm. In particular, two modifications were adopted, including the removal of two control parameters related to the gravitational constant in the original algorithm and the utilization of the global best solution to constitute the next generation of agents. Hence, the proposed algorithm is called the simplified global gravitational search algorithm (SGGSA), which has demonstrated better optimization performance compared to those of other techniques, including the original GSA. By conducting several evaluation tests using different nonlinear time-variant dynamical systems, the effectiveness of the proposed control structure was confirmed in terms of control precision and robustness against external disturbances. In addition, the MRWNN has exhibited a superior control performance compared with other related controllers.

Keywords. Integrated feedforward-feedback control structure; wavelet neural network; recurrent wavelet neural network; gravitational search algorithm; genetic algorithm.

1. Introduction

The increased complexity of modern industrial systems has rendered linear modeling and control techniques insufficient tools that will inevitably result in poor control performance. The reason behind this performance degradation in linear control techniques can be attributed to the description of the complex and nonlinear physical system behavior by just an approximated linear model [1]. To tackle this difficulty, more attention has been paid to utilize intelligent control methods that can be directly applied to handle the complexity and nonlinearity of the systems. In particular, artificial intelligence techniques can be combined with conventional control schemes to form efficient nonlinear control systems. To this end, among several control schemes, the feedback (FB) and the feedforward (FF) strategies are the most commonly applied schemes due to their simple configuration and good control performance. However, each of these structures has certain drawbacks and limitations that might negatively influence the accuracy of the overall control system [2–4]. To alleviate this problem, the FF and the FB control strategies can be combined to constitute a more powerful control framework, which is generally referred to as the FF-FB control scheme.

In this regard, due to its simplicity in implementation and widespread acceptance, the proportional-integral-derivate (PID) controller, or one of its reduced forms, namely the PI or the PD controller, has been extensively applied as a feedback controller for synthesizing the FF-FB control scheme to solve various control problems [3–9]. Nevertheless, to maintain an acceptable control performance, the gains and time constants of the PID controller must be re-tuned frequently according to the variations in the system operating conditions and the surrounding environment. Moreover, as a linear controller, the PID controller's ability to handle the nonlinear and time-varying behaviors of complex systems is questionable. Using another FF-FB control scheme, the swing-up control of a triple pendulum on a cart was considered in [10]. In particular, the authors utilized the optimal control theory to design a state feedback controller after linearizing the triple pendulum system model. In [11], the feedback error learning strategy was adopted for the load frequency control in the restructured power system. This control approach uses the output of a classical PD controller as a training signal for

Published online: 06 October 2020

optimizing the parameters of a type-2 fuzzy logic FF controller. As another linear FF-FB control system, the FB controller was designed by the H_2 mixed sensitivity approach in [12]. It is worth noticing that in the aforementioned works, the FB controllers were designed at the nominal operating conditions since these controllers are based on linear control techniques. Hence, such controllers might fail to guarantee the best control performance over a wide range of operating conditions, especially for complex nonlinear systems.

The artificial neural network (ANN) represents an important computational intelligence technique, which has been widely applied for solving different identification and control problems. In this context, some efforts have been expended on utilizing the ANN as the FB controller in the FF-FB control strategy. For instance, Li *et al* [13] proposed a hybrid control approach that comprises a FF controller and a single neuron FB controller for the ship fin stabilizer system. A gradient-based method was employed to adjust the connection weights of the FB controller. However, as a limitation in this method, an additional neural network was employed to identify the system and then to provide the Jacobian information required by the gradient-based method. Following the same control strategy, Baruch *et al* [14] designed a control system that incorporates a FF controller, a FB controller, an identifier, and a noise-rejecting filter to control the model of a nonlinear wastewater treatment bioprocess. More specifically, three recurrent neural networks (RNNs) were exploited to act as the FF, the FB, and the identifier in the proposed structure. The Levenberg-Marquardt (LM) algorithm, which is a gradient-based method, was applied to train the above networks. The task of the RNN identifier is to estimate the system states which are then used to design the FB controller. However, besides the complexity of this method, which requires an additional network to form the FB control law, a proper selection of the filter equation must be made in advance, where inappropriate filter dynamics can deteriorate the control system performance.

Recently, a new neural network variant, namely the wavelet neural network (WNN), has gained widespread popularity among researchers due to its superior approximation ability compared to the conventional ANN. However, the utilization of the WNN as a FB controller in the FF-FB control scheme has not been considered in general. As far as nonlinear time-varying systems are concerned, feedforward neural networks, including the multilayer perceptron (MLP) and the WNN, produce inadequate control performance compared to recurrent networks, which have a supporting memory to improve their approximation ability [15]. Pursued from that, as an improved version of the WNN, the self-recurrent wavelet neural network (SRWNN) has the added advantage of storing the previous wavelon layer state to be used in producing the network output [16].

There is an increased understanding that a proper selection for the neural network training algorithm represents an indispensable requirement for the success of the overall control system. In this respect, the gradient-based methods, such as the back-propagation (BP) and the LM algorithm, are the most widely used methods for implementing the FF-FB control schemes [13, 14]. Nonetheless, these methods are characterized by their slow convergence rate, the dependency on an appropriate selection for the learning and inertial factors, and the tendency to get trapped in local minima of the search space [17]. Furthermore, when these methods are used to train the FB controller in the FF-FB structure, a supplementary neural network identifier is needed to provide the necessary Jacobian matrix, which complicates the control design and adds an extra computational burden. The above difficulties associated with gradient-based methods have rendered the researchers to utilize evolutionary algorithms (EAs), which are more likely to arrive at the global optimal solution of a particular problem. As a novel evolutionary algorithm, the gravitational search algorithm (GSA) was recently developed by utilizing Newton's laws of gravity and motion [18]. However, for a good optimization performance, there should be a good balance between the exploration and the exploitation capabilities of the optimization method. Unfortunately, the original GSA is characterized by its good exploration search but weak exploitation for the promising regions in the search space [19].

The main objective of the present work is to develop an intelligent FF-FB control structure, in which a modified recurrent wavelet neural network (MRWNN) is employed in the FF and the FB loops to control nonlinear discrete-time systems. Particularly, the MRWNN is proposed to improve the approximation ability of a recently published recurrent WNN [16, 20]. To avoid the drawbacks of gradient-based methods, an enhanced version of the GSA, namely the simplified global gravitational search algorithm (SGGSA), is proposed to optimize the parameters of both the FF and the FB controllers. The simulation results of controlling different nonlinear systems, including time-variant ones, have confirmed the applicability of the proposed FF-FB control structure to control various nonlinear systems. The organizational flow of the remaining sections includes the following: Section 2 gives the details of designing the FF-FB control structure considered in this work. The proposed MRWNN structure is discussed in section 3. Section 4 provides an outline of the original GSA and clarifies the proposed SGGSA. Section 5 presents the results of several evaluation tests and comparative studies to demonstrate the effectiveness of the FF-FB control structure. Finally, the conclusions from this work are given in section 6.

2. The feedforward-feedback control structure using the MRWNN

As mentioned before, each of the FF and the FB control strategies has certain advantages and limitations. In particular, the FB control method is characterized by its simple

design and satisfactory control performance. However, if the controlled system has a particular time-delay, the FB controller will not directly affect the system until a certain period has elapsed. This delay in the FB controller response might deteriorate the overall control performance and cause stability problems [2]. In contrast, the FF controller can detect variations in the reference signal in advance and deliver its corresponding action directly to the controlled system [3]. Moreover, since no feedback signal is generally required, the FF controller does not cause stability problems [4]. However, to design an efficient FF controller, there is a need for an accurate inverse model of the system, which is not straightforward to attain, especially for complex nonlinear systems. Furthermore, the FF controller lacks the ability to handle adverse operating conditions that the control system might encounter, such as parameter variations in the system dynamics and the unexpected external disturbances. In light of the above facts, by combining the FF and the FB control strategies, a more powerful control structure can be materialized, which is generally known as the FF-FB control structure whose block diagram is illustrated in figure 1. The following sections elucidate the design procedure of the FF and the FB controllers.

