

Article

Fault Restoration of Six-Axis Force/Torque Sensor Based on Optimized Back Propagation Networks

Xuhao Li ^{1,2} , Lifu Gao ^{2,3}, Xiaohui Li ⁴, Huibin Cao ^{2,3,*} and Yuxiang Sun ^{2,*}

¹ Institutes of Physical Science and Information Technology, Anhui University, Hefei 230031, China

² Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China

³ Department of Science Island, University of Science and Technology of China, Hefei 230026, China

⁴ Beijing Institute of Control Engineering, Beijing 100080, China

* Correspondence: hbcao@iim.ac.cn (H.C.); yxsun@iim.ac.cn (Y.S.); Tel.: +86-0551-6559-2409 (H.C.)

Abstract: Six-axis force/torque sensors are widely installed in manipulators to help researchers achieve closed-loop control. When manipulators work in comic space and deep sea, the adverse ambient environment will cause various degrees of damage to F/T sensors. If the disability of one or two dimensions is restored by self-restoration methods, the robustness and practicality of F/T sensors can be considerably enhanced. The coupling effect is an important characteristic of multi-axis F/T sensors, which implies that all dimensions of F/T sensors will influence each other. We can use this phenomenon to speculate the broken dimension by other regular dimensions. Back propagation neural network (BPNN) is a classical feedforward neural network, which consists of several layers and adopts the back-propagation algorithm to train networks. Hyperparameters of BPNN cannot be updated by training, but they impact the network performance directly. Hence, the particle swarm optimization (PSO) algorithm is adopted to tune the hyperparameters of BPNN. In this work, each dimension of a six-axis F/T sensor is regarded as an element in the input vector, and the relationships among six dimensions can be obtained using optimized BPNN. The average MSE of restoring one dimension and two dimensions over the testing data is 1.1693×10^{-5} and 3.4205×10^{-5} , respectively. Furthermore, the average quote error of one restored dimension and two restored dimensions are 8.800×10^{-3} and 8.200×10^{-3} , respectively. The analysis of experimental results illustrates that the proposed fault restoration method based on PSO-BPNN is viable and practical. The F/T sensor restored using the proposed method can reach the original measurement precision.

Keywords: force/torque sensor; back propagation neural network; fault restoration; coupling; particle swarm optimization



Citation: Li, X.; Gao, L.; Li, X.; Cao, H.; Sun, Y. Fault Restoration of Six-Axis Force/Torque Sensor Based on Optimized Back Propagation Networks. *Sensors* **2022**, *22*, 6691. <https://doi.org/10.3390/s22176691>

Academic Editor: Roberto Teti

Received: 2 August 2022

Accepted: 1 September 2022

Published: 4 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multi-axis F/T sensors play an important role in high-end manufacturing fields. For example, F/T sensors can provide feedback for researchers to achieve fine grab in deep sea and the remote control in comic space [1–3]. However, high pressure under deep sea and high vacuum, high-low temperatures in comic space and other adverse factors will cause damage to F/T sensors [4]. It can be helpful and economical if the damaged F/T sensors can be restored rather than replaced when the damage is not fatal. Therefore, fault restoration is meaningful and necessary for multi-axis F/T sensors.

A six-axis strain gauge sensor is taken as an example of multi-axis F/T sensors in this paper, and the proposed method can be applied to other types of multi-axis F/T sensors with minor modifications. An elastic body and several strain gauges constitute the six-axis F/T strain gauge sensor, and every four strain gauges (two strain gauges for semi-bridge) pasted on an elastic body compose a Wheatstone bridge. Because all strain gauges are pasted on the same elastic body, applying a load in one direction will induce deformation not only on the corresponding dimension, but also on other dimensions, which

is called coupling. The coupling effect will reduce the measurement precision of six-axis F/T sensors, and scholars have proposed numerous decoupling methods to eliminate the effects of coupling. Song et al. [5] proposed a robust static decoupling algorithm for 3-axis force sensors based on ε -SVR, and Liang et al. [6] proposed a decoupling method based on parallel voltage extreme learning machine (PV-ELM) for six-axis F/M sensors. Nevertheless, if we make use of the coupling effect and find the correlations among all dimensions, it is feasible to achieve restoration of slight damages of F/T sensors.

The key to fault restoration for F/T sensors is to find out how the output of one dimension is affected by the loads from other dimensions. Mechanical analysis and finite element analysis (FEA) can reveal inherent correlations between dimensions. Niu et al. [7] analyzed the structure of a six-dimensional parallel-mechanism force sensor and proposed a new structure to minimize the coupling effect. Nevertheless, mechanical analysis and FEA are costly and demand too much prior knowledge. Machine learning has attracted much attention recently. Neural network is a major branch of machine learning, which is widely used for regression and classification in engineering, economics, and other fields [8,9].

Back propagation (BP) neural networks have simple structures and high efficiency, which makes them popular in fault detection and restoration [10–13]. BPNNs pass information through multiple hidden layers and calculate gradients of weights and biases by the back-propagation algorithm. Hyperparameters such as the counts of layers and the width of hidden layers will effect the convergence rate and performance of BPNN. Hyperparameters are normally selected by blind trails which are costly and not satisfying. Particle swarm optimization is a famous multi-objective optimization algorithm, and it is utilized for acquiring optimal solutions in math and engineering [14–17]. Lin et al. [18] proposed an adaptive dissipative particle swarm optimization (ADPSO) algorithm, which is used to solve the resource balancing optimization problem for different network-plans scales.

