Implementation of Speed Sensorless Induction Motor drives by Fast Learning Neural Network using RLS Approach

Yoon-Ho Kim, Member, IEEE   Yoon-Sang Kook

Dept. of Electrical Engineering, Chung-Ang University
221 HukSuk-Dong, DongJak-Ku, Seoul, 156-756 Korea
Phone : +82-2-820-5290   Fax : +82-2-812-1407

Keywords : RLS(Recursive Least Square), NN(Neural Network)

ABSTRACT - This paper presents a newly developed speed sensorless drive using RLS based on Neural Network Training Algorithm. The proposed algorithm has just the time-varying learning rate, while the well-known back-propagation algorithm based on gradient descent has a constant learning rate. The number of iterations required by the new algorithm to converge is less than that of the back-propagation algorithm. The theoretical analysis and experimental results to verify the effectiveness of the proposed control strategy are described.

Ⅰ. INTRODUCTION

The Recursive Least Squares is well known as a state estimation method for a nonlinear system, and can be used as a parameter estimation method by augmenting the state which unknown parameters. A multi-layered neural network is a nonlinear system having a layered structure, and its learning algorithm is regarded as parameter estimation for such a non-linear system[1]. In this paper, a new real-time learning algorithm for a multi-layered neural network is derived from the RLS. This method minimizes the global sum of the squared errors between the actual and the desired output values iteratively. The weights in the network are updated upon the arrival of a new training sample and by solving a system of normal equations recursively. Since this RLS-based learning algorithm approximately gives the minimum covariance estimate of the weights, the convergence performance is improved in comparison with the backward error propagation algorithm using the steepest descent techniques[2]. The proposed algorithm, though computationally complex has an adaptive varying learning rate, while the back-propagation algorithm has constant learning rate. The proposed learning algorithm usually converges in a few iterations and the error is comparable to that of the well-known back-propagation algorithm.

Ⅱ. FLUX ESTIMATOR WITH FILTER CONCEPT

Induction motor rotor fluxes are selected to represent the desired and estimated state variable. The following two independent estimators, in the stationary frame, are generally used to derive these rotor fluxes.

A. Current Model of Rotor Circuit

The rotor flux estimator can be formed if the stator current and the rotor speed are measured in real time. It can be represented as follows.

$$\dot{\hat{\lambda}}^s_{dqr\_cm} = \left(-\frac{1}{\tau_r}I + \omega_r J\right)\hat{\lambda}^s_{dqr\_cm} + \frac{L_m}{\tau_r}i^s_s \qquad (1)$$

Where, $\tau_r = L_r / R_r$, $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$,

$i_s = [i_{ds} \quad i_{qs}]^T$, $\lambda_r = [\lambda_{dr} \quad \lambda_{qr}]^T$.

$v_s = [v_{ds} \quad v_{qs}]^T$.

B. Voltage Model of Stator Circuit

The voltage model utilizes the stator voltages and currents, but not the rotor velocity. It is commonly used to implement direct field orientation without speed sensors for low cost drive applications. The rotor fluxes in the stationary d-q reference frame can be obtained,

$$\dot{\hat{\lambda}}^s_{dqr\_vm} = \frac{L_r}{L_m}\{(v^s_s - R_s i^s_s) - \sigma L_s \dot{i}_s\} \qquad (2)$$

Where, $\sigma = 1 - \frac{L_m^2}{L_s L_r}$

C. Rotor Flux Estimator Using Filter Concept

It is well known that the current model is heavily dependent on the parameter accuracy. Similarly, though the voltage model has less sensitivity on the parameter accuracy, the low speed sensitivity is a well acknowledged limitation of this observer due to the stator resistance and the offset problem. Thus, by utilizing the current model in the low speed range and voltage model in the high-speed range, more accurate rotor flux can be obtained in wide speed range. In this paper, the filter concept is used to utilize the current model in low frequency region and the voltage model in high frequency region. The resultant rotor flux is obtained from the low pass filtered current model rotor flux and the high pass filtered voltage model rotor flux. The resultant rotor flux observer is written as:

$$\hat{\lambda}^s{}_{dqr} = [HPF]\hat{\lambda}^s{}_{dqr\_vm} + [LPF]\hat{\lambda}^s{}_{dqr\_cm} \tag{3}$$

