Development of a Hybrid Knowledge-Based System for Condition Monitoring and Diagnosis of Rotating Machinery

Siyu Zhang

A Thesis
in
The Department
of
Mechanical Engineering

Presented in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada
August 1995

ISBN 0-612-05078-5

ABSTRACT

Development of a Hybrid Knowledge-Based System for Condition Monitoring and Diagnosis of Rotating Machinery

Siyu Zhang, Ph.D.
Concordia University, 1995

A new approach to hybrid knowledge-based systems (KBSs) for rotating machinery monitoring and diagnostics is developed, incorporating the latest developments in AI techniques and expert systems technology. This approach employs the vibration signature as the diagnostic signal, and neural networks are used to perform numerical processing of diagnostic data to enable condition identification and classification of fault patterns, and the quantification of fault or malfunction development. Neural network solutions to the above two problems, particularly the solutions using Self-Organizing Maps (SOM) are sought and obtained. For trending and quantifying fault development, a method which employs multiple-index based trend analysis is proposed and implemented. To address this problem, an appropriate Self-Organizing Mapping algorithm is developed from first principles.

A prototype expert system, designated RMD-KBS (Rotating Machinery Diagnostic Knowledge-Based System), is designed and fully developed. This is an on-line diagnostic system in which both the symbolic and numerical processing are deeply coupled. The

"Object-Oriented Programming (OOP) technique" is employed in such a way that its computational advantages are exploited in RMD-KBS. In addition, the RMD-KBS is designed and developed with the ability to possess a number of the most desired computational and functional capabilities of a diagnostic KBS for industrial applications. In order to validate the RMD-KBS diagnostic expert system and to assess its performance, experimental data from a class of real-life industrial machine systems have been collected. Comparison of the diagnostic results provided by the RMD-KBS with the faults known to be present, established the efficiency, accuracy and superiority of the proposed prototype system.

Dedicated to

my wife, Hongyu Yu
our son, Hanpo Yu
and our Parents

Acknowledgements

The author wishes to express his deep gratitude to his supervisors Dr. R. Ganesan, Dr. G. D. Xistris and Dr. S. V. Hoa for their guidance, encouragement, and constructive criticism throughout the course of this research work. Thanks to Dr. Hoa, the Chairman of the Department for his involvement in the thesis work, and for arranging financial support under difficult circumstances. Sincere thanks to Dr. Xistris for his timely help and assistance in the editing of this thesis. In particular, Dr. Ganesan deserves extra appreciation for spending those continuous long hours with the author discussing the research work, thesis presentation and the key publications, which are the outcome of this work. A word of appreciation to Dr. T. S. Sankar, who initiated many of the concepts involved in this work and provided indispensable guidance throughout the formative stages of this research.

I like to thank very much the research associates and graduate students in the Intelligent Diagnostics System Laboratory (IRIS), in particular Mr. A. Jaura, Mr. M. El-Karmalawy, Mr. F. Zhou, Mr. Y. Sun and Mr. M. N. Moorthy, for their helpful discussions and comments, and also for providing me with a pleasant work atmosphere.

The research reported in this thesis is a part of a major project on System Diagnostics and Reliability under the IRIS Network of Centres of Excellence Programs funded by the Government of Canada.

It would not have been possible for me to carry out my doctoral studies without the help, encouragement and moral support provided by my wife, Mrs. Hongyu Yu. The author wishes to thank the members of his family, back home in China, for their love, encouragement, understanding and moral support that paved the way for the successful completion of this research work.

CONTENTS

LIST OF FIGURES xii

LIST OF TABLES xix

ACRONYMS xx

NOMENCLATURE xxi

1. INTRODUCTION

1.1 Machinery Monitoring and Diagnosis 1

1.2 KBS Technology for MMD 3

1.3 Current Issues in Computer Aided MMD 6

1.4 Potential Applications of Neural Networks in MMD 8

1.5 Scope and Objectives of the Thesis 9

1.6 Organization of the Thesis 11

2. ASPECTS OF KNOWLEDGE-BASED SYSTEMS AND NEURAL NETWORKS

2.1 Knowledge-Based Systems and Applications 14

2.2 KBS Design Techniques 17

2.2.1 Knowledge Representation 18

2.2.2 Inference and Control Strategies 19

2.2.3 Reasoning with Uncertainty and Imprecision 20

2.2.4 Knowledge Source and Acquisition 20

2.2.5 Learning 21

2.2.6 Programming Language and KBS Development Tools 22

2.2.7 Interfaces 22

2.3 Existing Systems 23

2.4 Artificial Neural Networks and Hybrid Systems 26

2.4.1 Aspects of Neural Computing 26

2.4.2 Hybrid Systems 31

3. DESIGN OF RMD-KBS

3.1 Diagnostics of Rotating Machinery 32

3.1.1 Diagnostics Based on Vibration Signals 32

3.1.2 Signal Feature Extraction 33

3.1.3 Identification of Machine Fault Pattern 34

3.1.4 Quantification of Fault Development 36

3.1.5 Information Needed in Rotating Machinery Diagnosis 38

3.1.6 Diagnostic Procedure 39

3.1.7 Knowledge Required in Diagnostic KBS 40

3.1.8 Importance of Learning Ability in Diagnostic KBS 42

3.2 Design of RMD-KBS 43

3.2.1 Outline of the Design 45

3.2.2 On-Line Monitoring and Diagnostic Capability 46

3.2.3 Coupled Symbolic and Numerical Processing 49

3.2.4 Incorporation of Neural Networks 52

3.2.5 Knowledge Modelling 57

3.2.6 Knowledge Representation 58

3.2.7 Diagnostic Strategy 62

3.2.8 Inference Schemes 65

3.2.9 Data Normalization 66

3.3 Discussion 67

4. FAULT CLASSIFICATION USING NEURAL NETWORKS

4.1 Neural Networks in Fault Detection Systems 70

4.2 Pattern Clustering and Classification 71

4.3 Self-Organizing Mapping Algorithm 74

4.4 Bearing Condition Classification 77

4.5 Rotor Condition Classification 85

4.6 Condition Classification with BPNNs 93

4.7 Discussion 99

5. MULTIPLE-INDEX BASED TREND ANALYSIS USING NEURAL NETWORKS

5.1 Multivariable Trend Analysis in MMD 103

5.2 Multi-Dimensional Regression and SOM Algorithms 106

5.3 Nonfunctional Mapping in Regression Problems 112

5.4 A New SOM Algorithm Based on Weighted Euclidean Distances 121

5.5 Multiple-Index Based Trend Analysis of Bearings 127

5.6 Multiple-Index Based Trend Analysis of Gear Boxes 134

5.7 Discussion 139

6. **IMPLEMENTATION OF RMD-KBS**

6.1 Introduction 143

6.2 The KBS Development Tool and Its Object-Oriented System 145

6.3 Knowledge Source and Acquisition 149

6.4 Outline of the RMD-KBS 151

6.5 Sensor and Data Acquisition System 154

6.6 Database 155

6.7 Numerical Analysis Routines 161

6.8 The Multiple Windowing User Interface 167

6.9 Inference Mechanism 168

6.10 Knowledge Bases 169

6.10.1 The SET-UP Module 172

6.10.2 The BROWSER Module 174

6.10.3 The MONITOR Module 175

6.10.4 The MENTOR Module 176

6.10.5 The MAIN-KBS Module 179

7. **PERFORMANCE AND APPLICATIONS OF RMD-KBS**

7.1 Initialization of RMD-KBS 191

7.2 Performance of Monitoring and Diagnosis Tasks 195

7.3 Training the Neural Networks 199

7.4 Applications to Industrial Machine Systems 201

8. CONCLUSIONS AND FUTURE WORK 241

REFERENCES 248

APPENDIX

A.1 Definition of the Diagnostic Indices 264

A.2 The Knowledge Editor and Confidence Factors of LEVEL5 270

LIST OF FIGURES

2.1	An overall classification of AI programs	16
3.1	The architecture of the hybrid RMD-KBS	47
3.2	The structure of a partitioned hybrid system	54
3.3	The diagnostic strategy used in RMD-KBS	63
3.4	The fault classification approach employed in RMD-KBS	64
4.1	Typical signal from an accelerometer for a defect-free bearing	79
4.2	Typical signal from an accelerometer for a bearing with ball defects	79
4.3	Typical signal from an accelerometer for a bearing with the inner ring sliding at the shaft	80
4.4	Typical signal from an accelerometer for a completely-damaged bearing	80
4.5	Data samples for bearing condition classification and the initial positions of the four weights	83
4.6	Data samples for bearing condition classification and the final positions of the four weights	84
4.7	Typical vibration spectrum for fault-free rotor system	86
4.8	Typical vibration spectrum for slight unbalance	86
4.9	Typical vibration spectrum for rotor unbalance	87
4.10	Typical vibration spectrum for resonance	87

4.11 Typical vibration spectrum for misalignment with unbalance 88

4.12 Typical vibration spectrum for misalignment 88

4.13 Typical vibration spectrum for oil whirl in the journal bearing 89

4.14 Data samples for rotor condition classification and the initial positions of the seven weights 91

4.15 Data samples for rotor condition classification and the final positions of the seven weights 92

4.16 Feed-forward back-propagation neural network 94

4.17 Feed-forward back-propagation neural network with one output unit 94

5.1 Nonfunctional mapping formed by the original SOM algorithm 110

5.2 Functional mapping formed by the CTM algorithm 110

5.3 The quadratic curve $f(x)$ and sampled data points $f_d(x)$ 114

5.4 A 20-unit network obtained using the SOM algorithm 114

5.5 Regression curve obtained using the SOM algorithm based on an 8-unit network 116

5.6 Regression curve obtained using the SOM algorithm based on a 16-unit network 116

5.7 Regression curve obtained using the CTM algorithm based on a 20-unit network 118

5.8 Regression analysis involving an "S" curve using the CTM algorithm based on a 20-unit network 118

5.9 Regression analysis of quadratic functions using the newly developed algorithm based on a 20-unit network 128

5.10 Regression analysis involving an "S" curve using the newly developed algorithm based on a 20-unit network 128
5.11 The four univariate functions of bearing life obtained from the on-line vibration signal 131
5.12 Projections on to the hyperplane of 8 knots of a nonfunctional map obtained using CTM algorithm 131
5.13 Random initial positions of 10 knots of a neural network used in the new algorithm 132
5.14 Performance of the new algorithm in training a network of 10 knots with a scaling factor s=2.0 132
5.15 Network of 50 knots trained by the new algorithm to fit 9 data points (s=1.4) 137
5.16 Disordered 30-knot network trained by the CTM algorithm 138

6.1 Outline of the structure of a LEVEL5 object 147
6.2 The structure of RMD-KBS 152
6.3 The structure of the database of RMD-KBS 157
6.4 The list of the elements of a database class named **dB3 sensor** 159
6.5 Windows display of the vibration signal and the spectrum using an EXTERN program 164
6.6 A plot shown by the external program of training the SOM for trend analysis 166
6.7 The connectivity between the modules of RMD-KBS 170
6.8 Windows that appear at the beginning of the SET-UP module 173

6.9	The knowledge tree that shows the structure of the knowledge base of the MONITOR	177
6.10	The hierarchical structure of the rules and the information flow in MAIN-KBS	180
6.11	The strategy of generating the hypotheses	183
7.1	The information flow within the RMD-KBS	205
7.2	The Central Panel of RMD-KBS	206
7.3	The window of SET-UP module for machine system registration	207
7.4	the window of SET-UP module for defining machine subsystems	208
7.5	Acquisition of the bearing specifications using SET-UP module	209
7.6	Acquisition of the information about sensors using SET-UP module	210
7.7	The definition of sensor locations for helping the information acquisition	211
7.8	List of the settings for signal processing and the threshold values of indices	212
7.9	The window for browsing and managing the data files	213
7.10	The BROWSER module for initiating the monitoring and diagnosis	214
7.11	A graph presenting the machine system to be diagnosed	215
7.12	The display of Central Panel when a machine has been selected	216
7.13	The graphical display of the current values of the monitoring indices	217
7.14	The display of MONITOR module that shows the data samples and spectrum	218
7.15	The display of MONITOR module that shows	

the orbit of the centre of rotor shaft 219

7.16 The display of the data samples and spectrum from the file AH3 220

7.17 The display of the orbit from files AH3 and AV3
using MONITOR module 221

7.18 The display of both the hypotheses and the diagnostic results
for the compressor unit 222

7.19 The hypotheses and the diagnostic results for a compressor
with unbalance problem 223

7.20 The information flow during the training and utilization of ANNs
for fault detection 224

7.21 The MENTOR module of RMD-KBS which performs
the training of neural networks 225

7.22 The query for the specifications of ANN training 226

7.23 A window displaying the training data samples and
the initial positions of the seven weights 227

7.24 Results of training the SOM network with seven weights 228

7.25 Values of the seven weights of the SOM network
for machine fault identification 229

7.26 Windows of the routine for training Back-Propagation neural networks 230

7.27 The display of the data samples and the initial weights
of the SOM network for trend analysis 231

7.28 The display of the results of training the SOM
network for trend analysis 232

7.29 The information about the sump pump system stored in the RMD-KBS 233

7.30 The diagnostic results about the condition of the motor of the sump pump system 234

7.31 The vibration signal from the motor of the sump pump system and the corresponding spectrum 235

7.32 The information about the bearing test machine stored in the RMD-KBS 236

7.33 The diagnostic results of the bearing condition 237

7.34 The information about the pump of boiler feed pump system 238

7.35 The diagnostic results of the pump condition 239

7.36 The vibration signal from the boiler feed pump and its spectrum 240

A.2.1 The Knowledge Editor of LEVEL5 OBJECT shell and its Database Editor 273

A.2.2 The Object Editor of LEVEL5, that is used to declare and modify classes and instances 274

A.2.3 The Windows Editor of LEVEL5. By using it a KBS developer can specify the style, size and location of a window of the user interface under design 275

A.2.4 The Display / Forms Editor of LEVEL5, that is used for the KBS developer to create a new display 276

A.2.5 The first step in using the Display / Forms Editor: to select a pre-designed form in the list 277

A.2.6 The second step in using the Display / Forms Editor: to define a system provided form for string query from the end-user of the KBS 278

A.2.7 The Rule / Demon / Method Editor of LEVEL5 279

A.2.8 The Agenda Editor of LEVEL5, that is used to select a number of class attributes as agenda 280

A.2.9 The Knowledge Tree of LEVEL5 which is an interactive environment for building, debugging, and maintaining the knowledge bases 281

A.2.10 The Values List of LEVEL5, that catalogues the current values, confidence factors (CFs), facets, and attribute names in the knowledge base 282

A.2.11 The History Report of LEVEL5, which creates a text file that is a chronological record of the events of a processing session 283

A.2.12 The session Monitor of LEVEL5, that follows and displays the current status of the reasoning season at run-time 284

LIST OF TABLES

2.1	Existing KBSs for rotating machinery diagnosis	24
5.1	Accuracy of bearing life estimation using different algorithms	135
6.1	The list of machine faults and malfunctions	182
7.1	Motor and pump specifications	202
7.2	Pump specifications	204

ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BP	Back-Propagation
BPNN	Back-Propagation Neural Networks
CF	Certainty Factor
DB	Data Base
ES	Expert System
FNN	Fuzzy Neural Network
IE	Inference Engine
KB	Knowledge Base
KBS	Knowledge-Based System
KE	Knowledge Editor
MMD	Machinery Monitoring and Diagnosis
NAR	Numerical Analysis Routines
OOP	Object-Oriented Programming
RMD-KBS	Rotating Machinery Diagnostic Knowledge-Based System
SOM	Self-Organizing Mapping (or Maps)
UI	User Interface

NOMENCLATURE

AM	arithmetic mean value in frequency domain
AR	average residual
AX	absolute mean value of vibration signal in time domain
$b(t)$	learning factor for training self-organizing mapping networks
b_0	initial value of the learning factor
b_s	final value of the learning factor
CR	crest factor of vibration signal in time domain
DX	standard deviation of vibration signal in time domain
d	desired output for training back-propagation networks
e_{aR}	average residual in percentage
e_a	average estimation error in percentage
f_{inn}	the frequency of ball passing inner race
f_{out}	the frequency of ball passing outer race
f_s	data sampling frequency (samples per second or Hz)
f_{spin}	the frequency of the ball spin
GM	Geometric mean value in frequency domain
$H(t)$	neighbourhood function for training self-organizing mapping networks
h_j	output of the j-th node in the hidden layer
$input_i$	input value from neuron i
IP	impulse factor of vibration signal in time domain

K	Kurtosis value of vibration signal in time domain
MM	matched filter root mean square value in frequency domain
net_j	activation value of the neuron j
out_j	output value of neuron j
PP	peak-to-peak value of vibration signal in time domain
$\boldsymbol{R}$	fault-symptom correlative relationship
$\boldsymbol{r}$	vector of the coordinate of a neural in the feature space
RMS	root mean square value of vibration signal in time domain
s	scaling factor
s_0	number of neurons in a self-organizing mapping network
t_m	maximum number of the training iterations
VX	variance of vibration signal in time domain
V_{jk}	weight from neuron j toward neuron k of feed-forward networks
W_{ij}	weight from neuron i toward neuron j of feed-forward networks
$\boldsymbol{W}$	weight vector of a neuron of self-organizing mapping networks
$\boldsymbol{X}$	feature (symptom) set
$\lvert x \rvert_{max}$	maximum absolute value of vibration signal in time domain
$\boldsymbol{Y}$	failure (machine condition) set
Y	indicator of machine condition
y_k	output of the k-th node in the output layer
$\boldsymbol{Z}$	data pairs of X and Y
α, $\alpha(H, t)$	learning rate for training self-organizing mapping networks
Δt	data sampling interval

δ	errors
θ, τ	bias parameters (internal offset)
λ, μ	learning rate for training back-propagation networks

CHAPTER 1

INTRODUCTION

1.1 Machinery Monitoring and Diagnosis

The rapid advance of technology and the pressure of worldwide competition, have placed an increased emphasis on industrial productivity and efficiency. Plants and machine systems have become massive and complex and further, they have been largely automated. To prevent catastrophic failures and minimize breakdowns in modern industrial machine systems is a challenging problem. The unexpected breakdown of machines is financially costly and it may lead to major accidents, such as airplane crashes and nuclear power station failures. Preventive maintenance through continuous monitoring can contribute to the avoidance of these problems.

Monitoring, Trouble-Shooting, and Diagnostics are terms (see Pau, 1981; Brawley et al. 1989) used by many instrument manufacturers and system management engineers. They represent the analysis processes and techniques which dynamically determine the health or performance of machine systems, and support machinery performance control and effective maintenance. The field of Machinery Monitoring and Diagnosis (MMD) has been receiving widespread attention because it enables greater plant availability, lower maintenance or replacement costs, increased productivity and prolongation of service life to be achieved without affecting operational safety. Also, it leads to achieving the required modifications of mechanisms so as to minimize both the noise due to excitations and the transmission of vibrations.

The main tasks of MMD are: to detect abnormal occurrences; to identify the component and root cause of an existing failure: to estimate the risk of damage and remaining life of a machine or machine component; and to provide diagnostic and maintenance data on processes ranging from manufacturing to new product design. Successful achievement of these objectives however, hinges on three major features of a condition monitoring system: (i) collection of diagnostic data that can adequately represent the occurrence and development of each and all of the faults or malfunctions in machine components: (ii) development and use of efficient data handling and information processing techniques that result in precise fault detection; and (iii) automation of the process supervision systems.

The primary goal of monitoring and diagnosis is to recognize the faults developing inside machines without disassembling or stopping the machine systems. Several different on-line measurements are being used to provide diagnostic information depending upon the intended use of the condition monitoring program. Recent studies have clearly shown (Collacott, 1977; Haddad, 1991), that on-line measurements of temperature, pressure, force, stress, noise levels, acoustic emissions and vibrations constitute reliable, complete and comprehensive diagnostic data. Moreover, each of these elements adequately reflects, at an early stage, the occurrence and development of different types of faults or malfunctions in the machine system. Numerous sophisticated diagnostic techniques have been developed and applied in industry. These range from oil debris monitoring (Lukas and Anderson, 1989), infrared measurement, ultrasonic flaw detection (Haramis et al, 1989), and reliability estimation (Xistris, 1979), to data analysis methodologies for various kinds of signals. Among all the signal processing techniques, vibration analysis has emerged as the most successful for mechanical system monitoring and diagnosis. In most of the present-day machinery diagnostic systems, the condition of machinery, particularly rotating machines (pumps, blowers, compressors, fans, turbines, gearboxes etc, usually rotor- or shaft-bearing systems), are both identified and classified through the analysis of on-line vibration signals (Lipovszky et al, 1990; Eshleman and Jackson, 1992). Also, machinery vibrations that arise due to faults or malfunctions have long been used as established performance indicators in monitoring systems for rotating machine elements such as bearings (Taylor, 1980) and gears (Randall, 1982; Zhang et al, 1986; Liu et al, 1991). For these reasons, on-line vibration signals have been used as the primary source of diagnostic data to monitor and diagnose the condition of rotating machinery in this research.

Development of computer-based monitoring and diagnostic systems (Lyon and Dejong, 1984; Zhang, 1990; Xu et al, 1991) is an important area in machinery monitoring and diagnostics, and it is the only way to perform automatic diagnosis. Design and development of condition monitoring and diagnostic techniques which can detect and classify both the presence and the severity of potentially dangerous faults causing malfunctions in mechanical machinery, are crucial to the automation of monitoring systems. There is a continuing need for new practical techniques in this field to improve the performance of machinery monitoring and diagnostic systems. In this area, Artificial Intelligence (AI) technology, especially the Knowledge-Based System (KBS) approach has been introduced into MMD to develop intelligent diagnostic systems.

1.2 KBS Technology for MMD

Diagnosis is the process that identifies the fault causing malfunction in a given system (Mussi and Morpurgo, 1990). The diagnosis and maintenance of complex mechanical systems have been considered to be an interdisciplinary engineering task. The key characteristic of machine fault diagnosis is both a knowledge-intensive and an experience-based activity (Pau, 1981; Doherty et al, 1994). Human beings possess an ability to process complex types of information and draw inferences. Expert diagnosticians and equipment operators are often more efficient than present-day automated MMD systems. Current diagnostic systems are usually designed as an integration of the measurement system, data collector, signal analyzer (or signal processing software) and data base (Good et al, 1989; Ramakrishna et al. 1992). In such a system the diagnosis is performed by means of numerical analysis and the results are numerical values of the monitoring parameters. One serious problem of these diagnostic systems is the lack of pattern classification methods to accomplish machine fault identification. Therefore, numerical analysis based monitoring systems (known as conventional diagnostic systems) cannot yield definite conclusions on the precise nature of faults. The final decision regarding machine condition has to be made by human experts based on their knowledge and experience.

Knowledge-based systems (Garcia and Chien, 1991) are computer programs that explicitly represent the knowledge of an expert in a particular subject domain, and emulate the human reasoning process by means of symbolic manipulation. A more specific definition of KBS and a survey of KBS design techniques are given in Chapter 2. Knowledge-based system methodology is very helpful for diagnosis (Wright and Bourne, 1988). From a consideration of the following major characteristics and problems of MMD (Zhang and Sankar, 1993), it is easy to recognize the advantages of using the knowledge-based approach to solve MMD problems.

1) Signals, condition attributes and symptoms of machinery malfunctions vary

from case to case. No single feature of the vibration waveform has been found to yield consistently accurate results for the full range of applications encountered in practice. The most important point is that the use of a number of parameters in combination, will give the best indication of machinery condition (Carlson et al, 1988; Tranter, 1989). Since valuable human expertise can be manipulated in knowledge-based systems, such systems are able to handle arduous tasks and consider multiple factors to carry out decisions or suggestions.

2) Machine condition is reflected or represented by machine signals and test results, where the indications are frequently indirect, unclear and unstable. Hence, diagnosis is usually based on uncertain and incomplete information. KBSs can process uncertain, incomplete and even inconsistent information.

3) Rigid and stiff regulations may not work well in machine condition evaluation. Diagnosticians prefer to make decisions based mostly on relative changes (Serridge, 1989), but not only to compare the absolute values of diagnostic indices with standards, such as ARP1587 (SAE, 1981), VDI 2056 (Stronach et al, 1984). The structure of a KBS and knowledge-based approach make it feasible to arrive at decisions based on heuristics and past experience stored in memory, and make it possible to update the stored knowledge.

4) A serious problem is the shortage of diagnosis experts or that they are seldom readily available at the specific location where they are needed (Minami and Hirata, 1991). A machine operator or engineer needs considerable time to acquire experience and knowledge about MMD methodologies, machine malfunctions and maintenance. KBSs are available all the time in the absence of a human expert, and they can deliver the knowledge from an expert's domain to the end-users.

Therefore, design of knowledge-based diagnostic systems is of current interest to diagnosticians in both the academic and industrial communities. Diagnosis has been concerned with a human activity and is becoming an important application area of KBSs.

1.3 Current Issues in Computer Aided MMD

In AI research, many methods have been developed and new approaches continue to emerge. Meanwhile, several knowledge-based diagnostic systems have been implemented successfully for various applications over the last decade. Detailed discussion of AI and KBS design aspects will be given in the following chapters.

Two major achievements in diagnostic KBS development are noted. The first is that the present design of diagnostic KBSs emphasizes representation of human diagnostic knowledge and heuristic reasoning. Some basic KBS methodologies, mainly symbolic processing schemes, have been employed in the reported systems. Advanced KBS techniques, such as fuzzy logic (Zadeh, 1988; Kandel, 1992) and blackboard systems (Tailor, 1988), have been utilized in a few diagnostic KBSs also. The second achievement is that various approaches for diagnosis have been proposed for different applications. These include the decision tree method (Pattipati et al. 1986), fault tree (Parsaye and Lin, 1987), goal/subgoal scheme (Allard and Faemmerer, 1987), cause-effect and problem-symptom diagrams (Cantu-Ortiz, 1991), case-based reasoning (Feret and Glasgow, 1992), and model-based diagnosis (Reiter, 1987; Iwamasa et al. 1992).

The research presented in this thesis is to design a rotating machinery diagnostic KBS. As pointed out in the next chapter, several KBSs have been developed to conduct rotating machinery diagnosis. Their key features can be summarized as follows: they are rule-based systems; about half of them are consultative systems and others are on-line or non-consultative systems; some of them utilize certainty factors to process uncertain information, and fuzzy measurement has been used in one of them; not much attention towards numerical processing of on-line machine signal has been paid in these designs. The overall assessment of these KBSs is that they have been designed in a conventional manner. New AI and KBS methodologies have not been employed to date in rotating machinery monitoring and diagnosis. Further advancement and new methodologies are clearly required.

Several critical shortages of effective methods in rotating machinery diagnostic KBS design have been identified through recent research work.

1) Computational schemes for manipulating symbolic information are typically less effective in dealing with numerical data (Kanal and Raghavan, 1992), but numerical processing is essential in rotating machinery monitoring and diagnosis besides symbolic processing.

2) It is difficult to manage large size rule bases and to keep the rules consistent (Dillon and Laughton, 1990), even though the rule-based method is widely used and suitable for diagnostic knowledge representation.

3) Knowledge acquiring and refinement is a bottleneck in KBS development (Diaz, 1990).

4) Lack of methods to perform machine fault identification and classification, and for fault developing trend analysis hamper the design of diagnostic KBSs.

It is difficult to solve the above problems with conventional symbolic processing techniques. One of the most promising new methods gaining prominence with AI researchers is the Artificial Neural Network (ANN) technique. As Zurada (Zurada, 1992) noted, "Perhaps the most likely applications for neural networks will be those involving classification, association, and reasoning." Therefore, the ANN technique is considered to be particularly useful in problems involving condition monitoring and diagnosis.

1.4 Potential Applications of Neural Networks in MMD

Artificial neural networks or neural networks (Rumelhart and McClelland, 1986; Zurada, 1992) are massively parallel interconnected networks that consist of basic computing elements called neurons. The ANN structure is based on our present understanding of the biological nervous systems (Lippmann, 1987). Neural networks can deal with noisy and approximate data; learn automatically from training data; learn incrementally; adapt to a changing environment; generalize to situations they have not seen before; and execute very quickly once they are trained (Rich, 1990). Presently, most applications of neural networks are in the areas of speech and image recognition.

In MMD, neural networks could be useful for signal feature extraction, fault pattern identification and classification, and fault development trend analysis in a conceptual consideration. Few cases of machinery diagnosis using neural network algorithms have been reported that show the introduction of ANN methods in MMD domain (see Chapter 2). Neural networks possess a high level of adaptivity that can not be obtained from completely-analytical or numerical procedures and further, they provide a data-based heuristic approach to machine condition monitoring and diagnostics of machine systems. A neural network can automatically store knowledge about the faults or malfunctions in the machinery system being monitored by learning from the historical data and possess characteristics of associative memory. ANNs thus have the capability to learn and store complex information about abnormal conditions of machinery from the faults identified and classified in the past via the associative memory skills. This associative diagnostic capability makes the neural networks superior to the conventional methods of machinery fault diagnostics. These advantages of the ANN technique can be seen in both the published literature on the application of ANNs to solve MMD problems and the research presented in this thesis.

Several hybrid systems incorporating ANNs to ES have been prototyped, such as a hybrid ES for performance monitoring and diagnosis of telecommunication cable distribution networks (Senjen et al, 1993), and another hybrid KBS for ship machinery on-line diagnosis (Weiskopf et al. 1990),

1.5 Scope and Objectives of the Thesis

This thesis concerns itself with the development of a new approach to hybrid knowledge-based systems for rotating machinery diagnostics. The new approach takes into account the nature and characteristics of condition monitoring and diagnostics of rotating machinery systems. Further, it incorporates the latest developments in AI techniques and expert systems technology so as to result in a more efficient diagnostic KBS for industrial applications. In the conventional approach to develop a KBS, the knowledge engineer acquires both information and knowledge from human experts in terms of IF-THEN statements and then formulates them into the rules of the KBS. This approach poses many difficulties that hamper the development of diagnostic KBSs. In the present work, it is recognized that neural networks can be used to acquire knowledge without the extraction of IF-THEN statements from human experts, through the training process. Based on this, a new approach in which neural networks are employed in conjunction with the KBS technology, is proposed and followed. In this approach, neural networks are utilized for numerical processing of diagnostic data. The numerical processing encompasses the monitoring and diagnostic problems of 1) the condition identification and classification of fault patterns, and 2) the quantification of fault or malfunction development. These two fundamental problems that are frequently encountered in machinery monitoring and diagnostics are formulated into the corresponding mathematical problems of clustering and feature extraction, and trend analysis. To trend and quantify the development of faults in a machinery system, the new approach of multiple-index based trend analysis is proposed and implemented. Neural network solutions to the above two problems, particularly the solution using Self-Organizing Maps (SOM), are sought and obtained. The efficiency and performance of SOM networks in solving clustering and feature extraction problems are demonstrated in this thesis. Further, it has been recognized that the second fundamental problem mentioned above corresponds to the regression analysis. The connection between Topological Maps such as SOM networks and regression analysis has been well established (Cherkassky and Lari-Najafi, 1991). So, the SOMs are deployed in the present thesis to solve the second fundamental problem of MMD. To trend and quantify the fault development, since both the existing algorithm using SOM and the Constrained Topological Mapping algorithm are inefficient and inaccurate, a new Self-Organizing Mapping algorithm is developed from first principles.

Based on this new approach, a prototype expert system, the RMD-KBS (Rotating Machinery Diagnostic Knowledge-Based System), is designed and fully developed. This is developed as an on-line diagnostic system and further, both the symbolic and numerical processing are deeply coupled. The key capability of a diagnostic system, the learning ability, which none of the existing diagnostic KBS possesses, has been embedded into the RMD-KBS through neural networks. Moreover, a new scheme called "Object-Oriented Programming (OOP) technique", which has also not been utilized in any of the existing rotating machinery diagnostic KBSs, is employed in RMD-KBS. Suitable design strategies have been developed for these purposes. In addition, the RMD-KBS is designed and developed to possess a number of the most desired computational and functional capabilities of a diagnostic KBS for industrial applications. The expert system development tool known as LEVEL5 OBJECT has been utilized. Knowledge obtained from published literature on rotating machinery diagnostics and the author's industrial experience has been embedded. In order to validate the newly-developed diagnostic expert system, and to assess its performance, experimental data from a class of real-life industrial machine systems have been collected. Diagnostic vibration signals obtained from actual machine systems, which contain known induced faults or malfunctions, have been processed by the RMD-KBS. The diagnostic results are compared with the induced faults to validate the operation of the prototype system.

1.6 Organization of the Thesis

A brief outline of the applications of AI technology in KBS, the important issues related to the design and development of KBSs, the merits and disadvantages of existing diagnostic KBSs and artificial neural networks is provided in Chapter 2. This chapter also contains a literature survey of existing methodologies for KBS design (including the conventional symbolic processing based approaches), the existing rotating machinery diagnostic KBSs and new trends towards employing neural network approaches in AI.

The new approach to the development of a hybrid diagnostic KBS for rotating machinery monitoring and diagnostics, is described in Chapter 3. The conceptual design of the RMD-KBS and the approach of embedding various special features, that are not available in any of the existing rotating machinery diagnostic systems, are detailed in

Chapter 3 (Zhang et al. 1994b; 1995a). In addition, in this chapter, (i) the nature and principal characteristics of rotating machinery monitoring and diagnostics are summarized in a way that guides the design of RMD-KBS, and (ii) the basic problem of machinery monitoring and diagnostics is formulated into a pattern clustering and classification problem, and a function regression and trending problem.

In Chapter 4, a new approach which uses SOM networks to achieve machine fault identification and classification (Zhang et al, 1994a; 1995c) is developed and described. The new approach of employing the multiple-index based trend analysis (Zhang et al. 1995b; 1995d) for both the quantification of fault development and the prediction of the condition of machine components, is developed and described in Chapter 5. Also, the new self-organizing mapping algorithm that has been developed to solve the above trend analysis problem (Zhang et al. 1994c; 1995e) is described in detail in this chapter.

The details regarding the implementation of the conceptual design that is developed in Chapter 3 are provided in Chapter 6. In this chapter, the characteristics and the details regarding the knowledge base, inference mechanism, various modules of knowledge-base etc., of the prototype KBS, the RMD-KBS are summarized. The flow of diagnostic information and the information about the physical and operational parameters of the monitored machine system, is given in detail in Chapter 7 along with the details about the functioning of the RMD-KBS. Details of the experiments on real-life industrial machinery systems, that are intended for the validation and the assessment of performance of the RMD-KBS, are also provided in Chapter 7. In addition, the diagnostic results obtained based on the vibration signals are provided in this chapter. Chapter 8 contains the conclusions and the summary of the results achieved in this thesis, along with suggestions for future work.

The Appendices contain details of signal processing methods, and of the expert system shell LEVEL5.

CHAPTER 2

ASPECTS OF KNOWLEDGE-BASED SYSTEMS AND NEURAL NETWORKS

Issues related to the design and development of knowledge-based systems, as well as, artificial neural networks are briefly discussed. A survey of existing works on these topics is provided, and various important aspects on the automation of condition monitoring and diagnostics of rotating machinery, are described in detail. In addition, existing knowledge-based systems for rotating machinery diagnostics and the design approach used, are presented.

2.1 Knowledge-Based Systems and Applications

The field of Artificial Intelligence (AI) encompasses a range of several disciplines, including problem solving, robotics and vision, knowledge-based systems (expert systems), natural language understanding, machine learning etc., that are realized through computer systems or programs. These systems frequently use various methods of symbolic inference and exhibit performance that may be called "intelligent". More details of basic AI concepts can be found in (Garcia and Chien, 1991) and (Barr, 1981-1989).

An Expert System (ES) is a computer program which is capable of offering solutions to specific problems in a given domain or which is able to provide advice, both in a way and at a level comparable to that of human experts in a given field (Keim and Nordmann, 1989). This is the outcome of several years of research in cognitive science and AI (Waterman, 1986).

Nowadays, it is common practice to use the term Knowledge-Based Systems (KBSs) to denote an extended category of AI programs which are able to access knowledge in a way of some marked degree of separation - or some definite decoupling - between their knowledge and control portions. The term Knowledge-Based Systems is generally employed to indicate information systems in which some symbolic representation of human knowledge is applied, usually in a way resembling human reasoning. Knowledge-based systems hence, denotes a wider range covering expert systems and some other AI programs such as vision systems, thought support tools, natural language systems etc (Garcia and Chien, 1991). When the particular method of representing the knowledge uses logical implications (or rules), and these are the bases for the inferences, these systems are designated as "rule-based systems". An overall classification of AI systems is shown in Figure 2.1.

A knowledge-based (or expert) system is typically made up of at least three components: a Knowledge Base (KB), an Inference Engine (IE), and a Global or Working Memory. The knowledge base contains the domain knowledge which is employed to solve problem. The working memory is used to store the current context and information gained from the user of the system. The inference engine uses the domain knowledge together with acquired information about a problem to provide a solution.

The separation of knowledge from inference and control is probably the most important concept to come out of I research. This is also the most notable difference between KBSs and conventional computer programs which are essentially of the sequential instruction set and execution type. The effect of this new approach is to make the system simulate human "thinking", and to be flexible in integrating new knowledge

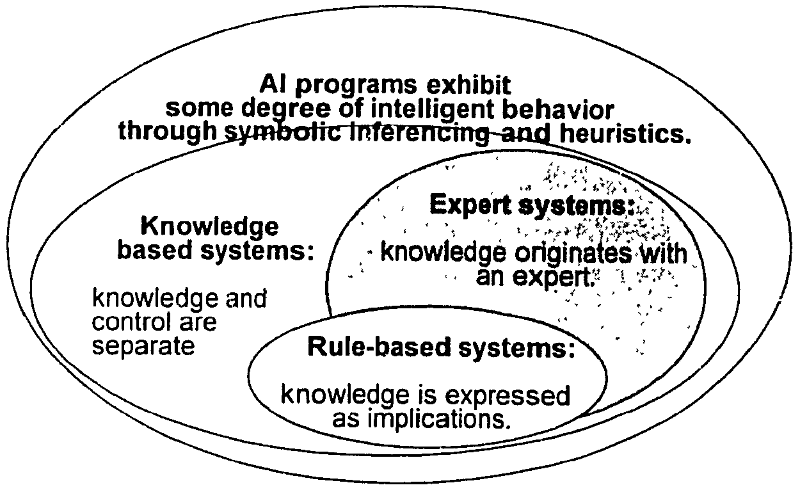


Fig. 2.1 An overall classification of AI programs [Garcia 91].

incrementally into the existing store of knowledge, and capable of letting anyone who can provide knowledge, create and update the program (Garcia and Chien, 1991).

KBSs have been applied in many fields. They have been used in training, planning, computer vision, computer aided design/manufacturing, scheduling, monitoring, configuring, diagnosis, speech understanding, debugging etc. Diagnosis has emerged as one of the most interesting and challenging applications of KBS technology. Among all of the early developments, the most influential one might be MYCIN, best described as a heuristic, rule-based system with certainty factors that aid in the diagnosis of and therapy for bacteraemia and meningitis (Buchanan and Shortliffe, 1984). Numerical implementations have been reported, such as the ESR project of the European power generation industry (Maile et al, 1991), the KBS applications on steel manufacturing in Australia (Lee, 1991), the five ESs developed in Mexico for equipment repair (Cantu-Ortiz, 1991), LES for troubleshooting in tele-communication switching systems (Laffey et al, 1986), FLES for aircraft on-board monitoring (Ali et al, 1986), SHARP 1 for consumer electronics repair (Persad and Wei-Muddeen, 1991), ATE for radar fault isolation (Chao et al, 1986), PRA for nuclear power plants (Parsaye and Lin, 1987) etc.

2.2 KBS Design Techniques

KBSs are more complex in design and more inclusive of techniques than conventional computer software. Many factors have to be considered in the development of a KBS. In general, the considerations deal with aspects of the system architecture, knowledge representation and modelling, inference and control, processing uncertain information, knowledge acquisition and learning, style of the user interface, explanation, data base structure, programming language or development tool, wide range applicability (adaptability), financial and time costs, and so on. The available techniques in some of the above areas, especially those involved in the design of diagnostic KBSs, are investigated in this section. More details on the approaches that are employed in the present development will be given in subsequent chapters.

2.2.1 Knowledge Representation

Knowledge is more complex than information, and more valuable. Knowledge is stored in the knowledge base of a KBS in a variety of ways. A representation of knowledge is a combination of data structures and interpretative procedures that will lead to knowledgeable behaviour.

The basic knowledge representation approaches defined in *The Handbook of Artificial Intelligence* (Barr, 1981-1989) are: state-space search, logic (or first-order logic), procedural representation, frames, scripts, analogical representation, production rules and semantic nets. A popular scheme that appeared in late 1980s, is object-oriented representation (Ramamoorthy and Sheu, 1988; Dillon and Laughton, 1990).

Some of the proposed approaches for diagnostic knowledge representation have been constructed from the above basic schemes, such as decision tree (Pattipati et al, 1986), fault tree (Parsaye and Lin, 1987), goal/subgoal representation (Allard and Faemmerer, 1987), cause-effect and problem-symptom diagrams (Cantu-Ortiz, 1991), models-based systems (Reiter, 1987; Iwamasa et al, 1992), and so on (see also Diaz, 1990; Garcia and Chien, 1991). The rule-based system has been most widely used to accomplish diagnostic knowledge representation.

Because the pattern of knowledge is very complex and variant, the knowledge representation methods currently used do not perform this task as well as expected (Diaz, 1990). There is much work underway in trying to develop robust knowledge representation schemes. Hybrid schemes, where more than one scheme is utilized to represent knowledge (Kanal and Raghavan, 1992), appear to be powerful new methods.

2.2.2 Inference and Control Strategies

Inference engine is the portion of a KBS that contains the inference and control strategies. It directs the search in the knowledge base to arrive at a conclusion appropriate to the context (Garcia and Chien, 1991). Inference is the process by which new facts are derived from established facts. The control is a group of methods used by the inference engine to determine the order in which reasoning occurs.

Forward chaining and backward chaining (Barr, 1981-1989) are two basic and widely used inference mechanisms in rule-based systems. Forward chaining (data-driven) goes forward through the chain of condition-action rules, from conditions that match the data in the working memory towards conclusions that may be established from these conditions. Backward chaining (goal-driven) goes backward through the chain of rules, from conclusions that have to be established towards the conditions that are necessary to establish these, or the data that has to be acquired. Some KBSs have combined these two reasoning algorithms to provide flexibility for different tasks.

The control and schedule of knowledge activation can be accomplished through a specification of the rule properties to affect the sequence of fire or fail rules (Level5, 1990a), such as "fire the first rule", "fire all rules in a group", "forget the current value of a variable" etc.

2.2.3 Reasoning with Uncertainty and Imprecision

Uncertainty in KBS is present in two forms: the data received by the KBS may not be completely reliable, may contain noise, or may be inconsistent with other data or with expectations; and the hypotheses and conclusions arrived at by the KBS may be inaccurate because of uncertain evidence (as in noisy data), or because t' ‹ knowledge about the process itself is uncertain, imprecise, and incomplete (Diaz, 1990). Therefore, a KBS must be able to function with missing, incorrect, inconsistent, approximate, or irrelevant data, and with uncertain, imprecise and incomplete knowledge.

Attempts towards representing uncertain facts (information and data) and imprecise knowledge have led to various formulations, including the Bayesian probability formulation (Sood et al, 1985; Kadesch, 1986), Dempster-Sharfer calculus (belief measures) (Dempster, 1967; Sharfer, 1976; Smets, 1990), confidence factors (CF) (Buchanan and Shortliffe, 1984; Feldman and Ballard, 1991) and fuzzy logic (Zadeh, 1988; Pedrycz, 1989; Xu and Zhang, 1990). Reasoning strategies have also been used in KBSs, such as fuzzy reasoning (Kandel, 1992), and reasoning with statistical knowledge (Bacchus, 1990).

2.2.4 Knowledge Source and Acquisition

Knowledge can be of many types and from different sources, mainly in terms of human expertise given by domain experts directly, and those in verbal and graphic manners in literature.

Knowledge acquisition has been found to be the major bottleneck in expert systems development and requires significant time and effort (Diaz, 1990). Many explanations about the problem were offered such as the lack of tools and the difficulty encountered in encoding an expert's behaviour (Tian and Yun, 1991). As the knowledge comes from domain experts and knowledge-based systems are usually designed by AI engineers, the knowledge acquisition is based on dialogue. To sort out the knowledge from the literature in a given domain, the help of an expert is needed. However, numerous proposed methods of knowledge acquisition have attempted to extract expertise either by human-to-human interviews or by some convenient man-machine interfaces, such as formatted input, pseudo-nature language, or high-level programming languages (Kim and Courtney, 1988; Gaines, 1988).

