SCIENCE CHINA
Information Sciences

• LETTER •

July 2020, Vol. 63 179201:1–179201:3
https://doi.org/10.1007/s11432-018-9514-9

Blocked WDD-FNN and applications in optical encoder error compensation

Fang DENG*, Jiachen ZHAO & Yeyun CAI

School of Automation, Beijing Institute of Technology, Beijing 100081*, China*

Received 27 April 2018/Accepted 29 June 2018/Published online 8 October 2019

Citation Deng F, Zhao J C, Cai Y Y. Blocked WDD-FNN and applications in optical encoder error compensation. Sci China Inf Sci, 2020, 63(7): 179201, https://doi.org/10.1007/s11432-018-9514-9

Dear editor,

Optical encoder is a kind of angle sensor, which has been applied in a variety of domains such as astronomy, vehicles and industrial production. However, there always exist errors between the measured angles and the real angles, reducing the performance and safety of systems. These measurement errors may raise from various sources, like error of installation and mechanical wear. The sensor errors can be classified into two categories: random errors which are difficult to completely eliminate and repeatable errors (systematic errors) which can be effectively reduced by the compensation techniques.

This study focuses on the problem of systematic error compensation, which can be described as follows: given the raw measurement x and the ideal measurement y, to find a compensation function $\tilde{x} = f(x)$ that minimizes the estimation error $\|\tilde{x} - y\|_2$, where $\|\cdot\|_2$ is 2-norm. Error compensation for optical encoders is a complex task because of the big volume and irregular nonlinearity of output data. However, neural networks (NNs) are particularly adept at solving this problem because of their ability of arbitrary nonlinear expressiveness. Reviewing the existing literature, we find that many studies have proposed NN-based algorithms to compensate errors for different sensors, such as multilayer perceptron (MLP) [1], radial basis function network [2], Fourier neural network [3], and recurrent fuzzy neural network [4]. However, almost all of these methods are suffering from the long training time, so these methods can only be applied on the low-precision sensors [5].

We propose a novel compensation method, called blocked weight direct determination–Fourier neural network (blocked WDD-FNN). The output series that sensor produced is usually very long, so using a single model to handle such a large amount data cannot gain a good compensation performance. To handle this problem, we first segment the output series into N blocks, and each block is compensated by a three-layer FNN so that the amount of data processed by each FNN is relatively small. In the training process, we use a parallel-computing framework where each FNN is trained in a separate processing core, and weight direct determination algorithm is used to determine the connection weights of FNNs.

Methodology. We introduce the structure of blocked WDD-FNN and specific steps to use blocked WDD-FNN to compensate sensor errors. The input is the original series containing all the raw outputs of the sensor to be compensated, and the original series is segmented into N blocks, called blocked series. Then N blocked series are compensated by N Fourier neural networks, and outputs of FNNs are the final compensated series. The specific steps are described as follows.

Step 1. Prepare the training set. As a supervised learning method, blocked WDD-FNN requires a training set for the sensor to be compensated. All the raw measurements x and ideal measurements y constitute a set of training samples

* Corresponding author (email: dengfang@bit.edu.cn)

$\{(x_1, y_1)\ (x_2, y_2)\ (x_M, y_M)\}$. The number of training samples M is determined by the resolution of the sensor. For example, a 15-bit angle encoder can generate 2^{15} outputs, so the training set has 2^{15} training samples. Generally, the training sets can be produced by the sensor precision testing equipments. The original series are divided into N blocks, so each blocked series contains $n = M/N$ samples. In the next step, each blocked series is compensated by an individual Fourier neural network.

Step 2. Build the blocked Fourier neural networks. We use three-layer Fourier neural networks to represent and compensate the nonlinear error of blocked series. The neurons in the input layer send the original data out, and all the input weights are set as 1, which is different from the back propagation network whose weights need to be adjusted during training. The excitation functions of hidden neurons are Fourier sin and cosine trigonometric functions, defined as follows.

$$\begin{cases} \varphi_1 = 1, \\ \varphi_{2i}(x) = \sin\left(\dfrac{2\pi ix}{T}\right), \\ \varphi_{2i+1}(x) = \cos\left(\dfrac{2\pi ix}{T}\right), \end{cases} \quad i = 1, 2, \ldots m, \quad (1)$$

where T is the periodicity of signal in the theory of Fourier approximation; the number of hidden neurons is $2m+1$; the output weights could be written as $w = [a_0, a_1, b_1, \ldots, a_m, b_m]^{\mathrm{T}}$. The weighted sum of all hidden neurons is calculated in the output layer, and the outputs of FNNs are the compensated measurements shown as

$$\tilde{x} = a_0 + \sum_{i=1}^{m}\left(a_i \cos\left(\frac{2\pi ix}{T}\right) + b_i \sin\left(\frac{2\pi ix}{T}\right)\right). \quad (2)$$

The final step is to train the FNNs.

Step 3. Train the FNNs with WDD. Training processes for all FNNs are similar, so we take one blocked series and respective FNN for example to introduce how we use weight direct determination algorithm to train the FNN. The goal of training FNN is to find a set of weights $w = [a_0, a_1, b_2, \ldots, a_m, b_m]^{\mathrm{T}}$ that minimizes the training error, i.e.,

$$\underset{\tilde{x}_j, j=1,2\ldots n}{\text{minmize}} \sum_{j=1}^{n} \|\tilde{x}_j - y_j\|_2. \quad (3)$$

For simplicity, we combine the (2) and (3), and write it in matrix as

$$\underset{W}{\text{minmize}} \|\Phi W - Y\|_2, \quad (4)$$

where Φ is the output matrix of hidden neurons, W is the output weight matrix, and Y is the ideal value vector. Φ, W and Y are respectively defined as

$$\Phi = \begin{bmatrix} \varphi_1 & \varphi_2(x_1) & \cdots & \varphi_{2m+1}(x_1) \\ \varphi_1 & \varphi_2(x_2) & \cdots & \varphi_{2m+1}(x_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1 & \varphi_2(x_n) & \cdots & \varphi_{2m+1}(x_n) \end{bmatrix},$$

$$W = \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Using the least squares method to solve (4), we can get the output weight matrix W, i.e.,

$$W = \Phi^{\dagger} Y,$$

where $\Phi^{\dagger}$ is Moore-Penrose pseudoinverse of Φ.