Where, $\hat{\lambda}^s{}_{dqr\_vm}$ denotes the voltage model rotor fluxes, $\hat{\lambda}^s{}_{dqr\_cm}$ denotes the current model rotor fluxes and $[HPF]$, $[LPF]$ denote the high pass filtering operation and the low pass filtering operation respectively. From (3), a flux angle can be detected, which enables direct field oriented control.

The filter can be designed in arbitrary order. For example, second-order filter is used, then

$$[HPF] = \frac{s^2}{s^2 + K_p s + K} \tag{4}$$

$$[LPF] = \frac{K_p s + K_i}{s^2 + K_p s + K_i} \tag{5}$$

The coefficients in (4) and (5) can also be determined by the filter concept. In case of Butterworth filter, the coefficients are related as

$$K_p = \sqrt{2}\omega_c,\ K_i = \omega_c^2 \tag{6}$$

where denotes the cut-off frequency of the filter. Note that this cut-off frequency is the transition point from current model to voltage model. This flux observer has less parameter dependency in high speed region and has higher immunity to noise and measurement error in low speed region.

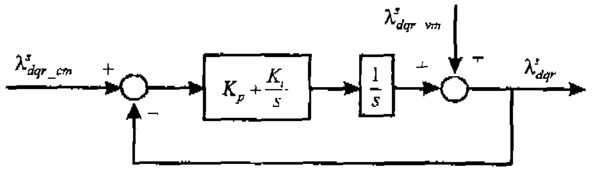

Fig. 1 Flux estimator with both voltage model and current model.

D. Estimator eigen value selection

First of all, to select the eigen values of the estimator, that of the controlled system should be predetermined from (1).

$$\det(sI - A) = \det\begin{bmatrix} s + \frac{1}{\tau_r} & \omega_r \\ -\omega_r & s + \frac{1}{\tau_r} \end{bmatrix} = \left(s + \frac{1}{\tau_r}\right)^2 + \omega_r^2 \tag{7}$$

$$\therefore S_1, S_2 = -\frac{1}{\tau_r} \pm j\omega_r \tag{8}$$

If a high eigen values are selected for estimator, the convergence rate is increased but the estimator stability will be diminished. In order to estimate stable flux quantities, the value of the estimator should be selected according to that of the system. In this paper, variable estimator eigen value is applied to satisfy the system requirements: i.e. rapid response and stable estimation. Thus, in this paper, the characteristic root presented by ω_c is determined as a function of the machine speed.

$$\omega_c = f(\omega_r) = \frac{1}{\tau_r}\left(\frac{15}{\omega_{r\_\max}}|\hat{\omega}_r| + 5\right) \tag{9}$$

Equation (9) shows that the estimated speed is used in calculating the cut-off frequency of the filter. In Fig. 2, the eigen valuc determination algorithm is illustrated.

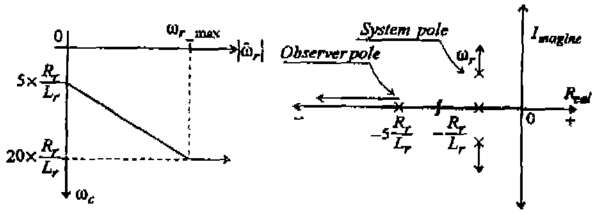

Fig. 2 Eigen value selection of the Flux Estimator.

Ⅲ. THE PROPOSED SPEED SENSORLESS CONTROL ALGORITHM USING RLS

The back-propagation algorithm can be summarized as follows[2]

$$w_{ji}^{k-1,k}(t+1) = w_{ji}^{k-1,k}(t) + \Delta w_{ji}^{k-1,k}(t) \tag{10}$$

where,

$$\Delta w_{ji}^{k-1,k}(t) = \eta\delta_j^k o_i^{k-1} + \alpha\Delta w_{ji}^{k-1,k}(t-1)$$

$$\delta_j = (t_j - o_j)f'(i_j^k) \quad \text{for the output layer}$$

$$\delta_j = f'(i_j^k)\sum_k \delta_k w_{kj} \quad \text{for the hidden layer}$$

The back-propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a feed-forward net and the desired output.

