Using Sensors Data and Emissions Information to Diagnose Engine’s Faults

Chunli XIE1, Yuchao Wang1, John MacIntyre2, Muhammad Sheikh2, Mustafa Elkady3

1 Traffic college, Northeast Forestry University, No.26 Hexing road Xiangfang district, Harbin city, Heilongjiang province, 150040, China, E-mail:XCL08@126.com

2 Faculty of Engineering and Advanced Manufacturing, University of Sunderland Sunderland, SR6 0DD, UK

3 Department of Mechanical Engineering, School of Engineering, Lebanese International University, Beirut, Lebanon & Department of Automotive Engineering, Faculty of Engineering, Ain Shams University, Cairo, Egypt

Abstract

This paper proposes using engine’s sensors data flow and exhaust emissions information to diagnose engine’s faults, enhancing the accuracy of fault diagnosis. Engine fault diagnosis model is built using both this information and the mature BP neural network and genetic algorithms. In order to verify the method, we build a test platform, which includes South Korea Hyundai fault test vehicle and X-431 diagnosis instrument and AUTO5-1 exhaust gas analyzer and computer. The diagnostic accuracy rate can reach 98.33%, which is higher than using sensors data flow or the exhaust emissions information alone.

Keywords: Fault diagnosis; Neural network; Sensors data flow; Emissions information; Engine

1. Introduction

The engine is the main source of power and an important component of the automobile and its performance directly affects vehicle’s safety and reliability. Due to their complex structure and heavy working conditions, the engine fault generally constitutes around 40% of the failures of the entire vehicle1. Hence, it is of a great importance to developing new approaches for quick and accurate condition monitoring and fault diagnosis of the engines.

In the past few decades, research on the engine fault diagnosis (EFD) technology has become an active area in the field of vehicle engineering2-4. Kowalski5 presented the concept of a multi-dimensional marine engine diagnostic tool. The dimensions of the tool were diagnostic signals, which form a vector in affine space. The distance of the resulting vector from reference vectors for considered technical states of the engine was the result of the diagnosis. Moreover, diagnostic signals, derived from the composition of the exhaust gas, were also considered. Kowalski also chose the nitric oxide, carbon oxide, carbon dioxide and oxygen contents in the...
exhaust gas and the temperatures behind all engine cylinders of the marine engine as diagnostic signals. Wu et al.\textsuperscript{6} described an internal combustion engine fault diagnosis system using the manifold pressure of the intake system. To verify the effect of their proposed system for identification, both the radial basis function network (RBFN) and generalized regression neural network (GRNN) were used and compared in their study. The experimental results indicated the proposed system using manifold pressure signal as data input was effective for engine fault diagnosis in the experimental engine platform. In addition, Chen and Randall\textsuperscript{7} used two methods to diagnose misfire fault. The first method was based on the torsional vibration signals of the crankshaft, while the second method was based on the angular acceleration signals (rotational motion) of the engine block. Following the signal processing of the experimental and simulation signals, the best features were selected as the inputs of ANN networks. The ANN systems were trained using only the simulated data and tested using real experimental cases, indicating that the simulation model can be used for a wider range of faults for which it can be considered valid. The final results have shown that the diagnostic system based on simulation can efficiently diagnose misfire, including location and severity.

In general, currently available methods for engine fault diagnosis are mainly classified into model-based, knowledge-based and data-driven methods\textsuperscript{8,9} according to Prof. PM Frank from Germany who is an international fault diagnosis authority.\textsuperscript{28} Model-based diagnostics threshold are very accurate, but it is very time-consuming and labour-demanding to identify the appropriate values of the model parameters. Thus, a model-based method is too expensive to comprehensively apply in practice. Moreover, due to the different natures of faults and the modelling uncertainty, no single model-based approach can diagnose all the faults. On the other hand, knowledge-based methods have the capability of handling the qualitative symptom descriptions\textsuperscript{9,10}. Lastly, the data-driven methods were widely used for engine fault diagnosis. The main advantage of such methods is that they do not require much (or none at all) expertise of the operator/designer since they are mainly based on the data collected from the process, which can be historical (off-line) or real-time (on-line). This is a very important feature since data-driven methods can cope with the problem of data drift and other unpredicted disturbances.\textsuperscript{11}

Based on these methods’ merits and demerits, we propose in this paper using both sensors data flow and emissions information to diagnose the engine faults, which can give the comprehensive running information of engine and improve diagnostic accuracy. In this research, BP neural network and GABP tools are used to verify the method. Tests are done using Roll Brake Test Systems to realize fault diagnosis on-line during vehicle running at a different speed. In addition, the structure and work principle of the electronic control system is clearly presented. Engine fault diagnosis model is described. Experiments have been tested, which include the data sampling procedure and the application of the proposed model. The training process of the model and the results of diagnostic tests online are also analyzed.

