

Nonlinear model predictive control based on fuzzy wavelet neural network and chaos optimization

Mehrnoush DAVANIPOUR*, Hanieh ASADIPOOYA
School of Electrical Engineering, Shiraz University, Shiraz, Iran

Received: 18.12.2015

Accepted/Published Online: 22.03.2017

Final Version: 05.10.2017

Abstract: In this paper a combined controller is proposed for nonlinear dynamical systems. The controller is constructed by a fuzzy wavelet network and nonlinear model predictive control. Chaotic optimization, which is fast and robust, is applied to generate optimized controlled input in nonlinear model predictive control. The ability of the fuzzy wavelet neural network and the proposed controller is shown by simulation. It is illustrated that the proposed method is able to increase the speed of tracking in addition to having very little steady state error. Using chaotic optimization makes the controller robust in the presence of noise.

Key words: Nonlinear model predictive control, fuzzy wavelet neural network, chaos optimization

1. Introduction

Nowadays nonlinear model predictive control (NMPC) including a great degree of challenges has attracted many researchers in the control area [1–8]. There are two requirements for NMPC to be designed successfully. On one hand a strong model is needed to be considered as a predictor and on the other hand a fast optimization algorithm capable of performing real-time optimization should be applied [1].

One of the successful strategies for dealing with the challenges related to NMPC is using intelligent networks [2–6]. In [3] an adaptive neural network was used to compensate for the error made by parameter variations. The method was suitable for model predictive control of nonlinear systems with parameter variations and uncertainties but the optimization method applied was a simple gradient-based method, which is slow, nonrobust in noisy conditions, and including high computation. In [4] a neurodynamic approach to NMPC based on two recurrent neural networks was presented. In that work a nonconvex optimization problem was decomposed via Taylor expansion and the high order unknown terms were estimated with an online supervised learning algorithm. In spite of the good performance of their controller, the complexity of model computation volume is very high since it needs to use Jacobian linearization. In [6] adaptive stable generalized predictive control using a fuzzy model was proposed to control nonlinear discrete-time with time-delay systems. The nonlinear model is constructed by some locally linear models so that GPC can be used as the predictive controller. Even though that method was able to capture the advantages of GPC, it could not cope with all of the nonlinearities in the system. As mentioned in [7] a combination of fuzzy logic, neural network, and wavelet theory could be a strong model capable of dealing with highly nonlinear systems. Thus it would be a suitable choice as the predictor in NMPC. While using this strong model they applied a gradient descent algorithm, which requires a long time and high computation for optimization.

*Correspondence: m.davanipour@shirazu.ac.ir

One of the modern optimization algorithms is chaos-based optimization [9–14]. Chaos is a kind of characteristics of nonlinear systems. It is a limited unstable dynamic behavior that represents sensitive dependence on initial conditions and includes infinite unstable periodic motions [9]. Chaos optimization is a robust mechanism that can escape from local minima and so it is suitable for global optimization. In [12], a stepped up chaos optimization algorithm has been used in to achieve maximum efficiency in photovoltaic systems. A combined algorithm that includes chaos optimization, particle swarm method, and local search has been applied in [13] in order to find the best subset of features. In [14], the search capability of binary particle swarm optimization has been upgraded using the chaotic search approach.

In the present paper a fuzzy wavelet neural network capable of capturing high nonlinearities of systems is used as a model in NMPC. It is seen that the FWNN can represent a good model for use in NMPC. However, as mentioned, another main problem about NMPC is having an effective optimization method that is suitable for online applications. Therefore, here chaos-based optimization, which is global, fast, and robust in noisy conditions, is applied to find the optimum control inputs. The simulation results show the very good and fast performance of the proposed method.

This paper is organized as follows: Section 2 includes NMPC. Section 3 gives a brief description of FWNN. In Section 4 optimization methods are completely described. In the next section, simulation results are provided. Section 6 includes concluding remarks.

