Operational fault diagnosis of manufacturing systems

W. Hu, A.G. Starr, A.Y.T. Leung
Manchester School of Engineering, University of Manchester, Manchester, M13 9PL, UK

Abstract

Among all kinds of possible faults in a manufacturing system, operational faults occur most often (about 70%). Efficient diagnosis of these faults is critical for improving the availability and productivity of the manufacturing system. This paper presents a hierarchical diagnosis model based on fault tree analysis and two other diagnosis models respectively based on the logic and sequential control of manufacturing systems which are usually controlled by a Programmable Logical Controller (PLC). With these models working together, the operational faults of a manufacturing system can be diagnosed completely. The models have been successfully applied to a PLC controlled flexible manufacturing system and have achieved good results.

Keywords: fault diagnosis, manufacturing system, fault tree analysis, PLC

1. Introduction

Automated plant such as a Flexible Manufacturing System (FMS) comprises many complex elements, and is quite different from ordinary machine tools because of modular architecture, distributed multi-level control and hierarchical system functions. Among all kinds of possible faults in a manufacturing system, operational faults occur most often (about 70%). The propagation of faults from parts or components to functional modules, to machine tools and hence to the system, has led human experts to adopt a diagnostic solution called hierarchical isolation when they are diagnosing a system fault. In addition, because such a manufacturing system is highly automated and integrated, many different fault modes exist as well as available diagnostic knowledge. Therefore, the fault diagnosis of such a manufacturing system cannot be carried out like that of an ordinary machine tool. A method for quick and comprehensive automation and integration is required.

In recent years, various strategies have been reported for the diagnosis of manufacturing systems. Toguyeni proposed reasoning mechanisms for the implementation of an on-line diagnostic system for FMS's, which are based on the distributed processing of symptoms [1]. Kim employed evidential reasoning to identify malfunctions of semiconductor manufacturing equipment by combining evidence originating from equipment maintenance history, on-line sensor data, and in-line post-process measurements [2]. Bohez and Thieravarut used a hybrid reasoning approach between a deep model and a shallow model for the diagnosis of computer numerically controlled machines [3]. Chevalier integrated causal reasoning and fuzzy logic reasoning for manufacturing line supervision and diagnosis [4]. To some extent, these diagnostic strategies have successfully been used and have solved some practical problems. However, there is little evidence to suggest that all the fault data, as well as available diagnostic knowledge, has been integrated in manufacturing systems.

Fault Tree Analysis (FTA) is a mature and efficient fault analysis method, which has been used in a variety of complex diagnostic applications such as digital fly-by-wire flight control systems [5], air-cooled turbo generators and spacecraft propulsion systems [6]. It is also used for fault identification and fault forecast [7]. FTA can help the maintenance personnel in finding the shortest path to a result in the diagnosis of complex machinery [8]. Furthermore, It has several unique advantages compared with If-Then-Else statements placed in the test software to direct troubleshooting.

In order to meet the quest for automation and flexibility, many manufacturing systems are controlled by Programmable Logical Controllers (PLC) [9]. This is because PLC's are adaptable, modular, user-friendly and acquired at low cost. However, because of the PLC's inflexible programming system, its capability in fault detection and diagnosis is limited. Operational faults associated with PLC control processes often confuse the maintenance personnel at workshop level. When such a fault occurs, about 80% of downtime is spent locating its source and only 20% is spent on the repair [10]. The availability and productivity of these PLC controlled manufacturing systems can be improved by shortening their downtime resulting from faults. This has led to the development of automatic diagnosis tools or systems based on PLC control.

Many diagnosis methods as well as systems have been reported in the literature. Jarvis proposed an approach which was used to develop a model of a PLC controlled assembly line. The objective of the approach was to simulate the sequence of manufacturing events that occur for each station in the assembly line. During the simulation, meaningful comparisons were made between the simulated state of the system and the observed state of the system (as specified by a snapshot of the PLC state) [11]. Plomp described a prototype of a support tool for PLC analysis. The tool was motivated by observations regarding the inefficiency of current PLC software debugging tools and the poor availability of cross-referenced documentation and manuals.

The prototype analyses the temporal signal dependencies within the PLC logic model and a history of logged values [12]. Matthias presented a method to model event based systems and described how post-mortem diagnosis based on the use of such models can be performed for PLC controlled manufacturing equipment [13].

In this paper, a hierarchical diagnosis model based on FTA is put forward. Two other models, logical diagnosis model based on PLC Logic Function Charts (LFC) and sequential diagnosis model based on PLC Sequential Control Process (SCP), are also presented. These models have been used in an existing diagnosis system for FMS's and are very suitable for fault diagnosis of PLC controlled manufacturing systems like a FMS.

2. Hierarchical Diagnosis Model Based on FTA

The FTA-based hierarchical diagnosis model for manufacturing systems follows the principle of fault tree construction and analysis. A manufacturing system is modularised in multiple levels according to the system function, working principle and expert experience, i.e. to describe the propagation process of the system faults in the form of a tree. For different levels in a fault tree, it adopts different concrete diagnosis methods, so as to locate the fault level by level until a specific level in the fault tree is reached and the corresponding result is obtained.

2.1 FTA method

FTA is a fault analysis method that is used to identify the cause(s) of a system fault hierarchically from the system level to the part/ component level. A fault can be located by analyzing the logical relationship between the system fault and its cause(s) along a fault tree.

