

Research on Neural Network Inverse model of Induction motor Drives

Wu Qinghui

College of Information Science and Engineering
Bohai University
Jinzhou, China

Lun Shuxian, Yin Zuoyou and Guo Zhaozheng

College of Information Science and Engineering
Bohai University
Jinzhou, China

Abstract—Since the realization of inverse model is very important for inverse decoupling control of induction motor (IM) drives, the purpose of this paper is to develop an efficient artificial neural network (ANN) based inverse model for IM drives. First, the existence of the inverse system for IM drives is proved by inverse system theory. However, the analytic inverse model is hardly applied in the engineering since it excessively depends on the parameters. Then a novel neural network based inverse model, which synthesizes non-analytic method and analytic method, is suggested in this paper. To accelerate the convergence speed of ANN and enhance its generalization ability, the nonlinear parts are realized by the analytic expressions and the corresponding results act as the inputs of network. A three-layered feed-forward ANN with 11-40-2 structure is introduced to approach the inverse mode of IM drives. This study shows that the procedure using ANN based inverse model is applicable to substitute the analytic inverse model of IM drives. Simulation results are given to verify the developed models.

Keywords—neural network; inverse decoupling control; induction motor drives; inverse system; non-analytic method

I. INTRODUCTION

Induction motors are a theoretically interesting and practically important class of nonlinear systems which constitutes a benchmark example for nonlinear control [1]. The control task is further complicated by the fact that induction motor is a nonlinear, multivariable system, and there is coupling between the stator and the rotor circuit, in addition, the parameters are highly uncertain. In order to control electrical torque exactly, the electrical torque is decoupled from the flux in the course of transient and steady processes [1]. However, in the vector control (VC) method, the decoupled relationship is obtained by the stator current vector divided into torque part and flux part in the rotor flux-oriented coordinates under the hypothesis that the rotor flux is kept constant. So the electrical torque is only decoupled asymptotically from rotor flux during steady state. Recently, decoupling control methods based on inverse system theory have been used in the design of induction motor drives for high performance applications [2].

Artificial neural network (ANN) have been used in various control field with the characteristics of self-adaptive and learning, nonlinear mapping, strong robustness and fault-tolerance. In the past few years, ANN technique has been also studied in electric drive [3-5]. In this paper, an ANN Inverse decoupling control of torque and stator flux is presented. First, the existence of the inverse system for IM drives is approved

by inverse system theory. To accelerate the convergence speed of neural network and enhance its generalization ability, a novel method of synthesizing neural network and analytic function is suggested, in which the nonlinear parts are realized by the analytic expressions and the corresponding results act as the inputs of network. A three-layered feed-forward ANN with 11-40-2 structure is introduced to approach the inverse mode of IM drives. Simulation results shows the feasibility of the proposed scheme.

II. DYNAMIC MODEL OF INDUCTION MOTOR DRIVE SYSTEM

For an induction motor, if the stator current and stator flux are selected as the state variables, the state equation is described as Eq. (1) in the stationary reference frame:

$$\left\{ \begin{array}{l} p \cdot i_{sd} = -\frac{1}{\sigma L_s} \left(R_s + \frac{R_r L_s}{L_r} \right) i_{sd} - n_p \omega_r i_{sq} + \frac{R_r}{\sigma L_s L_r} \psi_{sd} \\ \quad + \frac{n_p}{\sigma L_s} \omega_r \psi_{sq} + \frac{1}{\sigma L_s} u_{sd} \\ p \cdot i_{sq} = n_p \omega_r i_{sd} - \frac{1}{\sigma L_s} \left(R_s + \frac{R_r L_s}{L_r} \right) i_{sq} - \frac{n_p}{\sigma L_s} \omega_r \psi_{sd} \\ \quad + \frac{R_r}{\sigma L_s L_r} \psi_{sq} + \frac{1}{\sigma L_s} u_{sq} \\ p \cdot \psi_{sd} = -R_s i_{sd} + u_{sd} \\ p \cdot \psi_{sq} = -R_s i_{sq} + u_{sq} \\ p \cdot \omega_r = \frac{1.5 n_p}{J} (i_{sq} \psi_{sd} - i_{sd} \psi_{sq}) - \frac{1}{J} T_l \end{array} \right. \quad (1)$$

where R_s and R_r are stator and rotor resistances, u_{sd} and u_{sq} are the stator voltage d-q part, i_{sd} and i_{sq} are stator current d-q part, i_{rd} and i_{rq} are rotor current d-q part, ψ_{sd} and ψ_{sq} are stator flux d-q part, ω_r is the rotor speed, n_p is the number of pole pairs and p is the derivative operator.

