A Novel Neural Network Vector Control Technique for Induction Motor Drive

Xingang Fu, *Student Member, IEEE*, and Shuhui Li, *Senior Member, IEEE*

***Abstract*—This paper proposes a novel neural network (NN)-based vector control method for a three-phase induction motor. The proposed NN vector control utilizes the rotor flux-oriented reference frame, and the role of the NN controller is to substitute the two decoupled current-loop proportional-integral (PI) controllers in conventional vector control techniques. The objective of NN training is to approximate optimal control and the NN controller was trained by Levenberg–Marquardt (LM) algorithm. Forward Accumulation Through Time algorithm for induction motor was developed to calculate Jacobian matrix needed by the LM algorithm. The simulations showed that the NN vector control can provide better current tracking ability than the conventional vector control, such as less oscillations and low harmonics. Especially, the NN vector control can better overcome the problem of detuning effects than the conventional vector control. The hardware experiments further demonstrated the great advantage of the NN vector control. The NN vector control can succeed in driving the induction motor without audible noise using relatively lower switching frequency or lower sampling rate compared with the conventional vector control, and thus has the potential to improve efficiency and reduce size and cost of an induction motor drive system.**

***Index Terms*—Approximate optimal control, forward accumulation through time, induction motor, Levenberg–Marquardt, neural network vector control.**

I. Introduction

NOWADAYS, three-phase induction motors have been widely used in industrial drives as they are rugged, reliable, and economical. Variable-frequency drives (VFDs) are becoming more popular in variable-speed services because VFDs offer especially important energy-saving opportunities. Thus, many researchers have studied VFD-based control strategies for high performance of induction motor drives. One of these popular methods is the vector control method [1].

Vector control of the induction motor can guarantee the decoupling of torque and flux control and thus control the induction motor linearly as a separated excited dc motor to offer better dynamic response [1]. However, recent studies indicated that the conventional standard vector control strategy has a competing control nature in its current control loop so that true decoupled d- and q-axis loop control does not exist [2]. As a result, the control performance is more sensitive to uncertainties such as unpredictable parameter variation, external load disturbances, and so on.

Therefore, it is important to investigate an effective and intelligent control scheme for enhanced induction motor drive performance. Most intelligent control research works focus on speed loop control based on neural networks. Peng and Zhao [3] proposed to use a radial basis function neural network in the speed loop to provide better speed control performance, which can avoid the limit calculation under the uncertainties in the induction motor. Ben-Brahim and Kurosawa [4] used backpropagation neural network to identify the mechanical speed of an induction motor and use neural network speed estimator to better control motor speed based on the conventional vector control method. However, the competing control nature associated with the current loop is not solved, and thus, the improvement only in speed loop does not help to achieve the true decoupled control. Some research proposed a neurocontroller for the current loop control of the induction motor used a Lyapunov-based online learning algorithm and showed its benefit, but the computational requirement of online learning is a big burden [5], [6].

In recent years, significant research works have been conducted in the area of dynamic programming (DP) [7] for optimal control of nonlinear systems [8]. Adaptive critic designs constitute a class of approximate dynamic programming (ADP) methods that use incremental optimization techniques combined with parametric structures that approximate the optimal cost and the control of a system [9], [10]. In [11], both heuristic dynamic programming and dual heuristic programming have been used to control a turbogenerator. In [12], an ADP-based neural network (NN) controller is trained and used to control a grid-connected converter system [12], [13], which demonstrated an excellent performance compared to the conventional standard vector controller.

However, no research has been conducted to develop current-loop vector controller for a three-phase induction motor using an ADP-based NN. The purpose of this paper is to investigate an NN-based vector control method for the induction motor. The special features and contributions of the paper include: 1) an approach to implement optimal vector control for an induction motor by using an NN; 2) an NN controller to substitute two decoupled proportioanl-integral (PI) controllers in current loop; and 3) a mechanism to train the NN controller by using a novel Levenberg–Marquardt (LM) + forward accumulation through time (FATT) algorithm.

The rest of the paper is structured as follows. The conventional vector control method for a three-phase induction motor is presented in Section II. Section III explains the proposed neural network vector control topology. How to train the NN vector

Manuscript received December 1, 2014; revised March 19, 2015; accepted May 18, 2015. Date of publication June 10, 2015; date of current version November 20, 2015. This work was supported in part by the U.S. National Science Foundation under Grant 1414379. Paper no. TEC-00802-2014.

The authors are with the Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL 35487 USA (e-mail: xfu@crimson.ua.edu; sli@eng.ua.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEC.2015.2436914

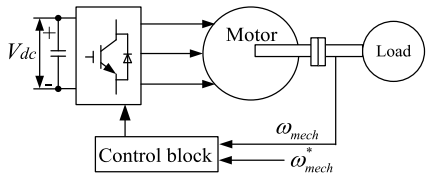

Fig. 1. Schematic of three phase induction motor drive.

controller is discussed in Section IV. Section V investigates the performance of the NN controller and compares it with conventional vector controllers in power converter switching environment. Section VI presents hardware experiment validation and comparison. Finally, the paper concludes with summary remarks in Section VII.

II. Conventional Vector Control

A. Induction Motor Model in d-q Frame

Fig. 1 shows a simple schematic of three-phase induction motor drive, in which a dc voltage source is on the left and the control block plays the role of regulating the motor speed through a voltage source pulse width modulation (PWM) converter.

In dq domain, the stator voltage equations is written as

$$\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} = R_s \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \frac{d}{dt}\begin{bmatrix} \lambda_{sd} \\ \lambda_{sq} \end{bmatrix} + \omega_{\rm sys}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \lambda_{sd} \\ \lambda_{sq} \end{bmatrix} \tag{1}$$

where R_s, $\omega_{\rm sys}$, v_{sd}, v_{sq}, i_{sd}, i_{sq}, λ_{sd}, and λ_{sq} stand for per-phase stator resistance, synchronous speed, stator d-axis voltage, stator q-axis voltage, stator d-axis current, stator q-axis current, stator d-axis flux, and stator q-axis flux, respectively.

Using the rotor flux-oriented frame, that is rotor q-axis flux $\lambda_{rq} = 0$, the rotor voltage equation is shown by

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = R_r \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \frac{d}{dt}\begin{bmatrix} \lambda_{rd} \\ 0 \end{bmatrix} + \omega_{\rm slip}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \lambda_{rd} \\ 0 \end{bmatrix} \tag{2}$$

where R_r, $\omega_{\rm slip}$, i_{rd}, i_{rq}, λ_{rd}, and λ_{rq} stand for per-phase rotor resistance, rotor slip speed, rotor d-axis current, rotor q-axis current, rotor d-axis flux, and rotor q-axis flux, respectively. Especially, the rotor slip speed $\omega_{\rm slip}$ equals the difference between the synchronous speed $\omega_{\rm sys}$ and rotor speed ω_m, e.g., $\omega_{\rm slip} = \omega_{\rm sys} - \omega_m$.

The flux linkage equation can be expressed as

$$\begin{bmatrix} \lambda_{sd} \\ \lambda_{sq} \\ \lambda_{rd} \\ 0 \end{bmatrix} = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix}\begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \\ i_{rq} \end{bmatrix} \tag{3}$$

where L_s, L_r, and L_m represent per-phase stator inductance, per-phase rotor inductance, and per-phase stator magnetizing inductance. Especially, $L_s = L_{\ell s} + L_m$ and $L_r = L_{\ell r} + L_m$. $L_{\ell s}$ and $L_{\ell r}$ refer to per-phase stator leakage inductance and per-phase rotor leakage inductance, respectively.

