Hindawi
Complexity
Volume 2017, Article ID 5813192, 11 pages
https://doi.org/10.1155/2017/5813192

Research Article

Parameter Optimization of MIMO Fuzzy Optimal Model Predictive Control By APSO

Adel Taieb, Moêz Soltani, and Abdelkader Chaari

Laboratory of Engineering of Industrial Systems and Renewable Energy, National High School of Engineers of Tunis (ENSIT), 5 Av. Taha Hussein, BP 56-1008, Tunis, Tunisia

Correspondence should be addressed to Adel Taieb; taeibadel@live.fr

Received 16 April 2017; Revised 18 June 2017; Accepted 22 August 2017; Published 9 October 2017

Academic Editor: Carla Pinto

This paper introduces a new development for designing a Multi-Input Multi-Output (MIMO) Fuzzy Optimal Model Predictive Control (FOMPC) using the Adaptive Particle Swarm Optimization (APSO) algorithm. The aim of this proposed control, called FOMPC-APSO, is to develop an efficient algorithm that is able to have good performance by guaranteeing a minimal control. This is done by determining the optimal weights of the objective function. Our method is considered an optimization problem based on the APSO algorithm. The MIMO system to be controlled is modeled by a Takagi-Sugeno (TS) fuzzy system whose parameters are identified using weighted recursive least squares method. The utility of the proposed controller is demonstrated by applying it to two nonlinear processes, Continuous Stirred Tank Reactor (CSTR) and Tank system, where the proposed approach provides better performances compared with other methods.

1. Introduction

Predictive control is a member of advanced discrete-time process control algorithms. This control algorithm is based on the use of an explicit process model to predict the manipulated variables and thus the future control actions are optimized throughout a finite horizon. To obtain a good performance, a process model describing the effects of all the different inputs on all the outputs must be developed. Although the linear model predictive control is suitable for processes that are not highly nonlinear, this strategy has been applied in the control of nonlinear systems, whether for SISO systems [1–3] or for MIMO systems [4–7]. But many industrial processes have strong nonlinearities and predictive control is applied in order to provide satisfactory control results. Two problems have appeared because of the introduction of nonlinearities in the predictive control.

(i) The first of the problems is that the modeling of processes is much more difficult and complex than the linear case. fuzzy logic is among the most used strategies in all areas [4]. Despite the fact that this strategy has been developed in the last few years, the fuzzy models of the TS type remain among the most used methods to deal with the MPC control for nonlinear systems (NMPC), because of their capacity to give an accurate approximation of the complex nonlinear MIMO systems.

(ii) The second important problem in nonlinear predictive control is the solving of the optimization problem. Conventional optimization methods, such as the gradient search method, used for designing the state feedback controller are restricted to the eigenvalues of the linear system matrix that not only increases the difficulty but also takes long time to find the global optimum solution [8]. Hence, evolutionary computation (EC) can be considered an alternative method to solve this type of optimization problem. In literature, plenty of works have been reported to solve the control optimization problems using EC techniques because they do not require explicit gradient information for optimization.

Particle Swarm Optimization (PSO), introduced by [9], is a population based metaheuristic search (MS) algorithm,

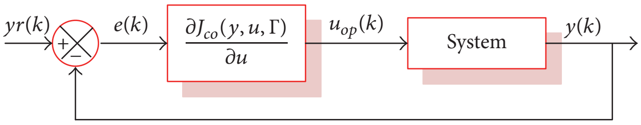


FIGURE 1: Problem principle of optimal control.

which emulates the collaborative behaviour of bird flocking and fish schooling in searching for foods. In addition, unlike other heuristic optimization methods, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. Since the introduction of PSO, many works based on MPC have been solved using PSO because it is not largely affected by the size and nonlinearity of the problem. Reference [10] introduces an approach for designing an adaptive fuzzy model predictive control using the PSO algorithm (AFMPC-PSO). In [11], a type of MPC is proposed using Chaos Particle Swarm Optimization (CPSO). First, for the modeling phase, the TS fuzzy model is employed to approximate the nonlinear system. Second, we introduce CPSO into MPC using a modified performance criterion in order to provide less computational controller's expression.

Although these methods have represented effective solutions to the problem of the MPC control of nonlinear systems, there is the problem of choice of control parameters. Several studies have shown the influence of these parameters on the quality of the responses of the systems treated. To overcome this problem, this paper presents a method of tuning the weight parameters of the performance function according to the output quantities detected from the system. One of the challenges in MPC is how control parameters can be tuned for various target systems, and the use of APSO for automatic tuning is one of the solutions. Firstly, for the modeling phase, the TS fuzzy model is employed to approximate the nonlinear system. In the second step, we used the principle of optimal control to calculate the control law of each linear subsystem. Then, we introduce APSO algorithm to determine the best weight parameters of the performance function using a performance criterion in order to improve the quality of the response with a minimum of energy.

The rest of the paper is organized as follows. Section 2 presents the influence of weights existing in the objective function on the quality of system performances. Section 3 reviewed the TS fuzzy model and the OMPC design method. The main contribution of this paper is presented in Section 4. In order to show the good performance of the proposal approach, simulation results are given in Section 5. Finally, Section 6 concludes.

2. Statement

The MPC control is a method of designing process control systems with feedback. This designing is carried out by the online repetition of a procedure that includes inputting data to a system of the initial input values. The principle of calculating the control law is the resolution of an optimal control problem using the present output values. So, the MPC control is a special case of the optimal control whose horizon is finite. It is recalled that this command minimizes the function described above. However, when the horizon is infinite, we speak of optimal control. The objective function is written as follows:

$$J_{co} = \sum_{i=1}^{n_y} \Gamma_{yi} \left(yr_i - y_i\right)^2 + \sum_{i=1}^{n_u} \Gamma_{ui} \left(u_i - ur_i\right)^2, \qquad (1)$$

where Γ_{yi}, Γ_{ui} are the weight values of the outputs and inputs. ur_i are the control instructions that are defined beforehand based on an expertise of the system. The partial derivative of the objective function described by (1) is as follows:

$$\frac{\partial J_{co}}{\partial u} = \frac{\partial \sum_{i=1}^{n_y} \Gamma_{yi} \left(yr_i - y_i\right)^2 + \sum_{i=1}^{n_u} \Gamma_{ui} \left(u_i - ur_i\right)^2}{\partial u}. \qquad (2)$$

By applying the principle of the optimal control, we obtain the following:

$$u_{op}(k) = \frac{\partial J_{co}\left(y, u, \Gamma_{yi}, \Gamma_{ui}\right)}{\partial u}. \qquad (3)$$

The structure of the loop system based on OMPC illustrating this method is shown in Figure 1. According to (1), the criterion J_{co} depends strongly on the weights $\Gamma_i = [\Gamma y_i, \Gamma u_i]$. These weights directly affect the performances index (Pi) of the system studied, such as overshoot (Ov%), settling time (Ts), rise time (Tr), and static error (Es).

