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. Simulation results are shown to verify 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:

\[
\begin{align*}
p \cdot i_{sd} &= -\frac{1}{\sigma L_s} \left( R_s + \frac{R_p L_s}{L_r} \right) i_{sd} - n_p \omega_r i_{sq} + \frac{R_r}{\sigma L_s} \psi_{sd} \\
&\quad + \frac{n_p}{\sigma L_s} \omega_r \psi_{sq} + \frac{1}{\sigma L_s} u_{sd} \\
p \cdot i_{sq} &= n_p \omega_r i_{sd} - \frac{1}{\sigma L_s} \left( R_s + \frac{R_p L_s}{L_r} \right) i_{sq} - \frac{n_p}{\sigma L_s} \omega_r \psi_{sd} \\
&\quad + \frac{R_r}{\sigma L_s} \psi_{sq} + \frac{1}{\sigma L_s} u_{sq} \\
p \cdot \psi_{sd} &= -R_s i_{sd} + u_{sd} \\
p \cdot \psi_{sq} &= -R_s i_{sq} + u_{sq} \\
p \cdot \omega_r &= \frac{1.5 n_p}{J} (i_{sq} \psi_{sd} - i_{sd} \psi_{sq}) - \frac{1}{J} T_j
\end{align*}
\]

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, \( \psi_{sd} \) and \( \psi_{sq} \) are stator flux d-q part, \( \omega_r \) is the rotor speed, \( n_p \) is the number of pole pairs and \( p \) is the derivative operator.

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 are given to verify the feasibility of the proposed scheme.

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. 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 are given to verify 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:

\[
\begin{align*}
p \cdot i_{sd} &= -\frac{1}{\sigma L_s} \left( R_s + \frac{R_p L_s}{L_r} \right) i_{sd} - n_p \omega_r i_{sq} + \frac{R_r}{\sigma L_s} \psi_{sd} \\
&\quad + \frac{n_p}{\sigma L_s} \omega_r \psi_{sq} + \frac{1}{\sigma L_s} u_{sd} \\
p \cdot i_{sq} &= n_p \omega_r i_{sd} - \frac{1}{\sigma L_s} \left( R_s + \frac{R_p L_s}{L_r} \right) i_{sq} - \frac{n_p}{\sigma L_s} \omega_r \psi_{sd} \\
&\quad + \frac{R_r}{\sigma L_s} \psi_{sq} + \frac{1}{\sigma L_s} u_{sq} \\
p \cdot \psi_{sd} &= -R_s i_{sd} + u_{sd} \\
p \cdot \psi_{sq} &= -R_s i_{sq} + u_{sq} \\
p \cdot \omega_r &= \frac{1.5 n_p}{J} (i_{sq} \psi_{sd} - i_{sd} \psi_{sq}) - \frac{1}{J} T_j
\end{align*}
\]

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, \( \psi_{sd} \) and \( \psi_{sq} \) are stator flux d-q part, \( \omega_r \) 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 and control electromagnetic torque exactly. Then the output equations are defined as
where

\[
y = \begin{bmatrix} T_e \\ \phi \end{bmatrix} = \begin{bmatrix} 1.5n_p(i_{sd}\psi_{sd} - i_{sd}\psi_{sq}) \\
\frac{\psi_{sq}^2 + \psi_{sd}^2}{2} \end{bmatrix}
\]

(2)

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

\[
\begin{align*}
p \cdot x &= f(x) + Bu \\
y &= h(x)
\end{align*}
\]

(3)

where

\[
f(x) = \begin{bmatrix} k_1 x_1 - k_2 x_2 x_5 + k_3 x_3 + k_4 x_4 x_5 \\
k_2 x_1 x_5 + k_2 x_2 - k_4 x_3 x_5 + k_3 x_4 \\
k_3 x_2 \\
k_6 (x_2 x_3 - x_1 x_4) - k_7 T_l 
\end{bmatrix},
\]

\[
x = \begin{bmatrix} x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 
\end{bmatrix},
\]

\[
h(x) = \begin{bmatrix} k_9 (x_2 x_3 - x_1 x_4) \\
\sqrt{x_3^2 + x_4^2} 
\end{bmatrix},
\]

\[
u = \begin{bmatrix} u_{sd} \\
u_{sq} \end{bmatrix}
\]

(4)

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

\[
d = A(x)u = \begin{bmatrix} y^{(1)} - g(x) \end{bmatrix}
\]

(5)

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

\[
d = A(x)u = \begin{bmatrix} y^{(1)} - g(x) \end{bmatrix}
\]

(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

\[
O = net\left(\begin{bmatrix} I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11} \end{bmatrix} \right)
\]

(8)

where

\[
I_1 = \frac{dy_1}{dt}, \quad I_2 = \frac{dy_2}{dt}, \quad I_3 = x_2 x_3, \quad I_4 = x_1 x_4;
\]

\[
I_5 = x_1 x_3 + x_2 x_4, \quad I_6 = x_3 I_5, \quad I_7 = x_3^2 + x_4^2, \quad I_8 = x_3 I_7;
\]

\[
I_9 = \frac{I_5}{\sqrt{I_7}}, \quad I_{10} = \frac{x_3}{\sqrt{I_7}}, \quad I_{11} = x_4 / \sqrt{I_7}.
\]

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 \( f_h(x) = \frac{1}{1+e^{-x}} \); 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.

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.

**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: \( L_s = 73.31\ mH \);
- Rotor inductance: \( L_r = 73.31\ mH \);
- Mutual inductance: \( L_m = 69.31\ mH \);
- 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 \( f_s = 200\ Hz \), 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.
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)
dfreqz(hn,1,fs1,fs1);
% filter computing of torque
for k=1:count8
    y1_1d(k)=0;
    for j=1:N+1
        if k>j
            y1_1d(k)=y1_1d(k)+hn(j)*Torque(k-j);
        end
        break;
    end
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
            y2_1d(k)=y2_1d(k)+hn(j)*Flux(k-j);
        end
    end
end
break;
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*Rs);
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);
et1=newff(minmax(p),[40 2]);
et1.layers{:}.transferFcn='tansig';
et1.trainParam.epochs=10000;
et1.trainParam.goal=0.25;
et1.trainParam.mu_max=2.0e+020;
et1.trainParam.min_grad=1e-30;
et_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
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');