Aircraft Fuel System Diagnostic Fault Detection Through Expert System

H. Long[1,2], X.M. Wang[2]
(1.Beijing Union University, Beijing 100101)
(2.Northwestern Polytechnical University, Xi'an 710072, China)

Keywords: *Aircraft Fuel System; Fault Diagnosis; Expert System; CLIPS*

Abstract

For the problem of fault diagnosis of aircraft fuel system, a new method for fault diagnosis of aircraft fuel system using CLIPS to build expert system is presented in this paper. By building Knowledge Base Infer Engine and using CB tool, the relevant software environment is set up. Consequently the fault diagnostic system of aircraft fuel system is successfully developed .The emulational results prove that the intelligence of expert system works perfectly and the faults of fuel system are diagnosed exactly and fast, furthermore, effective methods of reconstruction can be brought forward in the expert system. It is entirely practicable in aircraft fuel fault diagnosis system.

1. Introduction

We are concerned here with the experiences of applying expert system to diagnostic fault detection in aircraft fuel system. Expert System, an important branch of Artificial Intelligence, was developed in 1960s, and it has been a main stream of fault diagnostic technology during a few past years [1]. It includes five parts [2], which are knowledge base; infer engine; database; explanatory interface and knowledge acquisition. Aircraft fuel system as an important system of aircraft affects the aircraft's safety greatly. It is necessary to develop a fault detection of aircraft fuel system. We believe that the direction of aircraft fuel system development is integration, automatization and intelligentization. The future aircraft fuel system must have the function of fault detection. Through the information of sensors and AI the aircraft fuel's fault was detected exactly and timely. By this idea, the paper studied the application of expert system to diagnose the fault in aircraft fuel system. Using BC and CLIPS (C Language Integrates Product System) language a digital simulation platform was designed. Finally the simulation example shows this simulation platform is a good simulation and test tool for aircraft fuel system fault detection.

We report mainly on our experiences of diagnosing the generic type of fault of aircraft fuel system through expert system. Based on detailed analysis in aircraft fuel system fault pattern, we set up the fault tree and ulteriorly build the knowledge base and infer engine, through the embedded programming of CLIPS language, the aircraft fuel system fault diagnostic expert system has been designed and realized. It proved that the faults of fuel system are diagnosed exactly and fast; furthermore, effective methods of reconstruction can be brought forward in the expert system.

2. Functions of the aircraft fuel fault diagnostic expert system

The functions of aircraft fuel fault diagnostic expert system include the following aspects.

- Fault Detection: By use of the forward inference and the backward inference, the accessory faults can be diagnosed either when we test or take outfield examinations. The working states and the diagnostic process would be relatively explained, further more, the suggestions of reconstruction would be provided, and also can be printed or saved in form of files.

- Knowledge Base Management: This function includes of setting up knowledge base 、maintenance of knowledge base (insert 、amend 、delete 、consistent check etc.).
- Counseling: This function includes of the consultation on fault diagnosis 、knowledge base management and the system setting.
- System Setting: This function includes of printer setup 、diagnostic process update and other settings.

3. Systematic Design

3.1 Knowledge Acquisition

The main sign of the differences between artificial intelligent system or expert system and other computer software system is the knowledge, on the other hand, the quality and quantity of knowledge is the key factor in determining one system's capability. Therefore, knowledge acquisition is always considered as one of the biggest difficulties in designing an expert system.

The knowledge acquisition methods of the expert system included four aspects: system working principle and expert's diagnostic experiences, fault tree, feasibility of fault occurrences and fault-handling methods.

1 In order to acquire useful diagnostic knowledge and design appropriate inference strategy, we analyzed the fault feature of aircraft fuel system, and the diagnostic experiences used by diagnostic engineers to fault identify.

2 Fault Tree：Having collected plentiful fault examples of accessories in tests and outfield practices, we set up the fault tree [3]. The fault tree includes all of faults in tests and outfield practices, at the same time; we can find out the reason of fault through the Fault Mode Effect Analysis of bottom fault incidents. The fig. 1 gives an integrant example of fault tree according to adding fuel system and offering fuel system.