2. Nonlinear model predictive control

In the NMPC, the controller is designed by minimizing a cost function as below [15]:

$$J(k) = \frac{1}{2} \sum_{j=1}^{N_p} (r(k+j) - y(k+j))^2 + \frac{1}{2} \sum_{j=1}^{N_u} q_j \Delta u^2(k+j-1), \tag{1}$$

in which N_p is the predictive output horizon, N_u is the control horizon, q_j is the weight, $r(k)$ is a known reference input, and the index j represents the future times. In situations where there is no model of the system, the exact values of $y(k+j)$ are not available. Thus, it is necessary to use an estimation $\hat{y}(k+j)$ instead. This estimation is obtained here using FWNN, which is described in the next section. To design the controller, the cost function in (1) is expressed in the vector form as follows:

$$\hat{J}(k) = \frac{1}{2} \hat{E}^T(k) \hat{E}(k) + \frac{1}{2} Q \Delta U^T(k) \Delta U(k), \tag{2}$$

where

$$\hat{E}(k) = [\hat{e}(k+1), \hat{e}(k+2) \dots \hat{e}(k+N_p)]^T \tag{3}$$

$$\hat{e}(k) = r(k) - \hat{y}(k) \tag{4}$$

$$Q = \text{diag}[q_1, q_2, \dots, q_{N_u}]^T \tag{5}$$

$$\Delta U(k) = [\Delta u(k), \Delta u(k+1) \dots \Delta u(k+N_u-1)]^T \tag{6}$$

The control vector is obtained by minimizing the cost function (2). In this paper it is obtained using chaos optimization, which is described in the next sections. The schematic of the proposed method is shown in Figure

1. In the figure, the unknown plant is modeled by FWNN, which will be described in the next section. In fact, the identification error shown in the figure, points to this procedure. Then the estimated output is applied to the model predictive controller in order to be used in the optimization process.

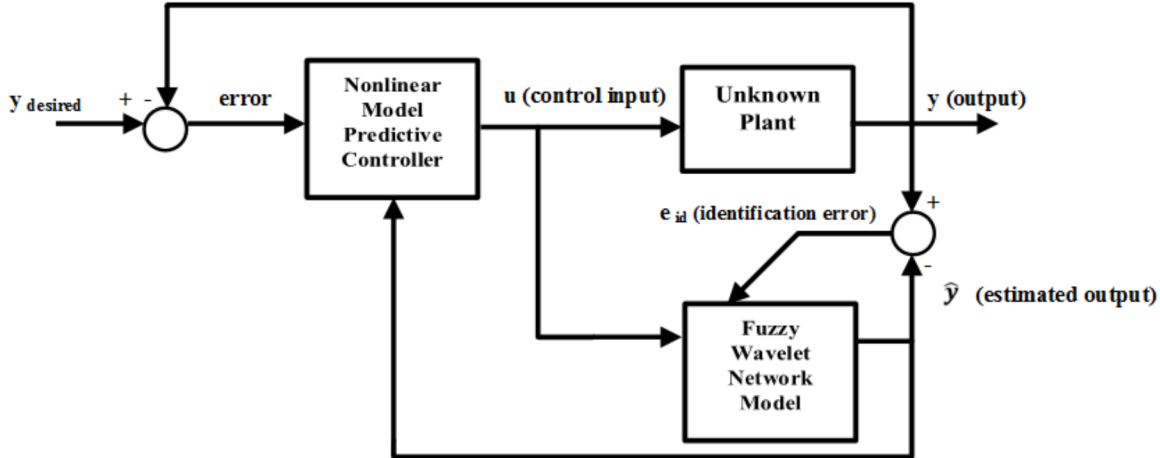


Figure 1. Structure of the proposed controller.

