International Journal of Computational Intelligence in Control, 7(2), 2015

NONLINEAR FUZZY NEURAL CONTROLLER DESIGN VIA EM-BASED HYBRID ALGORITHM

Ching-Hung Lee and Yu-Chia Lee

Department of Mechanical Engineering, National Chung Hsing University, No. 250 Kuo Kuang Rd., Taichung, Taiwan, R.O.C.

ABSTRACT: In this paper, we propose a hybridization of electromagnetism-like (EM) algorithm and particle swarm optimization (PSO) method to design recurrent fuzzy neural systems for nonlinear control. The hybrid algorithm (called modified EMPSO) combines the advantages of EM and PSO algorithms to enhance the performance of optimization. The main modification from EM algorihtm is the random neghborhood local search is replaced by PSO algorithm with an instant update strategy. Each particle's velocity is updated instantaneously and it provides the best information for other particles. Thus, it enhances the convergence speed and the computational efficiency. Simulation results of nonlinear systems control and two-degree-of-freedom helicopter system are shown to illustrate the modified EMPSO has the ability of global optimization, faster convergence, and higher accuracy.

Keywords: Electromagnetism-like algorithm, hybrid, particle swarm optimization, fuzzy neural system, nonlinear control

I. INTRODUCTION

Fuzzy systems and neural networks are being used successfully in many application areas [4-8, 11, 12, 28-32]. Based on the approximation ability, many adaptive control techniques are accompanied with them for approximation of system functions or controllers. A recurrent fuzzy neural network (RFNN) system is proposed to identify and control nonlinear systems in literature [6, 8]. For temporal problems, the RFNN system is more suitable than feed-forward (static) system. Recently, some other recurrent fuzzy neural systems have been also proposed [4, 8-10, 29, 32, 33]. They have the ability of storing the past information of system. With the advantages, this study develops a RFNN-based control scheme for nonlinear systems via the proposed hybrid algorithm.

Many literatures have been proposed to deal with the designing and training of fuzzy neural systems [1-10]. For solving this problem, the gradient-descent method is widely used and it is a powerful technique [1, 6, 8, 11, 12]. It may rapidly obtain a local minimum and cannot find the global solution. Therefore, searching the global optimum of optimization problems and apply in the training of fuzzy neural systems are important. Recently, a novel global optimization algorithm that deduces from the electromagnetism theory in physics was presented, called electromagnetism-like mechanism algorithm (EM) [13-15]. It simulated the electromagnetism theory by considering each individuality (or candidate solution) to be an electrical charged. These charges move in the direction by the Coulomb's law and superposition principle, i.e., the force is directly proportional to the product of their particles and in inversely propositional to the distance between the charges. Thus, it has advantages of global optimization and multi-point searching [13-20]. However, the convergent speed of EM algorithm is slow when the local search procedure is absent [13]. On the other hand, the local search procedure can improve the convergence of EM. However, it is coordinate by coordinate, stochastically, and very complex in computational consideration. Thus, the highly computational complexity is the major drawback. Hence, a modified local search procedure by modified particle swarm optimization (PSO) is adopted to enhance the performance.

Corresponding author: E-mail: chleenchu@dragon.nchu.edu.tw

The PSO is easy to implement and has been empirically shown to perform well on many applications [21-27]. Each particle has a fitness value and a velocity to adjust its moving direction according to the best experiences of the swarm to search for the global optimum. Recently, a method of updating velocity strategy for PSO was proposed to obtain a better performance [23, 24]. In order to enhance the performance of EM and to accelerate its convergent speed, a modified update strategy in PSO is adopted for the recurrent fuzzy neural controller design.

This paper proposes a hybridization of EM algorithm and PSO, called modified EMPSO, to design the RFNN controller for nonlinear systems control. The modified EMPSO algorithm improves the optimization performance of EM and PSO algorithms. The major drawback of EM, computational complexity, is solven by replacing the randomly neghiborhood local search using PSO with an instant update strategy. Thus, the modified EMPSO combines the advantages of multi-point search, global optimization, and faster convergence. In addition, the modified EMPSO does not need any system gradient information. We use the modified EMPSO and RFNN to develop the controller scheme for solving nonlinear system control problems and tracking of 2-degree-of-freedom-helicopter system [34]. Simulation results show that the modified EMPSO has the ability of global optimization, advantages of faster convergence and higher accuracy.

The paper is organized as follows. Section II introduces the recurrent fuzzy neural network system. In Section III, the EM algorithm, PSO, and the proposed modified EMPSO algorithm are introduced. Section IV shows the simulation and comparison results of nonlinear system control and tracking control of 2-DOF-helicopter system. Simulations are shown and demonstrate the performance of the modified EMPSO. Finally, the conclusion is given in Section V.

II. RECURRENT FUZZY NEURAL SYSTEM

Many results have been obtained by using fuzzy neural systems for system identification and control [1, 5-8, 11]. These methods optimize the fuzzy neural systems by using learning algorithms to adjust the systems' parameters. However, their application fields are limited in static problems due to the static property. Hence, a recurrent fuzzy neural network (RFNN) having dynamic fuzzy reasoning is proposed for solving temporal problems [6, 8]. With dynamic fuzzy reasoning, the RFNN is more effective than the conventional fuzzy neural systems [4, 7, 8, 24, 28-30]. Therefore, we here use the RFNN to develop a control scheme via the modified EMPSO for nonlinear systems control.

Figure 1 shows the diagram of the RFNN system, where G represents the Gaussian membership function. The RFNN has n input variables, m term nodes for each input variable, p output nodes, and $m\times n$ rule nodes. Therefore, a RFNN system with m rules consists of $n+(n\times m)+m+p$ nodes. The RFNN system is introduced briefly here. Layer 1 accepts input variables. Its nodes represent input linguistic variables. The nodes in this layer only transmit input variables to the next layer directly, i.e., $O_i^1 = u_i^1$. Layer 2 is used to calculate Gaussian membership value, i.e., $O_{ij}^2 = \exp\left[-\frac{(u_{ij}^2 - m_{ij})^2}{(\sigma_{ij})^2}\right]$, where m_{ij} and σ_{ij} are the center and the width of the Gaussian function, $u_{ij}^2(k) = O_i^1(k) + O_{ij}^2(k-1)\cdot\theta_{ij}$. Nodes in this layer represent the terms of the respective linguistic variables. Nodes at layer 3 represent fuzzy rules. Layer 3 forms the fuzzy rule base. Links before layer 3 represent the preconditions of the rules, and the links after layer 3 represent the consequences of the rule nodes, i.e., $O_i^3 = \prod_j O_{ij}^2$. Layer 4 is the output layer. Each node is for actual output to be pumped out this system. The links between layer 3 and layer 4 are connected

