A method of error compensation on sensors based on neural networks algorithm

Zhu Wei, Zeng Zhe-zhao, Huang Dan
College of Electrical and Information Engineering
Changsha University of Science and Technology
Chiling Road, No.45, Changsha, Hunan 410076
China
hncs6699@yahoo.com.cn

Abstract: - To improve the error compensation precision on sensors, a method of the error compensation on sensors based on the neural network algorithm is proposed. The convergence of the algorithm is researched. The theory gist to select learning rate is provided by the convergence theorem. To validate the validity of the algorithm, the simulating example of the error compensation on sensor was given. The result shows that the error compensation approach on sensors using the neural network algorithm has a very high accuracy. Thus, the method of the error compensation proposed is effective.

Key-Words: - Neural networks; sensor; error compensation; algorithm; convergence

1 Introduction

It is well known that sensors are usually sensitive to temperature, and also have nonlinear error. Therefore, to produce high-precision sensor must compensate for its error. There are many error compensation[1-4]，such as optimal linear fitting method[2,4], polynomial curve fitting approach based on least square method[6], compensation method of nonlinear inverse function[5], and nonlinear compensation method based on neural network algorithm[7],etc. Usually, linear fitting method and nonlinear compensation method of inverse function have larger compensation error，and lower compensation accuracy, hence, are seldom adopted. Polynomial curve fitting approach based on least square method has a higher precision of compensation，but when there are many data, it is easy to appear the oscillatory, and is unable to obtain the coefficients of polynomial, thus its application is restricted. The document [7] introduced a nonlinear compensation method based on neural network algorithm, and obtained better error compensation, but because activation functions of hidden layer neurons are non-orthogonal functions, convergence is very slow using the BP algorithms to train neural network, and it is necessary to update weights of the two-layer neurons. It is very obvious to need large computation with bad real-time performance. In order to improve the error compensation accuracy and the real-time performance of sensors, an error compensation method of sensors using the neural network algorithm with orthogonal basis functions is proposed.

2 Neural network model

The neural network model is showed in figure1. Where, the activation functions of hidden layer neurons are as follows:

$$c_0(x_k) = 1,\ c_1(x_k) = \cos\left[\frac{\pi}{b}(x_k - a)\right],\ldots\ldots,$$

$$c_N(x_k) = \cos\left[N\frac{\pi}{b}(x_k - a)\right] \quad (1)$$

Weights form the input layer to the hidden layer is identically equal to 1, and weights from hidden layer to the output layer are $w_n(n = 0,1,\cdots N)$. The $y(x_k)$ is the neural network output, and training samples set is：$\{x_k,\ \ y_d(x_k)|_{k=0,1,\cdots,m}\}$.

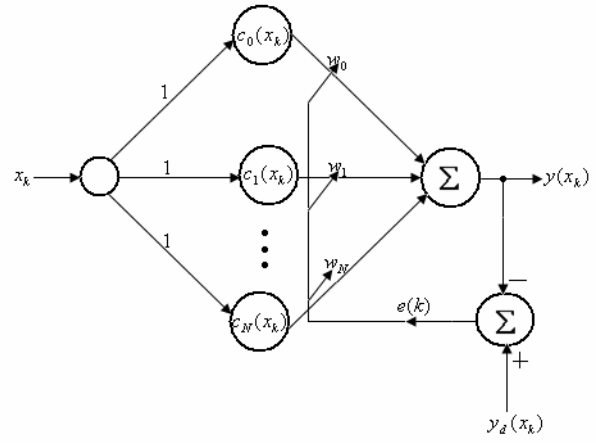

Figure 1 the model of neural network based on cosine basis functions

2.1 The algorithm description

According to figure 1, the neural network algorithm is as follows:

Given an arbitrary initial approximation weights: $w_n(n=0,1,\cdots N)$, then the output of neural network is as

$$y(x_k)=\sum_{n=0}^{N} w_n c_n(x_k) \tag{2}$$

And an error function can be obtained:

$$e(k)=y_d(x_k)-y(x_k) \tag{3}$$

Define an objective function J as follows:

$$J=\frac{1}{2}e^2(k) \tag{4}$$

To minimize the objective function J , the weight w_n is recursively calculated via using a simple gradient descent rule:

$$w_j^{k+1}=w_j^k-\mu\frac{dJ}{dw_j^k} \tag{5}$$

Where η is learning rate and $0<\mu<1$, and $\Delta w_j^k=-\mu\dfrac{dJ}{dw_j^k}$.