3. Fuzzy wavelet neural network

The structure of the FWNN used in this paper is depicted in Figure 2. It is constructed using some fuzzy rules [16,17]:

$$R^i : \text{If } x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i, \dots, \text{ and } x_q \text{ is } A_q^i \text{ Then } y_i = w_i \sum_{j=1}^q \psi_{ij}(x_j), \tag{7}$$

in which x_j ($1 \leq j \leq q$) and y_i ($1 \leq i \leq c$) is the j th input and the output of the local model for rule R^i respectively, w_i is the weight coefficient from the inputs and i th output, and ψ_{ij} is a wavelet family defined in the following form:

$$\psi_{ij}(x_j) = \psi\left(\frac{x_j - b_{ij}}{a_{ij}}\right) a_{ij} \neq 0 \tag{8}$$

In (7), $\psi_{ij}(x)$ is produced from a single mother wavelet function by dilations and translations (a, b) . According to Figure 2, the output of FWNN is calculated as

$$\hat{y} = \frac{\sum_{i=1}^c \mu_i y_i}{\sum_{i=1}^c \mu_i}, \text{ where } \mu_i(x) = \prod_{j=1}^q A_j^i(x_j) \tag{9}$$

for $i = 1, 2, \dots, c$, $j = 1, 2, \dots, q$. Here c is the number of fuzzy rules and q is the dimension of the input vector. The Gaussian membership functions are selected to describe the linguistic terms.

$$A_j^i(x_j) = \exp\left[-(x_j - c_j^i)^2 / \sigma_j^{i^2}\right] \tag{10}$$

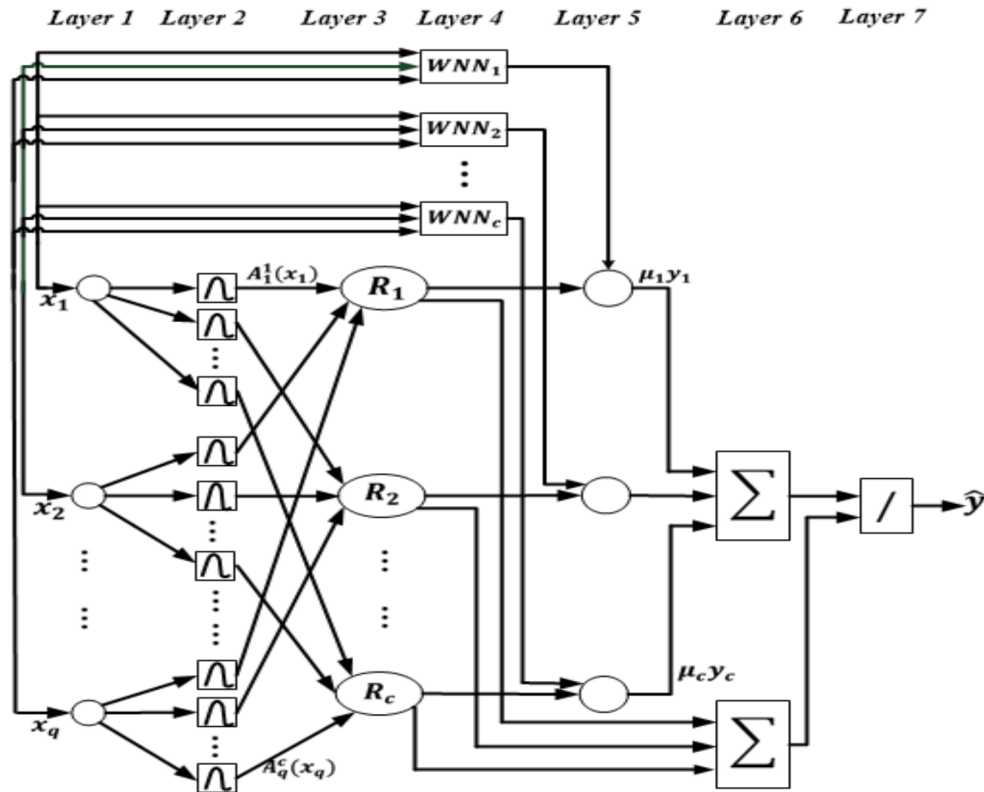


Figure 2. Structure of FWNN.