The paper is organized as follows. Section 2 provides the research background, which includes electronic control system of the engine, BP neural network, Genetic Algorithms and GABP model. Section 3 presents the tests and experiments including experimental instruments and data acquisition. In Section 4, the algorithm model is trained using Matlab. Section 5 describes the test result of a simulation experiment. Finally, Section 6 presents the concluding remarks for this paper.

2. Background

2.1. Electronic Control System of Engine

The electronic control system of the engine is composed of sensors, actuators and engine control unit (ECU). The basic control loop is shown in Figure 1, which is Sensors detect the status and condition of the engine and send the parameters to the ECU. ECU is electronic control unit which controls a series of actuators of the internal combustion engine to ensure engine’s optimal performance. It does this by reading values from a multitude of sensors within the engine bay, then interpreting the data using multidimensional performance maps (called lookup tables), and adjusting the engine actuators according to the information. Before ECUs, air-fuel mixture, ignition timing, and idle speed were mechanically set and dynamically controlled by mechanical and pneumatic means\textsuperscript{12}. The components of sensors and actuators and basic function of ECU are shown in Figure 2.

![Fig. 1. The basic control loop of engine.](image-url)
so nodes of input and output are confirmed. Table 1 lists the BP neural network model parameters.

<table>
<thead>
<tr>
<th>Item name</th>
<th>Content in detail</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neurons node number</td>
<td>Input layer</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Hidden Layer</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Output layer</td>
<td>6</td>
</tr>
<tr>
<td>Network structure function</td>
<td>Input—Hidden layer transfer function</td>
<td>Purelin</td>
</tr>
<tr>
<td></td>
<td>Hidden—Output layer transfer function</td>
<td>Tansig</td>
</tr>
<tr>
<td>Training parameters</td>
<td>Training function</td>
<td>Trainlm</td>
</tr>
<tr>
<td></td>
<td>Training times</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>Training goal</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Study rate</td>
<td>0.01</td>
</tr>
</tbody>
</table>

**2.2. BP neural network**

Back propagation (BP) neural network algorithm is a multi-layer feed forward network trained based on an error back propagation algorithm and is one of the most widely applied neural network models. The BP network is used to learn and store a great deal of mapping relations of input-output model. In this network, there is no need to disclose in advance the mathematical equation that describes these mapping relations. Its learning rule is to adopt the steepest descent method in which the back propagation is used to regulate the weight value and threshold value of the network to achieve the minimum error sum of square. The BP neural network toolbox in Matlab is chosen to get the ideal fault diagnosis results in this study. In order to obtain the best BP neural network structure, a large number of simulation experiments were conducted. Considering the mean squared error and percent error, hidden layer neurons of BP model are confirmed using minimum error and fast calculation speed, so hidden layer neurons are 17. Hidden and output layer incentive functions are purelin and tansig respectively to adjust weight values. The Figure3 shows the two functions.

**2.3. Genetic Algorithms(GA)**

Genetic Algorithms (GA) are probabilistic search algorithms that emulate the biological evolution of a population, by applying genetic operators that allow the recombination of the individuals, in order to strengthen their performance with regard to a quality function. The selection of fitness function directly influences the convergence speed of genetic algorithm and whether it can find the optimal solution. Individual fitness (F) is calculated by Equation 1.

$$ F_i = k \sum_{j=1}^{n} |y_j - O_j| $$  

Where $y_j$ is expectation output of node $i$, and $O_j$ is forecast output. $k$ is coefficient, which is a uniform random number rand between 0 and 1. $n$ is an individual number.