FTA is suitable for manufacturing system fault diagnosis because it has following characteristics:

- FTA can be used deeply to analyse a specific fault level by level. It uses clear graphics to vividly describe the internal logic relationship between the part/component faults and the system fault.
- The fault tree can clearly indicate a system fault is related to which part(s)/component(s), what the relationship is, and how strong the relationship is. It can also be seen whether a part/component fault will cause a system fault, what the effect is, how great the effect is, and its mechanism.
- A fault tree is a clear illustration for those management and maintenance personnel who have never participated in the system design and trial-manufacture, which will greatly shorten the training time of the maintenance personnel, and therefore cut down the expense for personnel training.
- The qualitative analysis of fault trees of a system may make the system designers clearly understand the fault modes and success modes of the system, so as to be able to find out the weak links in the design scheme, and take corrective measures.

To construct fault trees for a manufacturing system, we must fully understand the system. In addition, we should have plenty of experience on the system operation and maintenance. Otherwise, some important events may be omitted during the fault tree construction, which may lead to incorrect results. The process of FTA is as in Fig. 1.

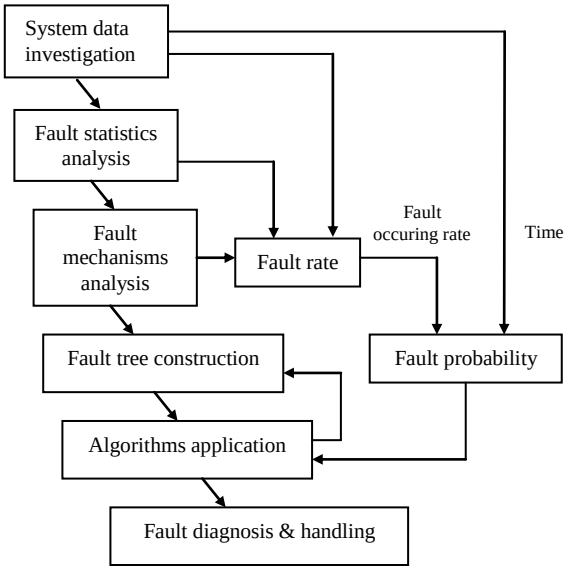


Fig. 1. The process of FTA.

2.2 Construction of the hierarchical model

A manufacturing system has a complex hierarchical architecture. From top to down, normally it can be divided into system level, materials flow sub-system level, production equipment sub-system level, functional module level and part/component level. Each level can be further divided into sub-levels and sub-sub-levels. Using a comprehensive fault tree to describe a manufacturing system fault, the fault tree will be like Fig. 2.

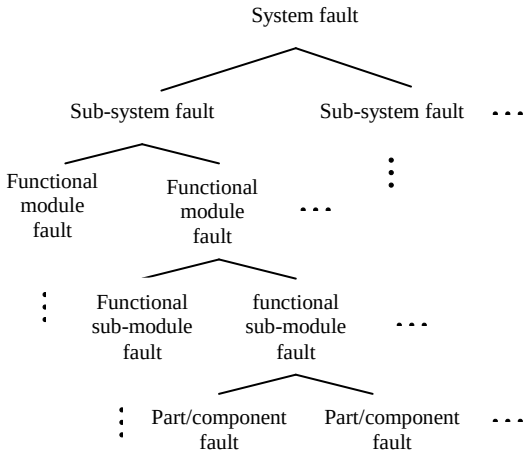


Fig. 2. The fault tree structure of a system fault.

That is to say, for a system fault, the fault sources at the first level are the sub-systems and the fault causes are the current faults of these sub-systems. The fault sources at the second and third levels are respectively functional modules and sub-modules, and the causes are the faults of these functional modules and sub-modules. Part/component faults are the lowest level faults which

are usually the possible final causes of a system fault. Therefore, the hierarchical diagnosis of a manufacturing system is essentially a search process using diagnostic strategies, along with hierarchical fault trees, from top down, i.e. from the top level system fault to part/component level faults.

According to function, the diagnostic knowledge in a manufacturing system is classified into meta-knowledge, principle knowledge and experiential knowledge. In simple terms, meta-knowledge is knowledge about knowledge. More precisely, it is the knowledge about how to use various principle knowledge and experiential knowledge. This kind of knowledge is used to guide the selection and use of all the available principle knowledge and experiential knowledge. Using meta-knowledge to guide the knowledge selection can avoid the unnecessary diagnostic reasoning, so as to improve the efficiency of diagnosis. Principle knowledge is the knowledge about the working principle of the system, which describes the propagation processes of the system faults from the view of the system working principle. Experiential knowledge is the knowledge about the expert experience. It explains the faults according to the mechanisms of the faults. The experiential knowledge selects the optimal problem solution.

2.2.1 Hierarchical modularization of diagnostic knowledge

In a manufacturing system, the knowledge associated with the diagnosis of a specific fault is only a part of the diagnostic knowledge bases. If we search all the knowledge bases, it will be slow and may not satisfy the requirement of real-time diagnosis. Therefore, measures must be taken to modularize the knowledge in different levels, so that the reasoning engine can select those relevant knowledge bases for diagnosis. A method is to decompose a manufacturing system into several sub-systems according to its function, meanwhile, decompose each sub-system into functional modules and sub-modules according to the function of this sub-system. All these make up the meta-knowledge base. There is only one meta-knowledge base for a manufacturing system. For each functional sub-module, a fault tree is built according to its working principle, which makes up a principle knowledge base. A principle knowledge base describes the fault propagation process of a relative functional sub-module from the view of its working principle. Similarly, in the form of a fault tree, the experiential knowledge associated with each principle fault are combined to make up an experiential knowledge base. When a human expert diagnoses a system fault, he/she usually first searches and finds out the faulty functional sub-module using meta-knowledge. Then he/she finds out a rough fault cause using the principle knowledge associated with the faulty functional sub-module, and finally analyzes the fault cause according to the relevant experiential knowledge until the final fault cause is found.

