Conférence Internationale en Automatique & Traitement de Signal (ATS-2017)
Proceedings of Engineering and Technology – PET
Vol.22 pp.53-60

Parameter optimization of MPC for TS fuzzy Model

Taieb Adel[#1], ChaariAbdelkader[*2]

Laboratory of Engineering of Industrial Systems and Renewable Energy,
University/Institution: High School of Sciences and Engineering of Tunis (ENSIT)
Town/City:Tunis State (US only): Tunis
[1]taeibadel@live.fr

Laboratory of Engineering of Industrial Systems and Renewable Energy
University/Institution: High School of Sciences and Engineering of Tunis (ENSIT)
Town/City:Tunis State (US only): Tunis
2asil.chari@esstt.rnu.tn

***Abstract* — This paper presents a model predictive control (MPC) strategy based on particle swarm optimization (PSO) to solve the nonlinear system control problem. First, a Takagi–Sugeno (TS) fuzzy model is established to approximate the behavior of these nonlinear systems. Then, a specially designed PSO algorithm is employed to calculate the MPC weight parameters. The combination between the MPC and the optimal control is added for the calculation of the control signal. Moreover, the PSO seeks the best MPC weight parameters guaranteeing desired performances of our output with a minimum of control signal. The numerical results for two physical examples show the effectiveness of our proposed method.**

***Keywords*— Takagi-Sugeno fuzzy model, model predictive control. particle swarm optimization, optimal control.**

I. INTRODUCTION

During the last decades, numerous successful applications of the technology because of the use of model predictive controller (MPC). The MPC is an advanced control [1], in which an optimization procedure is performed to calculate optimal control actions. It uses a model of the process explicitly to obtain the control signal which minimizes an objective function. In the predictive group, the generalized predictive control (GPC) is the most widely used method in power plant control ([2]; [3]). For instance, a nonlinear GPC based on neuro-fuzzy network for controlling the superheated steam temperature of a 200 MW power plant [4]. Several other approaches were proposed in the literature to solve the predictive control problem. Some of them are based on numerical optimization algorithms. Some others are based on optimization algorithms such as the genetic algorithms (GA), Ref. [5] proposed a hybrid predictive control strategy to regulate the temperature of a batch reactor by minimizing both the trajectory error and the control energy based on GA. However, GA has the disadvantage of premature and slow convergence rate, and needed many parameter settings. Recently, many studies have proposed the evolutionary computation technique based on Particle Swarm Optimization (PSO) [6, 7]. They have been successfully applied to solve various optimization problems. Indeed, Rfe. [8] presents a predictive controller based on recursive linear models. Where the optimization problem is solved using the PSO algorithm. In the same context, the PSO have been applied successfully to optimize the control law of a multivariable generalized predictive control ([9, 10, 11].In These papers, the PSO is used to finally tune predictive control (NMPC) law of obtained by minimizing a defined objective function. In other methods, PSO solve the problem of calculating the best MPC weight parameters (MPC-PSO) [12]. This motion keeps the principle of calculation of the control signal. But, using this method results in several disadvantages especially for nonlinear systems. The MPC is a method of designing control with feedback by solving an optimal control problem from the present time point, and using the output values for resolution of the problem at the next time point. So, Firstly, it is difficult to obtain a mathematical expression for MPC feedback performance. Secondly, In any case, feedback performance depends greatly on the optimal control parameters.

In this paper, Firstly, for the modelling phase, the T–S fuzzy model is employed to approximate the nonlinear system. Secondly, we treated the MPC problem who's the optimal control problem is solved repetitively online using the values of our system, which might be interpreted as the inverse problem of finding the parameters of the optimal control problem so as to obtain desirable plant outputs. And we proposed a parameter optimization method for optimal control to improve feedback performance.

The remainder of this paper is organized as follows. In Section 2, a brief overview of the MPC system and its parameter tuning. The Formulation of MPC parameter optimization problem is detailed in Section 3 with two examples to show the effectiveness of our proposal for two linear systems in Section 4. In the section 5, a proposed algorithm with PSO for parameter optimization of MPC system. An applying our algorithm on nonlinear systems in section 6. Simulation results and conclusion are given in section 7 and section 8 respectively.