2.2.5 Learning

AI researchers endeavour to make machines learn, which is another way of considering knowledge acquisition and refinement by automatic means. Knowledge acquisition through machine learning is a difficult process.

The different types of learning can be classified broadly as learning by memorizing, by instruction, by analogy, by induction from examples and by unsupervised learning through observation and discovery (Carbonell et al, 1983). This categorization is based on the increasing complexity of the inferencing process (Chitoor et al, 1991). Some prototypes or substructures of KBSs for learning have been proposed, such as PLAND (Whitehall, 1989) and ACES (Pazzani, 1989). But most expert systems do not have the capability of supervised learning or self-learning at all. Neither do the existing rotating machinery diagnostic KBSs.

2.2.6 Programming Language and KBS Development Tools

A KBS can be built with computer languages or, by means of expert system development tools (shells). The two higher-level languages commonly used for AI programming are LISP and PROLOG. Some others are the object-oriented languages SMALLTALK, C++, Objective-C; rule-based programming environment OPS5; frame language FRL; and general purpose program languages as BASIC, PASCAL, FORTRAN, FORTH, C (Ishii, 1988; Majstorović, 1990), and PL/I (Butler et al, 1988). Using programming languages to build KBSs facilitates the implementation of new methods, but is time and effort consuming.

An expert system shell is basically a programming environment with an embedded inference engine, a knowledge editor, an empty knowledge base, and a designable or modifiable user interface. It is a knowledge input and management tool. It usually provides modules for accessing data files, devices or external programs. There have been more and more powerful ES shells produced in the last few years (Garcia and Chien, 1991). Several new KBSs have been developed using ES shells (Ramu, 1991) to exploit a higher design efficiency and overall convenience. The main disadvantage of using shells is that the KBS designers have to follow a rigid pattern and are limited to the methods offered by the tools.

2.2.7 Interfaces

The user interface is the working level of a KBS to communicate with the end-users through a computer screen, keyboard and mouse. The design of the user interface is critical in attaining widespread system acceptance (Irgon et al, 1990). The user interface could be a menu-driven system (Kirk et al, 1989), text windows (Carlson et al, 1988), graphic windows (Altrock et al, 1991), or drawing support from advanced commercial window systems (e.g. Microsoft® Windows™, X-Windows™ etc). The last style is the newest, most powerful and user-friendly interface, similar to the user interface of shells: ZERO (Ueno et al, 1991), NEXPERT OBJECT™ (TATA, 1989) and LEVEL5 OBJECT™ (Level5, 1990a).

AI scientists think of "data" as describing the fact associated with a specific object or situation. Many knowledge-based systems need to access databases which contain the data to perform their tasks, such as computer aided design (Rosenman et al, 1986), planning (Artiba, 1991) and diagnosis (Irwin and Orden, 1987). NEXPERT and LEVEL5 shells enable applications to communicate directly and dynamically with databases (i.e. Oracle, Informix, RDB, dBASE, Ingres, Sybase and Lotus 1-2-3) during their inference processes.

2.3 Existing Systems

A dozen knowledge-based systems for rotating machinery monitoring and diagnosis have been documented in the literature. Five KBSs along with their principal characteristics are listed in Table 2.1.

The first system in Table 2.1, DXPERT (Piety and Corley, 1989), is a consultative (advisory, or off-line) one. A consultative system works together with its user through dialogue to get input information (facts or data) required to diagnose the problem. DXPERT is a small rule-based system. It has employed a method to measure the uncertainty associated with the input, which represents **Unknown** by 50% and **True Fact** by 100%. It has been used in resolving the real world problems in six different kinds of

Table 2.1 Existing KBSs for rotating machinery diagnosis.

No	Name	Date & Country	Domain	AI Topic	Knowledge Base	Inference Engine	Uncertainty Modeling	Size	Interfaces	Lan./Tool Hardware
1	DXPERT [Piety 89]	1989 USA	Vibration Problems	Consultative	Rule	Backward Chaining	CF on Input Facts	300 Rules	Non-graphic	on IBM PC
2	TUMAD [Keim 89]	1989 Germany	Rotor Dynamic Problems	Consultative	Frame & Rule	Forward & Backward Chaining	Possibility Measure on Diagnosis	2,500 Rules	Non-graphic	MED2 on IBM AT
3	- [Carlson 88]	1988 USA	Rotating Machine Diagnosis	On-Line	Rule	Forward & Backward Chaining	Fuzzy Measure on Input	100 Rules	Menu Driven, Graphics	GEN-X on IBM PC
4	VARMINT [Liddle 93]	1993 Canada	Vibration Analysis	Non-Consultative	Rule	Forward & Backward Chaining	Possibility Measure on Diagnosis		MS Windows	NEXPERT on IBM AT
5	- [Schindler 89]	1989 France	Pump, Fan & Alternator	On-Line	"Machine Object" & Rules	Forward Chaining			Menu Driven, Graphics	IROISE on HP 9000

rotating machines. Similar systems are ROMAD (Mahalingam and Sharma, 1986), KLUE (Karel and Kenner, 1988), KDS (Zhao and Luo, 1989), and the one presented in (Kirk et al, 1989).

The second system, TUMAD (Keim and Nordmann, 1989) is a large system with a more complete knowledge of the domain. It uses frames to define machine objects, and "context-sensitive" rules to represent diagnostic knowledge from the Sohre Chart (Sohre, 1980). The inference strategy has been the mixed forward and backward chaining. It uses a point system (0 to 41) to denote the possibility of suspect diagnosis. This prototype has been successfully tested and then deployed on pump monitoring applications. A similar system, TURBOMAC (Stuart and Vinson, 1985), was developed using the KBS development tool, RuleMaster in 1985, and encompasses about 4,000 rules, but no frames.

The third system was developed by General Electric (GE) Company (Carlson et al, 1988). It works in on-line mode when connected to a machine monitoring system (named TMGR, a HP based software package) through a numeric-to-symbol software (named FEAT), by which the monitoring results can be transferred to **True**, **False**, or **Can't Answer** statements based on fuzzy sets. Then, the expert system takes over to answer its own questions. The expert system portion alone is almost the same as a consultative system, and actually can work in a man-machine interactive mode. Similar systems are AMDS (Gauger and Smee, 1989), AMETHYST (Bernhard, 1989; Milne, 1990), a system presented in (Petersen, 1990), and another one in (Hill and Baines, 88).

The fourth system (Liddle, 1993) has been developed using the shell NEXPERT, with a modern user interface under Microsoft Windows. It uses rules and objects with inheritance to represent knowledge. Knowledge about machine structures, data needed to calculate some key frequencies, and the amplitude alarm level for those frequencies can be specified during the installation stage of this system. It has functions of machine selection, data transfer to "facts", generating symptoms, and diagnosis analysis. It has been able to save up to 75% analytical time in industrial applications. A similar on-line turbine-generator monitoring and diagnostic KBS developed in Japan is reported in (Kato et al, 1990). The latter system is large with about 5,000 rules to analyze 220 measuring points, and it runs under X-Windows on an Engineering Work Station.

The last one (Schindler et al, 1989) in Table 2.1, has a certain amount of knowledge of signal processing to make up for its lack of diagnostic knowledge. It can invoke numerical processing subroutines in diagnostic inference when it needs additional data from the machine being monitored. This ability has not been found in other rotating machinery diagnostic KBSs.

2.4 Artificial Neural Networks and Hybrid Expert Systems

2.4.1 Aspects of Neural Computing

Artificial Neural Networks (ANNs), or simply Neural Networks, are also known as connectionist systems, or parallel distributed processing models (Lippmann, 1987). A neural network is a massively parallel, self-adaptive, interconnected network of basic elements called neurons (Nigrin, 1993). These basic elements loosely model the mechanics of computations in the neurobiological processes of the brain (Zurada, 1992). The early ANN models had been proposed more than 40 years ago, and the more recent works by Hopfield (1986), Rumelhart and McClelland (1986a), Sejnowski and Rosenberg (1986), Feldman (1982), Grossberg (1986) and others have led to a new resurgence of the field.

Neurons of neural networks possess very simple computational abilities but the interactions between them do allow for parallel processing of information that is inputted as a set of stimuli to the array of input processors. The functions of these neurons are (i) to qualify the input signals by evaluating their intensity, (ii) to compute a weighted sum of these inputs using connection weights thereby comparing it with the threshold values associated with individual neurons, and (iii) to fire an output signal if the strength of the input signal builds up to the threshold level or, as in many cases, to obtain an output signal by processing the weighted sum of the input signal through an activation function. The sequence of operations is given below:

1) First, the input signals are multiplied by the connection weights W_{ij} and the effective input to the element j is the weighted sum of these inputs given by:

$$net_j = \sum_{i=1}^{n} input_i * W_{ij} = \sum_{i=1}^{n} x_i W_{ij} \tag{2.1}$$

2) Any one of the three functions, i.e. the sum function, activation function and learning function, is executed depending upon the intended use. The sum function compares the net_j of Eq. (2.1) with the threshold value of the neuron with which it is associated, to determine the presence of the output signal and its level. The activation function processes the net_j through a mathematical function with asymptotic behaviour; the commonly-used function is a sigmoid and it is given by

$$out_j = f(net_j) = \frac{1}{1 + \exp(-(net_j + \theta_j))} \tag{2.2}$$

where θ_j is a bias parameter that is used to modulate the output of the neuron. The learning function is usually associated with a neuron that is provided with a simple form of memory; it makes it possible to store the results of the previous operations to obtain a sort of learning behaviour.

As mentioned earlier, the significant computational superiority of a neural network is derived by interconnecting a large number of neurons into a single network. The overall performance is dictated by the scheme of connectivity and network architecture. The network architectures can roughly be divided into three categories (Kohonen, 1984) as follows: (i) Feed-forward Networks (Rumelhart et al, 1986a; 1986b), (ii) Feed-back Networks (Hopfield, 1982) and (iii) Self-Organizing Networks (Kohonen, 1990). Among these, in the feed-forward networks sets of input signals get transformed into sets of output signals, and the desired input-output transformation is usually determined by external, supervised adjustment of system parameters. In feed-back networks, the input information is used to define the initial activity state of the feed-back system, and the final (asymptotic) activity state is identified as the outcome of computation, after some transitions. In the last category, neighbouring neurons in a network compete in their activities by means of mutual lateral interactions and develop adaptivity to specific detectors of different signal patterns.

The feed-forward networks which have a layer of neurons to which the external stimuli are presented, a series of hidden layers (in some cases) and a layer of neurons at which the output is made available, have three distinct forms of architecture:

Flat Networks: Here inputs are directly mapped into the output. These are fully-connected networks with each and every input neuron influencing all output neurons. These networks do not adequately model the non-linear input/output relationship and the only way of introducing the non-linearity is through the use of activation functions.

Hidden-Layer Networks: Here the external stimuli are received by the input layers and the output layer transmits a signal as the response to this input, thus making the hidden layers not to interact with the external environment but only to accept or output information within the overall system (refer Figure 4.16). If the non-linear activation functions are not used, it is possible to convert a multilayer linear network into an equivalent single layer network.

Functional-Link Networks: Relatively recent, this network is similar to flat networks but has additional input nodes wherein the non-linear transformations of the basic input quantities are performed. These are more suitable to realistic applications, and the learning times are far more economical than those of multilayer networks (Pao, 1989).

Self-organizing mapping networks are sheet-like artificial neural networks, the neurons of which become specifically tuned to various input signal patterns or classes of patterns, so that each neuron or group of neurons acts like a separate decoder for the same input (Kohonen, 1990). The self-organizing network has been intended as a viable alternative to more traditional neural network architectures, and these have been successfully used for tasks similar to those to which other more traditional networks have been applied. Further details about these networks are given later in this thesis.

Design and development of a neural network for a particular application involves first the selection of an appropriate architecture for the selected type of network. Once a suitable architecture is chosen, the network is trained either by the supervised learning method or by the unsupervised learning method, so as to produce an acceptable output for a given input pattern. Self-organizing maps can be trained by the latter method. Various methods of network training are available under the category of supervised learning techniques and, the essential details of the back-propagation approach are given in Chapter 4. With the steepest gradient-descent algorithm, the back-propagation approach is fundamentally looking to minimize the mean square error between the actual network output and the expected output signal. The back-propagation method is suitable for the supervised training process of the functional-link neural network, where the process of weight updating is required only between input and output layers. However, it may be noted that the nature of the gradient descent method leads to a low rate of learning near local minimum, which increases the number of iterations when a higher value of accuracy is prescribed.

A large number of ANN models has been established. Some of the most popularly used models are Hopfield network (Hopfield, 1984), Hamming network (Lippmann, 1987), Kohonen's Self-Organizing Mapping (SOM), Back-Propagation Neural Network (BPNN) (Rumelhart et al, 1986b), Counter-Propagation Network (CPN) (Nielsen, 1987), Adaptive Resonance Theory (ART) (Grossberg, 1976), Bidirectional Associative Memory (BAM) (Kosco, 1988), Recurrent Network (Elman, 1991), and Regression Network (Specht, 1991). Actually, a designer can modify and re-design the structures of neural networks so as to meet particuiar functional demands (Nigrin, 1993).

Current research has shown that ANNs have many outstanding properties. They can operate in real-time under the presence of noise; can perform fast and slow learning, or self-organize using unsupervised learning; they can use feedback expectancies to bias classification; can perform context-sensitive recognition and process multiple pattern simultaneously; can combine existing representations to create categories for novel patterns and scale well to large problems (Nigrin, 1993).

The most remarkable advantage of using neural networks is that they can deal with uncertain information contained in noisy data, represent complex and non-linear relationships between their input and output, process 0-1 logical or numerical information very fast after being trained, learn new knowledge automatically and incrementally from training data, and adapt themselves to new problems.

Applications of ANNs can be found in many areas. Most of them are out of the scope of this research. It seems that most applications are in robotics and control, speech and image processing fields, where ANNs have been used in clustering, feature extraction, associated memories, pattern classification, pattern recognition, signal processing, control, optimization, non-linear process formulation, function approximation and so on. Some examples have been given by Zurada (1992). In the MMD area, attempts to use ANNs to solve rotating machinery diagnostic problems have just started. The examples in published literature include (Shi et al, 1990; Kim et al, 1991; Dimarogonas, 1992; Kuczewski and Eames, 1992; Liu and Mengel, 1992; Liu and Anantharaman 1993).

2.4.2 Hybrid Systems

The conventional approach to building a KBS requires human experts to formulate the rules by which the input data can be analyzed. Neural networks can acquire knowledge without the extraction of IF-THEN rules from a human expert provided that the number of training vector pairs is sufficient to form all decision regions. Thus, neural networks would be able to ease the knowledge acquisition bottleneck that has been hampering the creation and development of conventional KBSs (Zurada, 1992). This realization has led to new research on hybrid systems. The strengths of KBS and ANN can be combined in hybrid systems thus providing the best features of both technologies. A spectrum of possible hybrid architectures has been proposed in (Rich, 1990). A number of approaches of hybrid KBSs for different applications can be found in the literature, including (Weiskopf et al, 1990; Wu, 1991; Khosla and Dillon, 1993; Leonard and Kramer, 1993) and few implemented systems (Fu, 1989; Senjen et al, 1993). Hybrid systems appear to have considerable potential for industrial applications (VerDuin, 1990).

CHAPTER 3

DESIGN OF RMD-KBS

The issues and problems associated with rotating machinery diagnostics, as well as, the types of information and knowledge which are essential to the diagnostic process are summarized in the first part of this chapter. Two fundamental diagnostic problems of rotating machinery monitoring viz., (i) Identification and Classification of Fault Patterns, and (ii) Quantification of Fault Development, are defined and formulated in Section 3.1. Rotating machinery units, such as pumps, fans, compressors, turbines and gearboxes are selected to illustrate the monitoring effectiveness of the RMD-KBS, which is the knowledge-based diagnostic system developed in this thesis. The design of the architecture of RMD-KBS is described in the second part of this chapter, wherein the on-line working mode of the KBS, the methodologies of coupling symbolic and numerical processes, the hybrid approach of utilizing the advantages of both artificial neural networks and conventional rule-based KBS technologies, the knowledge representation schemes, and the diagnostic strategy implemented in the KBS, are detailed.

3.1 Diagnostics of Rotating Machinery

3.1.1 Diagnostics Based on Vibration Signals

Diagnosis of rotating machinery units such as pumps, fans, compressors, turbines, gearboxes etc. for faults or malfunctions during service, has long been performed utilizing on-line diagnostic measurements. On-line measurements of operational variables such as temperature and pressure, state and response variables such as noise levels and vibration amplitudes, constitute the domain of diagnostic signals. Among them, the vibration signal has been shown (Lipovszky et al, 1990; Eshleman and Jackson, 1992) to be very representative of the operational state of rotating machinery systems. Diagnostic methods based on vibration signature have thus established themselves to be highly successful, reliable and efficient. As mentioned in Chapter 1, vibration-based techniques of condition monitoring have been shown to be capable of detecting a broad range of defects in rotating machinery and in a wide array of other machine components. Vibration signals obtained from real-life industrial systems, however, possess considerable variability and further, this variability is distinctly different from the background noise that arises due to signal detection, acquisition and conditioning devices. Moreover, the observed variability has been recognized (Collacott, 1979; Eshleman and Jackson, 1992) to be a direct characteristic of many faults or malfunctions in machine systems. A given state of machine fault or machine condition, is normally represented by a combination of several features of the diagnostic signatures, such as peak-to-peak value of the vibration amplitude, crest factor, Kurtosis factor etc (see Appendix A.1). Changes in the values of these features show symptoms of the corresponding machine fault or condition. The on-line vibration signal is employed in this thesis as the principal diagnostic signal for rotating machinery systems.

3.1.2 Signal Feature Extraction

For monitoring and diagnostic purposes, it is always desirable to reduce the large amount of information contained in the raw vibration signal, to a single index or number that represents the overall characteristics of the signal. This procedure, known as signal feature extraction, apart from helping the machine operator to take remedial action, enables automated comparison with both previous experience with similar machines and with available standards. Such a capability is an essential property of intelligent diagnostic systems. To determine such indices, statistical, time series and probabilistic approaches have been developed (Cempel, 1988; Zhuge et al. 1990) in order to qualify and quantify the variability in the obtained data. More recent methods of fault detection have been developed for use in machine monitoring systems (Danai and Chin, 1991).

3.1.3 Identification of Machine Fault Pattern

Detection of any change in the operational characteristics of a machine through the observation of corresponding changes in its vibration signature leads to the detection and identification of possible faults that cause machinery malfunctions. This problem is one of the most fundamental in rotating machinery diagnostics, and is widely known as Fault Pattern Identification and Classification. Since each of the vibration signature parameters is sensitive to a few particular types of defects, the fault that has occurred can be diagnosed. Further, as mentioned in Chapter 1, simultaneous consideration of several monitoring indices leads to a reliable diagnosis of machine faults. This is due to the fact that there is no single index that can completely represent the conditions of a machine and further, different types of malfunctions are distinctly reflected in different diagnostic indices.

Fault pattern identification and classification, however, involves feature extraction through processing of the large amount of information contained in the vibration signal. The feature extraction approach that is followed in the knowledge-based diagnostic system developed in this thesis, involves the representation of the diagnostic information (characteristics of the signal) contained in the on-line vibration signal by a set of numerical indices. This set of indices which is representative of various failures in a machine is called a feature (symptom) set and is denoted by the column vector $X=\{X_1, X_2, ..., X_N\}^T$. The elements X_i, $i= 1, 2, ..., N$, of the feature set are selected based on previously established descriptors of machine condition monitoring such as peak-to-peak value, root mean square value etc (see Appendix A.1). Selection of a feature set for a particular machinery system, however, depends upon the type of failure or malfunction of the machinery system to be included in the condition monitoring program. For instance, the peak-to-peak value is useful in the detection of unusual impulses generated by bearing defects (Lipovszky et al, 1990); the vibrational energy at the machine rotating frequency in the vibration power spectrum, is applicable to both rotor unbalance and rotor resonance problems (Eshleman and Jackson, 1992).

Various different machine faults or malfunctions that can be clearly identified from the on-line vibration signal are taken as the elements of the column vector $Y=\{Y_1, Y_2, ..., Y_M\}^T$ which is designated as the failure (condition) set. Selection of the elements of Y depends on the particular machine system being monitored. For example, typical faults in a rotor system could include (Eshleman and Jackson, 1992) unbalance, misalignment, resonance, oil whirl etc.

The fundamental problem viz, the identification and classification problem of machinery monitoring and diagnosis (MMD) can now be described as

$$Y \Leftarrow R \Rightarrow X \tag{3.1}$$

where R is the fault-symptom correlative relationship between the feature vector and the failure vector, i.e. a transformation set. To find or define a proper set R, is to describe the relationship between each of the elements of X and Y. After the R has been defined, it can be applied on an observation of the feature vector X to find the corresponding failure vector Y. This is essentially a problem of condition identification. The transformation set R can be an influence matrix as in pattern recognition methods (Danai and Chin, 1991), a fuzzy matrix relationship as in fuzzy classification methods (Xu and Zhang, 1990), or a group of inference rules as in expert systems. It actually represents a feature mapping between the vector spaces of X and Y. In the present design, it is represented by neural networks. A new approach of using neural networks to identify and classify the condition of rotating machinery is developed in Chapter 4, and it is implemented in the Numerical Analysis Routines of RMD-KBS described in Chapter 6.

3.1.4 Quantification of Fault Development

Once the qualitative information about the malfunctions in a particular machinery system has been obtained for diagnostic and monitoring purposes, it is then necessary to quantify both the current stage of the malfunction and its time evolution. This is another fundamental problem in MMD, i.e. the Quantification of Fault Development or Machine Condition, which is intended to determine the health of a machine system through analysis of historical data that are obtained through continuous measurements. This analysis is mathematically known as fault development trend analysis, or trend analysis in short. This is a diagnostic technique widely employed in industry for machine condition monitoring. It is essentially a function regression problem in terms of mathematical statistics. Normally the data is collected over a period of time (or over the number of cycles of operation, or over any other suitable independent parameter) showing the continuous changes in the behaviour of a machine system as a function of time (or process parameters). Existing studies that employ trend analysis for machine condition monitoring, rely on a univariate function in the form of

$$Y = f(X) + e \tag{3.2}$$

Here, e is an unknown arbitrary disturbance, X is a monitoring index being extracted from the machine signal such that its value represents the operative states of a machine system, and Y is an indicator that is defined to monitor the operation of the machine system or the performance of its components, and to predict the degree of damage present. The task of trend analysis is to find the function f defined from a given set of data pairs $Z_i=(X_i, Y_i)$, $i=1, 2, \ldots, n$. The function f represents a curve in the regression plane displaying continuous changes of system condition. Once evaluated, the function f has several important applications. The most common application of f is to determine the current condition and to predict the future behaviour of the machine system being monitored. For instance, after observing the current values of X from a machine system, the residual service life can be approximated from Eq. (3.2). This serves as a basis for suggesting suitable machine maintenance programs.

Since simultaneous consideration of a number of monitoring indices is important not only in machine fault detection but also in fault development trend analysis, diagnostics using a Multiple-Index Based Trend Analysis is proposed as a new approach for MMD in this thesis. This approach is developed in Chapter 5, in a form that is highly suitable to knowledge-based diagnostic systems. Self-organizing neural networks are developed to perform the task of trend analysis and these networks are embedded in the Numerical Analysis Routines of RMD-KBS (Rotating Machinery Diagnostic Knowledge-Based System). The new approach is then embedded in the RMD-KBS. To this end, the fault development trend analysis problem is formulated as

$$Y = f(X) + e \tag{3.3}$$

where $X \in R^N$ is a vector consisting of several monitoring indices. The task of the newly-defined trend analysis is to find the multivariate function f defined in the $(N+1)$-dimensional space, from a given set of data pairs $Z_i=(X_i, Y_i)$, $i=1, 2, ..., n$.

3.1.5 Information Needed in Rotating Machinery Diagnosis

In the machinery diagnostic processes, sensorial signals give the most important information about the condition of machinery. In addition to the sensorial diagnostic signals, several types of information about the rotating machine system are also needed to diagnose its condition. The most necessary information for a monitoring and diagnostic system is listed below:

1) Information about the structure of the machine system, including the number of subsystems, the type of each subsystem (e.g. a gearbox, a pump), the specifications of elements such as bearings and seals, and the position of each machine in the system train. This information is used at different stages of the diagnosis. For instance, the number of teeth in a gear is used as input information to the calculation of the gear tooth-meshing frequency, which in turn is a reference frequency that determines the sampling frequency for the vibration signal to be acquired from the corresponding gearbox.

2) The values of the machine operational parameters (e.g. machine rotational speed, level of load, pressure etc) that show the operational state of the machine system being monitored. Some of the operational parameters can also be used as diagnostic signals. For example, if the value of the lubrication oil pressure is abnormal, it indicates

that the lubrication system may have a malfunction.

3) The location and type of each and all of the sensors mounted on the machine system. This information is used in the selection of a proper scheme for signal acquisition and processing, and is also used to identify the defective machine component.

4) The threshold values of all of diagnostic indices. These are important reference values for the assessment of machine health condition. The threshold values of a monitoring index can be determined from previous experience or available standards.

The information listed above is usually treated as "facts" that are used in the knowledge-based reasoning. From the view point of information processing, this information is in two basic forms, i.e. qualitative descriptions and quantitative values.

3.1.6 Diagnostic Procedure

Having obtained both the qualitative and quantitative descriptions of a fault as well as the description of the machinery system, diagnosis may be initiated. Generally, human experts perform the diagnosis of rotating machines through the following sequence:

1) When some of the on-line measured data show that the machine under monitoring may be in an abnormal condition, the diagnosis will start. The first stage of the diagnosis is to acquire vibration signals and other monitoring signals.

2) Based on a cursory analysis of the collected signals (for instance, checking the vibration level at certain measuring points), the diagnostician will put forward hypotheses of the most likely malfunctions.

3) A selective problem-oriented analysis will be considered so as to focus the diagnosis on the most likely causes.

3.1.7 Knowledge Required in Diagnostic KBS

The most important knowledge required to diagnose the condition of rotating machinery is the human knowledge and experience of both fault detection and machine condition determination. As mentioned in the previous chapters, a large amount of knowledge regarding rotating machinery monitoring and diagnostics has been accumulated in the last three decades. The diagnostic knowledge can be arranged into several categories. This is based on a consideration that, different categories of knowledge play different roles in knowledge-based processing, and further, their representation may need different schemes in building a diagnostic KBS.

The first category is "fault model directed heuristic knowledge" which is based on the known relationships between symptoms and faults/failures. A fault model can be a description of fault-symptom relationship in empirical qualitative mode. This is similar to what has been represented using rules in the existing rotating machinery diagnostic KBSs. In addition to the fault models, this category of knowledge contains several heuristic observations, such as the possible or frequently observed types of faults in a certain type of rotating machine, and the selection of a signal or a number of monitoring parameters as diagnostic indices which are sensitive to the occurrence of a certain fault or the changes in machine condition. Heuristic knowledge can help a KBS to generate hypotheses, narrow down the solution space, and to efficiently arrive at a correct diagnosis.

It has been recognized previously in this chapter that rotating machinery monitoring and diagnostics are mainly based on numerical signal analysis. Hence, quantitative knowledge about fault patterns is necessarily required to make diagnostic judgements based on the values of the monitoring indices. This assortment of knowledge has been implemented through conventional rules in a few existing rotating machinery KBSs, that can reason out numerical information. In the present approach, the main concern is to establish and utilize quantitative models of the fault-symptom correlative relationships.

In the entire process of rotating machinery monitoring and diagnosis, knowledge of the monitoring signal acquisition and processing, feature extraction and selection is also necessarily required. Hence, there is a need for another category of knowledge, which deals with the usage of the techniques and methods employed in signal acquisition, signal processing, feature extraction, and machine condition identification. This category of knowledge includes both, the heuristics to guide the selection and use of a method for certain diagnostic purposes, and the quantitative settings involved in the usage of any method.

For instance, when a gearbox is diagnosed, the spectrum of the vibration signal generated by the gearbox should be investigated. In the spectrum, the amplitudes at the tooth-meshing frequency and its harmonics are important features to be considered in the diagnosis. Further, a proper sampling frequency for the acquisition of vibration data should be selected to assure data accuracy. However, in the existing diagnostic KBSs for rotating machinery monitoring, only some elements of this knowledge are implemented, since there is no signal processing employed in most of them.

The knowledge required to perform machinery diagnosis can also be grouped in two types from another viewpoint: the knowledge regarding qualitative aspects and that regarding quantitative aspects. Therefore, a single scheme for knowledge representation, e.g. the widely used symbolic representation, is not fully suitable for diagnostic tasks. Hybrid approaches, that incorporate both the symbolic and numerical representation and processing are required in machinery monitoring and diagnosis applications.

3.1.8 Importance of Learning Ability in Diagnostic KBS

It has been mentioned in Chapter 2 that learning is considered to be both knowledge acquisition and refinement by automatic means. When KBS technology is employed in performing rotating machinery diagnosis, a certain level of capability to learn new knowledge is important and required to supplement and refine the existing knowledge stored in a KBS. The diagnostic knowledge stored in a KBS is normally considered to be incomplete and imprecise. Often the bottleneck in formulating a diagnostic system is the lack of a qualitative and/or quantitative model of fault-symptom relationships. In turn, this is due to the lack of understanding of fault induction and propagation mechanisms in machine systems. When a KBS has a certain level of learning capability, the new knowledge can be added to the existing knowledge, thus making it more complete. For instance, a fault that is unknown to a KBS may appear in a machine. When it occurs, the faulty behaviour of the machine may be detectable by the KBS, since certain changes in the diagnostic indices will show that the machine is in an abnormal condition. If this is the case, the learning ability can help the KBS to learn the new fault pattern from the observations of the new event.

The knowledge stored in a KBS can also be refined through learning so as to make it more precise than before. As mentioned in Chapter 1, in many situations the indication of a machine fault or malfunction by the monitoring indices is indirect, unclear and unstable. This inherent uncertainty is an important characteristic of machinery diagnosis via signal analysis. For instance, if unbalance is an incipient defect in a rotor system, the magnitude of the spectral amplitude at the rotational frequency must be greater than the value at the same frequency in the vibration spectrum obtained when the rotor was in good condition. This piece of diagnostic knowledge has been proven to be true through many case studies. However, such knowledge may not be sufficient to establish a precise quantitative model to detect unbalance. The model needs to be refined during the diagnostic practice of the KBS, and it can be refined only when the KBS possesses a learning capability. Learning capability in a KBS can improve its adaptability to a wider range of rotating machine systems, since the variations of the values of diagnostic indices that are influenced by particular characteristics of an individual machine can be learned by the diagnostic KBS.

3.2 Design of RMD-KBS

Rotating machinery diagnosis is basically a signal analysis process which involves the complex tasks of feature extraction, fault pattern identification and trend analysis. A knowledge-based diagnostic system should be able to perform these tasks with high precision. Various different types of information and knowledge regarding the machinery system being monitored need to be considered in the diagnosis process. A knowledge-based diagnostic system should be able to gather the needed facts and numerical values, to store them in proper form, and to make full use of them in performing the diagnostic tasks. The way in which the diagnosis is carried out by human experts should be duplicated by the knowledge-based diagnostic system. There is a need to establish a certain level of learning ability in any diagnostic KBS. Also, in the design of a KBS, the ability to handle a variety of applications is an important consideration.

In Chapters 1 and 2, several knowledge-base approaches for different diagnostic applications have been mentioned. The decision tree (Pattipati et al, 1986), fault tree

(Parsaye and Lin, 1987), and goal/subgoal schemes (Allard and Faemmerer, 1987) are some of the early approaches reported in the literature. They use logic operators such as AND, OR to connect the faulty events to the basic component events (i.e. symptoms) (Wang, 1990). Embedded in these approaches is the idea of constructing hierarchical links between the rules in a diagnostic KBS.

A case-based reasoning (Kolodner, 1991) system stores past experience in the form of cases in the system case-base. When a new problem arises, the system retrieves the cases most similar to the current problem and then combines and adapts them in order to derive and criticize a solution. After a problem is solved, a new case can be created and stored in the case-base. However, the case-based reasoning approach has been found to lead to lack of flexibility in the way the acquired knowledge is stored, indexed and subsequently, retrieved and re-used (Feret and Glasgow, 1992).

Another KBS approach known as the model-based diagnosis (Reiter, 1987; Iwamasa et al, 1992), has also been in use. In this approach, the model of a physical device that represents structural, behavioral and functional knowledge, is used to provide the expected behavioral data which is compared against the observed data from the device under examination (Yu and Biswas, 1992). The models are based on theoretical formulations, simulation or experimental data. To be effective, model-based reasoning systems require an accurate model of the physical system in order to reason out the expected behaviour. The models of the physical system must also be complete and independent from the reasoning unit of the KBS (Adamovits and Pagurek, 1993). The model-based approaches are considered to be able to handle new faults, but the KBS itself can be very complex in its structure. Most model-based diagnostic KBSs have been designed for fault detection in electrical circuits, since the behaviour of circuit elements can be modeled. In many other applications, model-based reasoning has been considered to be not fully suitable to the diagnosis (Schönwälder et al. 1991), since there are less available models which can be used to predict accurately the expected behaviour of the physical system of interest. Rotating machinery diagnosis is one such domain, where models that are able to support the above reasoning approach are not readily available.

Another approach to diagnostic reasoning is the experiential one (Reiter, 1987), in which heuristic knowledge plays a dominant role. The corresponding diagnostic reasoning systems attempt to codify the fault-symptom relationships, rules of thumb, statistical intuitions, and past experience of human diagnosticians who are considered experts in the task domain. This approach is quite close to the way in which a human expert recognizes machine condition. All the existing KBSs listed in the last chapter have employed this approach. In the design of RMD-KBS, the diagnostic reasoning is also based on this approach.

3.2.1 Outline of the Design

The scope of the design of RMD-KBS is limited to rotating machinery diagnosis based on vibration signals. The functions and characteristics which have been built into the RMD-KBS are as follows:

1) It is an on-line monitoring and diagnostic system.

2) Both symbolic and numerical processing techniques are employed, and further, they are tightly (deeply) coupled. Symbolic reasoning controls the diagnostic process.

3) It is a hybrid system, in which artificial neural network algorithms are incorporated in order to perform several diagnostic tasks and to provide learning ability to the system.

4) The knowledge in the KBS is modeled into different categories, and is represented by several schemes. This is done in order to achieve control of the diagnostic process, and to make the KBS accurate in the diagnosis, flexible for knowledge refinement, and adaptable to individual real-world applications.

5) The diagnostic strategy utilized in RMD-KBS is close to the standard procedure used by diagnosticians.

6) It is conceived for real-world applications and many aspects are considered in the design in order to make the RMD-KBS more user-friendly.

RMD-KBS is designed as an integrated system that consists of an on-line vibration data measurement and acquisition component, a database, a group of numerical processing routines, the neural networks, several knowledge bases, an inference engine and a multiple windowing user interface. The architecture of RMD-KBS is shown in Figure 3.1, wherein the functions to be performed by each of the above elements and the connections between them are briefly indicated. The implementation details are given in the following subsections and in Chapter 6.

3.2.2 On-Line Monitoring and Diagnostic Capability

The RMD-KBS is designed as an on-line rotating machinery monitoring and diagnostic system. The KBS can automatically perform signal acquisition, processing, feature extraction, and diagnostic analysis to detect machinery faults, with minimum run-time user interactions. It is based on the recognition of the fact that a consultative KBS cannot perform the diagnosis precisely, and its end-users have to be familiar with many techniques involved in machinery diagnostics.

It has been seen in Chapter 2 that about half of the existing KBSs for rotating

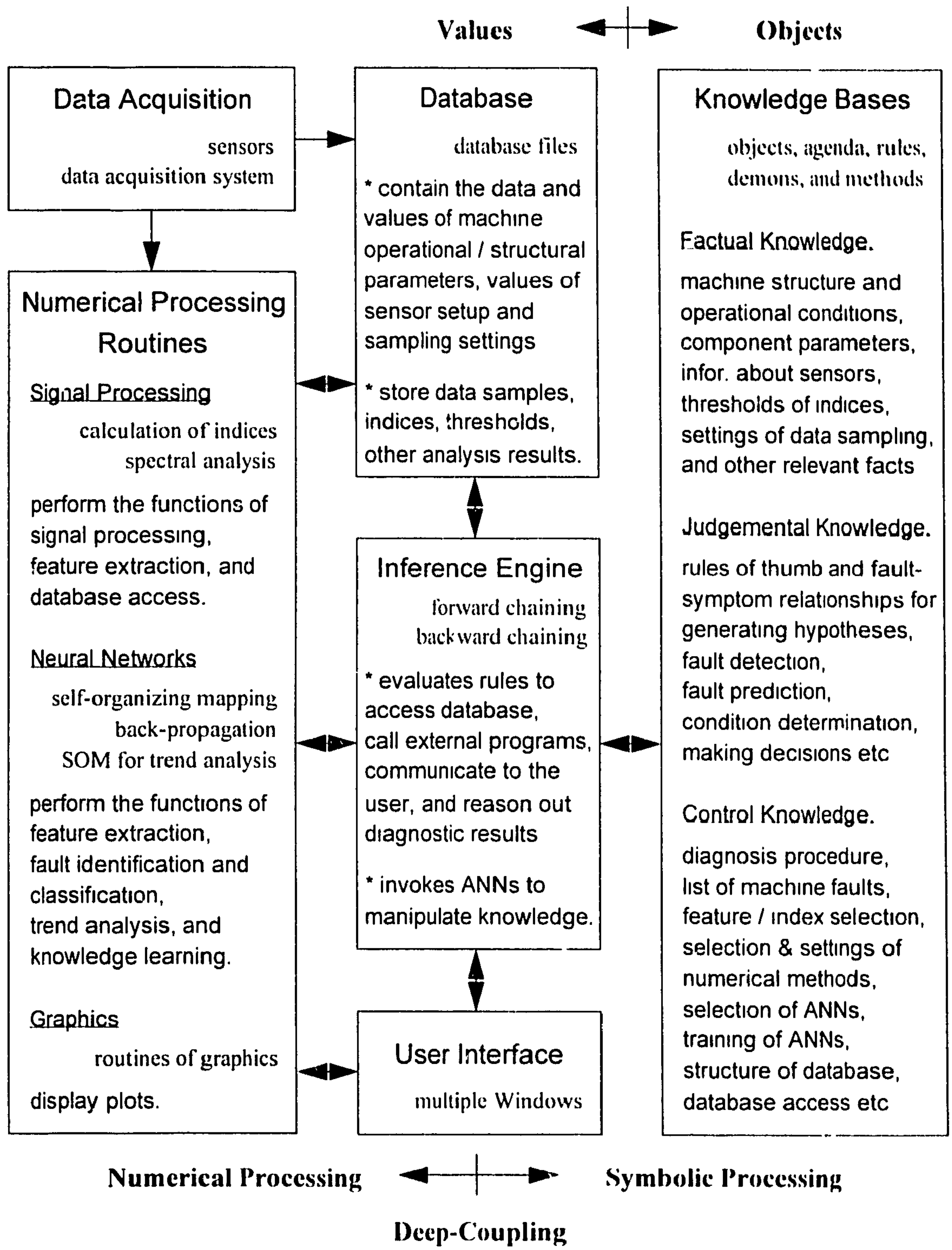


Fig. 3.1 The architecture of the hybrid RMD-KBS.

machinery diagnosis, such as TURBOMAC (Stuart and Vinson, 1985), KLUE (Karel and Kenner, 1988). KDS (Zhao et al, 1989), TUMAD (Keim and Nordmann, 1989), DXPERT (Piety and Corley, 1989) and the KBS reported in (Kirk et al, 1989), are consultative systems. They are off-line systems and perform the diagnosis based on the information provided by their end-users through a dialogue during the run-time. The existing KBSs lack the knowledge of signal analysis and of quantitative measurement of the diagnostic indices. These are pitfalls, especially from the view point of industrial applications. For example, the following rule in the TUMAD system (Keim and Nordmann, 1989) is typical of the rules employed to represent diagnostic knowledge in existing consultative systems:

IF:	**(**	**Machine has sleeve bearing**
	OR	**Machine has long seal)**
	AND	**Vibration is 42 to 48% of running frequency**
THEN:	**Suppose diagnosis is oil whirl**	

This rule qualitatively describes the human knowledge about the machine problem known as oil whirl, where the information needed to detect the oil whirl problem by this rule is listed as a set of conditions in its IF-part. A user of such a consultative KBS may be able to provide the KBS with precise information about the bearing type and seal length of elements in the machine system. But, in order to tell the KBS the fact whether a significant peak is appearing in the range between 42 to 48% of the rotational frequency of the machine in the vibration spectrum, which is a primary symptom of oil whirl, the end-user must first perform the necessary signal measurement, signal processing and the spectral analysis tasks.

Moreover, in the case that a consultative KBS does not know how to quantitatively measure the numerical values of diagnostic indices, many computational and judgemental aspects are actually left to its users. For instance, the expert system presented in (Kirk et al, 1989) asks its users, "**Is it true that it has high pressure?**" The possible user responses to this question are **Yes**, **No** or **Don't Know**. Hence, in this case, the user has been assumed as being able to judge what are the values of the pressure that are considered to be high. If the user can only answer the question about the pressure by **Don't Know**, the KBS can certainly not conduct a precise diagnosis of machine faults.

In practice, consultative diagnostic KBSs depend on considerable human/machine interaction before arriving to a conclusion (Mielnik, 1990). Such a KBS performs only a part of the work, since the signal processing and certain determinations have to be done by its users. On the other hand, an on-line monitoring and diagnostic KBS works in an automatic mode which can perform the entire process from signal acquisition to knowledge-based reasoning. Since there is more knowledge stored in it than in a consultative KBS, there will be less work that is left to its end-users. This advantage renders on-line systems more suitable for industrial applications.

3.2.3 Coupled Symbolic and Numerical Processing

The expert's diagnostic knowledge is about both the qualitative and quantitative aspects of the methods employed and of the fault-symptom relationships. Therefore, both symbolic and numerical processing methods are essential in solving rotating machinery diagnostic problems. In the existing KBSs, symbolic processing is the key approach where both information (facts and data) and knowledge take the form of symbolic representation, such as frames and rules. Previous research (Kanal and Raghavan, 1992) has shown that the computational schemes used to manipulate symbolic information are typically less effective in dealing with numerical data. A more effective means of using AI approaches is to couple symbolic and numerical computing in the KBS (Wang, 1989; 1990). Such systems are known as coupled systems (Bhandarkar and Suk, 1991). The KBS developed in this thesis belongs to this category.

The RMD-KBS is designed to possess both a numerical processing unit that is a group of routines in C++ codes, and a symbolic processing unit that is an inference engine with several knowledge bases that use objects and rules to represent knowledge. The rule-based reasoning in RMD-KBS will control both the information flow and the diagnosis procedure. A deeply-coupled system is designed, in which symbolic and numerical processing are coupled in an extensive manner. The symbolic processing unit in this system has the knowledge of numerical processing routines linked with it, so that it is able to reason with regard to the applications of those numerical processes and to analyze their results. The numerical processes employed for signal sampling, processing, analysis, and for graphics, are utilized together in order to solve a given complex diagnostic problem. Thus the problem can be solved more efficiently than would have been possible with a single numerical process alone. This deep-coupling strategy is designed with the aim of facilitating the representation of the entire human diagnostic knowledge, into the RMD-KBS.

The existing on-line or non-consultative KBSs for rotating machinery diagnosis (Carlson, et al. 1988; Hill and Baines, 1988; Gauger and Smee, 1989; Milne, 1990; Petersen, 1990; Kato et al, 1990; Liddle and Reilly, 1993) can also be considered as coupled systems, and be decomposable into a numerical processing unit, a symbolic processing unit and a coupling unit in between. However, they are shallow-coupled systems in which the symbolic component has limited knowledge of the numerical processes involved.

As can be seen from the third system illustrated in Section 2.3, the numerical processing unit is actually a computerized on-line monitoring system. The numerical values of the monitoring parameters, are converted into statements such as **True**, **False**, or **Can't Answer** by a numeric-to-symbol software (i.e. the coupling unit). Further, the expert system unit of the above system, takes over the symbolic descriptions of the on-line vibration signals to answer its own questions. Moreover, the numerical processing unit of such a KBS provides the values of all diagnostic indices to the symbolic processing unit en-block, without any judgement on whether a particular index is related to the current diagnostic task or not. The symbolic processing unit can not control the signal acquisition and the feature extraction. Obviously, there is a lack of knowledge in the symbolic processing unit of this KBS, about how and when to execute a numerical process. Neither has the expert's diagnostic knowledge been fully embedded. So, the coupling strategy does not use the numerical processing methods and data intelligently. The symbolic processing unit in this system is quite similar to a consultative KBS, except that it obtains the values of the machine signal features from the on-line monitoring system linked to it, and not from the end-users. Moreover, since the transformation between the numerical and symbolic processes may bring in a problem of information loss, a reduction in the accuracy of the monitoring parameters is not avoidable, which in turn affects the accuracy of the diagnosis.

