Research on Fault Diagnosis Method of Industrial Robots Based on Case-Based Reasoning

Lin ZHANG* and Chun-ling WANG
Institute of Robot Engineering, AnHui SanLian University, HeFei, AnHui, China
*Corresponding author

Keywords: Case-based reasoning, Industrial robot, Fault diagnosis, Text frequency.

Abstract. The application enterprises of industrial robots generally do not have the ability to independently diagnose and repair the faults of industrial robots. When robots fail and stop operation, they often need to notify external service providers to diagnose and repair the fault sites of enterprises, which seriously affects the production tempo of enterprises. The loss of the enterprise is aggravated. Therefore, according to the dynamic evolution of the industrial robot fault diagnosis method and the characteristics of the case reasoning method, this paper applies the Case-Based Reasoning method to the industrial robot fault diagnosis. By using the concept of Inverse Document Frequency (IDF) index in information theory, the case organization, case retrieval and case retention technology in CBR technology are designed, which provides an effective method for the diagnosis of industrial robot faults.

Introduction

Industrial robot is an intelligent equipment, which integrates mechanical, electronic, control, computer, sensor, artificial intelligence and other interdisciplinary advanced technologies. It is of great significance to improving the level of intelligent manufacturing in manufacturing industries. Industrial robots, as the main automatic equipment in modern manufacturing industries, is widely used. At present, all countries attach great importance to the technical research of industrial robots. The possession of industrial robots has become one of the important indicators of measuring the overall strength of a country's manufacturing industry.

Although industrial robots have played an increasingly important role in modern manufacturing and are rapidly gaining popularity in the world, especially in China. However, structure of industrial robots, which incorporates a variety of high technologies, give high demands on the technical skills of maintenance technicians. At present, the application enterprises of industrial robots do not have the ability of independent maintenance. When the robots break down, it is often necessary to notify the external service come to the enterprise fault site to provide diagnosis and repair. This process requires a lot of time. Because maintenance technicians cannot have detailed data of daily operation of the faulty robot, fault phenomena and operational data before and after the fault occurred, they often cannot quickly diagnose faults and handle the fault after the fault site, which drags slowly the recovery of normal production time, severely affects the company's production beats, and exacerbates the losses of the suspended companies. Therefore, advanced fault diagnosis technology is very necessary to ensure the efficient and stable operation of industrial robots.

Case-based Reasoning technology originated from the description of Dynamic Memory by Roger Schank of Yale University in 1982. It is a relatively new leaning method of knowledge-based problem solving in the field of artificial intelligence. The problem is solved by constructing and building a case library with rich subjects, reusing or revising the previous solution to similar problems. In the problem solving mechanism, CBR adopts a case-based reasoning strategy and imitates analogy in the human decision-making process to effectively solve unstructured and knowledge-poor domain problems [1-11].

Although case-based reasoning does not improve the professional skills of industrial robotic maintenance technicians, it is a good way to digitize and apply maintenance experience. Through case inference of industrial robot faults, we can analyze and diagnose machine faults, provide
corresponding maintenance suggestions, and even guide operators to solve common problems, so as to ensure the efficient and stable operation of industrial robots.

**Case-Based Reasoning Model of Fault Diagnosis of Industrial Robots**

By comparing the fault appearances and diagnosis results encountered during application of a certain type of industrial robot, a Case-Based Reasoning model framework for fault diagnosis of industrial robots is showed in Figure 1 below:

In the Case-Based Reasoning model of industrial robot fault diagnosis, the fault representations of industrial robots are first standardized and described, and the fault features are expressed in the form of feature vectors, and the search to find the closest fault case in the fault case library. If the case with the same or similarity within the threshold is found, the diagnosis plan of the old case can be reused directly. Otherwise, the diagnosis plan is re-modified according to the most similar matching cases to form a new case and saved in the case base.

![Figure 1. Case-Based Reasoning model for fault diagnosis of industrial robots.](image)

**Key Technologies of Case-Based Reasoning for Fault Diagnosis Model of Industrial Robot**

The case-based reasoning process mainly includes four key technologies: case knowledge representation, case retrieval, case reuse/case modification, and case study.

**Knowledge Representation of Fault Cases**

Before using CBR, data should be first cleaned and sorted. Due to various reasons such as the maturity and operation of technology of a certain type of industrial robot, various problems occur during the process. However, whether the problem was recorded, whether the diagnosis result and the repair process can be expressed, it will affect the construction of the case base. When the same problem is expressed, the storage structure, feature representation, and evaluation content of the data often differ greatly with time and space. Therefore, many data are difficult to compare on the same platform.

