Neurocomputing 149 (2015) 215–223

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Improved incremental Regularized Extreme Learning Machine Algorithm and its application in two-motor decoupling control

Jin-Lin Ding [a,*], Feng Wang [a,b], Hong Sun [a], Li Shang [a]

[a] Suzhou Vocational University, Suzhou, Jiangsu Province 215104, China
[b] School of Electrical and Information Engineering, Jiangsu University, Jiangsu, Zhenjiang 212013, China

ARTICLE INFO

Article history:
Received 15 August 2013
Received in revised form
28 January 2014
Accepted 4 February 2014
Available online 30 September 2014

Keywords:
II-RELM
Decoupling control
Cholesky factorization without square root
Generalized inverse
Vector mode
Changing rate of learning error

ABSTRACT

Regularized Extreme Learning Machine (RELM) is an ideal algorithm for regression and classification due to its fast training speed and good generalization performance. However, how to obtain the suitable number of hidden nodes is still a challenging task. In order to solve the problem, a new incremental algorithm based on Cholesky factorization without square root is proposed in this paper, which is called the improved incremental RELM (II-RELM). The method can automatically determine optimal network structure through gradually adding new hidden nodes one by one. It achieves less computational cost and better accuracy through updating output weights. Finally, neural network generalized inverse (NNGI) based on II-RELM is applied to two-motor synchronous decoupling control. Simulation indicates that the proposed algorithm has excellent performance in prediction control. It realizes the decoupling control between velocity and tension.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Inverse method is an effective way for feedback linearization of complex nonlinear system in multivariable decoupling control. However, the method is excessively dependent on accurate mathematical model and difficult to solve, which restricts its application. To overcome these defects, Dai et al. [1,2] put forward neural network inverse (NNI) control strategy, which builds a dynamic inverse model of the original system based on neural network (NN) and integrator. In [3] Dai and Liu proposed NN α-th order inverse system, and applied it in the two-motor variable frequency speed-regulating control. Neural network generalized inverse (NNGI) is presented in [4–6], which can transform the multi-input multi-output (MIMO) nonlinear system into a number of single-input single-output (SISO) linear subsystems with open-loop stability. The constructed pseudo-linear composite system (PLCS) realizes the decoupling control between speed and tension. Experimental results show that the method has good static and dynamic decoupling performance. In [7–11] some methods of building NNI are also presented. On-line adjustment control of two motor speed-regulating system based on NNGI is applied in [12]. The proposed method achieves strong stability and robustness.

Despite the many achievements that have been acquired in NNI, the slow training speed of traditional NN restricts its application in control system. Extreme learning machine (ELM), a novel learning algorithm for single hidden-layer feed forward neural networks (SLFNs), has been proposed in [13–22], and has been applied in various fields [23–26]. Within the framework of ELM, the input weights and hidden biases are randomly chosen, and the output weights are analytically determined by using Moore–Penrose generalized inverse. Compared with the traditional gradient-based learning algorithms, ELM not only learns much faster with higher generalization performance, but also avoids many difficulties that are faced by gradient-based learning methods such as slow learning rate, the existence of local minima and over tuning issues. ELM has far less computational burden than evolutionary algorithms. Its performance in terms of accuracy is also satisfactory [17].

Although the ELM model has many advantages, it may lead to over-fitting problem [14,15]. Compared with the original ELM algorithm, regularized ELM (RELM) based on the structural risk minimization principle and weighted least square has significantly improved generalization performance in most cases without increasing training burden. In [15] RELM achieves better robustness than ELM, and overcomes the over-fitting issue of ELM.

However, it is still unknown how to choose the optimal number of hidden nodes in RELM. A common approach to searching such fixed network size is by trial and error. This approach, although straightforward, is computationally expensive and does not guarantee that the selected network size will be close to optimal. In [27] researchers have proved in theory that for SLFNs with additive nodes, input weights and biases of hidden nodes may be frozen once they have been tuned. They need not be further adjusted when new nodes are added.

* Corresponding author.
E-mail address: djinlin@jssvc.edu.cn (J.-L. Ding).

http://dx.doi.org/10.1016/j.neucom.2014.02.071
0925-2312/© 2014 Elsevier B.V. All rights reserved.

In order to solve the determination problem of hidden nodes when RELM is applied to chaotic time series prediction, incremental RELM algorithm based on Cholesky factorization (CF-RELM) is proposed [28], which can determine the optimal hidden nodes automatically. Simulation results show that the method has higher prediction accuracy and less computational cost. In [29], Huang et al. further proved that an incremental ELM (I-ELM) can enable SLFNs work as universal approximators. Such a new network is fully automatic in the learning process, and no manual tuning of control parameters is needed. In [30] Huang and Chen provided a modified incremental algorithm based on a convex optimization method to further improve the convergence rate of I-ELM by allowing properly adjusting the output weights of the existing hidden nodes when a new hidden node is added. To obtain compact network architecture, enhanced I-ELM (EI-ELM) is presented [31]. At each learning step, EI-ELM picks the optimal hidden node among several randomly generated hidden nodes, which leads us to the smallest residual error. Compared with the original I-ELM, EI-ELM achieves faster convergence rate and much more compact network architecture. In [32] an error-minimization-based method (EM-ELM) is proposed, which can grow hidden nodes one by one or group by group. The output weights will be incrementally updated which significantly reduces the computational complexity.

The paper puts forward a new incremental RELM algorithm, denoted by improved incremental RELM (II-RELM). The method can automatically determine optimal network structure through gradually adding new hidden nodes one by one, and has less computational cost and better accuracy through updating output weights than conventional I-RELM. II-RELM is applied to approximate the NNGI of the two-motor synchronous system (TMSS), and to connect identified inverse model with the original system. The formed PLCS realizes the decoupling control of tension and speed. In this paper, there are two major contributions. (1) II-RELM based on Cholesky factorization without square root is proposed. The algorithm has faster learning speed and less computational cost than traditional I-RELM. (2) II-RELM is applied to TMSS.

This paper is organized as follows. Section 2 analyzes the proposed II-RELM. In Section 3 mathematical model of TMSS is described. Section 4 presents simulation of NNGI based on II-RELM. Section 5 demonstrates the experimental results of TMSS based on II-RELM. Section 6 is the conclusion.

2. Proposed II-RELM algorithm

2.1. Brief of RELM

In this section, we briefly describe the essence of RELM [15,16].