2. THE MPC SYSTEM AND ITS PARAMETER TUNING

In this work, we formulate a feedback system using MPC and the problem of parameter optimization. For this, we considered the variable k as a discrete time. Assume that the

ISSN 2356-5608

model of controlled object is given at the kth time point along with the detected state quantity $x(k)$.
Then we applied the principle of optimal control for finding the optimal control input series $\hat{u}(.|k) = \hat{u}(i|k)$ $\{i = 1,2,...,I\}$ for a finite time interval I. So the corresponding optimal state quantity series $\hat{x}(.|k) = \hat{x}(i|k)$ is formulated as follows :

$$\hat{u}(.|k) = \arg\min_{u(.|k)} \sum_{i=0}^{I} F(x(i|k), u(i|k), w) \quad (1)$$

$$x(i+1|k) = \hat{f}(x(i|k), u(i|k)) \quad i = 1,2,...,I \quad (2)$$

Here $u(i|k)$ and $x(i|k)$ denote the input to the plant model and the plant state quantity at instant $k+i$, F denote the performance function, it included the parameters w. The problem is solved by using the detected state quantity $x(k)$ of the system at constant as the initial state $x(0|k)$ and only $u(0|k)$ is applied as the input $u(k)$ to the process. This procedure is repeated at $k = 1,2,...,K$. The relationship between the input series $u(.)$ thus obtained and the observed state quantity series $x(.)$ of model is described for convenience as follows:

$$x(k+1) = f(x(k), u(k)) \quad k = 1,2,...,K \quad (3)$$

Here we assume that the initial state $x(1)$ is given at the start of control. This repetitive procedure of solution of the optimal control problem, and the inputting of the initial value of the optimal control. The input to process can be expressed by the time series shown in Figure 1. The process of generation of the input $u(.)$ to the system corresponding to its detected state $x(.)$ can be expressed using the operator G for solving the optimal control problem:

$$u(k) = G(x(k), w) \quad (4)$$

The feedback function $G(x(k), w)$ depends on the time-varying parameters w of optimal control. The above is the principle of feedback control using MPC. Figure 2 shows the principle of feedback control MPC using an operator $G(x(k), w)$. As can be seen from the diagram, the feedback function depends on the time-varying parameters w of optimal control.

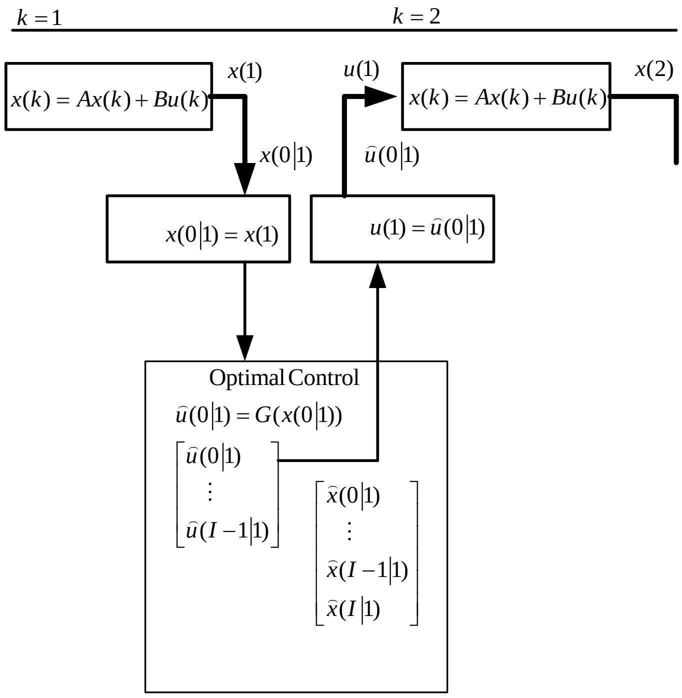

Fig. 1. Time series of MPC

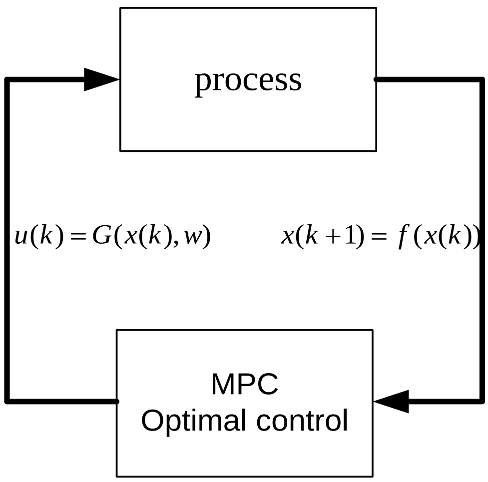

Fig 2. Feedback process of model predictive control

3. FORMULATION OF MPC PARAMETER OPTIMIZATION PROBLEM

The feedback function of the MPC system described above depends strongly on the parameters w included in the optimal control problem. For this purpose, the system model is often made linear, and the performance function is set in quadratic form:

$$F(x,u,w) = \sum_{i=1}^{n} w_{xi}(x_i - r_{xi})^2 + \sum_{i=1}^{m} w_{ui}(u_i - r_{ui})^2 \quad (5)$$