In this work, we proposed a fault restoration method based on the particle swarm optimization (PSO) algorithm optimizing back propagation neural networks (BPNN). After that, we conducted a coupling experiment to research the impact of coupling effects and obtain the transfer expression for the six-axis F/T sensor. Finally, some simulations are conducted to evaluate the proposed method. The rest of this paper is organized as follows: the concept and algorithm of BPNN and the PSO algorithm are briefly reviewed, then a novel fault restoration method for six-axis F/T sensors is proposed based on PSO-BPNN. Experiments including sensor calibration and model training are presented. Finally, the performance of the proposed method is discussed through the analysis of experiment results.

2. Methodology

2.1. Back Propagation Neural Networks

A BP neural network consists of one input layer, one output layer and several hidden layers. Each hidden layer owns an activation function, which calculates responses of layer nodes. Assuming $\mathbf{X} = \{x_1, x_2, \dots, x_m\}^T$ is the input vector, $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}^T$ is the output vector, the typical structure of BPNN is shown in Figure 1.

2.1.1. Forward Propagation

The input data flows through a series of hidden layers in turn, and the output of hidden layers can be expressed as follows:

$$\mathbf{u}^h = f(\mathbf{W}^h \times \mathbf{X} + \mathbf{b}^h), \quad (1)$$

where

$h = \{1, 2, \dots, H\}$ is the index of layer,

$\mathbf{W}^h \in \mathbb{R}^{n \times l}$ and $\mathbf{b}^h \in \mathbb{R}^l$ are the weight matrix and the bias vector of h 'th layer, respectively,

$f(\cdot)$ the nonlinear activate function.

The model output Y can be obtained by the same formula in (1), except that the output activation function is set to linear function:

$$\hat{Y} = f(\mathbf{V} \times \mathbf{u}^h + \mathbf{b}^o), \quad (2)$$

where $\mathbf{V} \in \mathbb{R}^{m \times n}$ and $\mathbf{b}^o \in \mathbb{R}^n$ are the weight matrix and bias vector of the output layer.

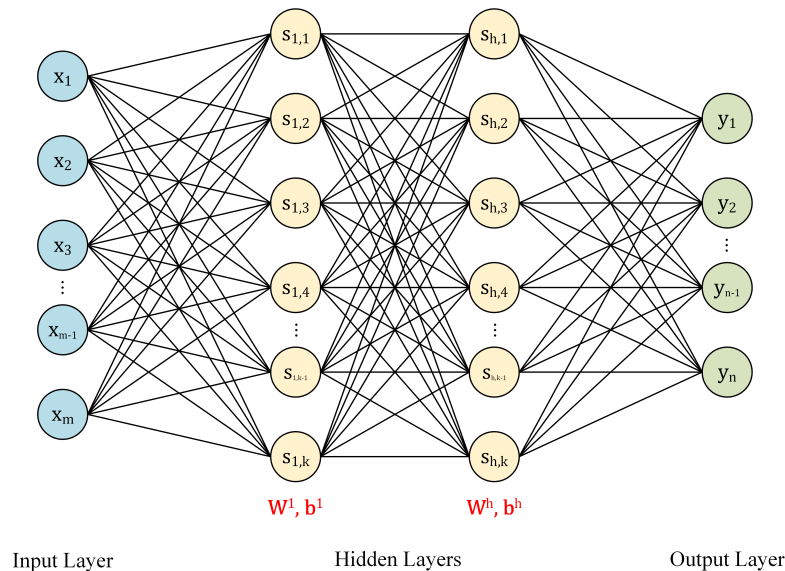


Figure 1. A typical structure of BPNN, $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and the bias vector, respectively.

2.1.2. Backward Propagation

The training principle of BPNN is to reduce the total errors in the dataset. Mean square error (MSE) is a common criterion for regression tasks, which is formulated as follows:

$$loss = MSE(Y, \hat{Y}) = \frac{1}{l} \sum_{i=1}^l (y_i - \hat{y}_i)^2, \quad (3)$$

BPNN adopts the back-propagation algorithm to train the network, which updates weights and biases along the gradient directions.

2.1.3. Repetition and Termination

The progress of forward and backward propagation are repeated iteratively, and the training terminates when it reaches the maximum iteration or the target error. The maximum iteration is essential for model training, and a suitable maximum iteration should be selected for keeping balance between under-fitting and over-fitting.

2.2. Particle Swarm Optimization

Meta-heuristic algorithms have attracted a lot of attention in many fields, such as mathematics, cyber-security, and the Internet of Things (IoT) [19,20]. These algorithms have three attractive characteristics: simplicity, flexibility, and the ability to avoid local optima. The particle swarm optimization algorithm is a meta-heuristic optimization algorithm, and it is inspired by crowd behavior observed in insects, fishes and birds.

Assuming $\mathbf{u}_i = (u_{i1}, u_{i2}, \dots, u_{in})$, $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{im})$ are the current position, and the speed of iteration i , respectively. $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{im})$ presents the optimal position and $\mathbf{p}_i^* = (p_{i1}^*, p_{i2}^*, \dots, p_{im}^*)$ is the global optimal position of each particle in iteration i . The basic evolution expression can be formulated as follows:

$$\mathbf{v}_{id}(t+1) = w \times \mathbf{v}_{id}(t) + c_1 \times r_1 \times (\mathbf{p}_{id}(t) - \mathbf{x}_{id}(t)) + c_2 \times r_2 \times (\mathbf{p}^*(t) - \mathbf{x}_{id}(t)), \quad (4)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (5)$$

where $i = \{1, 2, \dots, n\}$ is the index of particle, $d = \{1, 2, \dots, D\}$ is the dimension index; $t = \{1, 2, \dots, T\}$ indicates the current generation and w is inertia factor; and r_1, r_2 are random variables in the range of $[0, 1]$.