There are two important parameters, the number of blocks N and the number of hidden neurons in FNNs $2m+1$, which influence the performance of blocked WDD-FNN. In practical applications, we can adjust these parameters by statistical method or simple trial and error method.

Experimental setup. We conduct several experiments, in which blocked WDD-FNN is used to compensate a 15-bit optical encoder. A precision detecting device for angle sensor is used to measure and record the output value within full range. PC serves as a controller for steeper motor. The high-precision stepper motor serves as a drive to provide driving rotation for the encoder, and the rotation is transmitted to the encoder through a speed reducer and interfaces. The data acquisition card collects the output x_i of the angle sensor at each detection position into PC. The ideal value y_i is given by the high-precision stepper motor. Using this device, we can get a training set to train the blocked WDD-FNN. To evaluate the performance of different algorithms, three common criterions are taken into account. The mean absolute error (MeanAbs), the maximum absolute error (MaxAbs) and the standard deviation of errors (STD) are defined as (5)–(7):

$$\text{MeanAbs} = \frac{1}{M}\sum_{i=1}^{M} |e_i|, \quad (5)$$

$$\text{MaxAbs} = \max\left\{|e_i|, i = 1, 2, \ldots, M\right\}, \quad (6)$$

$$\text{STD} = \left[\sum_{i=1}^{M} (e_i - \bar{e}_i) / (M-1)\right]^{\frac{1}{2}}, \quad (7)$$

where $e_i = \tilde{x}_i - y_i$, $\bar{e}_i = \frac{1}{M}\sum_{i=1}^{M} e_i$, M is the number of samples.

A comparative study including least square estimation model (LSE), back-propagation neural network (BP-net) and standard FNN is conducted. BP-net has 15 hidden neurons with sigmoid function and is trained by Levenberg-Marquardt method (LM). In LSE, the dispersed data is imitating to linear function curve aiming at least square error. FNN is trained by LM method without parallel computation. We also compare the running time of all algorithms. Note that 50 trials have been conducted for and the average performance is recorded.

Results and discussion. Table 1 shows the raw measurement errors before compensation and the errors compensated by different algorithms. Before compensation, MeanAbs, MaxAbs and STD of measurement errors are 0.5387°, 1.4654° and 0.4610°, respectively. Although LSE has the lowest time complexity, it cannot reduce the errors effectively with an almost unchanged STD. BP-net and FNN get similar performance, where MeanAbs and STD reduced by an order of magnitude. However the main disadvantage of BP-net and FNN is their long training time, which is nearly 30 times that of the blocked WDD-FNN. Blocked WDD-FNN has the best compensation performance, and MeanAbs, MaxAbs and STD of measurement errors are respectively reduced to 0.0229°, 0.3281° and 0.0212°. The decrease of the average absolute error and the maximum absolute error indicates that the average and the worst level of error are reduced. The standard deviation of errors indicates that the error dispersion is smaller after compensation. The training time of blocked WDD-FNN is 1.57 s, which is only longer than LSE. The relatively small sub-training set for each FNN and WDD algorithm is the main reason why blocked WDD-FNN can achieve the high compensation performance in a short training time.

Table 1 Experimental results (° is the unit of angle)

	MeanAbs	MaxAbs	STD	Time (s)
Pre-compensation	0.5387°	1.4654°	0.4610°	
LSE	0.3919°	0.9796°	0.4607°	**0.27**
BP-net	0.0747°	0.2770°	0.0876°	44.00
FNN	0.0732°	0.3281°	0.0864°	40.60
WDD-FNN	**0.0229°**	**0.1108°**	**0.0212°**	1.57

Conclusion. We propose a novel sensor compensation algorithm called blocked WDD-FNN. In blocked WDD-FNN, a set of FNNs is used to present the nonlinear error, and a parallel-computing WDD algorithm is used to train the FNNs. There are two main advantages of blocked WDD-FNN. (1) It has the ability to compensate each point of sensors instead of probabilistic sampling. (2) The parallel computing framework and high training speed of WDD significantly reduce the training time. The experiments with a 15-bit optical encoder demonstrate the effectiveness of the blocked WDD-FNN. In the future, we will try to explore efficient methods for parameter determination of blocked WDD-FNN.

Acknowledgements This work was supported by Beijing Nova Program (Grant No. xx2016B027).

References

1 Cotton N J, Wilamowski B M. Compensation of sensors nonlinearity with neural networks. In: Proceedings of IEEE International Conference on Advanced Information Networking And Applications, Waina, 2010. 1210–1217
2 Rădulescu A. Neural network spectral robustness under perturbations of the underlying graph. Neural Comput, 2016, 28: 1–44
3 Deng F, Chen J, Wang Y Y, et al. Measurement and calibration method for an optical encoder based on adaptive differential evolution-Fourier neural networks. Meas Sci Technol, 2013, 24: 055007
4 Wu B, Wu K, Lü J H. A novel compensation-based recurrent fuzzy neural network and its learning algorithm. Sci China Ser F-Inf Sci, 2009, 52: 41–51
5 Yan H H, Deng F, Sun J, et al. An NN-based SRD decomposition algorithm and its application in nonlinear compensation. Sensors, 2014, 14: 17353–17375