Therefore, the diagnostic knowledge in a manufacturing system can be described in the hierarchical modular form in Fig. 3. Each module represents a knowledge base at that level. The modularization of the knowledge bases is in accordance with the FTA method. The knowledge in every knowledge base is in the modular structure of a fault tree as well.

2.2.2 Hierarchical decomposition of the system function

Like the fault tree structure of a system fault in Fig. 2, the hierarchical decomposition proceeds as follows:

- the function of the system is represented by those of its sub-systems;
- the function of each sub-system is represented by those of its modules;
- the function of each module is represented by those of its sub-modules.

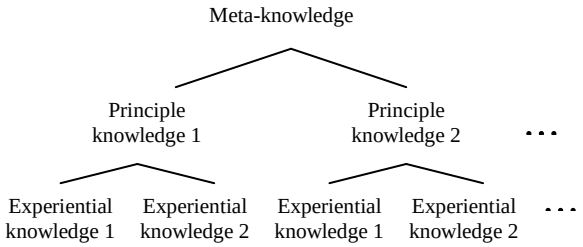


Fig. 3. The modular hierarchy of diagnostic knowledge.

The result of decomposition is a fault tree which has four layers. Each represents a new hierarchical decomposition of the system function, except the first layer which is the top event. The events at this layer are also the possible causes of an upper layer system or sub-system or functional module or sub-module fault. Each functional sub-module at the lowest layer, i.e. bottom event, corresponds to a physical structure of the system, which is associated with a principle knowledge base. That is to say, if a faulty sub-module is found, the corresponding principle knowledge base is obtained, which can be used for further diagnosis. Thus, the knowledge about the hierarchical decomposition of the system function can be used to guide the selection of principle knowledge. According to the above definition, this kind of knowledge is taken as meta-knowledge.

2.2.3 Fault decomposition based on the system working principles and expert experience

The decomposition of a fault divides a functional sub-module of the manufacturing system into several layers according to the system working principles, expert experience and physical structure. The fault propagation process can be represented using the causality between modules at the higher layers and the faults at the lower layers of the module. This is a convenient method for the analysis of fault propagation from the lowest principle faults to the functional sub-module faults.

In general, the correct operational behaviour of a manufacturing system may be characterized by a series of state transitions of the system, used during the manufacture of a product [14]. These state transitions occur because of the proper functioning of causal agents responsible for the transitions. The state transitions are monitored through multiple sensors in a manufacturing system. The monitored information is finally sent to the system control mechanisms such as a PLC. The information includes various state signals that indicate the system operating state, I/O signals and position signals in the PLC. The operation of a manufacturing system is performed through transiting signal status according to the logical relationships among signals to drive the corresponding physical mechanisms. If a state transition is not completed or is in error, the functional sub-module is considered to be faulty. At this time, a rough physical position where there is a fault can be located using the state signals in the PLC and the logical relationships among these signals. The detailed algorithms for PLC based fault detection will be presented in the next section. This knowledge of system workings is called *principle knowledge*. A fault tree built like this is also called a principle fault tree. In the tree the top event represents a functional sub-module fault. Each middle event represents a fault of a physical component of the functional sub-

module. Bottom events represent the lowest physical faults that may be located according to the system working principle.

After a fault is located to a specific position using meta-knowledge and the above mentioned principle knowledge, sometimes it is still not the final fault position. Because of the limit and incompleteness of information in the PLC, further diagnosis can only be carried out by expert experience, using expert experience to analyze the current fault layer by layer till the final fault cause is found. This kind of knowledge is called experiential knowledge which is normally represented in the form of a rule. A rule-based fault tree is also called a rule tree. The final diagnostic decision is always obtained in the rule(s) at the lowest layer or from the bottom events of a rule tree.

2.3 Diagnostic reasoning procedure

The diagnostic reasoning procedure derived from the hierarchical model is shown in Fig. 4.

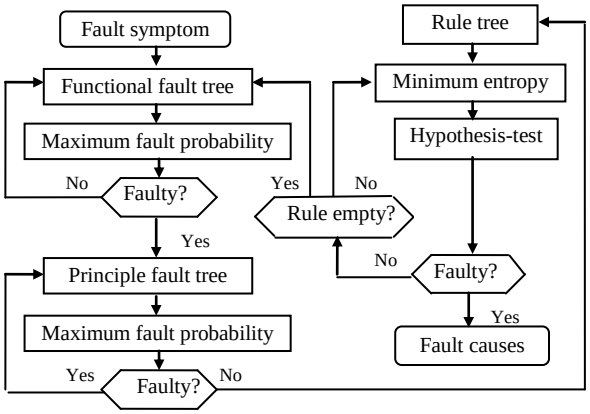


Fig. 4. Integrated diagnostic reasoning process.