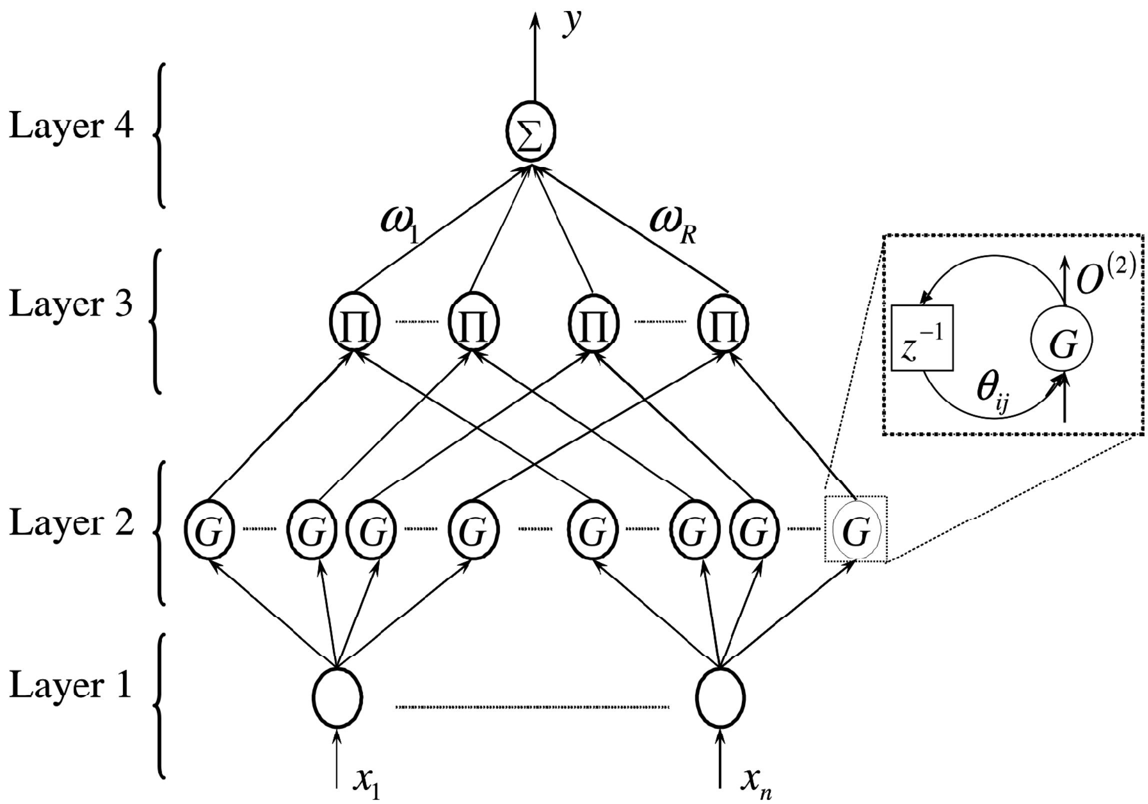


Figure 1: Network Diagram of RFNN System [6, 8]

by the weighting value ω_p^i, i.e., $y_j = O_j^4 = \sum_i^m O_{ij}^3 \omega_i^4$. As the previous statement, the RFNN has adjustable parameters m, σ, θ, and ω, which is denoted by $W = [m, \sigma, \theta, \omega]^T$.

To realize dynamic fuzzy rules, the RFNN system is inherently a recurrent multilayered connectionist network. By adding feedback connections in the second layer of the RFNN system, temporal relations are embedded in the network which is used to memorize past information. For a RFNN system with n inputs $x_1, x_2, \ldots, x_n$ and one output y, each dynamic fuzzy if-then rule in RFNN is in the form of

$$\text{Rule } j\text{: IF } z_1 \text{ is } A_{1j} \text{ and } \ldots\ z_n \text{ is } A_{nj}, \text{ THEN } y \text{ is } \omega_j \tag{1}$$

where z_j is $O_{ij}^{(2)}(k-1)\cdot\theta_{ij} + O_j^{(1)}(k)$ which includes the past information and the current input; A_{1j} is a fuzzy set and for inference output y, w_j is the consequent part parameter.

As discussion of our previous result [6, 8], the RFNN has the ability of universal approximation, i.e., the RFNN can be used to identify a nonlinear dynamic system from the system input and output signals. In addition, the RFNN has a smaller network structure and a smaller number of tuning parameters than that of the fuzzy neural systems [6, 8]. The RFNN has the ability of storing system past information due to the present of feedback layer. It also increases the learning speed of RFNN [6, 8].

III. HYBRIDIZATION OF ELECTROMAGNETISM-LIKE AND PARTICLE SWARM OPTIMIZATION ALGORITHMS

This section introduces the proposed hybrid learning algorithm, modified EMPSO, for designing the RFNN controller. The modified EMPSO combines the advantages of EM and PSO algorithms to result faster convergence and higher accuracy. Figure 2 depicts the flow chart of the proposed modified EMPSO algorithm. At first, the optimization problem should be defined, for design of RFNN controller, the controller parameters $W = [m, \sigma, \theta, \omega]^T$ is represented to be a particle. For nonlinear systems control problem, the tracking error is the difference between the reference trajectory and system actual output, i.e., $e(t) = y_r(t) - y_p(t)$. The root-mean-square-error (RMSE) of tracking error is adopted to be the objective function.

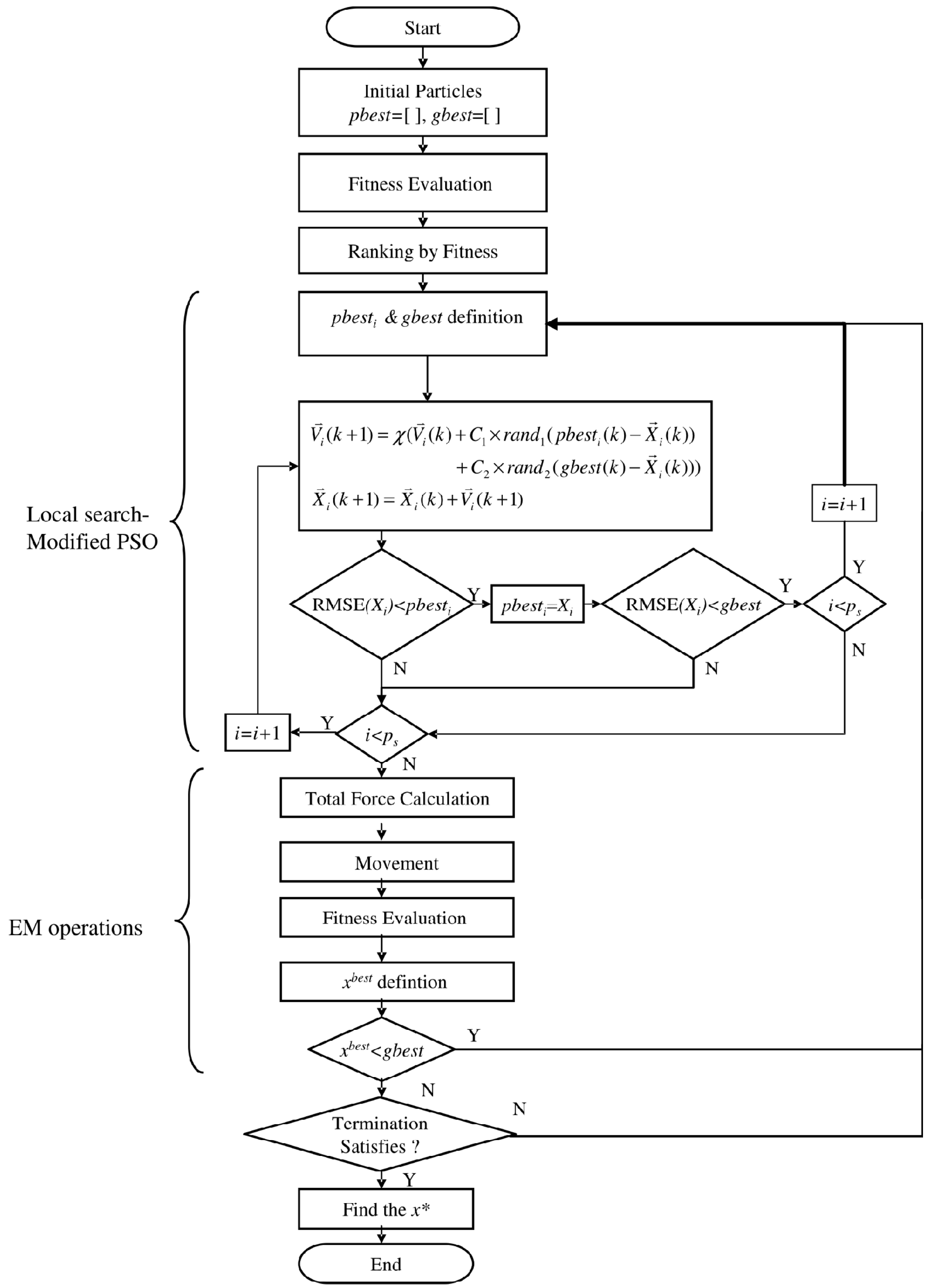


Figure 2: Description of the Modified EMPSO Algorithm

The initial particles are randomly selected from the searching space and its initial position and velocity should be set. After initialization, evaluation phase should be done. Each particle is evaluated and ranked by the corresponding RMSE of tracking error. The particle having the smallest RMSE value is selected to be *gbest* for PSO and x^{best} for EM. Each particle's information is updated by using the historical best information. In order to make better use of the beneficial information and enhance the convergent speed, we use the instant update strategy that all charges and particles are updating its velocity instantly. Each particle can update its individual information one by one and produce new best particle. When the new particle is defined, it should be determined whether it is better than the *gbest* or not. When the new *gbest* is produced, it is used to provide the information for next particle. Every particle gets the newest information to update the velocity by this strategy.