2.1 *The FF controller design*

As a preliminary design step to realizing the control system considered in this work, a suitable FF controller should be synthesized. In this regard, the basic concept behind designing a nonlinear FF controller is to develop an accurate model that describes the inverse dynamics of the nonlinear system. This inverse model is then applied as the FF controller [21]. More precisely, consider the nonlinear system described by the following equation [22]:

$$y(k+1) = f(y(k), y(k-1), \ldots, y(k-n+1), u(k), u(k-1), \ldots, u(k-m)), \quad (1)$$

where n is the number of previous outputs, m is the number of previous inputs, and f is a nonlinear function. To control the system in eq. (1), a model representing the inverse system dynamics should be attained, so that:

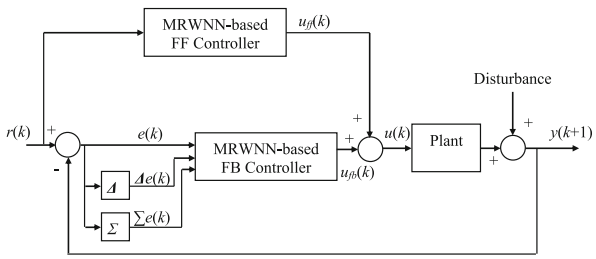

Figure 1. A block diagram of the FF-FB control structure, in which the MRWNN is used in the FF and the FB loops.

$$u(k) = f^{-1}(y(k+1), y(k), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m)), \quad (2)$$

where $f^{-1}$is a nonlinear function representing the inverse system dynamics. In this context, ANNs can be trained to acquire the nonlinear function of eq. (2). Specifically, the MRWNN is exploited in this work to accomplish this control objective, as illustrated in figure 2.

The training process is performed by adjusting the parameters of the MRWNN to minimize the mean squared error (MSE) criterion given below:

$$J = \frac{1}{N}\sum_{k=1}^{N}(u(k) - \hat{u}(k))^2, \quad (3)$$

where N is the number of training samples, while $u(k)$ and $\hat{u}(k)$ represent the system input and the MRWNN output, respectively. After the training stage is completed, the MRWNN structure is used as the FF controller by replacing the output of the plant $y(k+1)$ in eq. (2) by the command signal $r(k+1)$, as can be seen from figure 1. However, as a weakness in this control method, the output error, i.e. the error between the command signal and the real system output, is not minimized during the inverse modeling stage. As such, the controller developed by this method might cause a steady-state error between the desired and the actual system outputs [23]. Therefore, to achieve satisfactory control accuracy, the FF controller is augmented with a FB controller, which is described in the subsequent section.

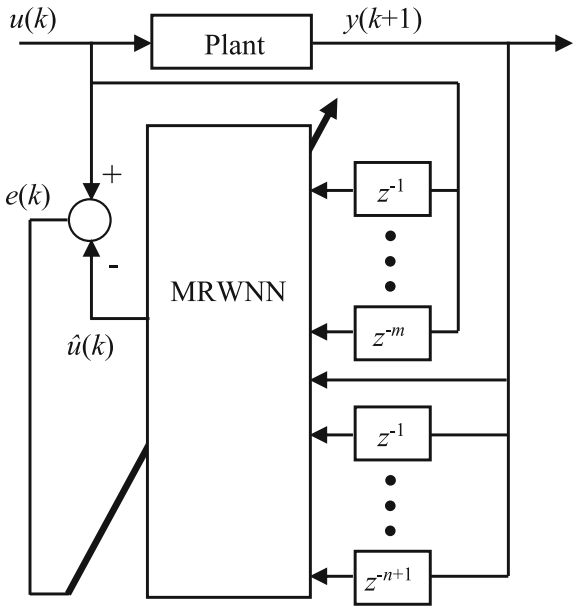

Figure 2. Training the MRWNN to act as an inverse model for the plant.

2.2 *The FB controller design*

Having developed an appropriate FF controller in the previous section, the next step is to design the FB controller to constitute the FF-FB control scheme. In essence, the FB control system is achieved by generating an error signal representing the variation between the command signal and the real system output. This error signal, which is also known as the control error, is used by the FB controller to determine the appropriate control action that forces the system output to follow the command signal. As illustrated in figure 1, the FB controller is implemented as a PID-like MRWNN controller in this work. Therefore, at a given time sample k, this controller receives three input signals representing the control error $e(k)$, the error rate of change $\Delta e(k)$, and the summation of errors $\sum e(k)$ to generate the FB controller output denoted as $u_{fb}(k)$. This FB control signal is then combined with the FF controller output $u_{ff}(k)$ to form the final control action $u(k)$, as shown in figure 1.

3. The MRWNN structure

In an attempt to boost the approximation ability of a previously reported recurrent WNN [16, 20], a suggestion was made to include feedback connections from the output node to the wavelon layer. Furthermore, the RASP1 function was used instead of the widely applied Mexican Hat function. These modifications to the original structure have contributed to the improvement in the network approximation ability, as will be demonstrated in section 5.4. The structure of the proposed MRWNN is depicted in figure 3.

As it is obvious from figure 3, the structure of the MRWNN comprises three layers, which are described in the following:

Layer 1: This is the input layer whose task is to transfer the input variables as they are to the next layer. In the present work, each of the FF and the FB controllers is implemented using the same MRWNN structure. However, the form of the input variables in layer 1 is different for each controller. More precisely, to employ the MRWNN as a FF controller using eq. (2), the form of the input variables should be as given below:

$$[y(k+1), y(k),\ \ldots, y(k-n+1), u(k-1),\ \ldots, u(k-m)] \tag{4}$$

On the other hand, utilizing the MRWNN structure as a PID-like FB controller entails the selection of the input variables to be; the control error $e(k)$, the error rate of change $\Delta e(k)$, and the summation of errors $\sum e(k)$, as it is apparent from figure 1.

Layer 2: This layer is known as the mother wavelet or the wavelon layer. As illustrated in figure 3, each node, or wavelon, in this layer receives three types of input variables namely, a weight from each input node, a self-feedback weight, and a feedback weight from the output node. For the jth wavelon, these input variables are used to compute the corresponding output using the following expression:

$$z_j = d_j\left(\sum_{i=1}^{Ni} v_{ji}x_i + \psi_j(k-1)\cdot\theta_j + y(k-1)\cdot\beta_j\right) - t_j, \tag{5}$$

where d_j and t_j are the dilation and the translation parameters of the j^{th} wavelon, respectively, Ni is the number of input variables in layer 1, v_{ji} is the weight that connects the i^{th} input node with the j^{th} wavelon, x_i represents the i^{th} input variable, $\psi_j(k-1)$is the preceding response from the j^{th} wavelon, $\theta_j$is the parameter related to the j^{th} self-feedback weight, $y(k\text{-}1)$ is the previous network output, and $\beta_j$is the parameter of the weight that connects the output node with the j^{th} wavelon.

Regarding the wavelet function, it is increasingly understood that selecting a suitable wavelet activation function is as critical as selecting the network structure and the training algorithm [24]. To this end, after several trials with various wavelet functions, it was found that the RASP1 function gives the best approximation performance compared to other function types. Therefore, the RASP1 function was adopted to compute the j^{th} wavelon output according to the following equation:

$$\psi_j(z_j) = \frac{z_j}{(z_j^2+1)^2}, \tag{6}$$

where $z_j$is the output of eq. (5).

Layer 3: In this layer, a single node produces the final output of the MRWNN structure using the following formula:

$$y = \sum_{j=1}^{Nw} c_j\psi_j(z_j) + \sum_{i=1}^{Ni} a_i x_i + b, \tag{7}$$

where Nw is the number of nodes in the wavelon layer, Ni is the number of nodes in the input layer, c_j signifies the weight connecting the j^{th} wavelon and the output node, a_i is the weight connecting the i^{th} input node and the output node, and b is a bias to the output node. Based on the preceding details, it is apparent that the MRWNN structure involves several modifiable weights, which can be included in the following set:

$$S = \left[v_{ji}d_j t_j c_j \theta_j \beta_j a_i b\right], \tag{8}$$

In order to use the MRWNN structure as a FF controller, the weights specified in eq. (8) should be trained based on minimizing the error between the input to the system and the output of the MRWNN structure, as indicated in figure 2. For this purpose, the MSE