The inertia factor w is negative. The bigger w means the better global search ability, while the smaller w means the better local search ability. Generally, dynamic strategies are adopted to adapt the inertia factor w to the searching progress, which is shown as follows:

$$w(t) = (w_{max} - w_{min}) \cdot \frac{(T - t)}{T} + w_{min}, \quad (6)$$

where w_{max} and w_{min} are the maximum and minimum inertia factor, respectively. T indicates the maximum iteration.

The pseudo code of PSO with dynamic strategy is shown in Algorithm 1.

Algorithm 1: Dynamic Particle Swarm Optimization

Inputs: The count of dimensions D , the count of particles N , the maximum iteration T , the boundary of inertia factor $[w_{min}, w_{max}]$.
Outputs: The optimal position of all dimensions.
Process:

1. **for** each particle i
2. Initialize velocity v_i and position x_i , set $p_i = x_i$;
3. Evaluate the fitness;
4. **end for**
5. $\mathbf{p}^* = \min(\mathbf{p}_i)$;
6. **while** $t \leq T$
7. **for** $i = 1$ to N
8. Update the velocity and position by (4) and (5);
9. Update the inertia factor by (6);
10. Evaluate the fitness of particle i ;
11. **if** $fitness(\mathbf{x}_i) < fitness(\mathbf{p}_i)$
12. $\mathbf{p}_i = \mathbf{x}_i$;
13. **if** $fitness(\mathbf{p}_i) < fitness(\mathbf{p}^*)$
14. $\mathbf{p}^* = \mathbf{p}_i$;
15. **end for**
16. **end while**
17. **return** $\mathbf{p}^*$

3. Fault Restoration Based on PSO-BPNN

The target for fault restoration is predicting the output of the broken dimension based on the correlations among other dimensions. Considering the complex of mutual impacts of coupling among dimensions, two schemes are assumed in this work: one arbitrary broken dimension restored by the other five dimensions and two arbitrary broken dimensions restored by the other four dimensions.

This chapter will describe how to apply the BPNN for restoring the broken dimension in the aforesaid two schemes and tuning hyperparameters of BPNN by the PSO algorithm. Lastly, the flowchart of the proposed method is presented.

3.1. Dataset Preparation

Assuming the input vectors are $\mathbf{X}_i = [x_i^1, x_i^2, \dots, x_i^m]^T$, and the output vectors are $\mathbf{U}_i = [u_i^1, u_i^2, \dots, u_i^n]^T$, where $(m, n) \in \{(5, 1), (4, 2)\}$. For the first scheme which restores one damaged dimension by the other five dimensions, input vectors contain voltages of five dimensions, and output vectors contain voltages of the broken dimension. For the

second scheme, which restores two damaged dimensions by another four dimensions, input vectors contain voltages of four dimensions, and output vectors contain voltages of the two broken dimensions. The data is collected in coupling experiments by exerting loads orderly on dimensions. About 70% of collected data will be separated for training, while the remaining 30% for testing.

3.2. The BP Network

3.2.1. Network Structures

The basic BP network consists of one input layer, one output layer and one hidden layer. Besides, the amount of hidden layer elements is selected by experience. In this work, the PSO algorithm is utilized to obtain the optimal counts of hidden layers and elements in each hidden layer. The counts of elements in input and output layers are decided according to Section 3.1.

3.2.2. Activation Function

Activation function $g(z)$ for all hidden layers is the hyperbolic tangent activation function, which is formulated as follows:

$$g(z) = \tanh(z).$$

Additionally, the linear function is selected for the output layers, which is shown as the following equation:

$$g(z) = z.$$

3.2.3. Cost Function

The BP networks use a cost function to calculate the error between predicted values and real values. Mean square error is selected to be the cost function in this work, which is suitable for regression task.

3.2.4. Maximum Iteration

The BP networks will update weights and bias during training progress iteratively. Early Stopping, a type of parameter fine tuning strategy, calculates the accuracy of a model at the end of each cycle and stop training when the accuracy is no longer increasing. Hence, appropriate iterations can benefit to alleviate the under-fitting or over-fitting problems. The PSO algorithm is utilized to search for the optimal target iteration, which enhances the model performance.

3.3. Model Optimization by PSO

This section will illustrate how PSO is applied to optimize hyperparameters for the BPNN. As shown in Section 3.2, there are three hyperparameters to be optimized in BPNN, including the counts of hidden layers, the counts of neurons in each hidden layer, and the maximum iterations.

3.3.1. Searching Space of Parameters

Considering the balance between model complexity and performance, we selected an approximate searching range for parameters, which is listed in Table 1.

Table 1. Searching ranges of hyperparameters.

Parameters	Ranges
counts of hidden layer	[1, 5]
neuron counts in each hidden layer	[5, 50]
maximum iteration	[500, 1000]

3.3.2. PSO Configurations

After a large number of experiments, the particle count of PSO was set to 50. The target iteration of PSO was set to 300 according to the model precision. The update strategy of inertia factor w is linear decreasing, as shown in (6).

