전력전자학술대회 논문집 1999. 7.21-24

유도전동기의 속도 센서리스 제어를 위한 신경회로망 알고리즘의 추정 특성 비교

*이 경훈, **국 윤상, ***최 원범, *김 윤호
*중앙대학교, **파워테크, ***여주대학교

Comparison of Different Schemes for Speed Sensorless Control of Induction Motor Drives by Neural Network

*Kyoung-Hun Lee, Yoon-Sang Kook, Yoon-Ho Kim, Won-Byum Choi
PowerTech 690-3, Keumjeong-Dong, GunPo-City, Dept. of Electrical Eng., Chung-Ang University

Abstract

This paper presents a newly developed speed sensorless drive using Neural Network algorithm. Neural Network algorithm can be divided into three categories. In the first one, a Back Propagation-based NN algorithm is well-known to gradient descent method. In the second scheme, a Extended Kalman Filter-based NN algorithm has just the time varying learning rate. In the last scheme, a Recursive Least Square-based NN algorithm is faster and more stable than the classical back-propagation algorithm for training multilayer perceptrons. The number of iterations required to converge and the mean-squared error between the desired and actual outputs is compared with respect to each method. The theoretical analysis and experimental results are discussed.

I. INTRODUCTION

System identification is a process aimed at establishing an adequate input/output relationship for unknown systems, and it is usually the first step taken by control engineers since control theory requires that we understand a system before we try to control it. Since the inception of artificial neural networks(ANN) many researchers have explored a wide variety of applications including identification of nonlinear dynamical system. Some of the advantages of using ANN as the model for system identification are: (i) ability to approximate arbitrary nonlinear functions to any degree of accuracy; (ii) they are adaptive, thus they can take data and learn from it, often capturing subtle relationships; (iii) they can generalize, therefore they can handle corrupt or incomplete data, thus providing a measure of fault tolerance; and (iv) they are highly parallel, which allows numerous independent operations to be executed simultaneously[1].

In general, an artificial neural network has a multilayer network structure. A widely used training method for a feed-forward multi-layer neural network (MNN) is the back-propagation algorithm developed by Rumelhart et al. in 1986, which is an iterative gradient algorithm designed to minimize the mean-square error between the desired output and the actual output for a particular input to the network with respect to the weights. Although it has worked successfully for a wide variety of applications, the standard back-propagation learning algorithm have several limitations. The long and unpredictable training process is the most troublesome, for example the rate of convergence is seriously affected by the initial weights and the learning rate of the parameters. In general, increasing the learning step size can speed up the convergence rate of the learning process, but it may also lead to divergence, paralysis, or continuous instability.

Many researchers have proposed modification of the classical back-propagation algorithm.

Wasserman incorporates several heuristics laws in the back-propagation algorithm, but they are difficult to describe systematically. Singhal and Wu incorporated a extended Kalman filering to improve the standard Steepest Descent technique. However, the computational complexity of this algorithm becomes intractable as the size of the MNN increases. Recently, another modified algorithm was derived by Scalero and Tepedelenliogu as an alternative to the back-propagation algorithm. It uses a modified form of the back-propagation algorithm to minimize the mean-square error between the desired output and the actual output with respect to the summation output (inputs to the nonlinearities). However, it is not a stable learning algorithm in practical real-life applications. Thus, a faster and more stable learning algorithm is desired and that is indeed the main purpose of this paper.

II. FLUX ESTIMATOR

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.

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

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

$i_s = [i_{ds} \;\; i_{qs}]^T \qquad \lambda_r = [\lambda_{dr} \;\; \lambda_{qr}]^T \qquad v_s = [v_{ds} \;\; v_{qs}]^T$

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,

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

where,

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

III. THE NEWLY PROPOSED SPEED SENSORLESS CONTROL ALGORITHM

A. Learning algorithm via the Back-Propagation

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

$$w_{ji}^{k-1,k}(t+1) = w_{ji}^{k-1,k}(t) + \Delta w_{ji}^{k-1,k}(t) \qquad (3)$$

where,

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

$\delta_j = (t_j - o_j) f'(i_j^k)$ for the hidden layer

$\delta_j = f'(i_j^k) \sum_k \delta_k w_{kj}$ for the output 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.