To show the importance of the choice of these parameters on the response of the system in closed loop, we consider the following a multivariable linear system. This system is described by the following equations:

$$\begin{aligned} x_{11}(k) &= 0.9401 x_{11}(k-1) + u_1(k-1) \\ x_{12}(k) &= 0.9524 x_{12}(k-1) + u_2(k-1) \\ x_{21}(k) &= 0.9083 x_{21}(k-1) + u_1(k-1) \\ x_{22}(k) &= 0.9306 x_{22}(k-1) + u_2(k-1) \\ y_1(k) &= -0.7664 x_{11}(k) + 0.9 x_{12}(k) \\ y_2(k) &= -0.6055 x_{21}(k) + 1.3472 x_{22}(k). \end{aligned} \qquad (4)$$

The criterion to be minimized is:

$$J_{co} = \sum_{i=1}^{2} \Gamma y_i \left(y_i - yr_i\right)^2 + \sum_{i=1}^{2} \Gamma u_i \left(u_i - ur_i\right)^2; \quad (i = 1, 2) \qquad (5)$$

with $yr_1 = 6.1$, $yr_2 = 12.8$, $ur_1 = 1$, and $ur_2 = 1$.

In a first step, the minimization of the objective function is obtained by canceling the gradient of J_{co} with respect to u_i; the expressions are obtained as a function of x_{ij} of (4) and Γ:

$$\begin{gathered}
u_1(k) = 2\Gamma_{u1} + \Gamma y_2\left(1.63\Gamma y_2(k-1) - 0.66x_{21}(k-1) + 1.52x_{22}(k-1) - 15,50\right) \\
+ \frac{\Gamma y_1\left(\left(1.37u_2(k-1) - 1.10x_{11}(k-1) + 1.13x_{22}(k-1) - 9.35\right)\right)}{\left(1.17\Gamma y_1 + 0.73\Gamma y_2 + 2\Gamma_{u1}\right)}. \\
u_2(k) = 2\Gamma_{u_2} + \Gamma y_1\left(1,37u_1(k-1) + 1.29x_{11}(k-1) - 1.54x_{12}(k-1)\right) + 10.98 \\
+ \frac{\Gamma y_2\left(1.36u_1(k-1) + 1,48x_{21}(k-1) - 3.37x_{22}(k-1) + 34.48\right)}{\left(1.62\Gamma y_1 + 3.62\Gamma y_2 + 2\Gamma_{u2}\right)}
\end{gathered} \tag{6}$$

Table 1 shows how the overshoot, settling time, rise time, and static error as the performance indices vary with the weight parameters $(\Gamma y_1, \Gamma y_2, \Gamma u_1, \Gamma u_2)$ of the performance function when outputs transitions y_1 and y_2 of the optimal control are used. As can be seen from the table, the transient characteristics depend strongly on the weight parameters $(\Gamma y_1, \Gamma y_2, \Gamma u_1, \Gamma u_2)$ of evaluation function (4).

The example presented shows the influence of the choice of the values of the weights existing in the expression of the optimal control law. So, the weight parameters directly affect Ov%, Es, ts, and tr.

3. MIMO TS Fuzzy Model and OMPC Design

Takagi and Sugeno proposed the well-known TS fuzzy model in [12] to describe the complicated nonlinear system. The TS fuzzy models are universal approximators capable of approximating any continuous function with certain level of accuracy [13]. Since the MIMO system can be divided into multiple input-single output (MISO) systems, we take MISO systems instead, to identify MIMO systems. It is assumed that an MISO system $F(x, y)$ is the system that needs identification, where x is the system input with $x \in R^M$ and y is the system output with $y \in R$.

The rules of TS fuzzy models, used in this work, have the following form:

$$\begin{aligned}
R_g\text{: IF}\quad & Z_{i1} \text{ est } \Omega_{ig1} \text{ et} \ldots Z_{iM} \text{ est } \Omega_{igM} \\
\text{THEN}\quad & y_{ig}(k) = A_{ig}y(k-1) + B_{ig}u(k-1), \\
& i = 1, \ldots, ny,\; g = 1, 2, \ldots, c_i,
\end{aligned} \tag{7}$$

where R_g represents the gth rule, c_i is the number of rules for the ith subsystem, M is the dimension of the input vector, Ω_{ig} is the fuzzy subset of the g-th rule, $Z_i = [Z_{i1}, \ldots, Z_{iM}] \in R^M$ is the input vector, and $A_g = [A_{g1}, \ldots, A_{gny}]$ and $B_g = [B_{g1}, \ldots, \mathrm{B}_{gnu}]$ are two polynomial vectors. The final output is calculated as the average of the outputs corresponding to the rules R_r, weighted by the normalized degree of completion (membership), as follows:

$$\hat{y}_i(k) = \frac{\sum_{g=1}^{c_i} \mathrm{w}_{ig}(Z_i)\, y_{ig}(k)}{\sum_{g=1}^{c_i} \mathrm{w}_{ig}(Z_i)} \tag{8}$$

$$\mathrm{w}_{ig}(Z_i) = \prod_{j=1}^{M} \mathrm{A}_{igj}(Z_{ij}). \tag{9}$$

The standardized degree of completion is described in the following form:

$$\mu_{ig}(Z_i) = \frac{\mathrm{w}_{ig}(Z_i)}{\sum_{g=1}^{c_i} \mathrm{w}_{ig}(Z_i)}. \tag{10}$$

The standard degree of achievement is actually the degree of activation of the corresponding local model in the region where the system evolves. The fuzzy subsets are generally Gaussian, triangular or sigmoid and must satisfy the following properties:

$$\begin{gathered}
\sum_{g=1}^{c_i} \mu_{ir}(Z_i) = 1, \quad \forall g = 1, 2, \ldots, c_i \\
0 \le \mu_{ig}(Z_i) \le 1.
\end{gathered} \tag{11}$$

The MPC control is a method of designing process control systems with feedback. This designing is carried out by the online repetition of a procedure that includes inputting data to a system of the initial input values. The MPC control is a special case of the optimal control. It is recalled that this control minimizes the function described above. However, when the horizon is infinite, we speak of optimal control for each linear subsystem.

The concept of the OMPC technique for TS system is utilized to design fuzzy controller. In this concept, the fuzzy controller rule shares the same premise part as the fuzzy system (8) and use same number of fuzzy rules.

The fuzzy controller is inferred as follows:

$$U_i = \frac{\sum_{g=1}^{c_i} \mathrm{w}_{ig}(Z_i)\, u_{ig}}{\sum_{g=1}^{c_i} \mathrm{w}_{ig}(Z_i)}. \tag{12}$$

In the OMPC, the cost index to be minimized is expressed as follows:

$$J_g = \sum_{i=1}^{n_y} \Gamma_{y_g}\left(yr_{ig} - y_{ig}\right)^2 + \sum_{i=1}^{n_u} \Gamma_{u_g}\left(u_{ig} - ur_{ig}\right)^2, \tag{13}$$

Table 1: Weight parameter Γ_i dependence of transient characteristic.