In the procedure, diagnostic reasoning based on the functional fault tree and principle fault tree is performed using the strategy of a breadth-first search combined with the fault probability of each node in the fault trees. Whether a node is faulty or not is determined by various signals in the PLC and the logical relationships among these signals. Reasoning based on experiential knowledge or rule trees is more complicated. For rules associated with machining process, reasoning is performed with the help of process monitoring results, while for other rules, reasoning is performed as a sequential hypothesis-test cycle. In the hypothesis-test cycle, a cost-weighted entropy criterion is used to choose the next part of the rule tree to be activated. This entropy criterion helps to select the rule that gives the maximum fault discernment per unit cost in cases where multiple tests might be performed.

Supposing R_{kj} is a node at the k-th layer of the rule tree and $R_{k+1,1}$, $R_{k+1,2}$,, …, $R_{k+1,m}$ are nodes at the (k+1)-th layer, the cost-weighted entropy of node R_{kj} can be calculated by the following equation:

$$H = -\sum_{j=1}^{m} w_j P_j \ln P_j \qquad m \geq 2 \tag{1}$$

where

$$1 \geq w_j \geq 0, \ \sum_{j=1}^{m} P_j = 1 \tag{2}$$

The weighting factor, w, is a normalized cost, determined by the ratio between the actual cost of a test operation and the maximum of the set of test costs for all components at the current rule layer. P_j is the probability with that $R_{k+1,j}$ is the cause of R_{ki}, under the condition that the test result of node R_{ki} is known. Cost-weighted entropy is used to select and activate a part of an experiential knowledge base or a rule tree. It selects the test that will give the most discernment at the lowest cost. That is to say, the entropy of the next rule to be tested must be the minimum.

2.4 Object-oriented implementation

The object-oriented programming method is well suited to the implementation of hierarchical diagnostic reasoning. Firstly, the hierarchy of diagnostic reasoning networks based on meta-knowledge and principle knowledge is very similar to the inheritance of classes in the object-oriented method. Secondly, fault probabilities and the relative cost of the test can be easily stored as slot values in the units representing the relative components. A slot in object oriented programming is used to represent a property. The modular structure of software like this is highly flexible and therefore is suitable for other diagnostic tasks.

A knowledge object can be represented in the Backus Naur Form (BNF), which is a standard format adopted in object oriented programming, as:

<Frame>::=Unit: <Frame name> in <Knowledge base name>
{Superclasses: <Superclass name>
{, <Superclass name>};}
{Subclasses: <Subclass name>
{, <Subclass name>};}
{Member of: <Class name>
{, <Class name>},}
<Slot>::= Member slot | Own slot: <Slot name> from
<Frame name>
Inheritance: <Inheritance attribute>
ValueClass: <Slot value class>
{Self-defined side: <Side value>}
Values: <Slot value>
<Frame name>::=<Character> {<Character> | <Number>}
<Slot name>::=<Character> {<Character> | <Number>}
<Inheritance attribute>:=override | union | METHOD
<Slot value class>::=integer | real | string | struct | rules | METHOD | <Frame name>
<Self-defined side>::=<Character> {<Character> | <Number>}
<Side value>::=<value> | <string>
<Slot value>::=<value> | <string>
<Character>::=A | … | Z | a | … | z
<string>::=<Character> {<Character> | <Number>}
<Number>::=0 | … | 9

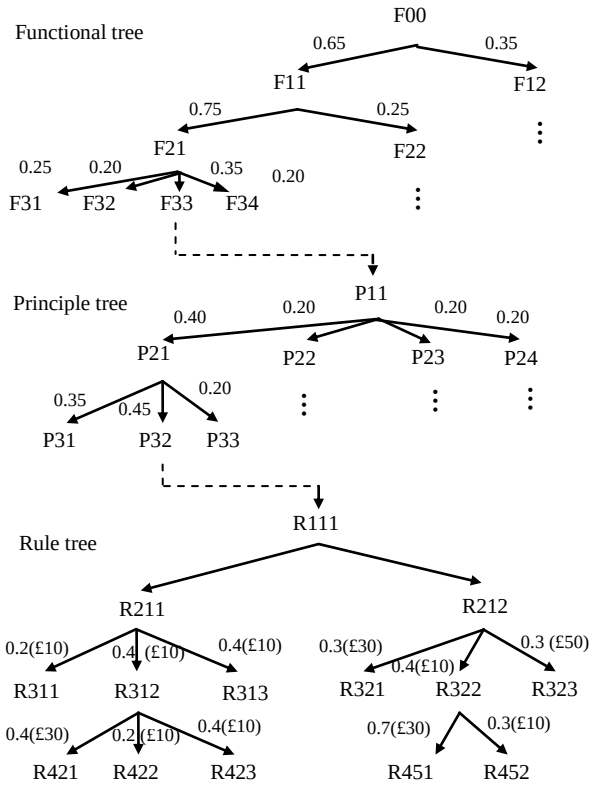
The ellipsis {} indicates that the contents may appear 0 to multiple times.

2.5 Example

The hierarchical diagnostic reasoning model has been implemented on a FFS-1500-2 FMS. The FMS consists of a PFZ1500 Flexible Manufacturing Cell (FMC), a KBNG85 Machining Center (MC) and an Automatically Guided Vehicle (AGV). The PFZ1500 FMC is made up of functional modules which include tool change, tool-head change, axis drive and

hydraulic drive. The axis drive can be further divided into spindle drive, X-axis drive, Y-axis drive, and Z-axis drive. This kind of decomposition is based on a FTA.

This example describes a failure of the PFZ1500 FMC because of some unexpected fault(s). The diagnostic search by the integrated hierarchical diagnostic model was conducted as shown in Fig. 5.