3 Feasibility of fault occurrence [4]: we do our best to collect the impersonal probability of fault occurrences, especially the faults with high level of occurrence probability in tests and outfield practices. If it is difficult to get the probability of fault occurrences, the experts will present it.

4 fault-handling methods: By understanding the accessories fault handling processes and fault judgment criterions, we also research how to deal with the system faults in order to ensure the aircraft flies normally and safely.

To sum up, the knowledge of the aircraft fuel fault diagnostic expert system is composed with Fault Tree、Experiential Knowledge 、Fault–handling Knowledge and Restructuring Knowledge.

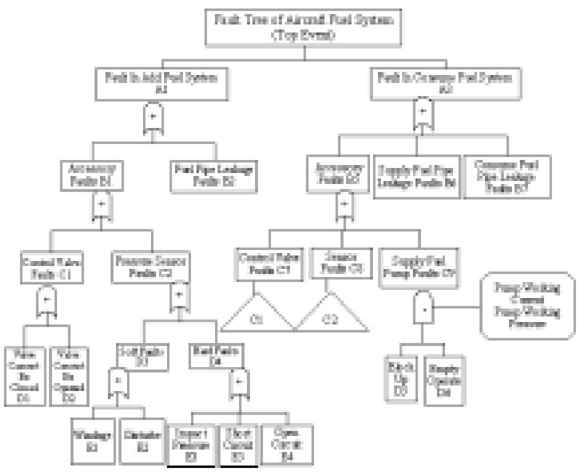

Fig.1. Fault tree of aircraft fuel system

3.2 Set up Knowledge Base

1. Table of Rules

Nowadays the most popular expert system style commonly bases on rules. The knowledge is ordinarily expressed as rule, and the rule is always composed with two parts (condition and action). The common form is as follow:

$$\text{Condition} \longrightarrow \text{Action}$$

This form is accessible 、easy to service and conveniently to infer. It is especially suit for describing the experienced knowledge and heuristic knowledge.

The fault tree matches knowledge described in form of rules individually. Every Crunode 、 Crunode's Parent Nodes and one of its Chile Nodes in fault tree denote one rule. According to the fault feature of aircraft fuel system, we designed the rule table and fact table.

In the rule table, the index is rule's serial number, and the other fields indicate the conditions and conclusions. The reliability factor is commonly included in the rule table for the purpose of inaccurate inference model. Because it is difficult to get the reliability factor of rule, we did temporarily not consider the reliability factor in this article. Every fact in fact table has an identifier number for the purpose of convenient reasoning function, and the form of rule table is as follow.

Table 1 Rule table structure of knowledge base

Rule number	Condition Number n	…	Conclusion Number
No.	Pn	...	C

Fact table includes all kinds of facts predictable and every fact is represented by a fact number. The fact index is the fact number; information related with the facts is in the other two items. The form of fact table is as follow.

Table 2 Fact table structure

Fact number	Content	Attribute
No.	…	…

In the table 1 and 2, the condition number and conclusion number in rule table appears like the fact number in fact table individually. Every piece of fact only has two meanings, one is 'True', the other is 'False', and so when the meaning of one condition or a conclusion is opposite to the corresponding fact, the number of them will take the negative value of fact number.

For example, if the meaning of one rule's condition is 'Alternating current pump is normal', and there is one fact with the fact number value '1' in fact table whose meaning is 'Alternating current pump broken down', the rule condition number should obviously take the value '-1'. The attribute of fact has two values: zero and one. Fact with value '0' expresses fault fact and the one with value '1' expresses working state fact.

2. Knowledge Base [5][6]

We programmed two fact templates to describe the working state of add fuel system and supply-consume fuel system.