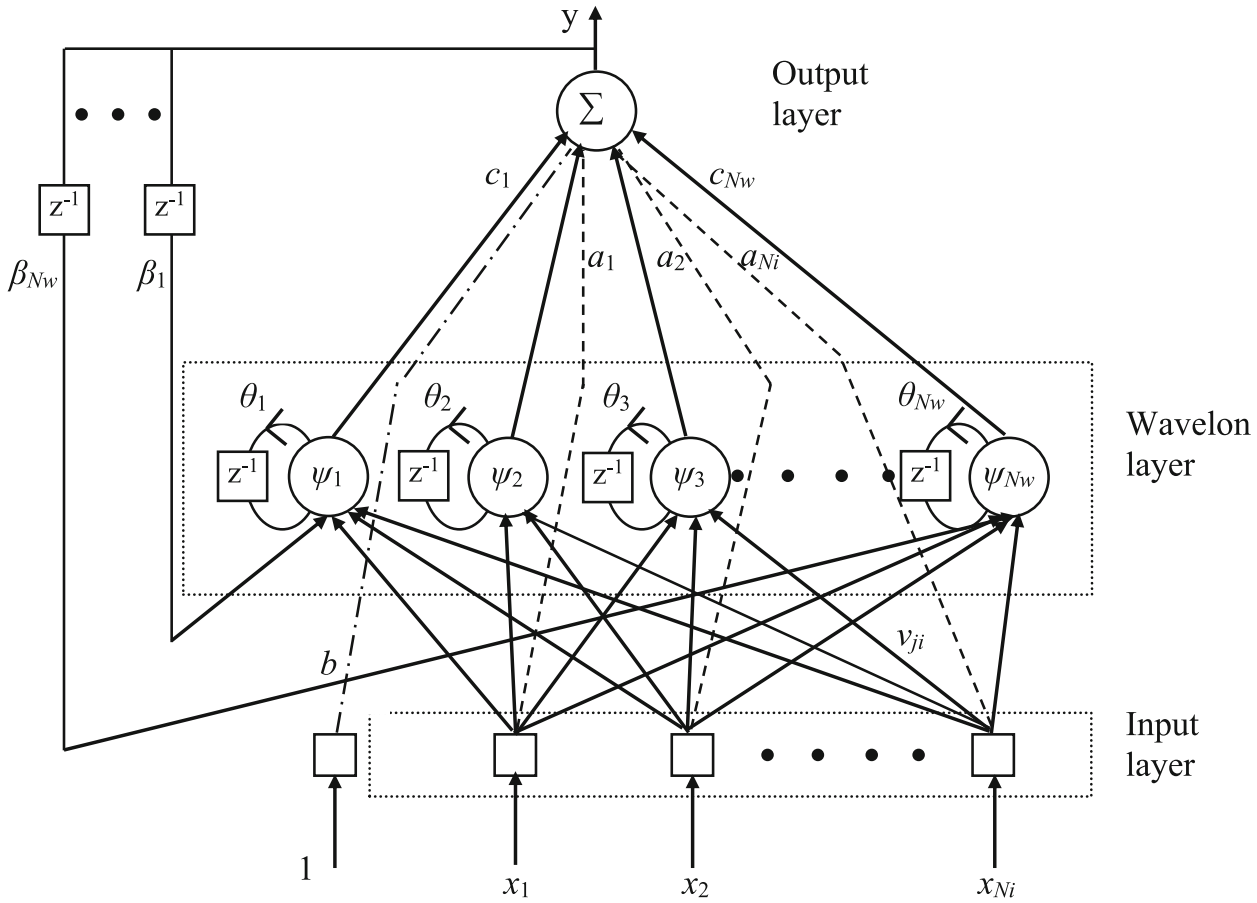

Figure 3. Structure of the MRWNN.

criterion defined in eq. (3) is utilized as the cost function to be minimized by the optimization method. While for the FB controller training, the same parameters of eq. (8) are optimized by minimizing the control error representing the difference between the desired and the actual system outputs. In this case, the integral square of error (ISE) criterion is employed as the cost function in the optimization algorithm. Specifically, the ISE has the following form:

$$J = \frac{1}{2}\sum_{k=1}^{N}(r(k) - y(k))^2, \tag{9}$$

where N is the number of time samples, while $r(k)$ and $y(k)$ represent the reference signal and the actual system output, respectively. In this work, the aforementioned optimization processes are performed by the proposed SGGSA. Therefore, the following section will shed some light on the original GSA, its proposed modified version, and the procedure to apply the latter as the training method for both the FF and the FB controllers.

4. The gravitational search algorithm

The gravitational search algorithm (GSA) is a nature-inspired optimization technique that has been recently proposed by Rashedi *et al* [18]. Ever since its first appearance, the GSA has attracted much attention in the literature due to its simple procedure and promising performance in solving various optimization problems [25–27]. In particular, the GSA has demonstrated better optimization results compared to other well-known EA techniques, such as the genetic algorithm (GA) and the particle swarm optimization (PSO) [28, 29].

As the main evolutionary principle, the GSA works on a population of agents representing candidate solutions to the optimization problem. Each of these agents possesses a specific mass, position, and velocity. These agents are search entities that cooperate with each other using Newton's laws of gravity and motion. At each iteration, the gravitational force applied by the top K best agents on the ith agent, denoted as $X_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,j}^t, \ldots, x_{i,D}^t]$, is calculated according to the following formula [18, 19]:

$$F_{i,j}^t = \sum_{k \in Kbest, k \neq i} rand_k \times G^t \times \frac{M_k^t \times M_i^t}{R_{i,k} + \varepsilon} \times \left(x_{k,j}^t - x_{i,j}^t\right), \tag{10}$$

where t indicates the current iteration, $i = 1, 2, \ldots, NP$, $j = 1, 2, \ldots, D$, NP is the number of agents in the population, D is the number of decision variables in each agent, $Kbest$ represents the set of the top K best agents, $rand_k$ is a random value within the range [0, 1], $G^t$is the time-dependent gravitational constant, M_k^t and $M_i^t$are the masses of agents k and i, respectively, $R_{i,k}$ denotes the distance between agents i and k, ε is a small constant, $x_{k,j}^t$ and $x_{i,j}^t$ are the jth decision variables of agents k and i, respectively. The gravitational constant in eq. (10) is calculated using the following expression:

$$G^t = G^0 \times \exp\left(-\alpha \times \frac{t}{t_{\max}}\right), \tag{11}$$

where G^0 is the initial value of the gravitational constant, $\alpha$is the decreasing coefficient, and t and t_{max} denote the current iteration and the maximum number of iterations, respectively. As the quality index, the mass of the ith agent is determined by the following equation:

$$M_i^t = \frac{q_i^t}{\sum_{k=1}^{NP} q_k^t}, \tag{12}$$

and q_i^t is calculated from:

$$q_i^t = \frac{f(X_i^t) - f(X_{Worst}^t)}{f(X_{Best}^t) - f(X_{Worst}^t)}, \tag{13}$$

where X_{Worst}^t and X_{Best}^t represent the agents with the worst and the best fitness values, respectively, and $f(\cdot)$is the fitness function. After finding the total gravitational force acting on each agent, the corresponding acceleration is computed as follows [18, 19]:

$$a_{i,j}^t = \frac{F_{i,j}^t}{M_i^t} = \sum_{k \in Kbest,\ k \neq i} rand_k \times G^t \times \frac{M_k^t}{R_{i,k} + \varepsilon} \times (x_{k,j}^t - x_{i,j}^t) \tag{14}$$

Once the acceleration is obtained, the new velocity and position of each agent are calculated by the following equations:

$$v_{i,j}^{t+1} = rand_i \times v_{i,j}^t + a_{i,j}^t \tag{15}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{16}$$

4.1 *The proposed simplified global gravitational search algorithm*

It should be borne in mind that a reasonable balance between the exploration and the exploitation abilities is an important requirement for the success of any evolutionary algorithm. Unfortunately, this balance is insufficient in the original GSA [19, 30]. In fact, the exploitation task is achieved by utilizing the top best agents in the current population to guide the search process. Nevertheless, this course of action does not preserve the global best agent obtained thus far, which means no information can be extracted from such a useful solution. Consequently, the exploitation search in the GSA lacks the valuable knowledge provided by the global best solution acquired so far [19]. This memoryless nature of the original algorithm might unsurprisingly lead the search process to stagnate in local optima and mislay the global optimal solution [29]. To overcome this difficulty, several attempts have been made to enhance the exploitation search of the original GSA. For instance, by borrowing the social and cognitive communications among the particles in the PSO, two acceleration constants were used to calculate the forces that act on the agents [28, 30]. However, this proposal requires the addition of two more control variables, which should be selected based on a trial and error basis. Following the same research direction, Darzi *et al* [29] utilized the local best solution of each agent to update its position. However, the global best solution was not considered in this method, which means disregarding a valuable search trajectory. To extend the searching space around the global optimal solution, Chao *et al* [31] used the standard normal distribution function along with information from the agents' velocity to mutate the global best agent. Similar to the limitation in [29], the global best agent was not used to guide the search direction for the other agents. In another attempt to enhance the GSA exploitation search, Yin *et al* [19] proposed to add a crossover operator similar to that employed in the genetic algorithm. This operator randomly inherits some information from the global best solution with a certain crossover probability, denoted as CR. However, selecting an appropriate CR value, which considerably affects the search performance, required a series of experiments by the authors. In addition, after several trials, the authors have used a value of 0.1 for CR, which means that the proposed crossover operator does not significantly affect the original GSA performance. It is worth pointing out that all the above-mentioned efforts have used the original control parameters of the GSA, namely G^0 and α in eq. (11). These two control parameters of the gravitational constant should be selected in advance based on a trial and error method. For a particular application, this selection process adds additional difficulty represented by choosing the proper settings for the above parameters.