F00:	FFS-1500-2 FMS	P33:	Spindle Transmission
F11:	Machine Tools	R111:	Spindle Motor
F12:	AGV	R211:	Motor Drive
F21:	PFZ1500 FMC	R212:	Control Circuit
F22:	KBNG85 MC	R311:	Mechanical Parts
F31:	Tool Change	R312:	Motor Temperature
F32:	Tool-head Change	R313:	Motor Connection
F33:	Machining Process	R321:	Circuit Connection
F34:	Hydraulic Drive	R322:	Power Connection
P11:	Machining Process	R323:	Control Amplifier
P21:	Spindle	R421:	Tool Edge
P22:	X-axis	R422:	Cooling System
P23:	Y-axis	R423:	Feed Load
P24:	Z-axis	R451:	Fuse
P31:	Spindle Feed	R452:	Power Switch
P32:	Spindle Motor		

Fig. 5. An example diagnostic reasoning procedure.

Depending on signals in the PLC and condition monitoring results, a first search through functional knowledge base (functional fault tree) leads to F33 (machining process), which is a terminal functional knowledge node. Principle knowledge base P11 (that corresponding to F33) is activated at this point and another search through the principle knowledge base (principle fault tree) leads to P32 (spindle motor), which is a terminal principle knowledge node.

Then experiential knowledge base (rule tree) R111 is activated and the cost-weighted entropy is computed for R211 and R212 groups, the results of which are in Table 1. In this situation, the node or rule 313 which has a minimum entropy is tested first and maintenance personnel are instructed to check the corresponding part of the system. If it is faulty, then diagnosis terminates, otherwise other observed symptoms will be checked before backtracking within the functional and principle knowledge bases.

Table 1
The computed results of entropy of nodes

R211 group		R212 group	
Nodes (R)	Entropy (H)	Nodes (R)	Entropy (H)
211	0.201	212	0.913
311	0.135	321	0.000
312	0.358	322	0.222
313	0.000	323	0.000
421	0.000	451	0.000
422	0.000	452	0.000
423	0.000		

Ultimately it was shown by test that the cooling system was faulty. The cooling oil pipeline was blocked so that the system could not provide a high enough pressure, which made the motor drive abnormal and the whole system failed to work. The fault was located and then the corresponding maintenance plan was suggested. After clearing the pipeline the system was restarted.

3. Diagnosis Models Based on PLC Control

The development of diagnosis models based on PLC control addresses operational fault diagnosis of manufacturing systems which are controlled by PLC's. They provide effective methods for maintenance personnel at workshop level to identify, classify, and correct operational faults occurring in production. The models make full use of the powerful I/O capacity of the PLC and various available control signals in the PLC.

3.1. Logic diagnosis model

When diagnosing a PLC controlled manufacturing system, maintenance personnel often focus on the LFC of the units that compose the system, and trace the faulty output along it. This method of diagnosis is based upon the concept of the manufacturing system as a transformer of power, information and material: the effect of a faulty unit is propagated with the LFC. Therefore the LFC-based diagnosis model which we call a *logical diagnosis model* will be effective for locating manufacturing system faults.

3.1.1 Construction of the logical diagnosis model

In this model, all variables associated with the LFC are described in a binary form. These binary variables include all the signals in the PLC. The model is constructed from these variables in accordance with LFC. The detailed algorithm for the model is as follows.

We assume that $S(x)$ is the state function of an operating state signal of the machine, $S(x)$=1 means that the operation associated with $S(x)$ is on, while $S(x)$=0 means that the operation associated with $S(x)$ is off. $\{s_k\}$ denotes the combination of several PLC signals connected by a logical "AND" which is written as "·" in model expressions, and $\varphi(\{s_k\})$ is the state of $\{s_k\}$. $\{s_k\}=s_1 \cdot s_2 \cdot \ldots \cdot s_k$.

In practice, the logical expression of $S(x)$ is given by a signal decomposition according to the LFC. Therefore, if we define $\varphi_{ji}(\{s_{ki}\})$ as the i-th term of $S(x)$ at the j-th level, then the result of the decomposition of $S(x)$ at the j-th level is

$$\varphi_{(j-1)i}(\{s_{ki}\}) = \varphi_{j1}(\{s_{k1}\}) + \varphi_{j2}(\{s_{k2}\}) + \cdots = \sum_i \varphi_{ji}(\{s_{ki}\}) \tag{3}$$

where $\varphi_{ji}(\{s_{ki}\})$ is a factor that makes $\varphi_{(j-1)i}(\{s_{ki}\})$=1. It may be a PLC signal or the combination of several signals connected by logical "AND". Except for the terms expressed by input signals or flag signals that cannot or need not be decomposed, other terms usually can be decomposed further according to the LFC in the PLC program.

The decomposition can proceed level by level in the same form till all the terms are expressed by input signals or non-decomposable flag signals. Then substituting the decomposition expression at every level into that at its higher level, we have

$$\varphi_{ni}(\{s_{ki}\}) \Rightarrow \varphi_{(n-1)i}(\{s_{ki}\}) \cdots \varphi_{1i}(\{s_{ki}\}) \Rightarrow S(x) \tag{4}$$

In the end we get the non-decomposable and minimized logical expression of $S(x)$, i.e.

$$S(x) = \varphi_1(\{s_{ki}\}) + \varphi_2(\{s_{ki}\}) + \cdots = \sum_i \varphi_i(\{s_{ki}\}) \tag{5}$$

where $\varphi_i(\{s_{ki}\})$ is also a factor that makes $S(x)$=1 and is composed of input signals or non-decomposable flag signals, $\varphi(\{s_{ki}\})$.