Subsequently, the EM-operation phase is used to optimize the RFNN system. By the technique of instant update strategy, all particles are evaluated to determine whether *gbest* is replaced or not. The particle with smallest RMSE value is defined to be x^{best}. If x^{best} is better than *gbest*, it would become the new *gbest*. The procedure will be stopped until the stop criterion (maximum generations) is satisfied. Detailed descriptions for the EM, PSO, and the modified EMPSO are introduced as below.

(A) Optimization Problem Definition for EM

In EM algorithm, each candidate solution is viewed as a charged particle [13-16]. The EM for optimization problems can be represented as

$$\begin{aligned}&\text{Minimize } f(x)\\&\text{subject to } \{x \in \Re^n \,|\, l_k \le x_k \le u_k, l_k, u_k \in \Re, k = 1, \ldots, n\}\end{aligned} \tag{2}$$

where n is the problem dimension, $f(x)$ is the objective function, and u_k and l_k are the corresponding upper bound and lower bound of parameter x_k. Each particle x represents a candidate solution with charge. EM utilizes the mechanisms of attraction and repulsion to determine the searching direction. The magnitude of force is calculated by the Coulomb's law.

(B) Initialization

For many applications, the real-value coding technique is used to represent a solution. In this study, each particle denotes a weighting vector $[m, \sigma, \theta, \omega]^T$ shown in Fig. 3, and the modified EMPSO is utilized to find the optimal value $[m^*, \sigma^*, \theta^*, \omega^*]^T$. At first, a proper population size is selected and the initial particles are randomly chosen from the searching space. The feasible region of RFNN parameters should be defined (i.e., u_k and l_k for m, σ, θ and w). Each *pbest* and *gbest* are set to be null (denoted by [ ]) at first. In addition, the stop criterion- maximum generations, is selected in this phase.

(C) Evaluation and Ranking

This phase is used to calculate the fitness values of entire particles. Each particle is evaluated by the given RMSE of tracking error to decide its survival or extinction in the next generation. This helps us to find superior particles, i.e., a particle having smaller RMSE value has a higher probability of survival. Three steps are done in this phase: *fitness values evaluation*, *ranking by*

m_{11}	m_{12}	…	σ_{11}	σ_{12}	…	θ_{11}	θ_{12}	…	ω_1	ω_2	…

Figure 3: Particle Representation of RFNN

fitness value, and *best particle definition*. To evaluate the performance of each particle in training the RFNN system, we define the RMSE as

$$f(x) \equiv \sqrt{\sum_{k=1}^{T} e^2(k)/T} \tag{3}$$

where e denotes the tracking error and T denotes the data number. Subsequently, all particles are ranked and indexed by their corresponding RMSE values. Finally, the particle having the minimum RMSE is stored in *gbest*, i.e., *gbest* = $\{x \mid \text{min RMSE of } e\}$.

(D) Local Search for the Modified EMPSO Algorithm

The local search phase is used to gather the local information for each particle x^j. Details of the local search procedure for EM algorithm can be found in literature [13-17]. As description in [17], for the case of RFNN controller design using EM algorithm, if M particles with N parameters are chosen, the random search needs at least $N \times M$ times of RMSE evaluation in a local search procedure. In order to reduce the computational complexity, we propose the modified EMPSO to enhance the performance.

Figure 4 shows a flow chart of modified PSO with an instant update velocity strategy. After evaluation phase, each particle updates its position and velocity in the local search procedure. In every searching-iteration, there are four attributes for particles in the search space to present their features: current position $\vec{X}_i$, current velocity $\vec{V}_i$, past best position $pbest_i$, and global best position *gbest*. Firstly, every particle updates its information by

$$\begin{aligned} &\vec{V}_i(k+1) = \chi(\vec{V}_i(k)) + C_1 \cdot rand_1(pbest_i(k) - \vec{X}_i(k)) + C_2 \cdot rand_2(gbest(k) - \vec{X}_i(k)) \\ &\vec{X}_i(k+1) = \vec{X}_i(k) + \vec{V}_i(k+1). \end{aligned} \tag{4}$$

If current position $\vec{X}_i$ has smaller RMSE value than the past best result $pbest_i$, it should be replaced. After replacing the $pbest_i$, we have to confirm whether $\vec{X}_i$ is better than the *gbest* or not. Conventional method of updating the best group particle is executed in the last of every generation while all particles have operated. Unlike this, the method of updating *gbest* here is updated instantaneously one by one, not until all particles have operated in a generation. Detailed flow chart of local search by PSO with an instant update strategy can be found in Fig. 4. Using the above modifications, the comparison results between the modified EMPSO and other algorithms (EMPSO, EM, PSO, and GA) are shown in Section IV.

(E) EM Operation - Total Force Calculation

As description of literature [13-16], there are three steps in the EM operation phase. They are “local search,” “total force calculation”, and “movement”, respectively. As above description, the local search is done by the PSO with an instant update strategy.

In this step, a charge is assigned to each particle of the population. The charge q^i of particle x^i is determined by

$$q^i = \exp\left[-n \times \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m} [f(x^k) - f(x^{best})]} \right], \qquad i = 1, 2, \ldots m \; . \tag{5}$$

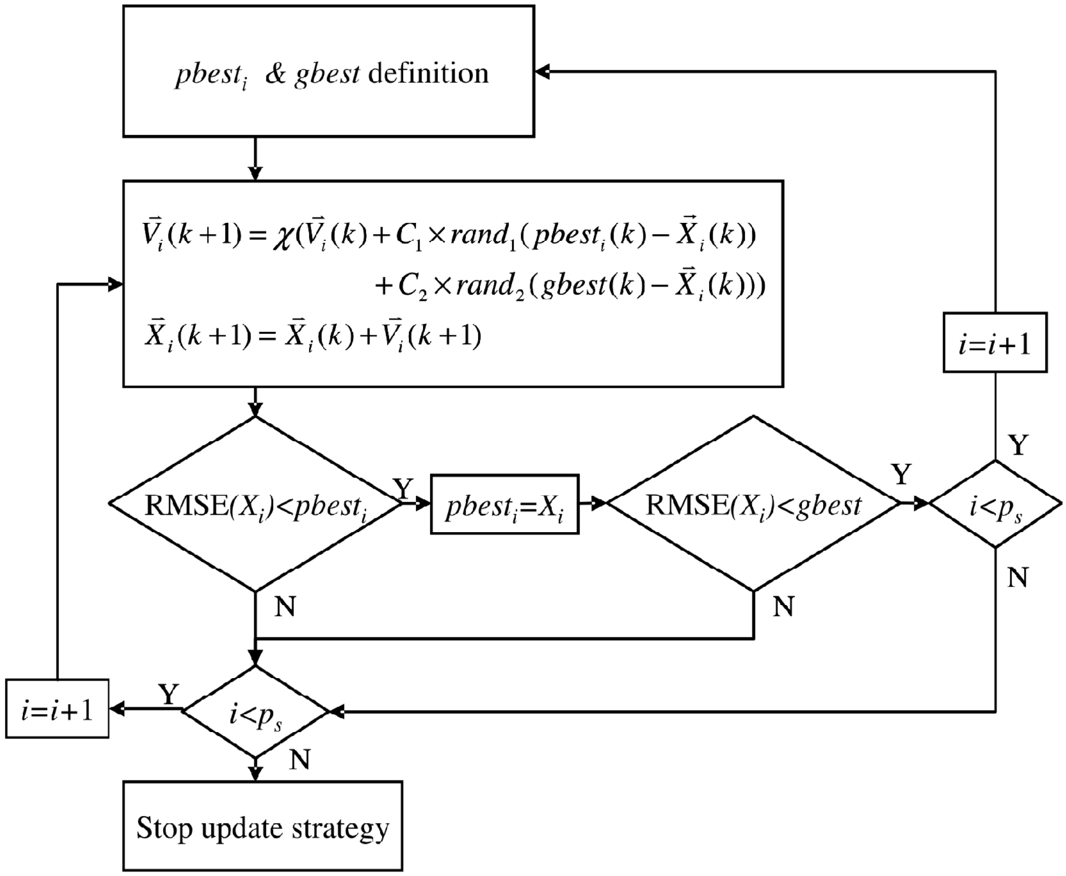


Figure 4: Flow Description of Local Search for Modified EMPSO Algorithm with an Instant Update Particle's Velocity Strategy