A. Learning algorithm via the Recursive Least Squares

We have reviewed how the back-propagation algorithm essentially implements gradient descent in sum-squared error. It should be noted, however, that the learning rate is constant, so we may have to consume more time to obtain a sufficiently convergent results, even though we can take into account a momentum term. Our main theoretical contribution here is to show that there is an efficient way of computing a time-varying learning rate.

Our learning strategy is based on regarding the learning of a network as an estimation (or identification) problem of constant parameters.

The output layer of the multi-layered neural network is expressed by the following models with nonlinear observation equations :

$$\hat{y}_j(t+1) = h(\hat{w}_{ji}(t+1)) = o_j^M(t+1) \qquad (11)$$

The recursive least squares method partitions the layers of an NN into a linear set of input-output equations and applies the common RLS algorithm to update the weights in each layer. The application of the RLS algorithm for a weight matrix update gives the following real-time learning algorithms :

$$\hat{w}_{ji}(t+1) = \hat{w}_{ji}(t) + K_{ji}(t)\left[y_j(t) - o_j^M(t)\right] \qquad (12)$$

$$K_{ji}(t) = \frac{P_{ji}(t+1\,|\,t)\Phi_{ji}(t)}{\left[\lambda I + \Phi^T{}_{ji}(t)P_{ji}(t+1\,|\,t)\Phi_{ji}(t)\right]} \qquad (13)$$

$$P_{ji}(t+1\,|\,t+1) = \lambda^{-1}\left[I - K_{ji}(t)\Phi^T{}_{ji}(t)\right]P_{ji}(t+1\,|\,t) \qquad (14)$$

where λ $(0 < \lambda \le 1)$ is the forgetting factor, $K_{ji}(t)$ is the gain matrix, $P_{ji}(t+1\,|\,t+1)$ is covariance matrix, $\Phi_{ji}(t)$ is the input to the layer, $y_j(t)$ is the desired output.

B. Speed sensorless control strategy

Two independent observers are used to estimate the rotor flux vectors: one based on (1) and the other based on (2). Since (1) does not involve the speed ω_r, this observer generates the desired value of rotor flux, and (2) which does involve ω_r may be regarded as a neural model with adjustable weights. The error between the desired rotor flux $\lambda^s{}_{dqr\_vm}$ given by (1) and the rotor flux $\lambda^s{}_{dqr\_cm}$ provided by the neural model (2) is used to adjust the weights, in other words the rotor speed ω_r.

The rotor speed can be derived using the RLS based on NN. The overall block diagram of speed sensorless control is shown in Fig. 3.

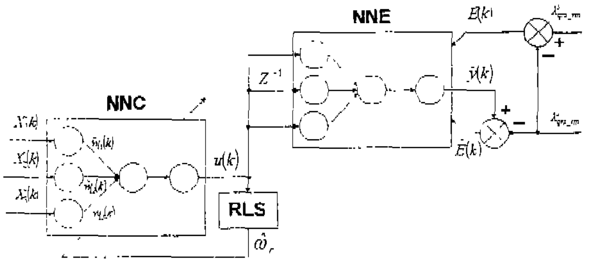

Fig. 3.Structure of RLS based on NN for $\hat{\omega}_r$ estimation.