Now, let $F(x)$ be the fault state function of the machine. $F(x)$=1 means that a fault has occurred, while $F(x)$=0 means that there is no fault at all. If $F(x)$ equals $S(x)$, all the fault terms that make $S(x)$=1 can be determined, which are expressed as $f_1(\{s_{k1}\})$, $f_2(\{s_{k2}\})$, …, respectively. That is

$$F(x) = S(x) = f_1(\{s_{k1}\}) + f_2(\{s_{k2}\}) + \cdots = \sum_i f(\{s_{ki}\}) \tag{6}$$

If $F(x)$ equals the inverse state of $S(x)$, the first step will be to extract the expression of the inverse $S(x)$. Each term of the expression is a combined pattern of causes of the manufacturing system fault, i.e.

$$\overline{S(x)} = \overline{\sum_i \varphi_i(\{s_{ki}\})} = f_1(\{s_{k1}\}) + f_2(\{s_{k2}\}) + \cdots = \sum_i f_i(\{s_{ki}\}) \tag{7}$$

Thus, the logical expression at the faulty state of the manufacturing system is obtained as

$$F(x) = \overline{s(x)} = \sum_i f_i(\{s_{ki}\}) \tag{8}$$

3.1.2 Diagnostic procedure by logical model

In the logical expression at a fault state of the manufacturing system, each term of the expression represents a possible combined signal pattern of fault causes. The next step is to analyze all the possible combined patterns, until an input signal that causes the fault or a non-decomposable flag signal is found. The detailed diagnostic procedure is as follows:

- Compare the faulty state of $S(x)$ with the current state of signals in PLC: if they are the same, then it is concluded that a fault has occurred.
- Establish a logical equation about the faulty state, i.e

$$F(x) = \sum_i f_i(\{s_{ki}\}) = 1 \tag{9}$$

- Substitute the actual state values of signals in PLC into the above equation and calculate if any term $f_i(\{s_{ki}\})$=1.
- Acquire the combined pattern corresponding to the term $f_i(\{s_{ki}\})$=1. The pattern shows the exact cause of the fault.

Using this diagnosis model, we must make sure that, at a fault state, the combined patterns in the logical expression cover all the possible fault causes, and are independent to each other.

3.2 Sequential diagnosis model

Many processes in a manufacturing system, such as tool exchange, are controlled sequentially. The sequential control is performed by a series of sequential commands. These commands lead to a dynamic change of the machine operating state. For the diagnosis of sequential control faults, a SCP-based diagnosis model which we call a *sequential diagnosis model*, is introduced.

3.2.1 Construction of sequential diagnosis model

The SCP-based sequential diagnosis model consists of a number of machine states and state changes in time sequence. It describes the sequential changes of the machine operating states. The action in a certain step is not only related to the control commands in this step, but also related to the step conditions in the previous step. The current step can only be started under the condition that the previous step has finished and the current control commands have been received. Whether a step is finished or not is decided according to its step conditions. So, the sequential diagnosis model can be constructed as follows.

We assume that $C(t)$ is the combined state of all the step conditions in the t-th step. Since each condition is normally a PLC signal, marked by $c_1(t)$, $c_2(t)$, …, thus

$$C(t) = c_1(t) \cdot c_2(t) \cdots = \prod_j c_j(t) \tag{10}$$

where "$C(t)$=1" indicates the step conditions are satisfied and the next step can be started, and "$C(t)$=0" indicates the conditions are not satisfied and the action sequence cannot be carried out. Similarly, the step conditions of the previous step is expressed by

$$C(t-1) = c_1(t-1) \cdot c_2(t-1) \cdots = \prod_j c_j(t-1) \tag{11}$$

Now we can let $I(t)$ be the combined state of all the control commands in the t-th step. Notice that every control command is also a PLC signal, marked by $i_1(t)$, $i_2(t)$, …, thus

$$I(t) = i_1(t) \cdot i_2(t) \cdots = \prod_j i_j(t) \tag{12}$$

where "$I(t)$=1" indicates the commands are received while "$I(t)$=0" indicates not received.

As mentioned above, if we let $F(t)$ be the faulty state of the step, "$F(t)$=1" indicates that the step is faulty. In the case where a fault exists, it is possible that

$$F(t) = C(t-1) \cdot \overline{I(t)} \tag{13}$$

When $F(t)$=1, $C(t$-1)=1 and $I(t)$=0, which means the previous step has finished and current step started, but the control commands have not been received. From

$$\overline{I(t)} = \overline{\prod_j i_j(t)} = \overline{i_1(t)} + \overline{i_2(t)} + \cdots = \sum_j \overline{i_j(t)} = 1 \tag{14}$$

the exact command that is not received can be found. It is also possible that

$$F(t) = I(t) \cdot \overline{C(t)} \tag{15}$$

When $F(t)$=1, $I(t)$=1 and $C(t)$=0, which means the current control commands have been received, but the action has not finished. Similar to the first case, from

$$\overline{C(t)} = \overline{\prod_j c_j(t)} = \overline{c_1(t)} + \overline{c_2(t)} + \cdots = \sum_j \overline{c_j(t)} = 1 \tag{16}$$

the exact condition that is not satisfied can be found.