- The fact template of add fuel system

```
(deftemplate AddFuel
        (slot engine)
        (slot fuelbox)
        (slot caseno)
        (slot value)
        (slot variable)
)
```

In this template, the engine name was expressed by slot engine. Because there are four engines in the aircraft fuel system discussed, the values of slot engine were described as 1、2、3 and 4. The fuel box name was expressed by slot fuelbox, and the values were described as BF(Backup Fuel Box)、FZ(Assistant Fuel Box) and YXH(Preparative Consume Fuel Box). The fact number was expressed by slot caseno. The variable corresponding fact was expressed by slot variable and its value was expressed by slot value.

For example：

```
    (AddFuel (engine 1)
                     (fuelbox BF)
                     (caseno 2)
                     (value 20)
                     (variable A))
```

The fact indicated that the prep-check fuel quantity of Backup Fuel Box which is in the group of fuel boxes related to the first engine in the add fuel system is twenty tons.

- The fact template of supply-consume fuel system

```
(deftemplate ConsumeFuel
          (slot engine)
          (slot caseno)
          (slot fuelbox)
          (slot variable)
          (slot value)
          (slot fjname)
)
```

Every accessory was named with different serial number in different associate systems. In this template, accessory's name of fuel system was described by slot fjname. Other slots expressed the same meanings with those in template AddFuel.

For example：

```
    (ConfuseFuel (engine 1)
              (caseno 18)
              (variable press)
              (fjname 2))
```

The fact indicated that the export pressure of the first alternating current pump which is in the group of fuel boxes related to the first engine in supply-consume fuel system is normal.

CLIPS language supports to Rule Based Programming、Object Oriented Programming and Process Oriented Programming. We adopt Rule Based Programming and Object Oriented Programming to express the knowledge. The applications of those two knowledge's representation modes are showed below.

In the fault diagnostic expert system of aircraft fuel system, a template is used to express the fault diagnostic rules, and the template below is about its representation mode.

```
(deftemplate rule
     (multislot if)
     (multislot then)
)
```

The template, whose name is rule, consists of two multi-field slots with the names: if and then. The combination of symptoms to one specific fault is listed in the multi-fields slot 'if', and the corresponding fault name is noted down in the multi-field slot 'then'. The form of fault name is ' (fault ? engine ? fuelbox ? fjname ? fault-type? fault-degree)' .

According to the constitution and fault condition of aircraft fuel system in the Object Oriented Programming, we decomposed the fault diagnostic knowledge and divided them into two parts: measuring layer and structure layer. The measuring point will get the working state information automatically. After dealing with the information, the programming function will translates the information into needed standard mode. The structure layer includes of engine name layer、system name layer、subsystem name layer and accessories layer. Above all, object oriented standard mode expresses the fault tree.

3.3 Infer Engine [5][6]

Since the rule of inference was built, we adopt the forward and the backward inference strategy. The heuristic search is used in the forward inference strategy, and the backward inference strategy is responsible for validating the search results. Main rules of inferences are as follows.

1. Modify part suited rule

```
(defrule modify-rule-match
     (declare (salience 20) )
       (ConfuseFuel ?case-variable ?case-value)
```

```
        ?f<- (rule(ifConfuseFuel ?case-variable
?case-value and ?rest))
=>
        (modify ?f (if ?rest))
)
```

In this rule, variable '?case-variable' stands for the variable (engine name, fuel box name, fact number etc.) corresponding the fact of expert system. The variable '?case-value' stands for the value of variable '?case-variable'.

By editing the rule step by step, either the fault expressed by the rule was identified, or the rule was deleted lacking of matched existent facts.

2. Remove not matched rule

When the knowledge does not match the existent fact it will be deleted, as a result, the program can execute rapidly and the suited workload will be decreased.

```
(defrule remove-rule-no-match
        (declare (salience 20))
        (ConfuseFuel ?case-variable ?case-value )
        ?f<-(rule (if ConfuseFuel ?case-variable
~?case-value))
=>
        (retract ?f)
)
```

The variable '?case-variable' stands for one working state parameter, such as voltage, current etc., and variable '?case-value' stands for the value of the working state parameter. If one knowledge's first slot name matches the variable '?case-variable', while it's value does not match the variable '?case-value', it is useless to be retract by the rule 'remove-rule-no-match'.