In the design of the RMD-KBS, there is communication of information between the numerical calculation routines and the KBS mainframe. These two units can exchange information with each other and invoke each other during the problem-solving phase. For example, the symbolic processes can transfer the problem specifications and numerical settings to the selected numerical processes. The numerical processing unit can also transfer the results of computation back to the symbolic processing unit, and then the symbolic processes will analyze those results. However, this KBS will not have a physical coupling unit that is used in the existing systems. The information transfer scheme is looked upon as a call for a remote procedure (external program) with the data regarding individual parameters, name of a data file, or a class of data in complex form as the message. The symbolic processing unit which issues a message becomes a client and waits for a reply from the numerical processing unit. This way, certain rules in the KBS can invoke numerical data processing routines and also receive the results of processing. Moreover, the diagnostic reasoning in this system is operated directly on the available numerical information. No translation is done unlike the existing KBSs. The purpose of this proposed deep-coupling strategy is to preserve the advantages of both pure numerical and pure symbolic processing techniques, and to improve the accuracy with which the diagnosis is performed.

3.2.4 Incorporation of Neural Networks

Rule-based expert systems and neural networks could be used to supplement each other in a hybrid system (Rich, 1990). Neural networks (connectionist reasoning) could be used within a symbolic system (Minsky, 1991) to reduce the search complexity, to produce a compact and well defined problem space, and to help overcome the often-encountered problem of unexpected interaction between rules due to the usage of too many rules.

The architecture of several hybrid KBSs reported in (Weiskopf et al, 1990; Kanal and Raghavan, 1992; Senjen et al, 1993), consists of both a connectionist and a rule-based reasoning module. In these cases, neural networks were employed for signal feature extraction and evaluation, as a preprocessor which took raw data as their input (Weiskopf et al, 1990). Then the outputs of the ANN module were passed to, and further analyzed by the rule-based reasoning module. Hence, these systems are called "partitioned hybrid systems" (Senjen et al, 1993), where two separated modules with clearly defined roles implemented for both knowledge representation and reasoning. A typical partitioned hybrid system is shown in Figure 3.2 (from Senjen et al, 1993). This hybrid system is quite similar to the existing rotating machinery diagnostic KBSs in the way in which numerical and symbolic processing are coupled. There is no interaction in this hybrid system between the two modules except data communication.

Both neural network components and the symbolic module are embodied in the RMD-KBS, and further, they are integrated together so as to share the knowledge representation and diagnostic reasoning on the same problem. In RMD-KBS, the symbolic processing unit has knowledge about machine faults diagnosis. It can heuristically select key features corresponding to a diagnostic problem, call signal processing routines to calculate the current values of the features, and then invoke, through the inference engine, a proper neural network to reason out based on the obtained values of the selected features. The neural networks invoked in the reasoning, also carry the knowledge in quantitative form about the fault-symptom relationships, so that they can evaluate the current condition reflected by the numerical values of those features. Further, the output results of this neural network will be used together with other information handled by the symbolic processing unit in order to render a diagnostic determination. In this hybrid system, the neural networks are components of the system itself. Each of them can perform

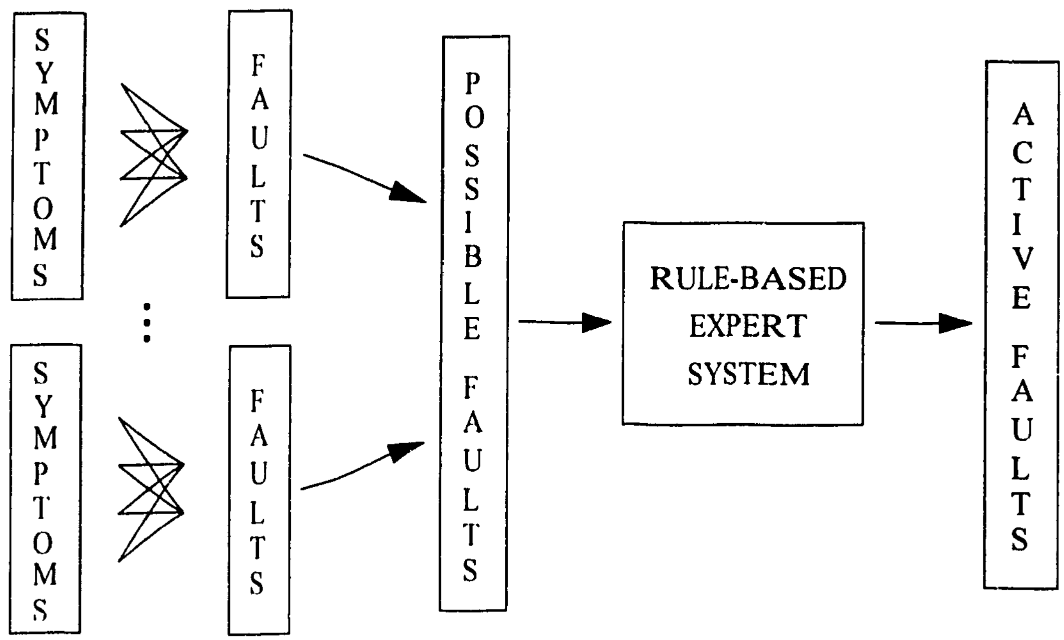


Fig. 3.2 The structure of a partitioned hybrid system (from [Sejen 93]).

certain tasks at various stages of diagnostic reasoning. The control as to when to execute which neural network, is performed by rule-based reasoning, in a way similar to the coupling between the numerical and symbolic process that is described in the last section.

There are two important ideas to be realized in this hybrid system. The first idea is that, instead of rule-based knowledge representation, the quantitative relationships between faults and symptoms will be mainly represented by the neural networks and will be used in their connectionist reasoning. The heuristic knowledge in qualitative mode, however, will be manipulated in rule-based knowledge representation. Therefore, the qualitative and quantitative modes of information and diagnostic knowledge could be handled using the most appropriate technology. The advantages of both symbolic and numerical processing technologies are thus properly utilized in the present hybrid system.

The second idea is that, any updating of the knowledge in this KBS depends on the type of knowledge to be added or modified. In many cases, new training of the neural networks is considered as the learning of new knowledge or refining of the existing knowledge in the KBS. Sometimes, the symbolic part also needs modification to add new rules or to refine existing rules. In general it can be said that, supplementing or modifying the symbolic part in KBSs is not a simple task. This task is usually performed by AI engineers who know the internal structure of the system and the connections between rules. Compared to this, the training of neural networks is much easier and faster. In the RMD-KBS, the task of adding new rules or modifying existing rules is reduced and simplified, since much of the representation and learning/refining of knowledge is assigned to neural networks. Under the present hybrid architecture of integrating the two technologies, both the neural network learning and rules modification can be easily performed.

It has also been observed that most of the previous applications of ANNs in solving various diagnostic problems, attempted feature extraction, wherein the inputs for the neural networks were raw data, such as time domain signal data (Weikopf et al, 1990; Leonard and Kramer, 1993; Liu and Anantharaman, 1993), or entire values of FFT spectral components (Kuczewski and Eames, 1992). In other applications, ANNs were used for fault pattern classification with key features of the diagnostic signals as their input (Kim et al, 1991; Liu and Mengel, 1992). In both cases, only back-propagation neural networks (BPNN, see Chapter 4) have been employed. The ANNs for feature extraction would be large in size, with up to 128 input neurons (Kuczewski and Eames, 1992) and 36 output neurons (Leonard and Kramer, 1993). The training of such large sized ANNs is relatively more difficult than the training of smaller sized neural networks, and further, a large amount of training data sets is required for the former. In the present design, the neural networks incorporated in RMD-KBS are intended to perform fault pattern identification and classification, and function regression for fault development trend analysis. Neural networks that take a number of diagnostic indices as their input and are smaller in size, are preferred. An individual neural network that is embedded in this KBS will perform the detection of limited types of machine faults based on the input set of diagnostic indices.

In the implementation of RMD-KBS, the neural networks are encoded in C++ as are the numerical routines. The communication between the neural networks and the symbolic processing unit of RMD-KBS is similar to the coupling between the numerical analysis routines and the symbolic processes.

3.2.5 Knowledge Modelling

The knowledge to be stored in the knowledge base of this system is a complex combination of different types of information. It can be sorted into three categories of knowledge: factual knowledge, judgmental knowledge and control knowledge. The incorporation of these three kinds of knowledge within a single system represents the most current trend in knowledge-base diagnostic systems (Wang, 1991).

Factual knowledge relates to the defined problems, settings, conditions and facts, including: 1) identification and structure of each machine system; 2) functional correlations between machine components; 3) parameters of the machine components; 4) general operational conditions; 5) sensor types, locations and measurement settings; 6) settings of data sampling; and 7) list of monitoring indices and their thresholds. Factual knowledge can be in many different formats, such as numerical values, symbolic descriptions, curves and graphs.

Judgmental knowledge is usually in the form of condition-effect pairs required in making suggestions, analysis and decisions. In this system, judgmental knowledge is considered to be that required in the processes of 1) generating hypotheses of machine faults (see Section 3.2.6); 2) reasoning on the numerical analysis results; 3) fault pattern identification and classification; 4) prediction of fault development or condition changes; 5) rendering final diagnostic results; and 6) making other pertinent decisions. The judgmental knowledge that is most important to this design is about cause and effect correlations between observable symptoms and defined faults (or malfunctions). This is known as fault pattern or fault model in the literature (Wang, 1990). A fault model is a description of fault-symptom correlative relationship, and of both the "correct behaviour" and "faulty behaviour" of a machine or its component.

Control knowledge guides the information flow, diagnostic procedure, and execution of the system. It covers the knowledge of 1) the diagnostic process sequences; 2) the list of the frequently-experienced failures (faults and malfunctions) of rotating machines; 3) the hints and clues for diagnostic feature selection; 4) the functions of each and all numerical processing routines including neural networks; 5) the selection and settings of numerical routines to suit the type of diagnosis requirement; 6) invoking signal processing routines and neural networks; 7) the database structure, access and management; 8) training neural networks; 9) the information, structure, and settings of the KBS itself; 10) control of the user interface.

3.2.6 Knowledge Representation

RMD-KBS is designed as an integrated system that has a single inference engine to work with several knowledge bases. Each of the knowledge base works with the inference engine which consists of a module to perform certain aspects of the diagnostic task. In such a design, an individual module will be relatively smaller in size, and requires less computer memory to run. The number of rules in each KB is smaller, so as to ease their management, and to speed up the inference searching among them. Such a design is believed to be better than developing a single huge knowledge base.

Different types of knowledge are represented in RMD-KBS by different schemes, such as "objects", "agenda", "rules", "demons", and "methods" (Level5, 1990a). These schemes of representing domain knowledge have long been in use. The agenda schedules the events which an application will follow, or the hypotheses the backward chaining which the inference engine will pursue. The rules and demons are the same in the statement format, but rules are fired by the backward chaining in the inference, while demons are fired by the forward chaining. A method contains a group of statements for knowledge-based processing. The control knowledge is represented by symbolic rules, demons and methods. The judgemental knowledge, especially the fault-symptom relationships will be represented in both connectionist scheme of neural networks and symbolic representation, in the form of rules, demons and methods. The neural networks contain quantitative knowledge about the relationship between the values of diagnostic indices and the occurrence of the corresponding faults. The rules represent the expert's heuristics knowledge of the fault-symptom relationships in a qualitative mode, rules of thumb and many other connections/relationships employed in the diagnosis process. The details of knowledge representation with each KB will be given in Chapter 6.

A new scheme, called Object-Oriented Programming (OOP) technique, which has not been utilized in existing rotating machine diagnostic KBS, is employed in RMD-KBS. Object-oriented computation provides a powerful means of controlling access to shared data, data abstraction, program modularization, and structural knowledge representation (Ramamoorthy and Sheu, 1988). In general, OOP is based on four concepts - object, message, class and instance. Essentially, an object encapsulates a set of private data and a group of procedures (methods or functions). The private data represent the "object task" assigned to this object, and they can only be accessed or modified with the activation of the group of methods in both symbolic and numerical modes. The procedures of the object are hence designed as object-oriented methods, and further, they accept messages through a call of them, that ask them to access or modify the data. Objects can be arranged in a hierarchy in which operations implemented at upper hierarchical levels can be automatically recognized at lower levels. In object-based systems, each object is defined by declaring a class and its instances. A class consists of attributes and functions, which define the structure of an object. In order to carry the values of the attributes, an occurrence of the class, called instance, must be created. For example, an object named Bearing Type is defined by declaring a class in RMD-KBS as follows.

```
Bearing Type
{
        rolling element
        sleeve bearing
        deflection pad
        ...
}
```

Any individual bearing in a rotating machine is one of the types listed in this class. In the KBS, several instances of the above object may be constructed with each instance intended to carry the information about a bearing in a machine system. The object may be inherited by another object at a lower level.

The point to be highlighted here is not the usefulness of OOP, but the method of using objects to represent the factual knowledge and to support the data sharing among the modules that constitute the RMD-KBS. In the KBS, one special type of objects can be linked to database files. They are named "database classes". A database class is composed of several attributes and a group of procedures for accessing the database file associated with it. The values (usually called facts) of the attributes are saved in the related data file. This way, the terms used to represent factual knowledge can be declared as attributes of database classes and stored in a knowledge base of RMD-KBS, while their values can be stored in the relative data files separately. The use of this scheme to represent the factual knowledge has two main advantages. The first advantage is that the facts (values of the attributes in database classes) can be acquired or modified through accessing the database files instead of modifying the knowledge base. Since access to database files is much easier than adding or modifying the statements in a KB, this scheme makes the information acquisition and system adaption to new applications, to be much easier than using other schemes. Therefore, this use of OOP in the design provides a flexible, riskless, and user-friendly environment, and with it, the end-users can perform many tasks without help from AI engineers.

The second advantage is that the object-oriented representation facilitates data sharing among the modules of RMD-KBS. It has been mentioned previously that this system consists of several numerical and symbolic modules. In its symbolic processing unit, there are several knowledge bases. The numerical processing routines and neural networks are written in C++, and the symbolic processing modules are several knowledge bases working with the same inference engine. Interfacing dissimilar languages in a coupled hybrid system composed of different environments, however, is a very complex task from an implementation point of view. OOP provides a bridge to link different modules, and also different knowledge bases. For instance, when a piece of information or knowledge is requested in different modules or knowledge bases, the same object can be declared in those modules. The values of the attributes of such an object can be communicated between modules through database files or through the working memory of the system.

3.2.7 Diagnostic Strategy

The procedure employed to diagnose the condition of machinery in the present design is shown in Figure 3.3. It attempts to simulate human performance more closely than available strategies. As mentioned previously, the RMD-KBS is connected to sensors mounted on the machine system. When this KBS detects abnormalities in the monitoring data, for example, when a signal value is greater than its prescribed threshold, it will start the diagnostic process. Diagnosis is performed in three steps. In the first step, based on the results of an analysis of the data obtained from a given sensor, a set of hypotheses of suspected machine faults is established from a forward chaining search in a group of rules. This step involves both signal processing and a rule-based reasoning through those rules that represent the expert's knowledge about the fault-symptom relationships The task is to determine the "best hypotheses" in a heuristic way to increase the efficiency of performance of the diagnosis. Since the establishment of a hypothesis is mainly based on human experience, a certain level of uncertainty is always involved. Confidence Factors (see Appendix A.2) are therefore used in the rules for generating the hypotheses.

The second step is to determine the "best signal analysis approaches and diagnostic indices" that yield the correct results. More machine signals will be collected from all the interrelated and critically-located sensors and more analysis will be carried out on the data. The above strategy is shown in Figure 3.4. The signal collection and analysis routines are problem-oriented and are optimized by diagnostic expertise embedded in RMD-KBS. For example, to confirm whether a hypothesis "Unbalance" is true or not, a group of carefully selected indices of vibration signals that are closely related to the rotor unbalance fault will be calculated from the raw data. The results that come from this stage of inference constitute the symptom vectors, that denote the current condition with respect

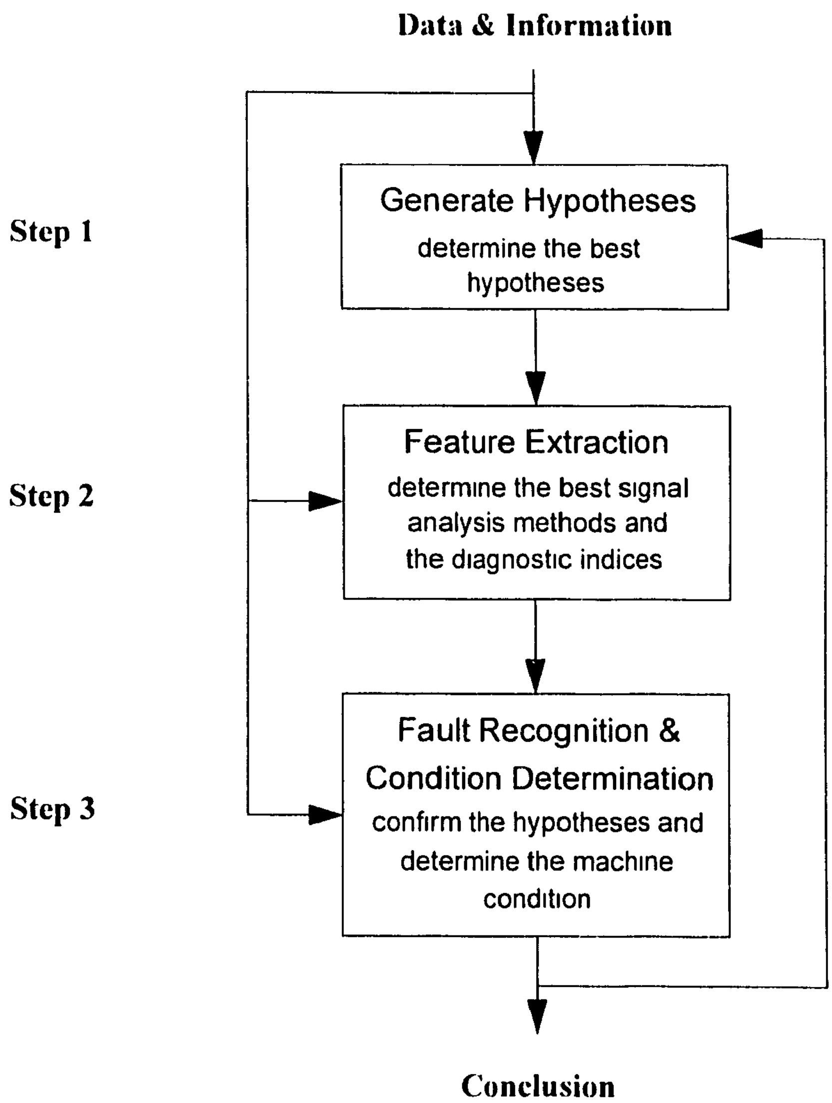


Fig. 3.3 The diagnostic strategy used in RMD-KBS.

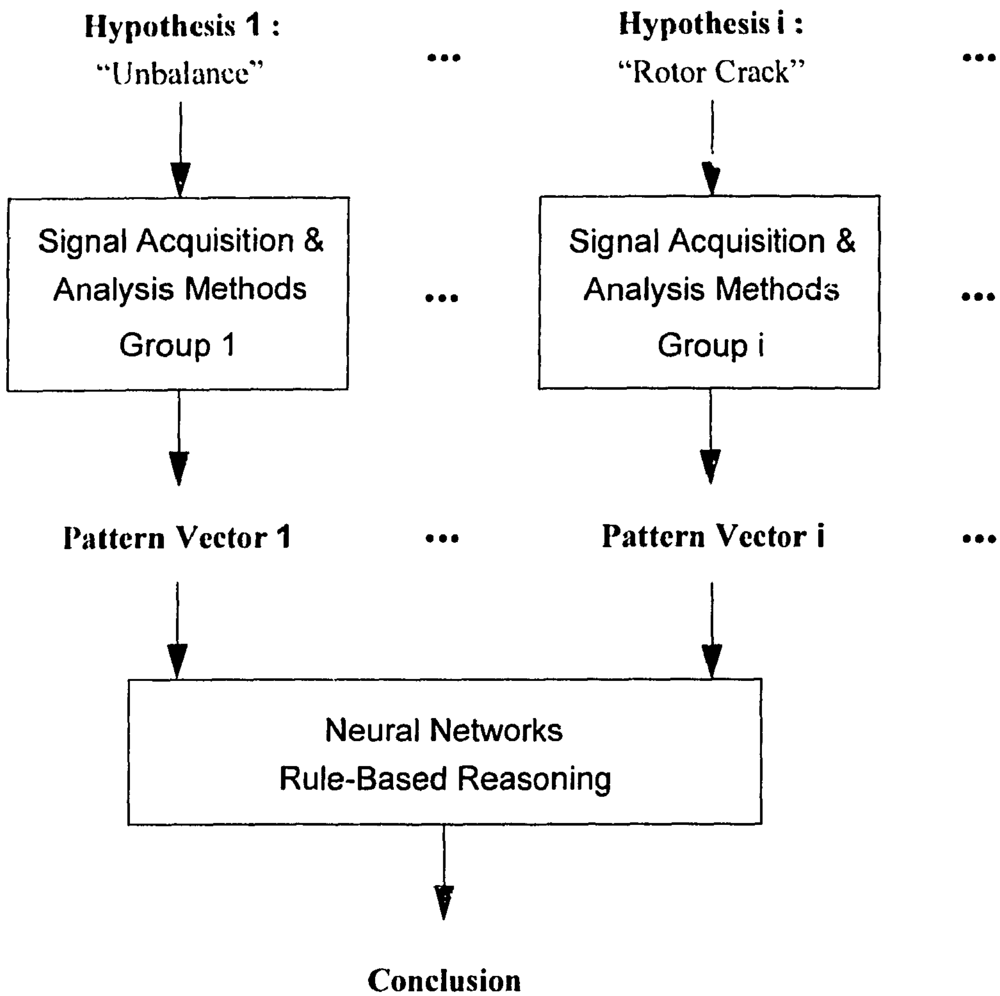


Fig. 3.4 The fault classification approach employed in RMD-KBS.

to each proposed hypothesis.

The third step is to confirm the hypotheses based on the observed evidence - the pattern (symptom) vectors. This part of the work focusses on the "pattern recognition and machine condition determination" by using the knowledge stored in the neural networks. Some well known patterns of rotating machine faults will be identified and classified by neural networks at this stage. Further, some determination tasks, which do not suit neural computing or need additional symbolic information, will be handled by rule-based reasoning. If the proposed hypotheses could not be confirmed, a new set of hypotheses will be introduced and proven through the same procedure as described above.

3.2.8 Inference Schemes

An expert system development shell, named LEVEL5 OBJECT™ is employed in developing RMD-KBS. This tool supports forward, backward chaining, and mixed-mode chaining. In this design, some steps in the execution of RMD-KBS are handled by means of forward chaining, including the control of user interface, acquiring information from the end-users, saving information to databases, displaying graphics, the procedures of generating the hypotheses and calling numerical routines. In the last step, the determination of machine condition will be reached through a mixed forward and backward chaining scheme. The forward chaining follows a data- or event-driven mechanism which provides control of the above steps in the execution of RMD-KBS. The verification of the previously selected hypotheses is naturally a goal- or hypothesis-driven process, that will be mainly realized by backward chaining, plus forward chaining in some cases where it is considered more suitable.

3.2.9 Data Normalization

In most of the diagnosis tasks, a particular condition of a machine system is distinguished based on the relative change in the values of diagnostic indices (Serridge, 1989), but not only based on the comparison between the absolute values of measurements to corresponding standards. This procedure actually allows normalization of the obtained data from the on-line measurement, that keeps the values of data comparable after the normalization. On the other hand, a computer-aided monitoring and diagnostic system may monitor more than one machine system and may get signals from more than one sensor. It is often required to normalize the values of on-line monitoring indices in order to simplify the judgemental task that is based on the relative changes. Data normalization has been widely used in computerized MMD systems (Watts and VanDyke, 1989). The data normalization method used in the RMD-KBS, is now defined.

Any absolute value of a diagnostic index to be used in RMD-KBS is normalized into a dimensionless value in the range of [0.0, 1.0] according to the following expressions.

$$x = \begin{cases} 0.0, & x^* \leq 0.0 \\ x^*, & 0.0 < x^* < 1.0 \\ 1.0, & x^* \geq 1.0 \end{cases} \tag{3.4}$$

where x is the normalized value of a diagnostic index, and x^* is determined by

$$x^* = 0.8 \frac{x_a - x_{min}}{x_{max} - x_{min}} + 0.1 \tag{3.5}$$

where x_a is the absolute value of an index obtained from on-line diagnostic signal, x_{min} and x_{max} are the respective thresholds of the minimum and maximum absolute values of this index. This method of data normalization is similar to that defined in (Liu and Mengel, 1992). Normally, the thresholds of an index are determined by three means: 1) by referring to the available standards or documentation provided by the manufacturer of the machine system; 2) thresholds are drawn out from historical observations of that diagnostic index through statistical methods; 3) based on the information provided by experienced experts or machine operators. Further, any threshold value can always be updated and rendered more representative, when sufficient data have been obtained from a machine system.

3.3 Discussion

The RMD-KBS is designed to possess certain special features that are not available in existing diagnostic KBSs. To cite just a few, they are: 1) the diagnostic process is controlled by symbolic reasoning, 2) several complex diagnostic tasks are performed by artificial neural networks, which also provide learning ability, 3) a deep-coupling exists between symbolic and numerical processing of the diagnostic information and on-line signals, 4) on-line diagnostic data measurement and acquisition system, database, numerical processing routines, knowledge bases, inference engine and user interface are all combined together to make the RMD-KBS, an integrated diagnostic system and not just a diagnostic KBS with the symbolic processing capability, 5) a new scheme for controlling access to shared data, data abstraction, program modularization and structural knowledge representation, called "Object-Oriented Programming Technique", is employed in RMD-KBS, 6) the advantages of SOM neural networks are employed so as to result in an efficient diagnostic process, and 7) a new SOM algorithm is developed and implemented into RMD-KBS for the quantification of fault development. A number of diagnostic indices, that can represent the signatures and features of the vibration signal, are used in the RMD-KBS. The definitions of these indices are listed in Appendix A.1.

CHAPTER 4

FAULT CLASSIFICATION USING NEURAL NETWORKS

A new approach to machine fault classification using neural networks (Zhang et al, 1994a; 1995c), is presented in this chapter. The problem of machine fault identification is formulated in terms of pattern clustering and classification concepts in Section 4.2. Diagnostic indices obtained from on-line measurements, corresponding to various faults arising in a machine system are used to constitute a multiple-element symptom vector. This formulation is of a more general nature in that it can be directly applied to any machine system. The suitability and efficiency of Self-Organizing Mapping (SOM) algorithms (Kohonen, 1990) in solving the resulting clustering and classification problem are outlined. A SOM network is developed to perform the clustering and feature extraction which takes the multi-dimensional data set as input and provides the condition of machinery systems as output. The details of the SOM algorithm as applied to the clustering and feature extraction problem, are described in Section 4.3. The developed approach is fully demonstrated through its application to the cases of both bearing condition identification and rotor system fault classification, which are presented in Section 4.4 and Section 4.5, respectively. For comparison, back-propagation neural networks (Rumelhart and McClelland, 1986a; Lippmann, 1987) are applied to the same classification problems, and the results are compared in Section 4.6. Discussion on the proposed new approach is presented in Section 4.7.

4.1 Neural Networks in Fault Detection Systems

Neural networks possess a high level of adaptivity that can not be obtained from completely-analytical or numerical procedures and further, they provide a data-based heuristic approach to the condition monitoring and diagnostics of machinery systems. A neural network can automatically store the knowledge about the faults in the machinery system being monitored by learning from the historical data, and also possesses the elements of associative memory. ANNs, via their associative memory skills, have the capability to learn and store complex information about abnormal machinery conditions from the faults identified and classified in the past. The associative diagnostic capabilities make neural networks superior to conventional methods of machinery fault diagnostics.

Recently, neural networks have been employed to handle problems of machinery monitoring and fault detection. They have been shown to be particularly useful for the analysis of machine degradation (Lee and Kim, 1993), condition monitoring of production systems (Rangwala and Dornfeld, 1990; Jiaa and Dornfeld, 1992), and diagnostics of machinery systems (Kim et al., 1991). However the published literature listed in Chapter 2, reports only back-propagation neural networks (BPNN). A BPNN is a multilayer feed-forward network which is one of the most widely used models and it is usually trained by means of error back-propagation algorithms. For instance, in the work of Kim et al (1991), BPNNs have been used to develop a simple diagnostic model for the purpose of fault detection in rotating machinery systems. In the work of Liu and Mengel (1992), back-propagation networks have been employed to classify the condition of ball bearings based on three indices extracted from the vibration signature. In a similar manner, Worden et al (1993) developed neural networks of the feed-forward back-propagation type that can locate and quantify damage in structural systems based on strain data records. The establishment of a BPNN requires the actual structure of the network, i.e. the number of layers and the number of neurons in each layer, to be known *a priori*. Further, network performance is quite sensitive to a particular architecture. The training of a BPNN is performed in a supervised learning mode through data pairs of both the input and desired output. Also, there is a strong possibility that the solution obtained using a back-propagation algorithm is not a global error minimum but a local one (Rao and Rao, 1993). The BPNN algorithms are slow in learning and a large amount of training data is required.

Self-organizing mapping (SOM) networks can successfully perform clustering and feature extraction without requiring information beforehand about the actual classes of objects. Hence, they are suitable for condition monitoring and diagnostics of machinery systems wherein minimal *a priori* information is available for training purposes. The associated one-layer neural network is developed during the process of SOM and the training of this network is performed in an unsupervised learning mode. The usage of the SOM networks to perform condition identification and classification is explored in this chapter.

4.2 Pattern Clustering and Classification

As mentioned in the previous chapter, condition monitoring and diagnostics of machinery based on measured on-line signals involve the extraction of features so as to determine whether a fault exists (detection) and if so, to assess the nature of the fault (classification). Typically, the inputs from which the signal features are extracted, are a series of large data samples or multi-dimensional sets of data vectors. Further, each and all of the set members consist of real numbers corresponding to the diagnostic signal for a given condition of the machinery. The data must be compressed to lower-dimensional performance-indicating vectors called features while still maintaining the same level of information with respect to machine performance. In Chapter 3, the set of indices representative of various faults in a machine has been defined as a feature (symptom) vector $\boldsymbol{X}=\{X_1, X_2, \ldots, X_N\}^T$. On the other hand, different machine conditions or malfunctions that can be clearly identified from the on-line machine signal, are taken as the elements of the failure (condition) vector $\boldsymbol{Y}=\{Y_1, Y_2, \ldots, Y_M\}^T$. The identification and classification problem of machinery monitoring can be described as

$$\boldsymbol{Y} \Leftarrow \boldsymbol{R} \Rightarrow \boldsymbol{X} \tag{4.1}$$

where $\boldsymbol{R}$ is the fault-symptom correlative relationship between the feature vector and the failure vector, i.e. a transformation set. To find a proper set $\boldsymbol{R}$, is to describe the relationship between each of the elements of $\boldsymbol{X}$ and $\boldsymbol{Y}$. After $\boldsymbol{R}$ has been defined, it can be applied on an observation of the feature vector $\boldsymbol{X}$ to find the corresponding failure vector $\boldsymbol{Y}$. This is essentially a problem of condition identification.

Patterns represented by several parameters (indices or signal features) in a multi-dimensional space, can be observed as a distribution of a number of clusters. For example, the multi-element vector $\boldsymbol{X}$ in Eq. (4.1), can be represented by a point in the multi-dimensional feature space. When a given group of data samples of $\boldsymbol{X}$ is obtained from a machine which has a certain type of malfunction, the data points will show similarity to each other. Hence this group of points should gather in the feature space close to each other as a cluster. Clustering can be viewed as a process of grouping similar objects and separating dissimilar ones (Zurada, 92). If it is assumed that the number of clusters, k, is known *a priori*, the patterns that correspond to a physical phenomenon can be mapped onto a vector as $\{W_1, W_2, ..., W_k\}$, where W_i denotes the "centre" position of the i-th cluster in the feature space. The problem of pattern clustering can now be generalized to the use of a small number of indicators (centres) to approximate the distribution of a large number of sets of data samples obtained from the physical process. Mathematically, the pattern clustering problem can be stated as (Kohonen, 1990)

$$E = \int \|X - W_c\|^r \, p(X) \, dX \qquad (4.2)$$

$$\|X - W_c\| = \min_i\{\|X - W_i\|\}$$

where $X \in R^N$ is a random vector with probability density function $p(X)$, $W_i \in R^N$, $i=1, ..., k$ are centres and dX denotes the volume differential in the N-dimensional space that corresponds to X. The problem is to place W_i in the X space such that the expected r-th power of the reconstruction error E is minimal. Such a distribution of W_i is said to be optimal in the sense of representing $p(X)$ by W_i. Since the problem has no closed-form solution, iterative algorithms are generally required to obtain the desired approximation.

A number of traditional algorithms (Zurada, 1992), such as K-means algorithm, maximum-distance technique and isodata algorithm are used in pattern mapping techniques. Even though the error E in Eq. (4.2) decreases monotonically when the K-means algorithm is used, it has been shown through empirical results that the convergence of the iteration process only leads to local minima and not to global minima. Further, this convergence to local minima highly depends on the initial positions of the centres W_i. Self-organization maps are more suitable for the pattern clustering and classification problem represented by Eq. (4.1) and have been successfully applied recently to the tasks of feature extraction and feature mapping. The weights of the networks are automatically adjusted to fit the distribution of input training data. The SOM neural network, which can be considered as a statistic classifier, can perform the linear or non-linear transformation that the set $\boldsymbol{R}$ does. Therefore, the SOM algorithm is employed in the present work and is summarized below.

4.3 Self-Organizing Mapping Algorithm

The distribution of training data in a higher-dimensional sample space can be topologically mapped onto a virtual one- or two-dimensional space that is represented by self-organized neurons. The SOM neural network is a set of these self-organized neurons and is represented as a discrete lattice of units. For a one-dimensional map, the lattice is represented by $\{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_k\}$, where $\boldsymbol{W}_i$ is a weight vector that has the same dimension as that of the training vectors.

The self-organization procedure consists of two steps. The first step is to find the best matching unit $\boldsymbol{W}_c$ to the current input vector $X(t)$, which is called a winning unit. The second step is to modify the weight vectors of both the winning unit and its neighbour units, so as to reduce the distance between them and the input vector. The updating process of the weights in discrete-time notation may be stated as

$$\boldsymbol{W}_i(t+1) = \boldsymbol{W}_i(t) + \alpha[X(t)-\boldsymbol{W}_i(t)], \qquad i \in H_c(t) \tag{4.3}$$

$$\boldsymbol{W}_i(t+1) = \boldsymbol{W}_i(t), \qquad i \notin H_c(t) \tag{4.4}$$

where,

$$\| X(t)-\boldsymbol{W}_c(t) \| = \min_i \{ \| X(t)-\boldsymbol{W}_i(t) \| \} \tag{4.5}$$

$i=1, \ldots, k$. $\boldsymbol{W}_c$ is the best matching unit corresponding to the current input data $X(t)$. Further, $H_c(t)$, designated as the neighbourhood function, is a symmetric index subset with the centre c and denotes the current spatial neighbourhood of W_c. In Eq. (4.3), the constant α is a scalar-valued adaptation gain called learning rate, such that $0<\alpha<1$. Both the regions of $H_c(t)$ and the value of α decrease with time t.

Since the learning rate α depends on both the training time and on the size of the neighbourhood, Eq. (4.3) can be rewritten as (Zurada, 1992)

$$W_i(t+1) = W_i(t) + \alpha(H_u,t)[X(t)-W_i(t)], \qquad i \in H_c(t) \tag{4.6}$$

By denoting the coordinates of cells c and i by the vectors r_c and r_i respectively, a proper value of $\alpha(H_u,t)$ can be obtained from

$$\alpha(H_u,t) = h(t)\exp[-\|r_i-r_c\|/\sigma^2(t)] \tag{4.7}$$

where $h(t)$ is the learning factor and the exponential term is the neighbourhood function. Both $h(t)$ and $\sigma(t)$ are suitable decreasing functions of time.

The training can be performed in an unsupervised mode and the network undergoes a self-organization process. Further, i is usually defined in a one- or two-dimensional metric space. The training algorithm is now described in terms of computational steps:

Step 1. Locate each of the k units randomly or at equal distances in the sample space, and consider this to be the initial position of the neuron.

Step 2. Given a randomly chosen input training data vector $X(t)$ in the sample space, find the unit j which is closest to $X(t)$ according to

$$\|X(t)-W_j(t)\| = \min_i\{\|X(t)-W_i(t)\|\} \tag{4.8}$$

where t is the discrete iteration step and $i=1, \ldots, k$. It may be noted that in the above expression, the Euclidean distance is used as a metric norm.

Step 3. Define a symmetric neighbourhood of units surrounding the winning unit j, $H_j(t)$, and adjust the weights of the winner and all its neighbourhood units according to

$$W_i(t+1) = W_i(t) + \alpha(H_{ij},t)[X(t) - W_i(t)], \qquad i \in H_j(t) \tag{4.9}$$

$$W_i(t+1) = W_i(t), \qquad i \notin H_j(t) \tag{4.10}$$

where $\alpha(H_{ij},t)$ is the scalar quantitative learning rate which monotonically decreases with the iteration number t.

Step 4. Reduce the neighbourhood function and learning rate $\alpha(H_{ij},t)$, increase iteration number t and return to step 2.

The learning rate and neighbourhood function that are used in the above algorithm can be given as below (Cherkassky and Lari-Najafi, 1991). The learning rate for the unit i in the neighbourhood of the winning unit j can be given by

$$\alpha(H_{ij}, t) = b(t)\exp\left[\frac{-|i-j|}{(b(t)s_0)^2}\right] \tag{4.11}$$

where s_0 is the number of neurons and $b(t)$ is the learning factor. For practical applications, the learning factor for the winning unit is given by the empirical relationship

$$b(t) = b_0(b_e/b_0)^{t/t_m} \tag{4.12}$$

where b_0 and b_e are the initial and final values of the learning factor, t_m is the prescribed maximum number of iterations which is usually defined as the product of the training set size and the number of times this set is recycled or repeatedly presented to the network during training. It may be noted here that whereas a gradual decrease in the learning rate is typical for neural network training, this is viewed as a special case of stochastic approximation (Cherkassky and Lari-Najafi, 1992), a similar decrease of the neighbourhood function does not have any obvious statistical interpretation.

The problem of machine fault identification has been formulated as a pattern clustering and classification problem earlier in this chapter. Therefore, a new approach for machine fault identification and classification is to cluster the patterns of different conditions of machinery by a number of neurons of a neural network through the above self-organizing process. After learning, each neuron goes to the centre of a cluster of data sample points in the feature space, and it represents a machine condition or fault corresponding to that cluster. Further, a new observation of the feature vector can be classified into one of those patterns, when it matches a pattern represented by the corresponding neuron. In such a way, the transformation between the feature vector and the fault vector is performed by a self-organizing map. Applications of the above SOM algorithm to condition classification of both a ball-bearing system as well as rotor systems are fully demonstrated in the next two sections, as examples of the implementation of the new approach.

4.4 Bearing Condition Classification

Vibration data were acquired from a type 308E ball-bearing having 8 rolling elements. The bearing was rotated at 1,470 rpm in a test machine and loaded with a circumferentially symmetric radial force of 20,800 N. An accelerometer was mounted on the bearing housing and its output was linked to a computer-based monitoring system. A sampling frequency of 5,000 samples per second was used and the digitized raw data stored in off-line files. Four different cases of bearing condition are considered: 1) defect-free bearing; 2) bearing with rolling element defects; 3) bearing with its inner ring sliding on the shaft; and 4) completely damaged bearing. Typical vibration signals of the four cases have been collected over multiple revolutions and a portion of the signals for each case is shown in Figure 4.1-4.4. As can be observed from Figures 4.1-4.4, the vibration level is very low when the bearing was defect-free; impulse peaks are generated due to ball defects; a low frequency (at the bearing rotational frequency) sine wave is present in the vibration signal which is due to sliding; and the signal is complex when the bearing is completely damaged. For bearing condition classification, these four cases are represented in Eq. (4.1) by the failure vector $Y=\{C_1, C_2, C_3, C_4\}^T$, where C_i is the i-th case of the above four cases under consideration. Four indices that are extracted from the bearing vibration signal are chosen as features: Peak-to-Peak value, Absolute Mean value, Crest Factor and Arithmetic Mean of the frequency spectrum. For clarity, these four indices are described briefly below:

The Peak-to-Peak value (PP) is the difference between the maximum and minimum amplitude of the vibration signal. It is sensitive to unusual impulses generated by bearing faults (Collacott, 1979).

The Absolute Mean (AX) is an index in the time domain and is the average value of absolute amplitudes according to

$$AX = |\bar{x}| = \frac{1}{n}\sum_{i=1}^{n}|x_i| \tag{4.13}$$

where x is a series of acquired data samples of x_i, i=1, 2, ..., n. The value of AX is small when most of the amplitudes of the signal are small and it is high when the bearing generates large-amplitude vibration.

The Crest Factor (CR) (Lipovszky et al, 1990) is another index in the time domain and is based on the maximum amplitude of the vibration as defined by

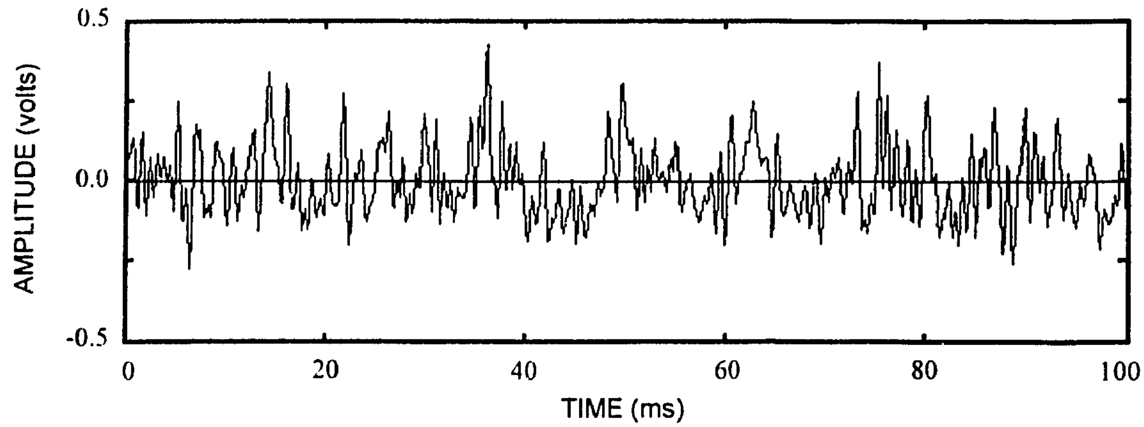


Fig. 4.1 Typical signal from an accelerometer for a defect-free bearing

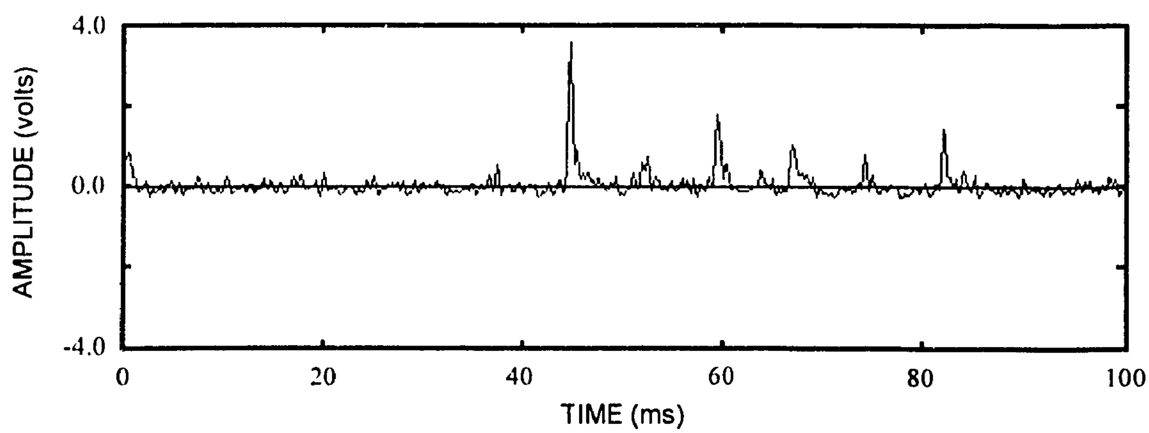


Fig. 4.2 Typical signal from an accelerometer for a bearing with ball defects.

