Monitoring instrumentation fault diagnosis and data interpretation

Autor(en): **Vann, Andy / Davis, John**

Objekttyp: **Article**

Zeitschrift: **IABSE reports = Rapports AIPC = IVBH Berichte**

Band (Jahr): **72 (1995)**

PDF erstellt am: **30.04.2024**

Persistenter Link: https://doi.org/10.5169/seals-54661

Monitoring Instrumentation Fault Diagnosis and Data Interpretation

Interprétation de données et diagnostique d'erreur dans les systèmes de surveillance

Dateninterpretation und Fehlerdiagnose bei Ueberwachungseinrichtungen

Andy VANN
Lecturer
University of Bristol
Bristol, UK

Andy Vann spent several years in industry before joining the University. His industrial experience includes contracting and highway design. His research has concentrated on the application of IT and AI to the interpretation of civil engineering monitoring data.

John DAVIS
Senior Lecturer
University of Bristol
Bristol, UK

John Davis is a Senior Lecturer in the Departments of Aeronautical and Civil Engineering at the University of Bristol. His research interests include work on hydraulics, telematics, civil engineering monitoring and Artificial Intelligence.

SUMMARY

This paper describes aspects of a knowledge-based system to interpret monitoring data. There has been a large amount of work in providing systems for specific monitoring tasks, such as in nuclear power station, human and structural health monitoring. However, to date there has been little work on support for general civil engineering monitoring tasks. The system proposed is intended to provide this general support, particularly for temporary monitoring programmes. As part of this, the system provides dynamic creation of rules to deal with user-generated models of instrumentation and signal conditioning.

RÉSUMÉ

Cet article décrit un système à base de connaissances pour l'interprétation des données expérimentales. Il y a eu beaucoup de recherche pour réaliser de tels systèmes permettant l'interprétation de données spécifiques, par exemple dans une centrale nucléaire, ou pour la surveillance médicale ou structurale. Il y a cependant peu de recherches réalisées pour la surveillance générale d'ouvrages de génie civil. Le système proposé offre cette aide, en particulier pour des programmes temporaires de surveillance. Le système propose un ensemble dynamique de règles afin de maîtriser les modèles d'instrumentation et de traitement de signaux.

ZUSAMMENFASSUNG

Der Beitrag beschreibt ein wissensbasiertes System zur Interpretation von Ueberwachungsmesswerten. Umfangreiche Entwicklungen galten speziellen Ueberwachungsaufgaben wie Kernkraftwerken, medizinischen und technischen Zustandsüberwachungen, doch floss bisher wenig Entwicklungsarbeit in die Unterstützung Ueberwachungsaufgaben im Bauwesen. Dieses System ist vor allem für temporäre Ueberwachungsprogramme gedacht. Ein Bestandteil ist die dynamische Erzeugung von Regeln zur Behandlung Anwender-generierter Modelle der Instrumentierung und der Signalverarbeitung.

1. INTRODUCTION

Monitoring is becoming increasingly important within civil engineering. Given this importance, there are significant problems with the process of monitoring which must be tackled. One of the most severe problems is that of data overload. Increasing the amount of monitoring increases the amount of data produced. To be useful, this data must then be interpreted so that it can be combined with other forms of engineering knowledge. In the Department of Civil Engineering at the University of Bristol we have been developing knowledge based systems to provide support for general monitoring activities. Much previous work has been carried out on providing knowledge based systems to interpret data from specific systems [1,2,3]. The amount of effort required to produce such a system is justified where the monitoring equipment is likely to form part of a permanent structural health monitoring system, however, to date little work has been carried out on generalised techniques for interpreting signals. This would be particularly useful where temporary monitoring programmes are set up for short periods of time. However, the work is also relevant to signal interpretation for long-term monitoring applications.

This paper describes a knowledge based system (IMCES) to interpret signals from civil engineering monitoring programmes. The system has been created using Kappa PC, an object-oriented KBS development tool. The aim of this work is to provide logging systems with local intelligence and to support data interpretation. This would help to reduce the interpretation load on engineers and also to reduce the amount of raw data logged by unsupervised systems. An overview of the work and the methods used to provide general signal descriptions have been reported elsewhere [4,5]. A brief restatement of this work is included below. This paper discusses how the knowledge about instrumentation is structured and how this knowledge is used for data validation and signal interpretation.