The object of control system is keep the stator flux altitude constant and control electromagnetic torque exactly. Then the output equations are defined as

$$\mathbf{y} = \begin{bmatrix} T_e \\ \phi \end{bmatrix} = \begin{bmatrix} 1.5n_p (i_{sq}\psi_{sd} - i_{sd}\psi_{sq}) \\ \sqrt{\psi_{sd}^2 + \psi_{sq}^2} \end{bmatrix} \quad (2)$$

The mathematical model of an induction motor can be expressed in the classical state-space representation as Eq. (3)

$$\begin{cases} p \cdot \mathbf{x} = f(\mathbf{x}) + B\mathbf{u} \\ \mathbf{y} = h(\mathbf{x}) \end{cases} \quad (3)$$

where

$$f(\mathbf{x}) = \begin{bmatrix} k_1x_1 - k_2x_2x_5 + k_3x_3 + k_4x_4x_5 \\ k_2x_1x_5 + k_1x_2 - k_4x_3x_5 + k_3x_4 \\ k_5x_1 \\ k_5x_2 \\ k_6(x_2x_3 - x_1x_4) - k_7T_l \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{sd} \\ \psi_{sq} \\ \omega_r \end{bmatrix}$$

$$h(\mathbf{x}) = \begin{bmatrix} k_8(x_2x_3 - x_1x_4) \\ \sqrt{x_3^2 + x_4^2} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix},$$

$$B = \begin{bmatrix} k_9 & 0 \\ 0 & k_9 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad k_1 = -\frac{1}{\sigma L_s} \left(R_s + \frac{R_r L_s}{L_r} \right), \quad k_2 = n_p,$$

$$k_3 = \frac{R_r}{\sigma L_s L_r}, \quad k_4 = \frac{n_p}{\sigma L_s}, \quad k_5 = -R_s, \quad k_6 = 1.5n_p/J, \\ k_7 = 1/J, \quad k_8 = 1.5n_p, \quad k_9 = 1/\sigma L_s$$

It can be seen from Eq. (1) to Eq. (3) that the induction motor is a 2-input, 2-output, nonlinear, strong coupling system.

III. ANALYSIS OF INVERSE SYSTEM

According to Eq. (3), it can be concluded that there is indirect relations between output variables and input control variables. In order to get direct relations, the output equation is differentiated by time as

$$\mathbf{y}^{(u)} = \begin{bmatrix} y_1^{(\alpha_1)} \\ y_2^{(\alpha_2)} \end{bmatrix} = \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \end{bmatrix} = g(\mathbf{x}) + A \cdot \mathbf{u} \quad (4)$$

where

$$g(\mathbf{x}) = \begin{bmatrix} k_6 [k_1(x_2x_3 - x_1x_4) + k_2(x_1x_3 + x_2x_4) - k_4x_5(x_3^2 + x_4^2)] \\ (k_5x_1x_3 + k_5x_4) / \sqrt{x_3^2 + x_4^2} \end{bmatrix}$$

$$A = \begin{bmatrix} k_6x_2 & -k_6x_1 \\ \frac{x_3}{\sqrt{x_3^2 + x_4^2}} & \frac{x_4}{\sqrt{x_3^2 + x_4^2}} \end{bmatrix}$$

In terms of invertible theorem in [3], the system is invertible. From Eq. (4), the control effort can be expressed as

$$\mathbf{u} = A^{-1}(\mathbf{x}) (\mathbf{y}^{(1)} - g(\mathbf{x})) \quad (5)$$

The inverse model and the original model constitute a generalized induction motor system, which is characterized by decoupling and linearization.