**2.4. GABP model**

Although the BP algorithm is successful, it has some disadvantages. Because the BP neural network has adopted the gradient method, the problems include a slow learning convergent velocity and easy convergence to local minima, which cannot be avoided. GA has been proven to effectively function with ANNs in determining the most compact architecture improving the estimation performance in a variety of problems. This paper chose GA to optimize BP neural network to avoid BP algorithm convergence to local minima. The weights and thresholds of BP neural network are optimized using GA in training process, so called this...
algorithm is genetic algorithm BP neural network (GABP). First, we should confirm the original parameters of GA. The population size is set to 10, and the evolution number is 50 times, crossover probability is 0.4, mutation probability is 0.2. Second, confirm fitness function using Equation 1. In fitness proportionate individuals are selected with a probability that is directly proportional to their fitness values. Evaluate the probability, $P_i$, of selecting each individual in the population:

$$P_i = F_i / \sum_{j=1}^{N} F_j$$

(2)

Where $N$ is population size.

Genetic operators are used to evolving the next generation of individuals from the current population. Here we choose crossover and mutation. So, next step is crossover and mutation using the equation 3 and 4.

According to the probability $P_i$, randomly select two antibodies $a$ and $a_i$ from the population, then the crossover operation of two antibodies in j-bit can be described as follows:

$$a_{ij} = a_i (1 - b) + a_j b$$

$$a_{ij} = a_i (1 - b) + a_j b$$

(3)

Where $b$ is a variable randomly distributed from 0 to 1. For the selected antibody $a_i$, the mutation of $j$-gene can be calculated as follows:

$$a_{ij} = \begin{cases} a_i + (a_j - a_{min}) \times f(g) & r > 0.5 \\ a_i + (a_{max} - a_j) \times f(g) & r \leq 0.5 \end{cases}$$

(4)

Where $a_{max}$ and $a_{min}$ are the low and upper bounds of the gene value respectively. $r$ is a variable randomly distributed from 0 to 1. Mutation probability is basically a measure of the likeness that random elements of your chromosome will be flipped into something else. For example, if your chromosome is encoded as a binary string of length 100 if you have 1% mutation probability it means that 1 out of your 100 bits (on average) picked at random will be flipped. So $f(g)$ is the mutation probability defined as follows:

$$f(g) = r_2 \left(1 - g / G_{\text{max}} \right)^2$$

(5)

Where $g$ is the current iterative number, $G_{\text{max}}$ is the maximal iterative number, $r_2$ is a variable randomly distributed from 0 to 1.

3. Tests and Experiments

The detection of faults in their early stages is beneficial for the avoidance of larger and more severe faults. Fault detection and diagnosis (FDD) methods are used to monitor the system and identify faults occurring and type and location. If a fault is correctly detected and diagnosed, corrective measures can be applied to repair the fault and reduce any further damage to the system. Many researchers carried out intensive research on FDD. Authors used the energy balance equations which balance the relation among pressure, mass flow rate and power at various locations to diagnose the faults of the space shuttle main engine. Authors used vibration signals to diagnose the faults of diesel internal combustion engine’s valve trains. Meanwhile, changes of exhaust emissions content reflect engine performance. Exhaust emissions compositions of engine mainly include HC, NOx, CO, CO2, O2, water vapour etc. They must change in some range due to the change of operating status at different conditions, various mechanical and electronic faults. So exhaust emissions can reflect engine faults. On the other hand, engine’s sensors data can also reflect engine’s condition and faults. In order to get the comprehensive information of engine to help us diagnose the faults accurately, sensors data and emissions information are all used to diagnose the faults of the engine, which are shown in Table 2. Table 2 presents data about the engine, which are called faults feature factors as inputs of the BP neural network and faults type, which are the output of the BP neural network.

<table>
<thead>
<tr>
<th>Input item</th>
<th>Feature factors(unit)</th>
<th>Output item</th>
<th>Faults type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_1</td>
<td>CO(10^{-2}\text{cc})</td>
<td>y_1</td>
<td>No fault</td>
</tr>
<tr>
<td>x_2</td>
<td>CO_{2}(10^{-2}\text{cc})</td>
<td>y_2</td>
<td>fuel injector fault</td>
</tr>
<tr>
<td>x_3</td>
<td>O_{2}(10^{-2}\text{cc})</td>
<td>y_1</td>
<td>Ignition coil fault</td>
</tr>
<tr>
<td>x_4</td>
<td>HC(10^{-6}\text{cc})</td>
<td>y_4</td>
<td>Throttle position sensor fault</td>
</tr>
<tr>
<td>x_5</td>
<td>NO_{x}(10^{-6}\text{cc})</td>
<td>y_4</td>
<td>air flow sensor fault</td>
</tr>
<tr>
<td>x_6</td>
<td>Crankshaft rolling speed(/min)</td>
<td>y_6</td>
<td>Oxygen sensor fault</td>
</tr>
<tr>
<td>x_7</td>
<td>Throttle position(mv)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_8</td>
<td>air flow sensor (HC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_9</td>
<td>Oxygen sensor(mv)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_{10}</td>
<td>Intake air temperature sensor(\degree C)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.1. Experimental instruments