The implementation process of the BPNN optimized using the PSO algorithm is illustrated in Figure 2:

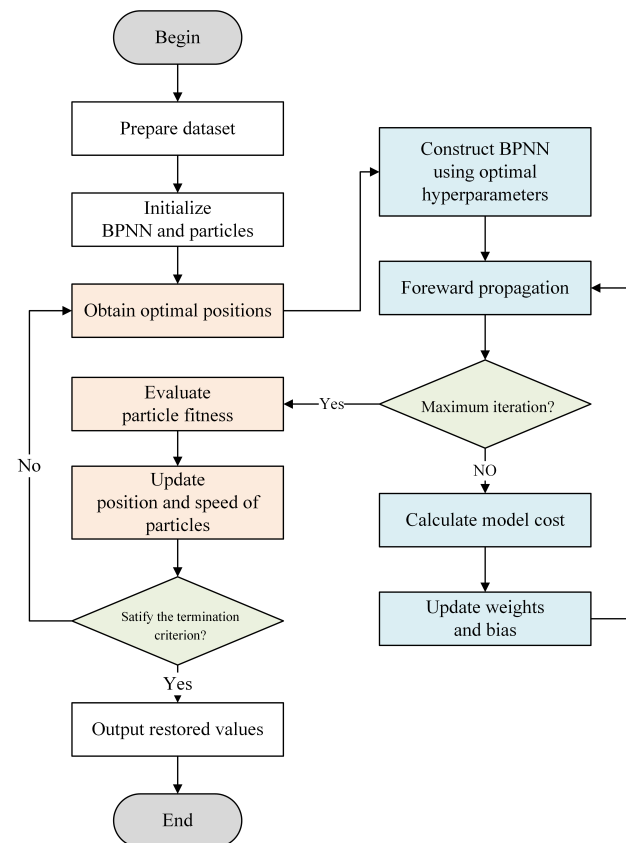


Figure 2. Flowchart of fault restoration for six-axis F/T sensors based on PSO-BPNN.

4. Experiments

A six-axis F/T sensor produced by the Institute of Intelligent Machines (IIM), Chinese Academy of Sciences (CAS), is utilized for experiments in this section. A unique double E-shape elastic body and several strain gauges constitute this six-axis F/T sensor [21]. The prototype and measurement circuit of the six-axis F/T sensor is shown in Figure 3, and the rated ranges of all dimensions are shown in Table 2. The experiment was programmed using MATLAB software and conducted on a PC which contains 3.6 GHz CPU and NVIDIA RTX 3070 GPU.

Table 2. Rated range of each dimension.

Dimensions	Ranges	Units
F_x	−1000 to 1000	N
F_y	−1000 to 1000	N
F_z	−1000 to 1000	N
M_x	−30 to 30	N·m
M_y	−30 to 30	N·m
M_z	−30 to 30	N·m

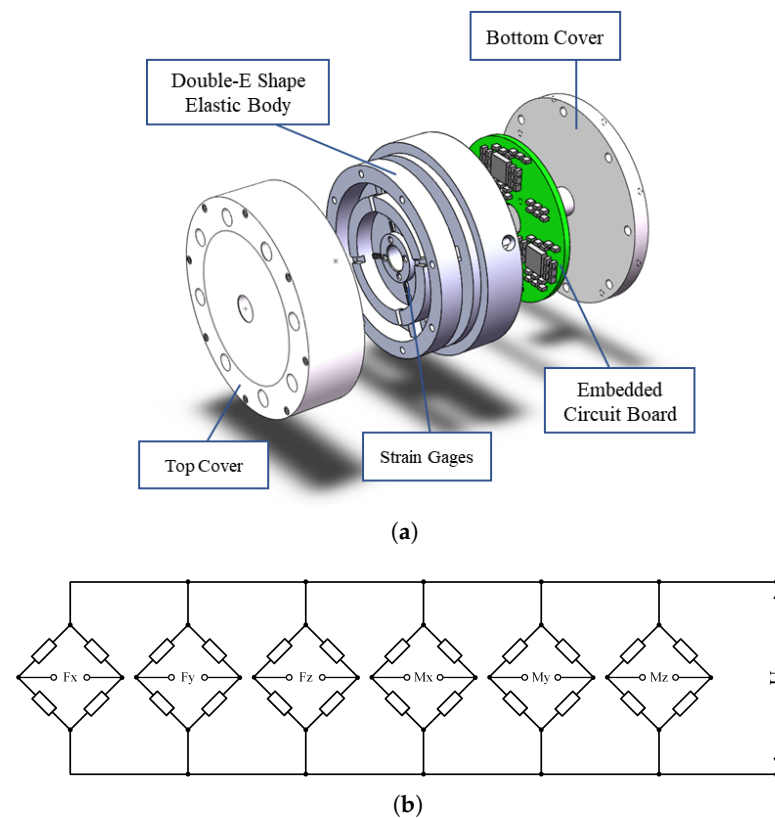


Figure 3. A novel six-axis F/T sensors designed by IIM, CAS. (a) The six-axis F/T sensor prototype. (b) The measurement circuit consists of six Wheatstone bridges.

4.1. Coupling Experiments

A coupling experiment was conducted to study the coupling effects among dimensions on the F/T sensor, and the experimental data were gathered for sensor calibration and model training. The coupling experiment is similar to the sensor calibration, which investigates the coupling output of all dimensions while the single dimension is loaded. Assuming $U = [u_1, u_2, \dots, u_6]^T$ represents the vector of voltage outputs of six dimensions, and $L = [Fx, Fy, Fz, Mx, My, Mz]$ indicates the measured load vector.

The main procedures of coupling and calibrating for six-axis F/T sensors apply a series of specific loads, which increase from minimum to maximum rated ranges with a certain step. These procedures were repeated three times in this coupling experiment, and the load vector L and voltage vector U were recorded at sample points for processing. The configuration of sample points is shown in Table 3, and the temperature and humidity of environment are 25 °C and 60%, respectively.