B. State-Space Model

The induction motor model (1)–(3) can be rearranged into the standard state-space representation shown by

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \end{bmatrix} = &-\begin{bmatrix} L_s & 0 & L_m \\ 0 & L_s - \dfrac{L_m^2}{L_r} & 0 \\ L_m & 0 & L_r \end{bmatrix}^{-1} \\ &\begin{bmatrix} R_s & -\omega_{\rm syn}(L_s - \dfrac{L_m^2}{L_r}) & 0 \\ \omega_{\rm syn}L_s & R_s & \omega_{\rm syn}L_m \\ \omega_r L_m & -R_r\dfrac{L_m}{L_r} & R_r + \omega_r L_r \end{bmatrix} \\ &\begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \end{bmatrix} + \begin{bmatrix} L_s & 0 & L_m \\ 0 & L_s - \dfrac{L_m^2}{L_r} & 0 \\ L_m & 0 & L_r \end{bmatrix}^{-1} \begin{bmatrix} v_{sd} \\ v_{sq} \\ 0 \end{bmatrix}. \end{aligned} \tag{4}$$

Define $\overrightarrow{i_{dqS}} = [i_{sd}\ i_{sq}\ i_{rd}]'$ and $\overrightarrow{u_{dqS}} = [v_{sd}\ v_{sq}\ 0]'$. To implement recurrent neural network (RNN)-based digital controller, the continuous state-space model (4) needs to be transferred into the equivalent discrete model (5) through either a zero-order or first-order hold discrete equivalent mechanism [14]

$$\overrightarrow{i_{dqS}}(k+1) = F\overrightarrow{i_{dqS}}(k) + G\overrightarrow{u_{dqS}}(k) \tag{5}$$

where k is an integer time step, F is the system matrix, and G is the matrix associated with the control voltage. Note that $\overrightarrow{i_{sdq}} = [i_{sd}\ i_{sq}]'$ are the state variables that needs to be controlled and $\overrightarrow{v_{sdq}} = [v_{sd}\ v_{sq}]'$ are the control actions.

C. Conventional Vector Control

The standard vector controller has a nested-loop structure consisting of a faster inner current loop and a slower outer speed and rotor flux loop. A fast inner current loop is important to ensure high-quality ac drive of an induction motor. The outer speed and rotor flux loop generates d- and q-axis current references to the inner current-loop controller, while the inner current-loop controller implements the final control function by applying a voltage signal to the PWM inverter.

The conventional control strategy [15] of the current loop is developed by rewriting (1)–(3) as

$$v_{sd} = \underbrace{\left(R_s i_{sd} + \sigma L_s \frac{d}{dt} i_{sd}\right)}_{v'_{sd}} + \left(\frac{L_m}{L_r}\frac{d}{dt}\lambda_{rd} - \omega_{\rm sys}\sigma L_s i_{sq}\right) \tag{6}$$

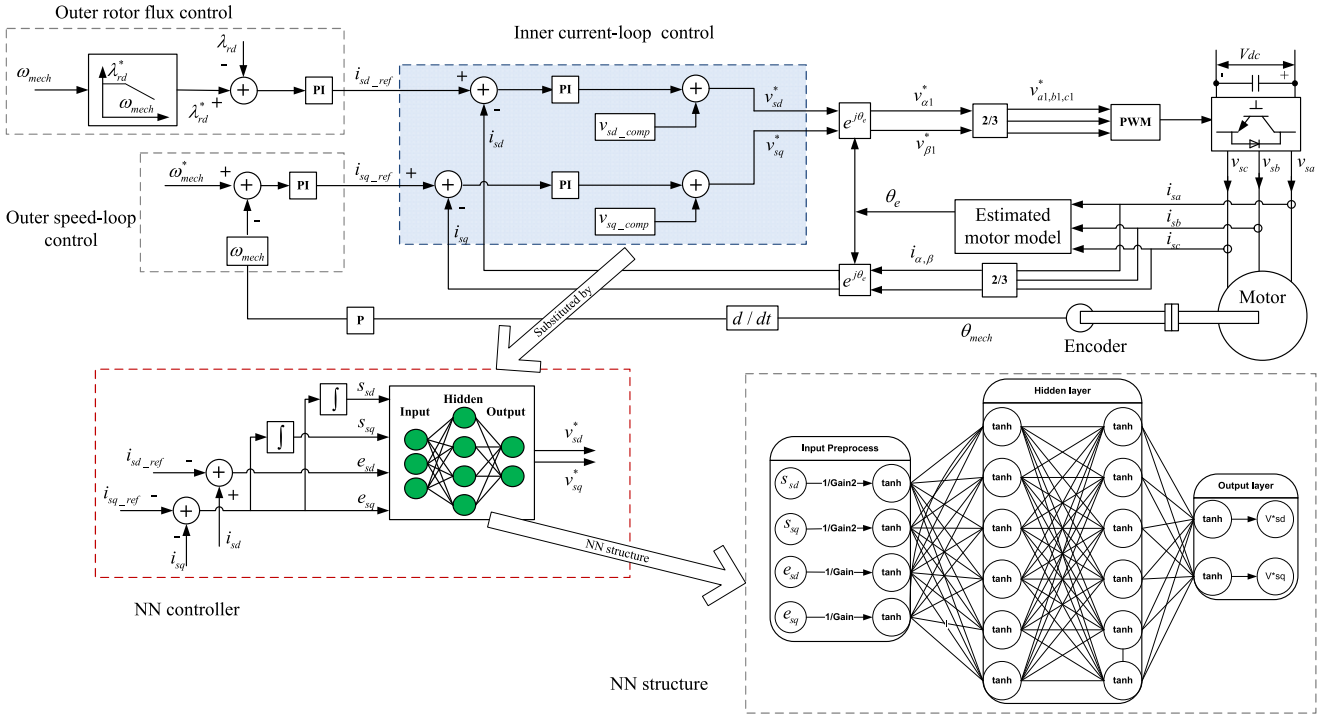

Fig. 2. Novel NN vector control for an induction motor.

$$v_{sq} = \underbrace{\left(R_s i_{sq} + \sigma L_s \frac{d}{dt} i_{sq}\right)}_{v'_{sq}} + \left(\omega_{\rm sys} \frac{L_m}{L_r} \lambda_{rd} + \omega_{\rm sys} \sigma L_s i_{sd}\right) \tag{7}$$

where σ is the leakage factor of the induction machine and defined as $\sigma = 1 - L_m^2/L_s L_r$ [1]. Those items denoted as v'_{sd} and v'_{sq} are treated as the state equations between the input voltages and output currents for the d- and q-axis current loops and the other terms are regarded as compensation items. Therefore, the corresponding transfer function $1/(R_S + s\sigma L_s)$ is used to design the current-loop controller [1].

The design of speed loop is based on the following equations [1]:

$$\frac{d}{dt}\omega_{\rm mech} = \frac{T_{em} - T_L}{J_{eq}},\ T_{em} = \underbrace{\frac{p}{2}\frac{L_m^2}{L_r} i_{sd}^*}_{K_T} i_{sq} \tag{8}$$

where p means pole numbers of the motor, i_{sd}^* is the reference value of i_{sd}, and K_T is constant.

III. Novel Neural Network Vector Control

A. Neural Network Vector Control Architecture

The proposed NN vector control architecture for the induction motor is shown in Fig. 2. The NN implements the fast inner current-loop control function. Due to the universal function approximation property [16], NN vector control, unlike conventional vector control, has the ability to achieve true decoupled torque and flux control. The outer control loops still utilize PI controllers and q-axis loop is used for speed control as shown in Fig. 2.

B. Neural Network Controller Structure

The proposed current-loop NN controller as shown in Fig. 2 contains two parts: input preprocessing block and a four-layer feed-forward network.