2. INSTRUMENTATION FAULTS AND DIAGNOSIS

2.1 General

Diagnosing instrument faults can be a time consuming and difficult business. During a monitoring programme the engineer is normally working from a set of records of instrument signals. Depending on how close the engineer is to the site and what level of support is available there, he or she may also have additional information about the system. For example, weather records and verbal reports of system behaviour. However, the engineer's first point of reference is often the electrical signals recorded from monitoring instruments. It is extremely important to identify and rectify faults in the instrumentation quickly so that data is not lost. The IMCES system is being developed to assist in this task.

The first interpretation task is to identify whether the signals are displaying any unusual characteristics. This has traditionally been carried out through the engineer looking at plots of the signals and identifying unusual trends or noise in the plots. 'Unusual' in this context means different from expectations based on three sources of knowledge:

- knowledge of the output ranges and characteristics of the instrumentation used
- knowledge of the physical characteristics of the system being monitored
- experience of the characteristics of the signals seen so far

If unusual behaviour is found, the engineer will try to use other sources of evidence to explain the behaviour. For example, the reports of system behaviour mentioned above or reports of visible faults in the monitoring system. The engineer will then decide whether the signals are a true record of the system behaviour or whether they could be due to some fault in the instrumentation and signal conditioning chain. Following this, the engineer may either revise his or her ideas of the state and behaviour of the system or visit the site and perform tests on the instrumentation to try to find and rectify the fault. A summary of the process for diagnosing instrumentation faults is shown in the flow chart in Figure 1.

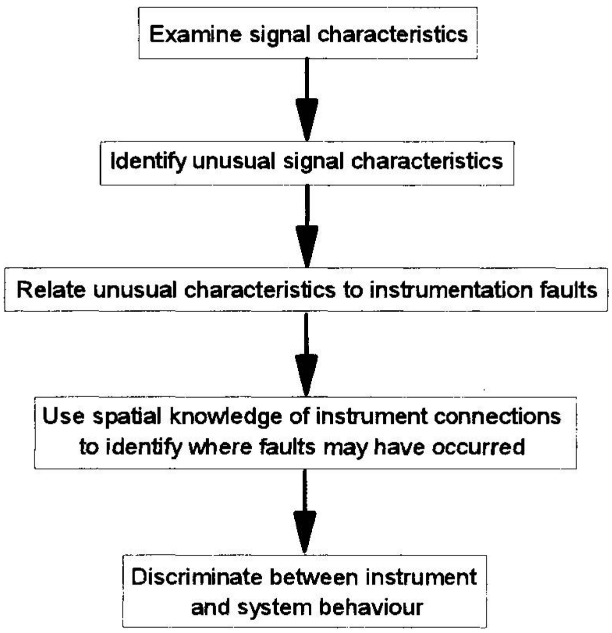

Figure 1 - Inferencing Process for Instrumentation Faults

There are clear benefits in providing some automatic data interpretation. Early automatic monitoring systems were limited to sending an alarm when the value of a particular signal crosses a given threshold. The position of the threshold would have been previously calculated from a numerical model of the system, No attempt would be made to diagnose the causes of the alarm, discern between instrument and system behaviour or draw together evidence from disparate sources. As mentioned above, work in the field of dam monitoring has tackled this issue [3]; we have been investigating the problem for use in general monitoring.

2.2 KBS Approach

We have emulated the traditional processes described above using a knowledge based system. Because the system is meant to be of general use, there are many problems to be solved. Firstly, in writing the system, we can have no knowledge of how it will be used. That is, for a given monitoring programme we do not know what types of instrumentation will be used, what types of signal conditioning will be used, how these will be connected together and what settings will be used. To overcome this problem, the user configures an object-oriented model of the instrumentation and signal conditioning as a first step in using the system.