Table 3. The calibration experiment configuration.

Dimensions	Load Points	Units
F_x	0, ± 200 , ± 400 , ± 600 , ± 800 , ± 1000	N
F_y	0, ± 200 , ± 400 , ± 600 , ± 800 , ± 1000	N
F_z	0, ± 200 , ± 400 , ± 600 , ± 800 , ± 1000	N
M_x	0, ± 10 , ± 20 , ± 30	N·m
M_y	0, ± 10 , ± 20 , ± 30	N·m
M_z	0, ± 10 , ± 20 , ± 30 , ± 40 , ± 50	N·m

4.1.1. Calibration

Calibrating for F/T sensors aims to build the transfer expression between loads and output voltages, and it can be formulated by the following equation:

$$L = W \times U + B, \quad (7)$$

where W is a calibration matrix, which consists of weights between loads and output voltages, and B is bias vector.

Least Square (LS) algorithm is commonly utilized to calculate the calibration matrix W and bias vector B . W, B obtained from LS in this calibration experiment are shown as follows:

$$W = \begin{pmatrix} 0.870 & -0.001 & 0.004 & 0.009 & -0.710 & -0.004 \\ -0.009 & 0.837 & -0.03 & 0.687 & 0.005 & -0.009 \\ 0.004 & 0.003 & 0.280 & -0.003 & 0.005 & 0.003 \\ -0.019 & -0.304 & 0.025 & 0.597 & 0.014 & -0.005 \\ 0.250 & -0.007 & 0.005 & -0.011 & 0.638 & 0.002 \\ -0.101 & -0.025 & 0.006 & -0.011 & 0.087 & 1.401 \end{pmatrix}, \quad (8)$$

$$B = [-0.925, -75.016, -49.692, 5.463, 164.978, -31.988]^T. \quad (9)$$

The transfer expression is obtained by substituting W and B into (7).

4.1.2. Coupling

Coupling effects can be illustrated by taking the load on one dimension as an independent variable, while the output voltages of all dimensions are dependent variables. As shown in Figure 4, coupling effects are observed between some dimension pairs. For example, the output voltages of dimension My have positive correlation with the loads on dimension Fx , the output voltages of dimension Fy have a negative correlation with the loads on dimension Mx . However, the loads exerted on dimension Fz and My cause minor effects on other dimensions, which means the structures of dimension Fz and My are somewhat independent of other dimensions.

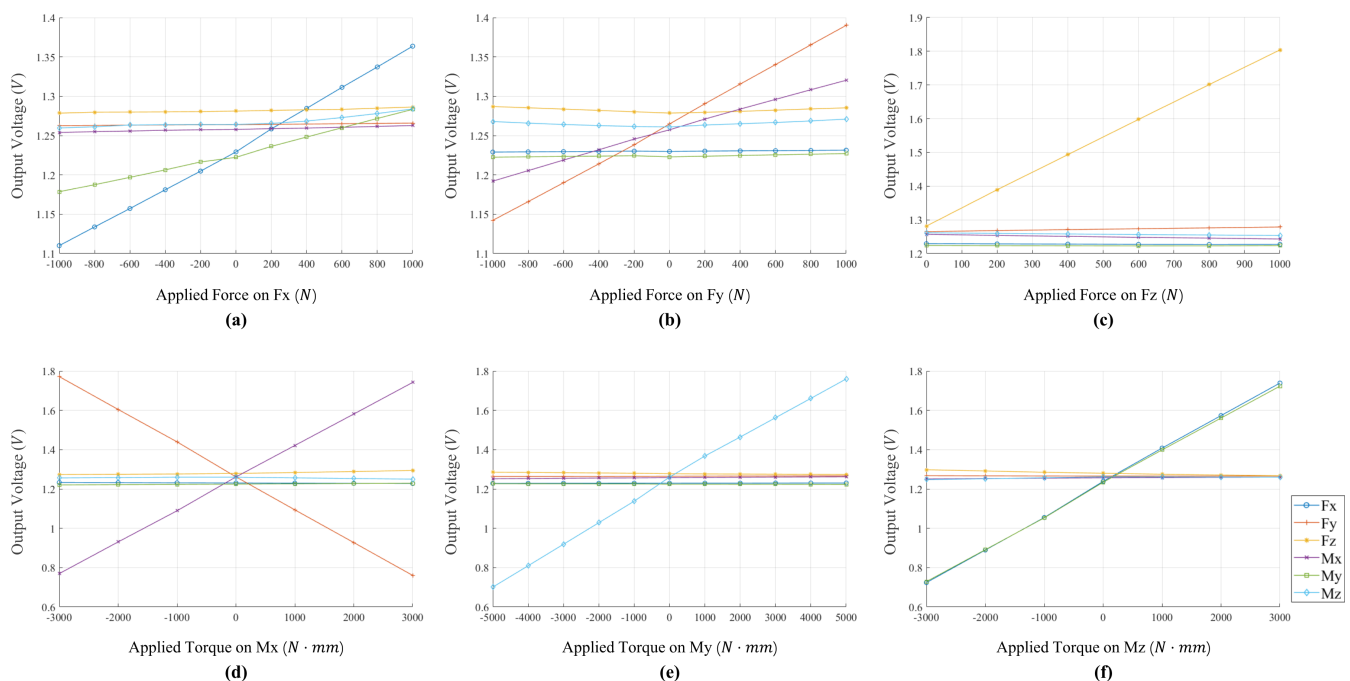


Figure 4. Voltage outputs of all dimensions when exerting loads on a single dimension. Not only does the loaded dimension have the output, other dimensions also have corresponding coupling outputs.