B. Learning algorithm via the Extended Kalman Filter

The multi-layered neural network is expressed by the following models with non-linear observation equations:

$$\hat{w}_{ji}(t+1) = \hat{w}_{ji}(t) + z(t) \qquad (4)$$

$$\hat{y}_j(t+1) = h(w_{ji}(t+1)) + \upsilon(t+1) = o_j^M(t+1) + \upsilon(t+1) \qquad (5)$$

where $\{z(t), \upsilon(t)\}$ are mutually independent, zero-mean noise with covariance matrix QandR regarded as a modeling error. Note that they can be considered pseudo-noises for tuning the gain of the extended Kalman filter. The application of the EKF to (4) and (5) gives the following real-time learning algorithms

$$w_{ji}(t+1) = w_{ji}(t) + K_{ji}(t)[y_j(t) - o_j^M(t)] \qquad (6)$$

$$K_{ji}(t) = \frac{P_{ji}(t+1|t) H_{ji}(t)^T}{[H_{ji}(t) P_{ji}(t+1|t) H_{ji}(t)^T + R]} \qquad (7)$$

$$P_{ji}(t+1|t) = P_{ji}(t|t) + Q \qquad (8)$$

$$P_{ji}(t+1|t+1) = [I - K_{ji}(t) H_{ji}(t)^T] P_{ji}(t+1|t) \qquad (9)$$

The filtered estimates of $w_{ji}^{k-1,k}$ at $k = M-1, K, 2$ $t+1$ are obtained by the following extended Kalman filter:

$$\hat{w}_{ji}^{k-1,k}(t+1) = \hat{w}_{ji}^{k-1,k}(t) + \eta_{ji}^{k-1,k}(t) \delta_j^k o_i^{k-1} \qquad (10)$$

where,

$$\eta_{ji}^{k-1,k}(t) = \frac{P_{ji}^{k-1,k}(t+1\ t)}{[H_{ji}(t)^T P_{ji}^{k-1,k}(t+1\ t) H_{ji}(t) + R]}$$

$$H_{ji}(t)^T = f'(i_j^k) o_i^{k-1}$$

$$\begin{cases} \delta_j^k = f'(i_j^k) \sum_{l=1}^{N_{k+1}} w_{jl}^{k\,k+1} \delta_l^{k+1} & \text{for } k = M-1, \Lambda, 2 \\ \delta_j^k = f'(i_j^k)(y_j - o_j^k) & \text{for } k = M \end{cases}$$

with initial conditions

$\hat{w}_{ji}^{k-1,k}(0) = \bar{w}_{ji}^{k-1,k} \qquad P_{ji}^{k-1,k}(0|0) = P_{ji}^{k-1,k}(0)$

C. Learning algorithm via the Recursive Least Square

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 :

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

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

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

$$K_{ji}(t)=\frac{P_{ji}(t+1\mid t)\Phi_{ji}(t)}{[\lambda I+\Phi^T_{ji}(t)P_{ji}(t+1\mid t)\Phi_{ji}(t)]} \quad (13)$$

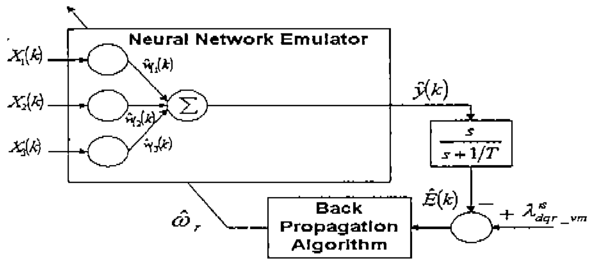
$$P_{ji}(t+1\mid t+1)=\lambda^{-1}[I-K_{ji}(t)\Phi^T_{ji}(t)]P_{ji}(t+1\mid t) \quad (14)$$

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

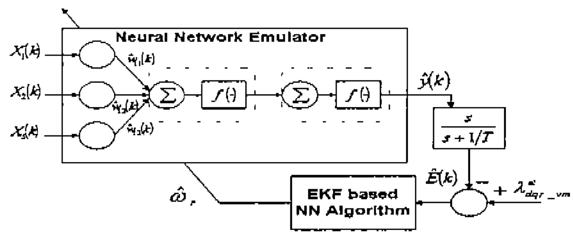
D. Speed sensorless control strategy

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 neuralmodel withadjustable 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 .