There is a further problem in terms of the rules which will diagnose the state of the system. In a knowledge based system involved in monitoring a nuclear plant, say, all of the entities with which it has to work are known. The names of these entities can be written into the rules as the KBS is created. In our system, the rules must be created once the user has constructed the instrumentation model. It would be possible to write general rules which could access knowledge in the model at run time. However, the rules would need to follow long chains of reference through objects and would be very unwieldy. We have therefore chosen to include code to create rules dynamically as the system runs. The KBS therefore actually generates the diagnosis rules itself. These rules can then operate on the knowledge the user has entered about the types of signal, the types of signal conditioning and the connections between the signal conditioning objects. This approach also allows the rules to be linked to the objects in an object-oriented fashion. Objects in the system are responsible for managing the diagnostic rules which relate to them.

2.3 Limitations of Rule-Based Systems

A final problem is that of drawing together evidence from the signals with evidence from knowledge of the behaviour of civil engineering systems and reports of events affecting the system. The engineer-interpreter calls on information from non-signal sources to help diagnose the recorded behaviour. This information is either in the form of a model of the system or in the form of a stimulus to that system. For instance an inspection record forms a snapshot of the state of the system which can inform a crude model of the system. The model can be used to make inferences. A construction log contains both a time varying model and a record of stimuli. Weather records on the other hand are purely stimuli. This information has both spatial and temporal components which are necessary for successful interpretation. Combining these sources of information is a much larger problem because there is no accepted method of encoding engineering knowledge or representing system behaviour. Once again, in a specific application most types of event could be entered into the knowledge base. In a system for general monitoring this is not possible. A method of encoding the whole of civil engineering knowledge would be required. One approach to encoding engineering knowledge is product modelling. This attempts to provide a universal data format which "seeks to transfer the engineering intent which underlies ... graphical representations" [6]. However, product models have not yet been able to transfer knowledge between domains. It may be that we need an alternative form of representation for engineering knowledge to deal with this problem.

2.4 Knowledge Acquisition

Knowledge acquisition for the system was carried out through unstructured interviews with staff members in the Department of Civil Engineering at Unversity of Bristol, and with industrial experts in monitoring and instrumentation. Experience gained from trial monitoring programmes has also been incorporated into the system. The scope of the system is necessarily limited at present, and we aim to carry out further work to expand the knowledge base.

3. GENERAL SIGNAL CHARACTERISTICS

Signals within the IMCES system are modelled by instances of the *Signal* class. At the class level, a set of slots are defined which represent the types of characteristic the signal can have. Examples include *ConstantlyDead* and *StationaryMean*. These slots are inherited by instances of the class. The slots are of Boolean type and their values are instantiated by rules. Signal processing is carried out to calculate statistical parameters of the signals and the rules then operate on these values. A fuller explanation of this is given in [5]. Once this process has been completed each signal instance has a set of slots describing signal characteristics, each with a TRUE or FALSE value.

4. INSTRUMENTATION HIERARCHY AND FAULT INFORMATION

4.1 Knowledge Representation

Knowledge about instrumentation and signal conditioning is contained within the system. Kappa PC uses a conventional object-oriented method of knowledge representation. Variables are defined as slots within objects and may be specified at the class or instance level. Object hierarchies exist to describe both types of instruments and signal conditioning. The object hierarchy for instrumentation is shown in Figure 2.