Based on the above analysis, coupling effects exist among dimensions of the six-axis F/T sensor, and we can use this phenomenon to restore one and two dimensions from damages. Besides, the effects of restoration are relative to the coupling level of dimensions.

4.2. Model Training

The data gathered in the coupling experiment contains the output of six dimensions. Since any dimension can be damaged in work, we assumed that one dimension was damaged in scheme 1 and restored it, then repeated this progress for all dimensions in turn. Similarly, we assumed that two dimensions were broken in scheme 2 and restored them by turns. Thus, there will be six combinations in scheme 1 and fifteen combinations in scheme 2.

To evaluate the effect of the proposed method, we take a basic BP network as a comparison, which consists of an input layer, an output layer, and two hidden layers.

4.3. Experiment Result and Analysis

After the restoration model is well trained, restoration experiments are carried out to evaluate the performance of the proposed method. Figure 5 shows how the trained model works in the measurement process. The testing errors indicated by the MSE of scheme 1 and scheme 2 are shown in Table 4 and Table 5, respectively. Besides, the convergence curve of training for dimension F_z is taken for a representative, which is shown in Figure 6.

Except for the MSE, we calculated the quote errors (QE) between the measurement outputs of the normal six-axis F/T sensor and restored one. The QE divides the full scale of the factor, which can make it easier to compare the performance of dimensions whose rated ranges are different. The quote errors can be calculate using (10).

$$QE(\hat{\mathbf{y}}, \mathbf{y}) = \text{average}_{y_i \in D} \left(\frac{|\hat{y}_i - y_i|}{y_{FS}} \right), \quad (10)$$

where $\hat{\mathbf{y}}$ and $\mathbf{y}$ indicate the restored measurement vector and original measurement vector, respectively. D is the training set or testing set, and y_{FS} is the measurement range of the corresponding dimension.

The quote errors of scheme 1 and scheme 2 in the training and testing sets are shown in Table 6 and Table 7, respectively. The measurement ranges are listed in Table 2.

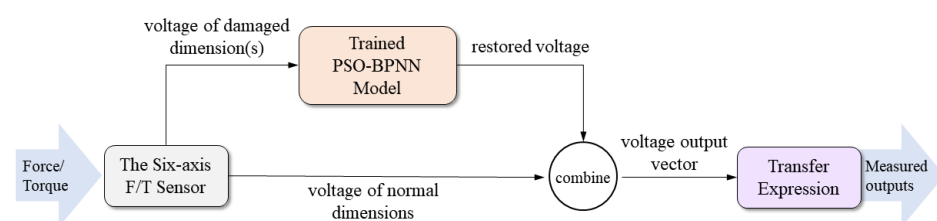


Figure 5. Workflow of trained PSO-BPNN in the measurement process.

Table 4. Testing error of restoring one dimension.

Dimensions	PSO-BPNN	Std-BPNN
F_x	1.066×10^{-5}	5.518×10^{-5}
F_y	6.888×10^{-7}	5.711×10^{-5}
F_z	1.914×10^{-6}	1.193×10^{-6}
M_x	4.147×10^{-7}	8.711×10^{-7}
M_y	7.432×10^{-7}	8.351×10^{-7}
M_z	5.553×10^{-5}	9.361×10^{-5}

Table 5. Testing error of restoring two dimensions.

Dimensions	PSO-BPNN	Std-BPNN
Fx, Fy	2.778×10^{-5}	2.428×10^{-5}
Fx, Fz	3.518×10^{-7}	2.701×10^{-6}
Fx, Mx	4.200×10^{-6}	2.153×10^{-6}
Fx, My	2.861×10^{-5}	1.353×10^{-4}
Fx, Mz	4.887×10^{-5}	3.928×10^{-5}
Fy, Fz	5.857×10^{-7}	5.812×10^{-6}
Fy, Mx	1.218×10^{-4}	2.218×10^{-4}
Fy, My	1.831×10^{-5}	9.515×10^{-6}
Fy, Mz	5.245×10^{-5}	5.429×10^{-5}
Fz, Mx	2.723×10^{-5}	1.500×10^{-4}
Fz, My	2.203×10^{-6}	2.885×10^{-6}
Fz, Mz	1.187×10^{-4}	1.287×10^{-4}
Mx, My	2.194×10^{-6}	1.351×10^{-5}
Mx, Mz	9.272×10^{-6}	2.054×10^{-5}
My, Mz	5.059×10^{-5}	6.253×10^{-5}

Table 6. Quote errors of restoring one dimension.

Dimensions	Training Set	Testing Set
Fx	7.524×10^{-5}	8.192×10^{-5}
Fy	5.419×10^{-4}	5.863×10^{-4}
Fz	1.193×10^{-4}	1.454×10^{-4}
Mx	3.900×10^{-3}	7.000×10^{-3}
My	7.100×10^{-3}	8.700×10^{-3}
Mz	4.020×10^{-2}	3.600×10^{-2}

Table 7. Quote errors of restoring two dimensions.