ISSN 2356-5608

Conférence Internationale en Automatique & Traitement de Signal (ATS-2017)
Proceedings of Engineering and Technology – PET
Vol.22 pp.53-60

$$\hat{f}(x,w) = Ax + Bu \quad (6)$$

In this case, w is a parameter vector composed of weights $w_{xi}, i = 1,2,...,n$ and $w_{ui}, i = 1,2,...,m$ of performance function F and a is a parameter vector composed of elements of matrices A, B of linear model $\hat{f}$. In order to improve the characteristics, the parameters w of optimal control must be tuned accordingly. The problem of parameter optimization can be formulated as follows:

$$\min_{w} J\left[x(.)\right]()$$

$$x(k+1) = f(x(k), u(k)) \quad k = 1,2,...,K \quad (7)$$

$$u(k) = G(x(k), w) \qquad k = 1,2,...,K \quad (8)$$

The performance function J is a functional that evaluates the characteristics of the state $x(.)$ in the feedback system with model predictive controller. However, this state quantity series $x(.)$ itself depends on the parameters (w) of optimal control problem that determines the feedback function.
So, this problem (5) can be formulated in terms of optimization of the parameters w as follows:

$$\min_{w} J\left[x(., w)\right] = \Phi(w) \quad (9)$$

The function Φ is a composite function that handles the values of the functional J as a direct function of parameters w . Therefore, now let us assume that the performance function J for the plant state quantity $x(.)$ evaluates the overshoot ratio M_i , the steady-state error E_i , the settling time Ks_i , and the rise time kr_i for every component xi of the state quantity.

$$J\left[x(.)\right] = \sum_{i=1}^{n} q_{1i} E_i \left[x(.)\right] + q_{2i} M_i \left[x(.)\right] + q_{3i} ks_i \left[x(.)\right] + q_{4i} kr_i \left[x(.)\right] \quad (10)$$

where $\left[q_{1i}, q_{2i}, ..., q_{ni}\right]$ $(i = 1,2,...,n)$ are the weights of the respective performance indices. We assume that the operation of a system can be repeated multiple times with the parameter w fixed at a trial value.

4. EXAMPLE

$$x(k+1) = 0.95x(k) + 0.5u(k) \quad k = 1,2,... \quad (11)$$

The performance function F for optimal control problem aims at a steady state of $x = 10$ at the input $u = 1$.

$$F(x,u,w) = w_x (x-10)^2 + w_u (u-1)^2 \quad (12)$$

$$u(k) = (200w_u + 40w_x - 19w_u x(k-1) / x(k-1)) / (10w_u + 40w_x) \quad (13)$$

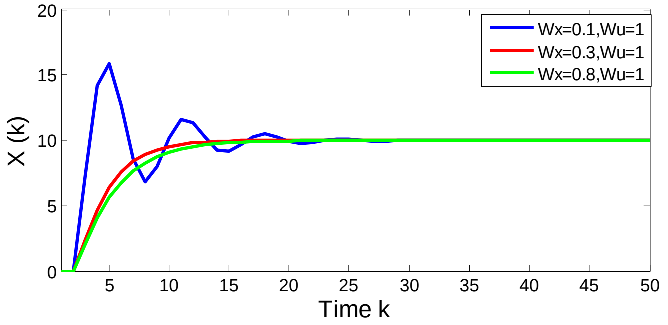

Fig 3. Outputs of system in terms of weight parameter

Figure 3 shows the state transition of the feedback loop when MPC is applied to our example for various weight parameters of the performance function. As can be seen from the diagram, the overshoot, settling time, and rise time vary strongly with the weight parameters (w_x, w_u) of the optimal control problem in MPC.

