Research of NCS Random Time Delay Prediction Based on FNN-PSO

Jian-He LEI, Jing-Xue LU, Bing-Xin XUE and Guang-Yi WANG

School of Information and Control Engineering, Qingdao University of Technology, Qingdao, 266520, China
1009856297@qq.com

Abstract. Networked control systems have been attracting considerable attention during the last decades, not only because of their theoretical value, but also of their widespread practical applications. In practice, networked control systems are inevitably affected by the random time delays. Aiming at the problem of system performance degradation caused by random time delay in networked control systems, a method of combining particle swarm optimization with fuzzy neural network is used to establish a networked control system time delay prediction model and predicts the future time delay. According to the simulation, we verify the effectiveness and feasibility of the proposed method. The results of simulation show that the fuzzy neural network optimized by particle swarm optimization has a high prediction accuracy for random time delay and is suitable for practical applications.

1. Introduction

The networked control system (NCS) refers to a closed-loop feedback control system formed by actuators, sensors and controllers in the control system to exchange information through the network. The addition of the network in the control loop makes the analysis of NCS complicated, especially the induced time delay in the network has an important impact on the performance of the system [1]. Therefore, accurate prediction of the network time delay is helpful to improve the performance of the NCS.

In recent years, a lot of research results have been reported on the network time delay prediction problem. In [2], a time-delay estimate algorithm for NCS based on least-square support vector machines is proposed. The authors in [3] use Auto-Regressive and Moving Average model to make forward time delay prediction and compare it with the prediction effect of Radial Basis Function neural network. In [4], a particle swarm optimization least square support vector machine method is used to establish an NCS time-delay prediction model and predict the future time-delay. For [5], the authors propose a novel method of NCS stochastic delay prediction based on the Autoregressive Intergated Moving Average. By using this method, high prediction accuracy can be achieved.

Aiming at the random time delay in NCS, this paper uses the FNN [6-8] optimized by PSO to predict the time delay, which makes up for the shortcomings of the conventional BP algorithm, such as slow convergence speed and easy to fall into local extreme points, and ensures the accuracy of prediction.
2. Fuzzy neural network optimized by particle swarm optimization algorithm

2.1. Fuzzy neural network model

Fuzzy neural network is a combination of fuzzy set theory and neural network. It not only has the large-scale parallel processing ability, self-learning ability and self-adaptive ability of neural network, but also can process fuzzy information and complete fuzzy reasoning function with superior performance.

The prediction model is a four-layer network model with multiple inputs and single output, as shown in Figure 1. The four layers are the input layer, the fuzzification layer, the fuzzy inference layer, and the output layer.

![Fuzzy neural network structure](image)

Figure 1. Fuzzy neural network structure.

The first layer is the input layer, the number of neurons in the input layer is the same as the number of input variables, and the input vector is \( x = (x_1, x_2, \ldots, x_n)^T \). \( y \) is the output layer variable. Since the output variable is the predicted value of future time delay and has only one, there is only one neuron in the output layer. The jth fuzzy rule of the system can be expressed as

\[
R_j: \text{if } x_1 = X_{1j}, x_2 = X_{2j}, \ldots, x_n = X_{nj}, \text{ then } y = Y_j
\]

where \( Y_j \) is the result of the jth rule, \( j \) depends on the number of neurons in the fuzzy inference layer; \( X_{ij} \) is the fuzzy subset corresponding to the input variable, and \( i = 1, 2, \ldots, n, j \) depends on the number of fuzzy subsets of each input variable.

The second layer is the fuzzification layer. In this layer, the membership degree of each input variable is calculated. The normal distribution function is used as the membership function, and the formula for calculating the ith input variable satisfying the fuzzy subset \( X_{ij} \) is as follows

\[
\mu_{X_{ij}}(x_i) = \exp \left[ -\frac{(x_i - m_{ij})^2}{\sigma_{ij}^2} \right] \tag{2}
\]

where \( m_{ij} \) is the expectation of normal distribution function, \( \sigma_{ij} \) is the standard deviation, and the membership degree of input variables is completely determined by parameters \( m_{ij} \) and \( \sigma_{ij} \).