The discrete state equation model of (2) can be rewritten as follows

$$\hat{y}(k) = \Phi^T(k)\hat{\theta}(k) \qquad (15)$$

where $\Phi^T(k) = \left[\hat{\lambda}^s{}_{qr\_cm}(k) \quad \hat{\lambda}^s{}_{dr\_cm}(k) \quad i^s{}_{qs}(k)\right]$

$$= \left[X_1(k) \quad X_2(k) \quad X_3(k)\right],$$

$$\hat{\theta}(k) = \left[1 - 1/\hat{\tau}_r \cdot T_s \quad \hat{\omega}_r \cdot T_s \quad L_m/\hat{\tau}_r \cdot T_s\right]^T$$

$$= \left[\hat{w}_{11}(k) \quad \hat{w}_{12}(k) \quad \hat{w}_{13}(k)\right]^T.$$

The new weight, $\hat{w}_{12}(k)$ is therefore given by

$$\hat{w}_{12}(k+1) = \hat{w}_{12}(k) + K_{ji}(k)\left[y_j(k) - \hat{y}_j(k)\right] \qquad (16)$$

where. $y_j(k) = \lambda^s{}_{qr\_vm}(k)$.

The estimated rotor speed $\omega_r(k)$ applied by RLS based on NN is computed as follows

$$\hat{\omega}_r(k+1) = \hat{\omega}_r(k) + K_{ji}(k)\left[y_j(k) - \Phi^T(k)\hat{\theta}(k)\right]/T_s \qquad (17)$$

where λ can be used to improve the characteristics of the transient response as follows :

$$\lambda(k) = \lambda_0\lambda(k-1) + (1-\lambda_0)$$

$$\lambda_0 = 0.98, \quad P_{ji}(0\,|\,0) = 500I$$

Ⅳ. SIMULATION RESULTS

A 22kW 4-pole IM is used for the simulation and experiment simultaneously. The proposed sensorless control of IM is shown in Fig. 5. The nominal parameters used for the simulations are given Table 1 as follows :

Table 1. Induction Motor Parameters

Rated Power	22kW	L_s	43.75mH
Rated Speed	2000rpm	L_r	44.09mH
Rated Torque	120Nm	L_m	42.1mH
R_s	0.115	J_M	0.1618kgm^2
R_r	0.0821	P	4

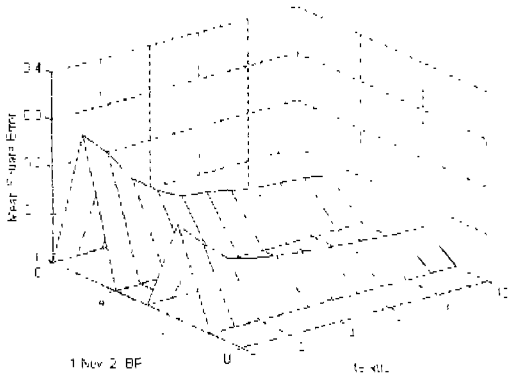

Fig. 4 Comparison of the mean-squared error versus the iteration number for New and BP algorithm

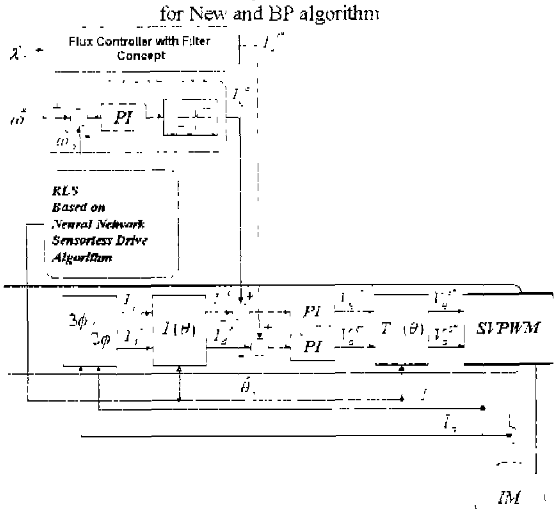
Flux Controller with Filter Concept
PI
RLS Based on Neural Network Sensorless Drive Algorithm
3φ/2φ
PI
PI
SVPWM
IM

Fig. 5 The block diagram of the overall control algorithm

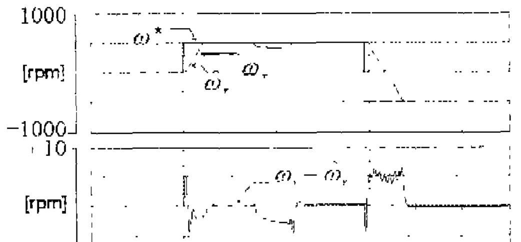

Fig. 6 The characteristics of speed step response (500[rpm]→-500[rpm], 0.5p.u. load).