Given N distinct training samples $\{(X_i, t_i)|X_i \in R^n, t_i \in R^m\}_{i=1}^N$, standard SLFNs with L hidden nodes and activation function $g(X)$ are mathematically modeled as

$$H\beta = T \tag{1}$$

where

$$H(a_1,\ldots,a_L,b_1,\ldots,b_L,X_1,\ldots,X_N) = \begin{bmatrix} g(a_1\cdot X_1+b_1) & \cdots & g(a_L\cdot X_1+b_L) \\ \vdots & \ddots & \vdots \\ g(a_1\cdot X_N+b_1) & \cdots & g(a_L\cdot X_N+b_L) \end{bmatrix}_{N\times L} \tag{2}$$

$$\beta = \begin{bmatrix}\beta_1^T \cdots \beta_L^T\end{bmatrix}_{L\times m}^T, \quad T = \begin{bmatrix}t_1^T \cdots t_L^T\end{bmatrix}_{N\times m}^T \tag{3}$$

$a_j = [a_{j1},\ldots,a_{jn}]^T$ is the weight vector connecting the jth hidden node and the input nodes, $\beta_j = [\beta_{j1},\ldots,\beta_{jm}]^T$ is the weight vector connecting the jth hidden node and the output nodes, $g(x)$ is the activation function, and b_j is the threshold of the jth hidden node. a_j and b_j can be arbitrarily assigned.

In order to improve the generalization ability of the traditional SFLNs based on ELM, Huang et al. proposed the equality constrained optimization-based ELM in [16]. In their approach structural risk regarded as the regularization term is introduced. The so-called RELM can adjust the proportion of structural risk and empirical risk through the parameter γ. The regression problem for the proposed constrained optimization can be formulated as

$$\begin{aligned} &\text{Minimize} \quad E(W) = \tfrac{1}{2}\gamma\varepsilon^T\varepsilon + \tfrac{1}{2}\beta^T\beta \\ &\text{Subject to} \quad H\beta - T = \varepsilon \end{aligned} \tag{4}$$

where γ is the proportion parameter of two kinds of risks, and ε is the regression error. To solve the above optimization problem Lagrange function is constructed as follows:

$$\xi(\beta,\varepsilon,\alpha) = \tfrac{1}{2}\gamma\varepsilon^T\varepsilon + \tfrac{1}{2}\beta^T\beta - \alpha(H\beta - T - \varepsilon) \tag{5}$$

Letting the partial derivatives of $\xi(\beta,\varepsilon,\alpha)$ with respect to β, ε and α be zero, we have

$$\begin{cases} \beta^T = \alpha H \\ \gamma\varepsilon^T + \alpha = 0 \\ H\beta - T - \varepsilon = 0 \end{cases} \tag{6}$$

From Eq. (6) we obtain

$$\beta = \left(\frac{1}{\gamma}I + H^TH\right)^{-1}H^TT \tag{7}$$

where I is the identity matrix, and the corresponding regression model of RELM is

$$Y = h(X)\beta = h(X)\left(\frac{1}{\gamma}I + H^TH\right)^{-1}H^TT \tag{8}$$

2.2. Improved calculation method of output weights

On the basis of Huang's RELM, the improved RELM algorithm is proposed by us. In Huang's RELM, the weight vector β is solved by inverse operation of high-order matrix. There exists large amount of calculation, and it reduces modeling efficiency. In order to simplify computation, a new method based on Cholesky factorization without square root [33] is introduced.

From Eq. (6) we obtain

$$\left(\frac{1}{\gamma}I + H^TH\right)\beta = H^TT \tag{9}$$

Let

$$\begin{aligned} A &= \frac{1}{\gamma}I + H^TH \\ B &= H^TT \end{aligned} \tag{10}$$

Then Eq. (9) can be rewritten as

$$A\beta = B \tag{11}$$

Since A is a symmetric positive definite matrix [28], it can be uniquely factored as

$$A = LDL^T \tag{12}$$

where

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{13}$$

$$L = \begin{bmatrix} l_{11} & & \\ \vdots & \ddots & \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 1/l_{11} & & \\ & \ddots & \\ & & 1/l_{nn} \end{bmatrix} \tag{14}$$

L^T is the transpose of L, and

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} / l_{kk}, \quad i \geq j \tag{15}$$

Substituting Eq. (12) into Eq. (11) leads to

$$LDL^T \beta = B \tag{16}$$

Denote

$$F = L^T \beta \tag{17}$$

Substituting Eq. (17) into Eq. (16) gives

$$LDF = B \tag{18}$$

Using Eqs. (14) and (18), we have

$$f_{ij} = \begin{cases} b_{ij}, & i = 1 \\ b_{ij} - \sum_{n=1}^{i-1} l_{in} f_{nj} / l_{nn}, & i > 1 \end{cases} \tag{19}$$

According to Eqs. (15), (17), (19), we can further obtain

$$\beta_{ij} = \begin{cases} f_{ij} / l_{ii}, & i = m \\ \left(f_{ij} - \sum_{n=1}^{m-i} l_{i+n,i} \beta_{i+n,j} \right) / l_{ii}, & i < m \end{cases} \tag{20}$$

In summary, given a training set $(X_i, t_i) \in R^n \times R^m (i = 1, \ldots, N)$, activation function $g(X)$ and number of hidden nodes L, then the improved RELM can be summarized as the following 6 steps:

Step 1: Define the number of hidden nodes L, randomly assign input weights a_i and hidden layer biases b_i $(i = 1, \ldots, L)$.
Step 2: Calculate the hidden layer output matrix H.
Step 3: According to Eq. (10), calculate the matrices A and B.
Step 4: Factorize the matrix A.
Step 5: According to Eq. (19), calculate the matrix F.
Step 6: According to Eq. (20), calculate the output weight β.

2.3. Our II-RELM

Research shows that the number of hidden layer neurons not only affects learning precision and generalization ability of RELM, but also is the key factor which must be determined in designing network architecture [17,28]. I-ELM algorithm achieves expected learning accuracy through adding the number of hidden layer neurons gradually. We proposed a new II-RELM in order to achieve more compact network architecture than the original I-RELM. The method can significantly reduce the computational complexity through updating output weights.

Assuming that the initial number of hidden nodes is L, hidden layer output matrix is H_L, output weight matrix is β_L, when the number of hidden layer neurons is increased to $L+1$, we have

$$H_{L+1} = [H_L : h_{L+1}] = [h_1 \ \ldots \ h_L \ : h_{L+1}] \tag{21}$$

$$A_{L+1} \beta_{L+1} = B_{L+1} \tag{22}$$

$$F_{L+1} = L_{L+1}^T \beta_{L+1} \tag{23}$$

where $h_i = [g(a_i \cdot X_1 + b_i) \ \ldots \ g(a_i \cdot X_N + b_i)]^T, i = 1, \ldots, L+1$.

Substituting Eq. (21) into Eq. (10) leads to

$$B_{L+1} = H_{L+1}^T T = [H_L : h_{L+1}]^T, \quad T = \begin{bmatrix} H_L^T T \\ h_{L+1}^T T \end{bmatrix} = \begin{bmatrix} B_L \\ h_{L+1}^T T \end{bmatrix} \tag{24}$$

$$A_{L+1} = \frac{1}{\gamma} I + H_{L+1}^T H_{L+1} = \frac{1}{\gamma} \begin{bmatrix} I_L & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} H_L^T \\ h_{L+1}^T \end{bmatrix} [H_L : h_{L+1}]$$

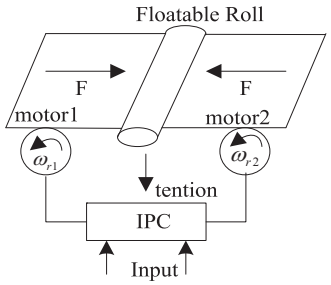

Fig. 1. The control diagram of TMSS.