A test platform is built to verify the proposed method. The platform includes South Korea Hyundai fault test vehicle, X-431 diagnosis instrument and AUTO5-1 exhaust gas analyzer as well as a computer. Figure 4 shows the diagnosis test vehicle, which has eight faults setting control modules. Their names are engine(ECU) faults, ABS(antilock brake system) faults, TCS(Traction Control System), FATC(Full Automatic Air Conditioning), TCU(Transmission Control Unit), ECS(Engine Control System), ETACS(Electronic Time Alarm Control System), Air bag. In this paper, the focus will only be on the engine faults setting module because it has a more direct connection with emissions and system can get the sensors data easily. The engine can be inserted into different sensors faults and actuators faults, then diagnose the inserted faults using the proposed method and get the diagnosis results. Comparing diagnosis results with the inserted faults, which can test the diagnostic system.

The X-431 diagnosis instrument is shown in Figure 5, which can get sensors data flow. Figure 6 shows the AUTO5-1 exhausts gas analyzer, which can get exhaust information of engine. The vehicle only runs at the idle condition, because we need to connect the experiment instruments with the vehicle and get the information. In order to simulate real road conditions, we use Burke Porter’s 3600 roll brake test system. Figure 7 shows the vehicle running on the platform. We can get the vehicle’s information on different speed and diagnose the faults real-time (on-line).

Fig. 7. Vehicle running on the 3600 roll brake test systems.

3.2. Experiment data acquisition

It should be noted that the use of artificial neural networks requires the collection of large amounts of data sets. They require obtaining more than one hundred sets of data for each operating condition, which are used to teach the network. We use the diagnostic port MITSUBISHI/HYUNDAI-12+16. The 16 frame port is standard RS232. Sensors data flow can be got using X-431 PCLINK data communication soft, which can realize the following functions shown in table3. Figure 8 represents the software main interface.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor and alarm</td>
<td>monitor the engine and give the alarm if the engine has abnormal.</td>
</tr>
<tr>
<td>Diagnose</td>
<td>diagnose the faults of the engine.</td>
</tr>
<tr>
<td>Restore and check</td>
<td>restore and check data of engine such as speed, coolant temperature and so on.</td>
</tr>
<tr>
<td>Reset system</td>
<td>reset system such as deleting fault code, recovering system etc.</td>
</tr>
</tbody>
</table>

The test vehicle and all the instruments can work normally before doing experiments in order to get the right information about the engine. Faults’ code must be cleared on the X-431 diagnostic instrument and the AUTO5-1 exhaust gas analyzer must be reset. All of the fault set modules of test vehicle must keep normal. We must keep the X-431 and the AUTO5-1 to work...
together, which can ensure we get the information on same time. Firstly, the vehicle runs at idle speed, and then it keeps running at this condition and gets the normal data. Parts of normal and different faults data engine at idle speed are shown in Table 4. Secondly, vehicle will accelerate until 1800r/min, and then the vehicle keeps running at this speed and gets the normal data. Parts of normal data and different faults data engine at 1800r/min are shown in Table 5. We performed these steps 20 times and got 120 groups data.

Table 4. Training data engine at idle speed.