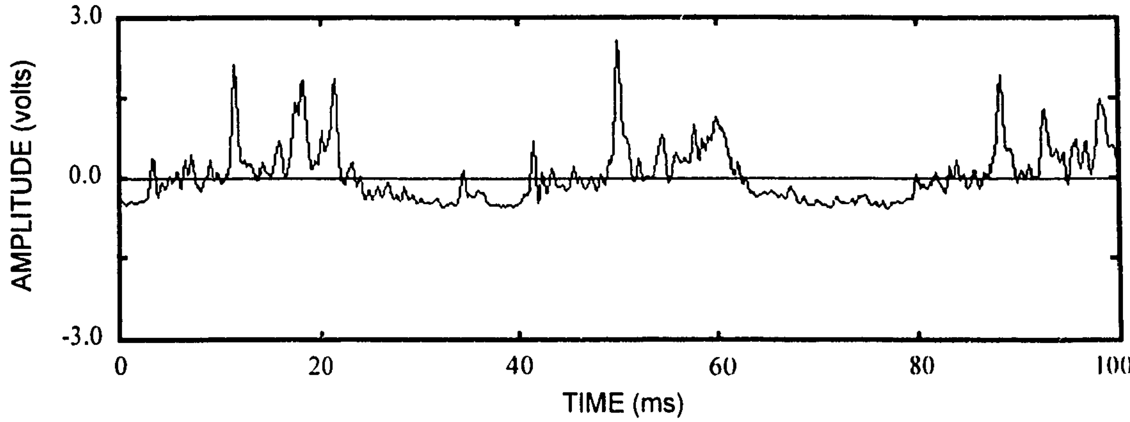


Fig. 4.3 Typical signal from an accelerometer for a bearing with the inner ring sliding at the shaft.

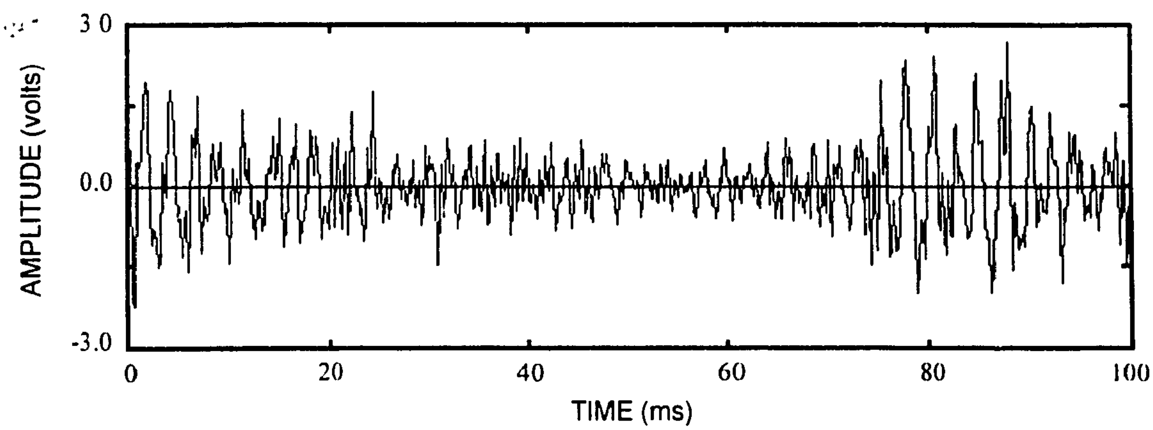


Fig. 4.4 Typical signal from an accelerometer for a completely-damaged bearing.

$$CR = \frac{|x|_{max}}{RMS} \tag{4.14}$$

where RMS is the root mean square of the signal. The RMS feature measures the average vibration energy. With the development of bearing faults, values of $|x|_{max}$ increase rapidly but the RMS value does not increase correspondingly at the early stages of the fault. However, when the bearing condition deteriorates due to further development of the faults, the RMS value will increase faster than $|x|_{max}$. Hence, the value of the index CR initially increases and then decreases during the development of bearing faults.

The Arithmetic Mean (AM) is an index defined in the frequency domain and is given by the empirical formula (Mathew and Alfredson, 1984)

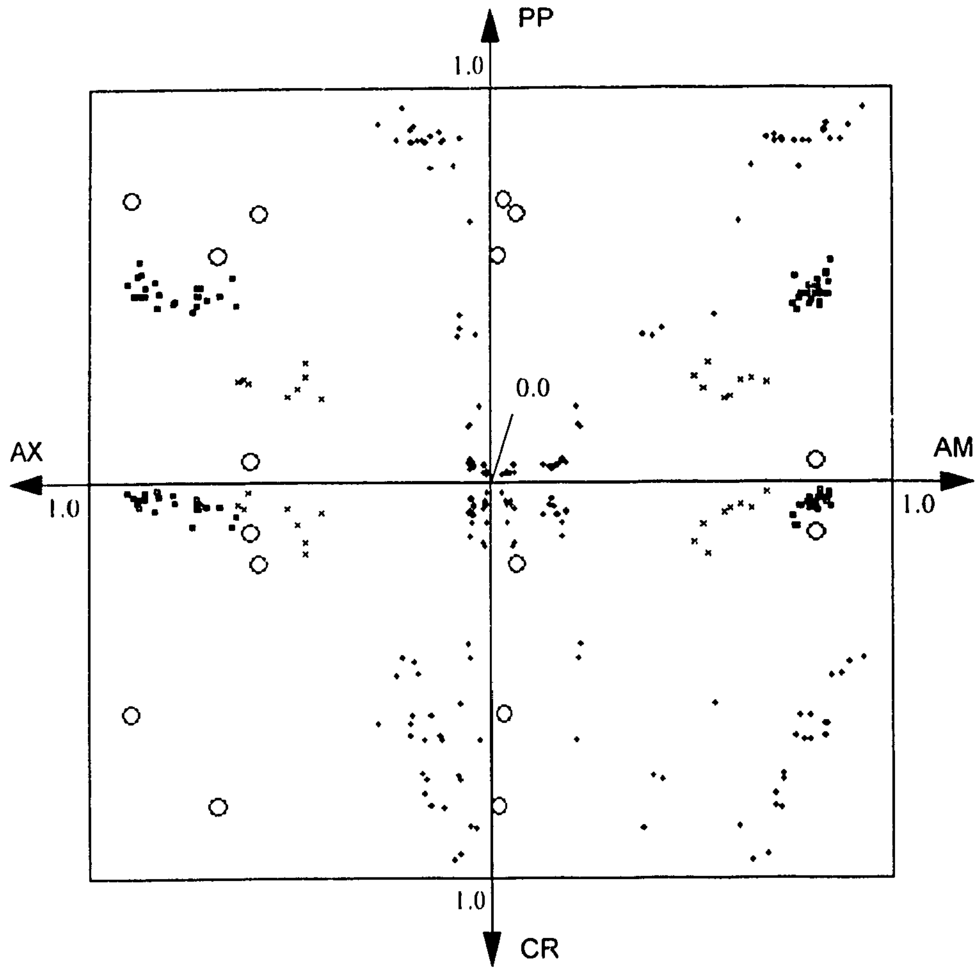
$$AM = 20\log\{(\frac{1}{n}\sum_{i=1}^{n} A_i)/10^{-5}\} \tag{4.15}$$

where A_i is the magnitude of the i-th frequency component in the Fourier spectrum of the bearing vibration signal. Normally, the value of AM increases monotonically within the service life of the bearing. Additional details about the above four indices have been published by El-Karmalawy (1993), and Kim and Lowe (1983).

The bearing conditions, corresponding to the vector $\boldsymbol{Y}$, are represented by the vector $\boldsymbol{X}$ of the above four indices as $X=\{PP, AX, CR, AM\}^T$. Feature mapping has been performed by a self-organization neural network with four neurons in an array which is represented by $\{\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3, \boldsymbol{W}_4\}$, where the weights are in the form, $\boldsymbol{W}_i=\{PP_i, AX_i, CR_i, AM_i\}^T$. The pattern recognition problem then reduces to mapping the data samples onto these four units of neural network with each one denoting a particular case of bearing condition. The trained neural network can then be used to perform condition classification. If a given set of the feature vector $\boldsymbol{X}$ is closest to a particular neuron $\boldsymbol{W}_i$, then $\boldsymbol{X}$ can be considered as the same pattern that is represented by the unit W_i.

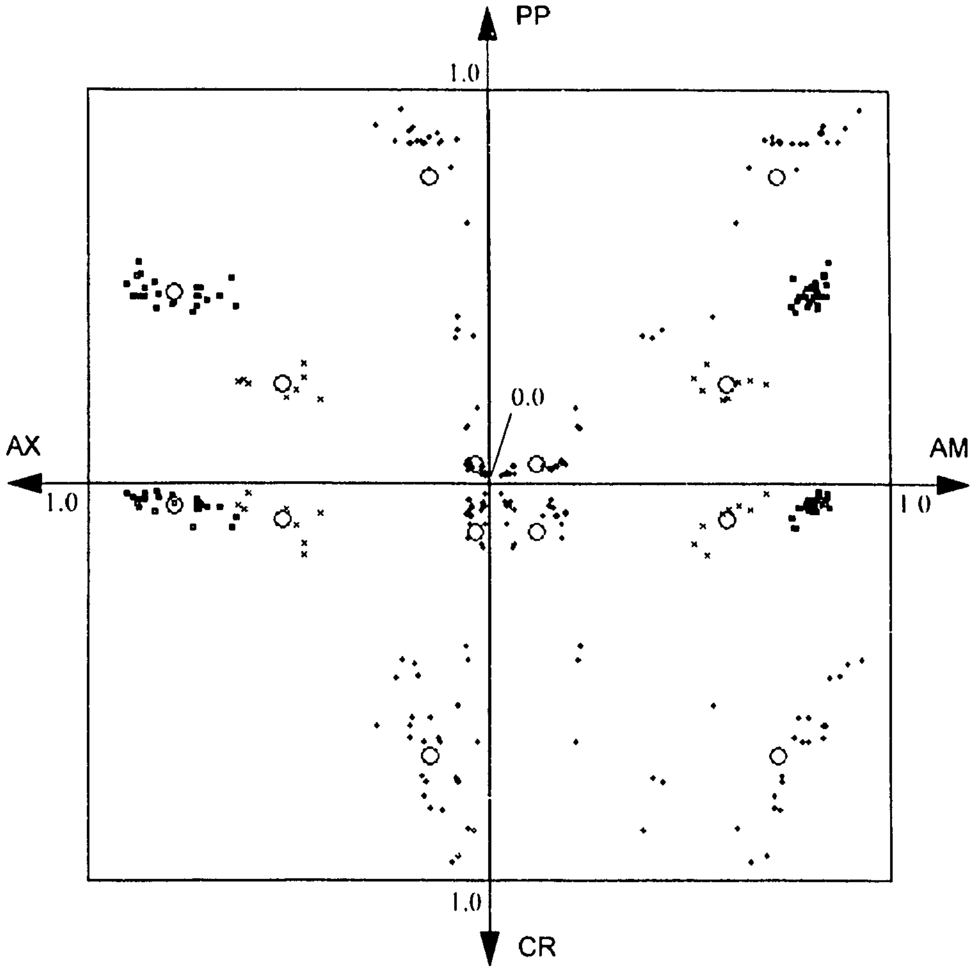
The training of the neural network was accomplished by a total of 80 sets of feature samples (vectors X). The selected samples comprised 24 sets from case C_1, 24 from C_2, 8 from C_3 and 24 from C_4. Before training, all the sample values of the indices PP, AX, CR and AM, were normalized so as to have the same range of values in [0, 1]. It may be noted that such a normalization scheme is necessary especially because the Euclidian distance is used as a metric norm. Otherwise, the co-ordinate axes of the feature space that is formed by the vibration indices will not be of the same scale. While calculating and using the Euclidean metric norm during the training of the SOM network, this situation causes unequal influence thus resulting in erroneous training result. The parameters in Eq. (4.12) of the learning factor were chosen as: the initial value b_0=0.6, the final value b_e=0.03, and the prescribed range of iterations t_m=120x80=9,600. This means that the 80 sets of data are recycled 120 times during training. Before each cycle of the data was presented as input to the neural network, the order of presentation was rearranged into a new random series. In other words, a sample set of features has been randomly selected from among the 80 sets during each training iteration. This way, the possible influence of the order of inputting the sample data sets has been eliminated.

The initial positions of neurons in the four-dimensional feature space has been randomly located as shown in Figure 4.5. In this figure, the entire data has been projected onto each of the hyperplanes determined by two orthogonal axes in the four-dimensional space. Four different types of dots are used to illustrate the data samples from the four cases (clusters). Similarly, the initial locations of $W_i(0)$, i=1, ..., 4, have also been projected onto the hyperplanes, which are represented by small circles in Figure 4.5. The final positions of the units, after training is completed, are shown in Figure 4.6, where



- bearing in good condition
- bearing with defects at balls
- bearing with its inner ring sliding on the shaft
- completely-damaged bearing

Fig. 4.5 Data samples for bearing condition classification and the initial positions of the four weights.



- bearing in good condition
- bearing with defects at balls
- × bearing with its inner ring sliding on the shaft
- ▪ completely-damaged bearing

Fig. 4.6 Data samples for bearing condition classification and the final positions of the four weights.

each of the four neurons has been located at the centre of the corresponding cluster.

Once the patterns have been classified and represented by a self-organizing map, the one-dimensional neural network can be applied to bearing condition identification. For any given feature set X, the pattern to which it belongs, can be determined by finding the matching unit closest to X using Eq. (4.5). The results of this classification are one-hundred percent correct for the prescribed data among which 80 sets or some others have not been used in the training. This excellent result demonstrates the potential application of self-organizing neural networks in pattern recognition and condition monitoring problems.

4.5 Rotor Condition Classification

The vibration signals were measured by velocity sensors from four rotating machines, two pumps, a motor, and a compressor. For each machine, the vibration signals from both, good and malfunctioning conditions were recorded at proper sampling frequencies. Data samples of digitized signal were provided by Mr. El-Karmalawy. The details of the set up, machinery and the measurement can be found in his thesis (El-Karmalawy, 1993). Rotor system problems were analyzed and classified into seven conditions based on previous knowledge and experience (El-Karmalawy, 1993). The observed seven different health conditions (cases) of the monitored rotating machines are: 1) good condition; 2) slight unbalance; 3) severe unbalance; 4) resonance; 5) misalignment with unbalance; 6) pure misalignment; and 7) oil whirl in journal bearing. The description of the above faults can be found in literature, such as (Eshleman and Jackson, 1992) and (Lipovszky et al. 1990). Typical signature spectra of the vibration signals for the seven cases are shown in Figures 4.7 to 4.13, respectively.

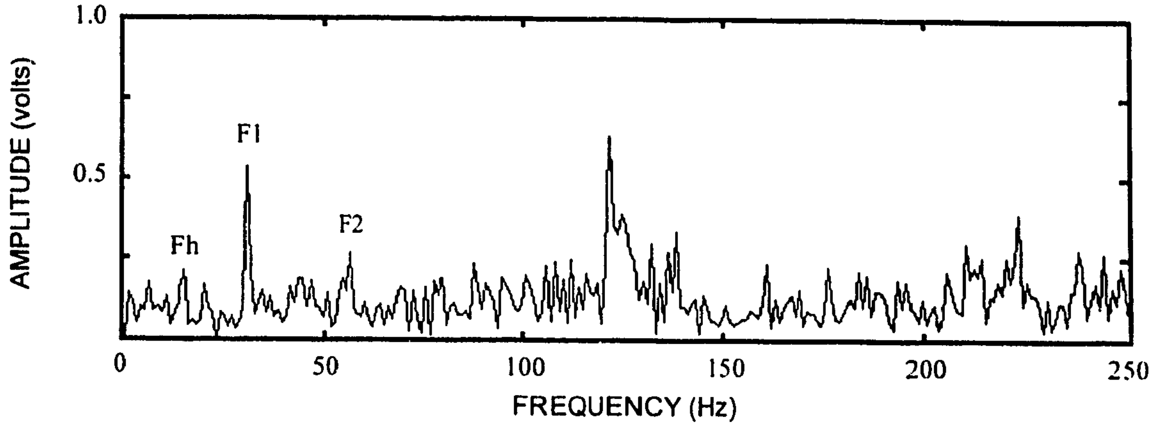


Fig. 4.7 Typical vibration spectrum for fault-free rotor system.

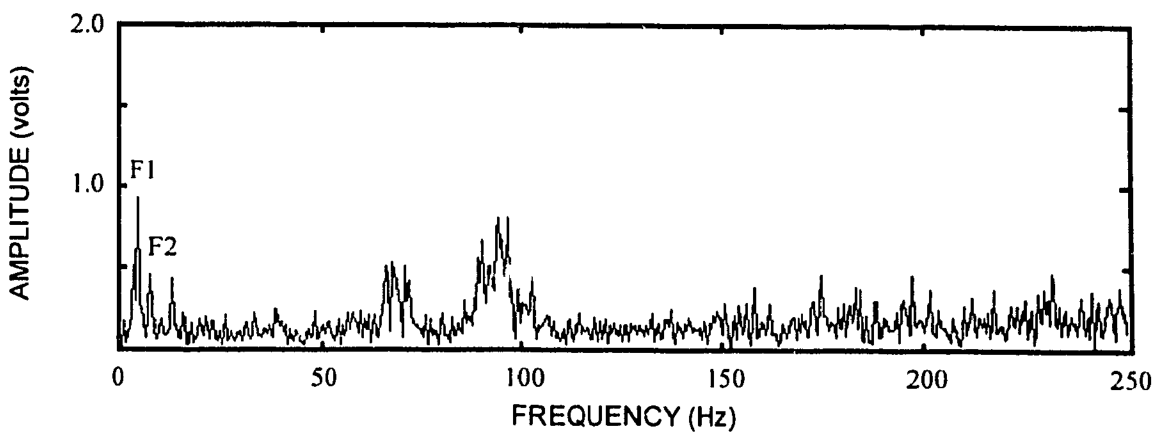


Fig. 4.8 Typical vibration spectrum for slight unbalance.

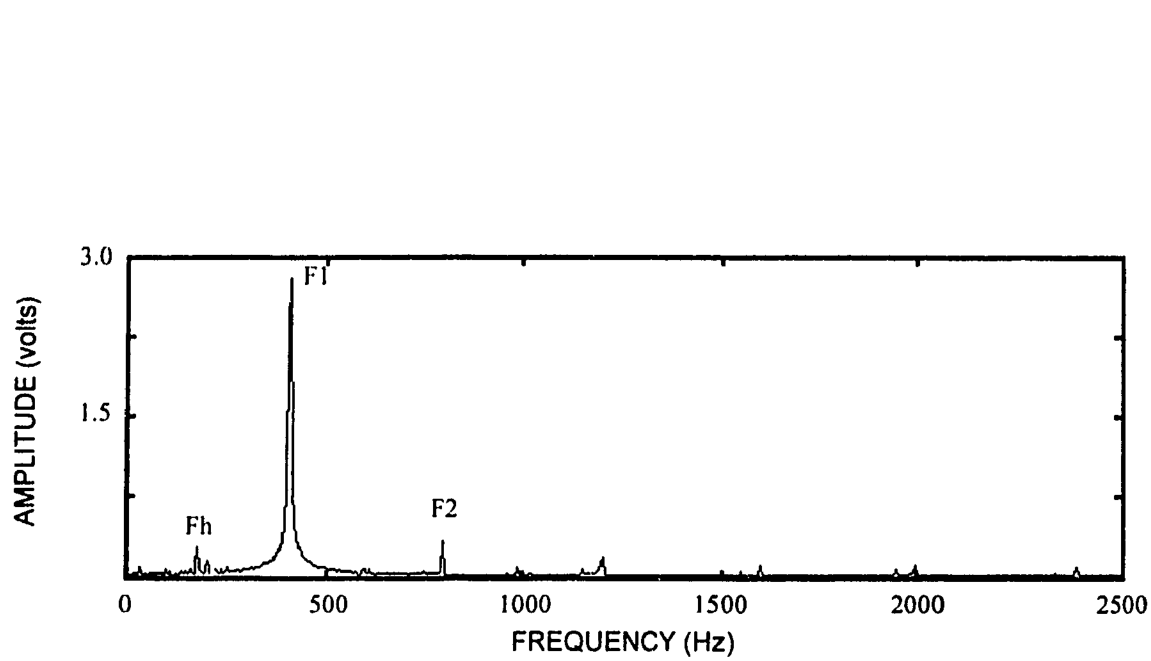


Fig. 4.9 Typical vibration spectrum for rotor unbalance.

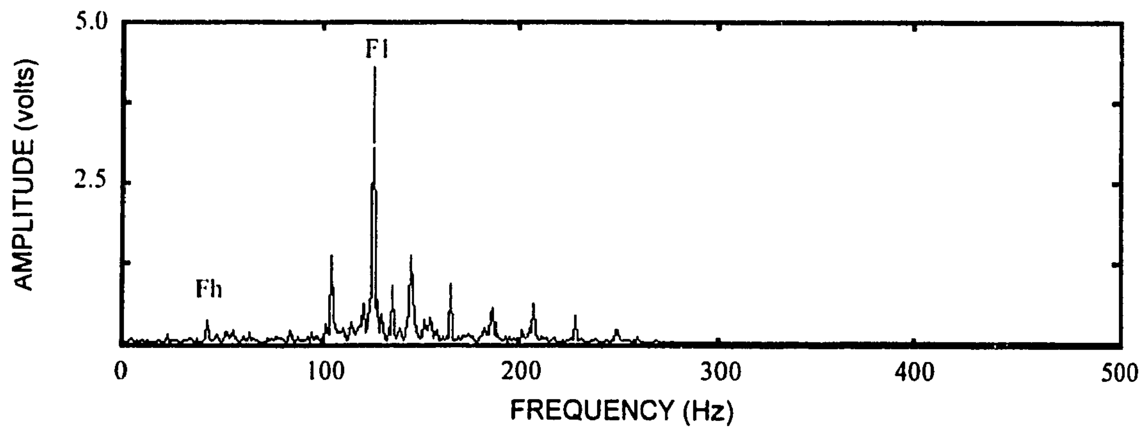


Fig. 4.10 Typical vibration spectrum for resonance.

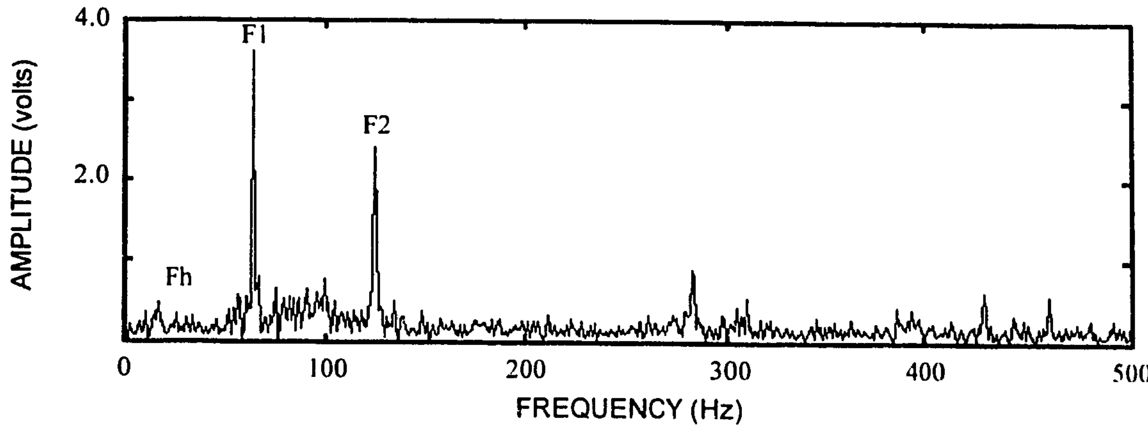


Fig. 4.11 Typical vibration spectrum for misalignment with unbalance.

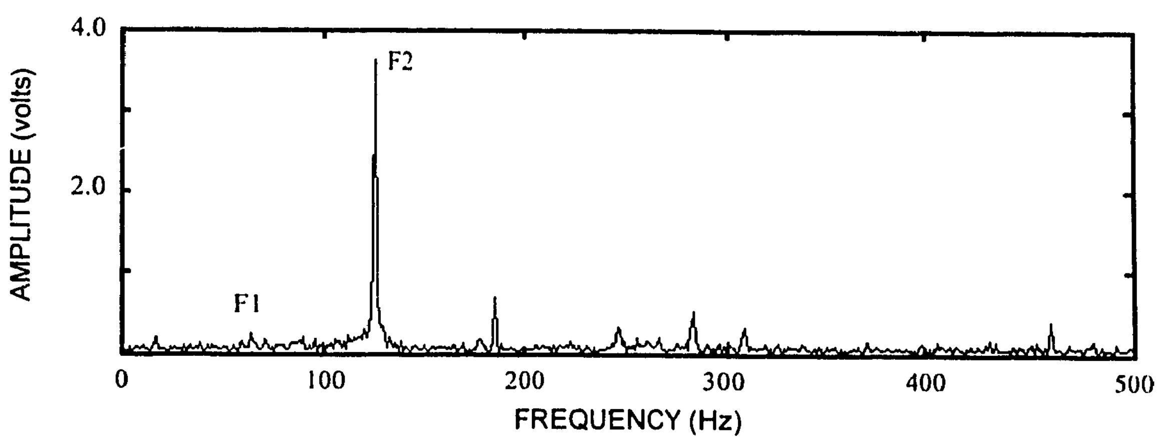


Fig. 4.12 Typical vibration spectrum for misalignment.

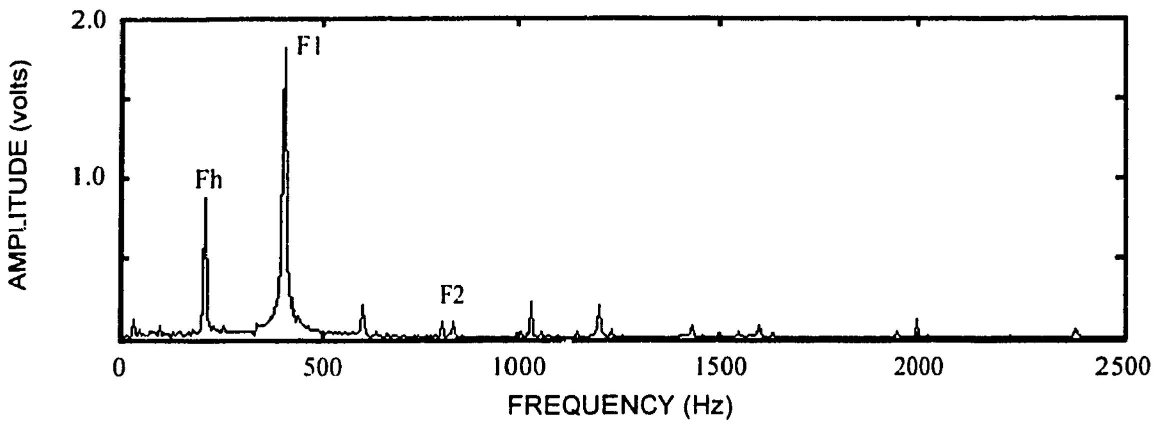
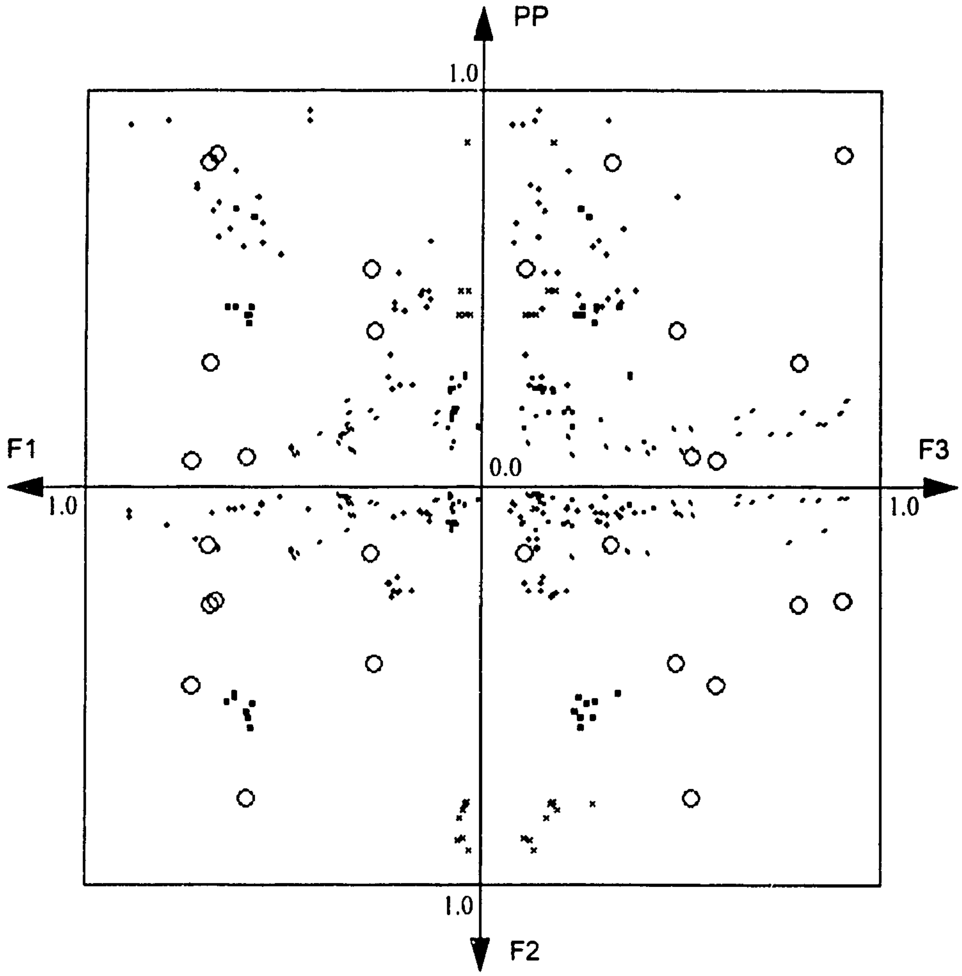


Fig. 4.13 Typical vibration spectrum for oil whirl in the journal bearing.

The following four features were chosen as system condition indicators: Peak-to-Peak value, magnitude at first harmonic of the rotational frequency, magnitude at second harmonic of the rotational frequency and the maximum magnitude of vibration power at a frequency which is around 42% of the rotational frequency. The Peak-to-Peak (*PP*) value shows the maximum variation of the amplitudes of the vibration signal in the time domain. The magnitude at the rotational frequency (*F1*) is important to problems of resonance and unbalance. The value of *F1* is small for a rotor in good condition, as in case 1 (Figure 4.7), and it is large for resonance conditions as in case 4 (Figure 4.10). For typical unbalance problems as in cases 2 and 3 (Figures 4.8 and 4.9), *F1* assumes intermediate values. The magnitude at the second harmonic of the rotational frequency (*F2*) is usually considered an indicator of misalignment problems. For a pure misalignment problem as in case 6 (Figure 4.12), the value of *F2* is even greater than that of *F1*. Both *F1* and *F2* show up in case 5 (Figure 4.11) where misalignment and unbalance are present. The maximum magnitude at a frequency around 42% of the rotational frequency (*Fh*) has been selected as the fourth index. The value of *Fh* depends on the stability of the oil film in journal bearing, i.e. it is indicative of the oil whirl problem. The vibration signal generated by an air compressor subject to incipient oil whirl has been recorded in case 7 (Figure 4.13).

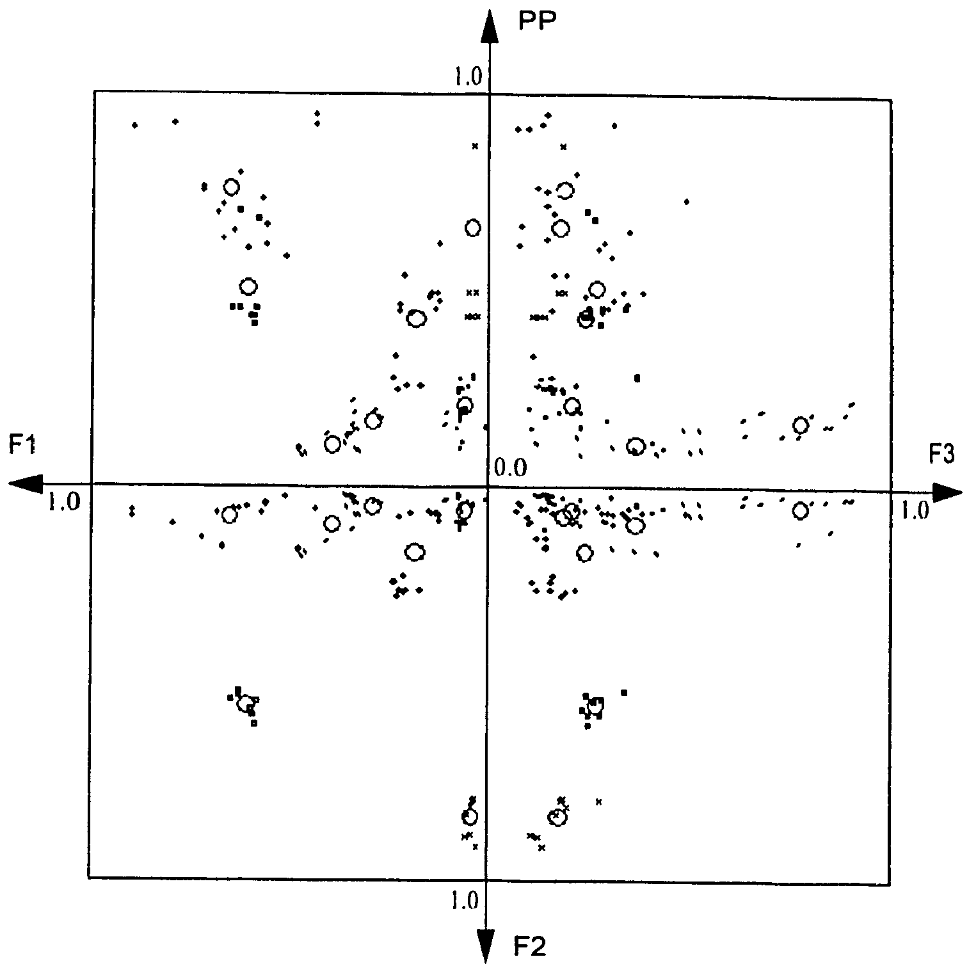
Since the vibration data were measured on different machine systems, the normalization for training purposes has been performed using the method presented in Chapter 3. The last step of the normalization is to force the ranges of the values of the four indices to be in [0, 1]. This essentially means that all the four indices have been considered to be equally important.

The feature vector has been defined as $X=\{PP, F1, F2, Fh\}^T$ and the failure vector



- good condition
- slight unbalance
- serious unbalance
- resonace
- misalignmant with unbalance
- pure misalignment
- oil whirl

Fig. 4.14 Data samples for rotor condition classification and the initial positions of the seven weights.



- good condition
- slight unbalance
- serious unbalance
- resonace
- misalignmant with unbalance
- pure misalignment
- oil whirl

Fig. 4.15 Data samples for rotor condition classification and the final positions of the seven weights.

as $Y=\{C_1, C_2, ..., C_7\}^t$. The neural network with seven neurons in an array is represented as $\{W_1, W_2, ..., W_7\}$, where the weight W_i is a column vector of the four indices, i.e. $\{PP_i, F1_i, F2_i, Fh_i\}^t$, i=1, ..., 7. The initial positions of the neurons have been randomly chosen and are shown in Figure 4.14. The four sub-parts of Figure 4.14 show the projections of the data and weights from the multi-dimensional space onto each hyperplane that is determined by the two axes in the four-dimensional space. Totally, 91 sets of data samples have been used for training. The number of sample sets for different cases is not the same. The maximum number of sample sets is 21 and the minimum is 8. The learning factor parameters in Eq. (4.12) have been chosen as b_0=0.8, b_c=0.03, and the maximum iteration t_m=140x91=12,740. The final positions of the weights are shown in Figure 4.15. As in the case of bearing malfunction classification, the usage of this neural network to identify any given index vector from among 91 data sets, has met with one-hundred percent success.

The results of using SOM to classify machine faults have been compared with back-propagation neural networks, and are presented in the next section after a brief introduction of BPNN algorithms.

4.6 Condition Classification with BPNNs

A back-propagation neural network (Rao and Rao, 1993; Elkordy et al, 1994) is illustrated in Figure 4.16. It consists of three layers: an input layer, a hidden layer, and an output layer. There are certain numbers of neurons (nodes or units) in each layer, and these neurons in the same layer are fully connected to the nodes in the next layer by a synaptic weight matrix. In general, there can be more than one hidden layer in such a network. A BPNN is a feed-forward multilayer network. The information contained in the

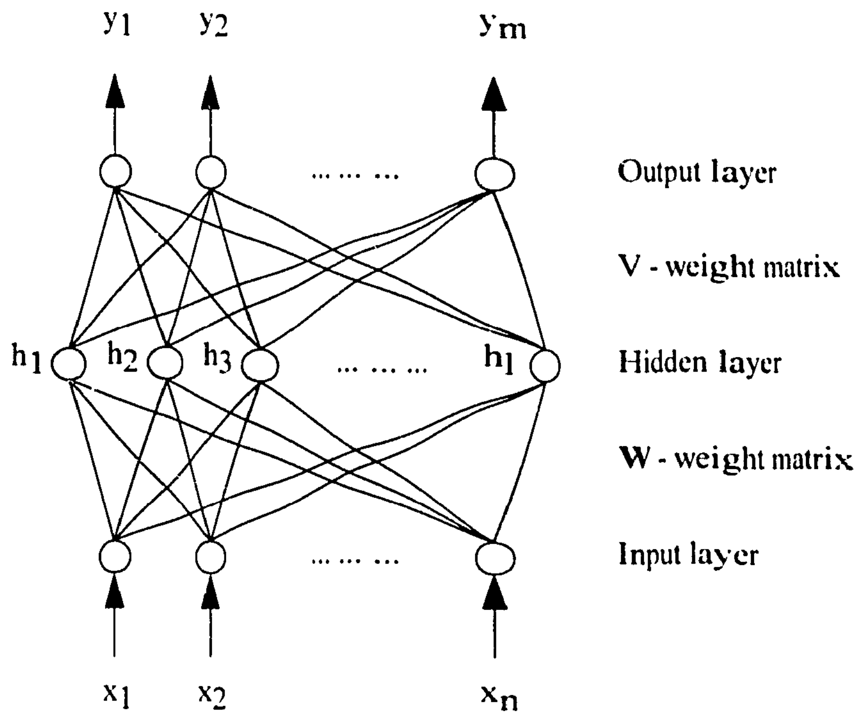


Fig. 4.16 Feed-forward back-propagation neural network.

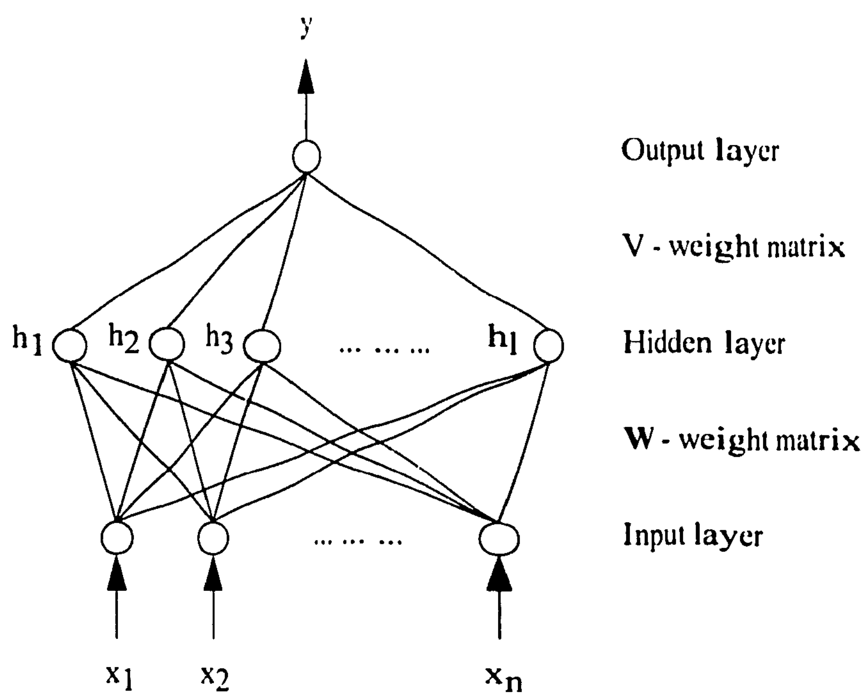


Fig. 4.17 Feed-forward back-propagation neural network with one output unit.

input signal is recorded into an internal representation by the hidden layers and weights which perform the transformation from the input to the output.

The training of a BPNN takes place in an iterative fashion, and is performed in a supervised learning mode with data pairs of both the input and desired output. Each iteration cycle involves a forward propagation step followed by an error backward propagation step to update the connection weights. The forward propagation step starts when the nodes in the input layer receive signal data. The forward propagation proceeds through the hidden layer up to the output layer by computing the activation values of the nodes in those layers. Each node receives its input from other nodes. The output of the j-th node in the hidden layer is calculated from (Rao and Rao, 1993):

$$h_j = f\left(\sum_{i=1}^{n} x_i W_{ij} + \theta_j \right), \qquad j=1, \ldots, l \qquad (4.16)$$

where, n is the number of nodes in the input layer, l is the number of neurons in the hidden layer, and x_i is the input received by this node from the i-th input neuron. W_{ij} is the connection weight which links the i-th input neuron to this node. θ_j represents an internal offset (threshold) value, and the activation function f is a sigmoid function defined as

$$f(z) = \frac{1}{1 + e^{-z}} \qquad (4.17)$$

Similarly, the output of the k-th node in the output layer is calculated from

$$y_k = f\left(\sum_{j=1}^{l} h_j V_{jk} + \tau_k \right), \qquad k=1, \ldots, m \qquad (4.18)$$

where, m is the number of neurons in the output layer, h_j is the j-th input which comes from the hidden layer, V_{jk} is the corresponding connection weight, and τ_k is the threshold.

The outputs of the network are the activation values of the output nodes. For the k-th node in the output layer during training, the error δ_k is computed from

$$\delta_k = y_k(1-y_k)(d_k-y_k), \qquad k=1, \ldots, m \tag{4.19}$$

where d_k is a given value of the desired output of node k corresponding to the present input. After the error values are computed, the error back-propagation step starts. The purpose of the back-propagation approach is to modify the connection weights and thresholds such that the difference between the calculated output of the network and the desired output is reduced. The weight between the j-th neuron in the hidden layer and the k-th output neuron is updated according to the following equation:

$$V_{jk}(t+1) = V_{jk}(t) + \mu h_j \delta_k \tag{4.20}$$

where, t is the discrete iteration step, and μ is a constant called learning rate. For the j-th neuron in the hidden layer, the error e_j is estimated from

$$e_j = h_j(1-h_j)\sum_{k=1}^{m} V_{jk}\delta_k, \qquad j=1, \ldots, l \tag{4.21}$$

The weights between the input layer and the hidden layer are updated through

$$W_{ij}(t+1) = W_{ij}(t) + \lambda x_i e_j \tag{4.22}$$

where λ is the learning rate. The offset parameters, τ and θ are treated as additional weight factors and updated as follows:

$$\tau_k(t+1) = \tau_k(t) + \mu\delta_k, \qquad k=1, \ldots, m \tag{4.23}$$

$$\theta_j(t+1) = \theta_j(t) + \lambda e_j, \qquad j=1, \ldots, l \tag{4.24}$$

This training process is repeated until the calculated outputs have converged sufficiently close to the desired output or, an iteration limit has been met. Forward and backward propagation in a BPNN which has more than one hidden layer is computed in a similar way.

A BPNN model with a single output neuron (Matteson et al, 1992), as shown in Figure 4.17, was employed to classify bearing faults by Liu and Mengel (Liu and Mengel, 1992). In that work, three vibration indices were used as inputs to the neural network, and the outputs were six different bearing states. The number of the input nodes was equal to the number of the vibration indices, i.e. three input neurons. After training, the single output neuron was able to give an output value, y, between 0.0 and 1.0 to denote the bearing condition (six different cases). BPNNs with one or two hidden layers, and with different numbers of neurons in each hidden layer were also tested.