Pi	$\begin{Bmatrix} \Gamma y_1,\Gamma y_2 \\ \Gamma u_1,\Gamma u_2 \end{Bmatrix}$			
	$\begin{Bmatrix} 1,1 \\ 1,1 \end{Bmatrix}$	$\begin{Bmatrix} 1,1 \\ 0.2,0.2 \end{Bmatrix}$	$\begin{Bmatrix} 1,0.1 \\ 1,0.1 \end{Bmatrix}$	$\begin{Bmatrix} 0.3,0.3 \\ 1,0.1 \end{Bmatrix}$
Es1	0.000	0.003	0.015	0.000
Es2	0.000	0.001	0.02	0.000
Ov1 (%)	40.41	32.09	09.33	26.33
Ov2 (%)	57.3988	36.36	25.33	53.20
Ts1	12.00	10.50	8.00	12.00
Ts2	11.00	13.00	16.00	17.00
Tr1	4.869	5.4701	4.6000	5.5637
Tr2	7.00	6.50	20.00	21.00

where Γ_{y_g}, Γ_{u_g} are the weight values of the outputs and inputs. ur_{ig} are the control instructions that are defined beforehand based on an expertise of the system. The general structure of the gth controller is then as follows:

$$\begin{aligned} R_g\text{: IF} \quad & Z_{i1} \text{ is } \Omega_{ig1}, \dots, Z_{iM} \text{ is } \Omega_{igM} \\ \text{THEN} \quad & u_{ig}(k) = \frac{\partial J_{ig}}{\partial u_{ig}}. \end{aligned} \tag{14}$$

The previous solution shows that the Γ weights directly affect the performance of the system. Indeed, despite the fact that the weight values Γ chosen can give good performance, we can not conclude that these are the best values.

4. Design of the Proposed FOMPC Controller

In this section, we describe a new method for determining the weight values of the objective function J_g of each local system.

So, an objective function is given, and the optimal control law of each local system is calculated. Then, the APSO algorithm intervenes to determine the optimal values of the weight. The next step is to design the global control law such that the proposed OFMPC-APSO presents the desired dynamic characteristics. The proposed FOMPC based on APSO combines both of the advantages of FOMPC and APSO algorithm. APSO algorithm is used to search and to fine tune the weight vector Γ_y and Γ_u of MPC controller. So, the expression of control law by the principle of optimal control is calculated. Once the latter has been established, the next step is to calculate the optimal weights such that the proposed algorithm presents the desired dynamic characteristics.

4.1. Design of the Linear Control Law. We consider parameter optimization problem as a simple problem of tuning only the weight Γ. Now let us assume that the performance function Φ for each output evaluates the (Ov%), (Ts), (Tr), and (Es). Let us define the performance function Φ as proposed in the following:

$$\Phi_g = q_1 \text{Es} + q_2 \text{Ov\%} + q_3 \text{Ts} + q_4 \text{Tr}. \tag{15}$$

Here (q_1, q_2, q_3 et q_4) are the weights of the respective performance indices. So, we obtain the optimal values of the weights of criterion (13) while respecting the following constraint:

$$\min_{\Gamma_{\min} < \Gamma < \Gamma_{\max}} \quad \Phi_g \quad (\Gamma). \tag{16}$$

With $\Gamma_{\min}$ and $\Gamma_{\max}$ the minimum and maximum limits are chosen.

The optimization problem given by (15) is a constrained nonlinear and nonconvex optimization problem, the solution of which is difficult and generally expensive in computing time. Different approaches were investigated to solve this problem, such as the numerical optimization techniques [14, 15], the metaheuristic based optimization algorithms [16–18], the linearization of the process fuzzy model [19], and the use of particular model structures to obtain a convex form for the cost function [20].

4.2. APSO Algorithm. The PSO algorithm is a type of stochastic global optimization algorithm for improving candidate solutions [9]. This algorithm iteratively explores a multidimensional search space with a swarm of individuals (referred to as "particles"), looking for the global optima [21, 22]. PSO has memory to store the knowledge of good solutions by all particles; in addition, particles in the swarm share information with each other. Therefore, due to the simple concept, easy implementation, and quick convergence, nowadays PSO has gained much attention and wide applications in solving continuous nonlinear optimization problems [23–26].

It is initialized with a population of random solutions, called particles, to find the optimal result. Each particle has a position and a velocity, representing a possible solution to the optimization problem and a search direction in the search space. In each iterative process, the particle adjusts the velocity and position according to the best experiences that are called the $p$best, found by itself, and $g$best, found by all its neighbors [27]. For every generation, the velocity and position can be updated by the following equations:

$$\begin{aligned} V_p^d(\text{t}+1) = {} & wV_p^d(\text{t}) + r_1 C_1 \left(p\text{best}_p^d - X_p^d(\text{t}) \right) \\ & + r_2 C_2 \left(p\text{best}_g^d - X_p^d(\text{t}) \right) \end{aligned} \tag{17}$$

$$X_p^d(\text{t}+1) = X_p^d(\text{t}) + V_p^d(\text{t}+1), \tag{18}$$

where $i = 1, 2, \dots, Np$ and Np is the number of particles, t is the number of iterations, and r_1 and r_2 are two random

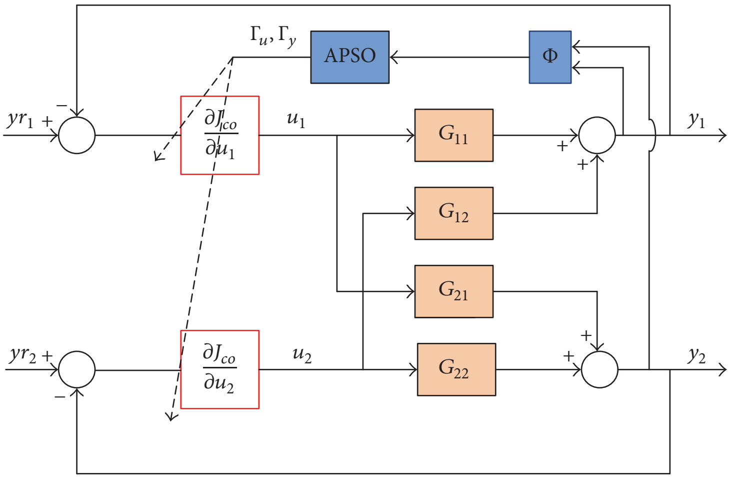


Figure 2: The control scheme.

Pahse 1. Identification of the model

Step 1. The nonlinear MIMO system is modeled by a TS fuzzy inference system.

Step 2. Identify of parameters using WRLS method.

Pahse 2. Application of Algorithm 1 *for each local model*

Step 1. The weight parameter Γ is specified.

Step 2. Give the objective function in the form of Eq. (13).