The step response of the proposed sensorless algorithm is shown in Fig. 6 when the speed reference is changed from 500[rpm] to -500[rpm]. As shown in Fig. 6, we can know that the speed error is limited by 0.5% of the rating speed. Also, The proposed learning algorithm usually converges in a few iterations and the error is comparable to that of the well-known back-propagation algorithm.

V. EXPERIMENTAL RESULTS

For the high performance IM drives, the overall IM drive system in Fig. 7 is implemented with a TMS320C31 DSP control board and a PWM IGBT inverter.

For actual load emulation, the DC generator is coupled to the IM. Actual rotor angle and machine speed are measured from an incremental encoder with 4096[ppr] resolution for monitoring. The sampling time of current controller loop is 250[μs] and that of the outer voltage regulating loop and speed loop is 2.5[ms]. The control algorithm including the proposed scheme was fully implemented with the software.

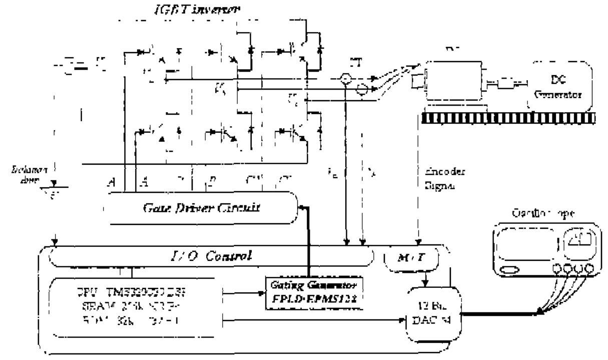

Fig. 7 The overall IM drive system.

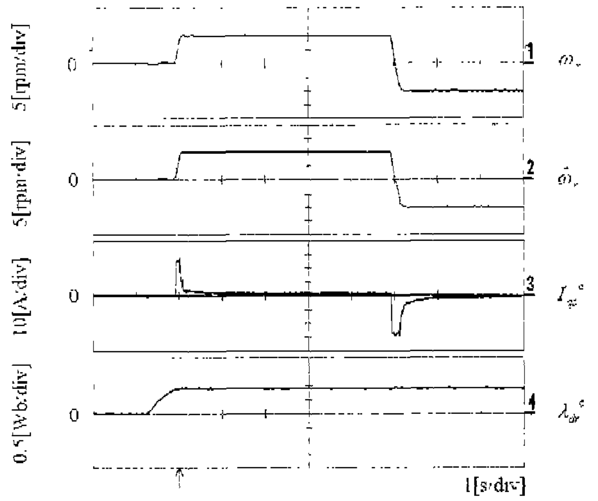

Fig. 8 The experimental waveforms of step response (+10[rpm]→-10[rpm], TL=0[p.u]).

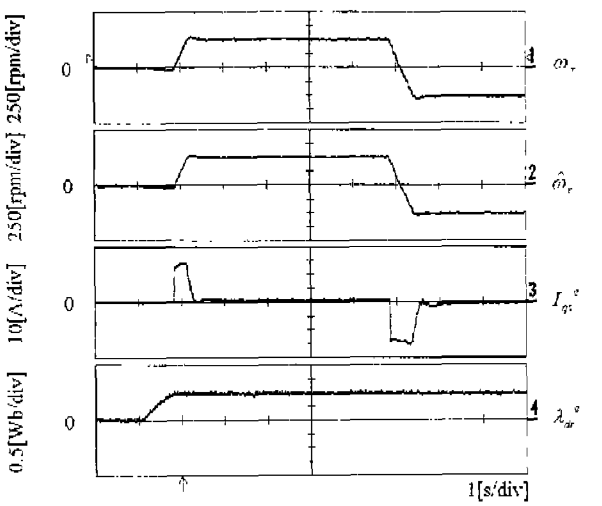

Fig. 9 The experimental waveforms of step response (-500[rpm]->-500[rpm], TL=0[p.u]).