5. USING A PSO PARAMETER OPTIMIZATION OF MPC SYSTEM

The PSO algorithm is a population-based search algorithm [13, 14, 15]. The very simple behavior followed by individuals emulates their own successes and the success of neigh boring individuals. The emergent collective behavior is that of discovering optimal regions of a high dimensional search space. In a PSO algorithm, each particle representing a potential solution is maintained within a swarm where the position of each particle is adjusted according to the experience of itself and its neigh bourse. The cost function used in this paper is given by equation (10) for tune the weight parameter w of the evaluation function for the optimal control problem by setting upper and lower limits to restrict the range of the weight parameters.

$$\min_{w} J(w), \; w_{\min} \le w_i \le w_{\max}, \quad i = 1,2,...,n+m \quad (14)$$

Here, the parameter w includes $w_{xi}, i = 1,2,...,n$ and $w_{ui}, i = 1,2,...,m$. In PSO, all of the particles iteratively discover the probable solution. Each particle moves to a new position according to the new velocity which includes its previous velocity, and the moving vectors according to the past best solution and global best solution. The best solution is then kept. Each particle in the swarm is iteratively updated according to the aforementioned attributes. Assuming that the

ISSN 2356-5608

cost function J is to be minimized, the new velocity of every particle is updated by:

$$v_{id}(t+1) = wv_{id}(t) + r_1c_1(Pb_{id}(t) - X_{id}(t)) + r_2c_2(Gb_d(t) - X_{id}(t)) \quad (15)$$

For all $j \in 1,...,N$, N is the dimension number, v_{ij} is the velocity of the j[th] dimension of the i[th] particle, p_{ij} is the particles' current position Pb_{ij} is local best position for particles the c_1 and c_2 denote the acceleration coefficients, r_1 and r_2 are elements from two uniform random sequences in the range (0, 1). The inertia weight ω is random numbers in the range (0, 1), and t is the number of generations. The new position of a particle is calculated as follows:

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (16)$$

The local best position of each particle is updated by:

$$Pb_i(t+1) = \begin{cases} Pb_i(t) \text{if } J(X_i(t+1)) \geq J(Pb_i(t)) \\ p_i(t+1), \text{ otherwise} \end{cases} \quad (17)$$

And the global best position Gb found from all particles during the previous three steps is defined as:

$$Gb(t+1) = \arg\min_{Pb_i} J(Pb_i(t+1)) \quad (18)$$

It assures that the global best position will not vanish during evolutionary process.

6. THE MPC SYSTEM AND ITS PARAMETER FOR NONLINEAR SYSTEM

a. FUZZY MODEL

A discrete-time nonlinear system can be described as follows:

$$\begin{cases} x(k+1) = F(x(k), u(k)) \\ y(k) = H(x(k)) \end{cases} \quad (19)$$

where $u(k) \in R$ and $y(k) \in R$ are the system input and output at time k respectively,

$x(k) \in R^n$ is the state vector of the system, $F \in R^n$ and $H \in R$ are nonlinear functions. The nonlinear system is decomposed into r subsystems such that each subsystem demonstrates a linear or nearly linear behavior. Using the T-S modeling methodology [16, 17, 18, 23], a fuzzy quasi-linear model, R_i or fuzzy implication, is developed for each subsystem. In such a model, the cause-effect relationship between control u and output y at sampling time k is established in a discrete time representation. The subsystems are defined in the fuzzy regions. R_i, $i = 1, 2, ..., n$. For both a controllable and observable system, $x(k)$ can be expressed as function of $y(k), \ldots, y(k-n+1), u(k), \ldots, u(k-n+1)$ and n represents the order of the system. Therefore, when the nonlinear system (1) is investigated around the origin, its equivalent system can be expressed as follows [19] :

$$y(k) = -a_1 y(k-1) ... - a_{na} y(k-na-1) + b_1 u(k-1) ... + b_{nb} u(k-nb) = \xi(k)^T \theta \quad (19)$$

$$\xi(k) = \left[y(k-1), ..., y(k-na-1), u(k-1), ..., u(k-nb)\right]^T \quad (20)$$

$\theta = \left[a_1, ..., a_{na}, b_1, ..., b_{nb}\right]^T$ is the referred to as the regression vector is. System (19) can be rewritten as Controlled Auto-Regressive Integrated Moving Average model (CARIMA) [20]

$$A(z^{-1}) y(k) = B(z^{-1}) u(k) \quad (21)$$

where $A(z^{-1})$ and $B(z^{-1})$ are polynomials in the backward shift operator z^{-1}

$$\begin{cases} A(z^{-1}) = 1 + a_1 + a_2 + ... + a_{na} \\ B(z^{-1}) = b_1 + b_2 + ... + b_{nb} \end{cases} \quad (22)$$

A T-S model consists of a set of fuzzy rules, each describing a local input-output relation as follows:

$$R^i : if\ y(k)\ is\ M_1^i, y(k-1)\ is\ M_2^i ...\ y(k-na-1)\ is\ M_{na}^i, u(k)\ is\ L_1^i, ...,\ u(k-nb)\ is\ L_{nb}^i \quad (23)$$

$$\text{THEN } y_i(k) = \frac{B^i(z^{-1})}{A^i(z^{-1})} u(k) \quad i = 1, ..., r$$

where M_j^i fuzzy set is corresponding to output $y(k-j)$ in the i^{th} FI, L_p^i fuzzy set is corresponding to output $u(k-p)$ in the i^{th} FI.