To avoid input saturation, the inputs are regulated to the range $[-1,\ 1]$ through a preprocessing procedure. The inputs to the feed-forward network are $\tanh(\overrightarrow{e_{sdq}}/\text{Gain})$ and $\tanh(\overrightarrow{s_{sdq}}/\text{Gain2})$, where $\overrightarrow{e_{sdq}}$ and $\overrightarrow{s_{sdq}}$ are error terms and integrals of the error terms. $\overrightarrow{e_{sdq}}$ is defined as $\overrightarrow{e_{sdq}}(k) = \overrightarrow{i_{sdq}}(k)$-$\overrightarrow{i_{sdq\_\rm ref}}(k)$ and $\overrightarrow{s_{sdq}}(k)$ is calculated by

$$\overrightarrow{s_{sdq}}(k) = \int_0^{kT_s} \overrightarrow{e_{sdq}}(t)dt \approx T_s \sum_{j=1}^{k} \frac{\overrightarrow{e_{sdq}}(j-1) + \overrightarrow{e_{sdq}}(j)}{2} \tag{9}$$

in which the trapezoid formula was used to compute the integral term $\overrightarrow{s_{sdq}}(k)$ and $\overrightarrow{e_{sdq}}(0) \equiv \overrightarrow{0}$.

The feed-forward network contains two hidden layers of six nodes each, and two output nodes, with hyperbolic tangent functions at all nodes. Two hidden layers were chosen to yield a stronger approximation ability [17]. The selection of the number of neurons in each hidden layer was done through trial and error tests. Basically, six nodes in each hidden layer can give good enough results.

Even though the feed-forward network in Fig. 2 does not have a feedback connection, the proposed NN current controller shown in Fig. 2 actually is a RNN because the current feedback signal generated by the system (4) acts as a recurrent network

connection from the output of NN back to the input. A recurrent network is potentially more powerful than a feedforward network and can exhibit temporal behavior, which is particularly important for feedback control applications [17].

According to Fig. 2, the NN current-loop controller can be denoted as $R(\overrightarrow{e_{sdq}}, \overrightarrow{s_{sdq}}, \overrightarrow{w})$, which is a function of $\overrightarrow{e_{sdq}}$, $\overrightarrow{s_{sdq}}$ and network weights $\overrightarrow{w}$. As the ratio of the converter output voltage $\overrightarrow{v_{sdq}}$ to the outputs of the current loop controller $\overrightarrow{v^*_{sdq}}$ is the gain of the PWM which is denoted as k_{PWM} [18], the control action $\overrightarrow{v_{sdq}}$ is then expressed by

$$\overrightarrow{v_{sdq}} = k_{\mathrm{PWM}}\overrightarrow{v^*_{sdq}} = k_{\mathrm{PWM}}R(\overrightarrow{e_{sdq}}, \overrightarrow{s_{sdq}}, \overrightarrow{w}). \tag{10}$$

IV. Training Neural Network Controller

A. Training Objective: Approximate Optimal Control

DP employs the principle of Bellman's optimality [7] and is a very useful tool for solving optimization and optimal control problems [9].

The DP cost function associated with the induction motor system is defined as

$$\begin{aligned} C_{dp} &= \sum_{k=j}^{\infty} \gamma^{k-j} U(\overrightarrow{e_{sdq}}(k)) \\ &= \sum_{k=j}^{\infty} \gamma^{k-j} \sqrt{[i_{sd}(k) - i_{sd\_\mathrm{ref}}(k)]^2 + [i_{sq}(k) - i_{sq\_\mathrm{ref}}(k)]^2} \end{aligned} \tag{11}$$

where γ is a discount factor with $0 < \gamma \leq 1$ and $j > 0$. U is called local cost or utility function. The function C_{dp}, depending on the initial time $j > 0$ and the initial state $\overrightarrow{i_{sdq}}(j)$, is referred to as the cost-to-go of state $\overrightarrow{i_{sdq}}(j)$ of the DP problem.

The objective of the training is to find an optimal trajectory of control action v_{dq1} that minimizes the DP cost C_{dp} in (11).

B. NN Training Algorithm: Levenberg–Marquardt + Forward Accumulation Through Time (LM + FATT)

1) Levenberg–Marquardt (LM) Algorithm: We chose to use the LM algorithm to train an NN because LM appears to be the fastest NN training algorithm for a moderate number of network parameters [19]. Also, LM usually can achieve better convergency performance than BPTT in training the RNN [20].

To implement LM training, the cost function defined in (11) needs to be rewritten in a sum-of-squares form. Consider the cost function C_{dp} with $\gamma = 1$, $j = 1$ and $k = 1, \ldots, N$; then, it can be written in the form

$$\begin{aligned} C_{dp} &= \sum_{k=1}^{N} U(\overrightarrow{e_{sdq}}(k)) \xLeftrightarrow{\text{def } V(k) = \sqrt{U(\overrightarrow{e_{sdq}}(k))}} \\ C_{dp} &= \sum_{k=1}^{N} V^2(k) \end{aligned} \tag{12}$$

and the gradient $\frac{\partial C_{dp}}{\partial w}$ can be written in a matrix product form

$$\frac{\partial C_{dp}}{\partial \overrightarrow{w}} = \sum_{k=1}^{N} V(k)\frac{\partial V(k)}{\partial w} = 2J_v(\overrightarrow{w})^T V \tag{13}$$

in which, the Jacobian matrix $J_v(\overrightarrow{w})$ is

$$J_v(\overrightarrow{w}) = \begin{bmatrix} \frac{\partial V(1)}{\partial w_1} & \cdots & \frac{\partial V(1)}{\partial w_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial V(N)}{\partial w_1} & \cdots & \frac{\partial V(N)}{\partial w_M} \end{bmatrix}, \; V = \begin{bmatrix} V(1) \\ \vdots \\ V(N) \end{bmatrix}. \tag{14}$$

Therefore, the weights update by using LM [19], [21], [22] for an NN controller can be expressed as

$$\triangle \overrightarrow{w} = -[J_v(\overrightarrow{w})^T J_v(\overrightarrow{w}) + \mu I]^{-1} J_v(\overrightarrow{w})^T V. \tag{15}$$

2) Forward Accumulation Through Time (FATT) Algorithm: The calculation of Jacobian matrix $J_v(w)$ needs to pass through the system equations (5). In order to find Jacobian matrix $J_v(\overrightarrow{w})$ efficiently, FATT is developed in this paper for the induction motor, which incorporates the procedures of unrolling the system, calculating the derivatives of the Jacobian matrix, and calculating the DP cost into one single process for each training epoch [20]. Algorithm 1 describes the whole algorithm. Denote $\overrightarrow{\phi_{sdq}}(k) = \sum_{j=1}^{k} \overrightarrow{i_{sdq}}(j)$ and $\frac{\partial \overrightarrow{\phi_{sdq}}(k)}{\partial \overrightarrow{w}} = \sum_{j=1}^{k} \frac{\partial \overrightarrow{i_{sdq}}(j)}{\partial \overrightarrow{w}}$ in the algorithm.

3) Combination of LM + FATT Algorithms: Fig. 3 presents the process of LM algorithm for training an NN controller and also demonstrates how to adjust μ dynamically to ensure that the training follows the decreasing direction of the DP cost function [19] and [17]. The weights update in (15) is handled by Cholesky factorization, which is roughly twice as efficient as the LU decomposition for solving systems of linear equations [23]. The training stops when the training epoch reaches a maximum acceptable value $\mathrm{Epoch}_{\max}$, μ is larger than $\mu_{\max}$, and the gradient is smaller than the predefined minimum acceptable value $\|\partial C_{dp}/\partial \overrightarrow{w}\|_{\min}$.

C. Training Implementation

Table I specifies the parameters of a three-phase induction motor [1], [24], which were used in NN training. Besides the LM + FATT algorithm shown in Section IV-B, we also used the following policies in the training [25].