Step 3. Calculate the control law u_i using Eq. (3).

Step 4. Compute Γ^{Best} that minimize the restricted function (15) using the procedure of Algorithm 2 and find the best particle labeled as $\Gamma = [\Gamma_{y_i}^{\text{Best}}, \Gamma_{u_i}^{\text{Best}}]$.

Step 5. Calculate the control law u_i using (3) According to the optimum values of Γ^{Best}

Pahse 3. Design the control law U via equation (12)

Algorithm 1: FOMPC-APSO algorithm.

numbers in the interval $[0, 1]$. C_1 and C_2 are positive constants. w is the inertia weight, a parameter used to control the impact of the previous velocities on the current velocity. If it is chosen properly, the particle will have the balanced ability of exploitation and development. w is updated as follows:

$$w = w_{\max} \left(\frac{w_{\max} - w_{\min}}{\mathrm{t}_{\max}} \right) \mathrm{t}, \tag{19}$$

where $w_{\min}$ and $w_{\max}$ are minimum and maximum values of w which are taken as 0.4 and 0.9, respectively.

In conventional PSO, the velocity of each particle in the next search is updated using the knowledge of its past velocity and personal and global best positions. However, the performance of PSO greatly depends on its parameters; it often suffers from being trapped in local optimum [28, 29]. Indeed, the inertia weight is the most important parameter to balance the local search ability and global search ability [30]. This balancing is a key role to improve the performance of PSO. However, the method of selecting inertia weight is difficult. And experiments show that particles can accumulate at point in local searching area. So, to avoid all these difficulties, an improved version of PSO has appeared; it is the APSO algorithm.

The basic idea of APSO is that the global best and the personal best position of particle always change over iteration and tend to the similar value if the swarm has approached the solution. The values of $p$best and $g$best are taken and are used to adjust the value of inertia weight by using feedback mechanism. In this condition, the inertia weight should be set to larger value. So, the balancing between local-global can be controlled based on the swarm condition.

The modified inertia weight is modified as follows:

$$\omega = \left(\omega_0 - \frac{p_{g_{\text{best}}}}{p_{p_{\text{best}}}} \right), \tag{20}$$

where χ_0 is the initial value of inertia weight.

The controller structure is illustrated in Figure 2. This Figure represents the case of a system with two inputs (u_1 and u_2) and two outputs (y_1 and y_2). yr_1 and yr_2 represent the references of each output, respectively.

4.3. FOMPC-APSO Algorithm. The general design steps of the FOMPC-APSO algorithm are summarized in Algorithm 1.

5. Simulation Study

In order to show the considerable contribution in the performance of the proposed control scheme, two highly nonlinear

```
S.1. Choose the weighting matrices $\Gamma = [\Gamma_{y_i} > 0, \Gamma_{u_i} > 0]$, $\Gamma_{\max}$, $\Gamma_{\min}$, the number of particles NP.
Initialize the position and velocity of each particle; fix learning factors $C1$ and $C2$; $\omega_0$ and the
number of iterations $t_{\max}$.
S.2. For t = 1 to $t_{\max}$ do
for each particle do
(1) Calculate the fittness value of each particle by minimizing the following Eq. (15)
(2) Find the individual best pbest for each particle and the global best gbest.
(3) Update the velocity and the position of each particle using equations (17) and (18), respectively.
end for
end for
S.3. Find the best particle labeled as $\Gamma = [\Gamma_{y_i}^{\text{Best}}, \Gamma_{u_i}^{\text{Best}}]$
```

Algorithm 2: Procedure of weight parameter optimization.

Table 2: Specification of the surge tank.

Parameter	Description	Normal operation condition
H_0	Initial value of tank level	0.15 [m]
H_s	Initial value of the output channel level	0.015 [m]
a	Section of the channel output	0.0001 [m^2]
A	Section of the tank	0.04 [m^2]
Q_0	The initial flow	0.00013 [m^3s^{-1}]
k_0	Constant	1
k_1	Constant	0.1

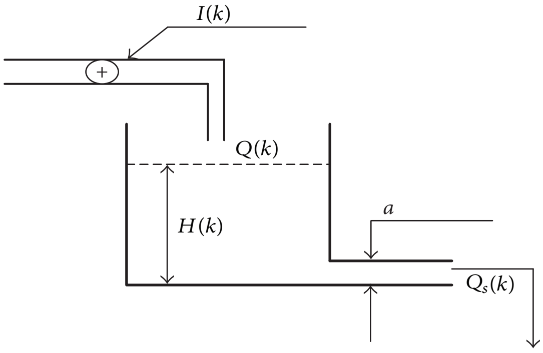


Figure 3: The surge tank system.

systems are selected. The first example is the surge tank. The second example is the CSTR. We compare our results with those obtained by other existing methods such as NMPC [31] and FMPC using the APSO algorithm [32].

In this paper, the Tr, Ts, Ov%, pic, and Es are used as the performance indexes.

5.1. Surge Tank System. The behaviour of the surge tank system, shown in Figure 3, is fed by a pump driven by a current $I(k)$ [10].

In Figure 3, $Q(k)$ is the feed rate, $I(k)$ is the supply current of the pump, H is the liquid level in the tank, Q_s is the output flow, a is the section of the output channel, A is the section of the tank, and H_s is the water level in the output channel. The mathematical model of this reactor is

(i) Model of the valve is as follows:

$$Q(k) = Q(k-1) + Te\left(-k_0 Q(k-1) + k_1 I(k-1)\right) \quad (21)$$

(ii) The change in water level in the tank is given by the following:

$$\begin{aligned} V(k) &= AH(k) \\ &= H(k-1) + Te\left(Q(k-1) - Q_s(k-1)\right), \end{aligned} \quad (22)$$

where $Q_s(k) = 0.6a\sqrt{2g(H(k) - H_s)}$.

The values of the constant parameters of this system are grouped in the Table 2.

Fuzzy modeling: 1000 pairs of data are used to identify the model using the FCM algorithm. For a good approximation of the plant, we suppose that the subsystems are in the third order. The model consists of two rules of the form:

$$\begin{aligned} R^1&: \text{If} \quad I_1 \text{ is } Q^1 \\ &\text{THEN} \quad H^1(k) \\ &\qquad = a_{11}H(k-1) + a_{12}H(k-2) \\ &\qquad\quad + a_{13}H(k-3) + b_{11}I_1(k-1) \\ &\qquad\quad + b_{12}I_1(k-2) \\ R^2&: \text{IF} \quad I_1 \text{ is } Q^2 \\ &\text{THEN} \quad H^2(k) \\ &\qquad = a_{21}H(k-1) + a_{22}H(k-2) \\ &\qquad\quad + a_{23}H(k-3) + b_{21}I_1(k-1) \\ &\qquad\quad + b_{22}I_1(k-2). \end{aligned} \quad (23)$$

Table 3: Pi performances obtained by the different algorithms.