In (10) σ_j^i and c_j^i determine the center and the half-width of the corresponding membership function. The Mexican hat wavelet function, which is a symmetric function, is used in consequent parts of each fuzzy rule.

$$\psi(x) = \frac{1}{\sqrt{|a|}} (1 - 2x^2) \cdot \exp(-x^2/2) \tag{11}$$

4. Optimization procedure

As discussed in the introduction section, using a fast and strong optimization method is an important factor in NMPC. Here chaos-based optimization, which can escape from local minima, is applied. There is a group of these kinds of algorithms [18]. Here, tent map chaos optimization is applied. To produce chaotic time series the equation of the tent map is as follows:

$$x_{i+1} = \begin{cases} 2x_i x_i & x_i \in [0, 0.5] \\ 2(1 - x_i) x_i & x_i \in [0.5, 1] \end{cases} \tag{12}$$

where x_i is the chaotic variable.

For NMPC the goal of tent map optimization is finding the vector of $[u(t), u(t + 1), \dots, u(t + N_u - 1)]$. For this purpose, it is necessary to assign some initial chaotic variables as $0 \leq x_{1,0}, x_{2,0}, \dots, x_{N_u,0} \leq 1$, which are selected randomly. Then for each variable the lower bound and upper bound are defined and indicated by l_{bd} and u_{bd} . Following that, the initial value for $[u(t), u(t + 1), \dots, u(t + N_u - 1)]$ and E (cost function) should be

determined and denoted as $[u(t)^*, u(t + 1)^*, \dots, u(t + N_u - 1)^*]$ and E^* . The main steps of the FWNN-NMPC based on tent map chaos optimization are explained below:

Step1. At the start, compute the cost function and determine $E([u(t), u(t + 1), \dots, u(t + N_u - 1)])$

Step2. Using tent map chaos optimization, find a new control vector that minimizes the cost function:

- Substitute $x_{1,k}, x_{2,k}, \dots, x_{N_u}, k$ in (12) to produce new chaotic variables by tent map where the index k is the number of steps.
- Compute the $(k + 1)$ th input vector by the following formula, which is called the first carrier wave:

$$\begin{bmatrix} u_{k+1}(t) \\ u_{k+1}(t + 1) \\ \vdots \\ u_{k+1}(t + N_u - 1) \end{bmatrix} = \begin{bmatrix} l_{bd-u(t)} \\ l_{bd-u(t+1)} \\ \vdots \\ l_{bd-u(t+N_u-1)} \end{bmatrix} + \left(\begin{bmatrix} u_{bd-u(t)} \\ u_{bd-u(t+1)} \\ \vdots \\ u_{bd-u(t+N_u-1)} \end{bmatrix} - \begin{bmatrix} l_{bd-u(t)} \\ l_{bd-u(t+1)} \\ \vdots \\ l_{bd-u(t+N_u-1)} \end{bmatrix} \right) \cdot \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ \vdots \\ x_{N_u,k+1} \end{bmatrix} \tag{13}$$

- Compute the cost function in (2) and assign the new values of inputs and cost functions as follows: If $E_{k+1} \leq E^*$ then $E^* = E_{k+1}$ and $[u(t)^*, u(t + 1)^*, \dots, u(t + N_u - 1)^*] = [u_{k+1}(t), u_{k+1}(t + 1), \dots, u_{k+1}(t + N_u - 1)]$, else do nothing. Repeat the above steps until no improvement in the value of the cost function is seen. Then follow the next steps:
- Again use the tent map equation in (12) to produce new chaotic variables. Then

$$\begin{bmatrix} u_{k+1}(t) \\ u_{k+1}(t + 1) \\ \vdots \\ u_{k+1}(t + N_u - 1) \end{bmatrix} = \begin{bmatrix} u^*(t) \\ u^*(t + 1) \\ \vdots \\ u^*(t + N_u - 1) \end{bmatrix} + \beta \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ \vdots \\ x_{N_u,k+1} \end{bmatrix} \tag{14}$$

search by the second carrier wave.

In (14), β is a chaos variable with small ergodic interval. It is normally chosen as a small positive constant, here $\beta = 0.001$.

- Compute the cost function in (2), and assign the new values such as follows:

If $E_{k+1} \leq E^*$ then $E^* = E_{k+1}$ and

$[u(t)^*, u(t + 1)^*, \dots, u(t + N_u - 1)^*] = [u_{k+1}(t), u_{k+1}(t + 1), \dots, u_{k+1}(t + N_u - 1)]$, else do nothing.