To address the above problems, this work presents a modified variant of the original GSA, namely the simplified global gravitational search algorithm (SGGSA). In the proposed algorithm, after performing the same computational steps described above, the exploitation ability is enhanced by utilizing the promising information extracted

from the global optimal solution attained thus far using the following formula:

$$x_{i,j}^{t+1} = x_{best,j}^{t} + \Phi_{i,j}(x_{r1,j}^{t} - x_{r2,j}^{t}), \tag{17}$$

where $r1$ and $r2$ denote two different integers randomly chosen from the set {1, 2, …, NP} representing the indices to the agents of the current population. These integers, $r1$ and $r2$, must not equal to the index of the current agent i, x_{best}^{t} indicates the global best solution obtained thus far, j represents an index to a decision variable, and $\Phi_{i,j}$is a random number from [− 1, 1]. For each agent, eq. (17) above is applied two times to adjust two randomly selected decision variables. Due to its powerful impact, this modification can compensate for the effect of the gravitational constant in the original GSA. Therefore, the gravitational constant, which requires two control parameters, is not included in the proposed algorithm. The above modifications result in a reasonable compromise between the exploration and the exploitation abilities in the proposed SGGSA. Therefore, the SGGSA has the potential to overcome other optimization methods, including the original GSA, as demonstrated in section 5.3.

4.2 *The procedure of utilizing the SGGSA as the training method in the FF-FB control structure*

The proposed algorithm clarified in the preceding section is employed to optimize the parameters of both the FF and the FB controllers considered in this work using the following steps:

Step 1: Initialize the maximum number of iterations and the number of agents (NP).
Step 2: By randomly generating NP agents within certain bounds, form an initial population to commence the search process. Each agent in this population represents a single MRWNN structure, which has the adjustable parameters given in eq. (8).
Step 3: Determine the cost function of each agent in the current population. To achieve this task, the MSE in eq. (3) is used in the FF controller training stage, whereas for the FB controller training stage, the ISE criterion defined in eq. (9) is used. Afterward, determine the fitness function of each agent by the following equation:

$$f(X_i^t) = \frac{1}{\text{Cost}\,function\,(X_i^t) + \varepsilon}, \tag{18}$$

where ε is a small value to evade the division by zero.
Step 4: Memorize the global solution acquired until now.
Step 5: Determine the mass of each agent according to eq. (12).
Step 6: For each agent, calculate the gravitational force applied by the top K best agents using eq. (10). Then, calculate the agent's acceleration, velocity, and position using eqs. (14), (15), and (16), respectively.
Step 7: As the proposed performance enhancement step, apply eq. (17) to adjust two randomly selected decision variables for each agent to exploit the promising knowledge provided by the global optimal solution obtained thus far.
Step 8: If the stopping condition, which is the maximum number of iterations in this work, is satisfied, the algorithm is terminated and the final optimized parameters of the MRWNN structure are given by the current global best solution. Otherwise, go to Step 3.

5. Simulation results

This section is dedicated to examining the applicability of the MRWNN-based FF-FB control structure to handle nonlinear discrete-time systems. For both the FF and the FB controllers, the SGGSA training procedure discussed in the previous section was employed to optimize the weights of the MRWNN structure. In the optimization process, 50 agents were utilized to form each population and the maximum number of iterations was chosen to be 500. In addition, six wavelons were used to constitute the wavelon layer in the MRWNN structure. For all the controlled systems, the above settings for the SGGSA and the MRWNN were adequate for achieving the required control objective.

5.1 *Control performance tests*

These tests aim at investigating the control precision of the proposed MRWNN-based FF-FB structure in controlling the nonlinear systems given below.

5.1.1 *Plant 1:* This is a complex, nonlinear, and time-variant plant representing a laboratory-scale liquid-level system. This system comprises a DC water pump that feeds a conical flask, which in turn feeds a square tank. The voltage to the pump motor is the manipulated variable, while the height of the water in the conical flask represents the controlled variable. The control objective is to make the water height follow some desired reference signal by manipulating the voltage to the pump motor. The following discrete-time expression represents the dynamics of this system [32, 33]:

$$\begin{aligned} y(k+1) =& a(k)y(k) + 0.3578u(k) - 0.1295u(k-1) - 0.3103y(k)u(k) \\ & - 0.04228y^2(k-1) + 0.1663y(k-1)u(k-1) \\ & - 0.03259y^2(k)y(k-1) - 0.3513y^2(k)u(k-1) \\ & + 0.3084y(k)y(k-1)u(k-1) + 0.1087y(k-1)u(k)u(k-1), \end{aligned} \tag{19}$$

where the time-variant coefficient is given by $a(k) = 1 - 0.4\sin\left(\frac{2\pi k}{T}\right)$, and T is selected to be 0.05. To realize the

FF-FB control structure depicted in figure 1, the MRWNN was trained to represent the inverse dynamics of Plant 1. In this context, to guarantee the accuracy of the model in representing the inverse system dynamics, the training data must be selected so that they cover a sufficiently large domain of the system input-output space [34]. Hence, the training signal, i.e., $u(k)$ in figure 2, is chosen to be a random signal, such that $|u(k)| <= 1$. To accomplish the inverse modeling task, the proposed SGGSA has been utilized for optimizing the parameters of the MRWNN structure. Figure 4 demonstrates the simulation results of controlling this plant.

As it is obvious from figure 4(a), the MRWNN has done well in representing the inverse dynamics of the system with an MSE value of 2.669×10^{-3}. Figure 4(b) illustrates the minimization of the MSE achieved by the proposed SGGSA. By applying the developed inverse model as the FF controller, the SGGSA method was used to train the PID-like MRWNN structure to act as the FB controller in the control system depicted in figure 1. The good control performance of the FF-FB control strategy in tracking a changing step signal is demonstrated in figure 4(c). As an important fact to conclude from figure 4(c), the MRWNN-based FF controller has shown notable generalization ability, since the random training signal used during the inverse modeling stage is entirely different from the changing step signal applied during the control stage. Figure 4(d) demonstrates the combined FF-FB control signal applied to Plant 1. The reduction in the ISE achieved by the SGGSA during the FB controller training is shown in figure 4(e).

5.1.2 *Plant 2:* As a nonlinear time-variant system, the following equation is utilized for this plant [1]:

$$y(k+1) = \frac{y(k)y(k-1)y(k-2)u(k)(y(k-2)-b(k)) + c(k)u(k)}{a(k) + y^2(k-1) + y^2(k-2)}, \tag{20}$$

where the time-variant coefficients are defined as: $a(k) = 1.2 - 0.2\cos\left(\frac{2\pi k}{T}\right)$, $b(k) = 1 - 0.4\sin\left(\frac{2\pi k}{T}\right)$, $c(k) = 1 + 0.4\sin\left(\frac{2\pi k}{T}\right)$, and T is selected to be 0.05. Following the same design steps used to control the previous plant, a random training signal, $u(k)$ (where $|u(k)| <= 1$), has been used to develop an MRWNN-based inverse plant model, which was subsequently used as the FF controller. Figure 5 shows the simulation results for Plant 2.

Figure 5(a) clearly shows that the MRWNN has successfully captured the inverse dynamics of Plant 2 with an MSE value of 0.966×10^{-3}. The efficiency of the SGGSA in minimizing the cost function can be seen in figure 5(b). Figure 5(c) indicates that the FF-FB control system was able to achieve the desired control objective by following the changing step signal. As in the case of the previous plant, the MRWNN-based FF controller has generalized its training very well by tracing a reference signal that differs from the random training signal utilized in the FF training stage. The FF-FB control action used to control Plant 2 is shown in figure 5(d). While figure 5(e) illustrates the best ISE values against iterations during the FB controller training phase.