3.2.2 Diagnostic procedure by sequential model

Under normal operating conditions, the PLC controls the manufacturing system according to the sequence of actions. At the same time, each step in the control sequence is monitored by the watch-dog-timer in the PLC. If the machine is in its normal condition, it will operate sequentially according to the preset control sequence. Therefore, if the machine control status is delayed too long at a certain action, it suggests the occurrence of a fault.

Upon the detection of a sequential control fault, diagnosis is carried out using the sequential diagnosis model. At first the current values of all the signals in PLC will be read. Then the start conditions of every step are analyzed according to these values. By doing so, the step where a fault has occurred can be determined. In the end, each control command and condition of the faulty step are checked, till the exact fault is located. This is the diagnostic procedure by the sequential diagnosis model.

3.3 Examples

The propagation of the effects of faults through a manufacturing system and its components is well described by the logical diagnosis model. The diagnostic algorithm models the human way of thinking in the diagnostic process. This is a static model and cannot represent the dynamic change of the machine operating state. However, the sequential diagnosis model can describe such a series of state changes, and can be used to identify the step in the operating sequence where the fault occurs and the precise fault cause.

The logical diagnosis model and the sequential diagnosis model are not alternative models, but are complementary to each other. First a faulty step in the control sequence is identified using the sequential diagnosis model. Commands issued in each control step activate certain units of the manufacturing system, so further diagnostic procedures can be performed using the logical diagnosis model, which corresponds to the activated part.

(1) Example for logical diagnosis model

The FFS-1500-2 FMS uses a SIEMENS U Series PLC. The SIEMENS U Series PLC has signals such as inputs (E), outputs (A), flags (M), times (T), counters (C), and data (D). Each item in the logical expressions above is a single signal or the combination of several signals via a logical "AND".

Here let us take the start conditions of the Numerical Control (NC) system in FFS-1500-2 FMS as an example. In the PLC program, we know that M132.4 is the flag signal indicating the start condition of the NC system. If we define $\overline{\text{Xm.n}}$ as the inverse state of Xm.n, according to the logical diagnosis model and the relevant LFC's in the PLC program, M132.4 can be decomposed as follows.

$$\text{M134.2} = \text{M132.0} \cdot \text{M132.3} \cdot \text{N1} \cdot \overline{\text{M21.2}} \cdot \overline{\text{M22.2}} \cdot \overline{\text{M23..2}} \cdot \overline{\text{M129.3}} \cdot \text{E7.1} \cdot \text{E7.3} \cdot \text{E7.5}$$

$$\text{M132.0} = \text{A9.6} \cdot \text{A18.4} \cdot \text{E19.0} \cdot \overline{\text{E23.1}} \cdot \overline{\text{M144.1}} \cdot \text{E20.6}$$

$$\text{N1} = \text{E30.5} + \text{N2}$$

$$\text{N2} = \text{M133.2}\,\text{E30.2}$$

where N1 and N2 are two middle flags, "+" denotes logical "OR". After being simplified,

$$S(x) = \text{M132.4} = \text{A9.6} \cdot \text{A18.4} \cdot \text{E19.0} \cdot \overline{\text{E23.1}} \cdot \overline{\text{M144.1}} \cdot \text{E20.6} \cdot \text{M132.3} \cdot (\text{E30.5} + \text{M133.2} \cdot \text{E30.2}) \cdot \overline{\text{M21.2}} \cdot \overline{\text{M22.2}} \cdot \overline{\text{M23.2}} \cdot \overline{\text{M129.3}} \cdot \text{E7.1} \cdot \text{E7.3} \cdot \text{E7.5}$$

The first possible fault is that the NC start conditions are not satisfied, in which case the state function $S(x)$ = M132.4 = 0. From the above expression, the logical expression at a faulty state, i.e. $F(x)$, can be obtained as

$$F(x) = \overline{\text{A9.6}} + \overline{\text{A18.4}} + \overline{\text{E19.0}} + \text{E23.1} + \text{M144.1} + \overline{\text{E20.6}} + \overline{\text{M132.3}} + \overline{\text{E30.5}} \cdot \overline{\text{M133.2}} + \overline{\text{E30.5}} \cdot \overline{\text{E32.2}} + \text{M21.2} + \text{M22.2} + \text{M23.2} + \text{M129.3} + \overline{\text{E7.1}} + \overline{\text{E7.3}} + \overline{\text{E7.5}}$$

The components related to the terms that make $F(x)$ = 1, are the potential fault locations.

(2) Example for sequential diagnosis model

The SIEMENS U Series PLC uses the programming language STEP 5. A PLC program coded with STEP 5 is divided into the following blocks:

- Organisation Block (OB)
- Program Block (PB)
- Step Block (SB)
- Function Block (FB)
- Data Block (DB)

The sequential control is carried out in the SB. Each SB contains a machine operation command. Several SB's form a control sequence by linking together in a specified order. Here we take the operation of the tool-head exchange in the FFS-100-2 FMS as an example to explain the sequential diagnosis model. The tool-head can be attached to the spindle so as to change the feed direction of tools. The tool-head is exchanged frequently, and various faults may occur in the process.

The tool-head exchange sequence programmed in the SB is described in Table 2, which includes a series of actions, from removing the old tool head to fitting a new one.

Table 2
Operation sequence of tool-head exchange

Step	SB	Action description
1	SB117	Tool-head magazine moves to the position of the old tool-head
2	SB118	Z-axis returns to the reference
3	SB119	Y-axis moves to the position to exchange tool-head
4	SB121	Z-axis moves to the position to exchange tool-head
5	SB122	Adapter loosens and the old tool-head is put into tool-head magazine
6	SB123	Z-axis returns to the reference
7	SB124	Tool-head magazine moves to the position of new tool-head
8	SB125	Z-axis descends to the position to pick the new tool-head

Table 2
Operation sequence of tool-head exchange

9	SB126	Adapter clamps the new tool-head
10	SB131	Z-axis returns to the reference
11	SB132	Y-axis returns to the machining position
12	SB133	Tool-head magazine returns to the reference

Now let us consider the first two steps, SB117 and SB118, and assume SB118 is the current step. The start condition of SB118 is that SB117 is finished.

From SB117 in the relevant sequential control program of PLC, we know that the start conditions of the current action (SB118) are:

$$C(t-1) = \mathrm{M143.0} \cdot \mathrm{M143.1} \cdot \mathrm{M160.1} \cdot \mathrm{M165.0} = 1$$

This identifies that:

- the internal cooling oil is stopped (M143.0=1);
- the spindle blower is turned off (M143.1=1);
- step running is enabled (M160.1=1);
- automatic operation of tool-head magazine is enabled (M165.0=1).

From the SB118 program we know that the control command of this step is:

$$I(t) = \mathrm{M227.4}$$

The start conditions of the next step (SB119) are:

$$C(t) = \mathrm{M159.0} \cdot \overline{\mathrm{M158.4}} \cdot \mathrm{M165.2} \cdot \mathrm{M134.5} \cdot \mathrm{E18.4} \cdot \overline{\mathrm{E18.5}} \cdot \overline{\mathrm{E38.2}} = 1$$

This identifies that:

- the middle variables are cleared (M159.0=1);
- Z-axis is at its reference position (M158.4=0);
- the tool-head magazine has moved to its right position (MM165.2=1);
- the spindle has been oriented (M134.5=1);
- the protection door for tool exchange manipulator is open (E18.4=1 and E18.5=0);
- the C-axis has returned to its reference position (E38.2=0).

These conditions are also used to determine whether SB118 is finished.

If $C(t\text{-}1)=1$ and $I(t)=0$, then the control command of step 2 is received. When $I(t)=1$ and $C(t)=0$, the current step is not finished. From the expression below we know that the components associated with the terms that make C(t)=1, are the potential fault locations:

$$\overline{C(t)} = \overline{\mathrm{M159.0}} + \mathrm{M158.4} + \overline{\mathrm{M165.2}} + \overline{\mathrm{M134.5}} + \overline{\mathrm{E18.4}} + \mathrm{E18.5} + \mathrm{E38.2} = 1$$

4. Conclusions

Manufacturing systems present an important domain for diagnostics applications. The development of advanced diagnostic techniques and systems can help to minimize downtime and maintain an efficient output. This is a need common to all manufacturing enterprises. The diagnosis models in this paper are developed to meet this need.

There are diagnostic functions available in modern controlled manufacturing systems. However, these diagnostic functions are still limited and need further development. The prospects are greater where larger investments are concerned, as the cost of a fault is higher.

The combination of the LFC– and the SCP–based diagnostic models offers significant advantages in accuracy and speed of identification and classification of faults in complex systems. Cost effective industrial applications have been shown on flexible manufacturing systems.

Good results have been achieved from this work. Future work will address the following areas:

- refine the reasoning algorithms, so as to improve their efficiency in diagnosis;
- investigate models that incorporate PLC control on continuous processes of the manufacturing systems, implementing a systematic integrated methodology for prediction, monitoring and diagnosis;
- define an embedded diagnosis system approach which will integrate the diagnostic models in the PLC's, so that faults can be diagnosed in real time.

References

[1] A.K.A. Toguyeni, E. Craye, J.C. Gentina, *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 4 (1996) 2774.

[2] B. Kim and G.S. May, *IEEE Trans. Components, Packaging, and Manuf. Tech., Part C: Manuf.*, 1 (1997) 39.

[3] E.L.J. Bohez and M. Thieravarut, *Computers in Industry*, 3 (1997) 233.

[4] E. Chevalier, J. AguilarMartin, G. BlanchiColomb, J.L. MesasLaserna, *Proc. IEEE Int. Conf. Fuzzy Systems*, 3 (1997) 1259.

[5] Y. Yao, X. Yang, P. Li, *AIAA/IEEE Digital Avionics Systems Conf. - Proc.* (1996) 479.

[6] Z. Fang, G. Feng, Z. Zhang, *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 1 (1996) 26.

[7] Y. Cai, *J. of Propulsion Tech.*, 1 (1997) 88.

[8] J.B. Kneale, *AUTOTESTCON* (*Proc.*), (1997) 100.

[9] J. Jarvis and D. Jarvis (eds.), *Software Engineering for Manufacturing Systems: Methods and CASE Tools, Chapman & Hall*, 1996.

[10] R.L .Kegg, *Annals of the CIRP*, 2 (1984) 469.

[11] J. Jarvis and D. Jarvis, *Proc. of 9th European Simulation Symposium*, (1997) 342.

[12] J. Plomp, P, Huskonen, E. Malm, *Proc. of Int. Conf. on Condition Monitoring and Diagnostic Engineering management*, (1997) 354.

[13] Z. S. Matthias, *Proc. of 6th Int. Conf. On Data and Knowledge Sys. for Manuf. and Eng. (DKSME'96)*, Tampe (Arizona), USA. (1996) 11.

[14] S.M. Alexander, C.M. Vaidya, J.H. Graham, *Computer-Elect. Engng*, 2 (1993) 175