The system output $y(k+1)$ is computed as the weighted average of the individual rules consequents:

$$y(k+1) = \frac{\sum_{i=1}^{r} \mu_i u(k) \frac{B^i(z^{-1})}{A^i(z^{-1})}}{\sum_{i=1}^{r} \mu_i} \quad (24)$$

μ_i is the normalized membership function of the inferred fuzzy set M^i where $M^i = \prod_{j=1}^{na} M_j^i$ and $\sum_{i=1}^{r} \mu_i = 1$.

The computing of the values for r model parameters in Eq. 3 is obtained by using weighted recursive least squares method to the N sample data ($\{x(k), y(k)\}$ as follows [21]:

$$\theta_i(k) = \theta_i(k-1) + L_i(k)\left[y_i(k) - \zeta(k)\theta_i^T(k-1)\right] \quad (25)$$

ISSN 2356-5608

$$Q_i(k) = \frac{P(k-1)\zeta^T(k)}{1/\mu_i + \zeta(k)P(k-1)\zeta^T} \quad (26)$$

$$P_i(k) = P_i(k-1) - Q_i(k)\zeta(k)P_i(k-1) \quad (27)$$

for $k = 1,...,N, P(k-1)$ is a covariance matrix and $Q(k)$ referred to the estimator gain vector. A common choice of initial value is to take $\theta_i(0) = 0$ and $P_i(0) = \alpha I$ where α is a large number. The synthesis of the proposed fuzzy controller is based on the architecture of the obtained fuzzy model. The controller has the following form [22]:

$$u(k) = \frac{\sum_{i=1}^{r} \mu_i u_i(k)}{\sum_{i=1}^{r} \mu_i} \quad (28)$$

b. ALGORITHM

The proposed predictive control algorithm is summarized as follows:

Phase 1: Construction of the fuzzy model using FCM algorithm

Step1. Given data $S = \{(x_1, y_1),...,(x_k, y_k)\}$ $k = 1,...,N$, set $m > 1$ and the metric matrix $A = I$. Select a termination threshold $\varepsilon > 0$ and initialize U^0 (e.g. random).
Repeat for $l = 1,2,...$

Step2. Calculate the cluster centers as follows:

$$v_i^l = \frac{\sum_{k=1}^{N} (\mu_{ik}^{l-1})^m x_k}{\sum_{k=1}^{N} (\mu_{ik}^{l-1})^m} \quad i = 1,2,...,r \quad k = 1,2,...,N \quad (29)$$

m is the fuzzy weighting exponent.

Step 3. Calculate distances as follows:

$$\rho_{ik} = (x_k - v_i^l)^T A (x_k - v_i^l) \quad (30)$$

Step 4. Update U^l with ρ_{ik} satisfy:

$$U_{ik}^l = \begin{cases} \mu_{ik}^l = \dfrac{1}{\sum_{j=1}^{r} (\frac{\rho_{ik}}{\rho_{jk}})^{\frac{2}{m-1}}} & if\ \rho_{ik} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

Until $\left\| U^l - U^{l-1} \right\| < \varepsilon$ then stop. Otherwise, set $l = l + 1$ and return to **Step** 2.

Step 5. Calculate values for r model parameters θ_i^r using WRLS method.

Phase 2: Parameter optimization for model predictive control

Step 0. The weight parameter $w(0)$ is specified, Initialize the particle swarm and $l = 0$ is set.

Step. 1. Calculate the plant state for each cluster according to Eq. (7) with a fixed initial state $x_i(1)$ $i = 1,...,r$.

Step 2. The evaluation score $J_i\left[x_i(.,w_i(1))\right]$ for each cluster is calculated.