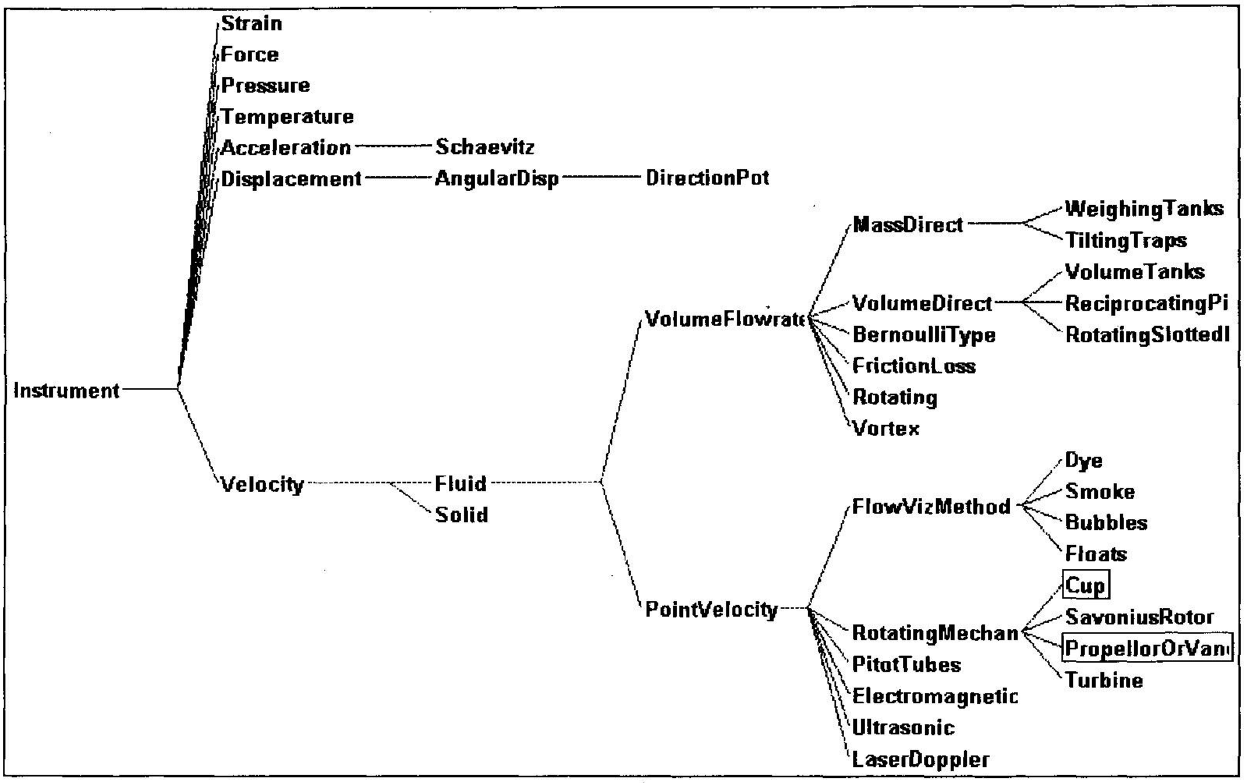

Figure 2 - Instrument Hierarchy

The hierarchy is divided according to the function and type of instrumentation. The objects within both instrument and signal conditioning hierarchies contain various kinds of knowledge:

- Signal type (pulse, analogue voltage, analogue current, digital voltage). This is represented as a slot value in the object class for each type of instrument.
- Output range and gain. This is represented as a slot value in the object class for each type of instrument.
- Expected signal characteristics. Each instrument object class contains a slot with a list of expected signal characteristics. Each signal has Boolean slots which represents whether these characteristics are true for that signal.
- Fault types and symptoms. These are represented as slots in the instrumentation and signal conditioning object classes.
- Connections between instrumentation and signal conditioning. These are represented as 'pointer' slots whose value is set within each individual instrument and signal conditioning instance. Kappa PC does not allow C++ style pointers (that is, variables which hold a memory address). The slots therefore contain the name of the connecting object and these names can be used to reference the objects.

The signal type and output range of the instruments is usually specific to each type. This is therefore defined at the leaves of the hierarchy tree. Expected signal characteristics are likely to be defined further up in the hierarchy. For example, no cup anemometers would be expected to show a negative speed, but this would not be true of all point velocity measuring instruments. The knowledge about cup anemometers therefore refines the knowledge about point velocity measuring instruments. Information about the types of fault from which instrumentation suffers may appear at a number of different levels. This is very important in a system which is meant to be generally applicable and which is expected to grow. Any new instrument which is added to the hierarchy should inherit the features of the class of instruments to which it belongs.