5.1.3 *Plant 3:* This plant represents the dynamics of a highly complex and nonlinear system, which is defined by the following difference equation [35]:

$$\begin{aligned} y(k+1) = &\frac{2.5y(k)y(k-1)}{1+y^2(k)+y^2(k-1)} + 1.2u(k) + 0.09u(k)u(k-1) \\ &+ 1.6u(k-2) + 0.7\sin(0.5(y(k) \\ &+ y(k-1)))\cos(0.5(y(k)+y(k-1))) \end{aligned} \tag{21}$$

The same design approach adopted for the previous plants was used to control Plant 3. The simulation results for this plant are presented in figure 6.

Evidently, figure 6(a) signifies that the MRWNN structure has performed well in the inverse modeling stage attaining an MSE value of 4.337×10^{-3}. A decrease in MSE values during the 500 iterations of the SGGSA is illustrated in figure 6(b). In spite of the complex and the highly nonlinear nature of Plant 3, figure 6(c) shows that the FF-FB control structure managed to follow the desired response with zero steady-state error and with some oscillations at each step change in the input signal. These oscillations were the direct result of the existence of the sine and cosine terms in the dynamics of this plant, as exhibited in eq. (21). Hence, unlike the control signals of Plants 1 and 2, figure 6(d) demonstrates some spikes and oscillations that were produced in the control signal to cope with the complexity and nonlinearity of Plant 3. To attenuate such oscillations, certain bounds could be imposed on the control signal. Additionally, figure 6(c) denotes the good generalization ability of the MRWNN-based FF controller. The minimization of the ISE against iterations is shown in figure 6(e).

5.2 *Robustness tests*

These tests were made to investigate the robustness ability of the MRWNN-based FF-FB control structure to handle the effects of external disturbances. To perform this test on each plant considered in the preceding section, a bounded disturbance with a magnitude of 10% of the system output was applied for 5 samples during two periods. The two periods are $121 \leq k \leq 125$ and $321 \leq k \leq 325$ for all controlled plants. It is worthy to notice that these disturbances were only applied during the testing phase, and not during the training phase of the FF and the FB controllers. This fact means that the FF-FB control system is not trained to deal with such disturbances. Nevertheless, figure 7 reveals that the FF-FB control system has accommodated the effects of these unexpected disturbances for all the

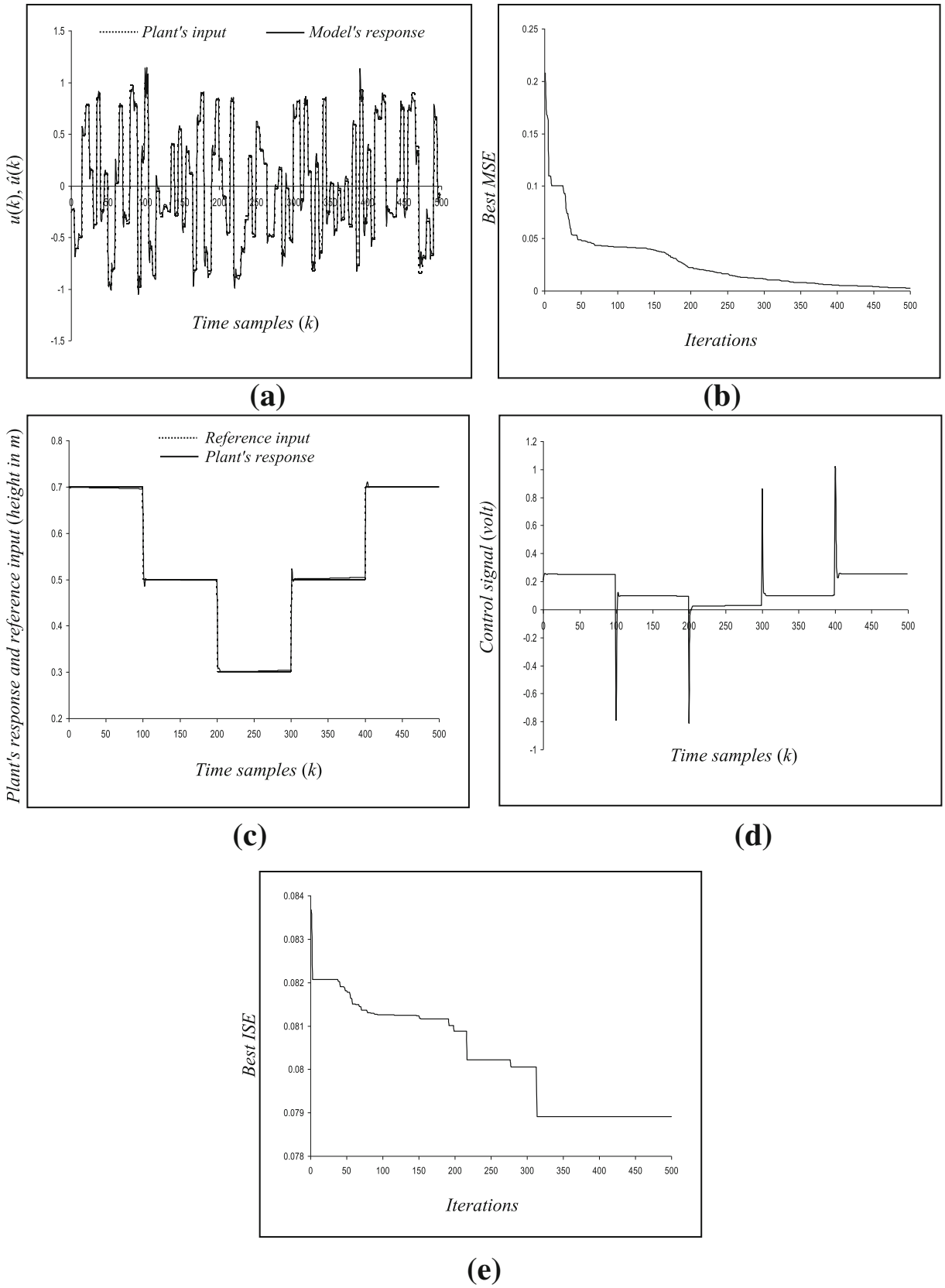

Figure 4. Plant 1 (a) plant's input and model's response, (b) best MSE against iterations, (c) plant's response, (d) control signal and (e) best ISE against iterations.