The third layer is the fuzzy inference layer. This paper defines the connection weights of the fuzzification layer and the fuzzy inference layer according to the fuzzy theory.

\[
\mu_j = \mu_{X_{1j}}(x_1) \times \mu_{X_{2j}}(x_2) \times \cdots \times \mu_{X_{nj}}(x_n) = \exp \left[ -\sum_{i=1}^{n} \frac{(x_i - m_{ij})^2}{\sigma_{ij}^2} \right] \tag{3}
\]

where \( \mu_{X_{ij}}(x_i) \) is the membership of the input variable \( x_i \) to the fuzzy subset \( X_{ij} \).
The FNN performs the fuzzy decision function from the third to the fourth layer. Assuming that there are \( m \) rules in the system, this paper uses the weighted average method to defuzzify and obtains the output as

\[
y = \frac{\sum_{j=1}^{m} w_j \mu_j}{\sum_{j=1}^{m} \mu_j}
\]

where \( w_j \) is the connection weight between the fuzzy inference layer and the output layer.

To train the FNN, the expectation \( m_{ij} \) and standard deviation \( \sigma_{ij} \) of the normal distribution function and the connection weight \( w_j \) need to be determined. Gradient descent method is commonly used in neural networks. The algorithm has slow convergence speed and is easy to fall into local optimization problems, so it is difficult to get the global optimal solution. In order to solve this problem, the PSO algorithm with global optimization performance is used to optimize the parameters of the FNN.

2.2. Particle swarm optimization algorithm

2.2.1. Mathematical description of particle swarm optimization. If a target is found in a D dimensional space, the population is composed of \( m \) particles. Each particle in the population has its own position, and the position of the \( i \)th particle is represented by \( \mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD}), i = 1, 2, \ldots, m, \) and its velocity is also denoted by a D-dimensional vector \( \mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD}). \) When each particle passes through a new position, it compares with all the positions that have passed before. When the \( i \)th particle flies to a certain position, the optimal position it passes is \( \mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iD}) \). By comparing the optimal positions of all particles in the particle swarm, the optimal positions of all particles are obtained and expressed as \( \mathbf{p}_g = (p_{g1}, p_{g2}, \ldots, p_{gD}) \). Each particle changes its position according to the following formula.

\[
v_{id}(t + 1) = w v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t))
\]

\[
x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1)
\]

where \( i \) is the number of particles, \( i = 1, 2, \ldots, m; \) \( d \) is the spatial dimension, \( d = 1, 2, \ldots, D; \) \( w \) is the inertia weight and is non-negative; the acceleration factors \( c_1 \) and \( c_2 \) are defined as constants; \( r_1 \) and \( r_2 \) are random numbers and are evenly distributed between 0 and 1; \( x_{id}(t) \in [-x_{max}, x_{max}], v_{id} \in [-v_{max}, v_{max}], x_{max} \) and \( v_{max} \) are non-negative.

2.2.2. Algorithm flow chart. The particle swarm algorithm flow chart is shown in Figure 2. The position and velocity of the particle are randomly initialized, the particle fitness value is calculated, and the particle position is compared with the optimal position to determine whether the convergence criterion is satisfied. If satisfied, the output particle position is regarded as the optimal position of the particle swarm; if not, the position and velocity of the particle are updated, and the iteration continues.

2.2.3. Fitness function and parameter optimization steps. The PSO is used to optimize the connection weight and membership function of the FNN. That is to say, the element of the position vector \( \mathbf{x}_i \) of the PSO is the connection weight of the FNN, the expectation and standard deviation of the membership function. The training of neural networks is mainly to minimize the parameter errors in the network. Therefore, the fitness function can be defined as

\[
f = \frac{1}{N_s} \sum_{i=1}^{N_s} (Y_i - y_i)
\]

where \( N_s \) is the number of training set samples; \( Y_i \) is the expected output; \( y_i \) is the actual output.

The steps of parameter optimization based on particle swarm optimization are as follows: 1) Initialization, randomly initialize the position and velocity of each particle, and determine parameters such as population number, inertia weight and acceleration factor; 2) Calculate the fitness of each
initial particle, $p_i$ is the optimal position of the current particle and $p_g$ is the optimal position of the particle swarm. 3) Update the position and velocity of the particles according to formulas (5) and (6) to generate new particles; 4) Return to step 2) until the convergence criterion is satisfied, the iteration stops, and the global extremum is the optimal solution of the training problem.

3. Prediction model establishment and simulation analysis

3.1. Prediction model establishment
In this paper, the rolling prediction method is used to predict the network time delay. The first five delay values are taken as input sample 1 and the sixth delay value as output sample 1 to form a sample pair. The second to sixth delay values are taken as input sample 2 and the seventh delay value as output sample 2 to form the second sample pair. By analogy, 300 sample pairs are constructed as data for training neural networks. After that, the 301st to 305th delay values are taken as input samples, and input to the trained FNN as an input vector to obtain the first network delay prediction value. Then, the 302nd to 306th delay values are used as input samples to obtain the second network delay prediction value, and by analogy, 20 network delay prediction values are obtained. The prediction flow chart is shown in Figure 3, and the modeling steps are as follows.