Pi	Algorithms		
	NMPC	FMPC-APSO	*FOMPC-APSO*
Tr	0.8111	0.9820	*0.7402*
Ts	5.6402	11.0788	*1.9936*
Ov%	6.0804	2.6922	*0.5017*
Pic	1.26	0.2055	*0.2162*
Es	0.00	0.00	*0.00*
W%	53.4	54.86	—

The vector of parameters of gth rule is obtained by using the WRLS. This fuzzy model is used to represent the process model in the controller:

$$\begin{aligned} a_{11} &= 1.952, \\ a_{21} &= 0.0487, \\ a_{31} &= 0.0522, \\ b_{11} &= 0.1131, \\ b_{21} &= 0.1106, \\ a_{21} &= 0.8421, \\ a_{22} &= 0.0808, \\ a_{23} &= 0.0764, \\ b_{21} &= 0.1668, \\ b_{22} &= 0.1697. \end{aligned} \tag{24}$$

Once the estimated model is obtained, we will investigate the optimal parameters of the FOMPC control law using the APSO algorithm. This gives the best results with these settings: $t_{\max} = 350$, $Np = 35$, $C_1 = C_2 = 2.05$, $r_2 = 0.2$, and $w_0 = 1.4$. The fitness function of the APSO algorithm is defined by the following:

$$J_i = \Gamma_{yi}\left(H_i - H_r\right)^2 + \Gamma_{Ii}\left(I_i\right)^2, \quad i = 1, 2. \tag{25}$$

Table 3 shows the performances obtained by these methods. In each interval time, we have changed the set point for evaluating each method to control a highly nonlinear system. The proposed method can generate a high quality solution within shorter calculation time and it tends to converge very fast compared to other methods. The comparison shows some interesting results. It is important to observe that, with FOMPC-APSO, the Ts has been reduced almost 2 times comparing with that obtained from the NMPC and has been reduced almost 5 times comparing with that obtained from the FMPC-APSO. The same observation can be made for the Ov%, where in the FOMPC-APSO case we notice a reduction of nearly 4 times compared with that obtained from FMPC-APSO and more than 10 times with NMPC without any increase in Tr. So, the proposed method is able to keep better stability with less control effort applied.

In fact, the proposed technique is able to achieve good performance using *53.4%* of total control energy consumed by FMPC-APSO as well as *54.68%* of FMPC-APSO. As it is presented in Table 3, the accuracy of our model outperforms that of other methods. To confirm these results further, Figure 4 shows the variations in liquid level in the tank when a step change is applied at 30, 60, and 90, respectively, by FOMPC-APSO, FMPC-APSO, and NMPC.

Figure 5 shows the manipulated responses when a disturbance is applied to the feed rate, at 40 and 80, respectively.

In conclusion, the proposed controller shows the best performance for both set point tracking and regulatory conditions for the entire range of the process as compared to the other controllers.

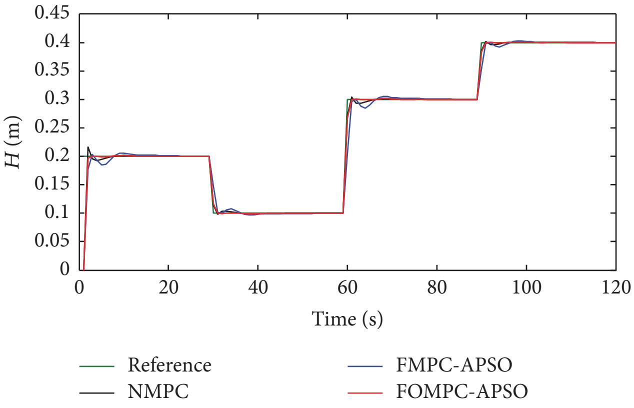


Figure 4: Evolution of liquid level H.

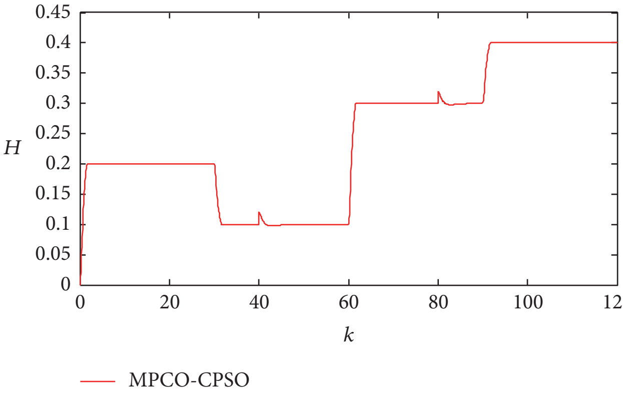


Figure 5: Evolution of liquid level H in the presence of disturbance.

Table 4: CSTR model parameters.

Description	Parameter	Nominal value
Product concentration	Ca	0.1 [mol L^{-1}]
Reactor temperature	T	438.51 [K]
Coolant flow rate	q_c	103.41 [min^{-1}]
Process flow rate	q	100 [L min^{-1}]
Feed concentration	C_{Af}	1 [mol L^{-1}]
Feed temperature	T_f	350 [K]
Inlet coolant temp	T_{cf}	350 [K]
Reaction volume	v	100 [L]
Heat transfer coefficient	h_a	$7 * 10^5$ [cal min^{-1} K]
Reaction rate constant	k_0	$7.2 * 10^{10}$ [min^{-1}]
Activation energy term	E/R	$1 * 10^4$ [K]
Heat of reaction	ΔH	$2 * 10^5$ [cal mol^{-1}]
Liquid densities	ℓ_c, ℓ	$1 * 10^3$ [g L^{-1}]
Specific heats	C_p, C_{pc}	1 [cal g^{-1} K^{-1}]

In fact, the proposed technique is able to achieve good performance using 53.4% of total control energy consumed by NMPC as well as 54.86% of FMPC-APSO. As it is presented in Table 3, the accuracy of our model outperforms that of other methods.

5.2. Continuous Stirred Tank Reactor (CSTR). The efficiency and the control accuracy of the proposed algorithm were investigated and compared to other control strategies by considering the control of a highly nonlinear MIMO process, namely, a Continuous Stirred Tank Reactor, where the model is presented in [33] and described by the following differentials equations:

$$\begin{aligned} C_a(k+1) &= C_a(k) + T_e\left(\frac{1}{v}q(k)\left(C_{Af} - C_a(k)\right)\right. \\ &\quad \left. - k_0 C_a(k)\, e^{(-E/RT(k))}\right) \\ T(k+1) &= T(k) + T_e\left(\frac{1}{v}q(k)\left(T_f - T(k)\right)\right. \\ &\quad + k_1 C_a(k)\, e^{-(E/RT(k))} \\ &\quad \left. + k_2 q_c(k)\left(1 - e^{-(k_3/q_c(k))}\right)\left(T_{cf} - T(k)\right)\right), \end{aligned} \tag{26}$$

where $k_1 = -\Delta H k_0/\ell C_p$, $k_2 = \ell_c C_{pc}/\ell C_p v$, and $k_3 = h_a/\ell_c C_{pc}$.