Step 3. We use the PSO updating formula (17) and (18) with regard to the upper and lower constraints on the weight parameters.

$$w^{pbest} = Pb = \arg\min_{w^p} \left\{ \Phi(w^p(h)) \middle| h = 1,2,...,l \right\} \quad (32)$$

$$w^{gbest} = Gb = \arg\min_{w^{pbest}} \left\{ \Phi(w^{pbest}) \middle| p = 1,2,..,P \right\} \quad (33)$$

Step 4. Calculate the control state for each subsystem.

$$u_i(k) = G_i\left(x_i(k\text{-}1), w_i^{gbest}, i = 1,...,r \ \ k = 1,...,N\right) \quad (34)$$

Step 5. The trial-and-error search is finished when the maximum number of iterations $l_{\max}$ is met.

Step 6. Calculate the overall control according to Eq. 31

7. SIMULATION RESULTS

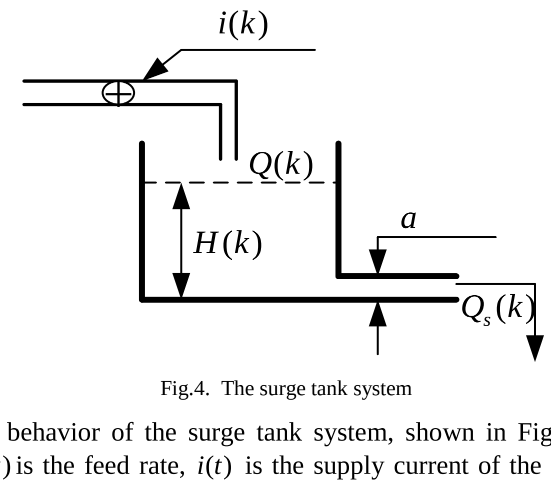

Fig.4. The surge tank system

The behavior of the surge tank system, shown in Figure 4, $Q(k)$ is the feed rate, $i(t)$ is the supply current of the pump, $H(k)$ is the liquid level in the tank, $Q_s(k)$ is the output Flow, a is the section of the output channel, This system can be represented by the following differential equations:

$$Q(k) = Q(k-1) + Te(-k_0 Q(k-1) + k_1 i(k-1)) \quad (35)$$

The change in water level in the tank is given by:

Conférence Internationale en Automatique & Traitement de Signal (ATS-2017)
Proceedings of Engineering and Technology – PET
Vol.22 pp.53-60

$$V(k) = AH(k) = H(k-1) + Te(Q(k-1) - Q_s(k-1)) \quad (36)$$

where :

$$Q_s(k) = 0.6a\sqrt{2g(H(k) - H_s)} \quad (37)$$

A is the section of the tank and H_s is the water level in the output channel.

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	$0.0001 \; m^3 s^{-1}$
k_0	Constant	1
k_1	Constant	0.1

We suppose that the subsystems are in the third order. We choose the degree of $A(z^{-1})$ and $B(z^{-1})$ respectively $na = 3$, $nb = 2$, the predictive step size is Ny =8, while the control step size is Nu =5. The iteration number for chaos searching is 75 and the iteration number for PSO searching is 350.

The model consists of two rules of the form:

$$R^1: \text{IF} \quad i_1 \text{ is } Q^1 \quad \text{THEN } H^1(k) = a_{11}H(k-1) + a_{12}H(k-2) + a_{13}H(k-3) + b_{11}i_1(k-1) + b_{12}i_1(k-2)$$

$$R^2: \text{IF} \quad i_1 \text{ is } Q^2 \quad \text{THEN } H^2(k) = a_{21}H(k-1) + a_{22}H(k-2) + a_{23}H(k-3) + b_{21}i_1(k-1) + b_{22}i_1(k-2)$$

For the analysis of this behavior, the reference signal changes as follows:

$$y_r(k) = \begin{cases} 0.2 & 0 < k \leq 30 \\ 0.1 & 30 < k \leq 60 \\ 0.3 & 60 < k \leq 90 \\ 0.4 & 90 < k \leq 1200 \end{cases}$$

The performance function selected for each cluster as follow :