1) l trajectories were used to train the RNN. l was chosen as 10 in this paper.
2) The initial current state $\overrightarrow{i_{sdq}}(0)$ for each trajectory was generated randomly.
3) The stator dq reference currents $\overrightarrow{i_{sdq\_\mathrm{ref}}}$ for each trajectory were generated randomly and changed every 0.1 s.
4) The initial weights $\overrightarrow{w}$ of the RNN were generated randomly.
5) The sampling time was chosen as $T_s = 0.1$ ms and the duration of each trajectory was set as 1 s.

Fig. 4 shows the learning curve for a successful training of the NN controller. The average DP cost per trajectory drops to a

Algorithm 1 FATT algorithm to calculate the Jacobian matrix and to accumulate DP cost.

1: $C_{DP} \leftarrow 0, \overrightarrow{e_{sdq}}(0) \leftarrow 0, \overrightarrow{s_{sdq}}(0) \leftarrow 0, \dfrac{\partial \overrightarrow{i_{sdq}}(0)}{\partial \vec{w}} \leftarrow 0, \dfrac{\partial \overrightarrow{\phi_{sdq}}(0)}{\partial \vec{w}} \leftarrow 0$

2: {Calculate the Jacobian matrix $J_v(\vec{w})$}

3: **for** $k = 0$ to $N-1$ **do**

4: $\overrightarrow{v_{dq1}}(k) \leftarrow k_{\mathrm{PWM}} R(\overrightarrow{e_{sdq}}(k), \overrightarrow{s_{sdq}}(k), \vec{w})$

5: $\dfrac{\partial \overrightarrow{s_{sdq}}(k)}{\partial \vec{w}} \leftarrow T_s \left[\dfrac{\partial \overrightarrow{\phi_{sdq}}(k)}{\partial \vec{w}} - \dfrac{1}{2} \dfrac{\partial \overrightarrow{i_{sdq}}(k)}{\partial \vec{w}} \right]$

6: $\dfrac{\partial \overrightarrow{v_{dq1}}(k)}{\partial \vec{w}} \leftarrow k_{PWM} \left[\dfrac{\partial R(k)}{\partial \vec{w}} + \dfrac{\partial R(k)}{\partial \overrightarrow{e_{sdq}}(k)} \dfrac{\partial \overrightarrow{i_{sdq}}(k)}{\partial \vec{w}} + \dfrac{\partial R(k)}{\partial \overrightarrow{s_{sdq}}(k)} \dfrac{\partial \overrightarrow{s_{sdq}}(k)}{\partial \vec{w}} \right]$

7: $\dfrac{\partial \overrightarrow{i_{dqS}}(k+1)}{\partial \vec{w}} \leftarrow F \dfrac{\partial \overrightarrow{i_{dqS}}(k)}{\partial \vec{w}} + G \dfrac{\partial \overrightarrow{u_{dqS}}(k+1)}{\partial \vec{w}}$

8: $\dfrac{\partial \overrightarrow{i_{sdq}}(k+1)}{\partial \vec{w}} \leftarrow$ the first two terms of $\dfrac{\partial \overrightarrow{i_{dqS}}(k+1)}{\partial \vec{w}}$

9: $\dfrac{\partial \overrightarrow{\phi_{sdq}}(k+1)}{\partial \vec{w}} \leftarrow \dfrac{\partial \overrightarrow{\phi_{sdq}}(k)}{\partial \vec{w}} + \dfrac{\partial \overrightarrow{i_{sdq}}(k+1)}{\partial \vec{w}}$

10: $\overrightarrow{i_{dqS}}(k+1) \leftarrow F \overrightarrow{i_{dqS}}(k) + G \overrightarrow{u_{dqS}}(k)$

11: $\overrightarrow{e_{sdq}}(k+1) \leftarrow \overrightarrow{i_{sdq}}(k+1) - \overrightarrow{i_{sdq\_\mathrm{ref}}}(k+1)$

12: $\overrightarrow{s_{sdq}}(k+1) \leftarrow \overrightarrow{s_{sdq}}(k) + \dfrac{T_s}{2} [\overrightarrow{e_{sdq}}(k) + \overrightarrow{e_{sdq}}(k+1)]$

13: $C_{DP} \leftarrow C_{DP} + U(\overrightarrow{e_{sdq}}(k+1))$ {accumulate DP cost}

14: $\dfrac{\partial \vec{V}(k+1)}{\partial \vec{w}} \leftarrow \dfrac{\partial \vec{V}(k+1)}{\partial \overrightarrow{e_{sdq}}(k+1)} \dfrac{\partial \overrightarrow{i_{sdq}}(k+1)}{\partial \vec{w}}$

15: the $(k+1)th$ row of $J(\vec{w}) \leftarrow \dfrac{\partial \vec{V}(k+1)}{\partial \vec{w}}$

16: **end for**

17: {On exit, the Jacobian matrix $J_v(\vec{w})$ is finished for the whole trajectory.}

small value very quickly within 100 iterations and is stabilized at this value, demonstrating a good convergence result of the LM training algorithm.

V. Performance Evaluation

To evaluate thetrained NN controller, integrated transient simulation system of a three-phase induction motor drive system was developed by using SimPowerSystems in MATLAB; see Fig. 5.

The PI parameters of both current-loop and speed-loop controllers for the conventional vector control method were tuned by using the the proportional-integral-derivative (PID) tuner function within the PID controller block in MATLAB. Fig. 6 shows the closed-loop simulink model used to tune the current-loop PI parameters. The transfer function in Fig. 6 is $1/(R_s + s\sigma L_s)$

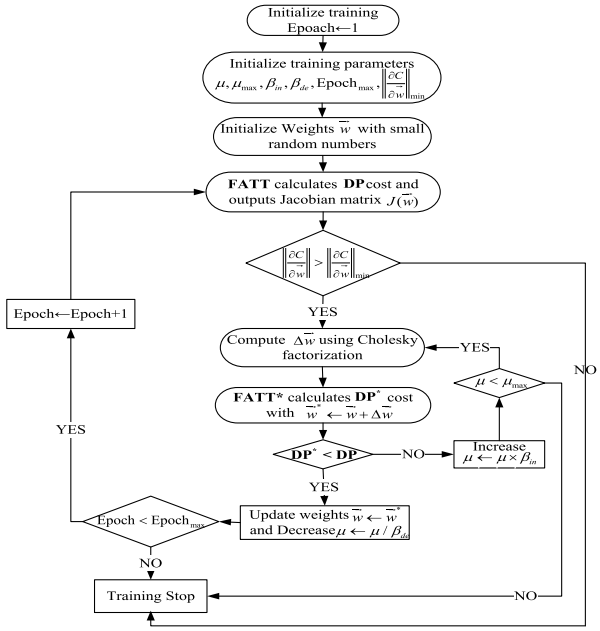

Fig. 3. LM + FATT algorithm for NN controller training. $\mu_{\max}$ stands for maximum μ, β_{de} and β_{in} signify the decreasing and increasing factors, respectively, $\mathrm{Epoch}_{\max}$ represents the maximum number of training epochs, $\|\partial C_{dp}/\partial \vec{w}\|_{\min}$ denotes the norm of the minimum gradient.