Here, we use Boolean eigenvectors to represent case knowledge. Because different fault characteristics reflect different problems, and the data is mostly unstructured data, we first establish a fault signature table, namely, the fault characteristics of a certain type of industrial robot encountered in the operation process (eg. start alarm, arm tremors, servos fever, etc.) to integrate and break it down into individual options. The fault feature statistics table as shown in Table 1 below:
<table>
<thead>
<tr>
<th>Fault number</th>
<th>Fault content</th>
<th>Property value</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fault features 1</td>
<td>0</td>
<td>0.86</td>
</tr>
<tr>
<td>2</td>
<td>Fault features 2</td>
<td>1</td>
<td>6.34</td>
</tr>
<tr>
<td>3</td>
<td>Fault features 3</td>
<td>0</td>
<td>6.69</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>i</td>
<td>Fault features i</td>
<td>1</td>
<td>6.69</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>j</td>
<td>Fault features j</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>Fault features n</td>
<td>1</td>
<td>9.12</td>
</tr>
</tbody>
</table>

Table 1. Fault Feature Statistics.

Assume that there are n fault features in the fault feature statistics table, and the indicators of each case can be represented by an n-dimensional feature vector $X$, where $X = (x_1, x_2, ..., x_n)$. In the vector, if the i-th fault feature does not occur, $x_i=0$, otherwise $x_i=1$.

It is easy to find that the weight of each fault feature should be different in the fault feature statistics table. The Inverse Document Frequency (IDF) in information theory has pointed out that the stronger the ability of a word to predict a topic is, the greater the weight should be. In other words, if a failure feature occurs rarely, then it is easy to make a diagnosis, and the weight should be greater.

Therefore, we use the IDF theory to define the weight of the i-th fault feature as $W_i = \log(D/D_i)$, where $D$ is the number of all cases in the fault case base and $D_i$ is the number of cases which fault feature i with an attribute value of 1.

Assume that there are 100,000 fault cases in the fault case database, of which 800 cases have fault feature i and 50000 cases have fault feature j, that is $D=100000$, $D_i=800$, $D_j=50000$, so the weight of fault feature i is $\log(D/D_i)=\log(100000/800)=\log(125)=6.96$, and the weight of the fault feature j is $\log(D/D_j)=\log(100000/50000)=\log(2)=1$. By analogy, we can get the weights of all the fault features, and then get the statistics table of the fault features with weights as shown in Table 2 below:

<table>
<thead>
<tr>
<th>Fault number</th>
<th>Fault content</th>
<th>Property value</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fault features 1</td>
<td>0</td>
<td>0.86</td>
</tr>
<tr>
<td>2</td>
<td>Fault features 2</td>
<td>1</td>
<td>6.34</td>
</tr>
<tr>
<td>3</td>
<td>Fault features 3</td>
<td>0</td>
<td>6.69</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>i</td>
<td>Fault features i</td>
<td>1</td>
<td>6.69</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>j</td>
<td>Fault features j</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>Fault features n</td>
<td>1</td>
<td>9.12</td>
</tr>
</tbody>
</table>

Table 2. Statistics of Fault Characteristics with Weights.

So, the attribute weight vector $W=(w_1, w_2, ..., w_n)^T=(0.86, 6.34, ..., 9.12)^T$ is obtained.

Although IDF shows good application effect on the distribution of information feature value weights, the original intention of IDF was to suppress the negative impact of non-meaningful high-frequency words in a document, but when the ratio of total document to keyword contained document is higher, frequency words will be highlighted. There is a question to be discussed here: common words are not equal to meaningless words, such as some public figures, hot events, etc. Similarly, the occurrence of low-frequency words will be treated as a high-weight keyword. This transition enlarges the importance of rare words. Given the inadequacies of the IDF, the frequency of the i-th keyword occurring among different classes will directly affect whether the keyword can be a feature word of the document. Therefore, an item can be added between the original document class to represent the feature. The distribution of words among various classes, namely, the inter-class dispersion of feature words. By the same token, we can also add the same degree of inter-class dispersion during robot fault diagnosis. The inter-class dispersion between classes...
describes the distribution of fault features in different models of the same industrial robot. The fault
features that are distributed in a certain model or a few models often have strong category
discrimination capabilities. Therefore, the fault features have strong inter-class dispersion. Suppose
an industrial robot has n models in total, and f(i) represents the frequency of occurrence of fault
feature i in some models, then \( f(i) \) represents the average frequency of occurrence of fault feature
i in all models. Therefore:

\[
\bar{f}(i) = \frac{1}{n} \sum_{k=1}^{n} f_k(i)
\]  