$$F_i(x,u,w) = w_{xi}(H_i - y_r)^2 + w_{ui}(u_i)^2, \; i = 1,2 \quad (38)$$

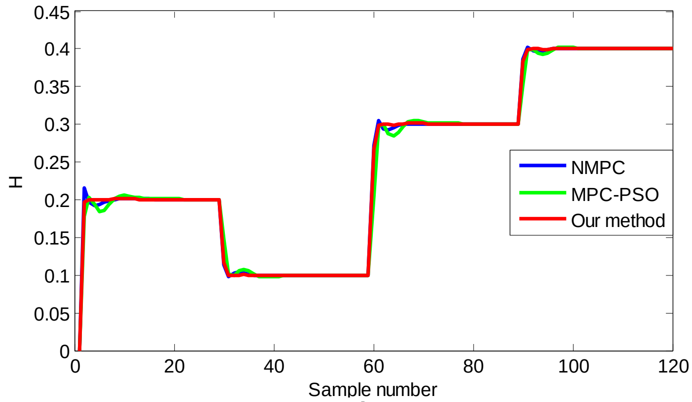

Fig 5. Surge tank response

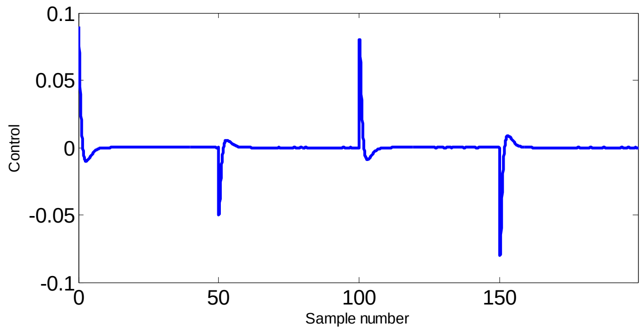

Fig 6. Control signal

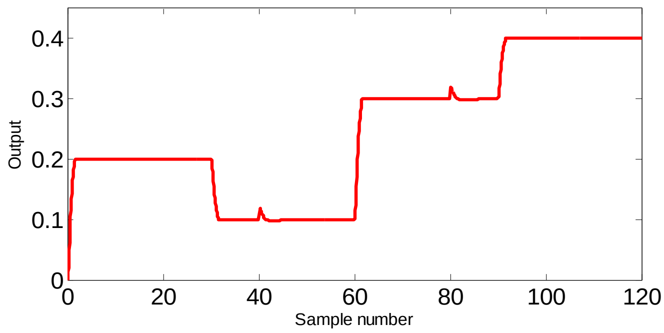

Fig 7. System response with disturbance of our method

Table 3: Comparison of performance of the MPC, MPC-PSO and our method

	Yousuf et all (2010)	Paul Mc Namara(2013)	Our method
RiseTime	0.8111	0.9820	0.7402
SettlingTime	1.9936	11.0788	5.6402
Overshoot	0.8388	2.6922	8.0804
Peak	0.2017	0.2055	0.2162
ESS	0.00	0.00	0.00

Table 3 shows the performances obtained by each method NMPC (Yousuf et all, 2010), MPC-PSO (Paul Mc Namara, 2013) and our method. In each interval time, we have changed the reference for evaluating each method to control our nonlinear system. Comparison was made between the proposed method and the other methods shows the results during the 120 iterations (Figure 5). The red line denotes

ISSN 2356-5608

the output of the T–S fuzzy model based on our method. The green and blue lines denote the outputs of the T–S fuzzy models based on MPC-PSO and NMPC, respectively. The proposed method shows obvious advantages over that of old methods. A comparison of control results between the three strategies are demonstrated in figure 5. The simulation results of different methods are given in Table 3. Same, these results demonstrate the superiority of our MPC method. So, one can see that the overshoots of the new method are much smaller than that of the other methods (NMPC, MPC-PSO), it is less than 30% compared to MPC-PSO and almost lower 10% compared to NMPC. And we can note also that the performance of our method is better than those of other methods in terms of Rise time, setting time, overshoot and peak time. Figure 6 shows the control signal obtained with our method. Besides, this control input is very smooth. Figure 7 shows that the algorithm has an enormous capacity for disturbance rejection. All these previous results clearly indicate that the proposed controller outperforms the other methods (NMPC and MPC-PSO) taking into account the responses by changing reference signals. Therefore, we can see that our proposed controller still holds the best performance with and without disturbance. Thus, it confirms the usefulness and robustness of this proposed controller.

8. CONCLUSION

This paper presents a model predictive controller MPC based on particle swarm optimization for T-S fuzzy modeling. In particular, we proposed a method of tuning the weight parameters based on the optimal control and of the performance function according to the output state quantities detected from the response of the system. The results showed that the proposed approach is efficient to tune MPC controller for nonlinear system. The effectiveness of the proposed controller has been tested in comparison with MPC-PSO and NMPC methods through the simulation studies of two benchmark problems. Results analysis show that the proposed method led to an improvement of the system characteristic such as overshoot, settling time and system response speed than the other recently reported methods in literature.

REFERENCES

[1]. Maciejowski, J. (2002), Predictive Control with Constraints, Prentice-Hall. Harlow, England.