1) Establish and train fuzzy neural network. The structure of the neural network is determined by clustering algorithm, and the number of fuzzy membership degrees is selected by Kohonen network [9]. It is determined that each input variable is divided into two fuzzy subsets, so the structure of the fuzzy neural network is 5-10-32-1. Then the training data are used to train the fuzzy neural network.

2) Optimize network parameters. Initialize the particle swarm optimization algorithm, set the population number to 30, the acceleration factor to $c_1 = c_2 = 2$, the fitness threshold to 0.001, and the maximum allowable number of iterations to 350.

3) Model prediction. The test data is used to verify the model and the final prediction results are obtained.

![Flow diagram of PSO](image-url)

**Figure 2.** Flow diagram of PSO.
3.2. Simulation results and discussion

Two kinds of fuzzy neural networks are constructed. One uses gradient descent method to train the neural network parameters (FNN), and the other uses particle swarm optimization (FNN-PSO). The performance comparison of the two types of fuzzy neural networks is shown in Figure 4.

\[
R_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}
\]  
\[
M_{\text{MAPE}} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|
\]  
\[
M_{\text{MAE}} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|
\]
where $y_i$ is the true value, $\hat{y}_i$ is the predicted value, and $N$ is the total number of samples.

**Table 1.** Error indexes of the prediction results.

| Prediction model | Error indexes |
|------------------|---------------|
|                  | RMSE/% | MAPE/% | MAE/% |
| FNN              | 34.75  | 5.94   | 34.65 |
| FNN-PSO          | 12.48  | 2.09   | 12.3  |

It can be seen from Table 1 that when predicting the network time delay value, compared with BP algorithm, such error indexes as RMSE, MAPE, and MAE of PSO algorithm are decreased by 22.27, 3.85, and 22.35 percentage points respectively. The error indexes prove the effectiveness of the PSO algorithm, and the prediction accuracy is higher than the conventional BP algorithm. According to Figure 4, it can be seen that the prediction model based on PSO and FNN has better fitting ability to actual data.

**4. Conclusions**

Aiming at the problem of random time delay affecting the performance of networked control system, a fuzzy neural network optimized by particle swarm optimization is proposed to predict random time delay. The simulation results show that the fuzzy neural network optimized by particle swarm optimization algorithm has higher accuracy for time delay prediction than the traditional BP algorithm, and reduces the computation time, which is more suitable for practical applications.

**References**

[1] Yan H C, Huang X H, Wang M. Delay-dependent stability of networked control systems with uncertainties and multiple time-varying delays[J]. Control Theory and Applications, 2008, 25(2): 303-306.

[2] Li C M, Xiao J, Zhang Y. Approach of adaptive prediction control on networked control systems based on least-squares support vector machines[J]. Journal of System Simulation, 2007, 19(15): 3494-3498.

[3] Xu P, Chen Q G, Ge Y, et al. Research of NCS forward time delay prediction based on ARMA model[J]. Computer Engineering and Applications, 2013, 49(20): 91-95.

[4] Tian Z D, Gao X W, Li K. Networked control system time-delay compensation algorithm based on predictive control[J]. Systems Engineering and Electronics, 2013, 35(10): 2165-2169.

[5] Xu W, Ge Y, Wang Y. Predicting NCS stochastic delay based on ARIMA[J]. Journal of Anhui Polytechnic University, 2016, 31(4): 72-76.

[6] JIANG S F, ZHANG C M, ZHANG S. Two-stage structural damage detection using fuzzy neural networks and data fusion techniques[J]. Expert Systems with Applications, 2011, 38(1): 511-519.

[7] Qian W, He Z X, Zhang D Y. Multi fire sensor characteristic parameters fusion algorithm based on fuzzy neural network[J]. Chinese Journal of Sensors and Actuators, 2017, 30(12): 1906-1911.

[8] Zhao J B, Xue B X. Fuzzy neural network for studying the prediction of dissolved oxygen in aquaculture ponds based on particle swarm[J]. Fishery Modernization, 2018, 45(4): 8-14.

[9] Gao F, Liu J, Li Y, et al. Identification of temperature measuring points for machine tools with Kohonen self-organizing competitive network[J]. China Mechanical Engineering, 2014, 25(7): 862-866.

[10] Huan J, Cao W J, Qin Y L, et al. Prediction of dissolved oxygen based on empirical mode decomposition and least squares support vector machine[J]. Fishery Modernization, 2017, 44(4): 37-43.