4.2 Rule and Knowledge Bases

An outline of the rule and knowledge bases required is shown in Figure 3. Our work so far has concentrated on the transducer, conditioning and connection rules and the instrument and signal conditioning behaviour. The inferencing process starts with examining the behaviour of individual instruments, signal conditioning units and the connections between them, shown in the centre of the diagram. Each of these may in itself be a hierarchy, for instance, there may be a number of channels passing through a filter box which need to be examined individually and then examined as a group - "do all the channels flag up behaviour x". The position rules (top left of the diagram) then use what is effectively spatial logic to examine where in the signal chain evidence of unusual behaviour occurs. The use of careful planning of the signal chain can help the interpretation process. For instance, if a system uses anemometers and accelerometers and has two filter boxes, passing half the signals from each transducer through each box will make filter box performance much easier to determine. The two ideal generalisations of this technique are firstly to provide redundant measurements by having more transducers than are needed and secondly by ensuring that each transducer signal path has a unique route through the signal conditioning chain. This will also help in the discernment between instrumentation system and observed system failure. If we can clear each item in the chain of malfunction then we can suggest that what is recorded is the actual behaviour of the monitored system.

The temporal rules (lower left) determine the time scale of behaviour. The problem of temporal reasoning is handled by dividing the signal records into windows. Within a window signal processing is used to determine whether changes in characteristics are sudden or gradual - are they step changes or drifts. The windows on the data are themselves tagged with a time stamp, so that the temporal problem is effectively transformed into a spatial one - 'is this event in front of or behind that one'. The data windows are also referenced to an event log so that unusual behaviour can be linked to reports of behaviour or faults noted in the log.

Firing the rules in this first part of the diagnosis produces evidence for examination by the rules which attempt to discern between instrumentation system behaviour and monitored system behaviour (the flow from left to right in Figure 3). Although this part has yet to be implemented some of the issues are becoming clear. While avoiding the temptation to limit the classification of behaviour to only known characteristics, there are some general rules we can apply. We have however to begin building up a hierarchical classification of civil engineering systems similar to the Instrument Hierarchy. This would include classes such as bridges with sub classes of perhaps suspension, cable stayed, glued segmental, with some further classification on size and use, such as long, medium or short span and foot, road or rail. However, this is not the only possible way of encoding the information. It may be that a representation based upon connectionist systems would be more appropriate for general use.