Dimensions	Training Set	Testing Set
Fx, Fy	2.254×10^{-4}	2.885×10^{-4}
Fx, Fz	1.690×10^{-4}	1.993×10^{-4}
Fx, Mx	3.700×10^{-3}	6.300×10^{-3}
Fx, My	6.300×10^{-3}	7.800×10^{-3}
Fx, Mz	1.690×10^{-2}	1.830×10^{-2}
Fy, Fz	1.498×10^{-4}	2.054×10^{-4}
Fy, Mx	4.400×10^{-3}	1.890×10^{-2}
Fy, My	2.500×10^{-3}	3.300×10^{-3}
Fy, Mz	1.800×10^{-2}	1.890×10^{-2}
Fz, Mx	4.300×10^{-3}	5.900×10^{-3}
Fz, My	2.300×10^{-3}	2.600×10^{-3}
Fz, Mz	1.560×10^{-2}	2.150×10^{-2}
Mx, My	6.100×10^{-3}	8.000×10^{-3}
Mx, Mz	2.420×10^{-2}	2.880×10^{-2}
My, Mz	1.760×10^{-2}	1.800×10^{-2}

As the experiment results show in Tables 4 and 5, the maximum and average testing errors of one dimension restoration are 5.553×10^{-5} and 1.1693×10^{-5} . The maximum and average testing MSEs of two dimensions restoration are 1.218×10^{-4} and 3.4205×10^{-5} . The testing MSEs show that the PSO-BPNN can accurately represent the correlations between dimensions in most conditions, and it is possible to restore the output of the damaged dimension with the remaining dimensions.

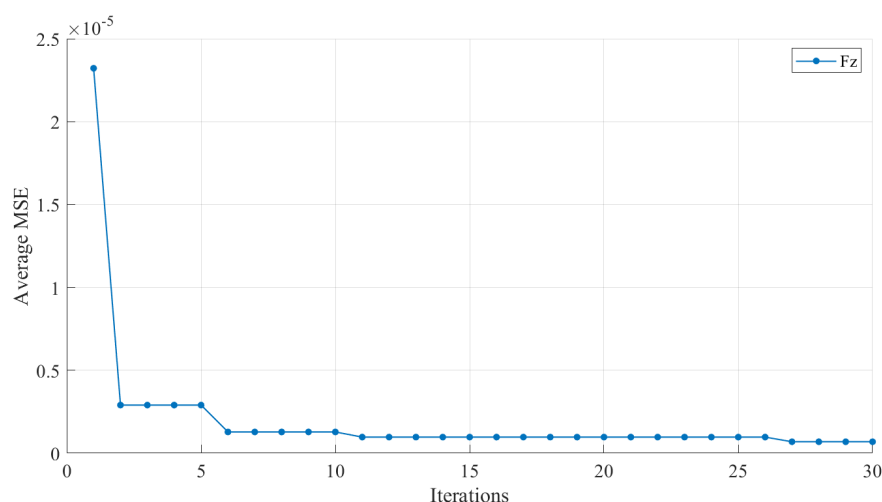


Figure 6. The convergence curve of training for dimension Fz .

The maximum and average quote errors of one dimension are 3.600×10^{-2} and 8.800×10^{-3} . The maximum and average quote errors of two dimensions are 2.42×10^{-2} and 8.200×10^{-3} . The quote errors in testing show that the measurement output of the restored dimensions can reach the original precision class, which means the proposed restoration method is feasible and satisfactory.

In addition, the average fitness of BPNN during training Fz is taken to illustrate the convergence characteristic of PSO, and the convergence curve is shown in Figure 6. As can be seen from Figure 6, the fitness continued to decrease before reaching the target iterations, which means that the PSO algorithm has good global search ability and is not easy to trap in local optima.

5. Conclusions

Multi-axis force/torque sensors are commonly deployed in robotics and industry. Due to the bad ambient conditions, multi-axis F/T sensors will be occasionally damaged, which is fatal to the control system. If one or two dimensions get damaged, it is costly and troublesome to replace the entire F/T sensor. Hence, fault restoration is meaningful and necessary for multi-axis F/T sensors.

Back propagation neural networks are popular for their simpleness and good performance. In this work, we adopted BPNN to represent the correlation between dimensions and build restoration models for damaged dimensions. Hyperparameters are essential for BPNN, and searching for good hyperparameters is low effective. The PSO algorithm is utilized to determine the optimal structure for BPNN in the proposed model. Moreover, a coupling experiment was conducted to assess the coupling effects of six-axis F/T sensors and provide data for model training.

The results of fault restoration experiments illustrated that the PSO-BPNN is feasible and suitable for fault restoration, and the PSO algorithm can provide optimal hyperparameters to improve the performance of BPNN. Multi-axis F/T sensors, which embed the fault restoration method, are more reliable and robust when working in the bad environment.

The proposed method performs well in fault restoration, but the model does not take into account the indirect coupling effect between dimensions. In future research, we will explore the inner mechanism of coupling effects and detect damage to multi-axis F/T sensors.