TABLE I
"Test" Induction Motor Parameters in Simulation

Symbol	Quantity	Value	Unit
V_g	induction motor rated voltage (rms)	460	V
J_{eq}	inertia	0.025	$\mathrm{kg \cdot m^2}$
p	pole pairs	4	
R_s	per-phase stator resistance	1.77	Ω
R_r	per-phase rotor resistance	1.34	Ω
$L_{\ell s}$	per-phase stator leakage inductance	13.9	mH
$L_{\ell r}$	per-phase rotor leakage inductance	12.1	mH
L_m	per-phase magnetizing inductance	368.7	mH

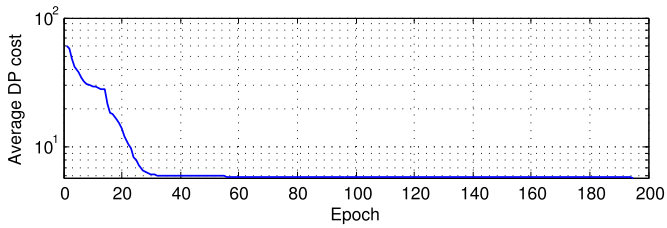

Fig. 4. NN controller learning curve.

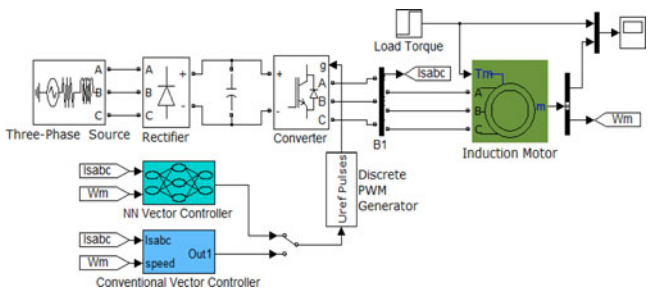

Fig. 5. Simulink model of a three-phase induction motor drive system.

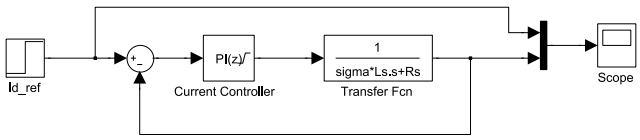

Fig. 6. Block diagram for tuning current-loop PI controller.

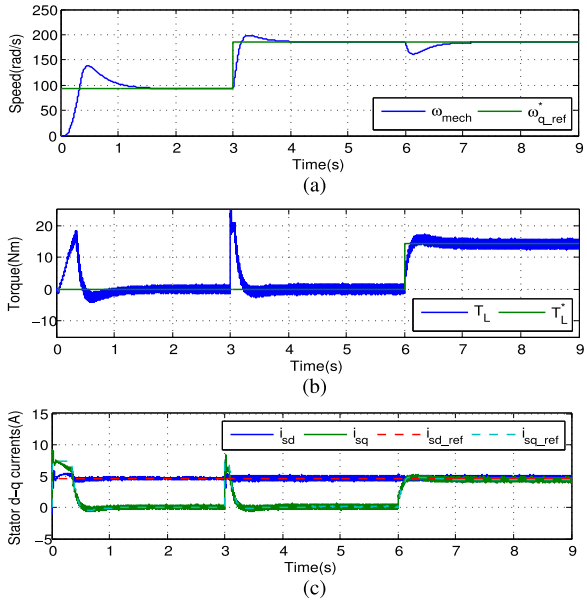

Fig. 7. Conventional vector control. (a) Speed. (b) Torque. (c) Stator dq currents.

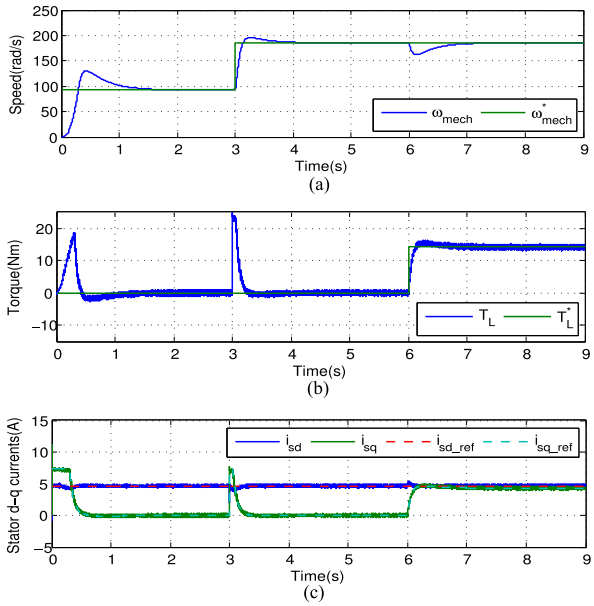

Fig. 8. NN vector control. (a) Speed. (b) Torque. (c) Stator dq currents.

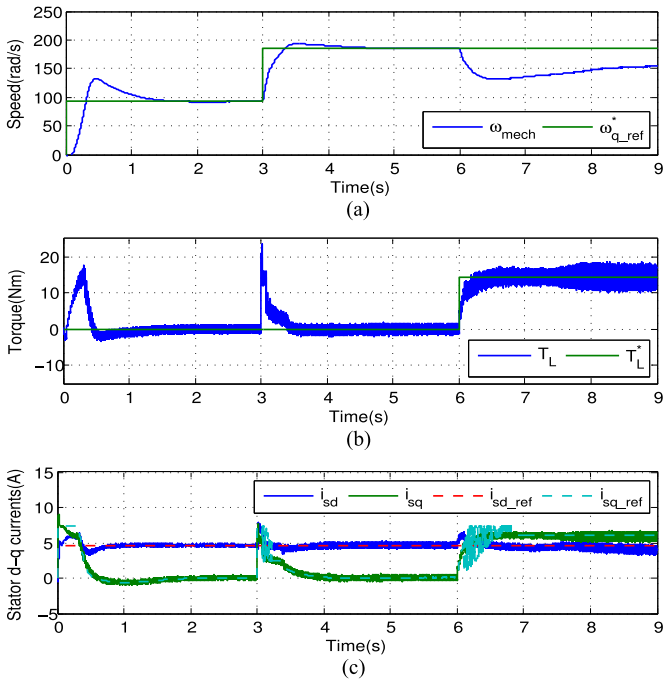

Fig. 9. Conventional vector control with new setting: $R_r^* = 3R_r$. (a) Speed. (b) Torque. (c) Stator dq currents.

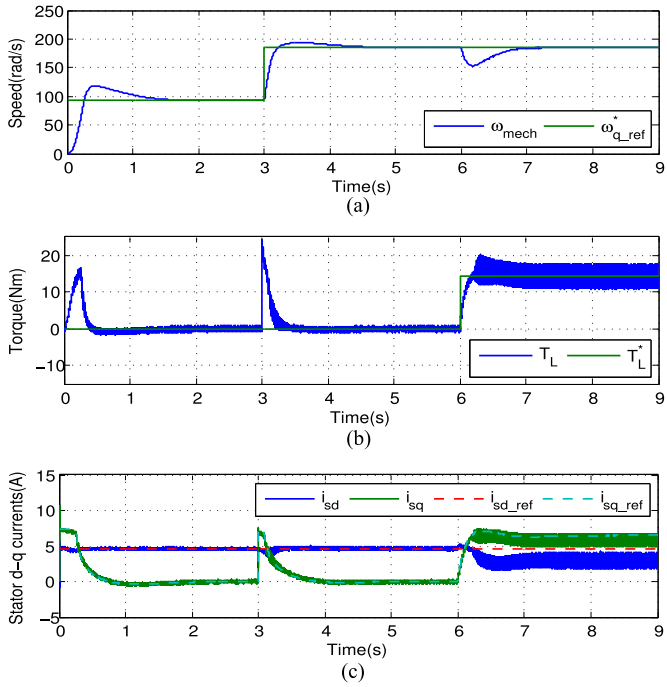

Fig. 10. NN vector control with new setting: $R_r^* = 3R_r$. (a) Speed. (b) Torque. (c) Stator dq currents.

corresponding to (6) and (7). For the current-loop controller, phase margin was set as 60° and the bandwidth was chosen as 2000 rad/s, which tends to yield the best results considering the PWM saturation constraints. For the speed-loop controller, the corresponding transfer function is $1/(K_T s J_{eq})$ corresponding to (8). As fast response is not required for the speed-loop controller, the crossover frequency was set as 20 rad/s and phase margins was also set as 60°.