As in the electromagnetic theory, the force is inversely propositional to the distance between two charges and directly proportional to the product of their charges. Hence, the total force on x^i computed by the superposition principle is

$$F^i = \begin{cases} \sum_{j\neq i}^{m} (x^j - x^i)\cdot \dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2} & \text{if} \quad f(x^j) < f(x^i) \\ \sum_{j\neq i}^{m} -(x^j - x^i)\cdot \dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2} & \text{if} \quad f(x^j) \geq f(x^i). \end{cases} \tag{6}$$

After comparing the RMSE values, the direction of the forces between the particle and the others is selected. The one with a better (smaller) RMSE value attracts the other particles and the particle with larger RMSE repels others. Therefore, x^{best} (or *gbest*) plays the role of an attractor.

F. EM Operation - Movement

After determining F^i, particle x^i moves in the direction by a random step length

$$x^i = \begin{cases} x^i + \lambda \dfrac{F^i}{\left\|F^i\right\|}(u_k - x_k^i) & \text{if} \quad F_k^i > 0 \\ x^i + \lambda \dfrac{F^i}{\left\|F^i\right\|}(x_k^i - l_k) & \text{if} \quad F_k^i \leq 0 \end{cases} \qquad k = 1,\ 2,\ \ldots,\ n,\ \ i = 1,\ 2,\ \ldots,\ m \;, \tag{7}$$

where λ = random(0, 1) is the random step length.

Subsequently, all particles are evaluated and the particle with the smallest RMSE value is defined as x^{best}. The *gbest* would be replaced again if x^{best} is better than *gbest*. Detailed procedure can be viewed in Figure 2.

IV. SIMULATION RESULTS

To show the performance of the modified EMPSO, two illustrated examples for nonlinear control systems using RFNN are presented. All simulation results were done by MATLAB in Intel Pentium 4 computer with clock rate of 3-GHz and 2G MB of main memory.

Example 1: Single-input-single-output system

Consider the tracking control of single-input-single-output nonlinear system [29]

$$y_p(k+1) = \frac{y_p(k)\cdot y_p(k-1)\cdot(y_p(k)+2.5)}{1+y_p^2(k)+y_p^2(k-1)} + u. \tag{8}$$

The reference trajectory is

$$y_r(k+1) = \begin{cases} 10, & \text{if } k \le 50 \text{ or } 100 < k \le 150 \\ 15, & \text{if } 50 < k \le 100 \text{ or } 150 < k \le 200. \end{cases} \tag{9}$$

Figure 5 shows the system control scheme using the modified EMPSO and RFNN.The RFNN system is used to play the role of off-line controller. The inputs of RFNN controller are plant past state variable $y_p(k\text{-}1)$ and the current reference trajectory $y_r(k)$, and the output is the current control signal $u(k)$ which generates a proper control force to derive the system to follow the reference trajectory y_r. Thus, the corresponding RMSE function of tracking error is defined

$$\text{RMSE} = \left(\sum_{k=1}^{200}(y_p(k+1)-y_r(k+1))^2/200\right)^{1/2}. \tag{10}$$

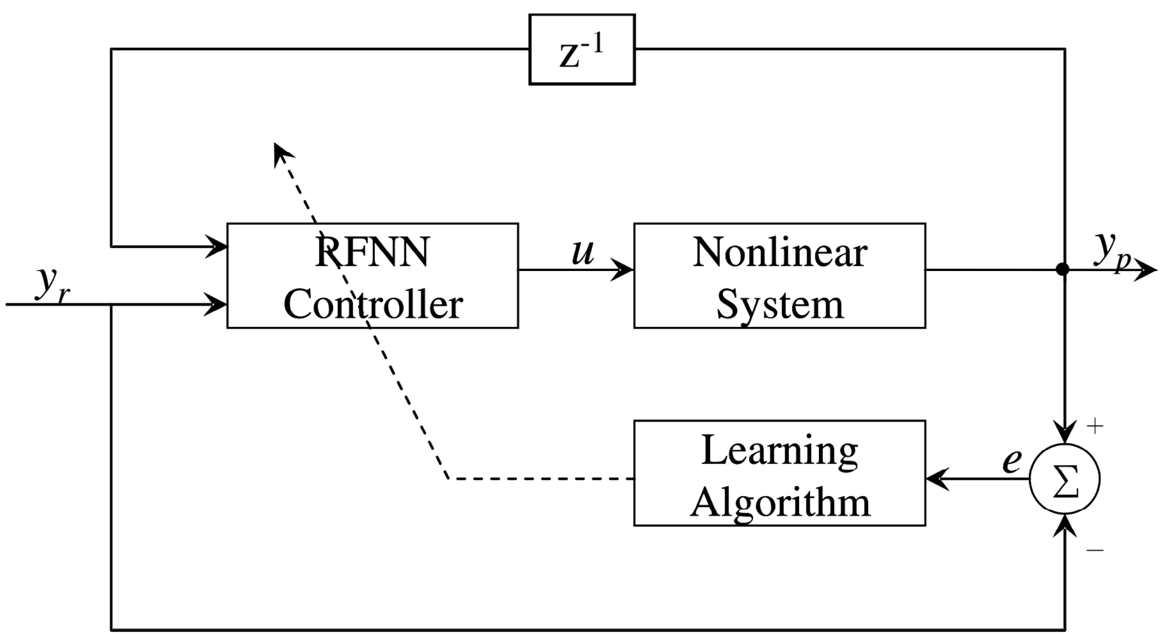


Figure 5: Dynamic System Control Configuration with RFNN Controller

To show the efficiency and effectiveness of the modified EMPSO, we have the comparison results with other multi-point algorithms, EM, PSO, EMPSO, and GA. Herein, we briefly introduce these used multi-point algorithms. (a) EM algorithm: as the above introduced in Section III. The local search is done coordinate by coordinate randomly [13-16]. (b) PSO algorithm: Each particle updates its information by the past best information and group best information, i.e., equation (4). (c) EMPSO algorithm [17]: hybrid algorithm of EM and PSO in which the local search operation of EM is replaced by traditional PSO technique. (d) GA algorithm: the individual updates by produce, cross, mutation operations. In this simulation, the stop criterion is chosen as the maximal generations to be 50 and the population size is set to be 30. Other parameters setting for algorithms are shown in Table 1. In addition, a RFNN system with 5 rules is chosen and the RFNN's initial parameters m, σ, θ, ω are chosen randomly between [-1, 1]. Hence, the total parameters number is 35.

Figures 6-7 and Table 2 show the simulation results of Example 1. Figure 6 shows the system trajectory after 50 generations (solid line: desired trajectory; dashed line: system actual output). It can be found that the RFNN controller performs well with less tracking error even the reference trajectory (set-point) is discontinuous. The comparison results of RMSE between the modified EMPSO and other algorithms as shown in Fig. 7 (solid green line: GA, solid pink line: PSO, solid black line: EM, solid blue line: EMPSO [17], and dashed line: modified EMPSO). Obviously, the modified EMPSO algorithm has better performance in RMSE value than others.