Author Contributions: X.L. (Xuhao Li) and H.C. conceived this research and designed the proposed model; Y.S. devised experiments; X.L. (Xuhao Li) programmed the algorithms; X.L. (Xuhao Li) and X.L. (Xiaohui Li) conducted experiments; X.L. (Xuhao Li) prepared this manuscript; L.G. revised the language expression and technical coherence. X.L. (Xiaohui Li) revised the technical description and text presentation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Key research projects supported by the National Natural Science Foundation of China, grant number: 92067205; in part by the Major Science and Technology Project of Anhui Province, grant number: 202103a05020022; in part by the Key Research and Development Project of Anhui Province, grant number: 2022a05020035; in part by the Strategic Priority Research Program of the Chinese Academy of Sciences, grant number: XDA22040303.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Subad, R.A.I.; Saikot, M.M.H.; Park, K. Soft Multi-Directional Force Sensor for Underwater Robotic Application. *Sensors* **2022**, *22*, 3850. [\[CrossRef\]](#) [\[PubMed\]](#)
- Meng, D.S.; She, Y.; Xu, W.F.; Lu, W.N.; Liang, B. Dynamic modeling and vibration characteristics analysis of flexible-link and flexible-joint space manipulator. *Multibody Syst. Dyn.* **2018**, *43*, 321–347. [\[CrossRef\]](#)
- Lin, S.; Zhang, L.X.; Zhou, S.J.; Zhang, C.F. Multi-dimensional dynamic measurement uncertainty analysis for the six-axis force sensor used in docking system. *Trans. Inst. Meas. Control.* **2018**, *40*, 3040–3048. [\[CrossRef\]](#)
- Ma, X.; Han, Z.; Zou, Y.; Ding, J. Review and Assessment of Spacecraft Mechanical Environment Analysis and Specification Determination. *J. Astronaut.* **2012**, *33*, 1–12.
- Song, A.; Wu, J.; Qin, G.; Huang, W. A novel self-decoupled four degree-of-freedom wrist force/torque sensor. *Measurement* **2007**, *40*, 883–891. [\[CrossRef\]](#)
- Liang, Q.; Long, J.; Coppola, G.; Zhang, D.; Sun, W. Novel decoupling algorithm based on parallel voltage extreme learning machine (PV-ELM) for six-axis F/M sensors. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 303–314. [\[CrossRef\]](#)
- Niu, Z.; Zhao, T.; Zhao, Y.; Hu, Q.; Ding, S. Design and Analysis of the Measurement Characteristics of a Bidirectional-Decoupling Over-Constrained Six-Dimensional Parallel-Mechanism Force Sensor. *Sensors* **2017**, *17*, 1985. [\[CrossRef\]](#) [\[PubMed\]](#)
- Lin, J.C.W.; Djenouri, Y.; Srivastava, G.; Fourier-Viger, P. Efficient evolutionary computation model of closed high-utility itemset mining. *Appl. Intell.* **2022**, *52*, 10604–10616. [\[CrossRef\]](#)
- Shao, Y.A.; Lin, J.C.W.; Srivastava, G.; Guo, D.D.; Zhang, H.C.; Yi, H.; Jolfaei, A. Multi-Objective Neural Evolutionary Algorithm for Combinatorial Optimization Problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ren, Q.; Zhang, N.; Song, Q.; Li, J. Short-Term PV Power Forecasting Based on BP Network Optimized by Logistic Improved ASO. In Proceedings of the 2021 6th International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 17–20 September 2021; IEEE: Shanghai, China, 2021; pp. 1135–1139. [\[CrossRef\]](#)
- Zou, J.; Liang, Q.; Xu, X.; Wang, Q. Transformer Fault Diagnosis Based on Optimized CPSO-BP Neural Network. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *781*, 042047. [\[CrossRef\]](#)
- Qie, X.H.; Kang, C.F.; Zong, G.C.; Chen, S.J. Trajectory Planning and Simulation Study of Redundant Robotic Arm for Upper Limb Rehabilitation Based on Back Propagation Neural Network and Genetic Algorithm. *Sensors* **2022**, *22*, 4071. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yu, H.H.; Wang, T. A Method for Real-Time Fault Detection of Liquid Rocket Engine Based on Adaptive Genetic Algorithm Optimizing Back Propagation Neural Network. *Sensors* **2021**, *21*, 5026. [\[CrossRef\]](#) [\[PubMed\]](#)
- Lin, T.; Fangjun, Q.; Kali, F.; Chunfu, H.; Zhi, W. Segmentation compensation method for FOG temperature error based on particle swarm optimization algorithm. *J. Chin. Inert. Technol.* **2019**, *27*, 505–509.
- Han, X.-Y.; He, P.; Pan, G.; Liu, Y.; Zhang, W. Temperature Compensation of Pressure Sensor Based on PSO-SVR. *Instrum. Tech. Sens.* **2018**, *8*, 9–12.
- Xie, H.; Zhang, L.; Lim, C.P.; Yu, Y.H.; Liu, H. Feature Selection Using Enhanced Particle Swarm Optimisation for Classification Models. *Sensors* **2021**, *21*, 1816. [\[CrossRef\]](#) [\[PubMed\]](#)
- Tan, X.P.; Su, S.J.; Zuo, Z.; Guo, X.J.; Sun, X.Y. Intrusion Detection of UAVs Based on the Deep Belief Network Optimized by PSO. *Sensors* **2019**, *19*, 5529. [\[CrossRef\]](#)
- Lin, J.C.W.; Lv, Q.; Yu, D.; Srivastava, G.; Chen, C.H. Adaptive particle swarm optimization model for resource leveling. *Evol. Syst.* **2022**. [\[CrossRef\]](#)
- Lin, J.C.W.; Zhang, Y.Y.; Zhang, B.B.; Fournier-Viger, P.; Djenouri, Y. Hiding sensitive itemsets with multiple objective optimization. *Soft Comput.* **2019**, *23*, 12779–12797. [\[CrossRef\]](#)
- Lin, J.C.W.; Srivastava, G.; Zhang, Y.Y.; Djenouri, Y.; Aloqaily, M. Privacy-Preserving Multiobjective Sanitization Model in 6G IoT Environments. *IEEE Internet Things J.* **2021**, *8*, 5340–5349. [\[CrossRef\]](#)
- Sun, Y.; Yu, Y.; Cao, H.; Shuang, F.; Gao, L.; Ge, Y. Research on Joint Torque Sensor for Space Manipulator Based on Redundant Measurement. *Chin. J. Sens. Actuators* **2018**, *31*, 1621–1627.