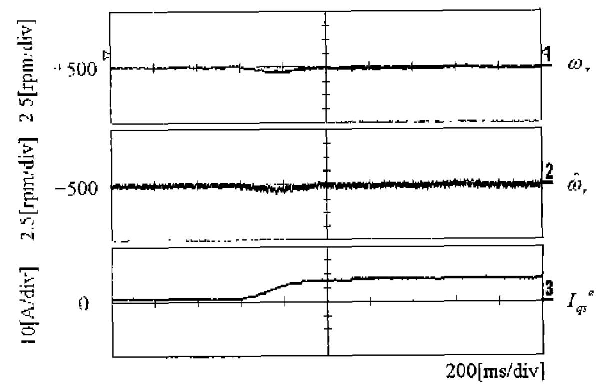

Fig. 10 The experimental waveforms of load step response (TL : 0[p.u] -> 0.5[p.u]).

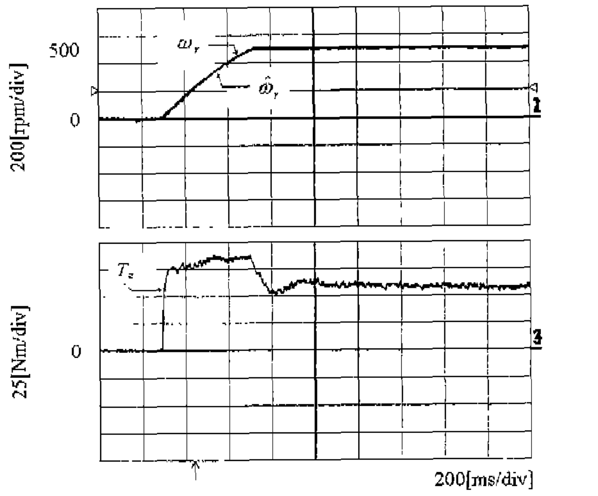

Fig. 11 The experimental waveforms of step response (0[rpm]->+500[rpm], TL=1.0[p.u]).

Experiments are conducted to evaluate the performance of the new speed sensor elimination algorithm based on the NN. The step response of the proposed sensorless algorithm is shown from fig. 8 to fig. 9 when the speed reference is changed with no load torque. It shows that the estimated speed is tracking the real one with good accuracy. Fig. 10 and fig. 11 show the characteristics of load torque response. As shown in those figure, the proposed algorithm works well in spite of the load torque variation.

VI. CONCLUSION

We have studied learning algorithm for multi-layered feed-forward type neural networks and proposed a new back-propagation algorithm that uses Recursive Least Squares algorithm to identify the connection weights of the network. This algorithm has the feature that the learning rate is time dependent, whereas the algorithm of conventional back-propagation has a constant learning rate. Using some simulation examples, we have made two points:

(a) when a sufficiently convergent solution is desired, the present method assures faster learning than the generalized delta rule.

(b) the proposed method works well even though the initial weights are relatively large, whereas the generalized delta rule is convergent only in the case where the initial weights are relatively small.

VII. REFERENCES

[1] Youji Iiguni, Hideaki Sakai and Hidekatsu Tokumaru, "A Real-Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter," IEEE, Trans. On Signal Processing, vol. 40, No. 4, pp. 959-966, Apr. 1992.

[2] Keigo Watanabe, Toshio Fukuda and Spyros G. Tzafestas, "Learning Algorithms of Layered Neural Networks via Extended Kalman Filters," *Int. J. Systems Sci.*, vol. 22, No. 4, pp. 753-768, 1991.

[3]Robert S. Scalero and Nazif Tepedelenlioglu, "A Fast New Algorithm for Training Feedforward Neural Networks," *IEEE, Trans. on Signal Processing*, vol. 40, No. 1, pp. 202-210, Jan. 1992