<table>
<thead>
<tr>
<th>Fault type</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>0.01</td>
<td>9.1</td>
<td>4.43</td>
<td>1</td>
<td>431</td>
</tr>
<tr>
<td>y2</td>
<td>0.07</td>
<td>7.8</td>
<td>10.82</td>
<td>1532</td>
<td>91</td>
</tr>
<tr>
<td>y3</td>
<td>0.08</td>
<td>7.8</td>
<td>10.81</td>
<td>1534</td>
<td>95</td>
</tr>
<tr>
<td>y4</td>
<td>0.01</td>
<td>10</td>
<td>9.32</td>
<td>1</td>
<td>984</td>
</tr>
<tr>
<td>y5</td>
<td>0.02</td>
<td>9.3</td>
<td>9.7</td>
<td>1</td>
<td>1008</td>
</tr>
<tr>
<td>y6</td>
<td>0.17</td>
<td>8.9</td>
<td>8.7</td>
<td>1344</td>
<td>38</td>
</tr>
<tr>
<td>y7</td>
<td>0.16</td>
<td>9.2</td>
<td>8.57</td>
<td>1344</td>
<td>44</td>
</tr>
<tr>
<td>y8</td>
<td>0.11</td>
<td>10.3</td>
<td>4.98</td>
<td>1543</td>
<td>89</td>
</tr>
<tr>
<td>y9</td>
<td>0.12</td>
<td>10.4</td>
<td>4.93</td>
<td>1543</td>
<td>92</td>
</tr>
<tr>
<td>y10</td>
<td>0.06</td>
<td>8.7</td>
<td>8.76</td>
<td>883</td>
<td>30</td>
</tr>
<tr>
<td>y11</td>
<td>0.07</td>
<td>9.1</td>
<td>8.52</td>
<td>915</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 4, continued:

<table>
<thead>
<tr>
<th>Fault type</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
<th>x10</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>718.8</td>
<td>507.8</td>
<td>23.1</td>
<td>39.1</td>
<td>17</td>
</tr>
<tr>
<td>y2</td>
<td>687.5</td>
<td>507.8</td>
<td>23.1</td>
<td>39.1</td>
<td>17</td>
</tr>
<tr>
<td>y3</td>
<td>718.8</td>
<td>507.8</td>
<td>31.2</td>
<td>58.6</td>
<td>16</td>
</tr>
<tr>
<td>y4</td>
<td>687.5</td>
<td>507.8</td>
<td>31.2</td>
<td>58.6</td>
<td>16</td>
</tr>
<tr>
<td>y5</td>
<td>718.8</td>
<td>507.8</td>
<td>43.8</td>
<td>39.1</td>
<td>17</td>
</tr>
<tr>
<td>y6</td>
<td>687.5</td>
<td>507.8</td>
<td>43.8</td>
<td>39.1</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 5. Training data of engine accelerate to 1800r/min

<table>
<thead>
<tr>
<th>Fault type</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>0</td>
<td>13.5</td>
<td>2.99</td>
<td>46</td>
<td>1397</td>
</tr>
<tr>
<td>y2</td>
<td>0</td>
<td>13.5</td>
<td>2.93</td>
<td>47</td>
<td>1590</td>
</tr>
<tr>
<td>y3</td>
<td>0</td>
<td>11.6</td>
<td>3.78</td>
<td>1</td>
<td>1115</td>
</tr>
<tr>
<td>y4</td>
<td>0</td>
<td>11.2</td>
<td>3.52</td>
<td>1</td>
<td>1487</td>
</tr>
<tr>
<td>y5</td>
<td>0.04</td>
<td>9.3</td>
<td>8.63</td>
<td>1</td>
<td>1228</td>
</tr>
<tr>
<td>y6</td>
<td>0.06</td>
<td>9.4</td>
<td>8.59</td>
<td>1</td>
<td>1215</td>
</tr>
<tr>
<td>y7</td>
<td>0.01</td>
<td>13.8</td>
<td>2.28</td>
<td>1</td>
<td>1412</td>
</tr>
<tr>
<td>y8</td>
<td>0.01</td>
<td>13.8</td>
<td>2.52</td>
<td>1</td>
<td>1027</td>
</tr>
<tr>
<td>y9</td>
<td>0.79</td>
<td>14.9</td>
<td>0.4</td>
<td>322</td>
<td>143</td>
</tr>
<tr>
<td>y10</td>
<td>0.84</td>
<td>14.9</td>
<td>0.4</td>
<td>356</td>
<td>133</td>
</tr>
<tr>
<td>y11</td>
<td>0.01</td>
<td>13.9</td>
<td>2.39</td>
<td>1</td>
<td>1675</td>
</tr>
<tr>
<td>y12</td>
<td>0.01</td>
<td>13.5</td>
<td>2.58</td>
<td>1</td>
<td>1916</td>
</tr>
</tbody>
</table>