If not specified, $T_s = 0.1$ ms was used in all simulation models of Section V and the switching frequency was chosen as $f_s = 6000$ Hz.

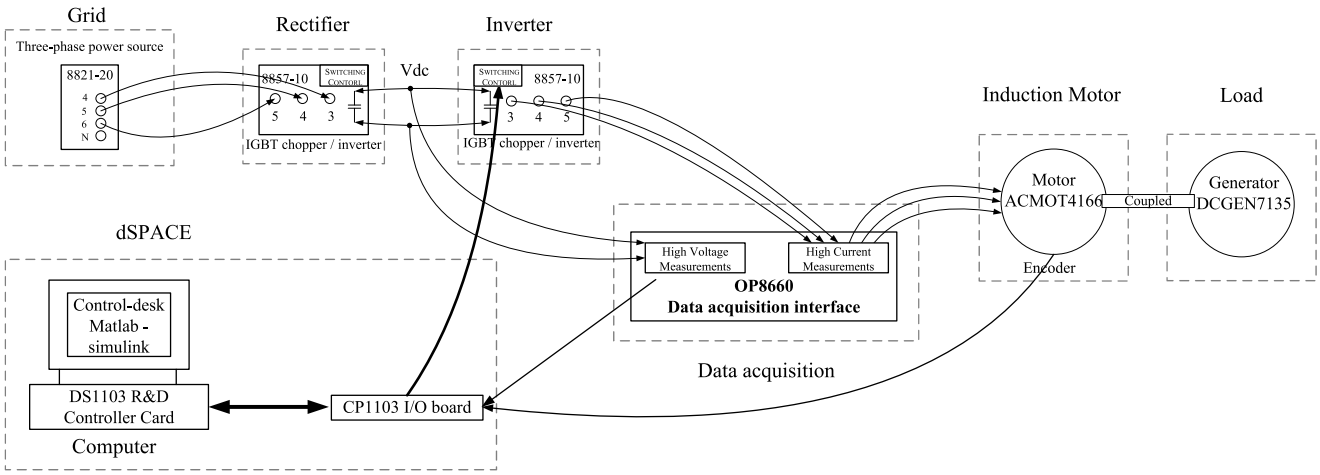

Fig. 11. Schematic of the induction motor drive experiment.

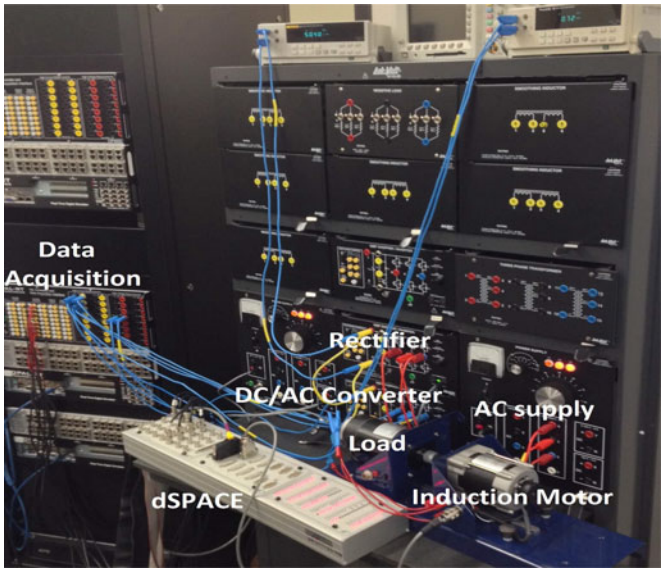

Fig. 12. Hardware laboratory testing and control systems.

TABLE II
INDUCTION MOTOR (ACMOT4166) PARAMETERS

Symbol	Quantity	Value	Unit
V_g	rated voltage (rms)	30	V
$J_{\rm eq}$	inertia	0.000225	$\rm kg \cdot m^2$
p	pole pairs	4	
$P_{\rm rated}$	rated power	120	W
$n_{\rm max}$	max speed	4000	rpm
$i_{\rm raeted}$	rated current	6	A
R_s	per-phase stator resistance	0.896	Ω
R_r	per-phase rotor resistance	1.82	Ω
$L_{\ell s}$	per-phase stator leakage inductance	1.94	mH
$L_{\ell r}$	per-phase rotor leakage inductance	2.45	mH
L_m	per-phase magnetizing inductance	46.2	mH

A. Tracking Reference

Fig. 7 shows the tracking reference performance under the conventional vector control and Fig. 8 gives the corresponding performance under the novel NN vector control.

During the test, no load torque was applied at the beginning and certain load torque was added at time $t = 6$ s [see Figs. 7(b) and 8(b)]. For speed reference tracking, although the same speed PI controller was used for both NN vector control [see Fig. 7(a)] and conventional vector control [see Fig. 8(a)], the NN controller still responses a little bit faster than the conventional controller. For stator dq current comparison [see Figs. 7(c) and 8(c)] and torque comparison [see Figs. 7(b) and 8(b)], the NN vector control provides less current oscillations than the conventional vector control.

B. Detuning Effects

For vector control of the induction motor (e.g., Fig. 2), an estimated induction model is used to estimate the rotor position. However, in practice, the estimated motor parameters may deviate a lot from its nominal values. This is particular true for the rotor time constant $\tau_r = L_r/R_r$, which mainly depends on rotor resistance that increases significantly as the rotor heats up. The variation of rotor resistance may be up to 100% and can hardly be recovered using thermal models and temperature sensors [26], [27]. This incorrect estimation will possibly cause a steady-state error using conventional vector control [1]. Thus, an examination was performed to evaluate the NN vector control under incorrect parameter estimation condition. In the examination, we chose an extreme condition and the rotor resistance was set as $R_r^* = 3R_r$ with all the other parameters kept unchanged. Figs. 9 and 10 show the examination results of the conventional vector control and the NN vector control, respectively.

The performance results demonstrated great advantages of the NN vector control over the conventional vector control. When certain load torque was applied to the induction motor, there was an obvious steady-state error in tracking the speed reference under conventional vector control [see Fig. 9(a)], while the NN vector control still performed very well [see Fig. 10(a)]. Further, the NN vector control [see Fig. 10(b)] showed lees torque oscillations than the conventional vector control [see Fig. 9(b)]. The stator dq current waveform revealed the reason why NN vector control can still perform well. The NN vector control can

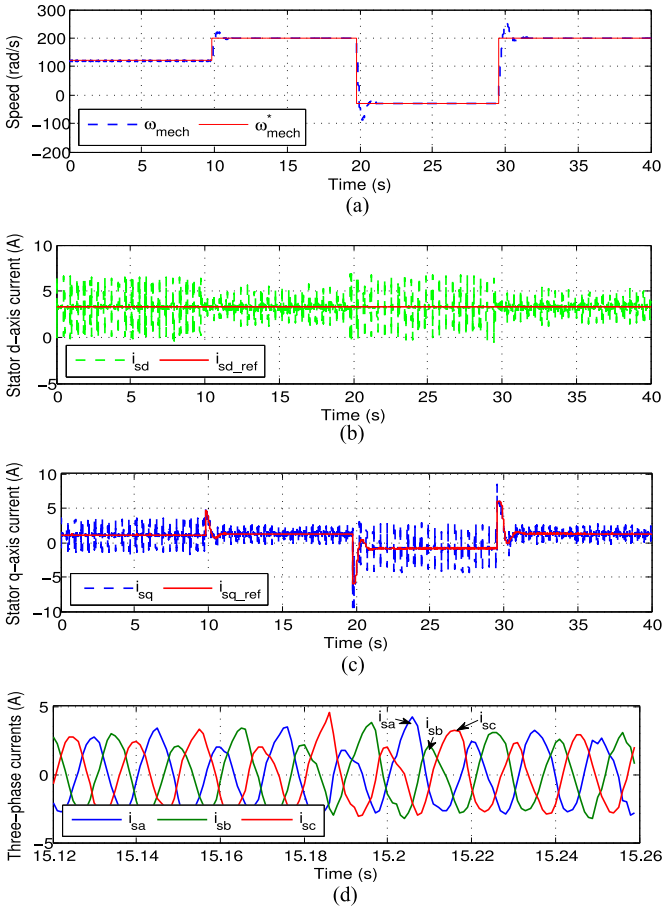

Fig. 13. Conventional vector control with $T_s = 0.5$ ms and $f_s = 6$ kHz. (a) Speed. (b) Stator d-axis current. (c) Stator q-axis current. (d) Three-phase currents.