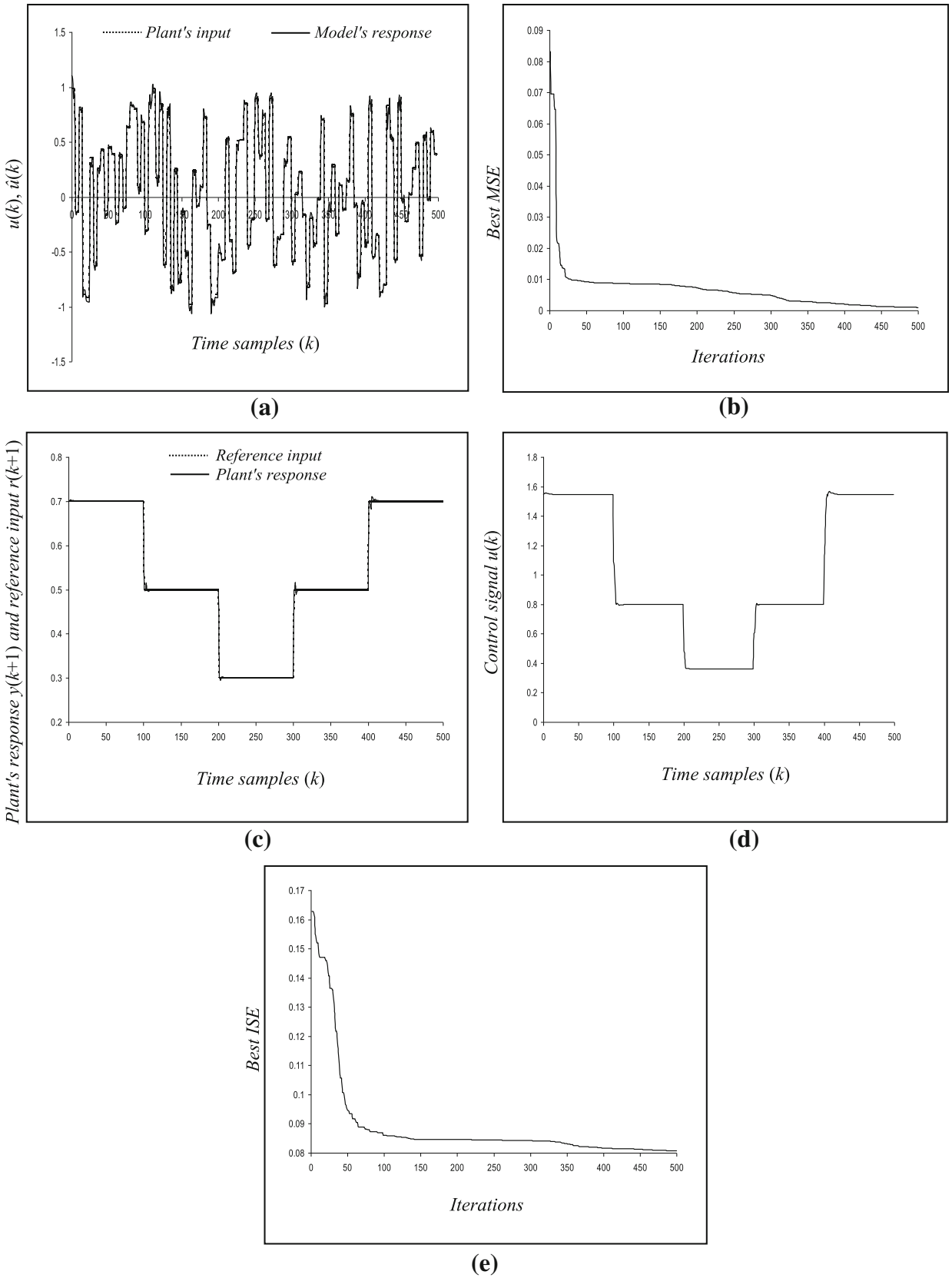

Figure 5. Plant 2 (a) plant's input and model's response, (b) best MSE against iterations, (c) plant's response, (d) control signal and (e) best ISE against iterations.

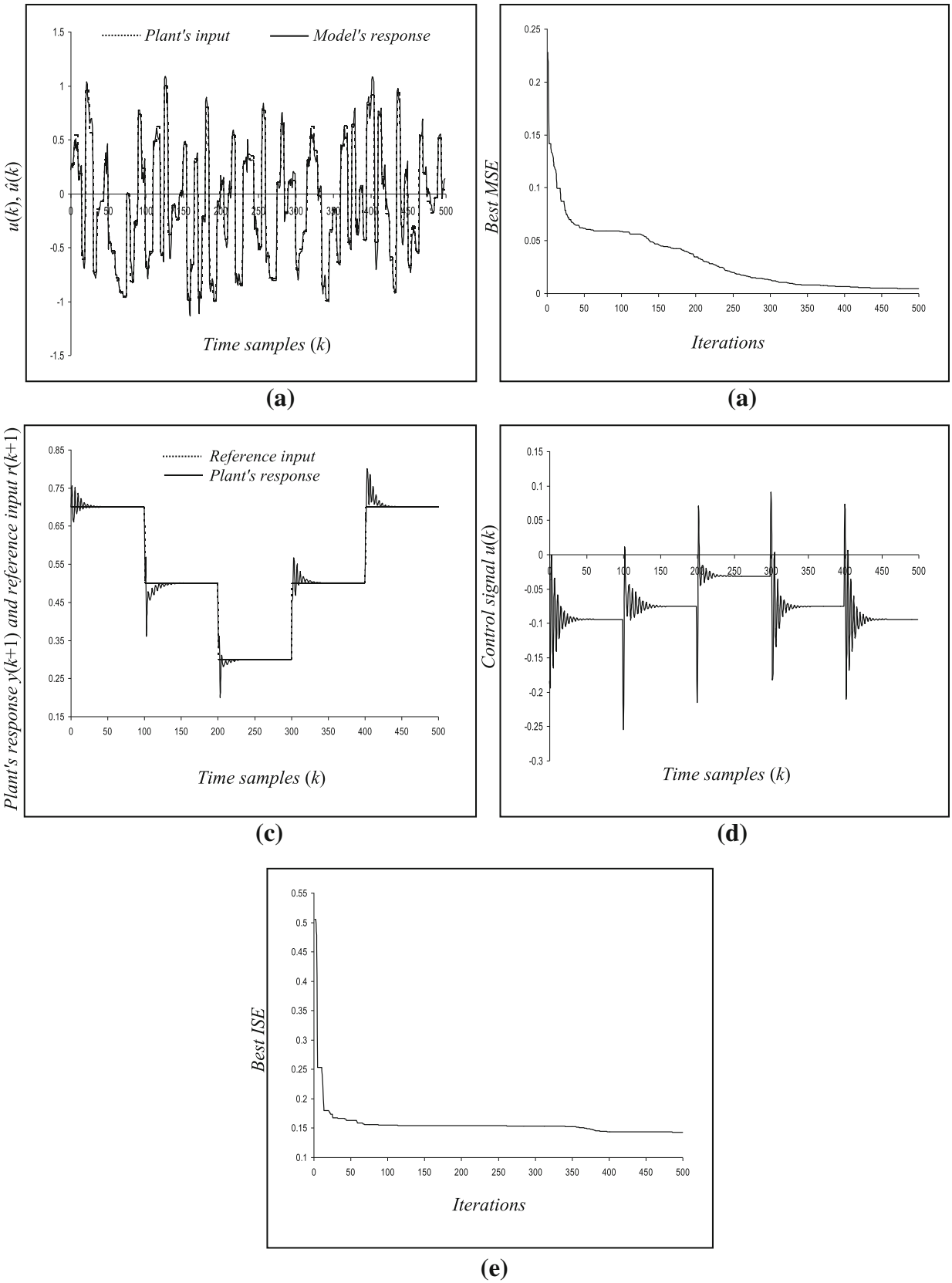

Figure 6. Plant 3 (a) plant's input and model's response, (b) best MSE against iterations, (c) plant's response, (d) control signal and (e) best ISE against iterations.