(1)

And the overall inter-class dispersion is:

\[
D(i) = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (f(i) - \bar{f}(i))^2} \quad \frac{f(i)}{\bar{f}(i)}
\]  

(2)

Substituting (1) into (2):

\[
D(i) = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (f(i) - \frac{1}{n} \sum_{i=1}^{n} f(i))^2} \quad \frac{1}{n} \sum_{i=1}^{n} f(i)
\]  

(3)

Combined with the original idea of IDF, if the fault feature in formula (3) only appears in a
certain model, then D(i) is 1, then the item has the strongest classification ability. And the fault
feature's frequency of occurrence of the class is equal, then the term is considered to have no
classification ability, and therefore D(i) is 0. This feature is an unnecessary feature and can be
discarded. It can be seen that the value of D(i) is between [0,1]. An IDF algorithm variant that takes
into account the degree of inter-class dispersion is:

\[
W_i = IDF \ast D(i) = (\log \frac{D(i)}{D_t}) \ast \left( \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (f(i) - \frac{1}{n} \sum_{i=1}^{n} f(i))^2} \right) \quad \frac{1}{n} \sum_{i=1}^{n} f(i)
\]  

(4)

However, although the degree of inter-class dispersion is considered here, if two fault features are
distributed in the same model, we still cannot accurately determine the distribution of the two fault
features. Therefore, we define the information entropy within the model so that it can reflect the
distribution of fault features within the model. For example, if the distribution of a fault feature i is
more uniform in a certain model, the greater the information entropy in the model is, the more the
fault feature i can reflect the feature information of the model, and the calculation formula of the
information entropy in the model is:

\[
E(t, C_k) = -\sum_{j} \frac{N_{d_j}}{N_{C_k}} \log \frac{N_{d_j}}{N_{C_k}}
\]  

(5)

Among them, Ndj indicates the frequency of occurrence of the j-th value (0 or 1) of the fault
feature i in the model Ck, and NCk indicates the frequency of occurrence of the fault feature i in all
models of the model Ck.

Finally, the inter-class dispersion degree and the information entropy in the model are integrated,
and the improved IDF algorithm is used to calculate the model differentiation so that the fault
features can be relatively accurately determined. So there are:
\[ W_i = \text{IDF} \ast D(i) \ast E(t, C_i) = (\log \frac{D_i}{D}) \ast \left( \frac{1}{n-1} \sum_{i=1}^{n} (f(i) - \frac{1}{n} \sum_{i=1}^{n} f(i))^2 \right) \ast (-\sum_{j} \frac{Nd_j}{NC_i} \log \frac{Nd_j}{NC_i}) \tag{6} \]

According to formula (6), using the improved IDF algorithm to select fault features, calculate the weight of each fault feature, and then pick out the N fault diagnosis with the largest weight as feature vectors for fault diagnosis.

Therefore, the knowledge of the fault case can be simply expressed by the following 2-tuple: Case=(FV, DT), where FV represents the feature vector of the fault feature (only feature vectors of 0 and 1), which is used when performing fault case search, the eigenvector and the weight vector of the fault feature can be matrix multiplied to form the weighted feature vector. DT represents the diagnosis suggestion. These diagnosis suggestions can also build statistics tables through the case and represent the diagnosis recommendation of each fault through the corresponding feature vector.

**Case Retrieval**

Case retrieval is the core of CBR. Its purpose is to retrieve reference cases as from a large number of cases as the basis for solving the current problem. The commonly used case retrieval strategies include the nearest neighbor strategy, induction index strategy, knowledge guidance strategy, and template retrieval strategy. In this paper, the nearest neighbor strategy is used, but the similarity calculation is not through the Euclidean distance but through the cosine theorem.

In the knowledge representation of the case, we have established a feature vector for each case, so we can use the cosine theorem to calculate the angle between the two feature vectors. Since the weight of all fault features is positive, the cosine between the two feature vectors is between 0 and 1. If the cosine value between two feature vectors is closer to 1, the angle between the two vectors is smaller. It means that the closer the fault features represented by the two feature vectors are. On the contrary, the closer the cosine value between the feature vectors is to 0, the larger the angle between the two feature vectors is. In other words, the correlation between the fault features represented by the two feature vectors is smaller.

We all know that the cosine of A is:

\[ \cos A = \frac{b^2 + c^2 - a^2}{2bc} . \]

At this point, if b and c are seen as two vectors starting from A, the above formula can be equivalent to:

\[ \cos A = \frac{<b,c>}{|b||c|} \]

where \(<b,c>\) represents the inner product of the vector and \(|b|\) and \(|c|\) represent the length of the vector.

Assuming that the attribute feature vector of the fault case X is \((x_1, x_2, ..., x_n)\), where \(x_i\) is 0 or 1, and the attribute weight vector \(W=(w_1, w_2, ..., w_n)^T\), then its weighted feature vector is: \((x_1, x_2, ..., x_n)\ast(W_1, W_2, ..., W_n)^T=(x_1W_1, x_2W_2, ..., x_nW_n)\).

Thus, assuming that the fault feature weighted feature vectors of the two fault cases A and B are \((a_1, a_2, ..., a_n)\) and \((b_1, b_2, ..., b_n)\), then the cosine of the included angle is:

\[ \cos \theta = \frac{a_1b_1 + a_2b_2 + \cdots + a_nb_n}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} \cdot \sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}} . \]

In other words, the smaller the \(\cos \theta\) value of two vectors is, the smaller the approximation degree of the vector is. On the contrary, the larger the value of \(\cos \theta\) is, the closer the two vectors are. When \(\cos \theta = 1\), the two vectors completely overlap. That means, the fault metrics of the case are exactly the same.

**Case Reuse/Case Modification**

When a new fault case occurs, we only need to calculate the cosine of the angle between each case in the new case and the case base. If \(\cos \theta = 1\), the current case is exactly the same as the new case and can be directly reused. Otherwise, we can sort by the value of \(\cos \theta\) from big to small, filter out a few cases that are closest to the new case, or set a threshold \(t\), and filter out all \(\cos \theta \geq t\) cases.
as the approximate cases. Since Case-Based Reasoning involves professional knowledge, it is difficult to fully rely on computers to automatically modify fault cases, especially to give diagnosis results and maintenance opinions. It requires manual intervention. However, if the DT in the 2-tuple representing case knowledge is also represented by a feature vector, then we can use the DT eigenvector of the selected approximate case as the Boolean operation "and" to determine the basic diagnosis and maintenance recommendations. Then the artificial intervention by experts to reduce the degree of manual intervention.

Case Study

The result of a new failure case obtained after manual intervention is not necessarily correct. It needs practice verification and the new failure case proved to be correct through practice verification can be added to the failure case base. When a new fault case is added to the fault case base, the weight vector needs to be adjusted. Each weight in the weight vector is calculated by $\text{IDF} \times D(i) \times E(t, C_k)$. When the weight vector is readjusted, because a new fault case is added, $D' = D + 1$. If the new case shows the $i$-th fault feature, $D'_i = D_i + 1$, otherwise $D'_i = D_i$. At this time, the corresponding IDF, $D(i)$, and $E(t, C_k)$ must be recalculated, and the $W_i$ value is finally modified.

Conclusion

This paper presents a Case-Based Reasoning method for fault diagnosis of industrial robots, and gives several key techniques in the reasoning process. The above demonstration model is a general model. In the application process, it is only necessary to convert the fault case characteristics and correspond to the relevant features in the model to perform fault diagnosis. However, this test is only a simple verification of the accuracy of the method. As for its effectiveness in the actual system application, it has to be verified, and there are still some problems to be resolved in several key technologies.

First, by integrating and decomposing each case into alternatives, the knowledge representation of the case is simple and easy to use, and it is suitable for most alternatives. However, the applicability of some fault features with reference data (for example, memory data differences, etc.) is not very good.

Secondly, the threshold $t$ mentioned in the case reuse technique also needs to be set by professionals. The threshold set in the diagnostic model is 95%, but it is clear that this threshold is too high and requires professionals to reset it. This will undoubtedly increase the level of human intervention.

Therefore, in practical applications, how to reduce the degree of human intervention and improve the work efficiency by simplifying the existing algorithms on the basis of ensuring the validity are all directions after the study.

Acknowledgements

This research is supported in the collection, analysis and interpretation of data by excellent young talents in colleges and universities support program with No. gxyq2017133, and AnHui SanLian University Collaborative Innovation Centre Special Research Projects with No. yjqr17002, and AnHui SanLian University Natural Science Foundation of Key Projects with No. KJZD2018014.

References