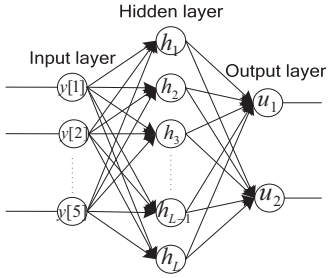

Fig. 2. The structure of SLFNs.

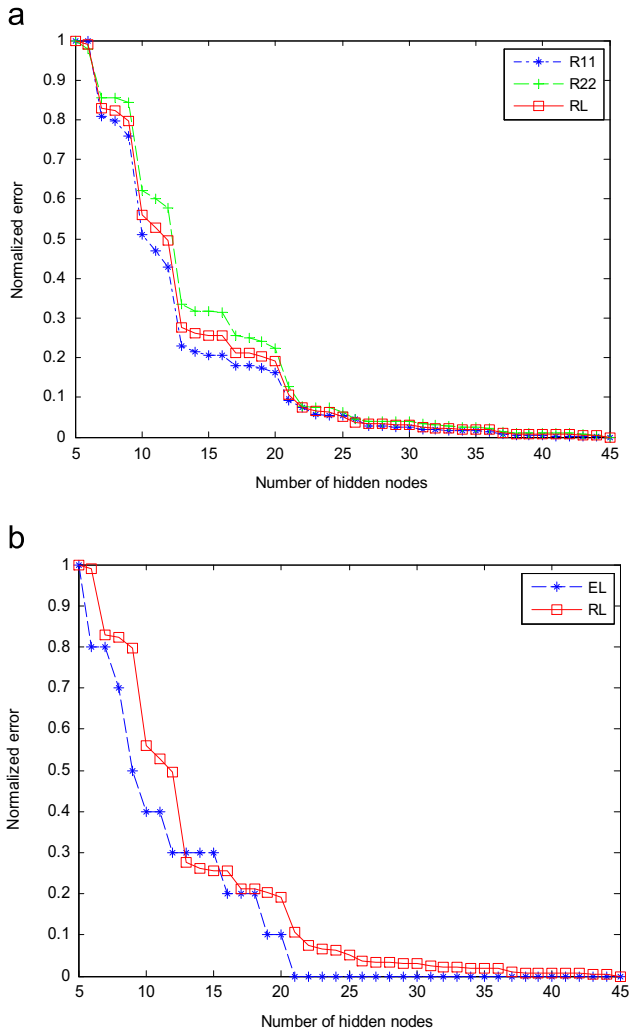

Fig. 3. Normalized errors of different hidden nodes. (a) Normalized errors of multiple outputs. (b) Normalized errors of total risk and RMSE.

$$= \begin{bmatrix} \frac{1}{\gamma} I_L + H_L^T H_L & H_L^T h_{L+1} \\ h_{L+1}^T H_L & h_{L+1}^T h_{L+1} + \frac{1}{\gamma} \end{bmatrix} = \begin{bmatrix} A_L & P_{L+1}^T \\ P_{L+1} & Q_{L+1} \end{bmatrix} \tag{25}$$

where $P_{L+1} = h_{L+1}^T H_L = [h_{L+1}^T h_1 \ \dots \ h_{L+1}^T h_L]$

According to Eqs. (12), (14), (25), we have

$$L_{L+1} = \begin{bmatrix} L_L & 0 \\ l_{L+1} & l_{L+1,L+1} \end{bmatrix} \tag{26}$$

where $l_{L+1} = [l_{L+1,1} \ \dots \ l_{L+1,L}]$, and

$$l_{L+1,j} = a_{L+1,j} - \sum_{k=1}^{j-1} l_{L+1,k} l_{jk} / l_{kk}, \quad j = 1, \cdots, L+1 \tag{27}$$

Using Eqs. (18), (24), (26) we get

$$F_{L+1} = \begin{bmatrix} F_L \\ f_{L+1} \end{bmatrix} \tag{28}$$

$$L_{L+1} D_{L+1} F_{L+1} = \begin{bmatrix} L_L & 0 \\ l_{L+1} & l_{L+1,L+1} \end{bmatrix} \begin{bmatrix} D_L & 0 \\ 0 & d_{L+1,L+1} \end{bmatrix} \begin{bmatrix} F_L \\ f_{L+1} \end{bmatrix} = \begin{bmatrix} L_L D_L F_L \\ l_{L+1} D_L F_L + f_{L+1} \end{bmatrix} = \begin{bmatrix} B_L \\ h_{L+1}^T T \end{bmatrix} \tag{29}$$

We can further obtain

$$f_{L+1,j} = b_{L+1,j} - \sum_{k=1}^{L} \frac{l_{L+1,k} f_{kj}}{l_{kk}}, \quad j = 1, \dots, L \tag{30}$$

Finally, β_{L+1} can be calculated according to Eq. (20).

Thus, we can obtain a fast incremental output updating method as represented by Eqs. (27) and (30). In the procedure we can make full use of storage information about L_L and F_L to fast calculate β_{L+1} through simple arithmetic when hidden nodes are added one by one. The method ensures learning accuracy and improves the training speed.

In summary, given a set of training data $\{(X_i, t_i) | X_i \in R^n, t_i \in R^m\}_{i=1}^N$, the maximum number of hidden nodes $L_{\max}$, and the expected changing rate of learning error ξ, we denote

$$R_{riskm} = \frac{1}{2} \gamma \varepsilon^T \varepsilon + \frac{1}{2} \beta^T \beta = \begin{bmatrix} R_{11} & \dots & R_{1m} \\ \vdots & \ddots & \vdots \\ R_{m1} & \dots & R_{mm} \end{bmatrix} \tag{31}$$

where R_{ii} is the total risk of the ith output, $i = 1, \dots, m$.

Let

$$R_L = \sum_{i=1}^{m} R_{ii} \tag{32}$$

$$\eta_j = |(R_{L-j} - R_{L-j-1}) / R_{\max}| \tag{33}$$

where $R_{\max} = \max(R_n, \dots, R_L)$, $j = 0, \dots, 4$. n is the number of system input.

R_L represents the risk sum of all m outputs and η_j represents the changing rate of R_L. Once η_j becomes smaller, R_L decreases slowly as the number of hidden nodes increases. At this time training RMSE E_L hardly reduces. This shows that RELM has the suitable number of hidden nodes.

The proposed II-RELM algorithm is described as the following:

While $L < L_{\max}$ and changing rate of learning error $\eta < \xi$.

Step 1: Let the number of hidden nodes $L = 1$, and calculate H_L, A_L and B_L .

Step 2: Calculate matrix L_L according to Eq. (14), and calculate F_L by Eq. (19).

Step 3: Calculate β_L by Eq. (20), and calculate prediction output Y_L by Eq. (8).