Repeat the above steps until no improvement in the value of cost function is seen.

Step 3. Use the obtained optimum values as the vector of controlled input.

Step4. For the next sample time go to step1.

5. Simulation results and discussion

In this section, two examples are considered for testing the suggested method. The software used for simulation is MATLAB and the sampling times considered in both examples are 0.1 s.

Example 1 Consider the nonlinear dynamic plant as below:

$$y(k) = f(y(k - 1), y(k - 2), y(k - 3), u(k), u(k - 1)), \tag{15}$$

in which

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2} \tag{16}$$

and $y(k - 1), y(k - 2), y(k - 3)$ are one-, two-, and three-step delayed outputs of the plant, $u(k)$ and $u(k - 1)$ are current and one step delayed inputs of the plant. As described in the previous sections, the unknown plant has been modeled and identified using FWNN. Here FWNN with only two rules has been constructed. Then the network is trained by the algorithm introduced in [16]. After the learning phase, the below input is used to motivate the trained network:

$$u(k) = \begin{cases} \sin(\pi k/25)k < 250 \\ 1.0250 \leq k < 500 \\ -1.0500 \leq k < 750 \\ 0.3 \sin(\pi k/25) + 0.1 \sin(\pi k/32) \\ +0.6 \sin(\pi k/10) 750 \leq k < 1000 \end{cases} \tag{17}$$

As can be seen from Figure 3 the FWNN has a high capability to identify the plant. Even though the plant includes nonlinear dynamics, FWNN can afford modeling of the unknown plant precisely, thanks to the ability of the wavelets used in the consequent parts of the network. After the identification stage, the proposed controller is applied to the plant. The reference signal, which is a unit step, is tracked by the output as illustrated in Figure 4. Figure 5 also shows the tracking error signal. As these figures show, the tracking is very fast, which is a consequence of both the model predictive controller and the strong optimization algorithm. In addition to the speed of tracking, this combination makes the proposed controller robust before the noise, as displayed in Figure 6. A summarized comparison of the proposed controller with some other methods can be found in Table 1. In the table MAE stands for mean absolute of tracking error. Using the FWNN, let the plant be modeled by fewer parameters in comparison to ANFIS. On the other hand, MPC along with the chaos optimization makes a fast and strong controller in comparison to the PID as the controller structure and the gradient descent as the optimization algorithm.

Table 1. Comparison of this work with other methods.

Method	Number of rules	Number of parameters	Settling time (s)	Overshoot (percentage)	MAE
Chaos-based FWNN-NMPC	2	18	25	6	0.012
Gradient-based FWNN-NMPC	2	18	35	15	0.018
Chaos-based ANFIS-NMPC	25	175	80	30	0.09
Chaos-based FWNN-PID	2	18	50	-	0.02

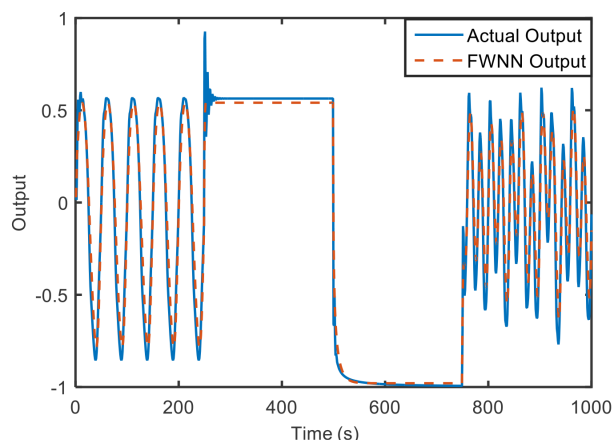


Figure 3. The comparison between the actual output and the output of FWNN (Exp. 1).