Such a BPNN model with four input nodes and a single output node has been applied to both bearing fault identification and rotor system condition classification as described previously. In the bearing state classification, the same 80 sets of data samples are used as the input data in the training of BPNNs. A training data set is formed as $\{PP, AX, CR, AM, d\}$, where PP, AX, CR and AM are the four indices being defined in Section 4.4, and d is the desired output corresponding to the input. Following the approach proposed in (Liu and Mengel, 1992), the desired output has been chosen as 0.2 for the bearing condition of defect-free, 0.4 for the case of a bearing with rolling element defects, 0.6 for a bearing with its inner ring sliding on the shaft, and 0.8 for a completely-damaged bearing. Based upon the output of the network, the condition of the bearing can be classified to be in four categories according to the following rules: good bearing, 0.100-0.300; bearing with defects, 0.301-0.500; bearing with its inner ring sliding on the shaft, 0.501-0.700; and completely-damaged bearing, 0.701-0.900.

BPNNs with a single hidden layer, but different numbers of nodes in the hidden layer, have been tested. The training has been performed by using a computer simulation program designed by V. B. Rao and H. V. Rao (1993). In training, the learning rate has been set at 0.1 for all tests. Since a smaller number of recycles has been unsuccessful in the tests, the repetitive cycles to present the training data to the network is set at 4,000. This means that the number of iterations is 80x4,000= 320,000. The result is that for a BPNN with less than 6 nodes in its hidden layer, the success rate depends on the neural network structure, ranging from 87.5% to 98.8%. A BPNN with more than 6 nodes in its hidden layer, can perform the classification with one-hundred percent success, which is the same as the performance of the SOM approach illustrated earlier.

The data samples used previously in the rotor system fault classification have also been applied to train BPNNs. As before, the training set is formed as $\{PP, F1, F2, Fh, d\}$. Here, the desired output d is chosen as 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8 to denote the seven different cases respectively, that were considered in Section 4.5. The rule for classification is that the output of the BPNN should be close to the desired output within the interval of $d-0.5<y<d+0.5$. The training rate is 0.1 and the number of recycles is 5,000. BPNNs with one or two hidden layers have been tested. In those with one hidden layer, the number of neurons in the hidden layer has been selected from 2 to 25. The best result reached is 90.1% for the neural network with 9 nodes in its single hidden layer. In tests with two hidden layers, all combinations of 5 to 16 nodes in the first hidden layer and 5 to 16 nodes in the second hidden layer have been tried. The best result is 95.6% which is from a BPNN with 12 nodes in the first hidden layer and 8 nodes in the second hidden layer.

4.7 Discussion

It has been stated in Chapter 3 that a fundamental problem of MMD is the machine fault identification and classification. In the earlier sections of this chapter, an effective way of using a self-organizing mapping algorithm to solve this problem is presented. This new approach is based on the interpretation of machine fault (condition) identification and classification problem to be a problem of pattern clustering and feature extraction from multi-dimensional data sets. Neural networks are developed based on the SOM algorithm and training is performed in an unsupervised mode. The training of such a computationally-effective neural network takes only a few seconds on an IBM PC 486 system and thus is cost-effective. Four diagnostic indices have been extracted from the vibration signature and have been used for condition classification. Neural networks are developed based on self-organizing mapping algorithm and the training is performed in an unsupervised mode. The establishment of this self-organization neural network is easier than the implementation of conventional expert system approaches. A neural network can learn diagnostic knowledge from historical data. In using expert system approaches, human knowledge about the classification problems needs to be sufficient and accurate, as well as to be available in explicit form for symbolic representation. Further, the reasoning scheme employed should be able to use the knowledge properly and to yield correct results.

The SOM algorithm needs only one piece of information, that is, how may patterns (clusters) of machine conditions are to be presented in the feature space. On the other hand, details as to which data sample set belongs to what pattern must be given during the training of a BPNN.

It is also observed that a monotonic increase or decrease in the number of units does not lead to precise maps, i.e. the precise representation of the transformation set $\boldsymbol{R}$. For a particular problem, the optimum number of units is dictated by the interpretation of the failure set. For the problem of rotor system fault classification, neural networks with 5 to 10 units have been attempted. It is observed that when the number of units is less than seven (the number of prescribed cases), two different rotor conditions are classified into the same condition. Also in some cases, half of the samples from a particular case are attributed to a neighbour cluster and the other half is attributed to some other cluster. On the other hand, when the number of units is greater than seven, new clusters are designated which contain samples from one or more conditions which are not relevant to these clusters. Moreover, the fact that the classification is sensitive to the values of *Fh* is also seen from this investigation. The influence of *Fh* is due to the amplification of the values of *Fh*, which were originally smaller than the values of other indices, during the process of normalization.

In both the bearing and rotor condition classification problems considered in the present chapter, a larger number of iterations is used so that a smooth learning process is achieved. Notwithstanding this, the training process took only a few seconds on an IBM PC 486 system. Further, initial and final values of the learning rate have been properly selected in order to achieve the best results in terms of accurate positioning of neurons in the multi-dimensional space of data sets. It is observed that for a fixed value of the final learning rate, a larger value of the initial learning rate leads to a larger variation of the weights of each and all neurons throughout a larger learning period. On the other hand, usage of smaller values of the initial learning rate may have led to a local minimum or erroneous positions of neurons through fixing the final locations of the weights at an earlier time during training. In the present work, the ranges of values of the initial and final learning rates were taken to be 0.4 to 1.0 and 0.01 to 0.07 respectively.

CHAPTER 5

MULTIPLE-INDEX BASED TREND ANALYSIS USING NEURAL NETWORKS

The development of a new approach to Multiple-Index Based Trend Analysis for MMD (Zhang et al, 1994c; 1995d) is presented. This new approach centres around the simultaneous use of a number of diagnostic indices to provide a precise determination and prediction of machine conditions. As explained in Section 5.1, the new approach formulates the trend analysis of MMD into a regression analysis of estimating an unknown multivariate function from the pertinent diagnostic signal. The neural network solutions based on Self-Organizing Mapping (SOM) and Constrained Topological Mapping (CTM) algorithms which have been developed quite recently for global nonparametric regression, are introduced in Section 5.2. In Section 5.3, simulation results are used to clearly show the failure of the SOM and CTM algorithms in performing multi-dimensional function regression. The fundamental reasons behind the encountered deficiency are then systematically brought out. A new self-organizing mapping algorithm (Zhang et al, 1995b; 1995e) which overcomes the limitations of the existing SOM and CTM algorithms is developed in Section 5.4. The efficiency and computational superiority of the new algorithm are illustrated through simulation experiments. Further, this newly developed algorithm is adapted to perform multivariable trend analysis. Applications of the new approach of multiple-index based trend analysis, that uses the new SOM algorithm to practical problems of predicting the service life of both a bearing system and a gearbox, based on on-line vibration measurements, are fully demonstrated in Sections 5.5 and 5.6. Discussion of both the new diagnostic approach and the new SOM algorithm is presented in Section 5.7.

5.1 Multivariable Trend Analysis in MMD

Trend analysis in MMD, as defined in Chapter 3, is a diagnostic technique to quantify development of incipient faults. This analysis uses functional models that show the trend of continuous changes in machine health condition, in the degree of damage or in the service life of a machine system. It is essentially a function regression problem in terms of mathematical statistics. To date, only a univariate function, as given below, has been used in trend analysis for machinery condition monitoring.

$$Y = f(X) + e \tag{5.1}$$

where e is an unknown disturbance, X is a monitoring index extracted from the on-line diagnostic signal, and Y is an indicator, employed to monitor the operation of the machine system or the performance of its components. In this approach, the determination or prediction of machine condition is based on the value of a single diagnostic index X. Further, the global parametric method is commonly used for estimating the function f from a given set of data pairs $\boldsymbol{Z}_i=(X_i, Y_i)$, $i=1, 2, \ldots, n$. In this method, the unknown function is assumed to have a fixed form, where this form can be represented as a combination of some basis functions. The simplest and most commonly used parametric method is linear regression in which the functional form is taken as

$$f(X) = aX + b \tag{5.2}$$

where parameters a and b are obtained by the method of least squares.

However, in condition monitoring and fault diagnosis of machine systems, the state, behaviour, characteristics and conditions of machines are represented by a group of indices that are obtained from the diagnostic signals, with each index reflecting a certain aspect of the signal. Further, it has been clearly shown (Tranter, 1989; Lipovszky et al, 1990), that more than one diagnostic index should be used if precise and complete information about the fault or malfunction in the machine system is desired. This is due to the fact that there is no single index that can adequately represent the conditions of a machine and further, different types of faults or malfunctions are distinctly reflected in various different diagnostic indices. Hence, the widely-used single-index based trend analysis can not provide complete and accurate information about the conditions and malfunctions of a machine system. Therefore, in trend analysis, a number of diagnostic indices should be considered simultaneously. It is observed from existing works that such a trend analysis involving many diagnostic indices has not been employed so far for system monitoring and diagnosis. This is due primarily to the inherent complexity and difficulty involved in modelling the multivariate diagnostic function f.

In order to make a collective use of several diagnostic indices in trend analysis, a new approach, designated Multiple-Index Based Trend Analysis, is proposed. This new approach uses multivariate functions to quantify the development of machine faults or the changes in machine conditions. It will be shown later in this chapter that the collective use of several diagnostic indices through a multivariate function leads only to the extraction of precise and complete information about the state or malfunctions of machine systems.

This new approach essentially involves a multi-dimensional regression analysis, that is a regression fitting of a function Y of N independent variables that are represented by a vector $\boldsymbol{X}$, from a given set of n data points denoted by $(\boldsymbol{X}_i, Y_i)$, i=1, 2, ..., n, in (N+1)-dimensional sample space. Mathematically, this problem is stated as finding the diagnostic function f such that

$$Y = f(\boldsymbol{X}) + e \tag{5.3}$$

where e is a disturbance assumed to have a zero mean. Further, it is assumed that the distribution of e may depend on the vector $\boldsymbol{X}$ and that the distribution of data points $\boldsymbol{X}_i$ in R^N is usually not uniform. In Eq. (5.3), the function f may represent either a multi-dimensional curve or a multi-dimensional surface, depending on the particular physical problem being addressed.

In condition monitoring and diagnosis of machine systems, the operative states of the machine system at any instant of time are represented in an N-dimensional vector. Also, the health condition, the degree of damage or the measure of service life of a machine system, are represented by a scalar variable which is obviously a function of the operative states of the machine system. Condition monitoring systems which use on-line vibration measurements as the diagnostic signal, usually vibration measurements are taken at pre-fixed instants of time T_i, i=1, 2, ..., n, (or over the number of cycles of operation, or over any other suitable independent parameter) from which N number of diagnostic indices are extracted. These diagnostic indices are representative of the operative states of the machine system and essentially constitute the N-dimensional operative state vectors $\boldsymbol{I}_i=\boldsymbol{I}(T_i)$, i=1, 2, ..., n. The data points $(\boldsymbol{I}_i, T_i)$, i=1, 2, ..., n, are used to estimate the functional dependency between the operative state vector $\boldsymbol{I}$ and the service life of the machine that is denoted by T. This functional dependency is then used to predict the residual service life of a machine system based on the sampled values of the operative state vector. The problem of condition monitoring can be posed as a mathematical regression problem represented by Eq. (5.3), if the operative state vector $\boldsymbol{I}$ and the service life of the machine system T are interpreted to be the vector $\boldsymbol{X}$ and the function Y respectively. In this case, (i) f is a multi-dimensional curve, (ii) the distribution of $\boldsymbol{X}_i$ in R^N need not be uniform but a uniform distribution of Y in the (N+1)-dimensional sample space can be achieved through the proper selection of observation time instants T_i.

5.2 Multi-Dimensional Regression and SOM Algorithms

Regression fitting of a function Y of N independent variables represented by a vector $\boldsymbol{X}$, from a given set of n data points $\boldsymbol{Z}_i=(\boldsymbol{X}_i, Y_i)$, $i=1, 2, \ldots, n$, is considered herein. The regression analysis is central to many problems of engineering in industry. However, these regression problems significantly differ among themselves in the nature, characteristics and availability of the data set $(\boldsymbol{X}_i, Y_i)$, and also in the physical system parameters that are represented by this data set. The goal of the above regression analysis is to estimate an underlying mathematical function based on a finite number of possibly inaccurate data points, for the purposes of data reduction, data interpolation and data prediction (extrapolation). There are several approaches that can be employed to solve this multi-dimensional regression problem, including neural network algorithms (Specht, 1991).

If the functional form of f in Eq. (5.3) is assumed through an exploratory study of the data points and by considering the particular application, parametric models of the function can be estimated. In global parametric regression methods, the functional form of the regression surface or the curve f is assumed to be known, and the problem is reduced to estimating the parameters of the regression surface or curve. For high-dimensional regression, suggesting a model for the regression surface that fits well with the data is extremely difficult (Cherkassky and Lari-Najafi, 1991), and the results of regression analysis are entirely dependent on the correctness of the assumed model.

On the other hand, if none or very few general assumptions about the functional form of f are made, a more complex problem, known as nonparametric regression analysis, is to be solved. Nonparametric methods of regression analysis eliminate the problem of predicting a priori the true functional form of f by making none or very few general assumptions about the regression surface or curve. When the underlying functional form is non-linear, the most common approach is to consider the function f to be a piecewise linear function made up of several linear pieces that are joined continuously at points known as knots (Friedman and Silverman, 1989). Assuming the coordinates of the knot positions to be fixed, there are several parametric models that can fit a wide range of training data sets reasonably well. The performance of the estimated function then depends on three factors, viz., the number, the relative location and the positioning of knots. However, nonparametric regression is cumbersome for the following reasons:

The finite number of sample points implies that the problem itself is ill-posed (Poggio and Girosi, 1990). Hence, the number of data points required to estimate a function with a prescribed accuracy should grow exponentially with dimensionality. Another difficulty is the existence of unknown additive output noise which in turn poses difficulties in distinguishing between the variations in sample data due to noise and the variation due to their dependencies over the process parameter changes (Geman et al, 1992). It has also been recognized that better results can be achieved through a dynamic knot positioning approach than through the fixed knot positioning approach (Cherkassky and Lari-Najafi, 1991). In these circumstances, the self-organizing mapping networks described in the last chapter, can be employed to perform the nonparametric regression analysis, if the units of the SOM network are interpreted as the knots (Cherkassky and Lari-Najafi, 1991; 1992).

As has been pointed out in Chapter 4, the original self-organizing topological map (Kohonen, 1990) essentially consists of an array of units which are interpreted in this context as knots, $\{W_1, W_2, ..., W_K\}$. They respond to the input signals in an *orderly* fashion. The response, in a high-dimensional sample space, is such that (i) the spatial locations of the neurons on a multi-dimensional map reflect the distribution of given (training) data points in the sample space; and (ii) the topological ordering of the neurons in the array reflects the ordering of the input signals drawn from the input sample space. In this formulation, the network model is basically a mapping of input signals onto a low-dimensional discrete lattice of units called neurons. During this mapping, both the data distributions and the neighbourhood relations in the input sample space should be preserved to the extent permitted by the units of the network model. It may be noted that the ordering of neurons is irrelevant when using Kohonen's SOM to perform data clustering, but it is relevant and very important when using the SOM algorithm to perform function regression.

Application of the self-organizing algorithm developed by Kohonen for regression problems, however, has not been successful. This is due primarily to the inability of the SOM algorithm to produce functional maps that do not violate the topological ordering of knots (Ritter and Schulten, 1989). During learning, the unit selected as the best match will experience maximum modification. It is possible for this knot to move far enough to cross its immediate neighbour. If this occurs, the topological order of the knots is said to be violated. The problem becomes much more serious for multivariate functions where the SOM algorithm fails so frequently. Figure 5.1 (from Cherkassky and Lari-Najafi, 1991) shows the disordering of units formed by the original SOM algorithm when it was trained using six data points that represent a sine function with error. In many regression problems, most or all of the independent variables have been considered to have uniform distributions and to be monotonically varying (increasing or decreasing), such that the violation of the order of units in a network is mainly observed in the N-dimensional subspace of independent variables. Making it a constraint that the topological order in N-dimensional input sample space be preserved during mapping, Cherkassky and Lari-Najafi (1991) proposed the so called "Constrained Topological Mapping" (CTM) algorithms for global nonparametric regression problems. Correct topological order in the subspace of independent variables is sought by this algorithm through finding the best matching unit in the same subspace of independent variables $\boldsymbol{X}$. Correspondingly, the learning steps are modified by using the Euclidean distance in the subspace $\boldsymbol{X}$ of independent variables for the selection of the best matching unit but the weight updating is still performed in the input sample space $\boldsymbol{Z}$ (like in SOM algorithm), based on the following expressions.

$$\boldsymbol{W}_i(t+1) = \boldsymbol{W}_i(t) + \alpha[\boldsymbol{Z}(t)-\boldsymbol{W}_i(t)], \qquad \boldsymbol{W}_i^*(t) \in H_c(t) \tag{5.4}$$

$$\boldsymbol{W}_i(t+1) = \boldsymbol{W}_i(t), \qquad \boldsymbol{W}_i^*(t) \notin H_c(t)$$

where

$$\| \boldsymbol{X}(t) - \boldsymbol{W}_c^*(t) \| = \min_i \{ \| \boldsymbol{X}(t) - \boldsymbol{W}_i^*(t) \| \}. \tag{5.5}$$

$\boldsymbol{W}^*$ is an N-dimensional vector of the projection of $\boldsymbol{W}$ onto the $\boldsymbol{X}$ subspace. $H_c(t)$ is a symmetric neighbourhood subset of the $\boldsymbol{X}$ subspace with the centre vector being $\boldsymbol{W}_c^*(t)$. The other forms of adaptation and neighbourhood functions can also be used as in the

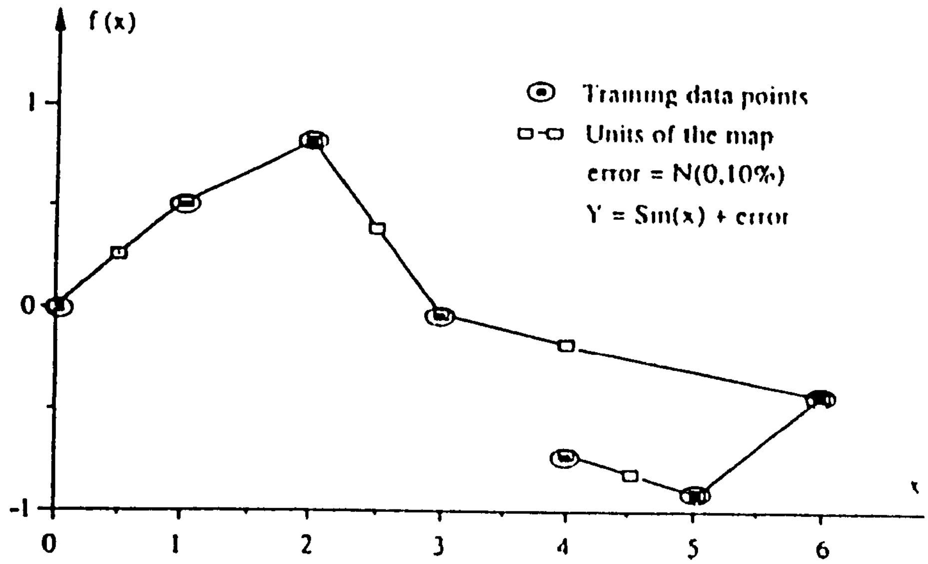


Fig. 5.1 Nonfunctional mapping formed by the original SOM algorithm.

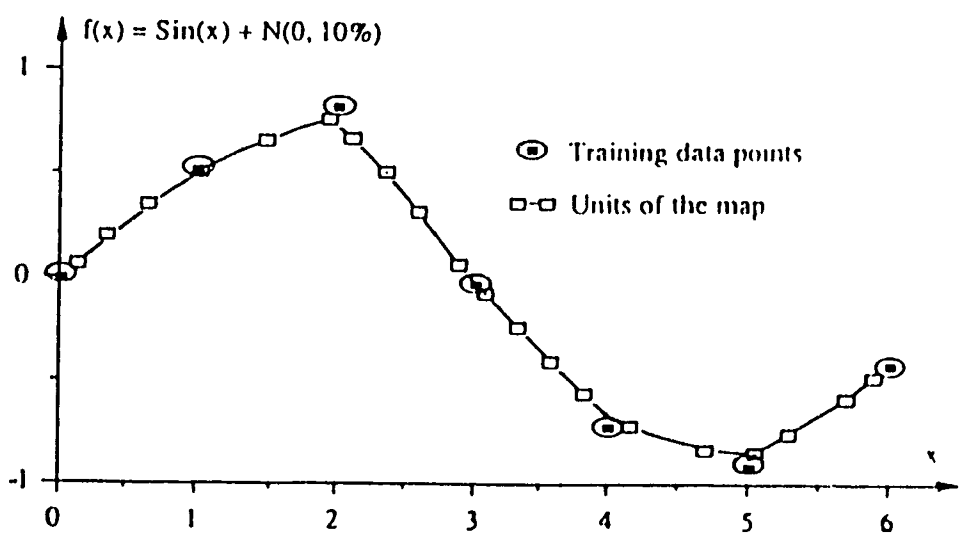


Fig. 5.2 Functional mapping formed by the CTM algorithm.

original SOM algorithm listed in the last chapter. Two different versions of the CTM algorithm have been proposed viz., Piecewise Linear Regression for Data Smoothing and Locally Constant Regression for High-Dimensional Problems. Hence, in order to find the best matching unit, the CTM algorithm selects a winning unit only according to the distance between the projections of a given sample and the units on the lower dimensional subspace of $\boldsymbol{X}$. The difference between the SOM and CTM in selecting the winning unit results in different outcomes while fitting functions of curves. Figure 5.2 (from Cherkassky and Lari-Najafi, 1991) shows the training result obtained using the CTM algorithm based on a 20-unit network to fit a sine function from the same data points as presented in Figure 5.1. The mapping can be seen to have no disordering problems. The CTM algorithm could be more successful than the original SOM algorithm, particularly when the independent variables have uniform distributions. However, if some of the N independent variables have non-uniform distributions, the CTM algorithm cannot avoid producing maps that have disordering of units. In fact, when fitting a high-dimensional curve, the CTM algorithm will perform the same operation in the N-dimensional subspace as the SOM algorithm does, to fit a curve in this N-dimensional subspace. If the original SOM fails in fitting the N-dimensional curve, the CTM will also fail in fitting the $(N+1)$-dimensional curve, especially when one or more independent variables have non-uniform distributions. For regression problems of trend analysis, wherein MMD, f is a multi-dimensional curve, application of both the original SOM and the CTM algorithms does not yield functional maps which do not violate (topological) order of the network units. Moreover, a map without disordering may or may not be a functional map for regression problems wherein f is a multi-dimensional curve, as can be seen from the following demonstrations. Both the SOM and CTM do produce nonfunctional maps as will be illustrated through simulation experiments in the next section. Detailed discussions of the reasons causing nonfunctional mapping problems are also included.

5.3 Nonfunctional Mapping in Regression Problems

The neurons of multi-dimensional maps move their spatial locations during the self-organizing process so as to represent the distribution of data points in the sample space. However, the way in which the spatial distribution of input data points is mapped onto a set of neurons through Kohonen's self-organizing process, may or may not be exactly what is expected for function regression analysis. A self-organizing map for regression analysis must satisfy some additional conditions. In self-organizing processes for function regression, in addition to responding to the distribution of the training data, the topological ordering of the neurons in a network should also reflect the ordered neighbourhood relations of input signals drawn from the input sample space. As a simple condition, it can be said that, if the values of the projection of given data points on an axis is monotonically increasing (or decreasing), the projection of the units onto the same axis should be increasing (or decreasing) in an orderly fashion. Self-organizing maps which are not such ordered maps, have been called nonfunctional maps. The following two simulation experiments have been performed to investigate the reasons behind nonfunctional maps.

Experiment 1

A single-variable function that represents a curve on the x-y plane in the following form is now considered.

$$f(x) = ax^2 + bx + c, \qquad 0 < x \le l \tag{5.6}$$

where a, b and c are constants, and l is the maximum value of x. Allowing for disturbances and noise, the above model becomes

$$y = f_d(x) = (a + e_1(x))x^2 + (b + e_2(x))x + c + e_3(x) + e_4 \tag{5.7}$$

where

$$e_1(x) = 0.2a\sin(2\pi x/20.0)$$

$$e_2(x) = 0.08b\sin(2\pi x/7.0)$$

$$e_3(x) = v\sin(2\pi x/4.0 + 0.3)$$

$$v = 0.1f(l)_{max} = 0.1f(l)$$

and e_4 is a Gaussian white noise, denoted by $\mathbf{N}(0, v)$, with zero mean and a standard deviation of v. In the above equation, $e_1(x)$, $e_2(x)$ and $e_3(x)$ are assumed forms of disturbance. The constants of Eqs. (5.6) and (5.7) are taken to be a=0.01, b=0.2 and c=80. The training data are uniformly sampled 4,480 times from the function $f_d(x)$ in the domain [0, 140] with a minimal sampling interval of 0.25, i.e. $Z_i=(X_i, Y_i)=(x_i, y_i)$, i=1, 2, ..., n, and n=4,480. Both the idealistic curve $f(x)$ without disturbance and the scatter-plot of the sample data points obtained from $f_d(x)$ are depicted in Figure 5.3. It can be seen that the projections of data points onto the x-axis are monotonically increasing and further, they are uniformly distributed. It is also evident in Figure 5.3 that the disturbance and noise are at a high level compared to the values of the function $f(x)$ itself.

A network of units $\{W_1, W_2, ..., W_k\}$, k=20, with $W_i=(X_i, Y_i)$ to represent the spatial location of the i-th knot in the x-y data sample plane, is established through the

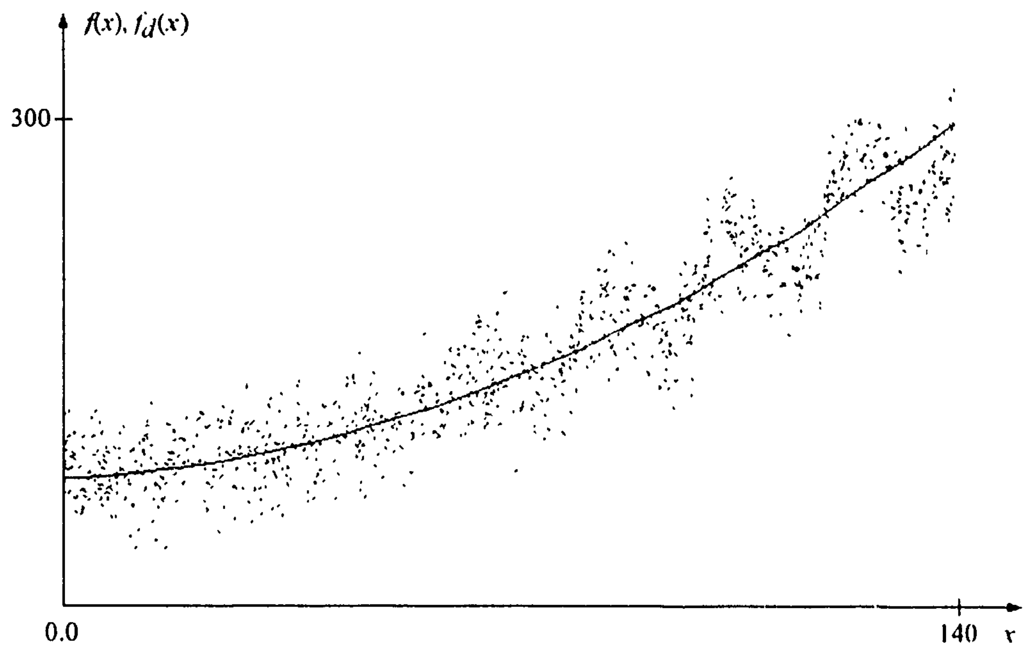


Fig. 5.3 The quadratic curve $f(x)$ and sampled data points $f_d(x)$.

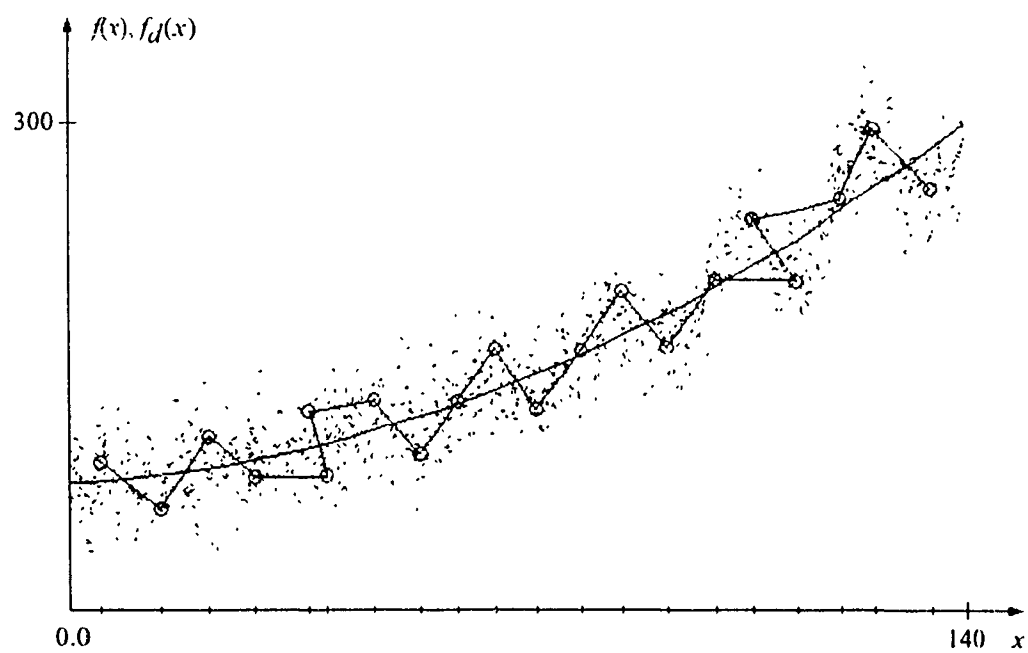


Fig. 5.4 A 20-unit network obtained using the SOM algorithm.

original SOM algorithm to estimate the underlying function from the data samples. The topological mapping is demonstrated in Figure 5.4, which shows that the order of the 20 units (shown by small circles linked by straight lines) measured along the x-axis has been violated. The knots are not in increasing order, and are not distributed uniformly along the x-axis as the distribution of the projections of the data samples on the x-axis should do. Another network with 8 units is trained by using the same algorithm, and the result is shown in Figure 5.5. All the 8 units are in the correct order. Obviously, there are some reasons that cause such discrepancies in regression mapping with different number of units. In order to establish the reasons behind the observed unit disorder, another network with 16 units is formed. The result is shown in Figure 5.6, wherein the order of units along the x-axis has not been violated. However, for two reasons, this regression curve is considered a nonfunctional map: (i) the projection of the unit positions onto the x-axis has not formed an uniform or close-to-uniform distribution, and (ii) the locations of the units close to the y-axis vary significantly around the curve $f(x)$, and hence, they do not properly represent the neighbourhood relations in the input sample space.

As mentioned earlier, the SOM and CTM algorithms differ in selecting the best matching unit to a given data sample. The SOM algorithm when followed for the selection of the winning unit may cause nonfunctional maps of the knots. The CTM algorithm leads to the selection of the best matching unit according to the distance measured only on the axis of the single variable x. This strategy forces, by the nature of self-organizing, the knots to have uniform or close-to-uniform distribution along the x-axis. During the learning process, since the projections onto the x-axis of both the data points and units have uniform distributions, a particular unit can only be matched to the closest samples lying between its immediate neighbours to the left and right along x-axis.

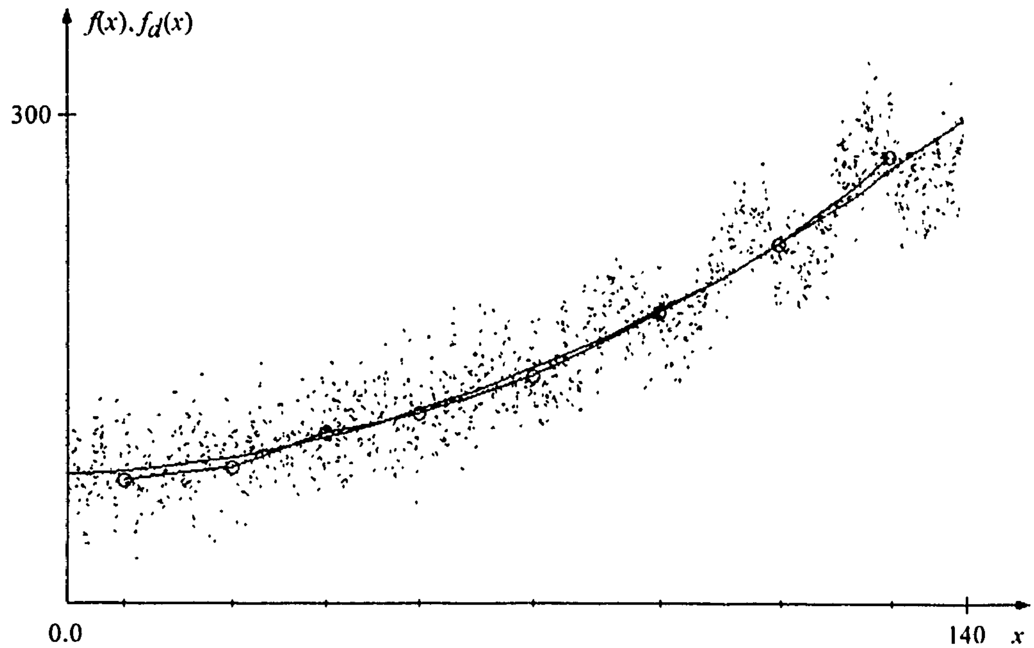


Fig. 5.5 Regression curve obtained using the SOM algorithm based on an 8-unit network.

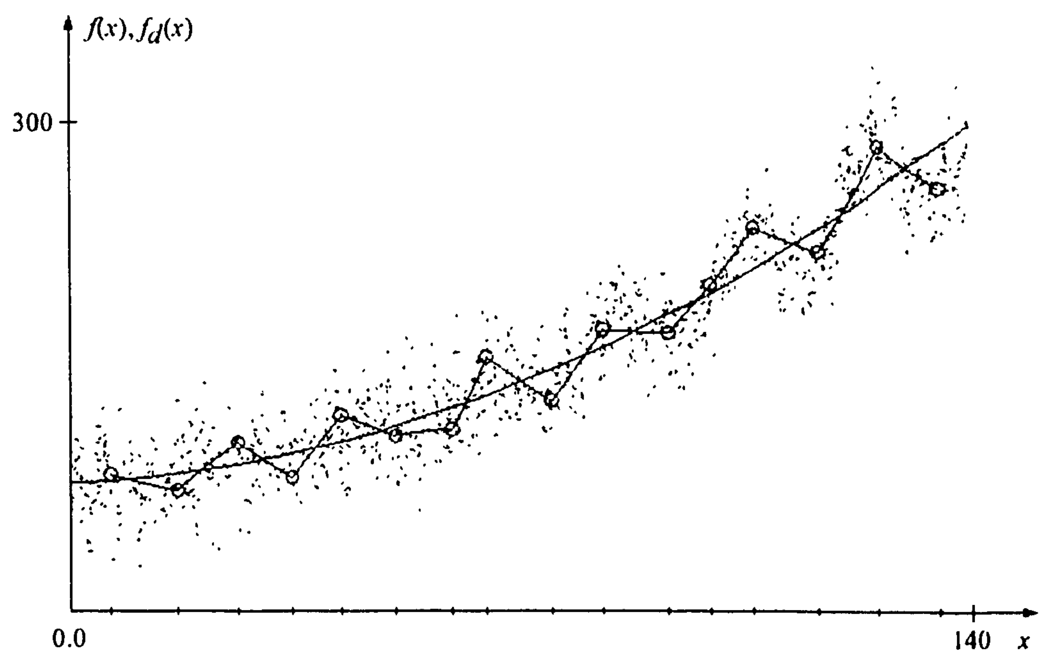


Fig. 5.6 Regression curve obtained using the SOM algorithm based on a 16-unit network.

Hence, such a selection of a winning unit will not be able to move any unit to cross its neighbours. And, the projection on the y-axis of the final position of a unit reflects the distribution of local samples around it. To be precise, one can draw two straight lines parallel to the y-axis. One is at the median between W_i and W_{i-1}, the other is at the median between W_i and W_{i+1}. The final position of W_i will be approximately located at the mass centre of the piece of data between these two lines. This capability of being able to "cluster" the data by itself is an important feature of the CTM algorithm. Figure 5.7 shows the result of mapping a network of 20 units using the CTM algorithm. The resulting curve can be considered as a proper map for the purpose of regression.

However, in certain cases, where the distribution of the values of one or more independent variables is non-uniform or/and with certain level of noise, the CTM learning algorithm could not avoid nonfunctional mapping. Consider the case of a two-variable function regression, where X and Y are two independent variables, and U is a function of them, that denotes a curve in three-dimensional space. Assume that the data vectors $\{X, Y, U\}^T$ are sampled from a machine system corresponding to discrete steps in X. This is the most common way of data collection. Assume further that a two dimensional curve is determined by the data vectors of $\{X, Y\}^T$ in the subspace, and assume also that the original SOM fails in fitting such a two-dimensional curve. While applying the CTM learning algorithm to this three-dimensional curve fitting problem, the winning unit has to be chosen based on the Euclidean distance on the X-Y plane. Since the projection of trained units on the X-Y plane in the CTM process is the same as that of SOM, the results become similar to that obtained using the SOM algorithm (to fit a curve on X-Y plane). Hence, the problem of disordering is still present with the CTM algorithm.

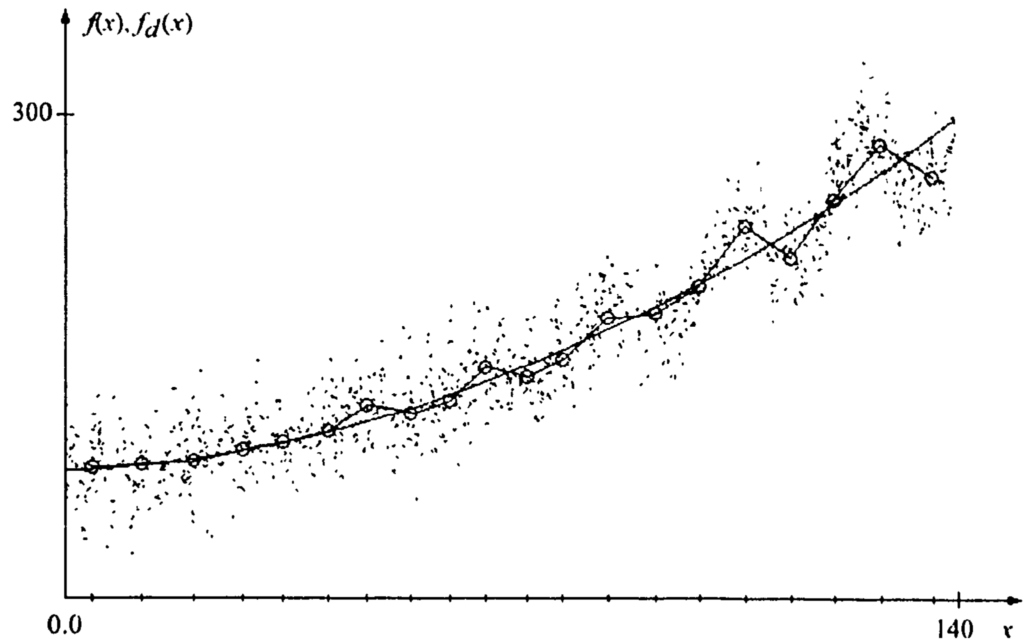


Fig. 5.7 Regression curve obtained using the CTM algorithm based on a 20-unit network.

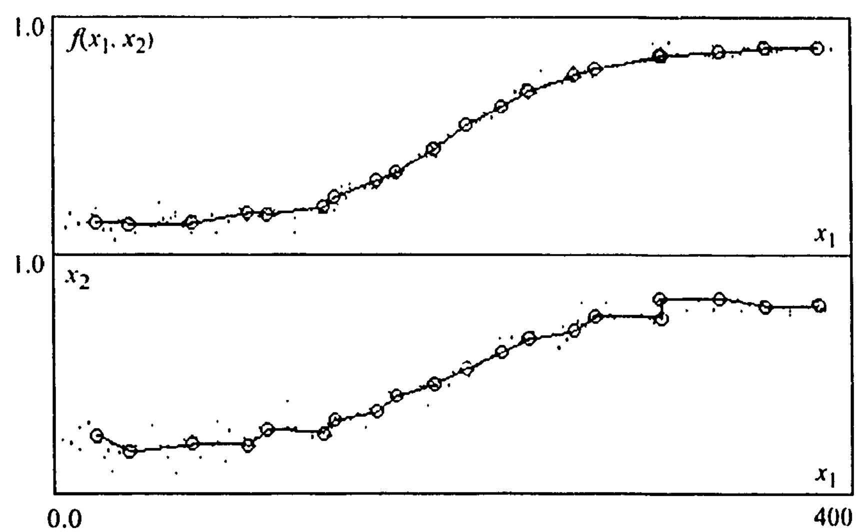


Fig. 5.8 Regression analysis involving an "S" curve using the CTM algorithm based on a 20-unit network.

Experiment 2

Another simulation experiment was performed to fit a high-dimensional "S" curve using self-organizing neural networks. In this simulation, the "S" curve is given by the sigmoid function

$$y = f_s(x) = \frac{1}{1 + e^{-\lambda x}}, \tag{5.8}$$

where x is an independent variable and, the constant λ controls the shape of the "S" curve. Such an "S" curve has been very useful in describing the cumulative fatigue damage process of many engineering materials.

A three-dimensional "S" curve in the sample space of $y=f(x_1, x_2)$ is given by

$$x_2 = \frac{0.6}{1.0 + e^{-0.03(x_1 - 0.5n)}} + 0.2 + e_1, \qquad 0<x_1<n \tag{5.9}$$

$$f(x_1, x_2) = \frac{0.75}{1.0 + e^{-0.03(x_1 - 0.5n)}} + 0.125 + e_2 \tag{5.10}$$

where n is the length of data samples, and e_1 and e_2 are Gaussian white noise defined by $N(0, v)$. The standard deviation v is expressed as a percentage of the function range. The training data was generated by Eqs. (5.9) and (5.10) with n=400, e_1=N(0, 25%) and e_2=N(0, 15%). The CTM algorithm has been tested in establishing a neural network with 20 neurons to fit the above three-dimensional curve. The initial locations of 20 neurons were set to be equidistant (uniform distribution) in the sample space. The data points and regression results are both shown in Figure 5.8, wherein the two plots show the projections of data points onto the x_1-y and x_1-x_2 planes. The projections of the final locations of the knots are shown with small circles linked by straight lines. From Figure 5.8 it can be seen that the order of some knots of the neural network has been violated and that some of the knots are too close to each other. Hence, the CTM algorithm has failed in solving this function regression problem.

As mentioned earlier, the original Kohonen algorithm performs poorly in fitting curves when the non-uniform distribution of a variable (e.g. y in Eq. (5.7)) has given undue influence to the selection of the best matching unit, even though another variable (e.g. x in Eq. (5.7)) has a uniform distribution. An explanation for the occurrence of disordering can be provided as follows:

1) The only difference among the three networks in Experiment 1, is that the number of units is changed. A lower number of units leads to (i) greater values of x (greater influence of x-values of data) in calculation of the Euclidean distance and vice versa, (ii) a trained network that will have a close-to-uniform distribution of units along the x-axis and vice versa.

2) From another viewpoint, the ratio of the band width of the values of y to the interval of x between two units is small, when a lower number of units is employed, and vice versa. The band width of y is large in the case of more noisy data.

3) The neurons are self-organised in a way to cluster the data points. The nonfunctional mapping problem is caused by the dislocation of units, and the worst case of disordering is also caused by this. When the noise level in the training data is high, the underlying function is complex, and the values of y give a greater influence than that of x through the Euclidean distance norm, it seems that nonfunctional mapping will result. This is the key reason for the occurrence of nonfunctional mapping.

It has also been observed in the present experiments, that disordering happens usually when two units are particularly close to each other so as to confuse the process of selecting the best match. Also, the level of noise gives a strong influence to the unit selection. This is due to the fact that, if there is a high level of noise e in Eq. (5.3), which in turn causes the y variation, the proper selection of the best matching unit through the Euclidean distance is hindered in the original SOM algorithm. On the other hand, the CTM algorithm selects the winning unit in the subspace of independent variables. When all the independent variables have uniform distributions, and this is the most common case especially in simulation experiments, the CTM algorithm organizes the units into uniform or close-to-uniform distributions, such that no two units would be too close to hinder the selection of the best match. Since there is normally no noise that is added to those monotonically increasing and uniformly distributed independent variables, CTM can guarantee the ordering of the units. It has already been shown that if there are one or more independent variables that have non-uniform distributions with a certain level of noise, CTM has the same problem as SOM does, in yielding nonfunctional maps. When the number of units is small, it is hard to observe disordering. When a large number of units is used, disordering occurs more often as has been observed in the simulation experiments.