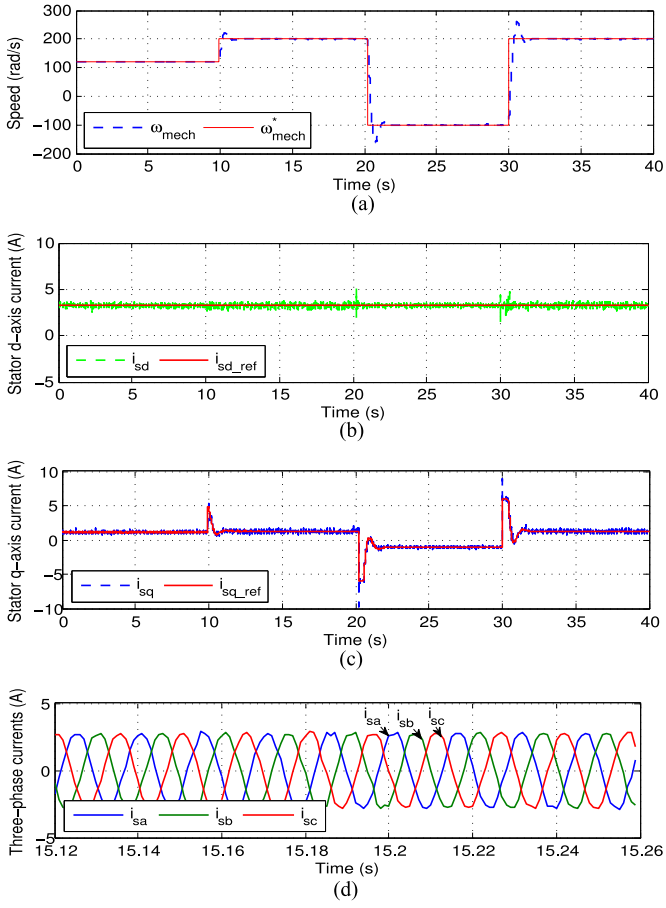

Fig. 14. NN vector control with $T_s = 0.5$ ms and $f_s = 6$ kHz. (a) Speed. (b) Stator d-axis current. (c) Stator q-axis current. (d) Three-phase currents.

better overcome the competing control problem and achieve the optimal d- and q-axis current control to meet the motor drive needs even under large motor parameter deviation conditions. This examination demonstrated an important feature of the proposed NN vector control that it can tolerate a wide range of system parameter change. As the neural network has adaptive and leaning abilities, thus it can provide better performance when facing uncertainties [16], e.g., rotor resistance changes. In addition, the training of the proposed NN is designed to approximate optimal control, which further tends to yield better performance when facing system parameter changes. The robustness of the NN controller is also demonstrated in other vector control applications [25].

VI. Hardware Experiment Validation

A. Experiment Setup

To further validate the proposed NN vector control, induction motor drive experiments were conducted based on dSPACE. Fig. 11 illustrates the experiment components and wire connections in the induction motor control experiments. Fig. 12 shows the hardware testing and control systems. Table II specifies all the parameters of the induction motor (ACMOT4166) [28]. The hardware test system took the following setups.

1) An LabVolt three power supply signifies the power source.
2) An ac/dc rectifier is connected with the power source.
3) Another dc/ac inverter is controlled by a dSPACE digital control system.
4) OP8660 collects the dc-link voltage and three-phase currents of the induction motor.
5) CP1103 I/O board connects the current and voltage measurements from OP8660, and the rotor position from the encoder and sends out control signals to the converter according to various control demands.

B. Experiment Results

Both conventional vector control and NN vector control used the same speed-loop PI controller with crossover frequency selected as 10 rad/s. The crossover frequency of current-loop PI controller was selected as 2000 rad/s for the conventional vector control, which tends to yield the best transient performance considering the PWM saturation constraints. The NN controller was retrained based on the induction motor parameters in Table II.

When the sampling time $T_s = 0.5$ ms and the switching frequency $f_s = 6$ kHz were chosen, Fig. 13 shows the experiment results of induction motor under conventional vector control, while Fig. 14 shows the results under the NN vector control. As both conventional vector control and NN vector control utilized the same speed PI controller, no noticeable differences can be observed in Figs. 13(a) and 14(a) between these two vector

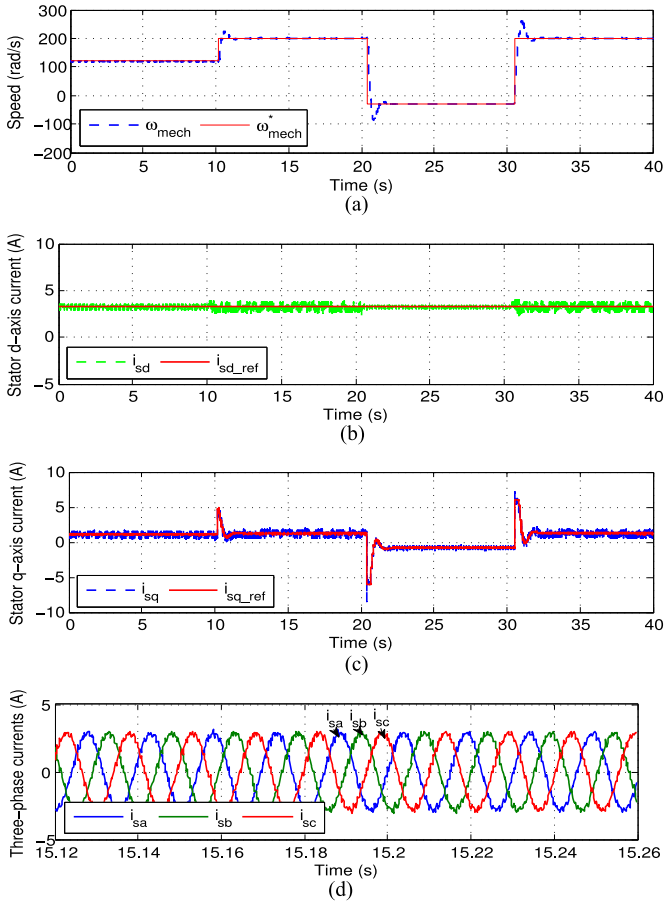

Fig. 15. Conventional vector control with $T_s = 0.1$ ms and $f_s = 6$ kHz. (a) Speed. (b) Stator d-axis current. (c) Stator q-axis current. (d) Three-phase currents.

control methods. However, big differences can be seen from current waveforms. No matter from d-current or q-current, the NN vector control demonstrated much less oscillations [see Fig. 14(b) and (c)] than the conventional vector control [see Fig. 13(b) and (c)]. Three-phase current waveforms [see Figs. 13(d) and 14(d)] furthermore revealed the fact. This kind of oscillations [see Fig. 13(d)] is not good for induction motors and would reduce the lifetime of induction motor. Large noise caused by this oscillation could be heard in the laboratory under the conventional vector control.