Differentiating (4) with respect to w_j^k , it can be obtained that

$$\Delta w_j^k=-\mu\frac{\partial J}{\partial w_j^k}=\mu e(k)\cos\left[j\frac{\pi}{b}(x_k-a)\right] \tag{6}$$

Substituting (6) into (5), we have

$$w_j^{k+1}=w_j^k+\mu e(k)\cos\left[j\frac{\pi}{b}(x_k-a)\right] \tag{7}$$

2.2 Research of the algorithm convergence

In order to ensure the absolute convergence of the algorithm proposed, it is important to select a proper learning rate μ . In the section, we present and prove the theorem about convergence of the algorithm. It is as follows:

Theorem 1: Suppose that μ is the learning rate, only when $0<\mu<2/\sum_{n=0}^{N}\left|c_n(x_k)\right|^2$, the algorithm is convergent.

Proof: Define $\mathbf{W}=[w_0,w_1,\cdots,w_N]^T$, a Lyapunov function as:

$$V(k)=\frac{1}{2}e^2(k) \tag{8}$$

then we have

$$\Delta V(k)=\frac{1}{2}e^2(k+1)-\frac{1}{2}e^2(k) \tag{9}$$

Since

$$e(k+1)=e(k)+\Delta e(k)=e(k)+\left(\frac{\partial e(k)}{\partial \mathbf{W}}\right)^T\Delta\mathbf{W} \tag{10}$$

And

$$\Delta\mathbf{W}=-\mu\frac{\partial J(k)}{\partial \mathbf{W}}=-\mu\frac{\partial J(k)}{\partial e(k)}\frac{\partial e(k)}{\partial \mathbf{W}}$$

$$= -\mu e(k)\frac{\partial e(k)}{\partial \mathbf{W}} \tag{11}$$

$$\Delta e(k) = \left(\frac{\partial e(k)}{\partial \mathbf{W}}\right)^T \Delta \mathbf{W} = -\mu e(k)\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2 \tag{12}$$

Where, $\|\mathbf{x}\|^2 = \sum_{i=1}^{n}|x_i|^2$ is the square of Euclidean norm.

According to (9), (10) and (12), we have

$$\begin{aligned}\Delta V(k) &= \Delta e(k)\left[e(k)+\frac{1}{2}\Delta e(k)\right] \\ &= -\mu e(k)\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2\left[e(k)-\frac{1}{2}\mu e(k)\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2\right] \\ &= \left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2 e^2(k)\left[-\mu+\frac{1}{2}\mu^2\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2\right]\end{aligned} \tag{13}$$

Since $\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2 e^2(k) \geq 0$, if the algorithm is convergent, i.e. $\Delta V(k) < 0$, then it is easy to see from (13) that:

$$-\mu+\frac{1}{2}\mu^2\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2 < 0 \tag{14}$$

Also since $\mu > 0$, we have

$$0 < \mu < 2/\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2 \tag{15}$$

According to (2) and (3), we have

$$\frac{\partial e(k)}{\partial w_n^k} = \frac{\partial e(k)}{\partial y(x_k)}\frac{\partial y(x_k)}{\partial w_n^k} = -c_n(x_k) \tag{16}$$

Therefore

$$\left\|\frac{\partial e(k)}{\partial \mathbf{W}}\right\|_2^2 = \sum_{n=0}^{N}|c_n(x_k)|^2 \tag{17}$$

Thus, we have

$$0 < \mu < 2/\sum_{n=0}^{N}|c_n(x_k)|^2 \tag{18}$$

2.3 Evaluation of the optimal learning rate

It is important to determine the magnitude of the learning rate μ during the training of neural network. Theorem1 indicates the theory criterion determining the magnitude of the learning rate μ. If learning rate μ is too large, the algorithm may produce oscillation and is not convergent at all. If learning rate μ is too small, the algorithm may be slowly convergent with more computation. In order to make the algorithm be convergent, according to experience, the optimal learning rate should usually be:

$$\eta_{opt}(k) = \begin{cases} 0.5, & \sum_{n=0}^{N}|c_n(x_k)|^2 \leq 1 \\ 1/\sum_{n=0}^{N}|c_n(x_k)|^2, & \sum_{n=0}^{N}|c_n(x_k)|^2 > 1 \end{cases} \tag{19}$$

2.4 Algorithm steps

INPUT: initial weights: $w_n^0(n=0,1,\cdots N)$; training sample set: $\{x_k,\ \ y_d(x_k)\}$; tolerance *Tol*; maximum number of iterations N; let k=0;

OUTPUT: The weights: $w_n^k(n=0,1,\cdots N)$

Step 1: While k<=N do Steps 2-5

Step 2: computing the output of neural network:

$$y(x_k) = \sum_{n=0}^{N} w_n c_n(x_k)$$

Let $e(k) = y_d(x_k) - y(x_k)$, and $J = \frac{1}{2}|e(k)|^2$

$$\text{If } \sum_{n=0}^{N}\left|c_n(x_k)\right|^2 \le 1$$

$$\eta_{opt}(k) = 0.5$$

$$\text{Else } \mu_{opt}(k) = 1/\sum_{n=0}^{N}\left|c_n(x_k)\right|^2$$

Step 3: Update the weights:

$$w_j^{k+1} = w_j^k + \mu e(k)\cos\left[j\frac{\pi}{b}(x_k - a)\right]$$

Step 4: If *J*<=*To*l then

OUTPUT $w_n^k(n = 0,1,\cdots N)$; (The procedure was successful.)

