ABSTRACT

PID neural network controller was designed, and the adaptive mutation particle swarm optimization (PSO) algorithm was adopted to optimize its initial weights. The adaptive mutation PSO-PIDNN learning algorithm of motor synchronization control was put forward. According to the need of engineering, the algorithm was simplified. In proportion of multiple motor synchronization control system simulation showed that the simplified PSO-PIDNN controller also can effectively realize the motor synchronization control, solve the problem about the parameter setting is not easy for traditional PID control, and has good stability, fast convergence speed, and no overshoot.

INTRODUCTION

Multi-motor synchronization control is the core problem in printing, paper, binding, textile and other mechanical transmission system with high precision, high speed. The stand or fall of synchronous control algorithm directly affects the system's reliability and the quality of the products[1].

Because of its simple structure, good robustness, high reliability and other features, traditional PID[2] synchronous control is widely used in the synchronous control system. But the key problem of designing PID controller is how to choose
the proportion, integral, differential coefficient, and it makes the application of PID controller is restricted. In order to overcome the shortcoming of traditional PID controller, control community puts forward some methods of PID setting. For example, the literature[3]set PID fuzzy control algorithm, but the realization of the fuzzy control is too dependent on human experiences and the scope of its application was very limited. In order to achieve effective control, the neural network was introduced into synchronous control system. Literature [4] uses the combination of neural network and fuzzy algorithm, but its underlying layer is overmuch, increase the complexity of the system and the amount of calculation. Literature [5] uses single neuron PID synchronous control, this kind of single-layer network only has linear classification ability, can't make simple logic operations, so this controller in the complex system is difficult to achieve good performance.

Due to the learning convergence speed of general neural network is slow, easily trapped in local minimum point and there is no guarantee for real-time control system. At the same time, the hidden layer unit number and the connection weights is difficult to determine, thus restricting its extensive applications in control system. Literature [6] proposes a new type of neural network model—PID neural network (PIDNN) model. The PIDNN is an amalgam of PID control and neural network, thus it has the advantages of neural network and PID control and overcome the shortcomings of traditional control method and the general neural network.

Particle swarm optimization (PSO)[7] is a kind of new, global optimization algorithm. It obtained the ideal effect in terms of solution to find the optimal solution. Compared with genetic algorithm, PSO has no crossover and mutation operation, and just through internal speed to update particles. Its principle is simpler, less parameters and easier.

In conclusion, this paper designs the PSO - PIDNN controller and simulated for the parallel multiaxial proportional synchronous control system. PSO-PIDNN uses PSO algorithm to define the PIDNN hidden layer to output layer weights, and makes it has better convergence direction from the start. Moreover, according to the engineering requirements, we make some optimization.

PROPORTION OF MULTI-MOTOR SYNCHRONIZATION CONTROL STRATEGY

In no axis synchronous motor control system, the running between the drive shafts need to maintain a certain speed. Synchronized proportional relationship commonly is \( w_1 y_1 = w_2 y_2 = \ldots = w_n y_n \). When the motor synchronous ratio \( w \) is 1, it's the most simple synchronous relationship. But in actual production, running between the motor need according to the different proportion, as: \( w_1 : w_2 : \ldots : w_n = \mu_1 : \mu_2 : \ldots : \mu_n \).
Current motor synchronization control strategy have parallel, master-slave control, virtual control total shaft coupling control, cross coupling control and deviation control and so on[8-11].

In parallel synchronous control, each shaft has a good synchronization performance. The system uses this control structure without instruction time lag, can extend the number of shaft and has a simple structure. In multi-axis synchronous control system, it is widely used. In this paper, the model uses parallel type synchronous control with 2 motors as actuators. The synchronous control method is shown in figure 1.

**PERMANENT MAGNET SYNCHRONOUS MOTOR**

In the motor shaft less synchronous control system, permanent magnet synchronous motor, with simple structure, reliable operation, high power factor, large torque at low speed, is widely used. But it is a nonlinear, strong coupling, multi-variable system, and need through some changes to convert to dc motor features and modeling[12]. In engineering design, in order to simplify the analysis of the problem, we have to simplify and get simplified model for permanent magnet synchronous motor[13]:

\[
G(s) = \frac{K_m}{JL s^2 + (JR + f f J) s + (R_f + K e K_m)}
\]  

(1)

\(L \cdot R \cdot I\) are the armature winding inductance, resistance and current; \(Ke\) is counter electromotive force proportion coefficient; \(Km\) is the moment coefficient of the motor; \(J\) is the moment of inertia of the motor shaft; \(fm\) is the damping coefficient.

Using the two same motor servo system, and the parameters for: \(J=0.1982013 \text{kg} \cdot \text{m}^2; fm=10-4 \text{N} \cdot \text{m/(rad} \cdot \text{s})\); \(R=25 \Omega; L=0.3744 \ \text{H}; \quad Km=0.727 \ \text{N} \cdot \text{m/A} \); \(Ke = 0.2 \text{V} / (\text{rad} \cdot \text{s})\), we can get:
\[
G(s) = \frac{1}{0.10207231 s^3 + 6.81578 s + 0.2034}
\]  

(2)

We have the discretization process for formula (2) and use zero order retainer with sampling period \( t_s = 0.01 \) s, having

\[
y(k) = 1.513 y(k - 1) - 0.5129 y(k - 2) + 0.0003968 u(k - 1) + 0.0003179 u(k - 2)
\]

(3)

SYNCHRONOUS CONTROL ALGORITHM WITH MULTI-MOTOR

**PID Neural Network (PIDNN)**

Traditional PIDNN is a three layer forward neural network with \( 2 \times 3 \times 1 \) structure. The features of PIDNN is that the middle layer defines the proportion, integral and differential three neurons, equivalent to put PID control into the neural network and make it has a good network dynamic characteristics. Figure 2 is the PIDNN structure.

**THE PIDNN CONTROLLER ALGORITHM AND VALUE FUNCTION**

Figure 2 illustrates the forward algorithm consists of input layer and hidden layer and output layer. PIDNN hidden layer has three neurons, they are proportion, integral, and differential dollar. The status of each neuron and input/output function in the forward algorithm as shown in literature [6].

Value function is the basis of neural network training, commonly used with the mean square error function. Whole neural network uses the minimum mean square error (MSE) as the training of learning principles and objectives. The weights of each layer of PIDNN controller forward propagation and error back propagation is constant adjustment, and this is neural network training process.

PIDNN according to the gradient descent method is used to correct the weights of the network. In order to improve the convergence speed of learning process, to add a momentum factor with weight learning. The following type:

\[
\Delta w_i(k) = -\eta \frac{\partial E(k)}{\partial w_i} + \alpha \Delta w_i(k - 1)
\]

(4)
In type, $\eta$ is the learning rate, $\alpha$ is the inertia coefficient, $k$ is the sampling time, $\Delta w_s(k)$ is a weight correction, $E(k)$ is the mean square error (MSE) value.

CONNECTION WEIGHTS INITIAL VALUE SELECTION OF PIDNN

(1) The initial value selection for input layer to hidden layer connection weights

According to the characteristics of the PID control algorithm: $e(k)=r(k)-y(k)$, selecting the input layer to hidden layer connection weights of the initial value and make the tracking error $e$ as an intermediary input neurons. So input layer to hidden layer connection weights of the initial value are

$$w_{1j} = 1, w_{2j} = -1$$

(5)

(2) The initial value selection for hidden layer to output layer connection weights

Select PIDNN hidden layer to output layer connection weights of the initial value and make the PID connection weights output when the initial value is equivalent to PID controller output. Selecting

$$w_1^* = k_p; w_2^* = k_i; w_3^* = k_d$$

(6)

Particle swarm optimization (PSO) algorithm is used to select the PIDNN $k_p$, $k_i$, $k_d$ initial value, so as to learning process is easy to fall into local optimal value and learning rate is slow when solving the initial weights randomly.

Particle Swarm Optimization (PSO)

The basic idea for PSO is to simulate the process of the flock foraging to search the global optimal point with system under a certain performance index.
According to the literature[7], the particles update their speed \( V_i \) and location \( X_i \) by following type. Namely, the standard PSO algorithm:

\[
V_i(k+1) = wV_i(k) + c_1 \times rand_1 \times (pbest_i - X_i(k)) + c_2 \times rand_2 \times (gbest - X_i(k)) \\
\]

\[
X_i(k+1) = X_i(k) + V_i(k) \\
\]

In type: \( i=1,2,3...M \), is populations in the algorithm; \( pbest_i \) is the optimal position of each particle; \( gbest \) is the best location for population; \( k \) is the number of iterations; \( w \) is inertia weight; \( c_1 \) and \( c_2 \) are two positive acceleration coefficient, generally taking \( c_1 = c_2 = 2 \); \( rand_1 \) and \( rand_2 \) as a random number between \([0, 1]\).

PARAMETER ANALYSIS [15]

The selection of appropriate inertia weight \( w \) can better balance the global search and local search ability of the algorithm. Inertia weight coefficient adjustment strategy which are frequently used has linear adjustment strategy and random inertia weight strategy.

Figure 3 is respectively using two kinds of inertia weight strategy simulation diagram. We can see from the figure 3, random inertia weight strategy achieving stability needs far less iterations than linear adjustment strategy. In the analysis of dynamic nonlinear problem of synchronous control, random inertia weight strategy was adopted to choose \( w \), namely: \( w=0.5+\text{rand}()/2 \)
ADAPTIVE MUTAION PSO ALGORITHM

Due to PSO has some disadvantages, such as premature convergence, low search accuracy and late iterative efficiency. This paper improves the standard particle swarm optimization algorithm, and blends the mutation of genetic algorithm into the particle swarm, that is to initialize some variables again at a certain probability. This can make the particles out of the previous optimal value position and carry out the search in a bigger space, thus improving the ability of algorithm to find the best value.

Introducing the mutation probability $P_j$. In order to avoid the mutation affects the stability of the algorithm, the mutation probability need to be selected reasonably. $P_j$ is too small may affect the stability of the system, if the $P_j$ is too large, will make small particle in the stagnant for too long, weaken the effect of mutation operator.

According to the experiment, taking $P_j>0.9$ and occurring variation. The MATLAB code is as follows:

![MATLAB code]

For $k=1:N$
$P_j=rand()$
If $P_j>0.9$,
$X_j(k)=rand$
end
end

Figure 4 is standard particle swarm optimization algorithm and improved particle swarm algorithm simulation results under the same condition. It can be seen
that the adaptive PSO algorithm can quickly jump out of local optimum and come into a stable state, get the optimal solution.

**ADAPTIVE MUTATION PSO ALGORITHM PROCESS**

The algorithm process is as follows:

1. The initialization of particle swarm, including population size $N$, the position of each particle $X_i$ and velocity $V_i$.
2. Calculate the value of each particle function, namely fitness value of the particle. According to the fitness value, the initial position of each particle is set to individual optimal position $pbest_i$, populations of the optimal location $gbest$ is set to the global optimal position of the initial particles.
3. According to the formula (7), (8) to update the particle's velocity $V_i$ and position $X_i$; To make adaptive mutation in position of the particles for meeting the conditions.
4. Calculate the fitness of each particle value, according to the fitness of each particle, compare the current position and its history optimal location, if it is better than the history optimal location, the current position will be as an individual optimal position, otherwise continuing to use the history optimal location.
5. For each particle, comparing its individual optimal location and population optimal position, if it is better than that of the optimal location with population, replacement; Otherwise unchanged.
6. Check whether achieve algorithm termination conditions, if satisfied, terminate the iteration; Otherwise, return (3).

**The PIDNN Algorithm Based on Adaptive Mutation PSO**

**ADAPTIVE MUTATION PSO - PIDNN ALGORITHM OPTIMIZATION**

In order to reduce control operation time in engineering, we need to simplify the control algorithm. According to a special relationship with the PIDNN and the PID controller and the analysis of the literature [16], the input layer to hidden layer weights with PIDNN can be fixed that maintain the initial values the same, only for the hidden layer to output layer weights are modified.

**PIDNN ALGORITHM PROCESS BASED ON ADAPTIVE MUTATION PSO**

1. The initialization of each particle search scope, speed, initial position, each group of particle number, group number, the learning efficiency, etc. Initialize the PIDNN input layer to hidden layer weights.
2. Run adaptive PSO algorithm, return population optimal position, namely the PIDNN hidden layer to output layer weights $k_p, k_i, k_d$.
3. Give step signal, the PIDNN forward calculation.
(4) According to the value function, carrying on the back propagation arithmetic, updating the PIDNN hidden layer to output layer weights.
(5) Compute system output.
(6) Return(3) and cycle calculation. When reach cycles, terminating calculation and drawing.

SIMULATION AND ANALYSIS

Making input \( r = 1 \), scale factor \( \mu_1 = 1, \mu_2 = 0.8 \); Using MTALAB simulation for parallel type proportional synchronization control system. Population size is 20, particle dimension is 3, iteration for 50 times, the input signal sampling points is 500. Particles search range is [-30, 30], maximum speed is [3, 3]. The simulation waveform in motor synchronization control system is shown in figure 5.

The simulation results can be seen in figure 5, the simplified adaptive mutation PSO - PIDNN algorithm enables the system axis 1 and 2 output without overshoot phenomenon, and can quickly reach the stable state, has good quickness and adaptability, meets the requirements of synchronous control system in industrial control. Moreover the PID parameters \( k_p, k_i, k_d \) selection does not need to rely on the experience of artificial selection and engineering subjects, and has a certain universality.

CONCLUSIONS

For proportion of synchronous control system based on multiple motor, the synchronous controller is designed, and adaptive mutation PSO-PIDNN synchronous control algorithm was presented. The initial PIDNN weight was
determined with adaptive mutation PSO algorithm, and it can solve the shortcomings that learning process is easy to fall into local optimal value and learning rate is slow when the initial weights is random, also avoid the disadvantages for the traditional PSO search precision is low and late iterative efficiency is not high. Then analyzing the selection of the PSO inertia weight, and choosing the weight adjustment strategy which suitable for synchronous control system. Finally, in the process of PIDNN training, network input layer and output layer weights are set for fixed values according to the actual system. It greatly reduces the computational complexity and improves the practicability of synchronous controller. Optimization of adaptive mutation PSO - PIDNN controller has good adaptability and quickness, no overshoot by simulation analysis. In addition, it solves the disadvantages of the traditional PID control parameters selection difficulty, and can effectively realize the motor synchronization control performance, can meet the engineering requirements about synchronized control system.

ACKNOWLEDGEMENTS

This work has been supported by the Production Research funding of joint innovation of Jiangsu Province of China (Grant No. BY2014004-08).

REFERENCES

5. Xingang-miao, Su-Wang, Lingpang-Han, etc. Based on the deviation coupling of single neuron synchronous motor control [J], Micro-motor, 2011, 2, 44-47. (In Chinese)