Table 4, continued:

<table>
<thead>
<tr>
<th>Fault type</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
<th>x10</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>1875</td>
<td>957</td>
<td>156.2</td>
<td>58.6</td>
<td>54</td>
</tr>
<tr>
<td>y2</td>
<td>1812.5</td>
<td>937.4</td>
<td>150</td>
<td>58.6</td>
<td>54</td>
</tr>
<tr>
<td>y3</td>
<td>1875</td>
<td>1074.2</td>
<td>187.5</td>
<td>39.1</td>
<td>32</td>
</tr>
<tr>
<td>y4</td>
<td>1812.5</td>
<td>1054.6</td>
<td>181.2</td>
<td>39.1</td>
<td>32</td>
</tr>
<tr>
<td>y5</td>
<td>1875</td>
<td>2382.7</td>
<td>368.8</td>
<td>39.1</td>
<td>0</td>
</tr>
<tr>
<td>y6</td>
<td>1812.5</td>
<td>2382.7</td>
<td>406.2</td>
<td>39.1</td>
<td>0</td>
</tr>
<tr>
<td>y7</td>
<td>1718.8</td>
<td>39.1</td>
<td>150</td>
<td>39.1</td>
<td>0</td>
</tr>
<tr>
<td>y8</td>
<td>1812.5</td>
<td>39.1</td>
<td>156.2</td>
<td>58.6</td>
<td>0</td>
</tr>
<tr>
<td>y9</td>
<td>1875</td>
<td>917.9</td>
<td>0</td>
<td>917.9</td>
<td>54</td>
</tr>
<tr>
<td>y10</td>
<td>1812.5</td>
<td>937.4</td>
<td>0</td>
<td>937.4</td>
<td>54</td>
</tr>
<tr>
<td>y11</td>
<td>1781.2</td>
<td>937.4</td>
<td>156.2</td>
<td>39.1</td>
<td>32</td>
</tr>
<tr>
<td>y12</td>
<td>1718.8</td>
<td>839.8</td>
<td>156.2</td>
<td>19.5</td>
<td>32</td>
</tr>
</tbody>
</table>

4. Algorithm model training

4.1 BP Neural network training

BP neural network model is built using Matlab software and trained using the data. Figure 9 shows the training interface, which gives the structure of the neural network, algorithms parameters, calculates progress and
training plots. Figure 10 presents change curve of BP neural network training mean squared error with training epochs. From Figure 10(a) we can see the best validation performance is $1.49 \times 10^{-5}$ at epoch 13, which mean the neural network convergence speed is very fast. Figure11 shows the BP neural network training objective gradient curve, while Figure 12 shows the linear correlation-regression curve of a relative output of the BP neural network. The training curves of the engine are provided using different data at the idle speed and accelerate to 1800r/min respectively.

**4.2. GABP training**

The best individual fitness curve and best fitness were obtained by training (showing in Figure 12). The steps of convergence GABP neural network model is shown in Figure13.

![Image](image.png)

Fig. 9. Neural network training interface.

Figure 11 shows that neural network model is trained to 24 epochs by iteration, and the declining gradient can reach required minimum. Target error is in the minimum range, which can illustrate that training of neural network model is ideal. It can meet convergence requirement. It can reach the minimum gradient 1e-005 and the minimum target error less than 0.001, which illustrate that the model can change and confirm the weights that link different layers. Figure 11(a) shows that best validation performance is 5.7463e-005 at epoch 24.
As it can be seen from Figure 12, the best individual fitness curve stabilizes after 110 generations, which indicates that the best individual has been achieved optimizing network to 110 generations. Comparing the same structure pure BP neural network, the GABP can improve the performance of the neural network, but it also increases the training convergence time.

4.3. Test on line
We test faults diagnosis on line using the model, which has been trained BP neural network. Table 6 provides a group of data engine accelerate to 1800r/min. The data have been normalized. The normalized calculated method is shown as follows:

Within a given bounded data set there is often a large difference between the maximum and the minimum values. When normalization is applied to a data set its’ upper and lower bounds are scaled to appreciably smaller magnitudes. We use the basic z-score normalization that is obtained by using Equation 6

\[
X' = \frac{X - \mu(X)}{\sigma(X)}
\]

Where \( \mu(x) \) and \( \sigma(x) \) denote the normal data and the standard deviation of \( x \), respectively. This normalization produces a data set, where each point has a normal close to zero and a variance close to one.