3. Find completely matched rule

The rule 'modify-rule-match' modified slot if step by step and rule 'remove-rule-no-match' deleted the rule useless, in the end, if one rule's final condition in slot if matches the existent facts, the expert system found a fault.

The form of this rule was programmed as follows：

```
(defrule rule-satisfied
        (declare (salience 20))
        (ConfuseFuel ?case-variable ?case-value)
        ?f<-(rule(ifConfuseFuel ?case-variable ?ca
se-value) )then ?engine ?fuelbox ?fjname ?fault-
type ?fault-degree))
=>
        (retract ?f)
        (assert(fault ?engine ?fuelbox ?fjname ?fau
lt-type ?fault-degree))
)
```

In this rule, when a fault was found, the related knowledge would be deleted, and new fault knowledge would be claimed by using function 'assert'. The fault knowledge expresses the fault's type and the serious degree.

4. Find faults

All of facts and knowledge having been dealt by the rules above, the faults will be found. The example below is about to find one alternating current pump's faults.

```
(defrule diesel-fault-found
        (declare (salience 10))
        (resultfile open)
        ?f<-(fault ?engine ?fuelbox ?fjname ?fault-
type ?fault-degree)
          (answer ? ?text ?fault-degree)
=>
        (retract ?f)
        (format    result    "%s    %s
n" ?text ?fault-degree)
)
```

The preemptive priority of this rule is designed as 10; while all of the above rules are designed as 20.It is ensured that the rule 'diesel-fault-found' would not be executed until other rules have been executed. This rule writes the diagnostic results in a result file using function 'format'.

The rule of 'open-result-file' and 'closeresult- file' can be programmed easily, so we did not describe their contents in detail in this article.

4. Results

The CLIPS tool can set up the aircraft fuel fault diagnostic expert system expediently, but it is difficult to set up the UI, for this reason, we had built the UI with C++ Builder programming tool, and embedded the CLIPS [7]. When one fault emergencies, the expert system will alarm the fault, show the fault information and put forward reconstructive suggestions.

One of fault diagnostic simulated results is shown in fig.2, and the diagnostic flow chart is indicated in fig.3.

In the fig.2, we set the quantity of fuel increment of preparative detection as 100 liters, when we begin to add fuel, the expert system check the quantity of fuel of all the add fuel boxes momentarily. (Because we did not add fuel in consume fuel box when adding fuel, the quantity of fuel of the fuel box is not display in the fig. 2). After preparative detection, the system reverts to the lingering detection. We set the quantity of fuel increment of lingering detection as 200 liters. When the quantity of fuel in add fuel boxes reaches the value we set just now, the quantity of fuel in Backup Fuel Box increases continually, the expert system identifies the fault that the add fuel valve can not be closed by analyzing working state information. At this time, the fault would be alarmed, and the fault information be illustrated in the UI.

If the expert system used practically, the working state information are transmitted from the accessories, and the expert system afford the fault information by analyzing the diagnostic knowledge.

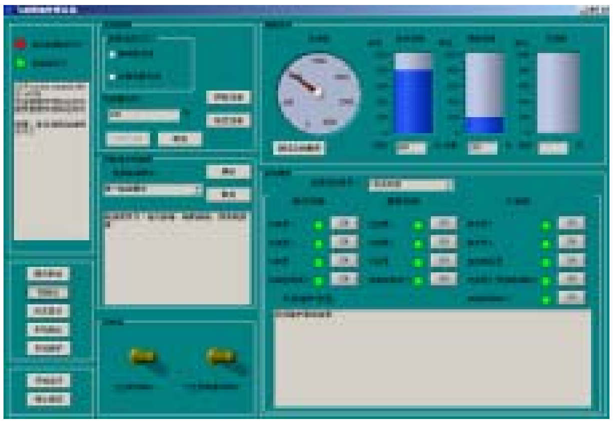

Fig.2. Valve can not be closed in add fuel system

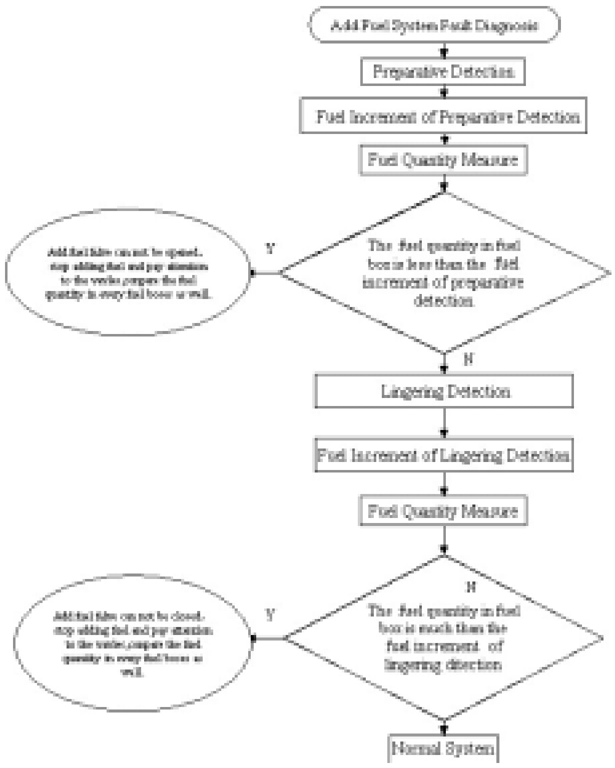
Add Fuel System Fault Diagnosis
Preparative Detection
Fuel Increment of Preparative Detection
Fuel Quantity Measure
The fuel quantity in fuel box is less than the fuel increment of preparative detection
Y
Add fuel Valve can not be opened, stop adding fuel and pay attention to the works compare the fuel quantity in every fuel boxes as well.
N
Lingering Detection
Fuel Increment of Lingering Detection
Fuel Quantity Measure
The fuel quantity in fuel box is much than the fuel increment of lingering direction
Y
Add fuel Valve can not be closed, stop adding fuel and pay attention to the works, compare the fuel quantity in every fuel boxes as well.
N
Normal System

Fig.3. Flow chart of add fuel system

Conclusion

The expert system, being a tool to aid the diagnostic process in aircraft fuel system, is achievable within current hardware and software technology. The simulation results prove that the intelligence of expert system works perfectly and the faults of aircraft fuel system are diagnosed exactly and fast. Furthermore, effective methods of reconstruction can be brought forward in the expert system. However the use of expert system is likely to remain somewhat difficulty to implement due to the difficulty of knowledge acquisition. Facts and rules are available from two different types of sources: experiential diagnostic engineer and documented data. At the same time, we have associated difficulties in assimilation and modeling. The problem here requires vast amounts of information spread throughout four departmental areas. To acquire this information would be a colossal task requiring much time and effort.

References

[1] George F. Luger. *Structure and strategy in solving artificial intelligent complex problems.* Beijing, Mechanical industrial press, 2004

[2] Wu Jinpei. *Intellective fault detection and expert system.* Beijing. Science press, 1997

[3] Zhang Rongxin. *Fault diagnostic expert system of hydraulic pressure system.* Mechanism engineering magazine, Vol.33 (7):32-35,2002.

[4] A.H. Haien. *The fault diagnosis and elimination of the hydrokinetic system.* Mechanical Industrial Press, 2000.

[5] *Giarratano. Ph. D. CLIPS user's guide version 6.10* 1998.8

[6] *CLIPS reference manual volume Ⅰbasic programming guide version 6.10* 1998.8

[7] *CLIPS reference manual volume Ⅱbasic programming guide version 6.10* 1998. 8