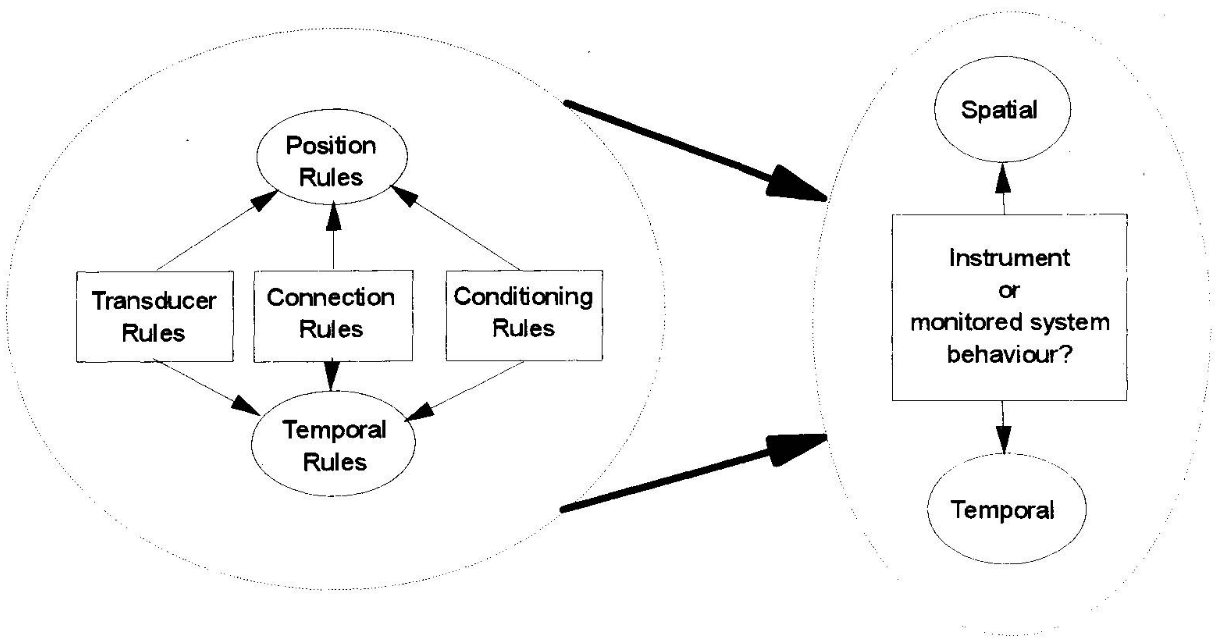

Figure 3 - Interpretation Rule and Knowledge Bases

The discernment rules will look at both frequency and time domains. We are fortunate that in the civil engineering domain the frequencies present in the systems are generally much lower than those expected in a faulty monitoring system (although faulty earthing can mislead an investigator). We can also make deductions based on the positions of the instruments. If the monitored system is actually behaving in a particular way, then several instruments will pick up the phenomenon in their own particular way. This of course again suggests that the placing of the instruments is of vital significance if we want to diagnose successfully.

As with the instruments the detection of unusual behaviour can be followed by some degree of diagnosis. If the frequency of vibration of a support cable on a cable stayed bridge drops then we can deduce that it is no longer carrying as much load. Such a database of information will take many years to build up, but it will not be superseded. It can be continually added to and made richer, even corrected as our understanding of systems improves. We must explore whether the same information can be used by those who are working on design and construction support systems. The links with product modelling in construction also need to be explored.

5. EXAMPLE

To demonstrate the inferencing process we will use an example taken from the Kessock Bridge field monitoring programme [7,8]. During the monitoring programme, a fault developed in the power supply to a filter box for some of the signals. The filters are used to remove unwanted high frequency components which can affect later signal processing. This fault manifested itself in a variety of ways including adding spikes into the signals and causing their mean level to drift and jump. One ten minute accelerometer record for the period during which the problem was occurring is shown in Figure 4.

Examination of the signal shows a clear jump in the mean level at the start of the record. The accelerometer from which this trace was taken recorded vertical movements of the bridge, and one would therefore expect the mean level to be about 1g. In practice, this offset was removed by the signal conditioning to make best use of the voltage range available. The mean level should therefore be approximately zero and a jump in the mean level is hard to explain. At this point, considerations about the behaviour of the system being monitored become important. A physical explanation for the shift could be that the accelerometer had suddenly rotated. This could be because the instrument had shifted on its mountings (unlikely) or that the bridge itself had moved (hopefully even more unlikely). A jumping mean level is therefore an unexpected characteristic for an accelerometer trace.