Dynamic model of the general system after decoupling control is described as

$$\mathbf{y}^{(1)} = \mathbf{v}, \quad i.e. \quad \begin{bmatrix} \dot{T}_e \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (6)$$

Under the assumption of exact decoupling by using Eq. (6), the decoupling control system with classic PID can obtain perfect torque response performance. But due to state estimation errors and IM parameter variations, the decoupling precision gets worse and the control performance is destroyed in the realistic applications.

IV. NEURAL NETWORK BASED INVERSE MODEL

In order to overcome the defect that the analytic method excessively relies on IM model and its parameters, an ANN based inverse decoupling control is researched in this paper.

A. Neural Network Design

According to Eq. (5), there are static relations but not dynamic relations between states and control effort in the inverse system. So a static neural network can be chosen to approach the nonlinear static function. In order to simplify the neural network structure, the Eq. (5) can be modified as

$$\mathbf{d} = A(\mathbf{x})\mathbf{u} = (\mathbf{y}^{(1)} - g(\mathbf{x})) \quad (7)$$

To accelerate the convergence speed of ANN, the nonlinear operations are realized by the analytic operation method and the corresponding results act as the inputs of network. Then the ANN is expressed linear structure as

$$O = net(I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11}) \quad (8)$$

where

$$I_1 = dy_1/dt; I_2 = dy_2/dt; I_3 = x_2x_3; I_4 = x_1x_4;$$

$$I_5 = x_1x_3 + x_2x_4; I_6 = x_5I_5; I_7 = x_3^2 + x_4^2; I_8 = x_5I_7;$$

$$I_9 = I_5/\sqrt{I_7}; I_{10} = x_3/\sqrt{I_7}; I_{11} = x_4/\sqrt{I_7}.$$

For Eq. (10), the network output is the linear combination of the inputs. Feed-forward ANN is used to construct an ANN decoupling controller. Generally, a three-layered feed-forward ANN with appropriate network structure and weights can approach any continuous function. So a 11-40-2 network is constructed, in which the neuron activation function of input layer is chosen as linear function $f_i(x) = x$; that of hidden

layer is the Sigmoid function $f_h(x) = 1/(1+e^{-x})$; and that of output layer is linear function $f_o(x) = x$.

B. Data Processing

The input I_1 and I_2 of ANN are very sensitive by high-frequency noise for using differential operates. Therefore, it is necessary to design digital low-pass filter for torque and stator flux.