[2]. Peng, H., Ozaki, T., Haggan, V. and Toyoda, Y. (2002), A nonlinear exponential ARX model-based multivariable generalized predictive control strategy for thermal power Plants, IEEE Transactions on Control Systems Technology, China, Vol. 10, no. 2, pp. 256–262.

[3]. Moelbak, T. (1999), Advanced control of superheater steam temperatures an evaluate on based on practical applications, Control Engineering Practice, vol.7, no.1, pp. 110-117.

[4]. Liu, X. and Chan, C. (2006), Neuro-fuzzy generalized predictive control of boiler steam temperature, IEEE Transactions on Energy Conversion,vol. 21, no. 4, pp. 900–908.

[5]. Causa, J. et al , (2008), Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor, Computers and Chemical Engineering, Vol. 32, No. 12, pp. 3254-3263.

[6]. Soltani, M., Chaari, A. and Benhmida, F. (2012), A novel fuzzy c regression model algorithm using new measure of error and based on particle swarm optimization, International Journal of Applied Mathematics and Computer Science, Vol. 22, No. 3, pp. 617-628.

[7]. Soltani, M. and Chaari, A. (2013), A novel weighted recursive least squares based on Euclidean particle swarm optimization, Kybernetes, vol. 42, no.2, pp. 268-281.

[8]. Coelho, L. and Mariani, S. (2009), A novel chaotic particle swarm optimization approach using Hénno map and implicit filtering local search for economic load dispatch, Chaos, Solutions and Fractals, vol. 39, no. 2, pp. 510–518.

[9]. Yousuf, M., Hussain, N. and Hamouz, Z. (2010), PSO based predictive nonlinear automatic generation control, Proceedings of the 12th WSEAS international conference on automatic control, modeling and simulation, USA, vol. 1, no. 4 pp. 87-92.

[10]. Duwaish, H. and Rizvi, S. (2010), PSO based Hammerstein modeling and predictive control of nonlinear multivariable boiler, Control Engineering practice, pp. 1-12.

[11]. Li, Z. (2010), Intelligent Fuzzy Predictive Controller Design for Multivariable Process System, Journal of Computational Information Systems, Vol. 6, No. 9, pp. 3003-3011.

[12]. Namaraa, P. Negenbornb, R. Schutter, B. and Gordon, L. (2013), Light body a Weight optimization for iterative distributed model predictive control applied to power networks", Engineering Applications of Artificial Intelligence, vol. 26, pp. 532–543.

[13]. Kenned, J. and Eberhart, R. (1995), Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948.

[14]. Shi, Y. and Eberhart, R. (1998), Parameter selection in particle swarm optimization, Evolutionary Pro-gramming", VII, LNCS, vol.1447, pp.591-600.

[15]. Yang, S. (2010), An Introduction with Metaheuristic Application, John Wiley & Sons, 2010.

[16]. Takagi, T. and Sugeno, M. (1985), Fuzzy identification of systems and its application to modeling and control, IEEE Transactions on Systems, Man and Cybernetics, vol.15, no.1, pp.116-132.

[17]. Espinosa, J., Vandewalle, J. and Wertz, V. (2005), Fuzzy Logic, Identification and Predictive Control, Springer, London, UK.

[18]. Bououden S, Filali S and Kemih K (2010), Adaptive fuzzy tracking control for unknown nonlinear systems", International

ISSN 2356-5608

Journal of Innovative Computing, Information and Control, vol. 6, no.2, pp.541-549.

[19]. Chenand, L. and Narendra, K. (2004), Identification and control of a nonlinear discrete-time system based on its linearization: A unified framework", IEEE Transactions on Neural Networks, vol.15, no.3, pp.663-673.

[20]. Boucher, P. and Dumur, D. (1996), La Commande Predictive", Editions Technip, Collection Methodes et Pratiques de l'Ingenieur, Paris.

[21]. Kung, C. and Su, J. (2007), Affine Takagi-Sugeno fuzzy modelling algorithm by fuzzy c-regression models clustering with a novel cluster validity criterion", IET Control Theory Application, Vol. 1 No. 5, pp. 1255-1265.

[22]. Lagrat et al (2008), Optimal adaptive fuzzy control for a class of unknown nonlinear systems, WSEAS Transactions on Systems and Control, vol. 3, no. 1, pp. 89-98.

[23]. Huang, C. (2012), Model based fuzzy control with affine T-S delayed models applied to nonlinear systems", International Journal of Innovative Computing, Information and Control, vol.8, no. 5(A), pp.2979-2993.

ISSN 2356-5608