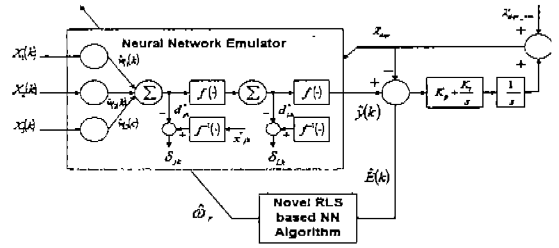
The rotor speed can be derived using the NN. The overall block diagram of speed sensorless control is shown in Fig. 1



(a) Back-Propagation algorithm



(b) Extended Kalman Filter algorithm



(c) Recursive Least Square algorithm
Fig 1

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

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

where,

$$\Phi^T(k)=[\hat{\lambda}^s_{qr\_cm}(k)\ \ \hat{\lambda}^s_{dr\_cm}(k)\ \ i^s_{qs}(k)] =[X_1(k)\ \ X_2(k)\ \ X_3(k)]$$

$$\hat{\theta}(k)=[1-1/\hat{\tau}_r\cdot T_s\ \ \ \hat{\omega}_r\cdot T_s\ \ \ L_m/\hat{\tau}_r\cdot T_s]^T =[\hat{w}_{11}(k)\ \ \hat{w}_{12}(k)\ \ \hat{w}_{13}(k)]^T$$

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

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

where, (16)

$$y_j(k)=\lambda^s_{qr\_vm}(k)$$

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

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

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

$$\lambda(k) = \lambda_0 \lambda(k-1) + (1-\lambda_0)$$
$$\lambda_0 = 0.98 \qquad P_\mu(0\,|\,0) = 500I$$

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. 2. 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
Pated Speed	2000rpm	L_r	44.09mH
Rated Torque	120Nm	L_m	42.1mH
R_s	0.115	J_M	0.1618 kgm^2
R_r	00821	P	4

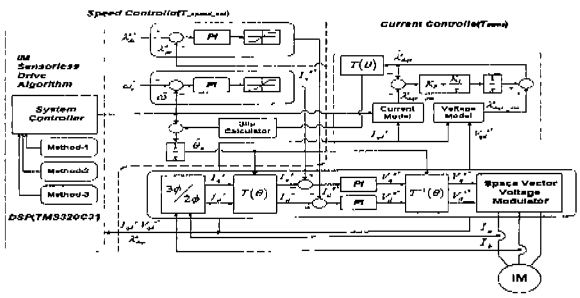
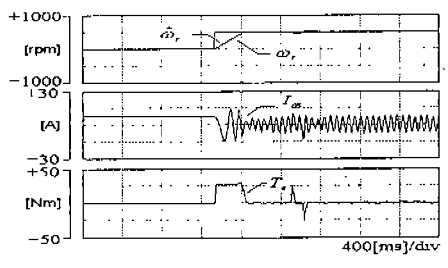
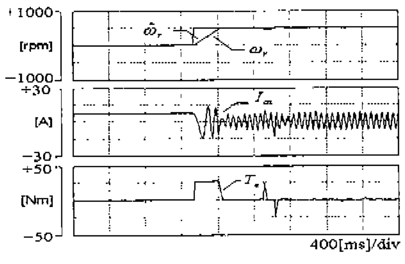


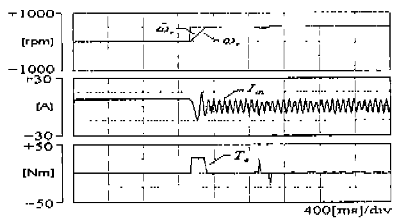
Fig. 2 The block diagram of the overall control algorithm.



(a) Back-Propagation algorithm



(b) Extended Kalman Filter algorithm



(c) Recursive Least Square algorithm

Fig. 3 The characteristics of speed step response (+500[rpm]-500[rpm], 0.5p.u. load).

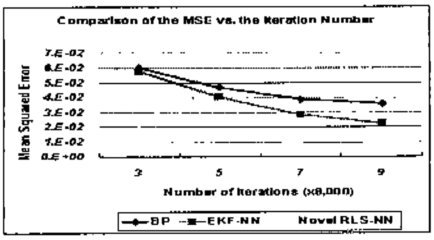


Fig. 4 The comparison of the mean squared error versus the iteration number for each NN algorithm.