Table 1
Parameters Setting of Algorithms for Example 1

Algorithms	*Parameter*
Modified EMPSO	$C_1 = C_2 = 2$; $u_k = 1$, $l_k = -1$, c = 1
EMPSO	$C_1 = C_2 = 2$; $u_k = 1$, $l_k = -1$, c = 1
EM	$u_k = 1$, $l_k = -1$
PSO	$C_1 = C_2 = 2$;
GA	copulation rate: 0.8; mutation rate: 0.1

Table 2
Comparison Results of RMSE for Example 1

Algorithm	*Best RMSE*	*Worst RMSE*	*Mean RMSE*
Modified EMPSO	0.1996	0.2572	0.2052
EMPSO	0.2398	0.3663	0.3021
EM	0.2844	0.3924	0.3366
PSO	0.3468	0.6156	0.4658
GA	0.7828	2.8918	1.8363

The comparison results of RMSE in independent 20 runs are shown in Table 2. From Table 2, the modified EMPSO algorithm has the best control performance (Best RMSE: 0.1996). We can also observe that the best, worst, and mean cases of the modified EMPSO are smaller than others. Besides, the corresponding computational effort of EM and the modified EMPSO are 2582.235 and 67.341 seconds. The modified EMPSO does reduce the computational complexity of EM. However, compared with PSO and GA, the function evaluation operation of each particle modified EMPSO should be done twice in one generation. This means that the computational effort is more complex than PSO and GA. From above, the effect of instant update strategy for optimization can be observed. Consequently, we can conclude that the modified EMPSO reduces the

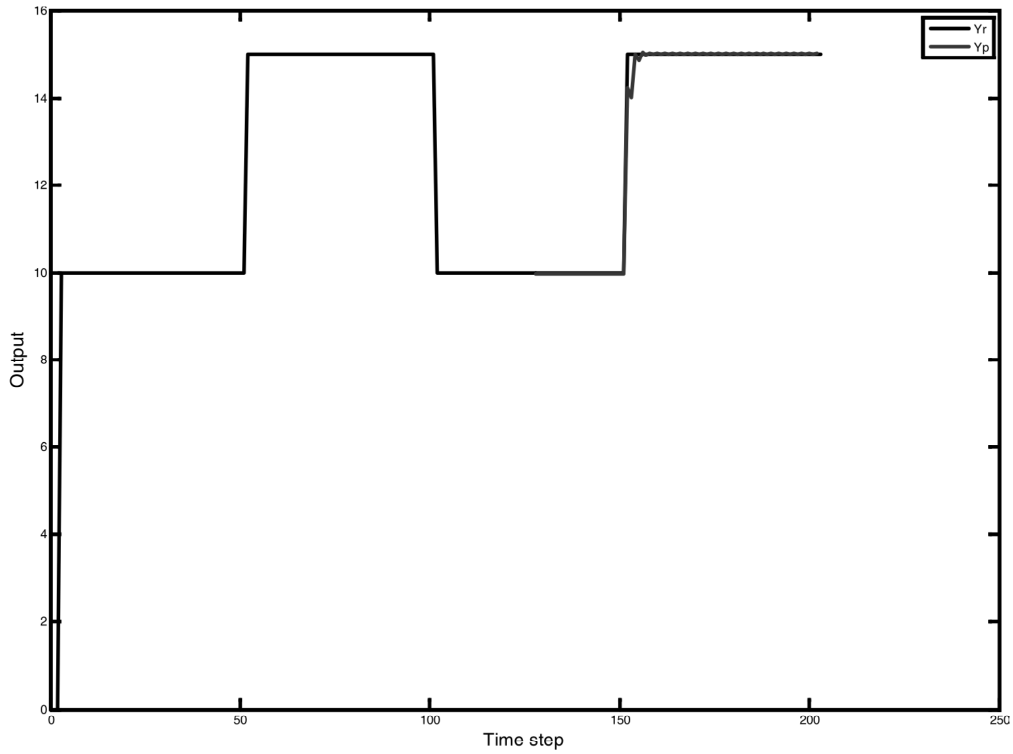


Figure 6: System Trajectories after 50 Training Generations of Example 1: (Solid line: Desired Trajectory; Dashed Line: System Actual Output)

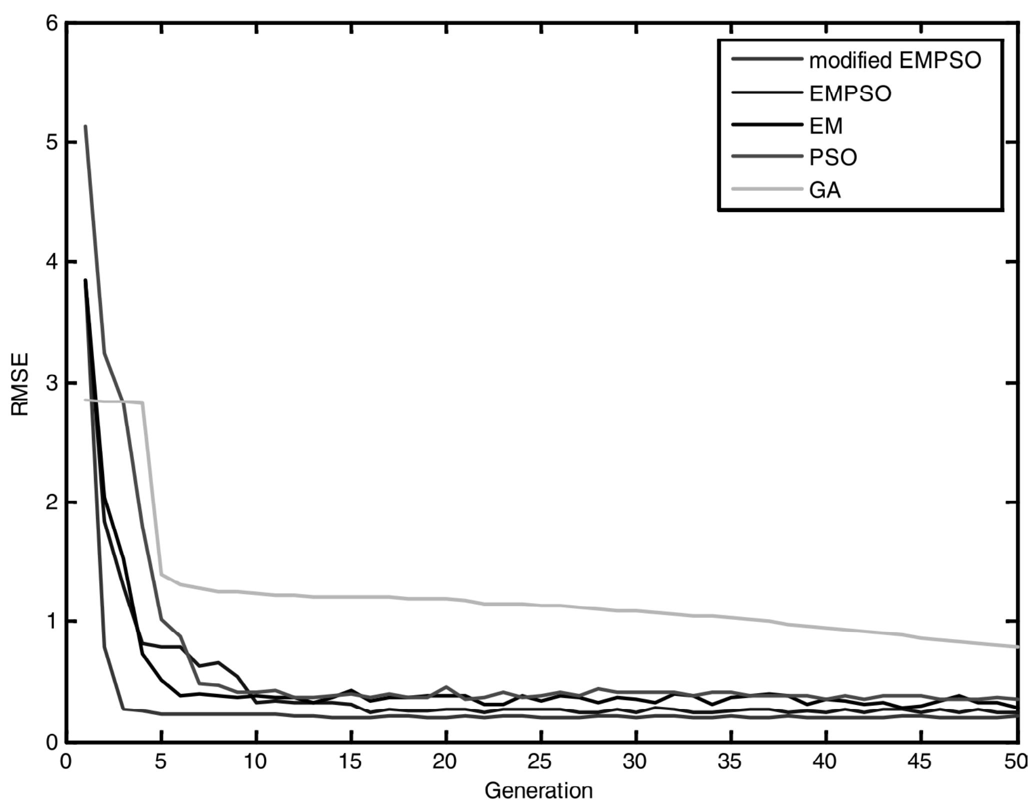


Figure 7: Comparison Results of Tracking Error in RMSE for Example 1: (Solid Green Line: GA, Solid Pink Line: PSO, Solid Black Line: EM, Solid Blue Line: EMPSO, and Dashed Line: Modified EMPSO)

computational complexity of multi-point optimization and has the ability of global optimization, advantages of faster convergence and higher accuracy.

Example 2: Tracking Control of 2-DOF-Helicopter System

To demonstrate the effectiveness of the proposed control scheme, consider the tracking control of the 2-DOF-helicopter system. The helicopter model is equipped with two dc motors which actuate two propellers directed in such way that the front (main) propeller controls the pitch motion, while the tail propeller controls the yaw motion. The inputs of the plant are the voltages $\mathbf{u} = [u_{\text{main}} \;\; u_{\text{tail}}]^T$ applied to the two motors. Also the pitch angle q_1 and the yaw angle q_2 are measurable. The relationship between voltages applied to the motors and the generalized torques produced along each axis of rotation is modeled by a static mapping of the form $\tau = \mathbf{B}(\mathbf{q})\,\mathbf{u}$.