Similar tests were done with different sampling time and switching frequency settings. Fig. 15 shows the control performance of conventional vector control when using $T_s = 0.1$ ms and $f_s = 6$ kHz. Under this setting, the high current oscillation and large noise were gone. However, even using higher sampling rate, the conventional vector control still showed relatively larger current oscillations [see Fig. 15(b)–(d)] compared with the NN vector control (see Fig. 14).

Table III gives a summary of comparison results, which demonstrated that the NN vector control can achieve very good performance under relatively low switching frequency and low sampling rate ($1/T_s$). The results are consistent with that reported in [29] when the switching frequency is increased up to 6 kHz; the audile noise can be reduced significantly under conventional vector control. The benefits of relatively low switching frequency or low sampling rate would decrease the power loss, improve efficiency, and reduce size and cost of the motor drive system. The reason behind this improvement is the better current control ability of the NN vector control than conventional vector control. The proposed NN vector control utilizes the exact state-space model of the induction motor and thus avoids inaccurate description of the induction motor. Also, the training of the NN controller is to approximate optimal control, which improves the current control performance and makes the proposed NN vector controller less sensitive to the harmonic current distortion.

TABLE III
HARDWARE EXPERIMENT COMPARISONS

Measure \ Method	Conventional vector control	NN vector control
$T_s = 0.5$ ms and $f_s = 6$ kHz	bad	good
$T_s = 0.1$ ms and $f_s = 4$ kHz	bad	good
$T_s = 0.1$ ms and $f_s = 6$ kHz	good	good

VII. CONCLUSION

The proposed NN vector control mainly focuses on improvement of current-loop control performance for an induction motor drive. The substitution of PI controllers in the current loop with the proposed NN controller has brought great advantages such as small oscillation, strong tracking ability. Most importantly, the NN vector control can better overcome the problem of competing control and detuning effects properly. Hardware tests showed that the NN vector control can succeed in driving an induction motor using relatively lower switching frequency or lower sampling rate compared with conventional vector control, which would benefit the induction motor drives in multiple aspects.

REFERENCES

[1] N. Mohan, *Advanced Electrical Drives: Analysis, Control And Modeling Using Simulink*. Minneapolis, MN, USA: MNPERE, 2001.
[2] S. Li and T. A. Haskew, "Analysis of decoupled d-q vector control in DFIG back-to-back PWM converter," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Tampa, FL, USA, Jun. 2007, pp. 1–7.
[3] K. Peng and J. Zhao, "Speed control of induction motor using neural network sliding mode controller," in *Proc. Int. Conf. Electr. Inf. Control Eng.*, Wuhan, China, Apr. 2011, pp. 6125–6129.
[4] L. Ben-Brahim and R. Kurosawa, "Identification of induction motor speed using neural networks," in *Proc. Power Convers. Conf.*, Yokohama, Japan, Apr. 1993, pp. 689–694.
[5] J. Restrepo, J. Viola, R. G. Harley, and T. G. Habetler, "Induction machine current loop neurocontroller employing a Lyapunov based training algorithm," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Tampa, FL, USA, Apr. 2003, pp. 1–8.
[6] J. Restrepo, B. Burton, R. G. Harley, and T. G. Habetler, "ANN based current control of a VSI fed AC machine using line coordinates," in *Proc. 5th IEEE Int. Caracas Conf. Devices Circuits Syst.*, Nov. 2004, pp. 225–229.
[7] R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.
[8] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *J. Guidance, Control, Dyn.*, vol. 19, no. 4, pp. 893–898, Jul./Aug. 1996.

[9] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.
[10] F. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag*, vol. 4, no. 2, pp. 39–47, May 2009.
[11] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 764–773, May 2002.
[12] S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Vector control of a grid-connected rectifier/inverter using an artificial neural network," in *Proc. IEEE World Congr. Comput. Intell.*, Brisbane, Australia, Jun. 2012, pp. 1–7.
[13] S. Li, M. Fairbank, C. Johnson, D. C. Wunsch, E. Alonso, and J. L. Proaño, "Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions," *IEEE Trans. Neural Netw. Learning Syst*, vol. 25, no. 4, pp. 738–750, Apr. 2014.
[14] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Reading, MA, USA: Addison-Wesley, 1998.
[15] J. Dannehl, C. Wessels, and F. W. Fuchs, "Limitations of voltage-oriented pi current control of grid-connected PWM rectifiers with LCL filters," *IEEE Trans. Ind. Electron.*, vol. 56, no. 2, pp. 380–388, Feb. 2009.
[16] J. Sarangapani, *Neural Newtwork Control of Nonlinear Discrete-Time Systems*. New York, NY, USA: Taylor & Francis, 2006.
[17] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*. Boston, MA, USA: PWS-Kent, 2002.
[18] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*. 3rd ed., Hoboken, NJ: Wiley, 2002.
[19] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.
[20] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg-Marquardt algorithm for optimal control of a grid-connected converter," *IEEE Trans. Neural Netw. Learn. Syst*, to be published.
[21] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. J. Appl. Math.*, vol. II, no. 2, pp. 164–168, 1994.
[22] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.
[23] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[24] Baldor electric company. [Online]. Available: http://www.baldor.com/
[25] X. Fu, S. Li, and I. Jaithwa, "Implement optimal vector control for LCL-filter-based grid-connected converters by using recurrent neural networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4443–4454, Jul. 2015.
[26] R. Marino, S. Peresada, and P. Tomei, "On-line stator and rotor resistance estimation for induction motors," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 570–579, May 2009.
[27] G. Kenné, R. S. Simo, F. Lamnabhi-Lagarrigue, A. Arzandé, and J. C. Vannier, "An online simplified rotor resistance estimator for induction motors," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 5, pp. 1188–1194, Sep. 2010.
[28] Induction motor (ACMOT4166) data sheet form Motorsolver LLC. [Online]. Available: http://motorsolver.com/wp/wp-content/uploads/2015/03/4-DYNO-IM-SPECS-small.pdf
[29] A. Malfait, R. Reekmans, and R. Belmans, "Audible noise and losses in variable speed induction motor drives with IGBT inverters-influence of the squirrel cage design and the switching frequency," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Denver, CO, USA, Oct. 1994, pp. 693–700.

Xingang Fu (S'14) received the B.S. and M.S. degrees in applied mathematics from the Ocean University of China, Qingdao, China, in 2005 and 2009, respectively. He is currently working toward the Ph.D. degree with the Electrical and Computer Engineering Department, University of Alabama, Tuscaloosa, AL, USA.

His main research interests include neural network control, machine learning, cloud computing, data mining, power systems, electric machines, and drives.

Shuhui Li (S'99–M'99–SM'08) received the B.S. and M.S. degrees in electrical engineering from Southwest Jiaotong University, Chengdu, China, in 1983 and 1988, respectively, and the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA, in 1999.

He was with the School of Electrical Engineering, Southwest Jiaotong University, from 1988 to 1995, where his fields of research interest included electrified railways, power electronics, power systems, and power system harmonics. From 1995 to 1999, he was involved in research on wind power, artificial neural networks, and applications of massive parallel processing. He joined Texas A&M University, Kingsville, TX, as an Assistant Professor in 1999 and an Associate Professor in 2003. He joined the University of Alabama, Tuscaloosa, AL, USA, as an Associate Professor in 2006. His current research interests include renewable energy systems, power electronics, power systems, electric machines and drives, and applications of artificial neural networks in energy systems.