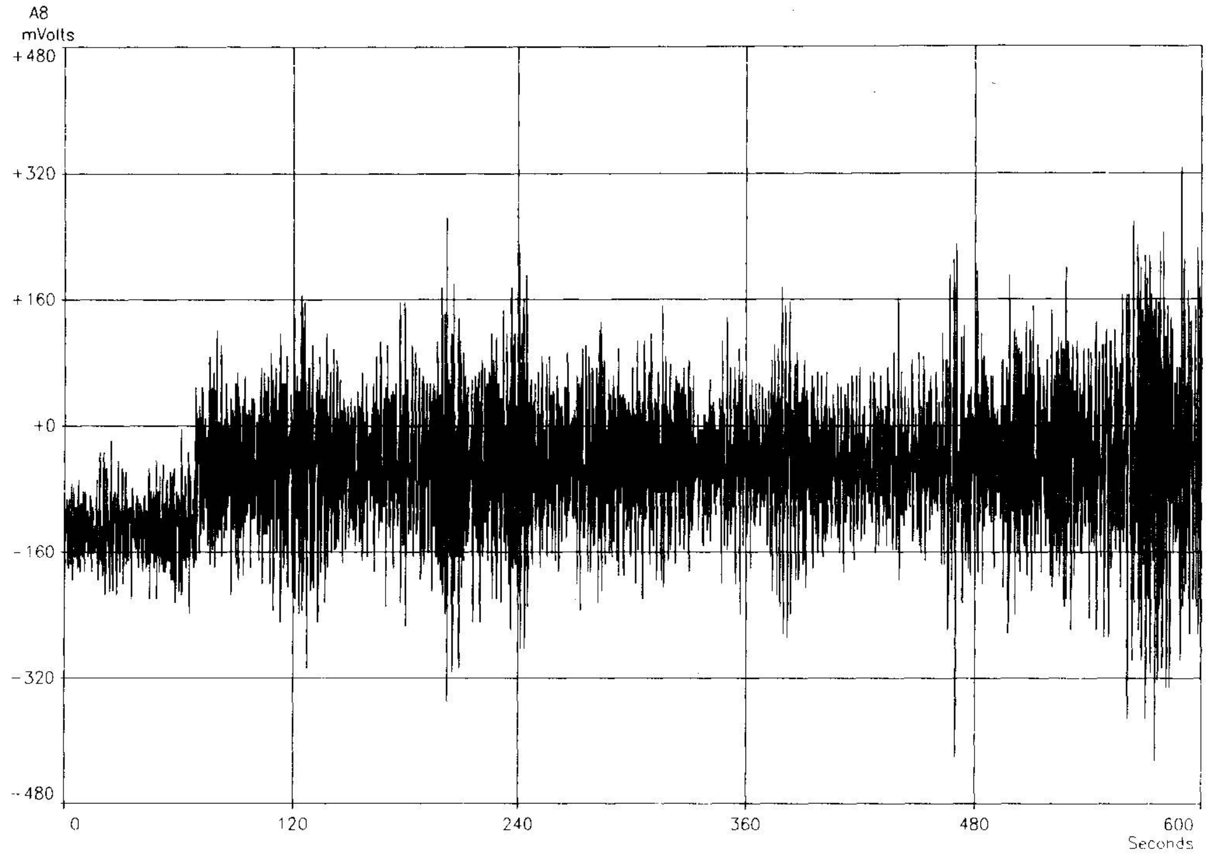

Figure 4 - Accelerometer Record Showing Change in Mean Level

As explained above, the system carries out signal processing to calculate statistical measures of the signal and the rule system then operates on these measures to assign TRUE or FALSE values to the signal characteristics. Following this, the diagnostic rules attempt to infer instrumentation faults from these characteristics. The inferencing sequence in which these rules work is shown in Figure 5. Rule names are shown in italics and the objects and slots on which they operate are shown in normal text. In Kappa PC's inferencing mechanism, when the value of a slot is altered it is placed on a list of slots for consideration by the rule base. The inferencing process therefore propagates forward until all slots have been considered or all rules have been used. The bodies of the rules operating during the inferencing process are given below.

Rule Acc1_1:
If GetNthElem(Acc1DatSmoBy20thDiffMaxabsrec:Values, 1) / Acc1Dat:Max > 0.001
Then Acc1:JumpingMean = TRUE;

Rule JumpingMean:
x|Signal: If x:JumpingMean And Not(Member?(x:ExpectedAttributes, JumpingMean))
Then x:HasUnexpectedAttributes = TRUE;

Rule SuspectChannel:
x|Signal: If x:HasUnexpectedAttributes Then EnumList(x:SigConChain, y, y:Suspect = TRUE);

Rule SuspectSigConModule:
x|SigConChannel: **If** x:Suspect **Then** x:SigConObject:Suspect = TRUE;

Rule FilterBox1_Chan_1_1:
If FilterBox1_Chan_1:Suspect And FilterBox1:Suspect And (
FilterBox1_Chan_1:Signal:JumpingMean And Not(Member?(
FilterBox1_Chan_1:Signal:ExpectedAttributes, JumpingMean)) Or
FilterBox1_Chan_1:Signal:IntermittentlySpiky And Not(Member?(
FilterBox1_Chan_1:Signal:ExpectedAttributes, IntermittentlySpiky)) Or
FilterBox1_Chan_1:Signal:ChangeableMean And Not(Member?(
FilterBox1_Chan_1:Signal:ExpectedAttributes, ChangeableMean)))
Then FilterBox1:PowerSupplyFault = TRUE;

cc1DatSomBy20thMaxAbsRec:Values
Acc1_6
Acc1:JumpingMean
JumpingMean
Acc1:HasUnexpectedAttributes
SuspectChannel
FilterBox1_Chan_1:Suspect
SuspectSigConModule
FilterBox1:Suspect
FilterBox1_Chan_1_1
FilterBox1:PowerSupplyFault