The model of full-actuated Euler-Lagrange system is given by

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q,\dot{q})\dot{q} + \mathbf{g}(q) = \tau \tag{11}$$

where

$$\mathbf{M}(q) = \begin{bmatrix} \vartheta_1 & 0 \\ 0 & \vartheta_2 + \vartheta_3 \cos^2 q_1 \end{bmatrix},\; \mathbf{B}(q) = \begin{bmatrix} -\vartheta_6 & \vartheta_7 \\ -\vartheta_8 \cos q_1 & -\vartheta_9 \cos q_1 \end{bmatrix},$$

$$\mathbf{g}(q) = \begin{bmatrix} -\vartheta_4 \cos q_1 + \vartheta_5 \sin q_1 \\ 0 \end{bmatrix},\; \mathbf{C}(q,\dot{q}) = \begin{bmatrix} 0 & \frac{1}{2}\vartheta_3 \sin(2q_1)\dot{q}_2 \\ -\frac{1}{2}\vartheta_3 \sin(2q_1)\dot{q}_2 & -\frac{1}{2}\vartheta_3 \sin(2q_1)\dot{q}_1 \end{bmatrix}.$$

Let $x_1 = q$, $x_2 = \dot{q}$, and $\mathbf{q} = [q_1 \;\; q_2]^T$, the state-space representation is

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 &= -[\mathbf{M}(x_1)^{-1}]\mathbf{C}(\mathbf{x}_1,\mathbf{x}_2)\mathbf{x}_2 - [\mathbf{M}(x_1)^{-1}]\mathbf{g}(\mathbf{x}_1) + [\mathbf{M}(x_1)^{-1}]\tau + \mathbf{D} \end{aligned} \tag{12}$$

where $\mathbf{x}_1 = [q_1 \; q_2]^T, \mathbf{x}_2 = [\dot{q}_1 \; \dot{q}_2]^T, \mathbf{u} = [u_1 \; u_2]^T,\; \tau = \mathbf{B}(\mathbf{q})\mathbf{u},$ and $\mathbf{D} = [d_1 \; d_2]^T$. The control objective is to design the control signals $\mathbf{u}$ such that the system output **x1** track a bounded reference trajectory $\mathbf{y}_r(t)$ ($\mathbf{y}_r = [y_{r1}, y_{r2}]$) asymptotically. The desired trajectories are chosen as $y_{r1} = \frac{\pi}{4}\sin(t)$ and $y_{r2} = \frac{\pi}{4}\cos(t)$. The sampling time is selected to be 0.01 second and the external disturbance is $\mathbf{D} = \left[0.5\sin\left(t + \frac{\pi}{6}\right) \cdot \sin 1.5t \sin 2t \cdot \sin\left(2t - \frac{\pi}{6}\right)\right]^T$. The initial conditions is chosen as $[x_1(0),\, x_2(0),\, x_3(0),\, x_4(0)] = \left[\frac{\pi}{18},\; 0,\; -\frac{\pi}{18},\; 0\right]$. The system parameters are chosen as previous literature shown in Table 3 [34]. The RFNN-based control scheme is shown in Fig. 8, and there are two RFNN controllers to generate proper signals u_{main} (u_1) and u_{tail}(u_2). The inputs of the RFNNs are the current tracking errors.

To evaluate the performance, we define the error function and the RMSE function as

Table 3
Plant Parameters of 2-DOF-helicopter System [34]

Parameter	*Value*	*Units*	*Parameter*	*Value*	*Units*
ϑ_1	0.0826	kgm^2	ϑ_6	0.0566	Nm/V
ϑ_2	0.0041	kgm^2	ϑ_7	0.0054	Nm/V
ϑ_3	0.0349	kgm^2	ϑ_8	0.0042	Nm/V
ϑ_4	0.1610	kgm^2/s^2	ϑ_9	0.0114	Nm/V
ϑ_5	0.4080	kgm^2/s^2			

$$\text{RMSE} = \left(\sum_{k=1}^{T}\left[e_1^2(k)+e_2^2(k)\right]/T\right)^{1/2}. \tag{13}$$

The following parameters for the modified EMPSO algorithm are chosen

– Maximum generation: 50

– Population size: 30

– Simulation time: 10 seconds

– Control constant: 1

– $C_1 = C_2$: 2

The RFNN's initial parameters m, σ, θ, ω are chosen randomly between [-1, 1] and the network structure is

– Rule number of RFNNs: 5

– Network structure of RFNNs: 2-10-5-1

– Parameter number of RFNNs: 35

After 50 training generations, simulation results are shown in Figs. 9-10. Figure 9(a) shows the reference trajectories and the actual system outputs of 2-DOF-helicopter system using the RFNN-based control scheme via the modified EMPSO (solid line: desired trajectories y_{r1} and y_{r2}; dashed line: the system actual outputs pitch angle q_1 and yaw angle q_2). The corresponding control effort for u_1 and u_2 are shown in Fig. 9(b). It can be observed that the RFNN-based control scheme via the modified EMPSO achieves the tracking control problem such that the system outputs follow the reference trajectories even the time varying external disturbance exists. Comparison results of RMSE between the modified EMPSO and other algorithms are shown in Table 4 and Fig. 10 (solid green line: GA, solid pink line: PSO, solid black line: EM, solid blue line: EMPSO and dashed line: modified EMPSO). As shown in Fig 10, we can observe that the RFNN-based

Table 4
Comparison Results of RMSE for Example 2

Algorithm	*Best RMSE*	*Worst RMSE*	*Mean RMSE*
Modified EMPSO	0.0796	0.257	0.205
EMPSO	0.1361	0.366	0.302
EM	0.1646	0.392	0.336
PSO	0.2034	0.615	0.465
GA	2.945	4.891	3.918