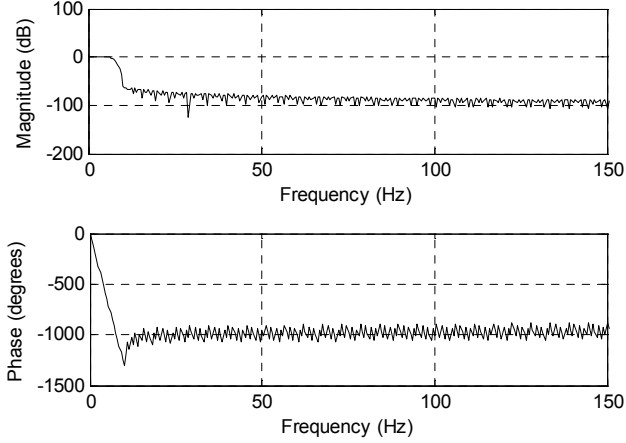


Figure 1. Amplitude and phase frequency characteristics of FIR filter

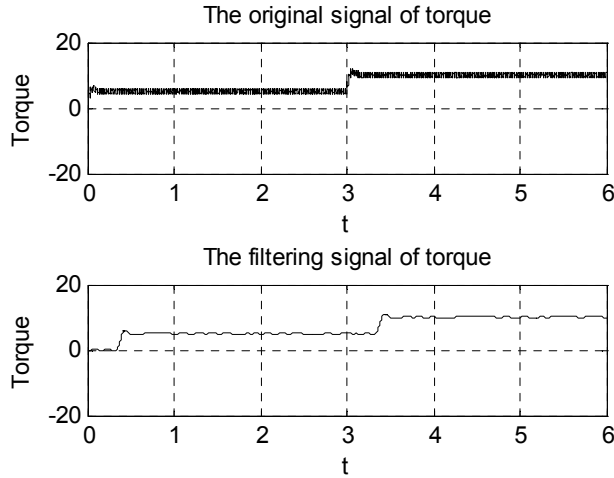


Figure 2. Comparative results of torque signal filter processing

A linear phase FIR filter is designed in this paper, whose idea is that according to given performance index of filter, the length N and window function W_n of filter are chosen, then finite unit impulse response sequence is determined by adding window. Its performance index demands include: 1) passband cutoff frequency is 10Hz; 2) stopband cutoff frequency is 15Hz; 3) passband ripple is 0.0005; 4) stopband ripple is 0.001; 5) sample frequency is 1kHz. The FIR filter is easy to be designed by using the order function and filter function of MATLAB (a part code seen in Appendix A), and its magnitude and phase frequency characteristic is seen in Fig. 1., and its comparative results are seen in fig. 2-3, which shows the high-frequency noise is well restrained.

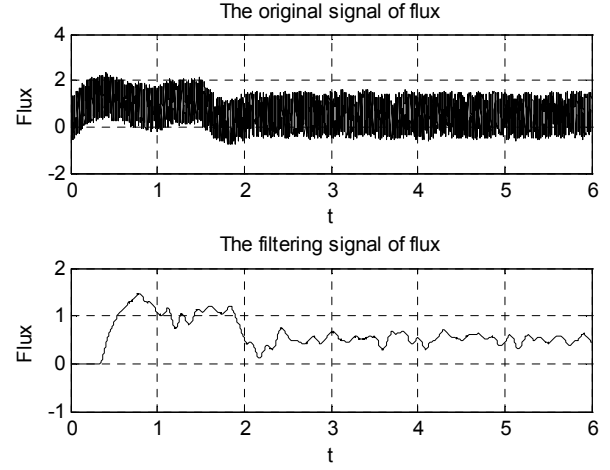


Figure 3. Comparative results of flux signal filter processing

V. SIMULATION RESULTS

The simulations have been performed for a standard 4kW, 4-pole, 220v, 50Hz, squirrel-cage induction motor having the following parameters:

Number of pole pairs: $n_p = 2$;

Stator resistance: $R_s = 0.435\Omega$;

Stator resistance: $R_r = 0.816\Omega$;

Stator inductance: $L_s = 73.31 \text{ mH}$;

Rotor inductance: $L_r = 73.31 \text{ mH}$;

Mutual inductance: $L_m = 69.31 \text{ mH}$;

Inertia: $J = 0.239 \text{ kg.m}^2$;

Friction coefficient: $B = 0.267 \text{ N.m.s/rad}$.

The sampling data set are produced by sampling frequency $f_s = 200\text{Hz}$, whose number is 1200, in which 2/3 of data sample is used to train ANN and the rest is applied to verify the ability of generalization. The goal of train-parameter is set 0.25. The training results is shown in Fig. 4, which shows the performance goal met after 4 epochs, *i.e.* good convergence performance.