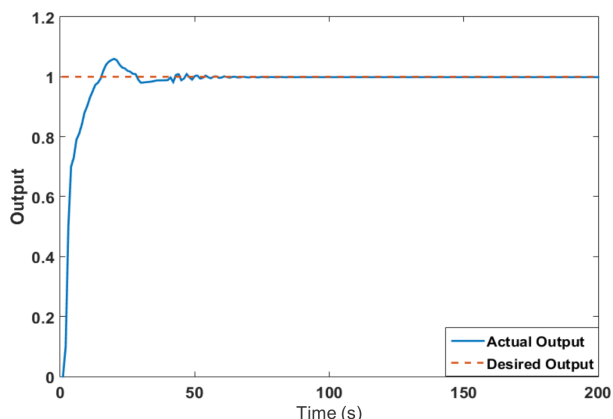


Figure 4. Performance of the proposed controller (Exp. 1).

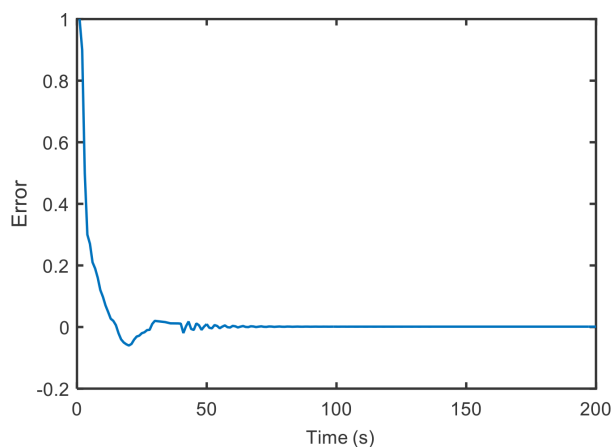


Figure 5. Tracking error signal (Exp.1).

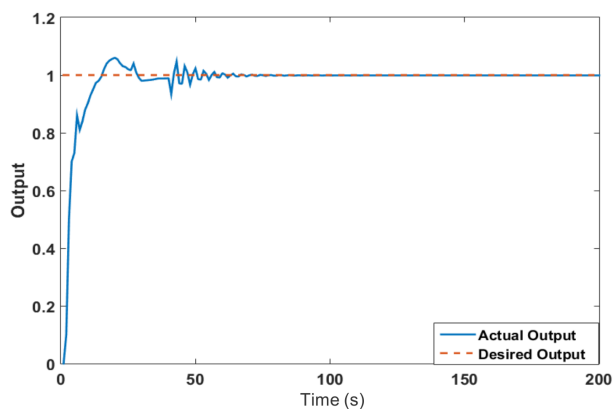


Figure 6. Performance of the proposed controller with output noise contamination (Exp. 1).

Example 2 In this example, a famous liquid level system is considered [19]. The following relations can describe the dynamic of the plant:

$$h(t) = h(0) + \frac{1}{A} \int_0^t (q_{in}(\tau) - q_{out}(\tau)) d\tau, \tag{18}$$

where

$$q_{in} = Q_{in} \sin(\theta(t)), \theta(t) \in \left[0, \frac{\pi}{2}\right] \tag{19}$$

$$q_{out} = a_{out} \sqrt{2gh(t)}, \tag{20}$$

in which h stands for level of the liquid and plays as the output, $a_{out} = 0.01m^2$, $A = 1m^2$, θ is the control valve's flap angle in rad , and $g = 9.81m/s^2$.

In this example, the plant is identified using the fuzzy wavelet network constructed with two rules as well. The result of modeling is depicted in Figure 7. One can see that the FWNN can identified the plant very accurately. Then, by utilizing the NMPC, the output is forced to track the reference signal. Figure 8 illustrates

the performance of the controller. For more details, the error signal is shown in Figure 9. The tracking is fast with very little steady-state error. The performance of the controller in the presence of noise is also verified and shown in Figure 10. The proposed controller and some other kinds of controllers have been applied to this nonlinear liquid level system and the results are summarized in Table 2 for a better comparison.