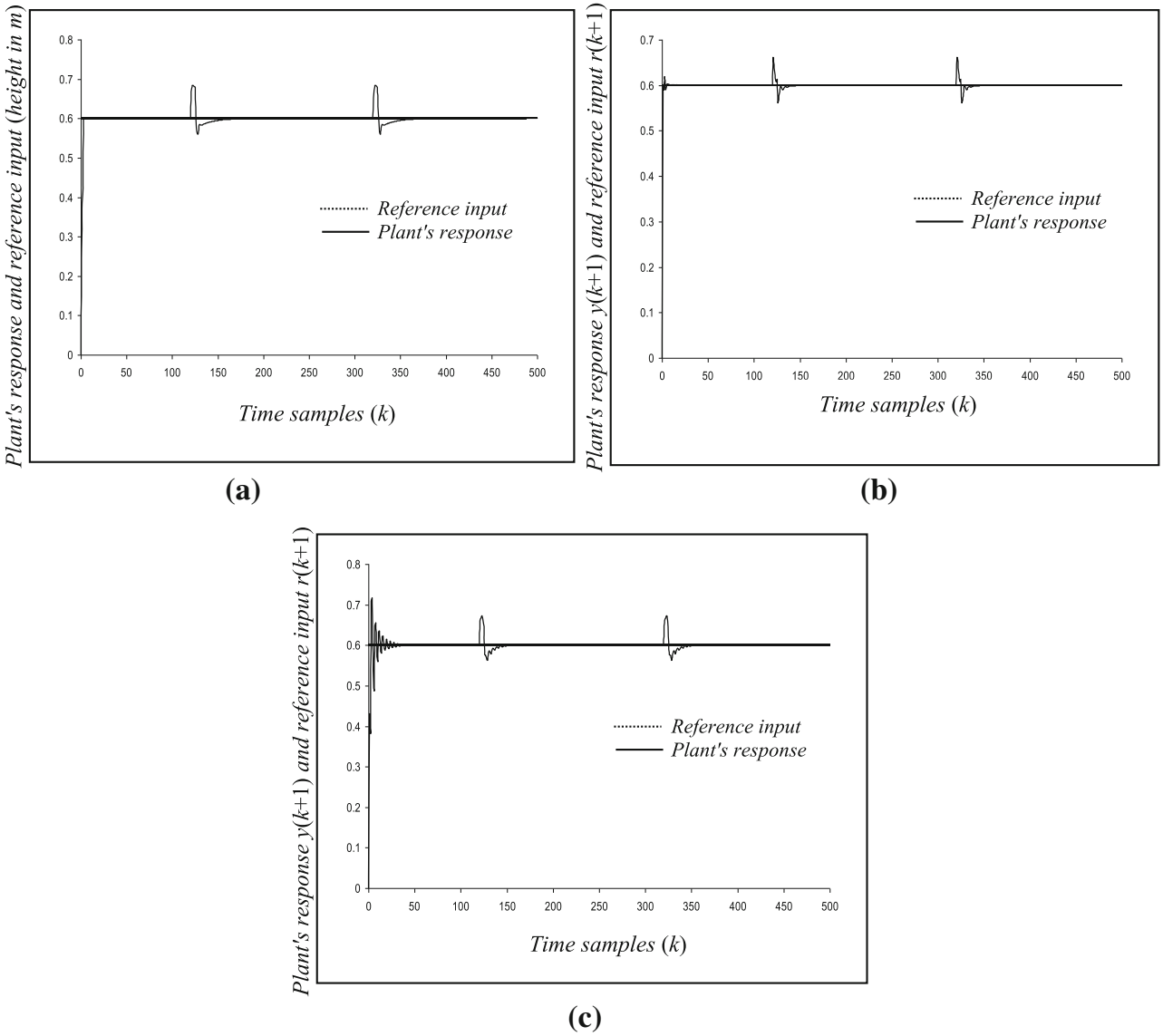

Figure 7. Disturbance rejection tests for (a) Plant 1, (b) Plant 2 and (c) Plant 3.

plants by recovering the desired response immediately after the influence of each disturbance.

5.3 *A comparative study with other optimization methods*

Since the proposed SGGSA is an enhanced version of the original GSA, it is important to compare the optimization results of the two algorithms. In addition, it is well-known that the genetic algorithm (GA) is regarded as one of the most popular types of evolutionary algorithms. Therefore, this section is devoted to comparing the optimization performance of each of the above algorithms along with another modified variant of the original GSA, which was proposed in [28] for enhancing the exploitation stage in GSA. This modified version was termed the gbest-guided GSA (GGSA). To carry out the comparative study, the same plants described in section 5.1 were used. Due to the stochastic nature of all the optimization methods considered in this study, each algorithm might generate different results for different runs. Hence, to handle this stochastic inconsistency, each algorithm was executed 10 times for each plant and the average result of these runs was adopted. Table 1 summarizes the results of this study.

In table 1, FF MSE and FF-FB ISE indicate cost functions for the FF and the FF-FB training phases, respectively, while FF Time and FF-FB Time indicate the training times for the FF and the FF-FB training phases, respectively. As indicated by the bold type, table 1 evidently

Table 1. Comparison results of the GA, the original GSA, the GGSA, and the proposed SGGSA as the training methods in the FF-FB control structure.

Optimization method	Criterion (average of 10 runs)	Controlled plant		
		Plant 1	Plant 2	Plant 3
GA	FF MSE	44×10^{-3}	8×10^{-3}	31×10^{-3}
	FF Time (sec.)	44.859	47.146	45.835
	FF-FB ISE	0.086	0.091	0.451
	FF-FB Time (sec.)	54.624	55.977	54.795
Original GSA	FF MSE	49×10^{-3}	9×10^{-3}	54×10^{-3}
	FF Time (sec.)	28.245	30.210	29.623
	FF-FB ISE	0.095	0.306	120.457
	FF-FB Time (sec.)	41.413	45.784	43.509
GGSA	FF MSE	48×10^{-3}	8×10^{-3}	59×10^{-3}
	FF Time (sec.)	**28.097**	29.893	29.351
	FF-FB ISE	0.086	0.097	1.009
	FF-FB Time (sec.)	41.185	44.160	42.606
The proposed SGGSA	FF MSE	$\mathbf{7 \times 10^{-3}}$	$\mathbf{1 \times 10^{-3}}$	$\mathbf{9 \times 10^{-3}}$
	FF Time (sec.)	28.110	**29.541**	**28.981**
	FF-FB ISE	**0.079**	**0.081**	**0.279**
	FF-FB Time (sec.)	**41.139**	**42.820**	**41.780**

reveals that the proposed SGGSA has attained the best optimization results for all the plants except an insignificant lead for the GGSA in achieving less FF training time for Plant 1. More precisely, the SGGSA has achieved considerably fewer values for the cost functions of the FF and the FF-FB training phases for all the plants. On the other hand, in terms of processing time, the SGGSA has taken the shortest times compared to the other methods with the only exception of the tiny difference achieved by the GGSA in the FF training time of Plant 1. In the sequel, the results in table 1 signify that the modifications proposed to the original GSA have resulted in better optimization performance without causing an additional computational burden. Moreover, the cumbersome effort exerted on deciding the best settings for the two coefficients of the gravitational constant has been eliminated in the proposed algorithm.

5.4 *A comparative study with other neural network structures*

In this section, the control performance of the proposed MRWNN structure is compared with those of other neural network structures, namely the original SRWNN [16, 20], and the multilayer perceptron (MLP). Each of the above networks was employed in the FF and the FB loops of the control system shown in figure 1. The SGGSA was used to optimize the parameters of all the networks considered in this study. The wavelon layer in both the original SRWNN and the MRWNN consists of six wavelons. As the mother wavelet function, the wavelons in the SRWNN use the Mexican hat function, while those in the MRWNN use the RASP1 function. For the MLP, six nodes were used in the hidden layer. Each of these nodes employs the sigmoidal activation function. The plants considered in the previous section are used in this study. Following the same strategy adopted in the previous section to handle the stochastic discrepancy, 10 runs were performed to control each plant and the average result was taken. Table 2 displays the results of this comparative study.

From table 2, it can be observed that the MRWNN has achieved the fewest MSE values for Plants 1 and 2, while the SRWNN has achieved the fewest MSE value for Plant 3. On the other hand, compared to the other networks, the MRWNN has resulted in significantly fewer ISE values for all the plants, which clearly indicates the superiority of the MRWNN to act as the FF and the FB controllers in the control system. In terms of the processing time, the MRWNN has required the shortest times except for a small difference in time taken by the MLP in the FF training of Plant 1.

5.5 *A comparative study with other control structures*

To validate the effectiveness of the utilized control structure, the performance of the FF-FB control structure is compared with those of the PID controller and the MRWNN-based FF controller. Using the same plants considered before, all the above controllers are optimized by the SGGSA. To accomplish a reliable comparative study, 10 runs were made for each plant to take the average result. Due to the complex task represented by requesting a linear

Table 2. Comparison results of the MLP, the SRWNN, and the MRWNN acting as the controllers in the FF-FB control structure

Network type	Criterion (average of 10 runs)	Controlled plant		
		Plant 1	Plant 2	Plant 3
MLP	FF MSE	8×10^{-3}	5×10^{-3}	12×10^{-3}
	FF Time (sec.)	**27.815**	30.401	31.108
	FF-FB ISE	0.083	0.107	1.084
	FF-FB Time (sec.)	43.255	46.615	45.367
SRWNN	FF MSE	9×10^{-3}	2×10^{-3}	$\mathbf{5 \times 10^{-3}}$
	FF Time (sec.)	30.065	32.986	32.491
	FF-FB ISE	0.087	0.105	2.142
	FF-FB Time (sec.)	44.841	47.905	46.885
The proposed MRWNN	FF MSE	$\mathbf{7 \times 10^{-3}}$	$\mathbf{1 \times 10^{-3}}$	9×10^{-3}
	FF Time (sec.)	28.110	**29.541**	**28.981**
	FF-FB ISE	**0.079**	**0.081**	**0.279**
	FF-FB Time (sec.)	**41.139**	**42.820**	**41.780**

Table 3. Comparison results of the PID controller, the MRWNN-based FF controller, and the MRWNN-based FF-FB control structure.

Control structure	Criterion (average of 10 runs)	Controlled plant		
		Plant 1	Plant 2	Plant 3
PID controller	ISE	0.139	35.729	119.199
	Time (sec.)	**17.095**	**19.027**	**17.294**
MRWNN-based FF controller	ISE	0.102	0.608	27.213
	Time (sec.)	28.110	29.541	28.981
MRWNN-based FF-FB structure	ISE	**0.079**	**0.081**	**0.279**
	Time (sec.)	41.139	42.820	41.780

controller to handle highly nonlinear and time-variant systems, the maximum number of iterations was set to 1000 in the SGGSA to train the PID controller, while only 300 iterations were used for the other two control structures. The comparison results are shown in table 3.