STOP

Step 5: Set k=k+1

Go back to step 2

Step 6: OUTPUT ('the method failed after N iterations);

(The procedure was unsuccessful.)

STOP

3 Simulation examples

3.1 Error compensation of magnetic sensor

As the Hall element is not only the magnetic sensitive element, but also the temperature sensitive element, therefore, both nonlinear error compensation and temperature error compensation of magnetic sensors should be considered. In order to validate the validity of the proposed algorithm, the nonlinear and temperature characteristic of magnetic sensor were compensated using the algorithm proposed. The data used in the algorithm were selected from the reference [4], it is shown in table1.

3.1.1 Nonlinear characteristic results using neural network algorithm

In the algorithm proposed, let a=50, and b=200. Take measured value of magnetic field as the input of neural network, and standard value of magnetic field as the training samples of neural network. Give *Tol*=10^{-6}, and produce random initial weights vector: **W**=*rand*(4,1). After neural network was trained for 6 times, neural network weights were as follows:
W=[120.0401585, -63.3564323, -0.0381913
-7.6457139]

3.1.2 Temperature characteristic results using neural network algorithm

Let a=1, and b=42. Use temperature as the input of neural network, and measured values of magnetic field as the training sample set of neural network. Give *Tol*=10^{-6}, and produce random initial weights vector: **W**=*rand*(5,1). After neural network trainings for 10 times, neural network weights were as:

W=[100.0000635, 0.0001683, 0.0000392,
-0.0002069, -0.0001578]

3.1.3 The compensation results of the magnetic sensor

According to the obtained above weights of neural network, the compensation results of nonlinear characteristic and temperature characteristic of magnetic sensors were showed in table 2.

Table1 [4]

Nonlinear characteristic		Temperature characteristic standard magnetic field given: 100mT	
Standard magnetic field: Bn/mT	Measured value of magnetic field Vx/mT	Temperature values T/℃	Measured value of magnetic field: Yt/mT
50.00	50.11	1.6	103.42
100.00	100.08	10.0	102.09
150.00	149.91	20.0	100.68
200.00	199.63	30.2	99.46
		39.9	98.49

Table2 The compensation results of magnetic sensor

Nonlinear characteristic		Temperature characteristic standard magnetic field: 100mT	
The standard magnetic field: Bn/mT	The compensation values of magnetic field: Vx/mT	Temperature values T/℃	The compensation values of magnetic field: Yt/mT
50.00	50.0005984	1.00	99.9999063
100.00	100.0007465	10.00	100.0004358
150.00	149.9990056	20.12	100.0000006
200.00	199.9998518	31.20	99.9999128
		43.00	99.9999834

4 Conclusion

We know from table2 that the nonlinear characteristic and temperature characteristic of the magnetic sensor were compensated very well using the neural network algorithm with cosine basis functions. The compensation accuracy is very high, such as, the compensation accuracy of the nonlinear characteristic reaches to 0.001%, and the compensation accuracy of the temperature characteristic reaches to 0.0004%. The range of the temperature compensation is from 1.0℃ to 43.0℃. Compared with reference [4], it has significant advantages. Therefore, it is an effective method of sensor error compensation.

References:

[1] Tandeske D. Pressure Sensors. Marcel Dekker, INC., 1991

[2] Huiqing Sun,Zhiyou Guo. Pressure sensor and its error compensation. *Sensor world*, Vol.8,No.3,2002, pp.14-16

[3] Jun Xu. Using single-chip processor software to achieve the sensor′s temperature error compensation. *Modern Electronics Technique*, Vol.26, No.10, 2002, pp.97-99

[4] Huiqing Sun,Zhiyou Guo. Technology of error compensation on sensors.*Chinese Journal of Sensors and Actuators,* Vol. 17, No.1, 2004, pp.90-92

[5]Desheng Li, Xinmin Zhao. An inverse function correction methods of sensor nonlinear characteristic [J]. *Chinese Journal of Scientific Instrument*, Vol.12, No.2, 1991, pp.215-218

[6]Yicai Sun. Normalizing the polynomial-match for a nonlinear function in sensors and solid electronics. *Chinese Journal of electronic devices*, Vol.27, No.1, 2004, pp.1-7

[7]Ya su, Yicai Sun. Comparing the different arithmetic methods for the offset drift compensation of pressure sensor . *Chinese Journal of Sensors and Actuators*, Vol.17, No.3, 2004, pp.375-378