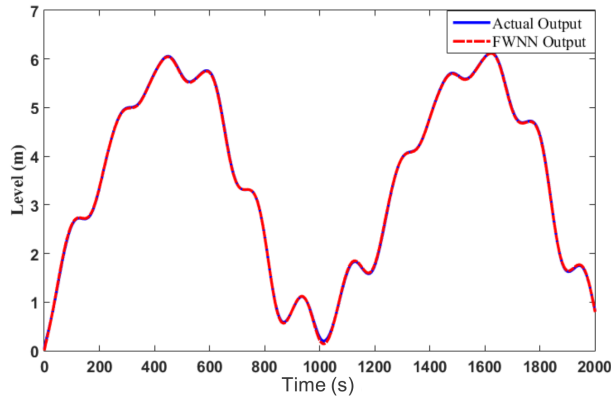


Figure 7. The comparison between the actual output and the output of FWNN (Exp. 2).

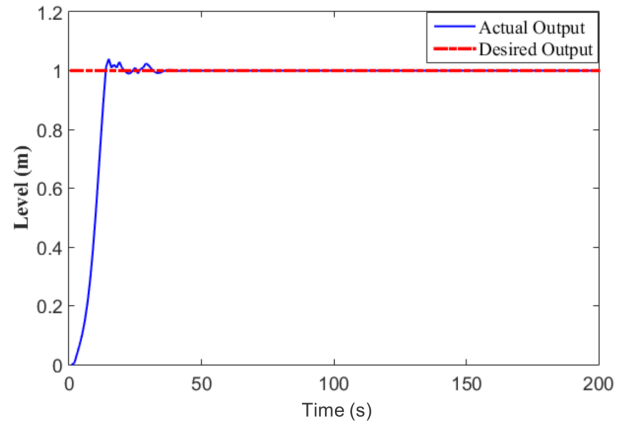


Figure 8. Performance of the proposed controller (Exp. 2).

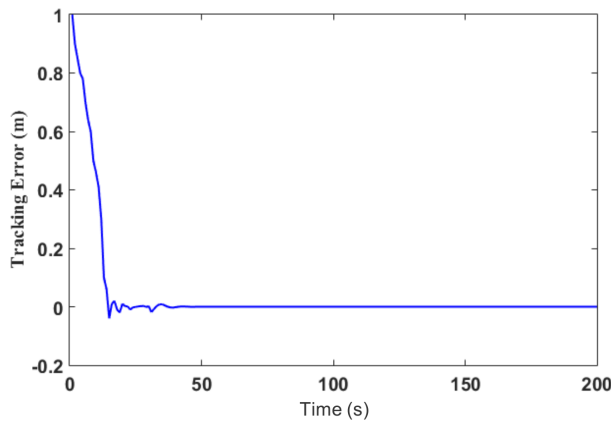


Figure 9. Tracking error signal (Exp. 2).

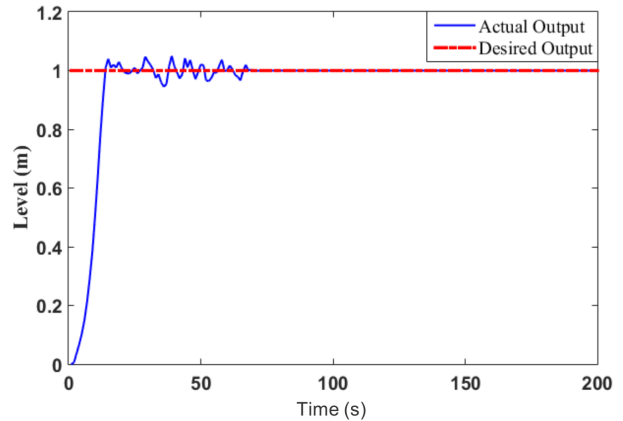


Figure 10. Performance of the proposed controller with output noise contamination (Exp. 2).

Table 2. Comparison of this work with other methods.

Method	Number of rules	Number of parameters	Settling time (s)	Overshoot (percentage)	MAE
Chaos-based FWNN-NMPC	2	18	16	3	0.005
Gradient-based FWNN-NMPC	2	18	38	12	0.012
Chaos-based ANFIS-NMPC	25	175	90	25	0.075
Chaos-based FWNN-PID	2	18	40	16	0.01

6. Conclusion

A combined controller constructed by fuzzy wavelet neural network and nonlinear model predictive control was proposed. The chaotic optimization algorithm, which is a kind of global and robust optimization algorithm, was

applied to obtain optimum inputs. The proposed controller was tested using simulation on both a nonlinear dynamic system and a nonlinear liquid level system. The results illustrated that the controller tracked the desired output in a high speed while having a small error.