The results in table 3 clearly highlight the superiority of the FF-FB control structure over the other structures, especially for Plants 2 and 3 for which the PID and the MRWNN-based FF controllers have resulted in unacceptable ISE values. Unsurprisingly, the FF-FB control structure took longer processing times since it uses two MRWNN controllers. However, this issue can be neglected in light of the superior control accuracy achieved by the FF-FB control structure for all the plants.

6. Conclusions

An integrated FF-FB control structure was presented in this paper to control nonlinear dynamical systems. In this intelligent control system, a modified structure of an RWNN, which was termed as the MRWNN, was employed in the FF and the FB loops. An enhanced version of the GSA, namely the SGGSA, was proposed to improve the searching capability of the original algorithm. In the SGGSA, two control parameters related to the gravitational constant in the original GSA were removed and the global best solution acquired thus far was exploited to form the next generation of agents. The SGGSA was used to optimize the parameters of both the FF and the FB controllers. Compared to other optimization techniques including the original GSA, the proposed algorithm has exhibited better optimization results. The results of several evaluation tests to control different nonlinear time-variant systems have demonstrated the effectiveness of the FF-FB control system with regard to control accuracy and robustness against external disturbances. From a comparative study with other neural network structures, the MRWNN has shown the best control performance. Moreover, another comparative study has clearly highlighted the superiority of the FF-FB control structure over the PID controller and the MRWNN-based FF controller.

Acknowledgements

The author would like to thank for the financial support provided by the University of Technology, Baghdad, Iraq, and the Ministry of Higher Education and Scientific Research, Iraq.

References

[1] Sahoo H K, Dash P K and Rath N P 2013 NARX model based nonlinear dynamic system identification using low complexity neural networks and robust H∞ filter. *Appl. Soft. Comput.* 13(7): 3324-3334

[2] Omatu S, Khalid M and Yusof R 1996 *Neuro-control and its applications*. Springer-Verlag London Ltd., Second Printing

[3] Zhu Z, Wang R and Li Y 2017 Evaluation of the control strategy for aeration energy reduction in a nutrient removing wastewater treatment plant based on the coupling of ASM1 to an aeration model. *Biochem. Eng. J.* 124: 44-53

[4] Beijen M A, Heertjes M F, Butler H and Steinbuch M 2016 Practical tuning guide to mixed feedback and feedforward control of soft-mounted vibration isolators. *IFAC-PapersOnLine* 49(21): 163-169

[5] Rollins D K, Mei Y, Loveland S D and Bhandari N 2016 Block-oriented feedforward control with demonstration to nonlinear parameterized Wiener modeling. *Chem. Eng. Res. Des.* 109: 397-404

[6] Yang Y -L, Wei Y -D, Lou J -Q, Fu L, Tian G and Wu M 2016 Hysteresis modeling and precision trajectory control for a new MFC micromanipulator. *Sensors Actuat. A-Phys.* 247: 37-52

[7] Lipiński K 2016 Modeling and control of a redundantly actuated variable mass 3RRR planar manipulator controlled by a model-based feedforward and a model-based-proportional-derivative feedforward–feedback controller. *Mechatronics* 37: 42-53

[8] Foo M, Kim J, Sawlekar R and Bates D G 2017 Design of an embedded inverse-feedforward biomolecular tracking controller for enzymatic reaction processes. *Comput. Chem. Eng.* 99: 145-157

[9] Wu J, Han Y, Xiong Z and Ding H 2017 Servo performance improvement through iterative tuning feedforward controller with disturbance compensator. *Int. J. Mach. Tool Manu.* 117: 1-10

[10] Glück T, Eder A and Kugi A 2013 Swing-up control of a triple pendulum on a cart with experimental validation. *Automatica* 49(3): 801-808

[11] Sabahi K, Ghaemi S and Pezeshki S 2014 Application of type-2 fuzzy logic system for load frequency control using feedback error learning approaches. *Appl. Soft Comput.* 21: 1-11

[12] Xie Y and Alleyne A 2014 Robust two degree-of-freedom control for MIMO system with both model and signal uncertainties. *IFAC Proceedings Volumes* 47(3): 9313-9320

[13] Li H, Guo C and Yin D 2011 Hybrid control model of IMWNN and SNLC based on WNNI for ship fin stabilizer. *International Conference of Soft* Computing *and Pattern Recognition*, Dalian, China, pp. 312-317

[14] Baruch I S, Escalante S F, Mariaca-Gaspar C R and Barrera-Cortes J 2007 Recurrent neural control of wastewater treatment bioprocess via marquardt learning. *IFAC Proceedings Volumes* 40(4): 289-294

[15] Sabahi K, Teshnehlab M and shoorhedeli M A 2009 Recurrent fuzzy neural network by using feedback error learning approaches for LFC in interconnected power system. *Energ. Convers. Manage.* 50(4): 938-946

[16] Lutfy O F 2017 Adaptive direct inverse control scheme utilizing a global best artificial bee colony to control nonlinear systems. *Arab. J. Sci. Eng.* 43(6): 2873-2888

[17] Huang H -X, Li J -C and Xiao C -L 2015 A proposed iteration optimization approach integrating backpropagation neural network with genetic algorithm. *Expert Syst. Appl.* 42(1): 146-155

[18] Rashedi E, Nezamabadi-Pour H and Saryazdi S 2009 GSA: A gravitational search algorithm. *Inform. Sciences* 179(13): 2232-2248

[19] Yin B, Guo Z, Liang Z and Yue X 2018 Improved gravitational search algorithm with crossover. *Comput. Electr. Eng.* 66: 505-516

[20] Lutfy O F and Dawood M H 2017 Model reference adaptive control based on a self-recurrent wavelet neural network utilizing micro artificial immune systems. *Al-Khwarizmi Engineering Journal* 13(1): 107-122

[21] Ramli N M, Hussain M A and Jan B M 2016 Multivariable control of a debutanizer column using equation based artificial neural network model inverse control strategies. *Neurocomputing* 194: 135-150

[22] Imtiaz U, Assadzadeh A, Jamuar S S and Sahu J N 2013 Bioreactor temperature profile controller using inverse neural network (INN) for production of ethanol. *J. Process Contr.* 23(5): 731-742

[23] Denaï M A, Palis F and Zeghbib A 2007 Modeling and control of non-linear systems using soft computing techniques. *Appl. Soft Comput.* 7(3): 728-738

[24] Zainuddin Z and Pauline O 2011 Modified wavelet neural network in function approximation and its application in prediction of time-series pollution data. *Appl. Soft Comput.* 11(8): 4866-4874

[25] Li L -L, Lin G -Q, Tseng M -L, Tan K and Lim M K 2018 A maximum power point tracking method for PV system with improved gravitational search algorithm. *Appl. Soft Comput.* 65: 333-348

[26] Wang M, Wan Y, Ye Z, Gao X and Lai X 2018 A band selection method for airborne hyperspectral image based on chaotic binary coded gravitational search algorithm. *Neurocomputing* 273: 57-67

[27] Choudhary A, Gupta I, Singh V and Jana P K 2018 A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing. *Future Gener. Comp. Sys.* 83: 14-26

[28] Mirjalili S and Lewis A 2014 Adaptive gbest-guided gravitational search algorithm. *Neural Comput. Appl.* 25(7-8): 1569-1584

[29] Darzi S, Kiong T S, Islam M T, Soleymanpour H R and Kibria S 2016 A memory-based gravitational search algorithm for enhancing minimum variance distortionless response beamforming. *Appl. Soft Comput.* 47: 103-118

[30] Bohat V K and Arya K V 2018 An effective gbest-guided gravitational search algorithm for real-parameter

optimization and its application in training of feedforward neural networks. *Knowl.-Based Syst.* 143: 192-207

[31] Chao Y, Dai M, Chen K, Chen P and Zhang Z 2016 A novel gravitational search algorithm for multilevel image segmentation and its application on semiconductor packages vision inspection. *Optik* 127(14): 5770-5782

[32] Ahmed M S 2000 Neural-net-based direct adaptive control for a class of nonlinear plants. *IEEE T. Automat. Contr.* 45(1): 119-124

[33] Sales K R and Billings S A 1990 Self-tuning control of non-linear ARMAX models. *Int. J. Contr.* 51(4): 753-769

[34] Shin Y C and Xu C 2009 *Intelligent systems: Modeling, optimization, and control*. CRC Press, Taylor & Francis Group, LLC, Boca Raton

[35] Hou Z and Jin S 2014 *Model free adaptive control: Theory and applications*. CRC Press, Taylor & Francis Group, LLC, Boca Raton