Figure 14 provides the parameters curve when the Oxygen sensor fault is set. Table 7 provides diagnosis
results of test data. It can be seen that the real output of the neural network is close to the expected output.

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.0045</td>
<td>0.0012</td>
<td>0.0201</td>
</tr>
<tr>
<td>2</td>
<td>0.0000</td>
<td>0.0038</td>
<td>0.0020</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>0.0001</td>
<td>0.0034</td>
<td>0.0028</td>
<td>0.0003</td>
</tr>
<tr>
<td>4</td>
<td>0.0000</td>
<td>0.0048</td>
<td>0.0012</td>
<td>0.0003</td>
</tr>
<tr>
<td>5</td>
<td>0.0007</td>
<td>0.0050</td>
<td>0.0001</td>
<td>0.3175</td>
</tr>
<tr>
<td>6</td>
<td>0.0000</td>
<td>0.0045</td>
<td>0.0012</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

Table 6. A group of test data.

<table>
<thead>
<tr>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
<th>x10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6282</td>
<td>0.3182</td>
<td>0.0520</td>
<td>0.0203</td>
</tr>
<tr>
<td>2</td>
<td>0.6282</td>
<td>0.3520</td>
<td>0.0650</td>
<td>0.0136</td>
</tr>
<tr>
<td>3</td>
<td>0.6282</td>
<td>0.8860</td>
<td>0.1538</td>
<td>0.0136</td>
</tr>
<tr>
<td>4</td>
<td>0.6282</td>
<td>0.0136</td>
<td>0.0520</td>
<td>0.0203</td>
</tr>
<tr>
<td>5</td>
<td>0.6282</td>
<td>0.3317</td>
<td>0.0000</td>
<td>0.3114</td>
</tr>
<tr>
<td>6</td>
<td>0.6282</td>
<td>0.3182</td>
<td>0.0520</td>
<td>0.0068</td>
</tr>
</tbody>
</table>

Table 7. Diagnosis results.

<table>
<thead>
<tr>
<th>Sample</th>
<th>BP neural network real output</th>
<th>Expected output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9886 0.1143 0.0707 0.0909 0.0903 0.0607 (100000)*</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.0986 0.9790 0.0789 0.0924 0.1022 0.0968 (010000)*</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.1018 0.1015 0.9408 0.1027 0.1428 0.1118 (001000)*</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.1193 0.0996 0.1175 0.9471 0.0732 0.0854 (000100)*</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.0617 0.0862 0.1401 0.0810 0.9181 0.0838 (000010)*</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0444 0.1025 0.0132 0.0429 0.0001 0.9327 (000001)*</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 14. Parameters curves when oxygen sensor has a fault.

5. Discussions

We used X-431 to get the test vehicle’s sensors data and used the AUTO5-1 exhaust gas analyzer to get the emissions information. These data were sent to BP neural network and GABP model to train the two models then using the trained model to diagnose faults.

We test 120 groups of data, which can be diagnosed are 118 groups of data. The accuracy of diagnosis can reach 98.33%. If only use sensors data flow and exhaust emissions information the accuracy of diagnosis is 93.33 and 83.33% respectively. It means that we use sensors data flow and exhaust emissions information to diagnose engine’s faults is valid and better than only use one.

We also diagnosed these data using GABP model, it can diagnose all faults accurately and the precision is higher than BP neural network, but training convergence time is increased.

It is must be pointed that we cannot guarantee the exhaust emission information keep up with the sensor data flow because they are collected using different equipment. X-431 gets the engine sensors data and then sends the data to the computer. AUTO5-1 exhaust gas analyzer gets the exhaust gas information and then sends the information to the computer. There is no connection between the two devices. They send information independently, so we cannot insure that they can get the engine’s information at the same time. How can we get synchronous data is needed to research deeply. The model should be modified further for raising the accuracy of fault diagnosis on the other hand.

6. Acknowledgement

This work is supported by the Fundamental Research Funds for the Central Universities (NO.2572018BG02).

References