References

- [1] Kayacan E, Ramon H, Saeys W. Robust tube based decentralized nonlinear model predictive control of an autonomous tractor trailer system. *IEEE-ASME T Mech* 2015; 20: 447-456.
- [2] Kaplanoglu E, Arsan T, Varsol HS. Predictive control of a constrained pressure and level system. *Turk J Elec Eng & Comp Sci* 2013; 23: 641-653.
- [3] Hedjar R. Adaptive neural network model predictive control. *Int J Innovative Comp, Inf & Cont* 2013; 93: 1245-1256.
- [4] Pan Y, Wang J. Model predictive control of unknown nonlinear dynamical systems based on recurrent neural network. *IEEE T Indust Elec* 2012; 59: 3089-3101.
- [5] Wang T, Gao H, Qiu J. A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. *IEEE T Neural Netw Learn Syst* 2016; 27: 416-425.
- [6] Chang YL, Tsai CC. Adaptive stable generalized predictive control using TSK fuzzy model for nonlinear discrete-time systems with time-delays. *Int J Fuzzy Sys* 2013; 15: 133-141.
- [7] Lu CH. Wavelet fuzzy neural networks for identification and predictive control of dynamic systems. *IEEE T Indust Elec* 2011; 58: 3046-3058.
- [8] Xiao H, Li Z, Yang C, Zhang L, Yuan P, Ding L, Wang T. Robust stabilization of a wheeled mobile robot using model predictive control based on neurodynamics optimization. *IEEE T Indust Elec* 2017; 64: 505-516.
- [9] Sridhar N, Ramrao N, Singh MK. Design of PID controller with auto tuned prefilter using chaotic optimization. In: *International Conference on Circuits, Communications, Control and Computing*; 21–22 Nov 2014; IEEE. pp. 141-145.
- [10] Xu H, Qian X, Zhang L. Study of ACO algorithm optimization based on improved tent chaotic mapping. *J Int Comp Sci* 2012; 9: 1653-1660.
- [11] Jahani R, Chahkandi Nejad H. Comparison of chaotic optimization algorithm and other evolutionary techniques for optimal unit commitment of power system. *Australian Journal of Basic and Applied Sciences* 2011; 5: 923-929.
- [12] Wang L, Wei X, Shao Y, Zhu T, Zhang J. MPPT of PV array using stepped up chaos optimization algorithm. *Turk J Elec Eng & Comp Sci* 2015; 23: 1748-1760.
- [13] Javidi MM, Emami N. A Hybrid Search Method of Wrapper Feature Selection by Chaos Particle Swarm Optimization and Local Search. *Turk J Elec Eng & Comp Sci* 2016; 24: 3852-3861.
- [14] Li P, Xu D, Zhou Z, Lee WJ, Zhao B. Stochastic optimal operation of microgrid based on chaotic binary particle swarm optimization. *IEEE T Smart Grid* 2016; 7: 66-73.
- [15] Camacho EF, Bordons C. *Model Predictive Control*. Berlin, Germany: Springer, 1999.
- [16] Davanipour M, Zekri M, Sheikholeslam F. Fuzzy wavelet neural network with an accelerated hybrid learning algorithm. *IEEE T Fuzzy Sys* 2012; 20: 463-470.
- [17] Lin FJ, Tan KH, Fang DY, Lee YD. Intelligent controlled three phase squirrel cage induction generator system using wavelet fuzzy neural network for wind power. *IET Renew Pow Gener* 2013; 7: 552-564.
- [18] Li B, Jiang WS. Optimizing complex functions by chaos search. *Cybernet Syst* 1998; 24: 409-419.
- [19] Engin SN, Kuvulmaz J, Omurlu VE. Fuzzy control of an ANFIS model representing a nonlinear liquid level system. *Neural Comput Appl* 2004; 13: 202-210.