Step 4: Calculate training error $\varepsilon_L = Y_L - T$, and calculate total risk R_L using Eqs. (31) and (32).

Step 5: Increase the number of hidden nodes L by 1: $L = L+1$, we use updating method to calculate L_L and F_L according to Eqs. (26) and (28), then jump to step 3.

Step 6: Calculate changing rate of learning error by Eq. (33) from $L = 6$. Once all the five continuous changing rates of learning error become smaller than the expected value ξ, the suitable number of hidden nodes can be set as the one corresponding to the first changing rate. The training process terminates. Otherwise, jump to step 5.

3. Mathematical model of TMSS

The control diagram of TMSS is shown in Fig. 1.

According to Hooke's law, considering the amount of forward slip, tension is described by the following equation:

$$\dot{F} = \frac{AE}{L_0}\left(\frac{1}{n_{p1}} r_1 k_1 \omega_{r1} - \frac{1}{n_{p2}} r_2 k_2 \omega_{r2}\right) - \frac{AV}{L_0} F = \frac{K}{T}\left(\frac{1}{n_{p1}} r_1 k_1 \omega_{r1} - \frac{1}{n_{p2}} r_2 k_2 \omega_{r2}\right) - \frac{F}{T} \tag{34}$$

where K is the transfer function, T is the change tension constant, F is the belt tension, r_1, k_1, ω_{r1} are the radius, the velocity ratio and the electrical angular velocity of the first pulley respectively, r_2, k_2, ω_{r2} are the radius, the velocity ratio and the electrical angular velocity of the second pulley respectively, A is the sectional area of the belt, E is Young's elasticity modulus of the belt, L_0 is the distance between the frames, V is the expected speed.

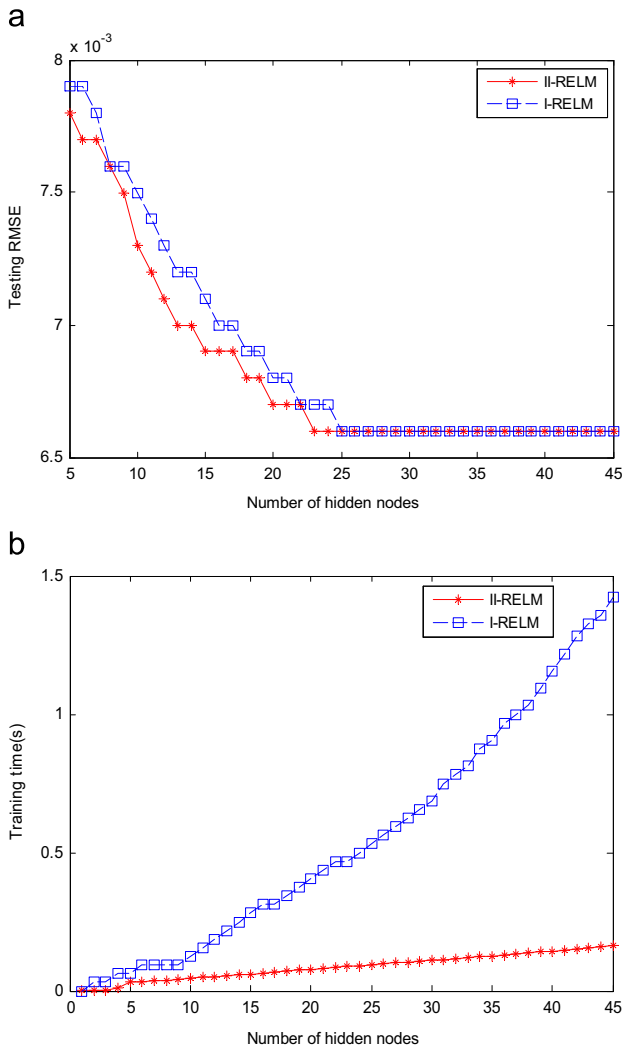

Fig. 4. Comparison of II-RELM and I-RELM. (a) Training RMSE of different algorithms. (b) Training time of different algorithms.

Under vector mode, the rotor flux approximately remains constant, the mathematical model of the system can be simplified as follows:

$$\dot{x} = f(x,u) = \begin{bmatrix} \frac{n_{p1}}{J_1}\left[(u_1 - x_1)\frac{n_{p1}T_{r1}}{L_{r1}}\psi_{r1}^2 - (T_{L1} + r_1 x_3)\right] \\ \frac{n_{p2}}{J_2}\left[(u_2 - x_2)\frac{n_{p2}T_{r2}}{L_{r2}}\psi_{r2}^2 - (T_{L2} - r_2 x_3)\right] \\ \frac{K}{T}\left(\frac{1}{n_{p1}} r_1 k_1 x_1 - \frac{1}{n_{p2}} r_2 k_2 x_2\right) - \frac{x_3}{T} \end{bmatrix} \tag{35}$$

where

State variable: $\mathbf{x} = [x_1, x_2, x_3]^T = = [\omega_{r1}, \omega_{r2}, F]^T$

Input variable: $u = [u_1, u_2]^T = [\omega_1, \omega_2]^T$

Output variable: $y = h(x) = [y_1, y_2]^T = [x_1, x_3]^T = [\omega_{r1}, F]^T$

J is the moment of inertia of the rotor, ω is the electric synchronous angular speed, ψ_r is the rotor flux, n_p is the pole of the motor, T_r is the electromagnetic time constant, and T_L is the load torque. Numbers 1 and 2 denote the first and second motors separately.

Taking the derivatives of output variables with respect to t, we have

$$\begin{cases} y_1^{(1)} = \dfrac{n_{p1}}{J_1}\left[(u_1 - x_1)\dfrac{n_{p1}T_{r1}}{L_{r1}}\psi_{r1}^2 - (T_{L1} + r_1 x_3)\right] \\ y_2^{(1)} = \dfrac{K}{T}\left(\dfrac{1}{n_{p1}} r_1 k_1 x_1 - \dfrac{1}{n_{p2}} r_2 k_2 x_2\right) - \dfrac{x_3}{T} \\ y_2^{(2)} = \dfrac{K}{T}\left\{\dfrac{1}{n_{p1}} r_1 k_1 \left[\dfrac{n_{p1}}{J_1}\left[(u_1 - x_1)\dfrac{n_{p1}T_{r1}}{L_{r1}}\psi_{r1}^2 - (T_{L1} + r_1 x_3)\right] - \dfrac{1}{n_{p2}} r_2 k_2 \left[\dfrac{n_{p2}}{J_2}\left[(u_2 - x_2)\dfrac{n_{p2}T_{r2}}{L_{r2}}\psi_{r2}^2 - (T_{L2} - r_2 x_3)\right]\right) - \dfrac{1}{T}\left[\dfrac{K}{T}\left(\dfrac{1}{n_{p1}} r_1 k_1 x_1 - \dfrac{1}{n_{p2}} r_2 k_2 x_2\right) - \dfrac{x_3}{T}\right]\right\} \end{cases} \tag{36}$$