The process describes the reaction that converts the product A into a new product B, the concentration C_a is the concentration of product A, and T is the temperature of the mixture. The reaction is exothermic and it is controlled by a coolant flow whose rate is represented by q_c. The temperature is controlled by changing the coolant and by controlling the temperature, and the concentration is also controlled. Ca_0 is the inlet feed concentration, q is the process flow rate, and T_f and Tc_f are the inlet feed and coolant temperatures. All these values are assumed constant at nominal values. In the same way, k_0, E/R, v, k_1, k_2, and k_3 are thermodynamic and chemical constants. The numerical values of these parameters are given in Table 4.

Fuzzy modeling: the above nonlinear model is used to produce input-output time data. The sampling time is set to 0.01 min. $[Ca(k-1),\ Ca(k-2),\ q_c(k-1),\ q_c(k-2),\ q(k-1)]$ and $[T(k-1),\ T(k-2),\ q(k-1),\ q(k-2),\ q_c(k-1)]$ are selected as the input vector. The rule numbers of the identified fuzzy models are four.

The APSO algorithm parameters are chosen as follows: the maximum number of APSO iterations $t_{\max} = 110$, number of particles $Np = 10$, $r_1 = 0.2$, $r_2 = 0.2$, $C_1 = 1.5$, $C_2 = 2.5$, and $w_0 = 1.4$.

The fitness function is selected as follows:

$$\begin{aligned} J_i &= \Gamma y_{1i}\left(Ca_i - Ca_r\right)^2 + \Gamma q_i\left(q_{ci}\right)^2 + \Gamma y_{2i}\left(T_i - T_r\right)^2 \\ &\quad + \Gamma_{q_i}\left(q_i\right)^2 \qquad i = 1, 2, 3, 4. \end{aligned} \tag{27}$$

The reference signals applied to the system are as follows:

$$\begin{aligned} Ca_r &= \begin{cases} 0.1 & 0 \le k < 50 \\ 0.05 & 50 \le k < 100 \\ 0.15 & 100 \le k < 150 \\ 0.25 & 150 \le k \end{cases} \\ T_r &= \begin{cases} 480 & 0 \le k < 50 \\ 365 & 50 \le k < 100 \\ 495 & 100 \le k < 150 \\ 465 & 150 \le k. \end{cases} \end{aligned} \tag{28}$$

Tables 5 and 6 contain the performance indices obtained for the outputs Ca and T by the NMPC [10], FMPC-APSO [11], and FOMPC-APSO algorithms. It summarizes the results of this example in terms of the Ov%, Tr, Ts, and Es. As seen

Table 5: Pi found by different methods.

Pi	Algorithms		
	NMPC	FMPC-APSO	*FOMPC-APSO*
Tr	1.4965	4.4693	*0.4135*
Ts	6.8561	7.2779	*3.5480*
Ov%	5.0973	0.1215	*0.00*
Pic	1.0510	1.0012	*1.0000*
Es	0.00	0.00	*0.00*
W%	48.94	47.28	—

Table 6: Pi found by different methods.

Pi	Algorithms		
	NMPC	FMPC-APSO	*FOMPC-APSO*
Tr	1.3711	3.0459	*0.4116*
Ts	9.2185	4.5471	*3.2626*
Ov%	5.4003	1.6309	*0.0011*
Pic	1.0540	1.0163	*1.0000*
ES	0.00	0.00	*0.00*
W%	48.14	49.07	—

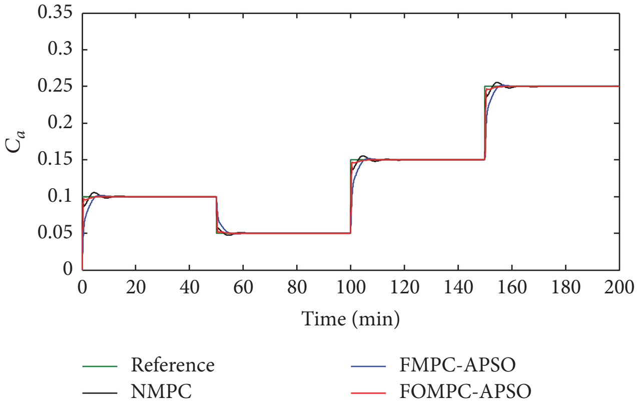


Figure 6: Evolution of concentration *Ca*.

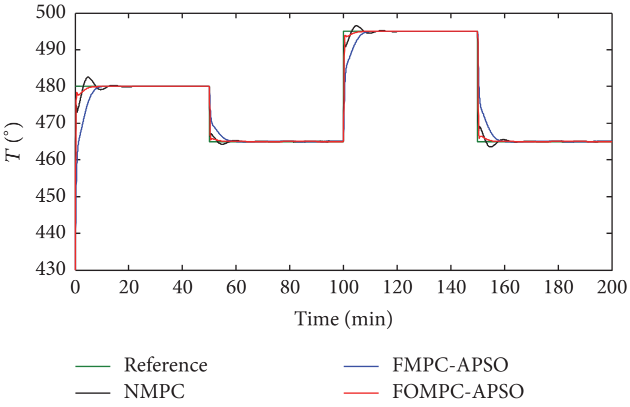


Figure 7: Evolution of temperature *T*.

in these tables, we can note that our method gives the best performance of all compared techniques.

They applied NMPC algorithm to control the concentration of product and it has the Tr and Ts values of 1.4965 min and 6.8561 min, while the FMPC-APSO approach has 4.4693 min and 7.2779 min. However, the corresponding Tr and Ts values for the same problem were 0.4135 min and 3.5480 and with no Ov%. This indicates that the proposed controller is able to perform faster than the other methods in real application environment. Figures 6 and 7 show the evolution of the *Ca* and *T* outputs from the three methods. From these figures, there is a perfect continuation of the signal of the setpoint whose FOMPC-APSO method has ensured good performances. We also note that our method provides more acceptable control effort regarding Figures 8 and 9.

In a second test, the disturbances on the system output in different times have been applied to validate the tracking of the reactor concentration. Thus, a disturbance of 0.002 mol/l at time $t = 82$ min and time $t = 130$ min is added. Figure 10 illustrates the disturbance rejection performance of the FOMPC-APSO controller. The results show that the adaptive controller has the ability to keep the process stable and regulate the outlet concentration at its desired set point value.