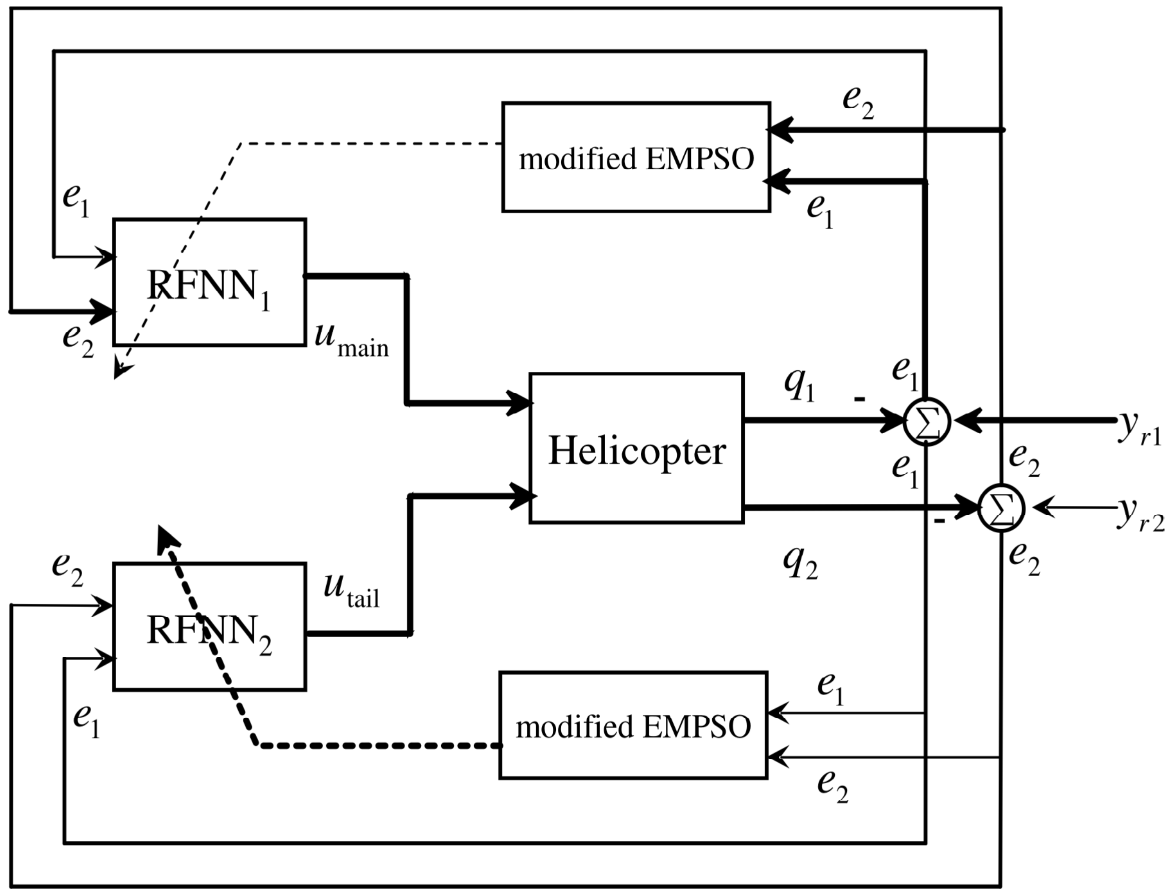
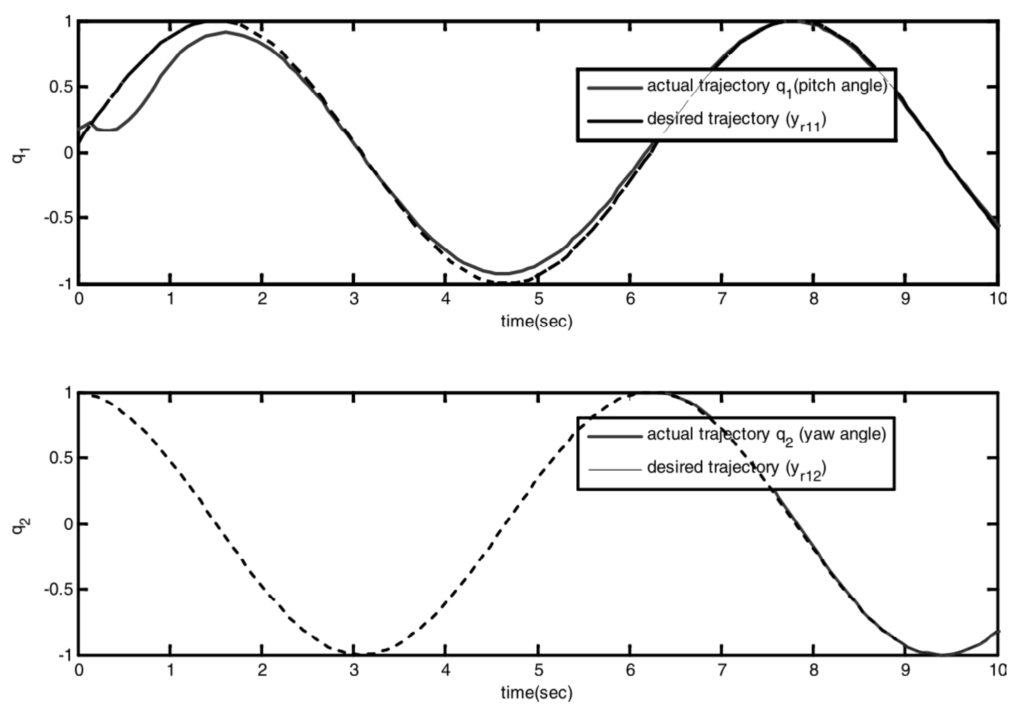
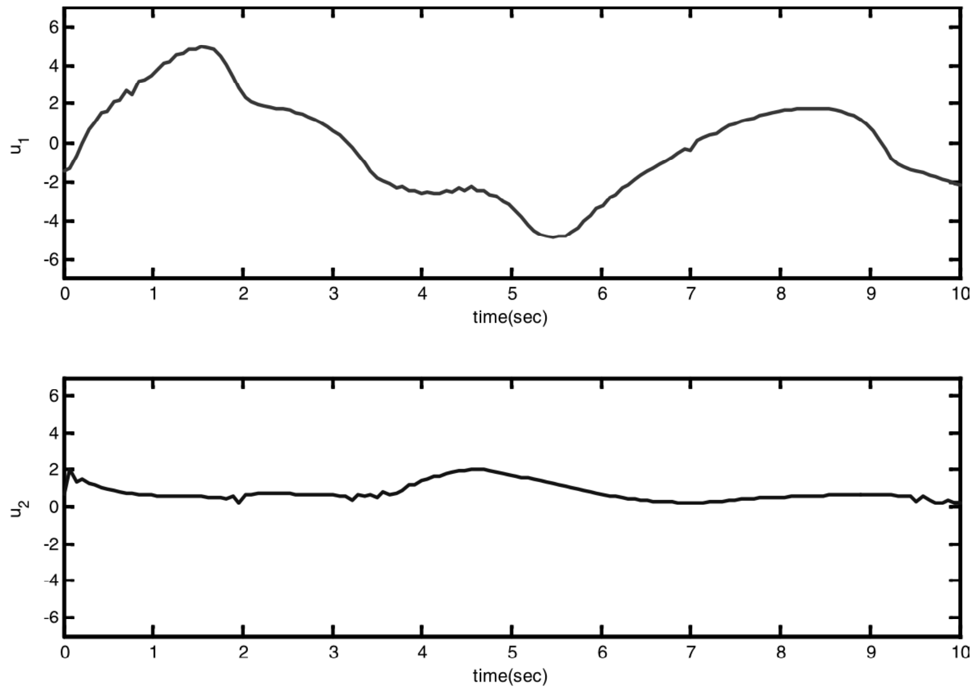


Figure 8: The Modified EMPSO-based RFNN Control Scheme for the two-DOF-helicopter System



(a)



(b)

Figure 9: Tracking Results of the Modified EMPSO-based RFNN Control Scheme for Example 2, (a) System Trajectories (Dotted Line: Desired Trajectories y_{r1} and y_{r2}; Dashed Line: the System Actual Outputs Pitch Angle q_1 and yaw angle q_2); (b) Control effort

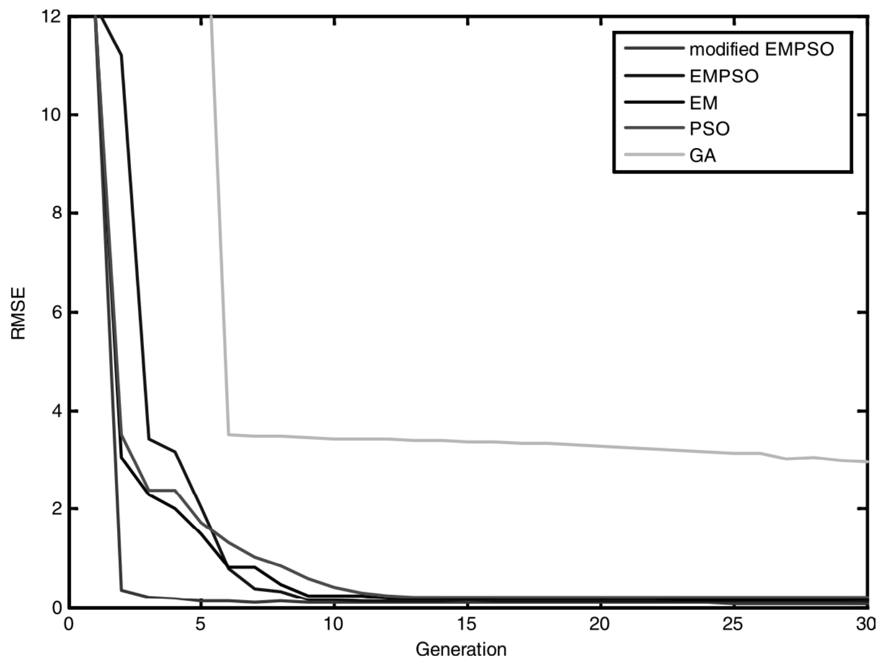


Figure 10: Tracking Error Comparison of Example 2: (Solid Green Line: GA, Solid Pink Line: PSO, Solid Black Line: EM, Solid Blue Line EMPSO and Dashed Line: Modified EMPSO)

control scheme via the modified EMPSO algorithm can converge faster than other algorithms and tracks the reference trajectories accurately. In addition, from Table 4, the modified EMPSO algorithm has the best control result (Best RMSE: 0.0796). It is obviously that the best, worst, and mean cases of the modified EMPSO are smaller than EMPSO, EM, PSO, and GA.

Furthermore, the proposed modified EMPSO algorithm does improve the control performance and has higher accuracy that results smaller RMSE and has faster convergence than the other algorithms.