Obviously $y_1^{(1)}$ explicitly includes u, $y_2^{(1)}$ does not explicitly includes u, therefore $y_2^{(2)}$ is computed, which includes u explicitly. So the Jacobi matrix is

$$A(x,u) = \begin{bmatrix} \frac{\partial y_1^{(1)}}{\partial u_1} & \frac{\partial y_1^{(1)}}{\partial u_2} \\ \frac{\partial y_2^{(2)}}{\partial u_1} & \frac{\partial y_2^{(2)}}{\partial u_2} \end{bmatrix} = \begin{bmatrix} \frac{n_{p1}^2 T_{r1}}{J_1 L_{r1}}\psi_{r1}^2 & 0 \\ \frac{Kr_1 k_1 n_{p1} T_{r1}}{TJ_1 L_{r1}}\psi_{r1}^2 & -\frac{Kr_2 k_2 n_{p2} T_{r2}}{TJ_2 L_{r2}}\psi_{r2}^2 \end{bmatrix} \tag{37}$$

$$Det(A(x,u)) = -\frac{n_{p1}^2 T_{r1}}{J_1 L_{r1}} \frac{Kr_2 k_2 n_{p2} T_{r2}}{TJ_2 L_{r2}} \psi_{r1}^2 \psi_{r2}^2 \tag{38}$$

When $\psi_{r1}^2 \neq 0$ and $\psi_{r2}^2 \neq 0$, the system is reversible. The relative order of system is $\alpha = (1,2)$, $\alpha_1 + \alpha_2 = 3 = n$(order of system), so the generalized inverse model is

$$u = \phi(y_1, \xi_1, y_2, \dot{y}_2, \xi_2) \tag{39}$$

where $\xi_1 = a_{21} y_1 + a_{22} \dot{y}_1$, $\xi_2 = b_{31} y_2 + b_{32} \dot{y}_2 + b_{33} \ddot{y}_2$.

In order to obtain ideal static and dynamic characteristics of PLCS, the expected poles of decoupled subsystems must be configured, $a_{21} = 1, a_{22} = 1, a_{31} = 1, a_{32} = 1.414, a_{31} = 1$. The generalized inverse system based on II-RELM is constructed to approximate nonlinear mapping $u = \phi(y_1, \xi_1, y_2, \dot{y}_2, \xi_2)$.

4. Simulation of NNGI based on II-RELM

4.1. Acquisition of training samples and II-RELM structure

Training samples are extremely important in the reconstruction of NNGI based on II-RELM. Reference signal should include all the work region of original system, which can ensure the approximate ability.

Firstly, a random model is constructed in Simulink to generate the given frequency u. Secondly, we take the given data as the input of TMSS. Speed output y_1 and tension output y_2 are collected every 0.1 s. The beginning and the end of sampled data are left out, $\dot{y}_1 + y_1, \dot{y}_2, \ddot{y}_2 + 1.414\dot{y}_2 + y_2$ are calculated in Matlab. Lastly, those data are normalized into $-1 \sim 1$.

We take $y = \{y_1, \dot{y}_1 + y_1, y_2, \dot{y}_2, \ddot{y}_2 + 1.414\dot{y}_2 + y_2\}$ as the input of SLFNs based on II-RELM, and take $u = \{u_1, u_2\}$ as the output. Among 8550 samples, the first 5600 data constitute the training sample and the remaining 2950 data constitute the test samples. The SFLNs includes 5 nodes in the input layer, L nodes in the hidden layer and 2 nodes in the output layer. The input weights a_i and the hidden layer biases b_i are arbitrarily chosen. The activation function of hidden-layer is selected as sigmoid function $g(x) = 1/(1 + e^{-x})$. The structure of SLFNs is shown in Fig. 2.

4.2. Simulation and analysis of NNGI based on II-RELM

In this section, we investigate the performance of the proposed algorithm II-RELM. Initial number of hidden nodes is set as 1 and is increased one by one. The expected changing rate of learning error is $\xi = 0.01$. The parameter γ is 256. Let $R_{\max}$ be R_5. Fig. 3(a) shows normalized risks of two outputs, where R_{11} and R_{22} are risks of u_1 and u_2 respectively, R_L is the sum of R_{11} and R_{22}. It can be seen that the normalized R_{11}, R_{22} and R_L decrease simultaneously and have similar trend with the increase of hidden nodes. In addition, normalized risks are less than 0.005 when the number of hidden nodes is equal to 22. This shows that the method can realize the regression of MIMO system.

Fig. 3(b) shows the curves of normalized risk and training root mean square error (RMSE) as the number of hidden nodes increases.

Table 1
The learning performance of different hidden nodes.

Hidden nodes	# Training time (s)	#Total risk (R_L)	# Training RMSE (E_L)	# Changing rate of learning error (η)
1	0.0005	45931	0.1798	–
2	0.0018	901.6739	0.0252	–
3	0.005	856.8452	0.0245	–
4	0.0088	637.3046	0.0211	–
5	0.0312	84.3095	0.0076	–
6	0.0345	84.1041	0.0074	0.002436262
7	0.0352	80.5565	0.0074	0.042078295
8	0.0384	80.4132	0.0073	1.70E−03
9	0.0422	79.8201	0.0071	0.007034794
10	0.0452	74.5641	0.007	0.062341729
11	0.0484	73.8934	0.007	0.007955213
12	0.0513	73.1329	0.0069	0.009020336
13	0.0545	68.3114	0.0069	0.057188099
14	0.0577	68.0029	0.0069	0.003659137
15	0.0609	67.8581	0.0069	0.001717481
16	0.0648	67.8365	0.0068	0.000256199
17	0.0679	66.9200	0.0068	0.010870661
18	0.0712	66.8810	0.0068	0.000462581
19	0.0748	66.7050	0.0067	0.002087546
20	0.0785	66.4050	0.0067	0.003558318
21	0.0824	64.5658	0.0066	0.021814861
22	0.0859	63.8766	0.0066	0.008174642
23	0.0894	63.6512	0.0066	0.002673483
24	0.0921	63.6145	0.0066	4.35E−04
25	0.0953	63.3852	0.0066	0.002719741
26	0.098	63.0076	0.0066	0.004478736
27	0.1011	62.9456	0.0066	0.000735386
28	0.1045	62.9249	0.0066	2.46E−04

E_L is the training RMSE. It can be seen that normalized R_L and E_L decrease gradually as the number of hidden nodes increases. There is a sharp decrease in R_L and E_L when the number of hidden nodes is smaller, this shows that training accuracy and generalization can be enhanced effectively by adding hidden nodes at this time. When the number of hidden nodes is increased to a certain level, E_L hardly reduces and R_L decreases slowly even if hidden nodes are added gradually. This shows that RELM has the suitable number of hidden nodes.

Fig. 4 shows training RMSE and training time obtained by II-RELM and I-RELM. Fig. 4(a) is the training RMSE of different algorithms. It can be seen that II-RELM can obtain better generalization performance than I-RELM. Fig. 4(b) shows the computational complexity comparisons of II-RELM and I-RELM. It can be seen that with the help of growing hidden nodes and incremental updating of the output weights, II-RELM is much faster than the conventional I-RELM.

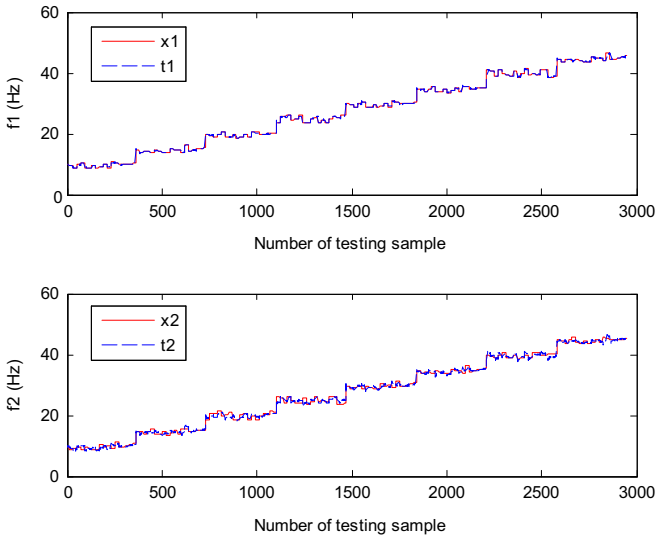

Fig. 5. Fitting curve of II-RELM.

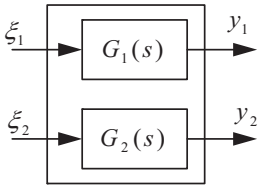

Fig. 6. Transfer block diagram of PLCS.

The learning performance of different hidden nodes is shown in Table 1. The changing rate of learning error η meets the judgment criteria presented in Section 2 that five continuous values must be smaller than the expected value 0.01 when the number of hidden nodes increases from 22. So the suitable number of hidden nodes is 22. Training time is the one corresponding to the last changing rate R_{26}. In addition, from Table 1 we can see that training RMSE E_L does not decrease once the number of hidden nodes is above 22. At this time the training RMSE E_L is 0.0066, training time is 0.098 s, which can meet the requirements of online learning for control system.

Finally, we carry out regression analysis of the trained RELM, the fitting curve is shown in Fig. 5, where t_1 and t_2 are the real frequency of two motors, x_1 and x_2 are the output frequency of NN. We can see that x_1 and x_2 converge to the expected output data set t_1 and t_2 respectively. It demonstrates that the model based on II-RELM has good generalization ability. Let $\xi_1 = \dot{y}_1 + y_1$, $\xi_2 = \ddot{y}_2 + 1.414\dot{y}_2 + y$ denote the input of NNGI and u_1, u_2 denote the output of NNGI, the generalized inverse system is designed.

5. Experimental results of TMSS based on II-RELM

5.1. Framework of TMSS based on II-RELM

The designed NNGI based on II-RELM is connected with original system, then a decoupled PLCS can be formed. Speed and tension are decoupled into independent linear subsystem, and the corresponding transfer function $G_1(s) = 1/(s+1)$, $G_2(s) = 1/(s^2+1.414s+1)$, PLCS block diagram is shown in Fig. 6. In order to enhance the robustness of control system, PID controller is designed, the closed

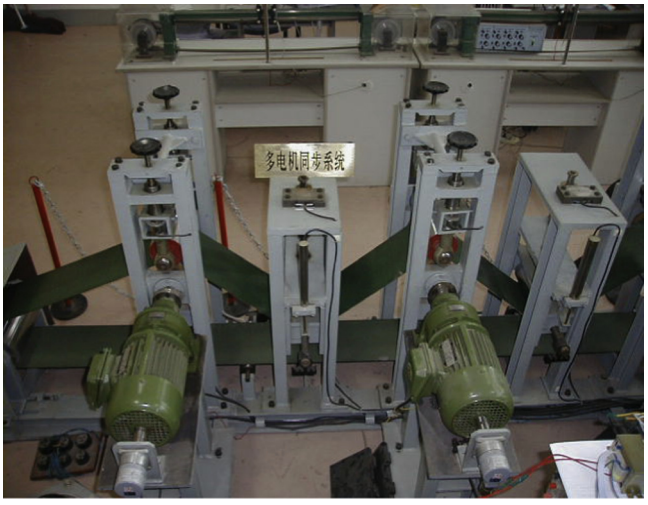

Fig. 8. Experiment platform photo of two-motor synchronous system.

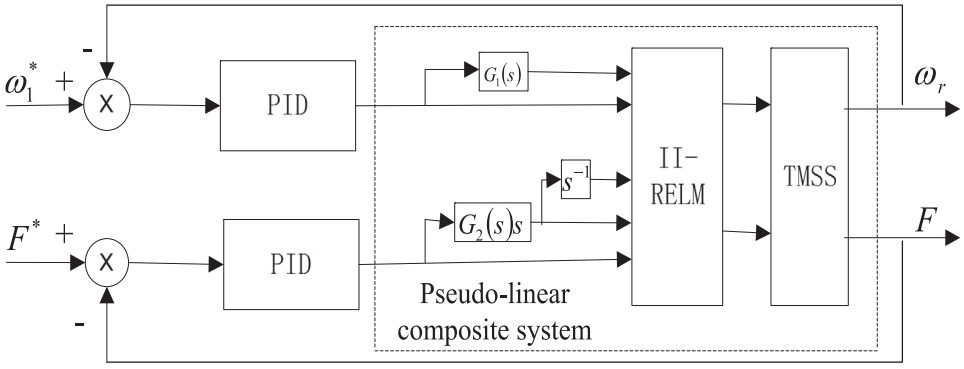

Fig. 7. The closed loop control block diagram of the two-motor NNGI.

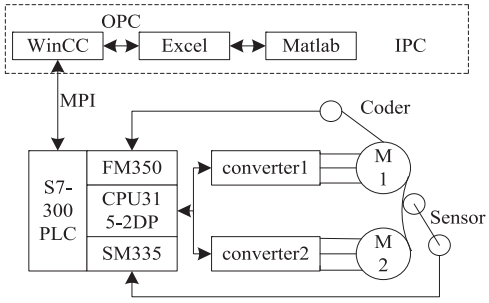

Fig. 9. Block diagram of two-motor synchronous system.

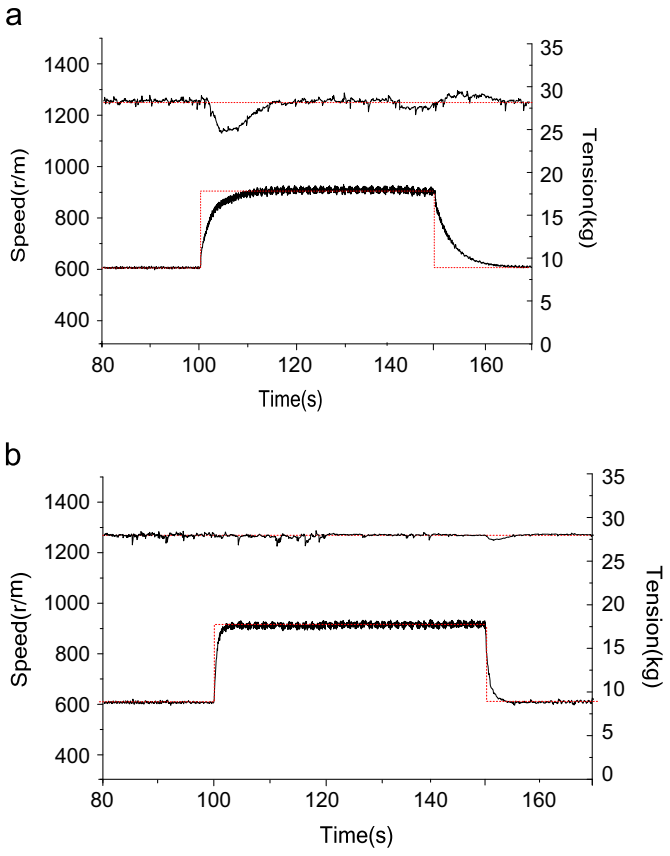

Fig. 10. The graph of response when the speed is suddenly decreased and increased. (a) PID Control, (b) NNGI Control based on II-RELM.

loop control system based on II-RELM is built. The control diagram is shown in Fig. 7.

5.2. Experiment and analysis of TMSS based on II-RELM

In order to investigate the performance of the proposed IIRELM, two-motor experimental platform is built, and it is shown in Fig. 8. The system includes Industry Personal Computer (IPC), monitoring software WinCC, S7-300 PLC, high-speed analog input/output module SM335, high-speed counter module FM350, a tension sensor, a photoelectric encoder, two Siemens MMV converters and two 2.2Kw 3-phase induction motors.

Fig. 9 is the structure diagram of TMSS. Two converters work in the vector mode, and PLC controls them by Profibus-dp. Wincc monitors PLC through MPI. OPC technology is used to exchange real-time data with Excel. Matlab directly calls data in Excel to train SLFNs based on II-RELM on-line. Then the optimal number of hidden nodes is obtained. Finally, the number of hidden nodes,

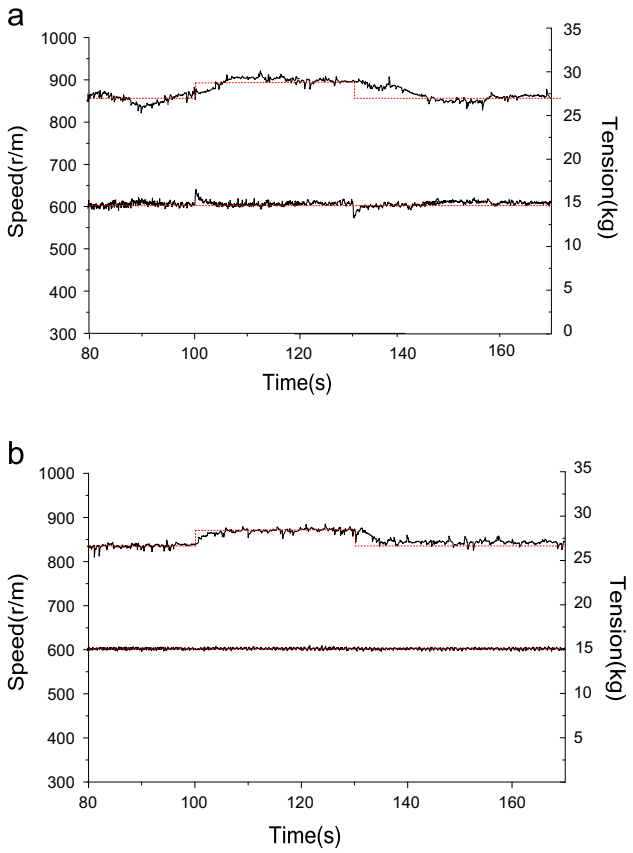

Fig. 11. The graph of response when the tension is suddenly decreased and increased. (a) PID Control, (b) NNGI Control based on II-RELM.

input weights, hidden biases and output weights are transferred to PLC through OPC, and stored in DB module, SLFNs based on the improved RELM can be constructed by the values in DB module.

Off-line 1000 groups of data are combined with 200 groups of updated data to train the NN in Matlab, and the initial number of hidden nodes is set to be 5. The PI regulator is added in speed pseudo-linear subsystem and the parameters are set as $K_p = 0.62, K_i = 30$. The PD regulator is added in tension pseudo-linear subsystem and the parameters are set as $K_p = 20, K_d = 120$. In addition, we find that training time is no more than 0.35 s and can satisfy on-line requirements. According to Fig. 7, expected speed and tension are set. The actual speed ω_r and tension F of experiment platform are sampled, and we can analyze the decoupling performance of the control system compared with traditional PID control.

First, when the given tension remains constant, the given speed is increased suddenly from 600 rpm to 900 rpm at 100 s and reduced to 600 rpm at 150 s, the response curves of speed and tension are shown in Fig. 10. Fig. 10(a) shows the graph based on PID. Fig. 10(b) shows the graph based on II-RELM. The upper curve is tension wave and the lower curve is speed wave. It is clear that larger fluctuation exists in tension response under traditional PID control when speed is increased or decreased suddenly than NNGI based on II-RELM.

Second, when the given speed remains constant, the given tension is increased suddenly at 100 s and reduced at 130 s, and the response curves of speed and tension are shown in Fig. 11. Fig. 11(a) shows the graph based on PID. Fig. 11(b) shows the graph based on II-RELM. The upper curve is the tension wave and the lower curve is the speed wave. The speed response curve based on II-RELM remains almost unchanged when tension increases or deceases suddenly. The result is better than PID control.

The experiment results indicate that when speed or tension increased or decreased suddenly, the influence on each other is very small under NNGI based on II-RELM and achieves better results than PID. The method realizes the decoupling between speed and tension. In addition, compared with PID control, NNGI system based on II-RELM has better tracking ability.

6. Conclusions

II-RELM learning algorithm based on Cholesky factorization without square root is proposed here, which can automatically determine the reasonable number of hidden nodes. The proposed method avoids inverse operation, reduces the computation complexity and has excellent performance in prediction. Finally, we use II-RELM to approximate the inverse of TMSS, which realizes the decoupling control between velocity and tension. This method can also be applied in other complex nonlinear control system, and has good application prospect.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 51077066, 61373098), the University Graduate Research Innovation Foundation of Jiangsu Province, China (Grant no. CXZZ13_0685), the Natural Science Foundation of Jiangsu Province, China (Grant no. BK2011319), the Innovation Foundation of Suzhou Vocational University (Grant no. 2011SZDCC01).

References

[1] X.Z. Dai, D. He, X. Zhang, T. Zhang, Mimo system invertibility and decoupling control strategies based on ann αth-order inversion, IEE Proc. Control Theory Appl. 148 (2) (2001) 125–136.
[2] Y. han Ding, X. zhong Dai, T. Zhang, Low-cost fiber-optic temperature measurement system for high-voltage electrical power equipment, IEEE Trans. Instrum. Meas. 59 (4) (2010) 923–933.
[3] X. zhong Dai, G. hai Liu, Neural network inverse synchronous control of two-motor variable frequency speed-regulating system, Acta Autom. Sin. 31 (6) (2005) 890–899.
[4] G. hai Liu, P. yuan Liu, Y. Shen, F. liang Wang, M. Kang, Neural network generalized inverse decoupling control of two-motor variable frequency speed-regulating system, Proc. CSEE 28 (36) (2008) 98–102.
[5] G. hai Liu, G. xue Yang, Y. Shen, Z. ling Chen, Internal model control of two motor variable frequency systems based on neural network generalized inverse, Trans. China Electrotech. Soc. 25 (11) (2010) 55–61.
[6] G. hai Liu, J. Zhang, W. xiang Zhao, Y. Zhang, Y. Jiang, Internal model control based on support vector machines generalized inverse for two-motor variable frequency system applications, Proc. CSEE 31 (6) (2011) 85–91.
[7] X. xing Liu, Y. wen Hu, Dynamic decoupling control of pmsm based on neural network inverse method, Proc. CSEE 27 (27) (2011) 72–76.
[8] Y. kun Sun, Y. Ren, Y. hong Huang, Decoupling control between radial force and torque of bearingless switched reluctance motors based on neural network inverse system method, Proc. CSEE 28 (9) (2008) 81–85.
[9] X. long Zhao, Y. hong Tan, Modeling hysteresis and its inverse model using neural networks based on expanded input space method, IEEE Trans. Control Syst. Technol. 16 (3) (2008) 484–490.
[10] D. chuan Yu, F. Liu, P.Y. Lai, et al., Nonlinear dynamic compensation of sensors using inverse-model-based neural network, IEEE Trans. Instrum. Meas. 57 (10) (2008) 2364–2376.
[11] G. hai Liu, Y. Zhang, H. feng Wei, W. xiang Zhao, Least squares support vector machines inverse control for two-motor variable frequency speed-regulating system based on active disturbances rejection, Proc. CSEE 32 (6) (2012) 138–144.
[12] G. hai Liu, J. feng Xue, M. Kang, P. yuan Liu, On-line adjustment control of two motor speed-regulating system based on neural network generalized inverse, Electr. Mach. Control 13 (4) (2009) 511–515.
[13] B. Li, Y. bin Li, Chaotic time series prediction based on elm learning algorithm, J. Tianjin Univ. 44 (8) (2011) 701–704.
[14] M. Han, D.-C. Li, An norm 1 regularization term ELM algorithm based on surrogate function and Bayesian framework, Acta Autom. Sin. 37 (11) (2011) 1344–1350.
[15] W.-Y. Deng, Q.-H. Zheng, L. Chen, X.-B. Xu, Research on extreme learning of neural networks, Chin. J. Comput. 33 (2) (2010) 279–287.
[16] G. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machines for regression and multi-class classification, IEEE Trans. Syst. Man Cybern. B Cybern. 42 (2) (2012) 513–529.
[17] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[18] G.-B. Huang, D. Wang, L. Yuan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2) (2011) 107–122.
[19] G. Huang, Q. Zhu, C. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of International Joint Conference on Neural Networks, vol. 2, 2004, pp. 985–990.
[20] Y. Lan, Y. Soh, G. Huang, Ensemble of online sequential extreme learning machine, Neurocomputing 2 (13–15) (2009) 3391–3395.
[21] M.B. Li, G.B. Huang, P. Saratchandran, N. Sundararajan, Fully complex extreme learning machine, Neurocomputing 68 (2005) 306–314.
[22] N.Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.
[23] H.X. Tian, Z.Z. Mao, An ensemble ELM based on modified adaboost. RT algorithm for predicting the temperature of molten steel in ladle furnace, IEEE Trans. Autom. Sci. Eng. 7 (1) (2010) 73–80.
[24] H. Rong, G. Huang, N. Sundararajan, P. Saratchandran, Online sequential fuzzy extreme learning machine for function approximation and classification problems, IEEE Trans. Syst. Man Cybern. B Cybern. 39 (4) (2009) 1067–1072.
[25] K.A.Z.L. Sun, T. Choi, A hybrid neuron-fuzzy inference system through integration of fuzzy logic and extreme learning machines, IEEE Trans. Syst. Man Cybern. B Cybern. 37 (5) (2007) 1321–1331.
[26] Z. Sun, T. Choi, K.F. Au, Y. Yu, Sales forecasting using extreme learning machine with applications in fashion retailing, Decis. Support Syst. 46 (1) (2008) 411–419.
[27] R. Meir, V.E. Maiorov, On the optimality of neural-network approximation using incremental algorithms, IEEE Trans. Neural Netw. 11 (2) (2000) 323–337.
[28] H.-L.W. Xian Zhang, Incremental regularized extreme learning machine based on Cholesky factorization and its application to time series prediction, Acta Phys. Sin. 60 (11) (2011) 1–6.
[29] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[30] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3062.
[31] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (2008) 3460–3468.
[32] G. Feng, G.-B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw. 20 (8) (2009) 1352–1357.
[33] B. Fang, J. Zhou, Y. Li, Matrix Theory, Tsinghua University Press, Beijing, China, 2006.

Jin-Lin Ding received both the B.S. and the M.S. degree in Jiangsu University, China. In 2005, she joined the Department of Electronic Information Engineering, Suzhou Vocational University, Jiangsu, China. During graduate school, she was engaged in decoupling control of multi-motor variable frequency speed-regulating system. Her current research interests include Electrical Machines Drive and Control, Intelligent Control and Measure, and so on. In these areas, she has published many refereed papers in technical journal and conference proceedings, and a few patents of utility model. She has also served as a member of Suzhou Science and Technology Association, China.

Feng Wang received the B.Sc. degree and the M.S. degree in China University of Mining And Technology. From 2005, he worked at the Department of Electronic Information Engineering, Suzhou Vocational University. At present, he is working toward the Ph.D. degree at the School of Electrical and Information Engineering, Jiangsu University, China. His research interests include Signal processing, Artificial Neural Networks and Intelligent Control.

Hong Sun received the Eng. D. degree in Mechanical Electronic Engineering from the School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007. Now, she is a teacher in the Department of Electronic Information Engineering, Suzhou Vocational University, Suzhou, China. Her current research interests include Intelligent Control, Industrial Control, and Robotics.

Li Shang received the B.Sc. degree and the M.Sc. degree in Xi'an Mine University in June 1996 and June 1999, respectively. And in June 2006, she received the Doctor's degree in Pattern Recognition & Intelligent System in University of Science & Technology of China (USTC), Hefei, China. From July 1999 to July 2006, she worked at USTC, and applied herself to teaching. Now, she works at the Department of Electronic Information Engineering, Suzhou Vocational University. At present, her research interests include Image processing, Artificial Neural Networks and Intelligent Computing.