6. Conclusion

In this paper, we have introduced the FOMPC-APSO controller applied to highly nonlinear systems. An approach of determining the optimum weights is developed by minimizing a chosen performance criterion using APSO algorithm. The proposed approach is based on the advantage of the TS fuzzy system in the modeling phase and the metaheuristic optimization APSO algorithm in a new structure predictive controller. The advantage of this structure is its ability to handle highly nonlinear systems regardless and keep a good stability in terms of overshoot, rise time, and settling time including disturbances. We have achieved these objectives

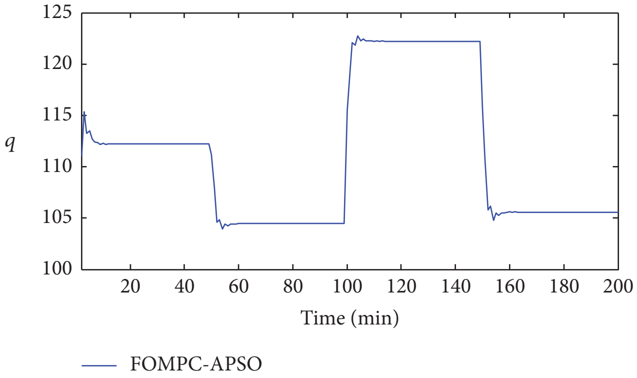


Figure 8: Evolution of control q.

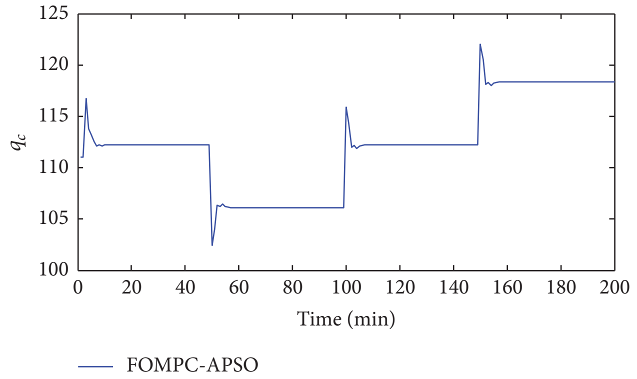


Figure 9: Evolution of control q_c.

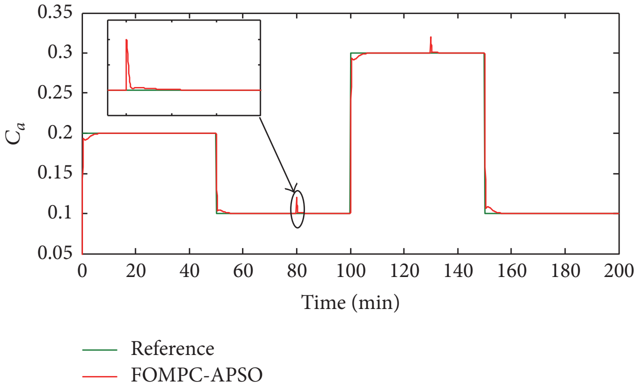


Figure 10: Evolution of concentration with external disturbances.

without any obligation increase in the control signal since we have injected the phenomenon of optimal control in the synthesis of our controller. Compared with other similar existing methods, the FOMPC-APSO algorithm enhances the convergence and accuracy of the controller optimization, which is much easier for implementation in real time.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. Alvarado, D. Limon, D. M. de la Peña et al., "A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark," *Journal of Process Control*, vol. 21, no. 5, pp. 800–815, 2011.

[2] C. Liu, W.-H. Chen, and J. Andrews, "Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers," *Control Engineering Practice*, vol. 20, no. 3, pp. 258–268, 2012.

[3] S. Hadj Saïd, F. M'Sahli, M. F. Mimouni, and M. Farza, "Adaptive high gain observer based output feedback predictive controller for induction motors," *Computers and Electrical Engineering*, vol. 39, no. 2, pp. 151–163, 2013.

[4] S. Bououden, M. Chadli, S. Filali, and A. El Hajjaji, "Fuzzy model based multivariable predictive control of a variable speed wind turbine: LMI approach," *Renewable Energy*, vol. 37, no. 1, pp. 434–439, 2012.

[5] V. Kirubakaran, T. K. Radhakrishnan, and N. Sivakumaran, "Distributed multiparametric model predictive control design for a quadruple tank process," *Measurement: Journal of the International Measurement Confederation*, vol. 47, no. 1, pp. 841–854, 2014.

[6] B. M. Al-Hadithi, A. Jiménez, and J. Perez-Oria, "New incremental Takagi-Sugeno state model for optimal control of multivariable nonlinear time delay systems," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 259–268, 2015.

[7] H. Gouta, S. H. Said, and F. M'Sahli, "Predictive and backstepping control of double tank process: A comparative study," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 33, no. 2, pp. 137–147, 2016.

[8] S.-J. Tsai, C.-L. Huo, Y.-K. Yang, and T.-Y. Sun, "Variable feedback gain control design based on particle swarm optimizer for automatic fighter tracking problems," *Applied Soft Computing Journal*, vol. 13, no. 1, pp. 58–75, 2013.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

[10] S. Bououden, M. Chadli, F. Allouani, and S. Filali, "A new approach for fuzzy predictive adaptive controller design using particle swarm optimization algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 9, pp. 3741–3758, 2013.

[11] A. Taeib, M. Soltani, and A. Chaari, "Model predictive control based on chaos particle swarm optimization for nonlinear processes with constraints," *Kybernetes. The International Journal of Cybernetics, Systems and Management Sciences*, vol. 43, no. 9-10, pp. 1469–1482, 2014.

[12] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[13] B. Kosko, "Fuzzy systems as universal approximators," in *Proceedings of the 1st IEEE International Conference on Fuzzy Systems (FUZZ '92)*, pp. 1153–1162, March 1992.

[14] P. Kittisupakorn, P. Thitiyasook, M. A. Hussain, and W. Daosud, "Neural network based model predictive control for a steel pickling process," *Journal of Process Control*, vol. 19, no. 4, pp. 579–590, 2009.

[15] A. Steinboeck, D. Wild, and A. Kugi, "Nonlinear model predictive control of a continuous slab reheating furnace," *Control Engineering Practice*, vol. 21, no. 4, pp. 495–508, 2013.

[16] H. Sarimveis and G. Bafas, "Fuzzy model predictive control of non-linear processes using genetic algorithms," *Fuzzy Sets and Systems. An International Journal in Information Science and Engineering*, vol. 139, no. 1, pp. 59–80, 2003.

[17] H. Jiang, C. K. Kwong, Z. Chen, and Y. C. Ysim, "Chaos particle swarm optimization and T-S fuzzy modeling approaches to constrained predictive control," *Expert Systems with Applications*, vol. 39, no. 1, pp. 194–201, 2012.

[18] Y. Li, J. Shen, K. Y. Lee, and X. Liu, "Offset-free fuzzy model predictive control of a boiler-turbine system based on genetic algorithm," *Simulation Modelling Practice and Theory*, vol. 26, pp. 77–95, 2012.

[19] J. A. Roubos, R. Babuska, P. M. Bruijn, and H. B. Verbruggen, "Predictive control by local linearization of a Takagi-Sugeno fuzzy model," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 37–42, May 1998.