V. CONCLUSION

In this paper, a hybrid learning algorithm- modified EMPSO with an instant update strategy has been proposed for the RFNN-based controller design. The modified EMPSO combines the advantages of EM and PSO algorithms to optain the properties of multipoint search, global optimization, and faster convergence. The random neighborhood local search of EM algorithm is replaced by PSO algorithm with an instant update strategy. Each particle's velocity is updated instantaneously and it provides the best information for other particles. Thus, it enhances the convergence speed and the computational efficiency. In addition, the modified EMPSO was used to design the RFNN controller parameters such that the nonlinear system output follows the reference trajectory. Illustration examples including the nonlinear system control and tracking of 2-DOF helicopter system were presented to show that the modified EMPSO has the ability of global optimization and effectiveness of higher accuracy and faster convergence.

Acknowledgement

The authors would like to thank the Guest Editor and anonymous reviewers for their insightful comments and valuable suggestions.This work was partially supported by the National Science Council, Taiwan, R.O.C., under contract No: NSC-97-2221-E-155-033- MY3.

References

[1] S. I. Birbil and S. C. Fang, "An Electromagnetism-like Mechanism for Global Optimization," *Journal of Global Optimization*, **25**(3), 263-282, 2003.

[2] S. I. Birbil, S. C. Fang, and R. L. Sheu, "On the Convergence of a Population-based Global Optimization Algorithm," *Journal of Global Optimization*, **30**(2), 301-318, 2004.

[3] S. I. Birbil and O. Feyzioglu, "A Global Optimization Method for Solving Fuzzy Relation Equations," *Lecture Notes in Artificial Intelligence*, **2715**, 718-724, 2003.

[4] P. C. Chang, S. H. Chen, and C. Y. Fan, "A Hybrid Electromagnetism-like Algorithm in Single Machine Scheduling Problems with Learning Effect," *Expert Systems with Applications*, **39**(3), 1259-1267, 2008.

[5] Y. C. Chen and C. C. Teng, "A Model Reference Control Structure using a Fuzzy Neural Network," *Fuzzy Sets and Systems*, **73**(3), 291-312, 1995.

[6] R. C. Eberhart and J. Kennedy, "A New Optimizer using Particle Swarm Theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, 39-43.

[7] W. A. Farag, V. H. Quintana, and L. T. Germano, "A Genetic-based Neuro-fuzzy Approach for Modeling and Control of Dynamical Systems," *IEEE Trans. on Neural Networks*, **9**(5), 756-767, 1998.

[8] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithm for Neural Networks," *IEEE Swarm Intelligence Symp.*, pp. 110-117, 2003.

[9] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On Fuzzy Modeling using Fuzzy Neural Networks with the Back-propagation Algorithm," *IEEE Trans. on Neural Networks*, **3**(5), 801-806, 1992.

[10] G. Hong and M. Z. Yuan "Immune Algorithm," *Proc. of the 4th World Congress on Intelligent Control and Automation*, **3**, 1784-1788, 2002.

[11] C. F. Juang, "A hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design," *IEEE Trans. on Systems, Man, Cybernetics- Part: B*, **34**(2), 997-1006, 2004.

[12] C. F. Juang, "A TSK-type Recurrent Fuzzy Network for Dynamic Systems Processing by Neural Network and Genetic Algorithms," *IEEE Trans. on Fuzzy Systems*, **10**(2), 155-170, 2002.

[13] J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization,” in *Proc. IEEE Int. Conf. Neural Networks*, pp. 1942-1948, 1995.

[14] D. H. Kim, “Parameter Tuning of Fuzzy Neural Networks by Immune Algorithm,” *IEEE Int. Conf. on Fuzzy Systems*, **1**, 408-413, 2002.

[15] C. H. Lee, “Stabilization of Nonlinear Nonminimum Phase Systems: an Adaptive Parallel Approach using Recurrent Fuzzy Neural Network,” *IEEE Trans. on Systems, Man, Cybernetics- Part: B*, **34**(2), 1075-1088, 2004.

[16] C. H. Lee and M. H. Chiu, “Adaptive Nonlinear Control using TSK-type Recurrent Fuzzy Neural Network System,” *Lecture Notes in Computer Science*, **4491**, 38-44, 2007.

[17] C. H. Lee and C. C. Teng, “Identification and Control of Dynamic Systems using Recurrent Fuzzy Neural Networks,” *IEEE Trans. on Fuzzy Systems*, **8**(4), 349-366, 2000.

[18] C. H. Lee and F. K. Chang, “Recurrent Fuzzy Neural Controller Design for Nonlinear Systems using Electromagnetism-like Algorithm,” *Far East Journal of Experimental and Theoretical Artificial Intelligence*, **1**(1), 5-22, 2008.

[19] C. H. Lee and Y. C. Lee, “An Improved Electromagnetism-like Algorithm for Neural Fuzzy Systems Design” *The* 16*th National Conference on Fuzzy Theory and Its Applications Conference*, Taoyuan, Taiwan, December, 2008.

[20] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Particle Swarm Optimization Algorithms with Novel Learning Strategies,” *IEEE Int. Conf. Systems, Man, Cybernetics*, **4**, 3659-3664, The Netherlands, Oct. 2004.

[21] J. J. Liang, A. K. Qin, “Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions,” *IEEE Trans. on Evolutionary Computation*, **10**(3), 281-295, 2006.

[22] C. J. Lin, C. Y. Lee, and C. C. Chin, “Dynamic Recurrent Wavelet Network Controllers for Nonlinear System Control,” *Journal of The Chinese Institute of Engineers*, **29**(4), 747-751, 2006.

[23] C. J. Lin and Y. J. Xu, “A Novel Evolution Learning for Recurrent Wavelet-based Neuro-fuzzy Networks,” *Soft Computing*, **10**(3), 193-205, 2006.

[24] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems*, Prentice Hall: Englewood Cliff, 1996.

[25] Ana Maria A. C. Rocha, Edite M. G. P. Fernandes, “On charge effects to the Electromagnetism-like Algorithm,” *Int. Conf. 20th EURO Mini Conf. “Continuous Optimization and Knowledge-Based Technologies*” May 20-23, Neringa, Lithuania, 2008.

[26] P. A. Mastorocostas and J. B. Theocharis, “A Recurrent Fuzzy-neural Model for Dynamic System Identification,” *IEEE Trans. on Systems, Man, Cybernetics, Part: B*, **32**(2), 176-190, 2002.

[27] K. E. Parsopoulos and M. N. Vrahatis, “On the Computation of all Global Minimizers through Particle Swarm Optimization,” *IEEE Trans. on Evolutionary Computation*, **8**, 211–224, 2004.

[28] Y. Shi and R. C. Eberhart, “A Modified Particle Swarm Optimizer,” in *Proc. IEEE Congr. Evol. Comput.*, 1998, 69–73.

[29] M. N. H. Siddique and M. O. Tokhi, “Training Neural Networks: Backpropagation vs. Genetic Algorithms,” *Proc. of Int. J. Conf. on Neural Networks*, **4**, 2673-2678, 2001.

[30] P. N. Suganthan, “Particle Swarm Optimizer with Neighborhood Operator,” in *Proc. Congr. Evol. Comput.*, Washington, DC, 1999, 1958–1962.

[31] C. S. Tsou and C. H. Kao, “Multi-objective Inventory Control using Electromagnetism-like Meta-heuristic,” *Int. Journal of Production Research*, **11**(1), 1-16, 2007.

[32] J. S. Wang and Y. P. Chen, “A Fully Automated Recurrent Neural Network for Unknown Dynamic System Identification and Control,” *IEEE Trans. on Circuits and Systems*-I, **56**(6), 1363-1372, 2006.

[33] P. Wu, W. H. Yang, and N. C. Wei, “An Electromagnetism Algorithm of Neural Network Analysis - An Application to Textile Retail Operation,” *Journal of the Chinese Institute of Industrial Engineers*, **21**(1), 59-67, 2004.

[34] G. Zarikian and A. Serrani, “Harmonic Disturbance Rejection in Tracking Control of Euler- Lagrange Systems: an External Model Approach,” *IEEE Trans. on Control Systems Technology*, **15**(1), 118-129, 2007.