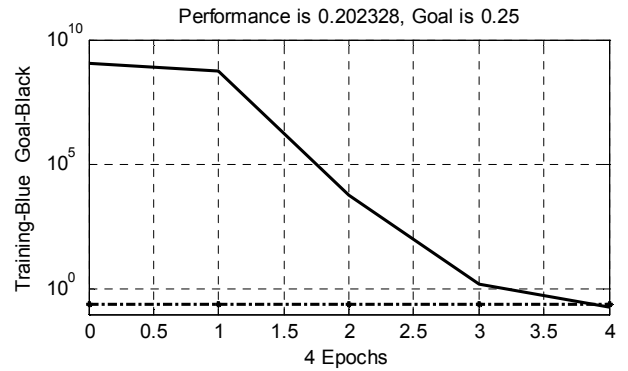


Figure 4. ANN training process

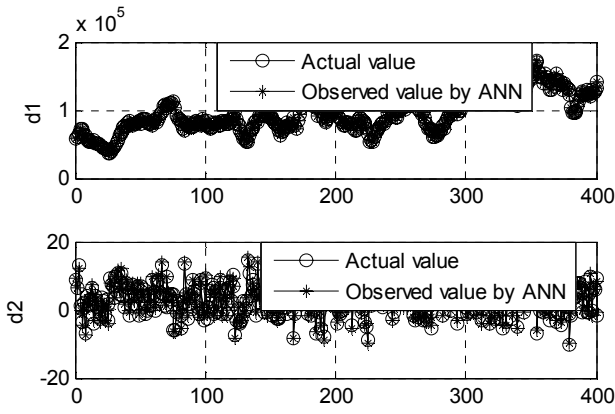


Figure 5. Comparative results between actual value and observed value by ANN

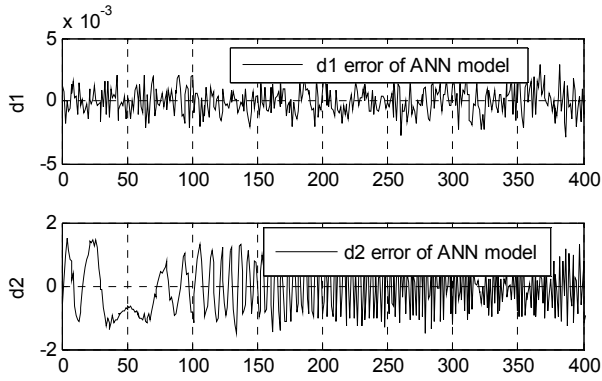


Figure 6. Observe error of ANN

The generalization ability tests are shown in fig. 5-6, in which fig. 5 is the comparative results between actual output and ANN output, and fig. 6 is the observed error of ANN. The part of code has been shown in Appendix B. The simulation results suggests that the ANN has good generalization ability.

VI. CONCLUSIONS

ANN based inverse model of IM is deeply researched in this paper. The major contributions of this paper are organized as followings: 1) the well restraining high-frequency noise; 2) the successful application of ANN with analytic function; and 3) the successful training of the proposed ANN.

ACKNOWLEDGMENT

The authors would like to acknowledge the economic support received from Liaoning Provincial Natural Science Foundation of China for realizing this work under the 20072199 Project.

REFERENCES

[1] R. Ortega, N. Barabanov and G. Escobar, "Direct control of induction motors: stability analysis and performance improvement," IEEE Transaction on Automatic Control, vol. 46, no. 8, pp. 1209-1221, 2001.

[2] H. Wang, W. Xu, T. Shen and G. Yang, "Stator Flux and Torque Decoupling Control for Induction Motors with Resistances Adaptation," IEE Proc. -Control Theory Appl, vol. 152, no. 4, pp. 363-370, 2005.

[3] E. Stanislav, "Input-output Decoupling Control of Induction Motors with Rotor Resistance and Load Torque Identification," in IEEE Mediterranean Conference on Control and Automation, Athens, pp. 24-27, 2007.

[4] Y. Jianghua, L. Wei and L. Wenqi, "PID Controller Based on the Artificial Neural Network", in International Symposium on Neural Networks, LNCS 3174, Dalian, pp. 144-149, 2004.

[5] W. Qinghui, L. Yi, Z. Dianjun and Z. Yonghui, "Adaptive Control for Induction Servo Motor Based on Wavelet Neural Networks," in International Symposium on Neural Networks, LNCS 3174, Dalian, pp. 156-162, 2004.

[6] W. Rong-Jong and C. Jia-Ming, "Intelligent Control of Induction Servo Motor Drive via Wavelet Neural Network," Electric Power System Research, vol. 61, pp. 67-76, 2002.

[7] Z. Xinghua and D. Xianzhong, "Speed Control System of Induction Motor Based on Inverse System Method," Control and Decision, vol. 15, no. 6, pp. 708-711, 2000 (in Chinese).

Appendix A

```
% FIR filter design
fn=[5 10];
A=[1 0];
dev=[0.005 0.001];
fs1=10^3;
[N,Wn,beta,ftype]=kaiserord(fn,A,dev,fs1);
hn=fir1(N,Wn,ftype,kaiser(N+1,beta));
figure(5)
freqz(hn,1,fs1,fs1);
% filter computing of torque
for k=1:count8
    y1_1d(k)=0;
    for j=1:N+1
        if k>j
            break;
        end
    end
    y1_1d(k)=y1_1d(k)+hn(j)*Torque(k-j);
end
figure(7)
subplot 211
plot(t,Torque);grid;title('The original
signal of torque');
subplot 212
plot(t,y1_1d);grid;title('The filtering
signal of torque');
% filter computing of flux
for k=1:count8
    y2_1d(k)=0;
    for j=1:N2+1
        if k>j
            break;
        end
    end
    y2_1d(k)=y2_1d(k)+hn(j)*Flux(k-j);
end
```

```

            break;
        end
    end
end
figure(8)
subplot 211
plot(t,Flux);grid;title('The original
signal of flux');
subplot 212
plot(t,y2_1d);grid;title('The filtering
signal of flux');

```

Appendix B

```
%induction motor parameters
```

```

Rs=0.435;
Rr=0.816;
Lls=4.0e-3;
Llr=2.0e-3;
Lm=69.31e-3;
J=0.089;
np=2;
Ls=Lls+Lm;
Lr=Llr+Lm;
Rou=1-Lm^2/(Ls*Lr);
k10=1.5*np;
k9=1/(Rou*Ls);
for i=1:count1
    A=[k10*(x2(i)-k9*x4(i)) k10*(k9*x3(i)-
x1(i));x3(i)/sqrt(x3(i)^2+x4(i)^2)
x4(i)/sqrt(x3(i)^2+x4(i)^2)];
    U_1(:,i)=A*U(:,i);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Neural network input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I1=y1;
I2=y2;
I3=x2.*x3;
I4=x1.*x4;
I5=x1.*x3+x2.*x4;
I6=x5.*I5;
I7=x3.^2+x4.^2;
I8=I7.*x5;
I9=I5./sqrt(I7);
I10=x3./sqrt(I7);
I11=x4./sqrt(I7);
p1=I1';
p2=I2';
p3=I3';
p4=I4';
p5=I5';

```

```

p6=I6';
p7=I7';
p8=I8';
p9=I9';
p10=I10';
p11=I11';
t1=U_1(1,:);
t2=U_1(2,:);
p=[p1;p2;p3;p4;p5;p6;p7;p8;p9;p10;p11];
t=[t1;t2];
P1=p(:,1:(count1-1)*2/3);
T1=t(:,1:(count1-1)*2/3);
net1=newff(minmax(p),[40 2]);
net1.layers{:,1}.transferFcn='tansig';
net1.trainParam.epochs=10000;
net1.trainParam.goal=0.25;
net1.trainParam.mu_max=2.0e+020;
net1.trainParam.min_grad=1e-30;
net_1=train(net1,P1,T1);
P_test=p(:,(count1-1)*2/3:count1);
tt=sim(net_1,P_test);
T_1=tt;
T_2=t(:,(count1-1)*2/3:count1);
E=T_1-T_2;
figure(1)
subplot 211
plot(E(1,:))
subplot 212
plot(E(2,:))
m=size(T_1);
n=1:1:m(2);
figure(2);
subplot 211
plot(n,T_1(1,:),'-o')
hold on
plot(n,T_2(1,:), 'r-*')
grid on
legend('Actual value','Observed value by
ANN');
xlabel('Sample dot');
ylabel('v2');
hold off
subplot 212
plot(n,T_1(2,:), '-o')
hold on
plot(n,T_2(2,:), 'r-*')
hold off
grid on
legend('Actual value','Observed value by
ANN');
xlabel('Sample dot');
ylabel('v2');

```