[20] Y. L. Huang, H. H. Lou, J. P. Gong, and T. F. Edgar, "Fuzzy model predictive control," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 665–678, 2000.

[21] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, Springer, New York, NY, USA, 1998.

[22] J. Yu, S. Wang, and L. Xi, "Evolving artificial neural networks using an improved PSO and DPSO," *Neurocomputing*, vol. 71, no. 4–6, pp. 1054–1060, 2008.

[23] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 81–86, Seoul, Republic of Korea, 2001.

[24] M. Rashid and A. R. Baig, "Psogp: A genetic programming based adaptable evolutionary hybrid particle swarm optimization," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 1, pp. 287–296, 2010.

[25] I. Pan and S. Das, "Fractional order fuzzy control of hybrid power system with renewable generation using chaotic PSO," *ISA Transactions*, vol. 62, pp. 19–29, 2016.

[26] M. A. Duarte-Mermoud and F. Milla, "Model predictive power stabilizer optimized by PSO," in *Proceedings of the IEEE International Conference on Automatica, ICA-ACCA*, October 2016.

[27] L. Zhao, Y. Yang, and Y. Zeng, "Eliciting compact T-S fuzzy models using subtractive clustering and coevolutionary particle swarm optimization," *Neurocomputing*, vol. 72, no. 10-12, pp. 2569–2575, 2009.

[28] P. Angeline, "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences," in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., vol. 1447 of *Lecture Notes in Computer Science*, pp. 601–610, Springer, Berlin, Germany, 1998.

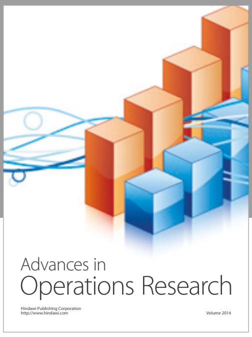
[29] B. Liu, L. Wang, Y. Jin, F. Tang, and D. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons and Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.

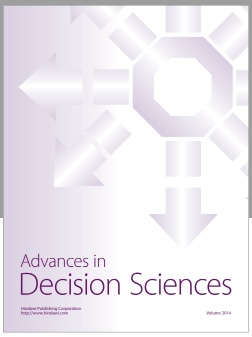
[30] Y. Shi and R. Eberhart, "A modifieded particle swarm optimization," in *Proceedings of the IEEE Int. Conf. Evol. Computer*, pp. 591–600, Anchorage, Alaska, USA, 1998.

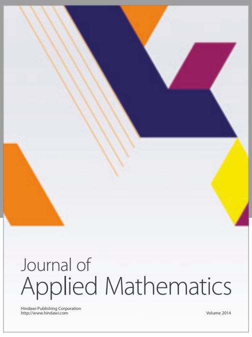
[31] L. D. S. Coelho and V. C. Mariani, "A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch," *Chaos, Solitons & Fractals*, vol. 39, no. 2, pp. 510–518, 2009.

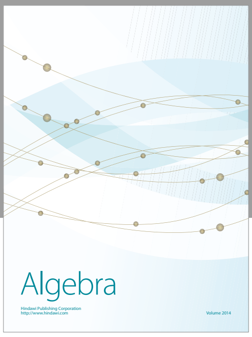
[32] O. Ait Sahed, K. Kara, and M. L. Hadjili, "Constrained Fuzzy Predictive Control Using Particle Swarm Optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2015, pp. 1–15, 2015.

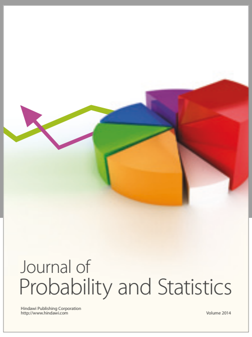
[33] M. Ikravesh, *Dynamic neural network control, ph.d. dissertation [Ph.D. thesis]*, In University of South Carolina, Columbia, SC, USA, 1994.


Advances in Operations Research

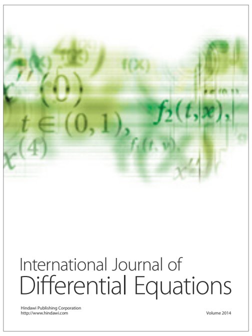

Advances in Decision Sciences

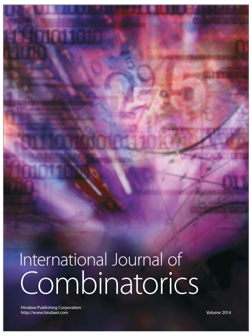

Journal of Applied Mathematics

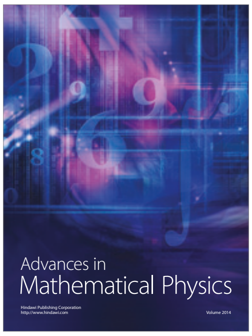

Algebra

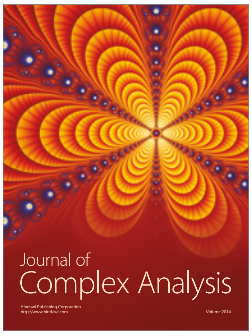

Journal of Probability and Statistics

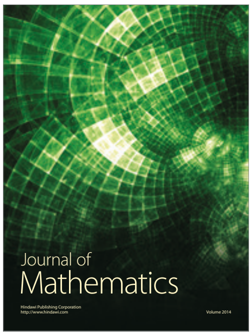
The Scientific World Journal

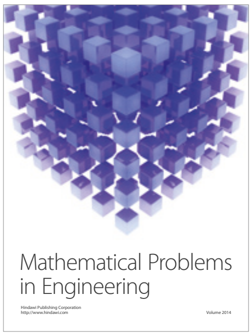

International Journal of Differential Equations

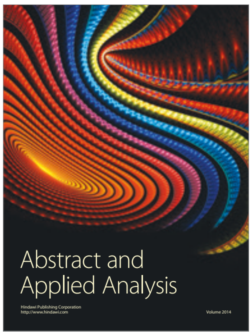

International Journal of Combinatorics

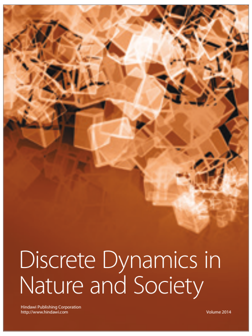

Advances in Mathematical Physics

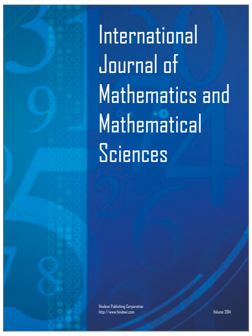

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

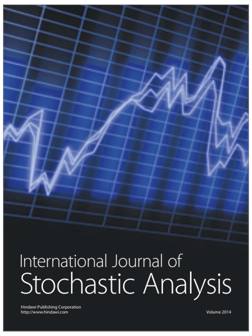

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization