<table>
<thead>
<tr>
<th>Title</th>
<th>Fault Diagnosis and On-Condition Maintenance of a Combined Heat and Power Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors(s)</td>
<td>Conroy, G. V.; Black, W. J.; O'Hare, G. M. P. (Greg M. P.)</td>
</tr>
<tr>
<td>Publication date</td>
<td>1989-04-20</td>
</tr>
<tr>
<td>Publication information</td>
<td>IEE Colloquium on Expert Systems for Fault Diagnosis in Engineering Applications [proceedings]</td>
</tr>
<tr>
<td>Publisher</td>
<td>IEEE</td>
</tr>
<tr>
<td>Link to online version</td>
<td><a href="http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=198287&amp;isnumber=5104">http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=198287&amp;isnumber=5104</a></td>
</tr>
<tr>
<td>Item record/more information</td>
<td><a href="http://hdl.handle.net/10197/4395">http://hdl.handle.net/10197/4395</a></td>
</tr>
<tr>
<td>Publisher's statement</td>
<td>Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.</td>
</tr>
</tbody>
</table>
Fault diagnosis and on-condition maintenance of a combined heat and power unit.

G V Conroy, W J Black and G M P O'Hare

Introduction

Recent economic trends, and growing market competition have necessitated a drift towards highly automated manufacturing techniques. The equipment commissioned in such environments is more often than not very expensive, sophisticated and intricate. The intricacy of such equipment makes it prone to failure. Since economic survival is often synonymous with the smooth operation of the equipment it is imperative that machine down time be kept to a minimum. Consequently, investment in Diagnostic and Preventive Maintenance systems makes economic sense.

The authors are involved with ongoing research in addressing the problem of constructing such a Diagnostic and Maintenance system for a particular client company. The company is involved in the manufacture of engine units installed at a range of sites throughout Britain. These units will provide a reliable and economic source of power, earning revenue from customers for the supply of energy rather than from the original sale of capital equipment. Consequently the cost of maintenance, repair and renewal of each of the engines is to be borne by our client company. It is therefore imperative to their business strategy that maintenance is effectively managed and its cost controlled.

Fault Detection

There are two sorts of detection involved in this particular enterprise, the detection of a real fault in some component of the system and the detection of below par behaviour of the unit. Real faults can often be detected by the odd behaviour of some system parameter, or by the condition monitoring system itself. The condition of a unit cannot normally be measured directly. The standard approach is to make indirect measurements of associated parameters and to compare these with the expected values of a healthy unit. Differences in the readings which are statistically significant are taken to be indicators of change in the condition of the unit under investigation. These differences are not easy to detect since differences can arise which are due to environmental factors unrelated to the health of the unit. An original intention of the company was to provide an 'ex-works' 'fingerprint' for each unit, representing its standard operating values. The intention underlying the fingerprinting was to avoid the inevitable variation between the behaviours of supposedly identical items. This approach was rejected in the early days because of the environmentally associated variation. Consequently an approach is being taken depending on the comparative changes of parameter trends which theoretically should behave in a consistently similar manner. Whenever such comparisons indicate abnormal behaviour an expert system attempts to diagnose the cause of the variation. Major faults can be picked up by the static detection of abnormal parameter readings, while off-tune condition is detected by the condition monitoring system. The fingerprint idea is still to be used but now referring to the related parameter trends, not to static behaviour.

G V Conroy, W J Black and G M P O'Hare are all members of the Department of Computation, UMIST, PO Box 88, Manchester, M60 1QD
Fault Diagnosis

Existing research has recognised that diagnosis is very much a deductive issue, whereby it involves identifying the cause of a particular problem given the occurrence of a specific symptom or set of symptoms. The majority of existing systems employ either symptom trees or cause-consequence diagrams to represent the inherent relationship between symptoms and causes. Our system uses the symptom tree approach embedded in a production rule expert system. A prototype version of a deep level system has also been tested.

The basic production rule system depends on the expertise elicited from the domain expert and is crucially dependent on his expertise in relating the displayed symptoms to their causes. This is very much an ongoing process since the unit in question is a new one specifically developed for the purpose. The expert has to verify for himself that his traditional knowledge still applies to the new unit. The system developed contains within it a database of historical information for each unit in the field, so that the expert mechanical engineer can check out trends and associations for all units.

Because the unit is new, it will inevitably arise that faults will arise that were not previously encountered, or foreseen, by the expert engineer, or that have not yet been incorporated into the system. In that case the deep level approach to diagnosis can be brought to bear. This deep level approach has the possibility of ultimately bringing the greatest benefits to the company since it depends purely on knowledge of system design and standard component behaviour, thus being easily brought on line whenever a new unit type is introduced, a situation already occurring.

Condition Monitoring

The approach taken by us in this respect is to continually monitor the behaviour of up to 32 parameters. Sensors on the unit record data values for every parameter and an on-board microcomputer carries out a preliminary assessment of the relevance of such data. Whenever exception conditions are recognised the micro will automatically dial up the central computer facility to request the expert system to diagnose the fault in question. In general the condition of the unit will not be that of an exceptional fault situation, but either the condition of normality or of a situation representing a degradation of performance. The latter situation will not cause the micro to invoke the central facility for immediate help but will be dealt with on the routine nightly call-up by the central facility. This nightly call-up is a procedure by which the central computer facility automatically dials up every unit, every night, and requests its data for the previous day. On receipt of the unit data the expert system is then invoked to act in its condition monitoring mode, whereby it is not seeking for a fault, but searching for evidence of degraded performance and the associated cause. The condition monitoring mode of operation is then one whereby the expert system is actively monitoring the behaviour of the unit for signs of going off-tune, followed by a recommendation for corrective action. The corrective action can be either that of suggested replacement of a part, or the scheduling of standard maintenance such as an oil change, etc. In the condition monitoring mode the expert system is actually acting as a fault prediction system. The system is designed to recognise non-critical poor performance situations before they become critical, which should reduce the cost of maintenance by taking appropriate preventive action and thus preventing a serious failure situation from arising.

The Expert System

The suitability of this problem for an expert system solution being appropriate is justified on several counts. The situation is one demanding considerable expertise for its solution; the expert mechanical engineer currently performing the task is needed elsewhere on other tasks, and the financial rewards are great for a successful system.
As mentioned already this system acts in two roles, that of a traditional fault diagnosis system and that of a condition monitoring system, with considerable overlap between the two roles. The knowledge base of associations between faults and causes and between off-tune performance and cause, is obviously a very similar relationship.

In many circumstances the difference between the two situations is one of degree rather than of kind.

The expert system is actually in two parts, one being the traditional shallow level expert system using the experiential associations of the expert, the other being the deep level system which is concerned with a model of the correctly functioning unit.

The shallow level system is implemented in Prolog. The choice of this language was because: no shell appeared to provide all the required functionality, interface to an existing database was required, knowledge representation issues were not clear cut initially and the developers were very familiar with the Prolog language. Speed of development was an initial requirement also, which suggested that the use of familiar tools was appropriate.

The term shallow is used because the knowledge possessed represents the 'compiled' knowledge of the expert, or his accumulated experience, with no understanding of the first principles upon which that knowledge is based. The rules are there because 'they work', the experience of the expert is the guarantee of this. Such a system cannot derive any conclusions, certainly not meaningful ones, about a new presenting situation.

The system is a 'rule based' one, that is the knowledge is expressed in rules of the form 'IF Condition(s) THEN Action(s). The Conditions are the sets of data readings which the domain expert has associated with the Actions of the rules. The Actions are of two forms, either they are final conclusions about some fault which the domain expert has identified, or they are intermediate level conclusions about which the system can reason further. The system reasons as far as possible with the data available, producing either a primitive fault as a conclusion or a generic fault (one representative of a group of faults).

A typical rule is:

\[\text{isa} (\_\text{Unit}_\text{id}, \text{Pump}_\text{failure}):-\]
\[\text{cert} (0.98),\]
\[\text{isa} (\_\text{Unit}_\text{id}, \text{Heat}_\text{recovery}_\text{cooling}_\text{fault}),\]
\[\text{TW1} (\_\text{Unit}_\text{id}, \text{high}),\]
\[\text{TW2} (\_\text{Unit}_\text{id}, \text{low}).\]

This is read as follows:

\[\text{isa} (\_\text{Unit}_\text{id}, \text{Pump}_\text{failure})\]: this means that the unit numbered \_\text{id} has a Pump_failure fault associated with it if the following conditions hold.

\[\text{cert} (0.98)\] indicates the degree of belief attached to this rule.

\[\text{TW1} (\_\text{Unit}_\text{id}, \text{high})\] indicates that the value of the parameter TW1 for unit numbered \_\text{id} is high, etc.

This is a generic fault, further rules are found in the knowledge base which will refine the fault to a more specific one.

The rules present are applied or 'fired' by a process known as pattern matching, the condition part of the rule forms the pattern to be matched, the match is made with the set of data readings supplied, if the match is successful the appropriate action is taken and the system cycles through the whole process again until no more matches can be made. Each action adds further data to the system allowing further new matches to be made.