Figure 5 - Inferencing Chain For Instrumentation Fault Diagnosis

The first and last rules in this sequence, *Acc1_6* and *FilterBox1_Chan_1_1* are created dynamically by the system at run time. They use the knowledge entered by the user in the instrumentation model. The object classes for each signal conditioning and instrumentation class contain a method which is triggered when the user completes the model. This method is essentially a template which uses the instance names entered by the user and generates the diagnostic rules. Knowledge relating signal characteristics to fault types is therefore stored implicitly within this method. The rules generated are free standing in that they are stored in the general rule base, but the instrumentation objects store the names of the rules so that they can be updated if the instrumentation model is altered. *Acc1_6* and *FilterBox1_Chan_1_1* therefore use the names of objects directly whereas the remaining rules use pattern matching to operate on all objects of a certain class.

The first rule, *Acc1_6* examines the statistical parameters of the signal shown in Figure 4 and decides that the signal has a jumping mean. The second rule, *JumpingMean* looks at the characteristics displayed by the signal and compares them with those expected. It then decides that the signal is displaying unexpected characteristics and flags this fact. The third rule *SuspectChannel* flags all channels through which the signal passes as suspect. The fourth rule, *SuspectSigConModule* flags the module containing those channels as suspect. The final rule *FilterBox1_Chan_1_1* compares the unexpected characteristic with those caused by a power supply fault and suggests that the filter box could be suffering from this fault. In practice, we would not decide that the power supply was faulty on the basis of a single channel. We would use our knowledge about all the channels in the filter box and the connections to other pieces of equipment. To improve the inferencing mechanism we therefore need to use an uncertainty handling mechanism to weigh the evidence from all the channels and assess the likelihood that a power supply fault has occurred. We are planning to tackle this stage next.

6. CONCLUSIONS

The IMCES KBS is successful in diagnosing faults in the limited domains we have used so far. However, to produce a general system we need to expand the amount of knowledge in the hierarchy and validate it against other data sets.

ACKNOWLEDGEMENTS

The work reported in this paper was funded by the Science and Engineering Research Council.

REFERENCES

1. COMERFORD JB, BLOCKLEY DI, DAVIS JP. A knowledge based system to interpret results from pulse-echo integrity tests. *Proc. Instn. Civ. Engrs*, Part 1, 1991, 90, Feb., 189-204.

2. MARTIN JHM, DAVIS JP & BLOCKLEY DI. Inference of Embankment Dam Safety by Combining Processed Geotechnical Instrument Data With Stored Engineering Knowledge, pp235-243 in *Proceedings of Conference on Geotechnical Instrumentation in Civil Engineering Projects - purpose - performance - interpretation*, Nottingham, ICE, London 1989.

3. LAZZARI M AND SALVANESCHI P. Artificial Intelligence and Monitoring of Dams. pp71-76 in *Applications of Artificial Intelligence in Structural Engineering*. Proceedings of the 1st Workshop of the European Group for Structural Engineering Applications of Artificial Intelligence, EPFL, Luasanne, Switzerland, March 21-22 1994.

4. DAVIS JP, VANN AM, Intelligent Monitoring of Civil Engineering Systems. *Applications of Artificial Intelligence in Engineering VII - Proceedings of AIENG92*. Southampton: Computational Mechanics Publications, 1992.

5. VANN AM AND DAVIS JP. IT Support for Civil Engineering Monitoring, *International Journal of Construction Information Technology*, Vol 1, No4, Winter 1993.

6. WATSON AS. To Product Models and Beyond. *SERC N&N Workshop*, 24-26th May 1993.

7. OWEN JS, VANN AM, DAVIS JP, BLAKEBOROUGH A. The Prototype Testing of Kessock Bridge: Response to Vortex Shedding. *WES Conference* 19-21 Sept 1994 (in press).

8. VANN AM, OWEN JS, DAVIS JP, BLAKEBOROUGH A. The Prototype Testing of Kessock Bridge: Long Term Monitoring of Response to Wind Excitation. *WES Conference* 19-21 Sept 1994 (in press).