The step response of the speed sensorless algorithm is shown in Fig. 3 when the speed reference is changed from 0[rpm] to 500[rpm]. As shown in fig. 3, we can know that the speed error of RLS-based NN algorithm is limited by 0.05% 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. Fig. 4 shows the comparison of the mean squared error versus the number of iteration for each method. Fig. 5 shows the system sensitivity to parameter variation. Fig. 6 shows the comparison of the mean squared error versus the noise sensitivity.

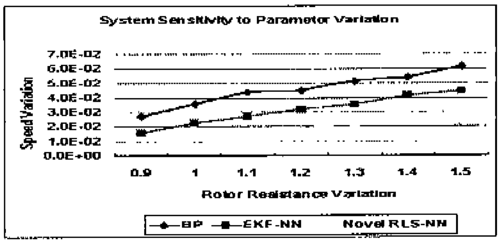


Fig. 5 The comparison of the speed variation versus the rotor resistance variation for each NN algorithm.

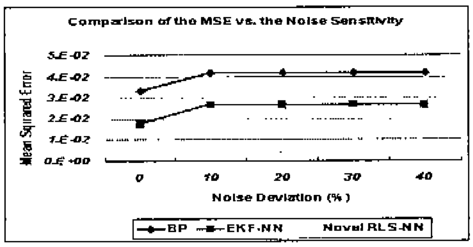


Fig. 6 The comparison of the mean squared error versus the noise deviation for each NN algorithm.

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.

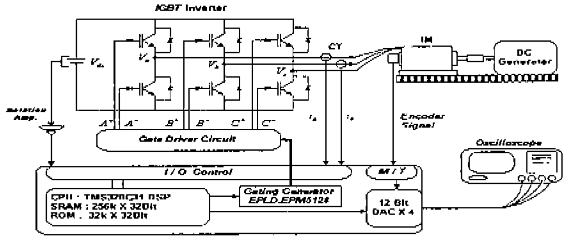
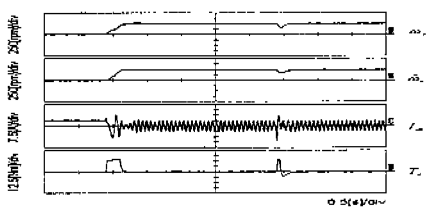
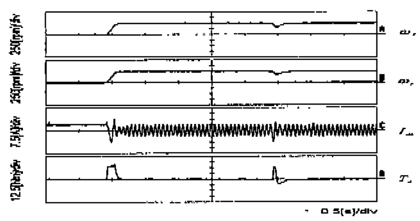


Fig. 7 The overall IM drive system.

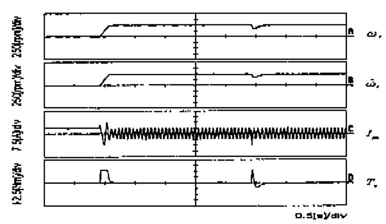
For actual load emulation, the DC generator is coupled to the IM. Actual rotor angle and machine speed are measured from an incremental encoder with 4096[ppr] resolution for monitoring. The sampling time of current controller loop is 250[s] and that of the outer voltage regulating loop and speed loop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



(a)Back-Propagation algorithm



(b) Extended Kalman Filter algorithm



(c) Recursive Least Square algorithm

Fig. 8 The experimental waveforms of step response (0[rpm]->+500[rpm], TL=0.5[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 speed sensorless algorithm is shown fig. 8 when the speed reference is changed with load torque.

It shows that the estimated speed is tracking the real one with good accuracy. The proposed algorithm works well in spite of the load torque variation and parameter variation.

VI. CONCLUSION

We have studied learning algorithm for multi-layered feed-forward type neural networks. Neural Network algorithm can be divided into three categories for speed sensorless control of induction motor drives.

1) Back Propagation-based NN algorithm
2) Extended Kalman Filter-based NN algorithm
3) Recursive Least Square-based NN algorithm

Table. 2 Comparison results on the speed sensorless systems.

	STE	DB	LS	PS	NS	C	CT
BP	3	3	3	3	4	2	2
EKFNN	2	2	2	2	2	3	4
RLSNN	1	1	2	1	2	2	2

1 : good, 4: bad,
STE : steady state error,
DB : dynamic behavior,
LS : low speed,
PS : parameter sensitivity,
C : complexity,
NS : noise sensitivity,
CT : computation time.

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.