The rules are held in the component of the Expert system known as the 'Knowledge Base', the actual inferencing or reasoning is carried out by a component called the 'Inference Engine'. The Inference Engine is also known as a 'Shell', holding the inference mechanism without the domain knowledge.

The Expert system supplied carries out the inferencing process by a technique called 'forward chaining', here the input data is used to trigger off rules to establish fault groupings, these are added to the knowledge base as new facts, the process then continues forward until a basic fault is identified. This technique allows early pruning of the search space of possible faults thus producing greater efficiency in the search for a possible fault. All possible faults are identified, not merely the most likely one, a ranking is produced to indicate the strength of belief in the conclusions reached.
The output of the Expert system is the full list of possible faults. The above refers to fault identification, however the system is intended to be used when the fault has not actually occurred but the unit may be tending to fail in that manner. Thus fault prediction is actually the rationale behind the Expert system’s use.

The deep level system prototype employs Qualitative reasoning techniques. The motivation for such an approach lies in the realisation that in many cases, empirical associations alone are not sufficient to describe the behaviour of systems. The deep level system implemented uses knowledge about the structure of the system to be examined together with information about the correct functioning of the system components. The functional behaviour of a component is independent of the actual system in which the component is embedded, and we can say that in such circumstances the system reasons from first principles. The relationships between the components are established by the fact that components share values for certain parameters, the connectivity of components is maintained in this manner without presupposing any inter-component behaviour. These readings refer to changes in the parameters, such as: they increase, decrease or remain constant.

A technique known as Constraint Suspension uses the structural knowledge, the component rules and the qualitative readings to predict faults, the basic idea for this approach is in [2], the technique works as outlined below.

The top level structure has each component checked out in turn for anomalous behaviour, each suspect component being added to a candidate list for further investigation. The system then in a hierarchical manner decomposes each suspect component into its own sub-components, which are checked out in an identical fashion. The decomposition continues until a base level or primitive component is reached which cannot be decomposed any further.

A component is checked out by removing it and its associated rules, or constraints, from the system. If the system now appears to behave in a satisfactory manner that suspended component is a candidate faulty one.

The structural knowledge is as below:

```
breakdown (heat_recovery_system,
    [oil_cooler(tW3,tW5a),
      ex_man_hex(...........),
      header_hex(...........),
      ex_hex(...........)]).
```

This implies that there is a breakdown of the structure named heat_recovery_system into its sub-components. In this case oil_cooler, ex_man_hex, header_hex and ex_hex. The parameters in parentheses after each sub-component, eg, tW3, tW5a, are the parameter readings which are available for that sub-component.

The functional behaviour of each component is supplied by the provision of a rule list for each component in the form:

```
rule_list (heat_recovery_system (tW1,.......),
    [mplus(tE6, tE1).......]).
```

This indicates that the component heat_recovery_system has a rule_list associated with it, which appears inside the following parentheses. A typical rule is quoted, mplus(tE6, tE1), this informs the system that the relationship between parameters tE6 and tE1 is governed by the mplus function, i.e. the two parameters are in a monotonic relationship to each other (they can only change in an identical manner).

Further Prolog rules define how such a function is evaluated.

The output from the system is in the form:

```
fault_found_in_component oil_cooler (tW3, tW5a)
fault_found_in_component ex_man_hex (......)
_Fault = [heat_recovery_system (......)]
```

The text is reasonably self-explanatory, the predicted faulty component is identified with its associated parameters following in parentheses.
The last component identified at a higher level is also output, here the main component possibly faulty is the heat_recovery_system with sub-components of oil_cooler,..., etc, being possible faults within that component.
The system being implemented in a hierarchic manner implies efficiency in the search for faulty components.

Conclusion

The expert system involved in fault diagnosis for a combined heat and power unit has been detailed in the above lines. A fuller description of the system design and how it is embedded in a full information system for the units involved appears within [1].
We conclude by stating that the traditional expert system approach need not be restricted to failure driven maintenance systems employing troubleshooting techniques. On the contrary predictive maintenance proves equally amenable to solution via an expert systems approach. Such systems by their very nature need to have the ability to employ deep reasoning when necessary. For this reason a causal or functional model of the engine is a prerequisite.

Acknowledgements

The authors would like to thank Dr. J. Packer, Mechanical Engineering Dept., UMIST and Mr. S. Almozaffar for their collaboration in this research. Thanks are also extended to our client company who owing to confidentiality must remain nameless.

References

[1] O'Hare, M P, Black, W J, Conroy, G V.

School of Engineering and Applied Sciences, Univ. of Sussex.