5.4 A New SOM Algorithm Based on Weighted Euclidean Distances

Application of a self-organizing mapping algorithm to a multi-dimensional regression problem is successful only when the units of the SOM network represent correctly the distribution of data in the (N+1)-dimensional input sample space Z, and there is no disordering among the units of the network. Normally, a uniform or close-to-uniform distribution of the units on one or more dimensions is desirable since it serves the objectives of functional mapping. Further, certain properties of the sampled data in practical problems should also be considered. In real-world nonparametric regression problems, such as trend analysis in machinery monitoring and diagnostics, the data corresponding to independent variables are obtained from on-line measurements, and hence, they can have any kind of distribution. It is quite possible, particulary in the problems of condition monitoring and diagnosis of machine systems, that the distribution of at least one variable in the input sample space is uniform. In the most widely employed Discrete Sampling Method (DSM) of data collection, one or more variables (independent or dependent) are continuously increased (or decreased) and the corresponding values of remaining variables are measured. Depending upon the physical characteristics of the system being monitored, sampled values of the remaining variables could have any kind of distribution including uniform or close-to-uniform. For instance, if the functional dependency between the independent variable vector $\boldsymbol{X}$ and the scalar dependent variable Y is linear, uniform distribution of one variable would call for a similar uniform distribution of the remaining variables. This is the case for the linear segments between any two knots in a piecewise linear regression curve, while the overall distribution of data may still be undefined. Weakly non-linear functional dependencies would also result in close-to-uniform distributions. During the learning process, a uniformly distributed variable will force the projection of the units (of the self-organizing network) on the dimension of this variable to be uniformly placed. It can also be observed that an uniform distribution of a regression variable can effectively control both the selection of the best matching unit and the weights of the self-organizing units.

In the previous section, it has been observed that (i) the original SOM algorithm positions the knots (or units) so as to reflect the distribution of training data in the input sample space $\boldsymbol{Z}$, and (ii) the CTM algorithm positions the knots so as to reflect the distribution of projections of data onto the lower dimensional subspace of independent variables $\boldsymbol{X}$. Further, selection of the best matching unit in the SOM algorithm is dictated by the values of both $\boldsymbol{X}$ and Y, whereas in the CTM algorithm it is dictated by the values of the projections of $\boldsymbol{Z}$ onto the dimension corresponding to the independent variable X. It would be more appropriate and effective, if both the values of $\boldsymbol{X}$ and Y are considered in the selection of the best matching unit but with different weights. The relative values of the weights could then be determined so as to reflect the influence of the uniformly distributed variable in determining the distribution of knots (or units). This strategy will particularly be useful to the regression problems wherein (i) f is a multi-dimensional curve, (ii) the input sample data are collected from a machine system using DSM of data collection, and (iii) the problem is a global nonparametric regression using piecewise linear functions.

In a self-organizing mapping process, once the best matching unit to an input sample data point is selected, adjustment of the weights of both the unit and its neighbour units actually moves all of them in the same direction towards the given data point. At this juncture, the following two cases may arise: (i) the given data point is in between the best matching unit and its closest neighbours, (ii) the given data point is not in between the best matching unit (to be occasionally selected) and its closest neighbours. In the former case, no disordering of knots is possible and in the latter case a disordering may occur along at least one axis (along which the Euclidean distance measurements were taken). Hence, the proper selection of the best matching unit dictates the ordering of units (or knots). The above strategy essentially focuses on this point, i.e. to maintain the distributions of projections of units along one or more axes as close to uniform as possible during the entire learning process, and to have the proper selection of the best matching unit based on weighted Euclidean distances measured along both independent and dependent (regression) variables. The units (or knots) of the SOM network will not be subject to disordering and better represent the influence of uniformly distributed data points.

A greater weight is given in this new algorithm to the variable that has a uniform distribution. The weight factor is called the "scaling factor" and it is denoted by the symbol s, since it actually changes the scales of one or more dimensions in the Euclidean space. In selecting the winning unit, the distance between the sample data $\boldsymbol{Z}=\{X_{1Z}, X_{2Z}, \ldots\}^T$ and a unit $\boldsymbol{W}=\{X_{1W}, X_{2W}, \ldots\}^T$ will be calculated by

$$
\begin{aligned}
\|\boldsymbol{Z}-\boldsymbol{W}\| &= \sqrt{(X_{1Z}-X_{1W})^2 + \ldots + (sX_{iZ}-sX_{iW})^2 + \ldots} \\
&= \sqrt{(X_{1Z}-X_{1W})^2 + \ldots + s^2(X_{iZ}-X_{iW})^2 + \ldots}
\end{aligned}
\tag{5.11}
$$

where it is supposed that X_i is monotonically increasing (or decreasing) and has a uniform distribution, and so it has been greatly weighted by $s>1$. Therefore, variables other than X_i can affect the winning unit selection only when the value of $s^2(X_{iZ}-X_{iW})^2$ is relatively small. This strategy helps the self-organization process in producing functional maps not only along the dimension of the variable that has a uniform distribution but also in other dimensions. The influence of noise will also be reduced. When the number of units is small, this scaling factor can be selected as 1.0 and the present algorithm then becomes the same as the original SOM algorithm. When a large number of units is used on a multi-dimensional map, a proper value of s is easy to select for producing functional maps. The algorithm is now given in terms of computational steps.

Step 1. Locate all the k units, $\boldsymbol{W}_i=\{\boldsymbol{X}_i, Y_i\}^T=\{X_{i1}, \ldots, X_{iN}, Y_i\}^T$, $i=1, 2, \ldots, k$, randomly or at equal distances in the sample space $\boldsymbol{Z}$ as their initial positions, and set the value of the scaling factor to be $s>1.0$ (there could also be more than one scaling factor).

Step 2. Given a randomly chosen input training data vector $\boldsymbol{Z}(t)=\{\boldsymbol{X}(t), Y(t)\}^T=\{X_1(t), ..., X_M(t), Y(t)\}^T$ in the sample space, find the unit j which is closest to $\boldsymbol{Z}(t)$ according to

$$\|\boldsymbol{Z}(t)-\boldsymbol{W}_j(t)\| = \min_i\{\|\boldsymbol{Z}(t)-\boldsymbol{W}_i(t)\|\} \tag{5.12}$$

where t is the discrete iteration step. In Eq. (5.12), the distance between $\boldsymbol{Z}(t)$ and $\boldsymbol{W}(t)$ is determined by

$$\begin{aligned}\|\boldsymbol{Z}(t)-\boldsymbol{W}(t)\| &= \sqrt{(X_{1Z}(t)-X_{1W}(t))^2 + ... + (sX_{iZ}(t)-sX_{iW}(t))^2 + ...}\\ &= \sqrt{(X_{1Z}(t)-X_{1W}(t))^2 + ... + s^2(X_{iZ}(t)-X_{iW}(t))^2 + ...}\end{aligned} \tag{5.13}$$

Step 3. Define a symmetric neighbourhood of units i surrounding the winning unit j, $H_{ij}(t)$, and adjust the weights of the winner and all its neighbour units according to

$$\begin{aligned}\boldsymbol{W}_i(t+1) &= \boldsymbol{W}_i(t) + \alpha(H_{ij},t)[\boldsymbol{Z}(t)-\boldsymbol{W}_i(t)], \quad && i\in H_{ij}(t)\\ \boldsymbol{W}_i(t+1) &= \boldsymbol{W}_i(t), && i\notin H_{ij}(t)\end{aligned} \tag{5.14}$$

where $\alpha(H_{ij},t)$ is the scalar learning rate monotonically decreasing with t.

Step 4. Reduce the neighbourhood and the learning rate, increase iteration number t and return to step 1, until certain number of iterations is performed.

Step 5. The accuracy of the regression results is measured by the Average Residual (AR) (Cherkassky and Lari-Najafi, 1991), which is given by

$$AR = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[Y_i - f(X_i)]^2} \tag{5.15}$$

where n is the number of data points, and Y_i belongs to a training data pair $\{\boldsymbol{X}_i, Y_i\}^T$. In the above equation, $f(\boldsymbol{X}_i)$ is approximated by the value of Y coordinate of the most matching unit W_j to the input sample $X(t)$ according to

$$\| X(t) - W_j^*(t) \| = \min_i \{ \| X(t) - W_i^*(t) \| \} \tag{5.16}$$

where $W^*(t)$ is the projection of the weight vector of a unit onto the X subspace. This step can be combined with step 4 for training, so that whenever the proposed algorithm converges to a stable AR value, or the value of AR achieves the prescribed level of (function approximation) accuracy, the training process will terminate.

In practical applications of SOM, CTM or the new algorithm presented above, the learning rate for unit i in the neighbourhood of unit j at time t is given by

$$\alpha(H_{ij}, t) = b(t) \exp\left[\frac{-|i-j|}{(b(t)s_0)^2}\right] \tag{5.17}$$

where $b(t)$ is called the learning factor, the exponential term is the neighbourhood function, and s_0 is the number of knots. The learning factor for the winning unit is given by the empirical relationship

$$b(t) = b_0 (b_e / b_0)^{t/t_m} \tag{5.18}$$

where b_0 and b_e are the initial and final values of the learning rate, t is the discrete step of iteration, t=1, 2, ..., t_m, t_m is the maximum number of iterations which is usually defined as the product of the training set size and the number of times this set is recycled or repeatedly presented to the network during training.

Experimental Results

The same simulation experiments performed earlier have been repeated using the new algorithm. In fitting the data generated by Eqs. (5.6) and (5.7), the parameters used in the learning factor defined by Eq. (5.18) are selected as b_0=0.7, b_e=0.03 and the number of iterations is taken as $t_m=5n$=22,400 (the same values have been used in those previous experiments.) The scaling factor has been selected as 2.0. The result of this regression analysis with 20 units is shown in Figure 5.9. The locations of the 20 units can be seen to fit the data very well. Comparing Figure 5.9 and Figure 5.7, it can be seen that the new algorithm yields better results than the CTM algorithm.

The three-dimensional "S" curve is now fitted by a 20-unit network using the new algorithm. The training parameters used in Eq. (5.18) are taken to be b_0=0.5, b_e=0.02 and $t_m=15n$=6,000. The scaling factor s has been taken equal to 2.0. The result is shown in Figure 5.10 in the same manner as in Figure 5.8. This regression analysis is successful and does not exhibit a disordering problem. Further, the distances between any two neighbour units along the x_1-axis are almost equal.

5.5 Multiple-Index Based Trend Analysis of Bearings

The conditions of a bearing system are now identified using the new trend analysis approach with the newly developed algorithm presented in the previous section (Zhang et al. 1995b). Diagnostic indices are extracted from historical vibration signal. Vibration data were acquired from a type 308E ball-bearing having 8 rolling elements. The set up for testing the bearing and signal acquisition system have been described in Chapter 4. An accelerometer was mounted on the housing and its output was linked to a computer-based monitoring system. When an abnormal vibration signal was generated, the computer started to collect vibration data after each time period of 4.5 minutes, until complete failure of the bearing. A sampling frequency of 4,000 samples per second was used and the digital data stored in 105 files. Each file was divided into two time series thus

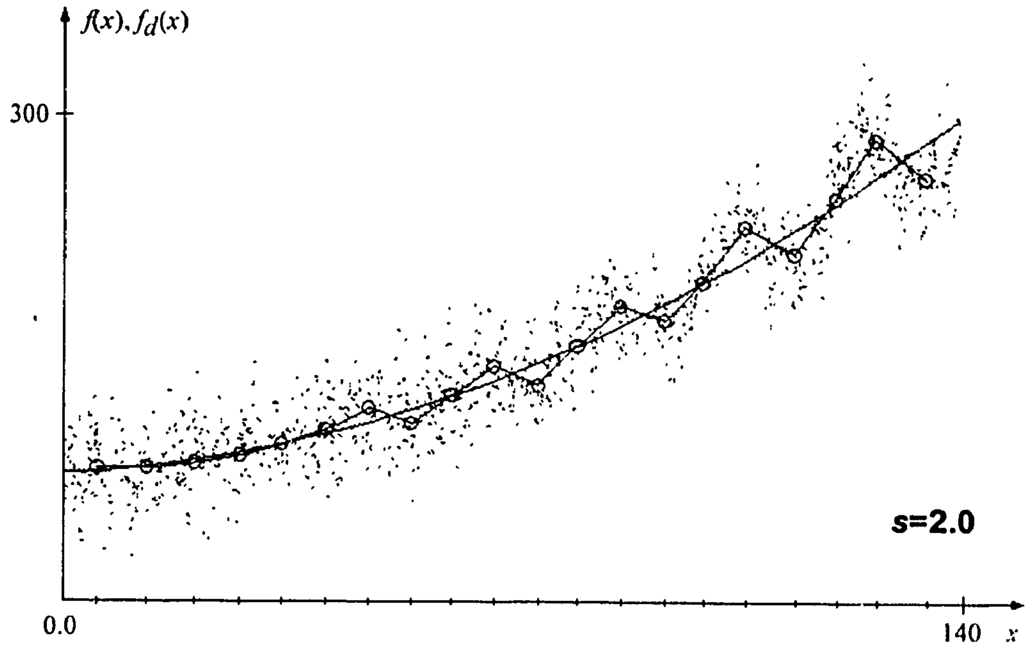


Fig. 5.9 Regression analysis of quadratic functions using the newly developed algorithm based on a 20-unit network.

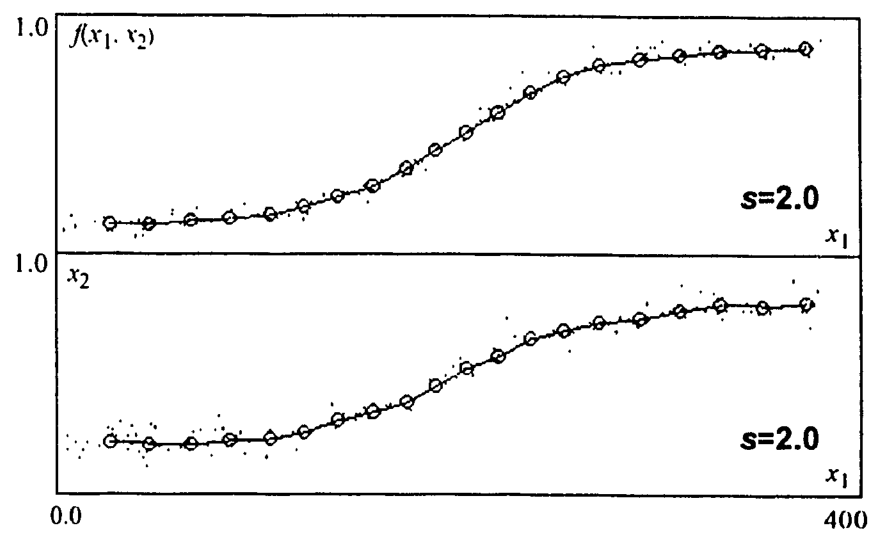


Fig. 5.10 Regression analysis involving an "S" curve using the newly developed algorithm based on a 20-unit network.

yielding 210 records. The malfunctions of the tested bearing were defects on the rolling elements and on the inner raceway. Four features being used in Chapter 4, the *PP*, *AX*, *CR*, and *AM* from the on-line bearing vibration signal have been chosen as diagnostic indices for trend analysis.

Trend analysis can be defined as finding a time *T*, that denotes the bearing life and is related to each and all of the four indices. Using the standard equation for a third order polynomial in *T*, and determining by the least square method the respective constants in these equations, the following were obtained:

$$PP = 19.013 + 0.143T + 8.603 \cdot 10^{-3} T^2 - 1.947 \cdot 10^{-6} T^3 \tag{5.19}$$

$$AX = 2.095 + 8.564 \cdot 10^{-3} T + 5.001 \cdot 10^{-4} T^2 + 2.676 \cdot 10^{-6} T^3 \tag{5.20}$$

$$CF = 3.4001 + 0.147T - 2.336 \cdot 10^{-3} T^2 + 9.468 \cdot 10^{-6} T^3 \tag{5.21}$$

$$AM = 104.307 + 0.178T - 3.696 \cdot 10^{-4} T^2 + 6.589 \cdot 10^{-7} T^3 \tag{5.22}$$

From the above four equations, the bearing life *T* can be expressed in terms of *PP*, *AX*, *CR* and *AM*. This way, a one-to-one correspondence between the bearing life and the diagnostic indices is observed. The data samples and corresponding curves of the above expressions are shown in Figure 5.11, wherein the horizontal axes of the four sub-plots denote time *T* and the vertical axes denote *PP* (upper-left), *AX* (upper-right), *CR* (lower-left) and *AM* (lower-right).

In multiple-index based trend analysis, the bearing life time *T* is determined based on the four diagnostic indices which are arranged in the form $X=\{PP, AX, CR, AM\}^T$. Since these four indices have been extracted from the same set of time series and further, since they can always be represented as four explicit functions of *T*, they actually represent a curve in a four-dimensional space. Each data point X_i is extracted from the data record corresponding to time T_i which is the bearing life time corresponding to the i-th data pair. The training data vectors have thus been defined as $Z_i=\{X_i, T_i\}^1=\{PP_i, AX_i, CR_i, AM_i, T_i\}^1$, $i=1, 2, ..., n$ with $n=210$, and this way the multivariable trend analysis essentially involves fitting a five-dimensional curve. Before training, the five variables have been scaled so as to lie approximately in the same range. The parameters of the equation for the learning factor, Eq. (5.18), have been taken to be $b_0=0.6$, $b_e=0.04$, $t_m=50n=10,500$. Considering the possible influence of the order of data presentation on the results of training, records of $\boldsymbol{Z}$ have been rearranged into a new random series prior to reusing them. The variable T has a uniform distribution over its range but the four indices are not distributed uniformly.

A neural network consisting of 8 units was first trained by the CTM algorithm. The training failed to generate a functional map, since the order of knots was violated. The resulting regression curve is shown in Figure 5.12, where the four sub-plots show its projection on to the four hyperplanes that are similar to those of Figure 5.11. The two dimensional figure plotted below the sub-plots shows the one-dimensional map of the units along T, which is the output of the training using CTM algorithm.

In using the new algorithm described in the previous section, a scaling factor has been used to multiply the uniformly distributed variable T. Different values of the scaling factor have been tested to estimate the functions corresponding to the number of knots ranging from 5 to 15. When the scaling factor is greater than 1.8, violation of the unit order has not been observed. A network with 10 units and a scaling factor of 2.0 is presented in Figures 5.13 and 5.14. Figure 5.13 shows the initial positions of the 10 units with random coordinates in the sample space. After training, each unit is moved to its

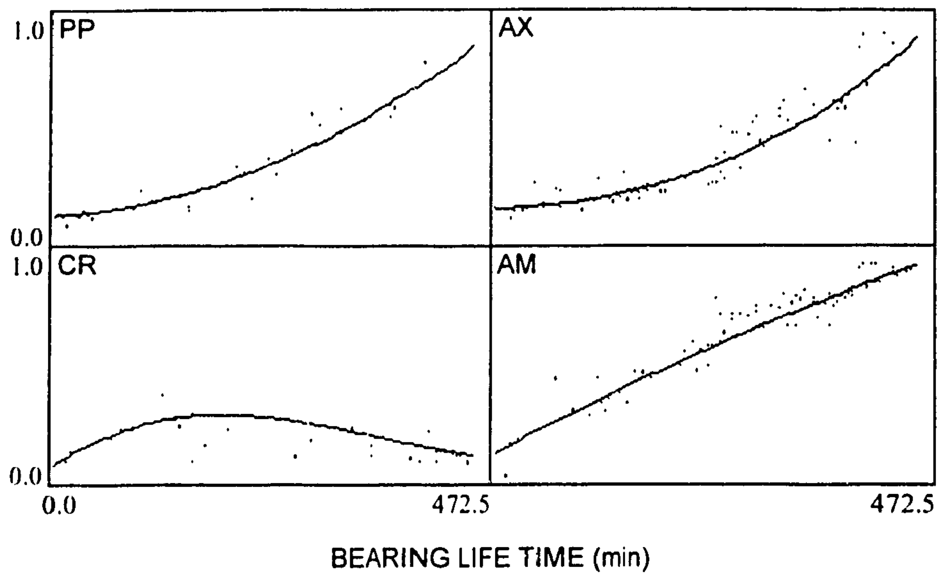


Fig. 5.11 The four univariate functions of bearing life obtained from the on-line vibration signal.

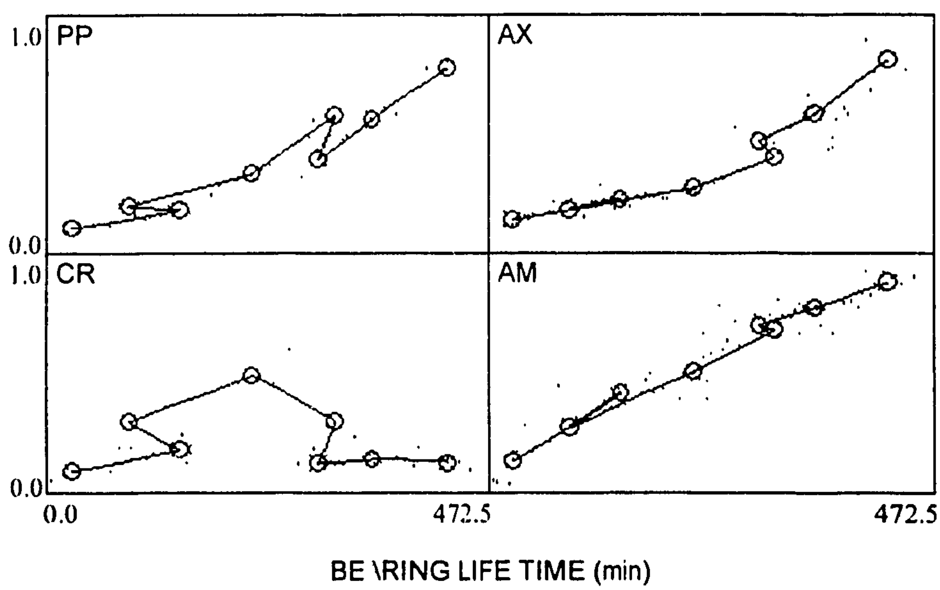


Fig. 5.12 Projections on to the hyperplanes of 8 knots of a nonfunctional map obtained using CTM algorithm.

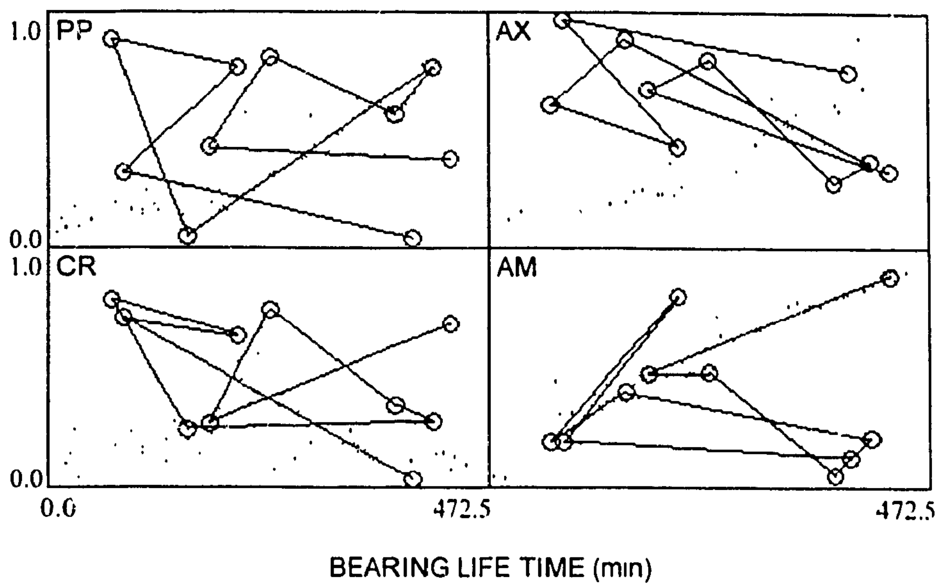


Fig. 5.13 Random initial positions of 10 knots of a neural network used in the new algorithm.

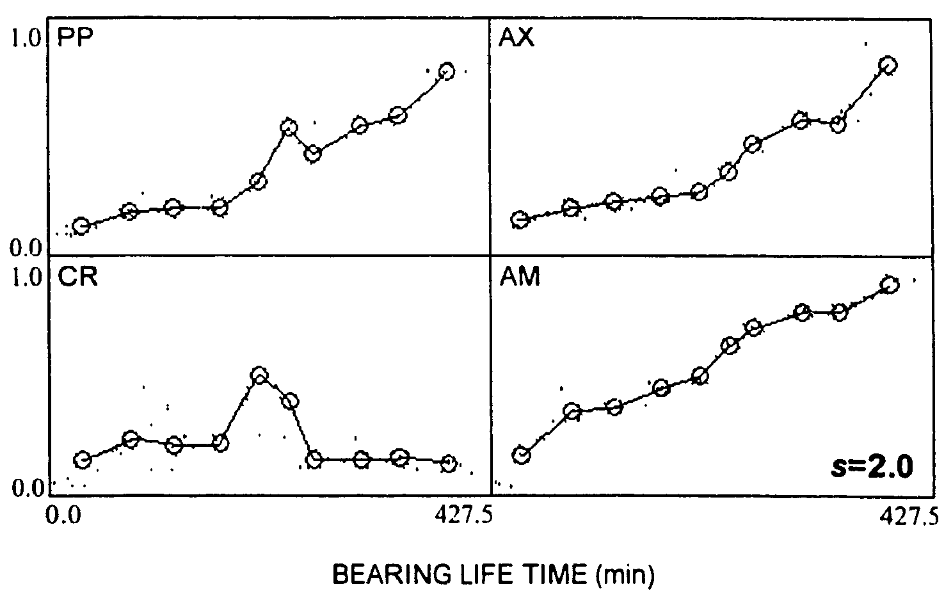


Fig. 5.14 Performance of the new algorithm in training a network of 10 knots with a scaling factor *s*=2.0.

final position in the $\boldsymbol{Z}$ space as shown in Figure 5.14. These ten units constitute the knots that connect the locally linear line segments, and the estimation of the unknown functions can now be obtained by piecewise linear methods. In our experiments, an easier approach has been used. That is, for any given index set $\boldsymbol{X}$ obtained from the same machine system, the corresponding bearing life T is simply approximated by the value T_j of the closest knot $\boldsymbol{W}_i$, which in turn is determined from the following equation

$$\| \boldsymbol{X}-\boldsymbol{W}_j^* \| = \min_i \{ \| \boldsymbol{X}-\boldsymbol{W}_i^* \| \}, \qquad 1 \le j \le k,\ i=1,\ 2,\ \ldots,\ k \tag{5.23}$$

in the same form as Eq. (5.16). The performance of the above network can now be measured by the average residual expressed as the percentage value, e_{1R}, which is the ratio between the average residual and the range of T. It is expressed as

$$e_{1R} = \frac{1}{T_a}\sqrt{\frac{1}{n}\sum_{i=1}^{n}[T_i-T_j]^2} \tag{5.24}$$

In the above equation, T_a represents the product of the number of data points and the sampling time interval, i.e. T_a=472.5 (minutes), which thus represents the entire monitoring time. Further, T_i is the life time corresponding to each data sample $\boldsymbol{X}_i$, i=1, 2, ... n, and T_j is the life time of the matched unit $\boldsymbol{w}_j$. This measure of error is in terms of standard deviation of the estimation error expressed as a percentage value. Another measure of error, e_{4I} is now defined as the average estimation error expressed as a percentage value:

$$e_{4I} = \frac{1}{T_a n}\sum_{i=1}^{n}|T_i-T_j| \tag{5.25}$$

where, T_a, T_i and T_j are the same as those defined in Eq. (5.24). The value of e_{4R} in the present case is 7.34% and that of e_{tt} is 5.23% for the 10-knots network established using a scaling factor of 2.0. They can now be compared with that obtained from single-variable functions, i.e. Eqs. (5.19)-(5.22): the values of e_{AR} and e_{tt} obtained from given values of PP are 20.41% and 17.82%, respectively. Corresponding values obtained from AX are 14.18% and 11.92% respectively, and from CR are 39.17% and 34.89% respectively. The best results obtained from AM have the values 9.92% and 7.57% respectively. For practical applications, the neural network established here consisting of 10 units is thus seen to be good enough for the purpose of estimating bearing life.

The number of knots is now increased and the performance of trained neural networks obtained using the present algorithm is compared with those obtained using both SOM and CTM algorithms, in Table 5.1. Two different values of scaling factor have been used. It can be seen from the accuracy of the results obtained using the present algorithm that even though the trend analysis is a high dimensional regression problem, wherein the variation of data is in large ranges, and the number of data points for learning is small, the method achieves reasonably good results.

5.6 Multiple-Index Based Trend Analysis of Gear Boxes

Application of multiple-index based trend analysis to the problem of life estimation of gear boxes is now considered. The on-line vibration signal was collected from a gear test machine during experiments with gear tooth surface damage due to fatigue (El-Karmalawy, 1993). Two identical spur gears with 30 teeth and a modal value of 6 have been used in the test apparatus. The rotating speed of the driving gear was 1,480 rpm, and so the gear tooth-meshing frequency is given by $f1$= rotating speed (rpm) x number of teeth / 60 = 740 Hz. An accelerometer was mounted on the gearbox housing, to pick

Table 5.1 Accuracy of bearing life estimation using different algorithms.

Number of knots	SOM algorithm (e_{AR}%, e_{AE}%)	CTM algorithm (e_{AR}%, e_{AE}%)	New algorithm with s=3.0 (e_{AR}%, e_{AE}%)	New algorithm with s=2.0 (e_{AR}%, e_{AE}%)
5	8.01, 6.26	8.23, 6.43	8.39, 6.55	8.08, 6.33
6	7.52, 5.75	7.51, 5.74	7.60, 5.42	7.59, 5.54
7	7.43, 5.55	7.47, 5.68	7.70, 5.44	7.58, 5.51
8	*	*	7.60, 5.38	7.35, 5.38
9	7.49, 5.49	7.46, 5.68	7.59, 5.28	7.49, 5.36
10	7.36, 5.37	7.31, 5.52	7.60, 5.21	7.34, 5.23
11	7.30, 5.42	*	7.53, 5.12	7.38, 5.12
12	*	*	7.44, 5.06	7.30, 5.15
13	*	*	7.48, 5.09	7.24, 5.11
14	*	*	7.33, 4.98	7.11, 5.09
15	*	*	7.24, 5.08	7.21, 5.10

* The network knots were disordered.

up vibrations in the horizontal direction. There were only nine digitized data records of the vibration acceleration signal obtained from the test. They were recorded one after another at approximately the same time interval. The sampling frequency was 6,400 Hz and further, 2,048 data points were stored in each record. Record 1 was obtained at the beginning of the test just after the gears started wearing in. Records 2 and 3 were collected during normal operating conditions. Records 4 and 5 correspond to gears that had very small pits introduced on a small number of teeth located near the pitch line. Records 6 and 7 were obtained from a condition when a large number of pits was distributed on the gear teeth. Record 8 was obtained when the gears had large pits where the biggest one was 1.25x2.5 mm^2. The last record, number 9, was obtained when the gears had practically failed.

Trend analysis has been conducted in order to approximate the unknown function that describes the whole process of the development of malfunctions (the process of damage), based on incomplete information provided by the nine records. Three indices have been extracted from the vibration acceleration signal, that include the *RMS* value of the time domain signal, magnitudes at gear tooth meshing frequency (*Mf1*) of 740 Hz and its second harmonic (*Mf2*) of 1,480 Hz. Previous research (El-Karmalawy, 1993) has shown that the *RMS* values are sensitive to changes in gear condition. The values of *Mf1* and *Mf2* are sensitive to the development of the defects on the surface of gear teeth. Using the variable T to denote the damage process, the training sample has been defined as being the set $\mathbf{Z}_i=[RMS_i,\ Mf1_i,\ Mf2_i,\ T_i]^T$, $i=1, \ldots, 9$. The training parameters of Eq. (5.18) have been taken to be $b_0=0.7$, $b_e=0.03$, $t_m=$ 3,600. And, the scaling factor applied to the variable T has been taken as 1.4. A one-dimensional map with 50 knots has been established through the previously described new algorithm. Figure 5.15 shows

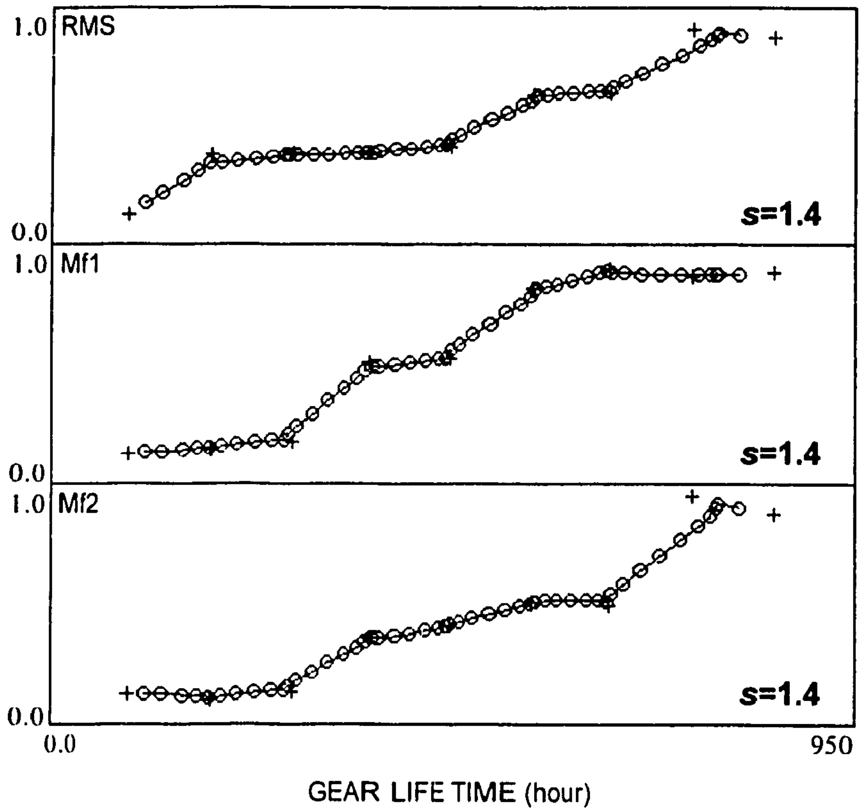


Fig. 5.15 Network of 50 knots trained by the new algorithm to fit 9 data points. (s=1.4)

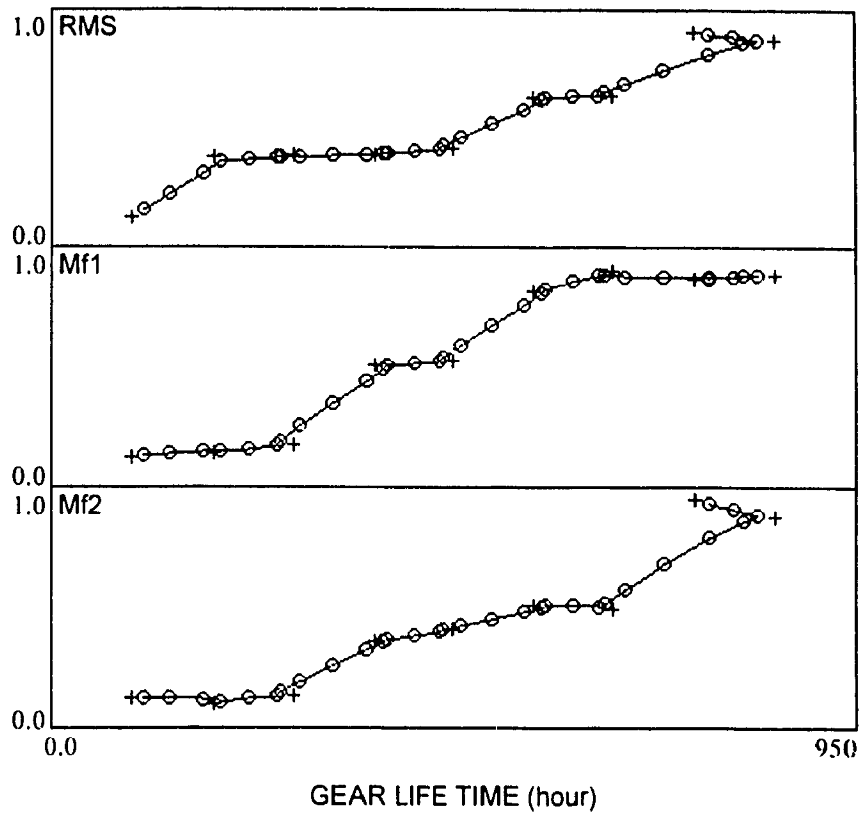


Fig. 5.16 Disordered 30-knots network trained by CTM algorithm.

how well the algorithm would allocate extra units. The projections of the damage function obtained using the present algorithm onto the hyperplanes determined by the three diagnostic indices of the gearbox are shown in Figure 5.15. It can be seen from the above figure that the new mapping algorithm has the ability to find the optimal locations of units, even though the number of knots is much more than the number of data points. A neural network with 30 knots has been attempted using the CTM algorithm for training but the network failed in that it produced a nonfunctional map as shown in Figure 5.16.

5.7 Discussion

A new approach to the condition monitoring and diagnosis of machine systems is proposed. In this approach, the multiple-index based trend analysis is used as a basis to (i) obtain information on the current state of the machine and to (ii) estimate with prescribed accuracy the service life of the machine system being monitored. On-line vibration measurements are used as the diagnostic signal and the variables for trend analysis are obtained from the signal through the indices that are extracted from the time series of the signal. The diagnostic indices are selected in such a way that each index is distinctly sensitive to a particular type of fault or malfunction in the machine system. Alternately, each and all of the commonly-observed faults or malfunctions in the machine component being monitored, are adequately reflected in the variables used in the trend analysis. That the proposed method produces more accurate results than the widely used single-index based trend analysis is demonstrated considering the case of a bearing system. The proposed new approach using multiple variable trend analysis has the following important advantages:

1) No prior knowledge of the unknown model of multivariate function is required.

2) More than one characteristic of the on-line vibration signal can be simultaneously considered.

3) Both large or small training data sets can be handled adequately.

The new neural network algorithm, is adapted to perform the above trend analysis. The unequal scaling of variables is central to the new neural algorithm which was originally proposed to improve the performance of Kohonen's self-organizing mapping algorithm for regression problems. This strategy is particulary useful to the regression analysis using piecewise linear functions, wherein the fitting of a multi-dimensional curve is involved. Disordering of knots or units is avoided through the proper selection of the best matching unit and its closest neighbours. Both independent and dependent variables are considered for the selection of the best matching unit but with different weights and further, the component of the input data vector that has a uniform distribution, whether it corresponds to independent or dependent variable, is assigned a greater value of weight. The natural ordering of units in both the projection subspace of independent variables and the dependent variables is preserved in this new SOM strategy. A greater emphasis to preserve the ordering of units along the component of input data vector that has a uniform distribution is embedded through assigning a greater value of weight to that component. The algorithm development has been oriented so as to avoid the nonfunctional maps obtained by both the CTM and SOM algorithms. Structuring a self-organizing neural network using the new algorithm for desired applications is easy and further, the training process is fast in that it takes only few seconds on a personal computer. It is shown that both the CTM and SOM algorithms, which are currently being used for function regression analysis, are not efficient and accurate for use in machinery monitoring problems. To demonstrate this, application of the original SOM and the CTM algorithms to the trend analysis of a bearing system and a gearbox is conducted. It is clearly shown that the topological order of units is violated in both cases thus leading to erroneous and unreliable results. The new neural network algorithm is shown to be superior to both the SOM and CTM algorithms in that it

1) does not violate the topological order of units;

2) does not produce nonfunctional maps when real-life data from practical machine systems are considered;

3) has the ability to optimally locate all the units even when the number of units is significantly large;

4) can estimate the multivariate curve from finite data points in addition to estimating the multi-dimensional surfaces; and

5) consumes less computational time but yields highly accurate estimation of multivariate functions such as the service life of practical machine systems. Thus, the present chapter contains both analytical and algorithmic developments for use in condition monitoring and diagnosis of industrial machine systems.

The scaling factor of the new algorithm has been shown to have a direct bearing on the order and locations of the units or knots, thus on the accuracy with which a regression curve is obtained. When the value of this scaling factor reaches a certain threshold, violation of unit order ceases to be present in the established map. When the number of data points is relatively large, increasing the value of the scaling factor beyond this critical value makes the knots possess a more uniform distribution along the weighted axes in the input sample space. However, equally spaced knots do not always yield an optimal regression curve, since the corresponding value of average residual is more than that of the network with unequally spaced units. In other words, the average error of uniformly distributed knots is smaller. From the simulation results, it is felt that the values of both average residual and average error simultaneously tend towards their corresponding minimum values when the scaling factor reaches its critical value. Increasing the values of the scaling factor beyond the critical value however, does not guarantee this trend.

CHAPTER 6

IMPLEMENTATION OF RMD-KBS

The structural implementation of the RMD-KBS (Rotating Machinery Diagnostic Knowledge-Based System) prototype system is detailed in this chapter. The KBS development tool that has been utilized in this application is briefly introduced in Section 6.2. The knowledge acquisition is described in Section 6.3. An outline of the system structure is given in Section 6.4 and this is followed by descriptions of all components and modules of the RMD-KBS in Sections 6.5 to 6.10. The coupling of numerical and symbolic processing, the hybrid knowledge representation and manipulation, the inter-task connection between the database and the knowledge bases, and the inter-process communication between the RMD-KBS and the external programs are all discussed in detail, as is the multiple windowing user interface.

6.1 Introduction

In Chapter 3, the conceptual design of a new knowledge-based diagnostic system for rotating machinery monitoring and diagnosis was described. Formulation of the fault detection problem into a neural network based pattern classification problem was discussed in Chapter 4. The multiple-index based trend analysis and its neural network formulation, was developed in Chapter 5 for use in the diagnostic system. The feasibility of the new conceptual design is the prime concern of this chapter. The conceptual design is realized and the prototype knowledge-based system is fully developed. The implementation details and the features of the knowledge-based diagnostic system are both provided in this chapter.

At this point, it is worthwhile to outline the major features and limitations, which have been considered in the development of the prototype. As shown in Chapter 3, the most remarkable characteristics of the design are

1) it is a hybrid system which employs both connectionist (neural networks) and rule-based reasoning approaches for knowledge representation and knowledge-based diagnosis;

2) it is a deeply-coupled numerical and symbolic processing system, which performs machine fault diagnosis based on vibration signal analysis; and

3) it has a certain level of learning capability, as a consequence of the neural networks employed in the system.

The design of the RMD-KBS orients itself towards industrial applications. Many important aspects regarding industrial applications have been taken into consideration. For instance, the RMD-KBS can itself perform suitable signal processing operations for feature extraction. Compared with existing KBSs for rotating machinery diagnosis, the RMD-KBS is highly automated. The user interface of RMD-KBS is well constructed to provide a convenient working environment under a simple operating procedure.

However, the development of a diagnostic KBS for real-world applications involves too much work that can not be performed by a single person in a limited time. As a prototype, the RMD-KBS is not considered to be a complete system. Both the diagnostic knowledge encoded in it and the functions it can perform now, are less than desired. The approach and methodology developed in this thesis have been implemented in RMD-KBS only at a level needed to demonstrate the feasibility of the design approach.

The RMD-KBS has been assembled using the KBS development tool LEVEL5 OBJECT™. The structure and capability of this tool, as well as its object-oriented system, are briefly introduced in Section 6.2. The knowledge source and acquisition are delineated in Section 6.3, before describing the details of the structural implementation of the RMD-KBS.

6.2 The KBS Development Tool and Its Object-Oriented System

The LEVEL5 OBJECT™ (Level5, 1990a; 1990b) is a KBS development tool available for the implementation of rule-based systems. It combines the versatility of object-oriented techniques with multiple interfacing strategies in a flexible windowing environment. The LEVEL5 system is composed of an inference engine and a knowledge editor. The inference engine supports both forward and backward chaining, and also mixed-mode (forward and backward) processing. It works with any knowledge base which is developed using its knowledge editor. The knowledge editor is a multiple windows environment (see Appendix A.2), that includes the utilities named Object Editor, Rule/Demon/Method Editor, Display Editor, Agenda Editor, Windows Editor, Database Interfaces, Graphical Display of Knowledge Tree, Session Monitor/Debugger, Values Report, History Record and External Program Interfaces. These utilities support the establishment of the knowledge base.

In a knowledge base, the information and knowledge can generally be represented using different schemes such as "objects", "agenda", IF-THEN-ELSE "rules"/"demons", "methods", grouped rules, and relational database models. The LEVEL5 objects encapsulate the knowledge structure, specific values, and procedures for processing the data. They are created using class declarations. The agenda schedules the events that the

LEVEL5 application will follow, or the hypotheses that the inference engine will pursue during backward chaining. The rules and demons are the same in the statement format, but rules are evaluated by the backward chaining during inference, whereas demons are fired by the forward chaining. A method consists of a group of statements for knowledge-based reasoning, that can perform a more complex task than a rule or demon. Two types of methods, named WHEN CHANGED and WHEN NEEDED, that are fired by forward and backward chaining during inference, can be defined using LEVEL5. These knowledge representation schemes have been described at length in (Level5 1990a) and the details of their usage have been documented in (Level5, 1990b). Further, associated with the rules or demons are the confidence factors (CFs) which are described in detail in Appendix A.2. The relational database models interface with external database files, for information exchange between a particular database and the other components of the KBS. Moreover, a KBS developer can design his/her own user interfaces in Microsoft® Windows™ format using LEVEL5. External computer programs, which are designed to perform specific numerical or symbolic computational tasks, can be invoked by the inference engine of LEVEL5. Further details about the utilities of LEVEL5 will be introduced in this chapter as and when required.

The RMD-KBS employs the object-oriented technique and the LEVEL5 system supports the implementation. The objects provide a template for classifying and organizing knowledge within a LEVEL5 application. A class, that is to be declared in order to create an object, is defined by a collection of characteristics called attributes, that define the structure of an object. The structure of a LEVEL5 object is outlined in Figure 6.1 (Level5, 1990a). It can be seen from Figure 6.1 that, a class possesses a name, certain assignable properties, several attributes, and at least one instance. The properties determine

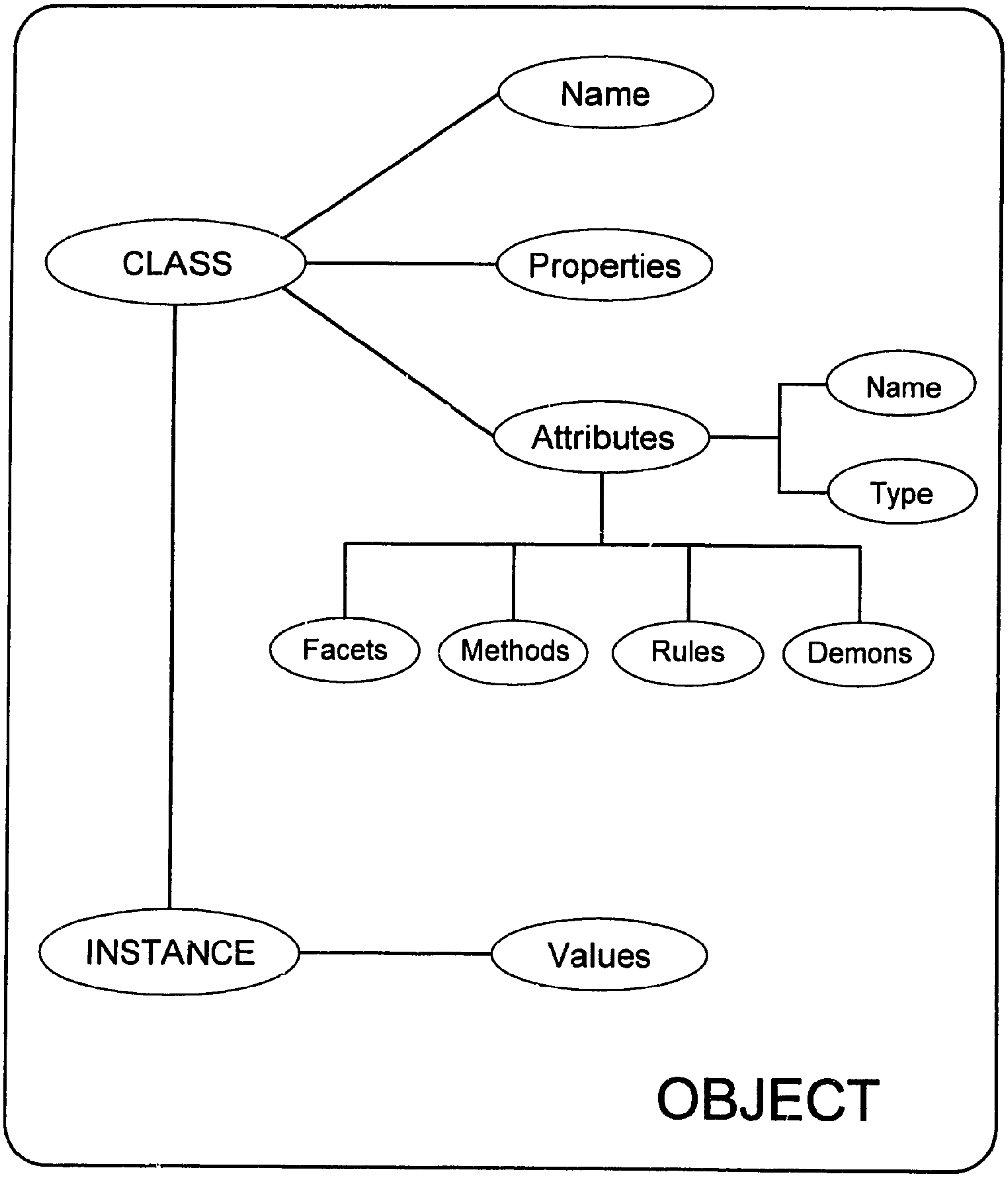


Fig. 6.1 Outline of the structure of a LEVEL5 object.

if the class can inherit the structure of another class, if the class is created from an external database file, or if the class is limited to a single instance. An attribute of the class has its type, and it can be used in the statements of the knowledge base as a symbolic element or variable. Facets, rules, demons and methods refer to and influence the attributes. For example, when the value of an attribute changes during a reasoning session, the rule that has previously been defined for this attribute, will be activated (i.e. fired). A class defines only the structure of an object. In order to carry the values of the attributes, an occurrence of the class called instance must be created. More than one instance can be created by a KBS developer from the same class. A KBS developer can use the Object Editor of LEVEL5 (see Figure A.2.2) to create classes and their instances, as well as to specify the properties of the class, and to select the type of each attribute. Moreover, LEVEL5 provides a number of system-defined classes such as **application**, **picturebox**, **promptbox**, **pushbutton**, **textbox**, **timer**, and **true-false box** (see Level5, 1990a), for KBS development. Typical examples of classes will be given in the following sections.

Among the family of currently-available KBS development tools, the LEVEL5 is a medium-sized and a fairly sophisticated system. Compared with other expert system shells such as NEXPERT OBJECT™ (TATA, 1989), the LEVEL5 is not a very powerful tool. However, this tool has been adequate for the implementation of RMD-KBS, since it provides all the utilities needed to support the development of the present diagnostic KBS. Nevertheless, significant amount of additional programming work had to be performed to fruitfully exploit the capabilities of LEVEL5 system.

6.3 Knowledge Source and Acquisition

The knowledge embedded in RMD-KBS has mainly been acquired from published literature on rotating machinery monitoring and diagnosis, and fault detection based on vibration signal characteristics. Diagnostics of rotating machinery employing the analysis of vibration signal has been well documented in the technical literature. The diagnostic knowledge to be implemented in the RMD-KBS is obtained from the literature. Two important articles (Sohre, 1980; Esheleman and Jackson, 1992) written by well known experts in this domain, have been used as the primary references. It may be noted here that the former article, (Sohre, 1980) has formed the knowledge source for several existing advanced KBSs, e.g. (Stuart and Vinson, 1985; Keim and Nordmann, 1989; Kato et al, 1990). Detailed information about the algorithms that are used in both vibration signal analysis and machine fault diagnosis has been acquired from relevant text books and monographs, including (Collacott, 1979; Shives and Mertaugh, 1986; Rao and Griffiths, 1990; Lipovszky et al, 1990; and Chen and Li, 1991). The illustrations and case studies, that have been documented therein have been taken into account. In addition, a large number of journal and conference publications have also been reviewed. These include the works by Sasaki and Tomita (1984), Cempel (1988), Boyce et al. (1989), Brawley et al. (1989b), El-Shakweer (1989), McFadden (1989), Zhuge et al. (1990), Saavedra et al. (1990), Grace (1990), Martin et al. (1990), Kadushin 1991), Tang et al. (1991), Leung (1992), Mechefske and Mathew (1992), Yan and Shimogo (1992).

The author has long worked on machinery monitoring and diagnosis of real-life industrial machinery systems and hence accumulated experiential knowledge. This experiential knowledge has served as a basis in sorting out the information provided in the published literature. Further, real-life vibration data have been obtained from machines currently in use and have been carefully analyzed in this investigation before encoding the relevant information. The detailed information about the machines monitored, the setup of signal measurement devices, the vibration data, the detected machine faults based on this vibration data, and the diagnostic method appropriate to each and all of the cases are available in the work of (El-Karmalawy, 1993).

The knowledge acquisition work involved in the present development of RMD-KBS is performed through the sequential investigation approach described below:

Step 1. Make a list of the faults and malfunctions in a rotating machinery system, that have been well defined, frequently observed and comprehensively studied in previous MMD research.

Step 2. Investigate for (i) the relationship between the faults and the machine type or structure, and (ii) the relationship between a fault and the operational conditions of a machine, if they are given in the literature.

Step 3. For each machine fault or malfunction, find the most appropriate diagnostic methods and indices, that have been proven or recommended in the published literature as the best ones for the detection of the fault.

Step 4. Sort out the relationship between a machine fault and the diagnostic indices used in detecting it, and describe the relationship qualitatively. The previous knowledge and experience about fault-symptom relationships in quantitative mode, that provide important reference values to be used in determining the numerical thresholds of the indices, and in the training of neural networks, are also obtained

Step 5. Corresponding to each diagnostic index, find the relevant equation or algorithm. Obtain the information of the sensors, the corresponding settings involved in the signal measurement and processing, if available.

Step 6. The raw information and knowledge can be gathered through the above five steps. Reorganize the accumulated material into different types of information and knowledge as defined in Chapter 3. In this step of knowledge acquisition, significant analysis tasks are involved, while determining the way in which the information and knowledge are used in the RMD-KBS.

The knowledge acquisition has been mainly performed at the early stage of this research work, and has continued throughout the development of the RMD-KBS, as and when necessary.

6.4 Outline of the RMD-KBS

The structure of RMD-KBS is shown in Figure 6.2. As an on-line monitoring and diagnostic KBS, it is designed to be an integrated system of eight components: the sensors, a data acquisition system, a database (DB), numerical analysis routines (NAR), a multiple windowing user interface (UI), an inference engine (IE), knowledge bases (KBs), and a knowledge editor (KE). These components are now described in the sequel.

The sensors and data acquisition system, including signal pre-preprocessing hardware (signal amplifiers, filters etc), can be linked to RMD-KBS. These are actually intended to pick up vibration signals form the machines being monitored, to digitize the measured analogue signal, and further, to input the obtained digital data samples to the RMD-KBS. They are designed so as to be controllable by the RMD-KBS, e.g. when they perform their tasks, the settings of the sampling conditions are sought from the KBS. The functions performed by physical sensory and data acquisition systems, are simulated through the use of off-line files of vibration data samples.

In the RMD-KBS, several types of information and vibration data are stored in

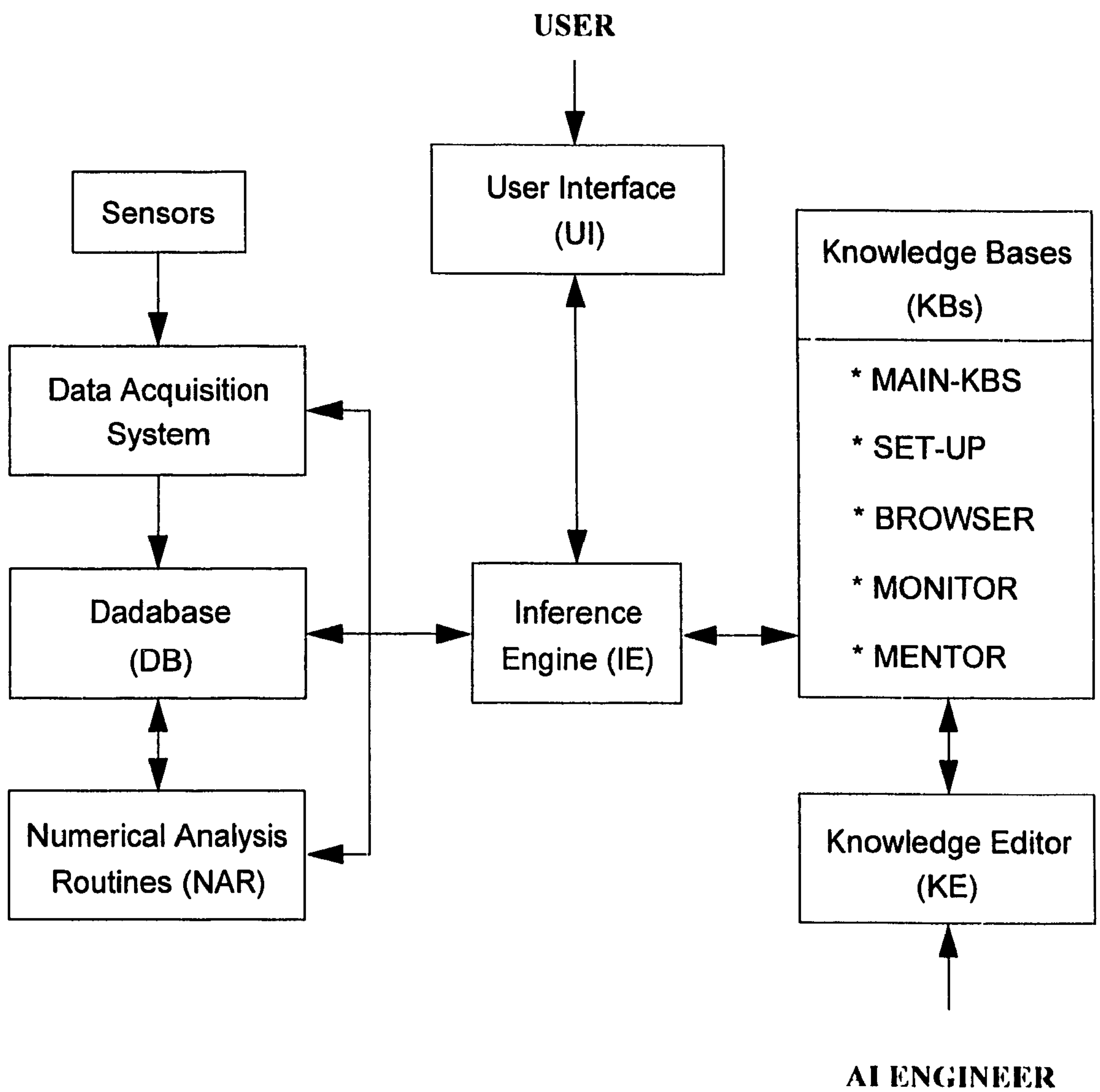


Fig. 6.2 The structure of RMD-KBS.

more than 200 database files that are managed by the DB. LEVEL5 has a database interface for communicating dBASE III® files through a special type of objects, known as database classes. Using the object-oriented programming (OOP) techniques in RMD-KBS, as mentioned in Chapter 3, the large amount of information (facts) required for the diagnosis is stored in the database, instead of being directly described through the symbolic statements in its KBs. This facilitates the acquisition and refinement of the factual knowledge, through database access rather than knowledge base modification.

A number of algorithms and their corresponding computer programs, for numerical data processing, realization of neural networks, and graphic displays have been developed to perform the tasks of vibration signal analysis, diagnostic index extraction, neural network based diagnosis, and to display the vibration data, the spectrum and other relevant plots on the screen. They have been encoded in C++ language as Microsoft® Windows™ applications (i.e. the programs can run under the Windows environment). The C++ routines are external programs to the KBS and they are invoked by the inference engine.

The multiple windowing user interface developed in this research is in Microsoft® Windows™ format, and performs the man-machine communication at the run-time of the system. The utilities of the LEVEL5 knowledge editor have been employed in the development of the user interface. The behaviour of user interface elements is controlled by rules, demons and methods. In order to enhance the user interface, several windows in a more complex style for graphical displays have been developed in C++.

The inference engine that is provided by the LEVEL5 OBJECT, supports three inference strategies, i.e. the forward chaining, backward chaining and mixed-mode chaining schemes. The selection of an inference scheme, however, depends on the tasks to be performed by a KBS using a particular knowledge base. For RMD-KBS, all the three inference schemes have been employed, at different stages of performing the monitoring and diagnosis tasks.

The knowledge editor embedded in the LEVEL5 system, has been adopted. The implementation of the symbolic processing modules has been performed using the utilities of the knowledge editor. It is an integral component of the RMD-KBS, and LEVEL5 is always available for the acquisition and refinement of the knowledge, as well as for the modification of the RMD-KBS. More information as to how the LEVEL5 knowledge editor has been used to build a KBS is given in the following sections.

The structure of the knowledge base of the RMD-KBS is designed as a set of several knowledge bases. Each of these KBs works with the LEVEL5 inference engine in order to constitute a symbolic processing module. Each module is designed to perform certain specific tasks, such as system initialization, display of the monitoring signal and indices, diagnostics and knowledge learning etc. While the RMD-KBS performs diagnosis, only one KB works with the inference engine of the system at a time. It may be noted that in such a design, the size of each KB, as defined by the number of rules contained in it, is relatively small. The relatively small size of KB will be more helpful in grouping different knowledge and in easing the inference, since any reasoning through a small group of rules will be certainly more efficient than searching a massive group of rules.

6.5 Sensors and Data Acquisition System

The sensors and data acquisition system are intended to obtain the vibration signal and provide the RMD-KBS with digitized data samples. The RMD-KBS has been designed to possess the capability of controlling the data sampling process. In the development of the prototype, however, the functions that are normally performed by the sensors and data acquisition system are simulated through accessing the off-line data files. Vibration signals have been acquired from several real-life industrial machines and also from (El-Karmalawy, 1993). The signals have been digitized and stored in a large number of off-line data files. These data files are organized by the database of the RMD-KBS. The information about the machines being monitored and the sensors (number, type, location etc) has also been obtained. The relevant data have been stored in the database of the RMD-KBS.

During the diagnostic process, the RMD-KBS asks the sensors for vibration data obtained under the sampling conditions or settings, as specified by it. This step of the diagnostic process is simulated as follows: when the RMD-KBS needs vibration data, it is provided by its database, instead of being directly provided by the data acquisition system. For the diagnosis, the required input information regarding real-life machines and sensors, is provided by the RMD-KBS itself. This way, the functionality of the sensors and data acquisition system is simulated without affecting the design of the RMD-KBS as an on-line monitoring and diagnostic KBS.

6.6 Database

The database of the RMD-KBS is able to handle various types of information and a large amount of vibration data, automatically and efficiently. In this manner, easy access to any piece of the information or any block of the data samples during diagnosis is assured. In the implementation of the RMD-KBS, this requirement has been achieved by structuring the database in an appropriate manner. It has also been designed such that the end-users are not requested either to learn the format of the database files or to create the files by themselves. The RMD-KBS provides the end-users with the utilities for accessing the database, so as to obtain the information stored in it, and to perform acquisition and refinement as may be needed. Descriptions of the utilities of RMD-KBS for database access will be given later in this chapter. Both the structure of the DB, and its intercommunication with other elements of the RMD-KBS are shown in Figure 6.3.

The LEVEL5 possesses a Database Interface (see Figure A.2.1), that can integrate a database file with a KBS. The database files must however be written in dBASE III[R] format, which uses a number of "fields" to define a complex data group and "records" to store the values of the "fields". A KBS developer can create a special type of object, called database class from the data file structure. The database class has the same number of attributes as the number of "fields", with each of them communicating a corresponding "field" of the dBASE III file, thereby reading from or writing the values into the file. The database class has also a group of system-defined procedures (commands or functions), that can be utilized (called in the knowledge base statements) to access the corresponding file.

The database of the RMD-KBS contains numerous database files that are external to the LEVEL5 system. As mentioned previously, there is a large number of off-line files that contain the vibration data samples. During the implementation of the RMD-KBS, a problem has been observed that the format of dBASE III files is not suitable to store the vibration data samples which are in a sample series format and not in tabular form. In order to handle the vibration data samples, as well as other types of information, two types of database files called data files and information files respectively, have been established in the DB. A data file is written in a text format as a series of data samples and can be accessed (both read and write) by the numerical processing routines of the RMD-KBS. They can also be edited by using any text editor. An information file is

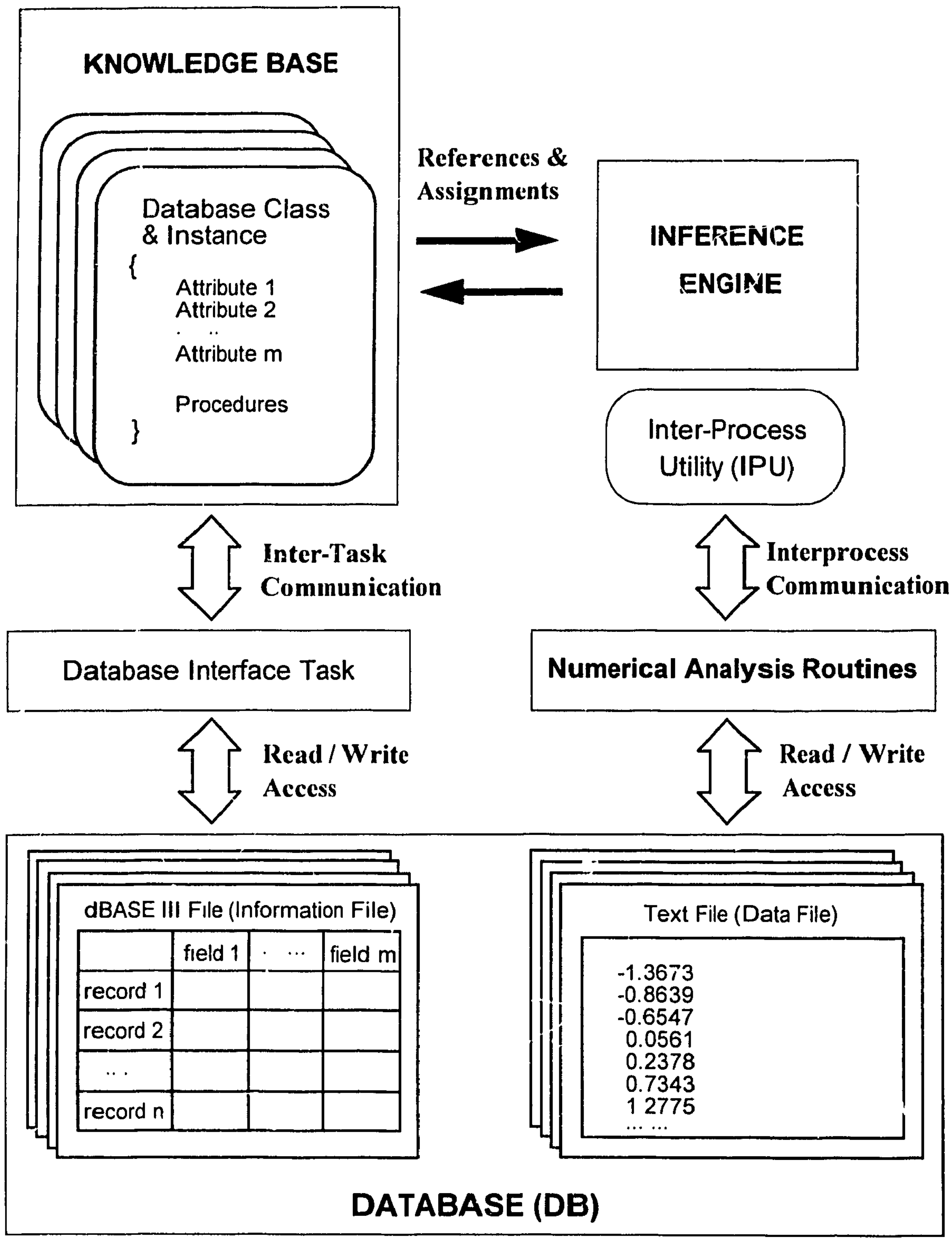


Fig. 6.3 The structure of the database of RMD-KBS.

established in the dBASE III structure that can be directly accessed by the KBS through the LEVEL5 database interface. The information files can also be edited by using the dBASE III database editor.

There are more than 200 data files in the DB of RMD-KBS, in which a large amount of data samples is stored. They include the data files of machine vibration signals, the files of some signal processing results, and those of historical data. In order to manage such a large number of files, the names and certain information about them are recorded in the dBASE III files that provide the information to the KBS. Based on the information, RMD-KBS can make a selection among the data files, and can assign a numerical processing routine to work on the selected data file at run-time.

Several types of information are stored in the DB under dBASE III format. The information includes the values of the structural and operational parameters of the machine system, the descriptions of machine components, the number, type and location of the sensors mounted on each machine, the signal processing settings, the values of the diagnostic index thresholds, and information about the data files themselves. There are 8 information files in the DB. The structure of these files is established using dBASE III+®, Version 1.0, and they are integrated into the RMD-KBS. In the knowledge bases of the RMD-KBS, 18 database classes are created corresponding to the 8 information files. They are designated as dB3 unitrgst, dB3 sensor, dB3 setting, dB3 bearing etc.

An example of a database class is shown in Figure 6.4. The database class illustrated is named dB3 sensor, and is linked to the dBASE III file named SENSOR.DBF that contains information about the number, type and location of the sensors mounted on the rotating machines being monitored. The dB3 sensor class, like other database classes, possesses both a group of attributes and a group of procedures

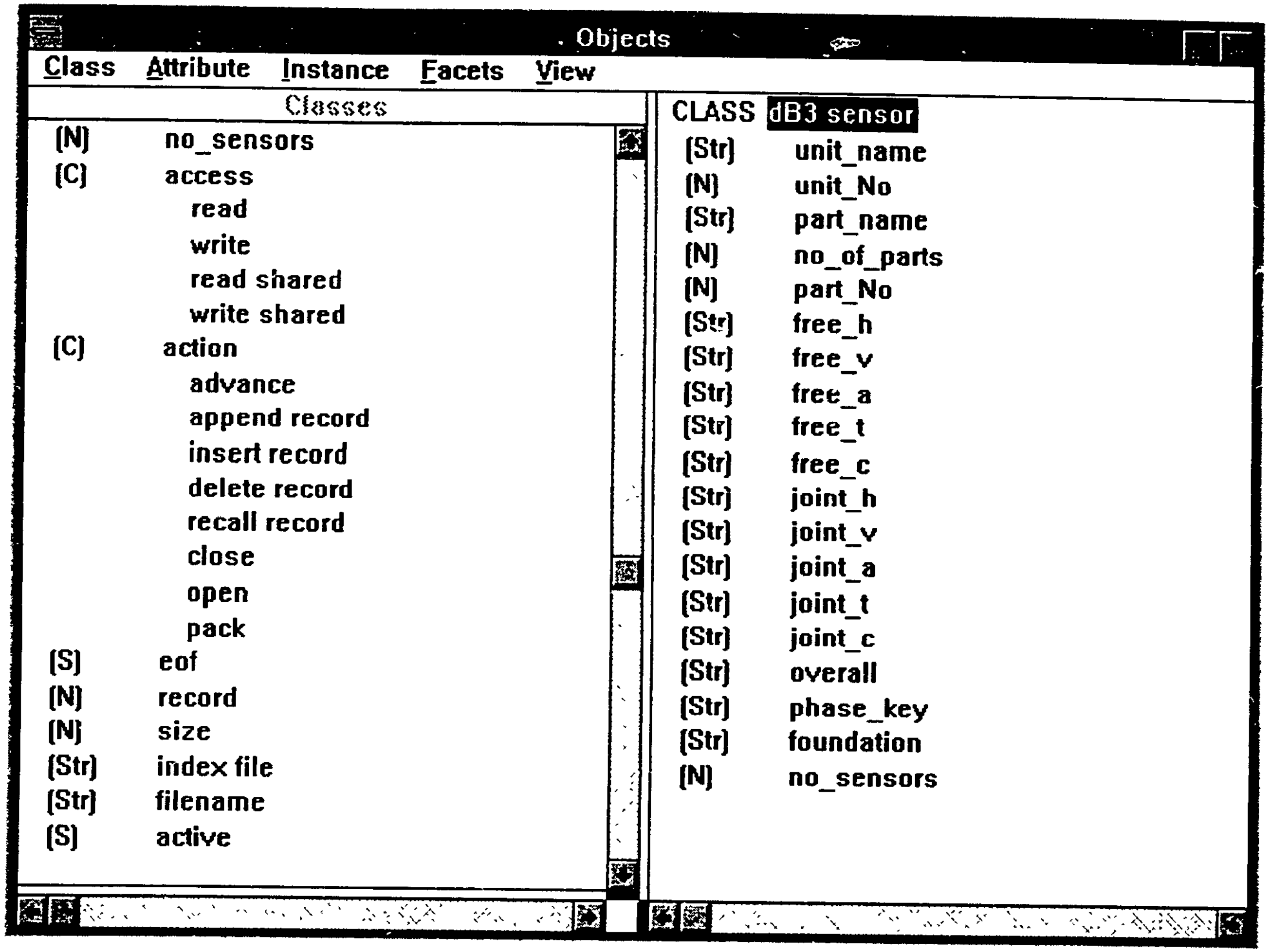


Fig. 6.4 The list of the elements of a database class named dB3 sensor.

(commands or functions). Its attributes are shown on the right window of Figure 6.4, which represents the structure of the corresponding information file with each attribute to communicate a field of SENSOR.DBF. The procedures of **dB3 sensor**, listed on the left window of Figure 6.4, are defined by the LEVEL5 system. These procedures can be called in the statements of rules, demons or methods, and are invoked in a reasoning session so as to open or close a file, to read or write a record, to insert or delete a record, and so on. For instance, the statements in the demon below ask the LEVEL5 inference engine to delete, from SENSOR.DBF, a record of previously stored information about the sensors on a machine system, that may be no longer used.

```
DEMON to delete record
IF to delete OF dunit actions AND cc OF variables > 0
THEN FIND dB3 sensor
        WHERE  unit_name OF dB3 sensor = string OF variables
        WHEN FOUND
                record OF dB3 unit := record OF dB3 sensor
                action OF dB3 sensor IS delete record := TRUE
                action OF dB3 sensor IS pack := TRUE
                action OF dB3 unit IS delete record := TRUE
                action OF dB3 unit IS pack := TRUE
                cc OF variables := cc OF variables - 1
        WHEN NONE FOUND
                to delete OF dunit actions := FALSE
FIND END
```

```
AND record OF dB3 sensor := 1
AND record OF dB3 unit := 1
AND LOOP
AND to delete OF dunit actions := FALSE
```

In the above demon, action OF dB3 sensor IS delete record := TRUE and action OF dB3 sensor IS pack := TRUE are the statements that perform the deletion of a record from the file.

As mentioned in Chapter 3, the DB is so designed as to enable the representation of factual knowledge. The files contain the "facts" (data, parameters, settings, conditions etc) of the factual knowledge. The knowledge is structured by objects (i.e. database classes) and corresponding attributes, which are in turn referred in the knowledge-based processing. Through such a shared representation, the acquisition and refinement of the factual knowledge is easily accomplished through a simple access to the database.

6.7 Numerical Analysis Routines

The RMD-KBS is designed as a deeply-coupled numerical and symbolic processing system which processes the vibration data by itself during diagnostic reasoning. It is also a hybrid system which employs connectionist technologies (i.e. neural networks) to represent the diagnostic knowledge in a quantitative mode. The neural network algorithms (see Chapters 4 and 5) perform their tasks through a numerical manipulation of data. Moreover, the KBS is required to display plots on the screen, to illustrate the data, information and some results of numerical computation. From a global view of the KBS, it can be seen that (i) the graphical display routines are utilized to enhance the user interface of RMD-KBS; (ii) the numerical processing capability of the system and the coupling between numerical and symbolic processing are accomplished through the design of the corresponding routines; and (iii) the neural network algorithms and the incorporation of neural networks with the rule-based symbolic processing modules of the system are also realized by invoking external programs.

The implementation of the above functions of the RMD-KBS is not feasible under the LEVEL5 development tool alone, since it does not support complex plots and numerical calculation. Hence, programs that are external to LEVEL5 are employed in the RMD-KBS. Routines for graphical display, numerical calculation and neural networks, are encoded in C++. They constitute the component named Numerical Analysis Routines (NAR). The RMD-KBS can invoke them via its symbolic statements through a LEVEL5 utility named Inter-Process Utility (IPU). The connection between the inference engine and the external routines is shown in Figure 6.3. Two types of external programs, named EXTERN and SERVER programs, can be called from the statements of rules, demons, or methods. An EXTERN program is a common Windows application (i.e. the program can run under Microsoft* Windows™) without any special codes. It can receive messages and different types of data from RMD-KBS. On the other hand, a SERVER program is specifically written to communicate with the LEVEL5 inference engine, that can also send its responses or calculation results back to the KBS.

The external programs of the RMD-KBS are encoded in C++, using the Borland C++ compiler, Version 3.0. They are designed for graphical display, signal processing and neural network applications. Some of them are designed as EXTERN programs, and the others, as SERVER programs, depending on the amount of information to be passed between a routine and the KBS. The routines for vibration signal processing can perform spectral analysis and statistical calculations in order to extract a number of indices from the vibration data samples. The mathematical expressions of the statistical methods implemented in the routines are listed in Appendix A.1. Further, the programs of neural network applications implement the self-organizing networks for fault pattern identification and classification, the back-propagation networks for machine condition classification, and the newly developed SOM algorithm for multiple-index based trend analysis. The neural network programs can also display plots on the screen. The way of calling external programs from the RMD-KBS is illustrated through the following examples.

The statements given below may be seen in a rule, demon or a method of RMD-KBS. They invoke an EXTERN program, named ANALYSIS.EXE to present both a set of vibration data samples and the FFT spectrum of the data set on a window, as shown in Figure 6.5.

```
ACTIVATE   "IPU, EXTERN, d:\l5o\z\ANALYSIS.EXE"
COMMAND  Command of Variables
```

The statement ACTIVATE is a command to call an external program through the LEVEL5 IPU, with the specifications of the type and name of the external program. The statement COMMAND is used to send a message carried by an attribute of a class, i.e. Command of Variables, to the external program. The message could include the name of the data file, the name of the machine, and the type and location of the sensor that picked up the vibration signal to be displayed on the screen.

The signal processing and vibration feature extraction routines are written as

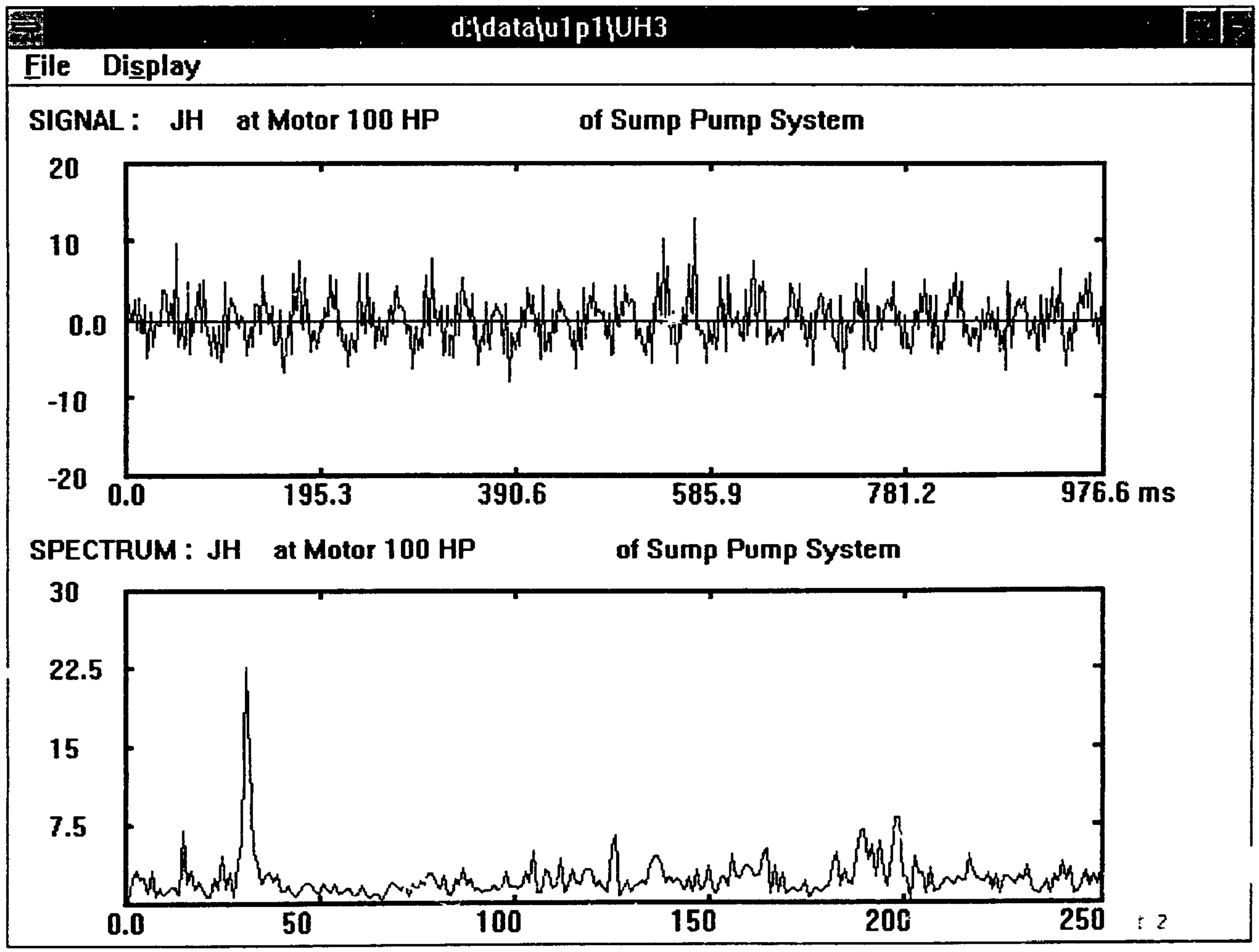


Fig. 6.5 Windows display of the vibration signal and the spectrum using an EXTERN program.

SERVER programs to LEVEL5, since more communication is required in calling them. The way to call a SERVER program is shown through the following example.

```
ACTIVATE   "IPU.SERVER, d:\l5o\z\PROCESS.EXE"
SEND       File Name of Bearing1
SEND       Analysis f of Setting
RECEIVE    PP of Indices
RECEIVE    AX of Indices
RECEIVE    CR of Indices
RECEIVE    AM of Indices
```

Via the above statements, the RMD-KBS calls a SERVER program, named PROCESS.EXE. The SEND commands send messages about the data file name and the value of the analysis frequency to the routine. Further, RMD-KBS asks the routine to calculate and then to return the values of four indices, namely Peak-to-Peak value, Absolute Mean value, Crest Factor, and Arithmetic Mean value respectively.

The programs of neural networks are written in both EXTERN and SERVER application forms. When the RMD-KBS trains a neural network for the purpose of learning knowledge, the corresponding program is designed as an EXTERN program. While the RMD-KBS invokes a trained neural network to perform fault classification or trend analysis, the corresponding routine is encoded in the SERVER application form, that can send back its calculation results to the RMD-KBS. The way of calling an EXTERN or SERVER program of neural networks is similar to either one of the examples given above. A neural network program may also draw some plots on the screen to illustrate its

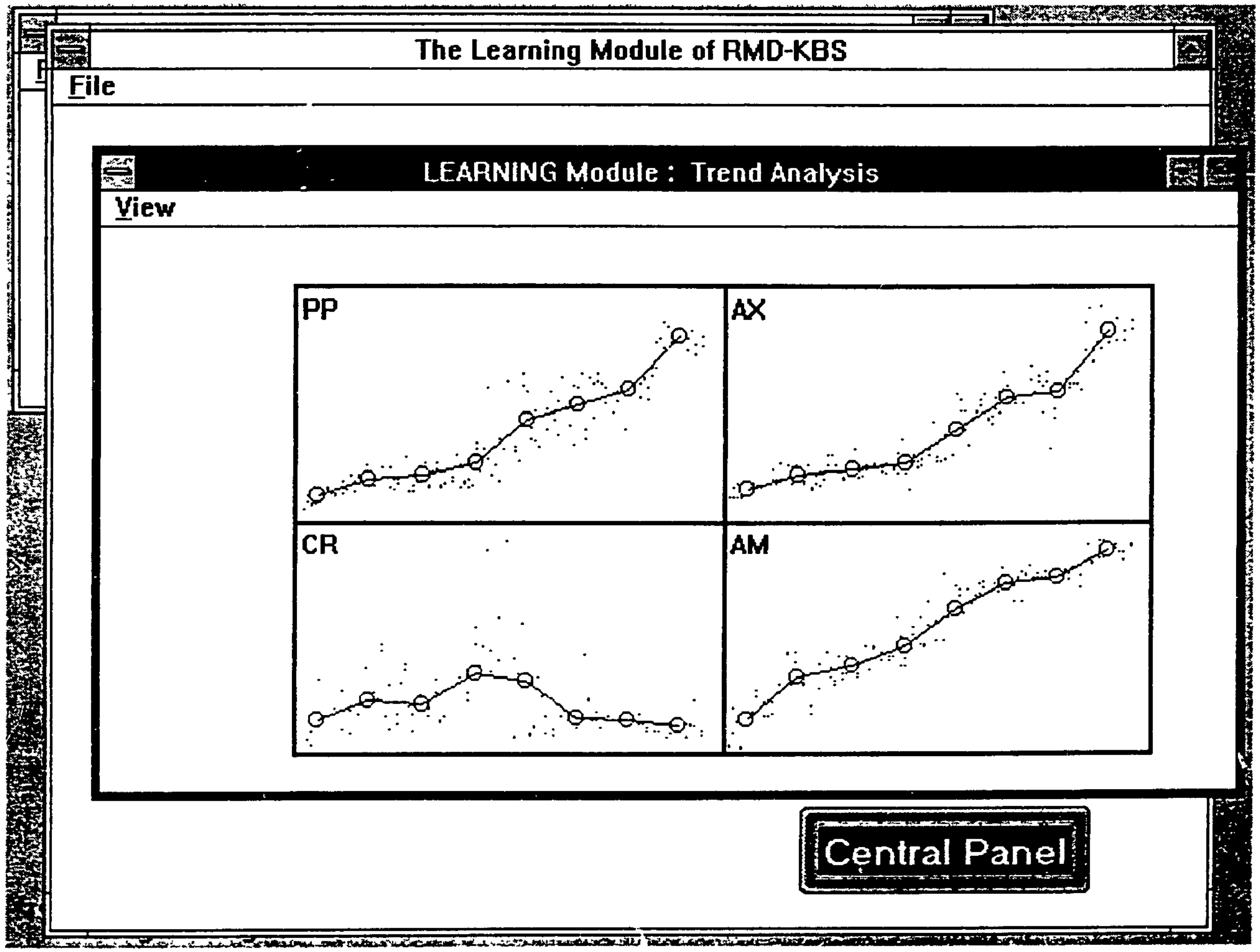


Fig. 6.6 A plot shown by the external program of training the SOM for trend analysis.

performance. Figure 6.6 is drawn by the program that uses the newly developed SOM algorithm to perform a multiple-index based trend analysis.

6.8 The Multiple Windowing User Interface

The user interface provides a man-machine communication environment. The end-users communicate with the RMD-KBS and thereby operate it through the user interface. The UI performs two fundamental functions: (i) obtain the input information from an end-user, e.g. to receive a value, a piece of information, or a user selection among several options; and (ii) post the output information such as figures, messages, and queries for user response. In this implementation, an input is called a user interface event, and the output is called a user interface action.

The UI of the RMD-KBS has been built in Windows format, and is composed of window frames, graphical displays on the windows, and various graphical elements that constitute a display. It is mainly built by using the utilities of LEVEL5. In addition some graphics programs have also been encoded in C++ to enhance the capability of the UI.

The development of the user interface is not only to create the windows and displays, but also to implement the functions that are desired to be performed by the displays and their elements. The man-machine communication is the most basic function of the UI. For example, a graphical element called Promptbox can be defined in a display which receives an input from the user. There must be an attribute of an object, that is attached to the Promptbox to receive the input value. After the input has been received, it may be considered as an UI event to drive the system to perform a task, depending on the design of the interactions between the UI events and the other parts of RMD-KBS.

The actions of the UI are fully controlled by the RMD-KBS. For example, at the stage of system development, a Picturebox, that is a square frame with a picture attached to it, can be defined within a display. When the Picturebox appears on a window, the corresponding picture will be posted within it. It is also allowed to attach another picture to the same Picturebox, at the time when something else is to be shown on the screen. This way, the RMD-KBS performs the communication with its user through its user interface. The UI has been designed to offer more convenience and clarity to the user, as can be seen from the demonstrations reported in the next chapter.

6.9 Inference Mechanism

The inference engine employed in the RMD-KBS is that of the LEVEL5 system. As mentioned previously, it can perform forward chaining, backward chaining and mixed-mode chaining. It can also calculate the confidence factors (see Appendix A.2) involved in the rules or demons during the evaluation. When the LEVEL5 is used to implement the RMD-KBS, the selection of an inference scheme that is highly suitable for the intended application becomes an important task. According to the selected inference scheme, the statements to be written in the corresponding KB could differ in format. For example, the rules and the WHEN NEEDED methods are evaluated and fired by the backward chaining scheme, while both the demons and the WHEN CHANGED methods are processed by the forward chaining scheme. In the RMD-KBS, both forward and backward chaining strategies are employed in order to optimize the performance of the diagnostic system. The details as to when and where the different inference strategies have been employed, will be given in the next section along with the description of the knowledge bases of the RMD-KBS. Furthermore, some schemes of inference procedure control, such as "fire the first" and "fire all" (to evaluated the first or all, of the rules in a rule group), are also employed.

6.10 Knowledge Bases

Five knowledge bases have so far been implemented in the RMD-KBS. Each one works together with the inference engine of LEVEL5, thereby constituting a symbolic processing module of the RMD-KBS. The symbolic processing modules thus constituted are named MAIN-KBS, SET-UP, BROWSER, MONITOR and MENTOR. This way, the entire knowledge has been partitioned into separate knowledge bases that are relatively smaller in size. The development and management of the small sized KBs are much easier than developing a single huge knowledge base.

More than 400 instances of 101 objects have so far been defined in the knowledge bases of the RMD-KBS. The attributes of the objects are referred in 57 rules, 136 demons and 89 methods that have been encoded in the KBs. It may be noted here that several methods have been used. Because a method contains many more statements than what a rule or a demon, it can perform complex tasks. Before describing the implementation of the five knowledge bases, two aspects, viz., (i) linking the modules, and (ii) sharing the data and information among the modules, will be considered.

The link between the five symbolic processing modules is shown in Figure 6.7. The MAIN-KBS is the one which can be linked to any other knowledge base, and there is no direct link between the other knowledge bases. At any time, there is only one knowledge base that is working with the LEVEL5 inference engine. A module is linked to another module using the CHAIN command of LEVEL5. When the LEVEL5 executes a CHAIN command, the current KB ends, and a new KB is invoked. The following

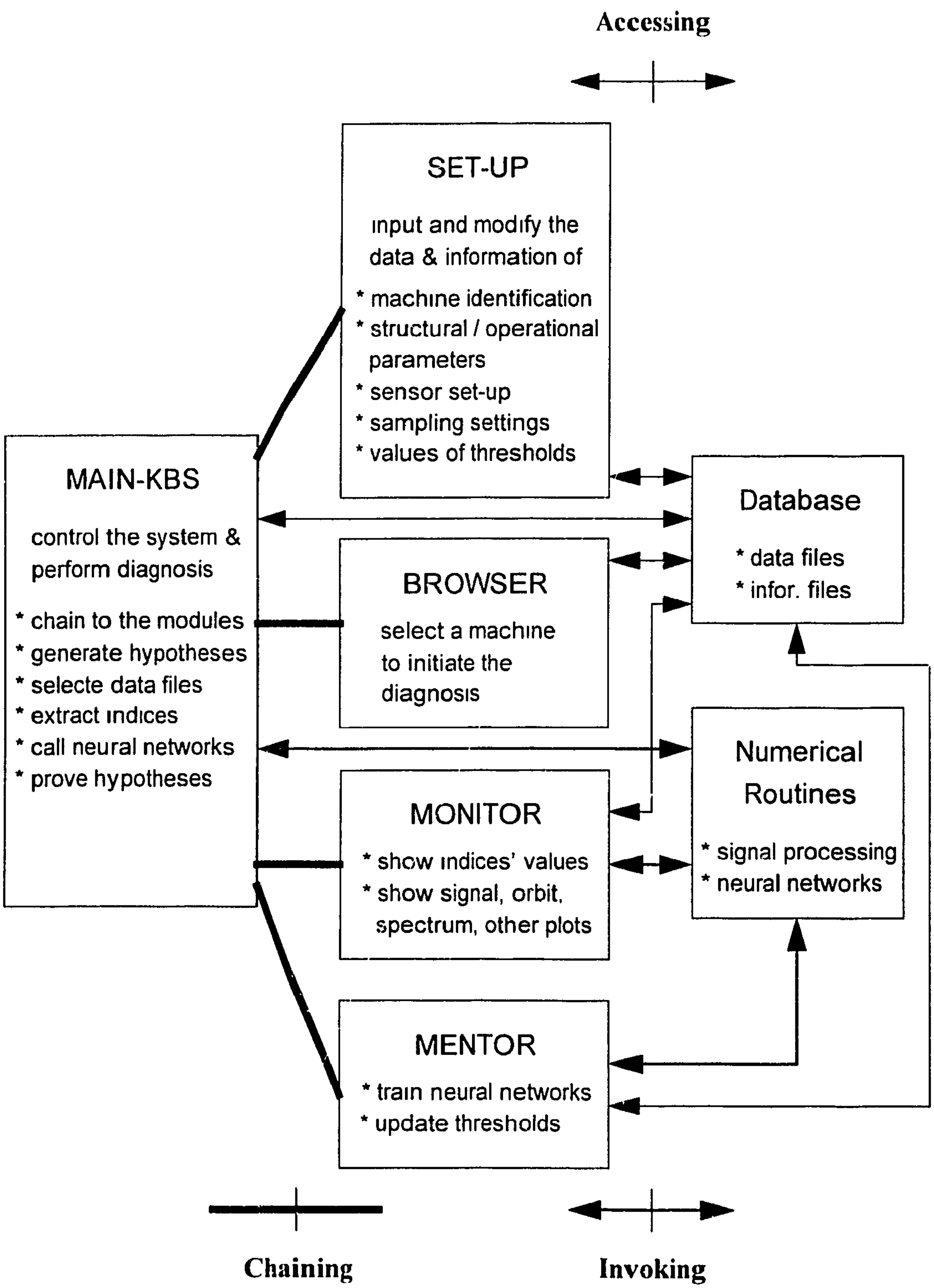


Fig. 6.7 The connectivity between the modules of RMD-KBS.

example shows a typical WHEN CHANGED method, that invokes the BROWSER from the MAIN-KBS.

```
WHEN CHANGED
 BEGIN
     record OF dB3 unitrgst := 1
     record OF dB3 bearing := 1
     record OF dB3 sensor := 1
     record OF dB3 setting := 1
     CHAIN  "browser"
 END
```

When a knowledge base is linked to another knowledge base, there may be a message that needs to be transmitted from the current KB to the new one. Further, different modules of the RMD-KBS may need to refer to the same information and data in performing their individual tasks. Data sharing between two knowledge bases is realized in two ways. The first one is the so called "values sharing" of object attributes. All of the knowledge bases have the same object, named domain, which is specified as a SHARED object. The current values of the attributes of domain can be directly passed to the new KB. This method is used in the implementation in order to carry the message that is immediately needed at the beginning of a new module. The second way of data sharing employs the database classes for data sharing between the knowledge bases. Database classes corresponding to the same database file can be established in different KBs. Hence, the different modules can get the same information from the database file.

Most of the data and information to be shared by the KBs are handled this way. The five modules are described in the following sections. The way with which they are used, and their performance are illustrated in the next chapter.

6.10.1 The SET-UP Module

The SET-UP is an important knowledge-based module, which is developed as a tool that helps the end-users to (i) initialize and adapt the RMD-KBS to any individual application, and (ii) refine the information that is stored in the information files. With this tool, the user can input the identifications of the machine system being monitored. Users can specify or modify the values of both the structural and operational parameters of the machines or the machine components. They can also input sensor information and assign monitoring index threshold values, using the SET-UP module. Figure 6.8 shows the first window of the SET-UP module.

The SET-UP module is not only a tool for the users to access the information files in the database, it also automatically determines the signal processing settings, such as the sampling frequency, filtering frequency, the frequencies at which the spectrum amplitudes should be taken as monitoring indices, and so on, based on knowledge of the signal measurement process and the signal processing methods, that are embedded in it. The following demons in a demon group are employed to determine the correct sampling frequency.

```
DEMON setup 4: sampling f
IF          to setup OF setting actions
AND         0 < rotating f OF limits
AND         rotating f OF limits < 10.0
```

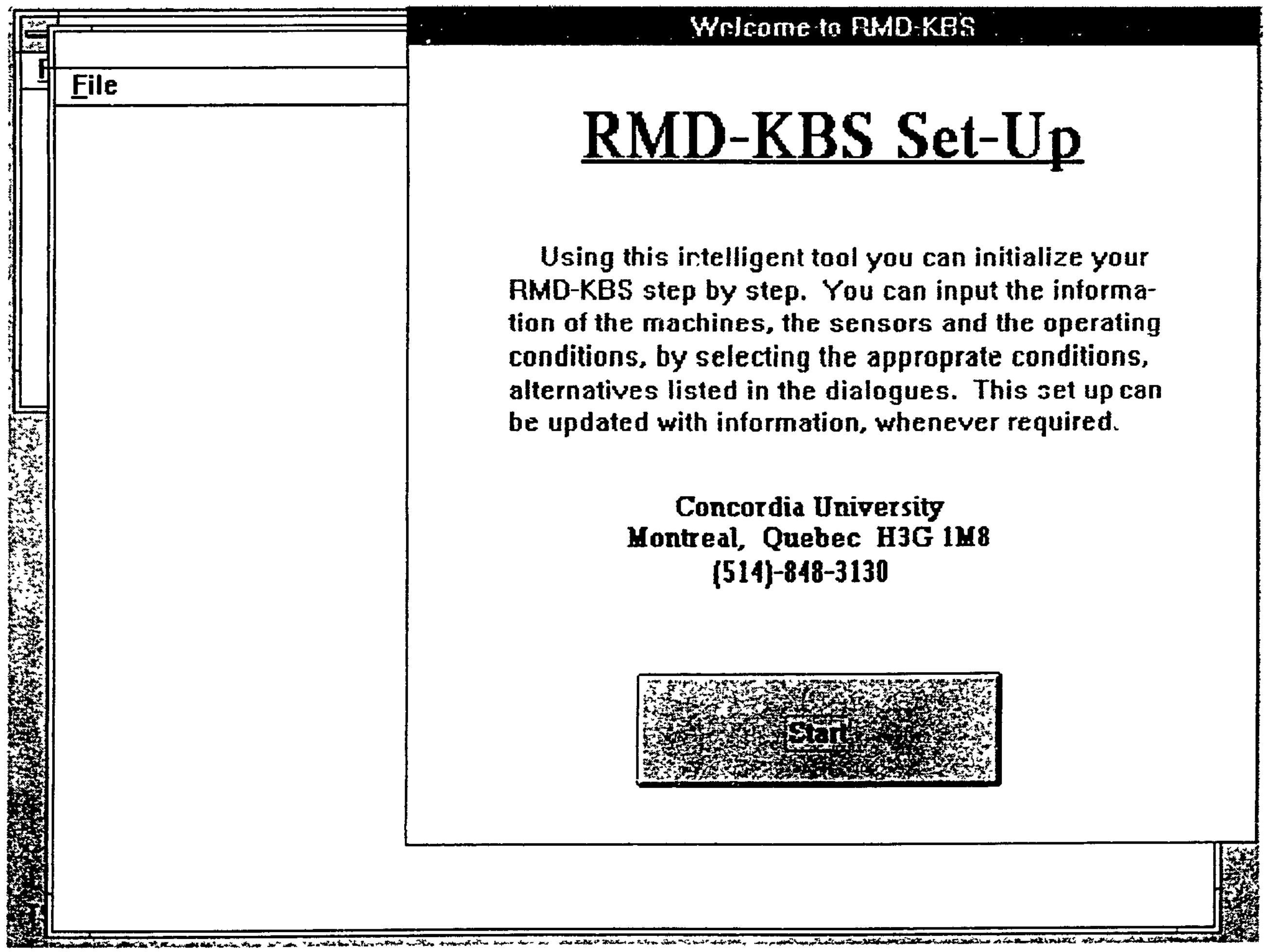


Fig. 6.8 Windows that appear at the beginning of the SET-UP module.

```
THEN        sampling f OF limits := 128.0


DEMON setup 5: sampling f
IF          to setup OF setting actions
AND         10.0 < rotating f OF limits
AND         rotating f OF limits < 20.0
THEN        sampling f OF limits := 256.0


... ...


DEMON setup 9: sampling f
IF          to setup OF setting actions
AND         rotating f OF limits > 240.0
THEN        sampling f OF limits := 4096.0
```

The SET-UP module has a sophisticated user interface to perform the man-machine communication needed for information acquisition. Inside the SET-UP, the inference is mainly driven by the user interface events. Further, the determination of the settings is also performed according to known information (i.e. the user input and other information). Hence, the forward chaining strategy is selected to work with the knowledge base of the SET-UP tool.

6.10.2 The BROWSER Module

The BROWSER is the smallest and the simplest module of the RMD-KBS. It is designed for the user to select a machine as the subject to diagnose by browsing through the information files of all the machines under monitoring. The BROWSER provides the user with information about the machine systems, their subsystems and components, the sensors and so on.

The RMD-KBS has been designed to monitor machine systems by acquiring vibration signals and calculating the values of a set of monitoring indices. The diagnosis process is triggered when an abnormal value of any monitoring index is detected. However, the initiation of a diagnosis task can also be performed by the user. When a particular machine has been selected by the user, the BROWSER will gather the necessary information and pass it to the MAIN-KBS. The statements in the knowledge base of BROWSER are written in a form suitable for forward chaining.

6.10.3 The MONITOR Module

In the diagnostic process, the end-users may want to check the current values of the monitoring indices and to see the vibration signal and the spectrum. The MONITOR module is intended to provide the user with the above information. The database can be accessed, graphical routines can be called to display the plots, and numerical routines can be invoked to perform the calculations by the MONITOR module. Further, this module is important to the system since it enables the experienced diagnosticians to compare the results of the knowledge-based reasoning with what they infer from the information and displays provided by MONITOR. The diagnosticians can thus decide whether the knowledge embedded in the RMD-KBS is not adequate and so should be refined, and in that case, what type of new knowledge is needed to be added into the system.

The MONITOR module is implemented through symbolic processing with the forward chaining scheme. Several graphical and numerical routines are involved in this operation. It can also access both the information and data files. Hence, it is quite large in size, i.e. it contains a large number of demons and methods. The graphical and numerical routines can be further improved so as to implement more methods that are available for vibration signal monitoring and analysis. Figure 6.9 shows a small part of the Knowledge Tree of MONITOR. The Knowledge Tree (see Figure A.2.9) is a utility of LEVEL5, which graphically shows the structure of a knowledge base.

6.10.4 The MENTOR Module

The MENTOR module is designed to perform two tasks: (i) to train the neural networks, and (ii) to adjust the values of the monitoring index thresholds. Three types of neural networks have so far been implemented in this prototype, i.e. back-propagation neural networks, self-organizing neural networks and the new SOM algorithm for multiple-index based trend analysis. The learning algorithms of these neural networks have already been described in Chapters 4 and 5. This module can select a data file from the historical data files and then perform the training of neural networks. When sufficient historical data have accumulated, the adjustment of the threshold values of diagnostic indices could be performed using statistical methods to calculate the proper values of both the minimum and maximum thresholds of an index. The MENTOR module also works with the forward chaining scheme, since the learning processes are data-driven in nature. It is an independent module that will not be involved in the diagnosis, but will be used when the user wants to train a neural network. A WHEN CHANGED method described below calls an external routine, that is one of two routines designed for training the SOM networks, according to the context of the reasoning session.

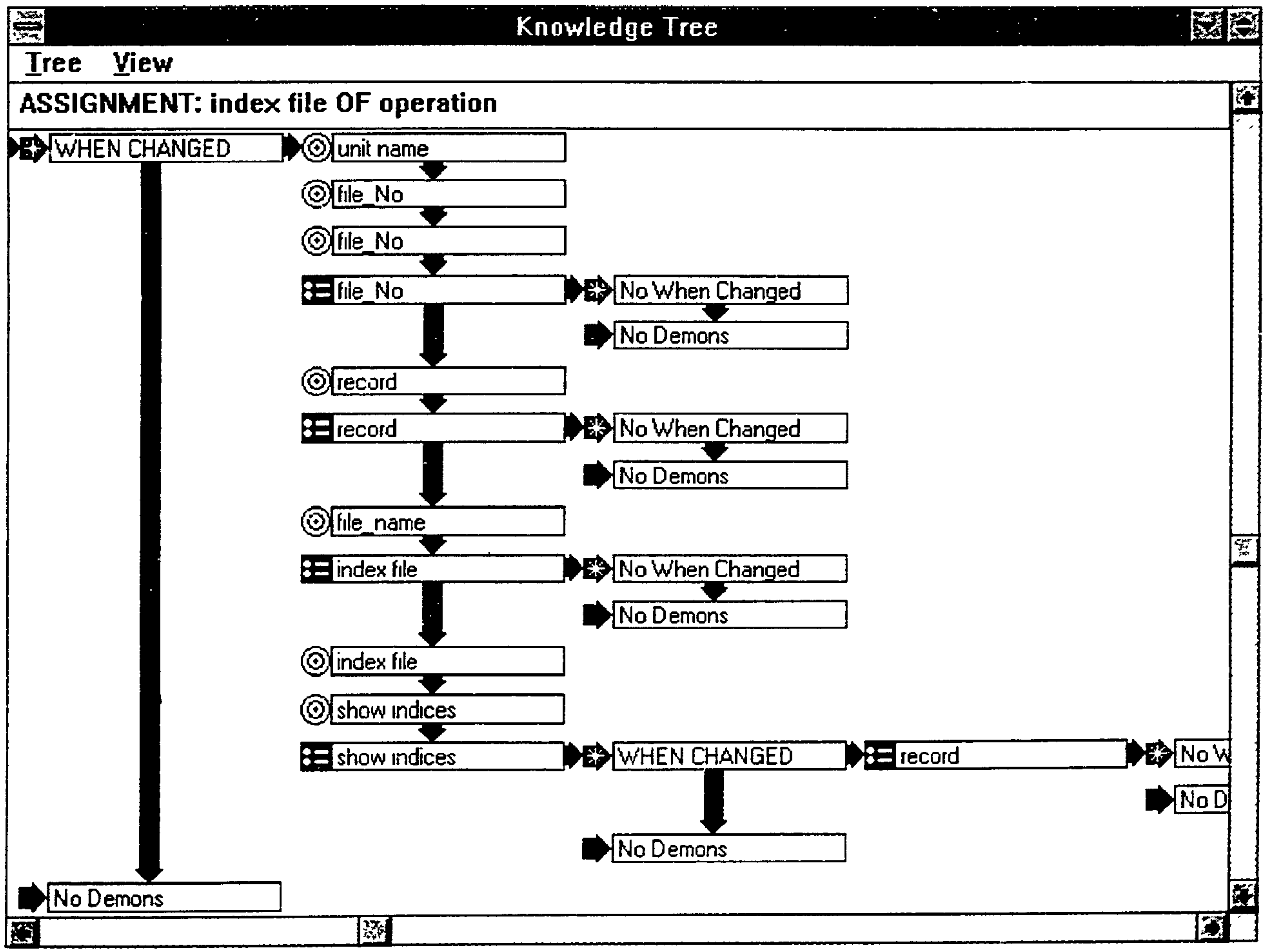


Fig. 6.9 The knowledge tree that shows the structure of the knowledge base of the MONITOR.

```
WHEN CHANGED
    BEGIN
    visible OF expand window := FALSE
    commands OF control := CONCAT( commands OF control, "+")
    commands OF control := CONCAT( commands OF control, nNodes
                                    OF control)
    commands OF control := CONCAT( commands OF control, "+")
    commands OF control := CONCAT( commands OF control, nCycles
                                    OF control)
    commands OF control := CONCAT( commands OF control, "+")
    IF case OF control = 1 THEN
        BEGIN
        ACTIVATE  "IPU,EXTERN,TRENDW.EXE"
        COMMAND  commands OF control
        END
    ELSE
        BEGIN
        ACTIVATE  "IPU,EXTERN,SOMW.EXE"
        COMMAND  commands OF control
    END
END
```

The attribute, nNodes OF control, carries the number of nodes of a new SOM network to be established. Another attribute, nCycles OF control, carries the value of the number of times the data sample sets are presented to the network during the training. The values of these two attributes are assigned by the user before firing the above WHEN CHANGED method. The external routines designed to perform the training of the two different types of SOM networks are named SOMW.EXE and TRENDW.EXE. In the above example and in the examples previously presented, it can be seen that the control of the symbolic process is performed by the statements in the KBs. The numerical processing routines are also controlled by the symbolic commands and statements. Such statements in the KBs represent the control knowledge which was summarized in Chapter 3.

6.10.5 The MAIN-KBS Module

The MAIN-KBS module performs the most important task of the diagnostic system. The knowledge regarding rotating machinery diagnosis is mainly manipulated in the knowledge base of this module. It can chain to any of the above four modules.

As mentioned in Chapter 3, the diagnostic strategy employed in the RMD-KBS, consists of three steps, i.e. (i) generate the hypotheses of the suspected fault; (ii) determine the relevant signal analysis methods and diagnostic indices; and (iii) determine the condition of the machine in light of the assigned hypotheses. The manipulation of the diagnostic knowledge by symbolic statements is described below.

There are five groups of rules, demons and methods in the KB of the MAIN-KBS module. The neural networks to be invoked by the MAIN-KBS can be considered as another group. Both the hierarchy of reasoning out the above six groups, and the information flow during the reasoning are shown as Figure 6.10. The diagnostic tasks and the reasoning process to be performed by MAIN-KBS is described below.

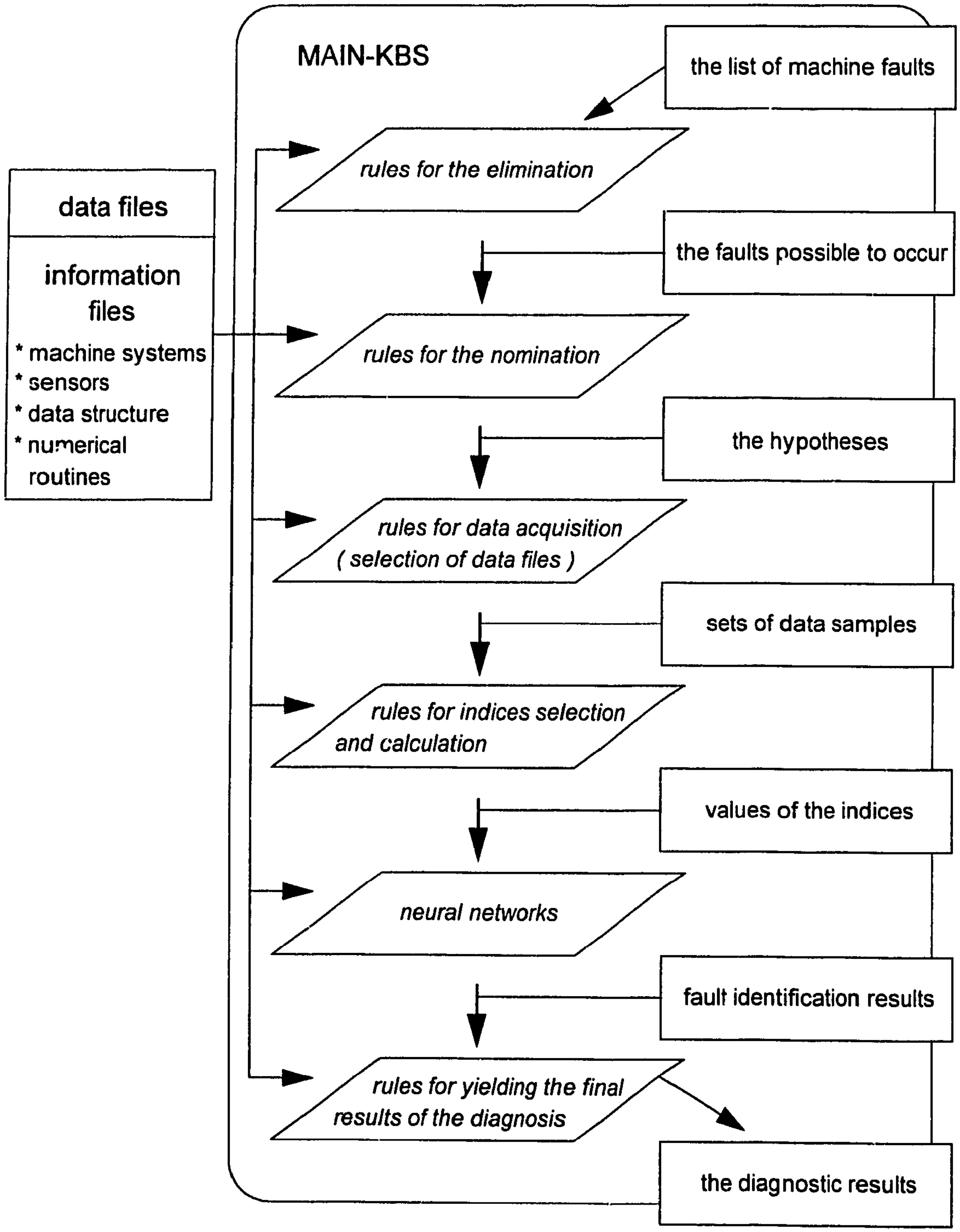


Fig. 6.10 The hierarchical structure of the rules and the information flow in MAIN-KBS.

Generating the Hypotheses of Machine Faults: In the KB of the MAIN-KBS, a list of faults and malfunctions which are frequently-encountered in rotating machinery and are documented in the published literature (Sohre, 1980; Dimarogonas, 1992; Eshleman Jackson, 1992), has been encoded. The faults or malfunctions are summarized in Table 6.1. The hypotheses to be generated by the MAIN-KBS are selected from the above list, based on the available information about the machine system, the values of the monitoring indices, the diagnostic knowledge and experience, and the heuristics and rules of thumb regarding fault-symptom relationships, which are stored in the MAIN-KBS in symbolic mode. The forward chaining inference scheme is followed through demons and WHEN CHANGED methods with confidence factors, to generate the hypotheses.

The strategy of generating the hypotheses basically involves two steps of reasoning as shown in Figures 6.10 and 6.11. After a machine, a sensor on it and the corresponding data file are all selected by the user, the values of several monitoring indices are calculated from the data samples in the file. These values are normalized to have a value between 0 and 1, using the method defined in Chapter 3. Subsequently, the hypotheses are generated by the hierarchically connected demons and methods in the first and second groups, through the following reasoning.

1) Determine the faults and malfunctions from the list, that can not occur or can not be detected in the machine being diagnosed, based on the input information and the available information about the structural and operational parameters of the machine, the information about the sensors, and the diagnostic knowledge and experience encoded in the MAIN-KBS. This step of reasoning is designated as "elimination", and actually cancels form the above list, the faults that can not occur in the subject machine. The remaining faults are marked as the "candidates" and they are evaluated by another group

Table 6.1 The list of machine faults and malfunctions.

Components	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5
Rolling element bearing	Wear	Pitting	Sliding		
Coupling	Misalign-ment	Resonance	Inaccuracy & damage		
Foundation	Structural resonance	Insufficient tightness	Resonance of support		
Gears	Wear	Teeth broken	Scoring	Pitting	Spalling
Journal bearing	Oil whirl	Eccentric			
Rotor / shaft	Unbalance	Cracked	Resonance	Run out	Bow
Seal	Oil seal induced vibration	Rub			
Thrust bearing	Wear	Rub	Damaged surface		

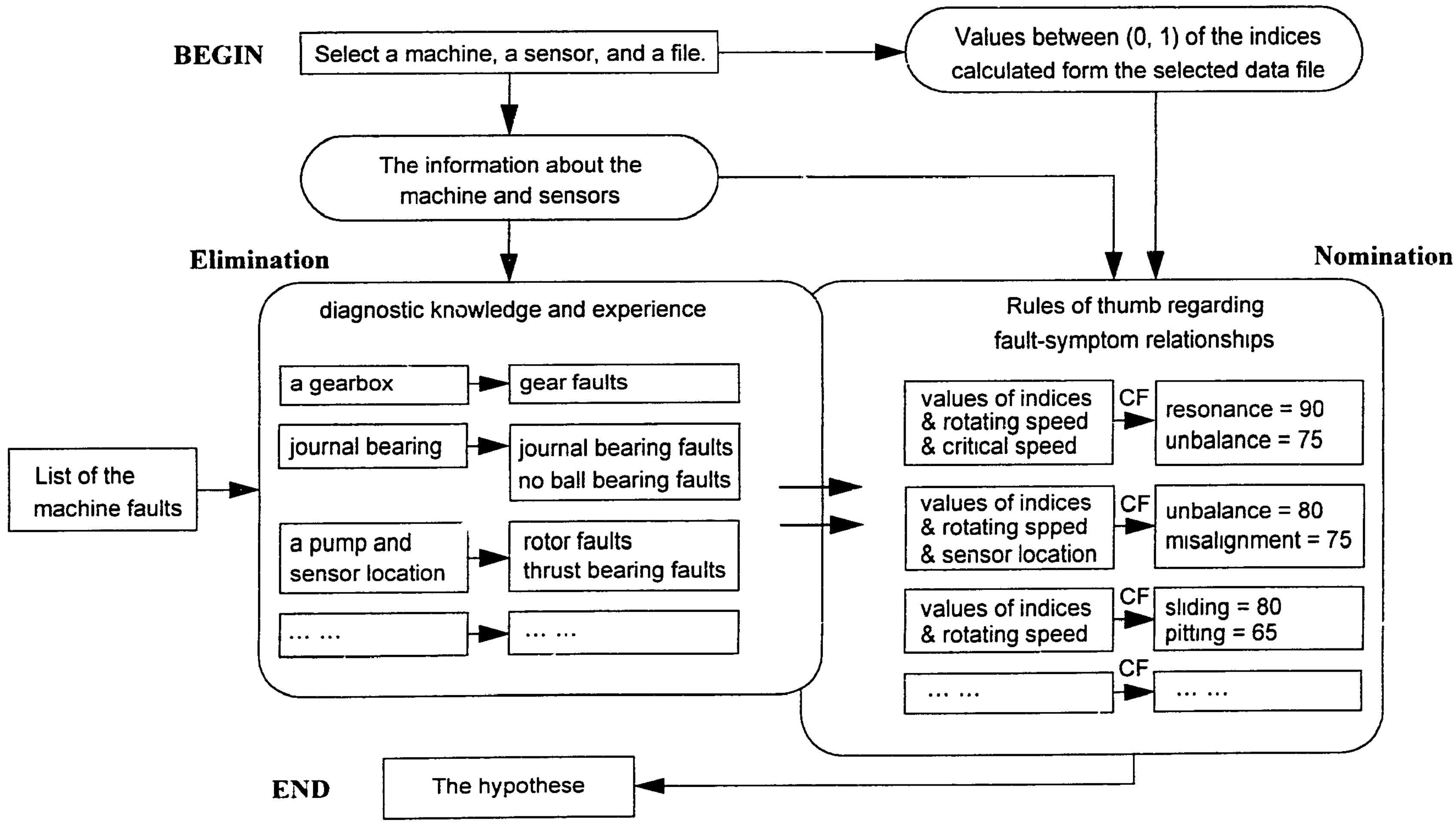


Fig. 6.11 The strategy of generating the hypotheses.

of demons and methods in the second group.

2) Evaluate the values of the monitoring indices based on rules of thumb regarding the fault-symptom relationships, in order to determine which faults and malfunctions are the most plausible faults, i.e. the hypotheses. The facts about both the machine system and the sensors should also be referred to this end. This step of reasoning is called "nomination", and selects a few faults and malfunctions as the hypotheses, i.e. the most likely faults.

For example, the demon below eliminates all the journal bearing faults, since the type of bearing in the monitored machine is a rolling element bearing. Meanwhile, tne demon sets the values of the attributes of the ball bearing fault to 100, so that further evaluation of other demons in the second group will be triggered at a later stage. The factual knowledge of the bearing type is actually provided by a database class, named dB3 unitrgst, which in turn gets the relevant facts from the corresponding database file.

```
DEMON bearing problems 1
IF      bearing OF dB3 unitrgst = "Rolling Element Bearing"
THEN    Wear OF ball bearing fault := 100
AND     Pitting OF ball bearing fault := 100
AND     Sliding OF ball bearing fault := 100
AND     Oil whirl OF journal bearing fault := 0
AND     Eccentric OF journal bearing fault := 0
AND     Damaged surface OF journal bearing fault := 0
```

Another demon given below, recognizes that no sensor has been mounted along the axial direction to measure the vibration signal. Then, it has to eliminate the category of thrust bearing problems, because these problems are difficult to identify based on the available information.

```
DEMON thrust bearing problems
IF          axial bearing OF dB3 unitrgst := "Doesn't have"
OR          ( free_a OF dB3 sensor = ""
            AND  joint_a OF dB3 sensor = "" )
THEN        wear OF thrust bearing fault := 0
AND         rub OF thrust bearing fault := 0
```

When the other demons similar to the above have been evaluated, the faults and malfunctions in the list are assigned a value of either 0 or 100. Those faults that are assigned a value of 100 will be further judged by other demons or methods, that represent the fault-symptom relationships, in the second group. At the stage of knowledge acquisition to establish the RMD-KBS, the fault-symptom relationships have been collected. For example, regarding rotor unbalance, the possible symptoms are (Sohre, 1980; Eshleman and Jackson, 1992): (i) a distinctly-high amplitude appears at the rotating frequency in the spectrum, with much lower level amplitudes at the 2nd and 3rd harmonics; (ii) the value of the amplitude at the rotating frequency increases when the rotating speed increases; (iii) the values of certain time domain indices are greater than their corresponding values at the normal operating condition of the machine, e.g. the Peak-to-Peak value, RMS value and the Absolute Mean value; (iv) the above symptoms can be obtained from the sensors at both the vertical and horizontal directions. The demon given below provides a heur:_tic suggestion about the current fault, based on the above knowledge about the occurrence of unbalance.

```
DEMON of unbalance
IF          Unbalance OF rotor fault = 100
AND         ar OF dB3 index > 0.25
AND         ( pp OF dB3 index > 0.3
            OR    rms OF dB3 index > 0.3
            OR    ax_mean OF dB3 index > 0.3 )
THEN        Unbalance OF rotor fault := 90 CF 90
```

A number of demons similar to the above, are available to represent the judgemental knowledge and determine which fault is possibly developing. Another demon given below considers the features of the vibration signal and an additional condition, as to whether the location of the sensor is close to the coupling of the machines.

```
DEMON of misalignment
IF          misalignment OF coupling fault = 100
AND         ( location OF sensor = "JH"
            OR    location OS sensor = "JV" )
AND         a2 OF dB3 index > 0.2
AND         ( pp OF dB3 index > 0.2
             OR   rms OF dB3 index > 0.25 )
THEN        misalignment OF coupling fault := 85
```

AND unbalance OF rotor fault := 60 CF 80

It may be noted here that confidence factors are employed in the above two examples. The usage of confidence factor scheme is helpful to represent the heuristic knowledge in the above examples, since the rules of thumb regarding the fault-symptom relationships are usually not in the form of simple YES or NO answers.

After this round of reasoning, the faults and malfunctions are assigned to a value between 0 and 100. Three of the faults with the highest values are selected as the suspected faults, i.e. the hypotheses. It has been mentioned in Chapter 3 that this step of diagnostic reasoning, yields reasonable hypotheses about the faults, that can narrow down the following steps of reasoning and make the system more efficient. The hypotheses are then set as agenda items to be proved through the backward chaining inference in the next two steps.

Feature Extraction: As shown in Figures 3.3 and 3.4, in accordance with the posted hypotheses, signal analysis methods are selected next in order to calculate the diagnostic indices which will quantify machine condition. The following three tasks are performed based on the corresponding knowledge encoded in the MAIN-KBS.

1) Determine from which sensors the vibration data should be acquired in order to prove the hypotheses.

2) Determine the analysis methods, such as time domain analysis, frequency domain analysis, and cepstrum, that should be employed to extract the features from the data samples.

3) Calculate the values of the diagnostic indices.

In the last step, several monitoring indices are used in generating the hypotheses. Their values are extracted from only one file, the user selected file, that stores the data samples from a sensor that is mounted on the selected machine. In the design of the RMD-KBS, data samples from other sensors will also be used in the next diagnosis step. In general, a monitoring system could have more than one sensor mounted on a single machine, to measure the vibration signals at different locations and in different directions. When a certain type of machine fault occurs, it may be detectable by more than one sensor at the same time. The diagnostic analysis, which considers the features of the vibration signals from more than one sensor, provides more accurate results. For example, the sensors mounted in the vertical and horizontal directions can both detect the presence of unbalance. The values of the indices that are sensitive to the rotor unbalance problem can be extracted form the two sets of data samples that are measured from the sensors along the vertical and horizontal directions, respectively. If both of them show unbalance, the diagnosis is quite obviously the unbalance problem. In the MAIN-KBS, additional data samples from other sensors (usually one), if they are available, will be taken into consideration. This task of obtaining additional data is performed by several rules and WHEN NEEDED methods, that are in the third group of rules (see Figure 6.10). When the sensors and the data acquisition system are connected to the RMD-KBS, this step of data file selection should be replaced by the acquisition of new data from the selected sensors.

As mentioned in Section 6.3, the best indices recommended in the literature, to diagnose each fault have been known through the knowledge acquisition. In the implementation of the MAIN-KBS, such human knowledge regarding the selection of signal analysis methods is encoded as follows: when a hypothesis needs to be proved, a corresponding set of indices is selected and extracted from the data samples. The values of the selected indices are extracted from the selected data files by invoking the corresponding routines which perform the calculations. The way in which the numerical routines are invoked to perform this task, has been illustrated in Section 6.7. The obtained set of indices is used as the pattern vector X of Eq. (3.1) for fault classification and identification through neural networks. The rules and methods that belong to the fourth group shown in Figure 6.10, are designed to perform this task. It may be noted from Chapter 4, that the same set of diagnostic indices is used to identify several different types of machine faults. The index set is a combination of the diagnostic indices that can be used for the identification of more than one type of fault. Consequently, a neural network that is intended to identify and classify fault pattern, is trained to detect more than one type of fault. In accordance with the diagnostic task to be performed, the MAIN-KBS selects a proper set of indices in this step.

In some cases, the SOM networks which perform the multiple-index based trend analysis have been established, and can be used to estimate the remaining life of a machine component. Then the values of the index set to be used for this purpose, are also calculated at this point. Most of the indices in this set are the same as the set used for fault identification, but it may be supplemented by a few additional indices.

Fault Recognition and Condition Determination: After calculating the values of the selected indices, these will be judged by a neural network which has the knowledge in numerical mode about the fault-symptom relationships. According to the posted hypotheses, a SOM network is invoked from the statements of rules. The SOM network will then return the results of fault identification back to the MAIN-KBS module.

Subsequently, the rules in the last group shown in Figure 6.10 will be evaluated to render the final diagnostic results. The rules in the last group are implemented as follows.

1) If there is only one sensor mounted on the machine, the rules will simply declare the results obtained from the neural networks as the final diagnosis result of the current condition of the machine.

2) In the case where two data files corresponding to two different sensors, when evaluated by the neural networks, yield the same results, this result will be declared as the final diagnosis result.

3) If the results deduced from the two different data files are not the same, then both results will be posted on the screen as the result of the diagnosis. This means that at the current instant of time, the symptoms of both faults are observed.

4) In some cases, the symbolic processing module will evaluate some other rules in this last group of Figure 6.10, so as to obtain the final diagnostic results. For example, a symptom of unbalance is that the amplitudes which appear at the rotating frequency and its harmonics increase when the rotating speed increases. The occurrence of this kind of symptoms may not be checked by the neural networks. In certain cases, they are important in distinguishing one fault from another. Hence, the symptoms of machine faults, in the above example, are checked by the symbolic rules at this final stage of diagnosis. A query may be posted to ask the user if the occurrence of a symptom is TRUE or FALSE before the KBS achieves the final diagnostic result.

5) If the diagnostic result at this point is "unknown fault", then the system will return to the beginning of step 2 and a new round of data collection and analysis will start.

CHAPTER 7

PERFORMANCE AND APPLICATIONS OF RMD-KBS

In this chapter, the flow of both the sensory data regarding the vibratory response and the information about the structural and operational parameters of the monitored machine, over the entire diagnostic process is illustrated. The diagnosis of oil whirl and unbalance problems in a compressor unit is considered. The functioning of various modules of the RMD-KBS and the sequence of operations in extracting the diagnostic result are explained in detail. Further, applications of the RMD-KBS to a class of industrial rotating machinery are demonstrated. Vibration measurements from the machine systems with known faults, have been obtained. Diagnosis results are obtained from the RMD-KBS, based on the sensory data to ascertain its performance and reliability.

7.1 Initialization of RMD-KBS

The information flow and the tasks to be performed by each module of the RMD-KBS are shown in Figure 7.1. The SET-UP module performs the acquisition of information for the initialization and refinement of the facts stored in the database. Using the SET-UP module, the user can type in the information, or by selecting one among several options provide the RMD-KBS with the facts about the structural and operational parameters of machine units. Before the RMD-KBS starts monitoring and diagnosing an individual machine system, information about both the machine system and the sensors should be provided. This step is in essence the initialization of the RMD-KBS. Whenever the facts need to be refined, the SET-UP module can be used to perform this task. The initialization of the RMD-KBS and the performance of the SET-UP module are now demonstrated through the case study given below.

The diagnosis of an air compressor that has oil whirl and rotor unbalance problems is considered. Oil whirl is a malfunction that is associated with fluid-film bearings in rotating machinery (Eshleman and Jackson, 1992). It tracks the operating speed and appears at a frequency about 0.35 to 0.47 times the operating speed. The rotor unbalance is a once-per-revolution fault which occurs when the geometric centre and the mass centre of a rotor do not coincide. The selected air compressor unit is a key machine system in a factory of chemical fibres, in the city of Yizheng, Jiangsu province, P. R. China. Several off-line data files, that contain the digitized vibration signal measured from the compressor unit, were provided by the Mechanical Engineering Department of Southeast University, Nanjing, P. R. China.

The RMD-KBS is started through executing the MAIN-KBS module. As shown in Figure 7.2, the first window of the MAIN-KBS module appears on the screen displaying the RMD-KBS Central Panel. There are seven buttons on the Central Panel. By clicking on any one of them using the mouse, the corresponding module will be invoked to perform its tasks for monitoring and diagnosis. The button with the label "Initiate Diagnosis" links to the BROWSER module; "Condition Monitoring" chains to the MONITOR module; "Refine Knowledge" calls the MENTOR module; "Browse Data File" invokes the Notepad program of Windows to show the contents of data files. If the button with the label "Begin Analysis" is selected, a diagnosis will begin. When the button labelled "Set-Up Module" is selected, then the SET-UP module, that is the module for the information acquisition, will be invoked.

The first window of the SET-UP module is shown in Figure 7.3. It can be seen that the window shows the name of the machine system, the location where the machine system is in service, the installation date of the machine, the first service due date, the number of major subsystems, and the reference number assigned to this unit. The above information is stored in an information file named UNITRGST.DBF. The information can be inputted, modified or deleted by the end-user, using the buttons labelled "Replace", "Append" or "Delete". The buttons with labels of "<<", "<", ">" and ">>" are designed to search within the database file to find the first, the previous, the next and the last record respectively.

After finishing the above step of the system set up, by pushing the button with the label "Next Step", another display as shown in Figure 7.4 will appear. This step is designed to define each subsystem (a machine or a major component) of the current machine unit. The compressor unit consists of a motor, two compressors and two gearboxes. Since no sensor had been previously used to monitor the motor and the gearboxes and further, no information about those subsystems is available, the compressor unit is described as a machine system of two compressors in the RMD-KBS. The information about one of the compressors is summarized in Figure 7.4.

More details about the machine components are also acquired in this step. For example, when the bearing type is given as a rolling element bearing, a small window will automatically appear on the screen to inquire about the specifications of the bearing. The windows and screen display appearing at this moment are shown in Figure 7.5. On the small window, the user can input the information about the number of rollers, the pitch diameter, the roller diameter and the bearing contact angle. Then, the SET-UP module will calculate the fundamental train frequency, the ball pass frequencies at the inner and outer race, and the ball spin frequency, using Eqs. A.1.15 to A.1.18 of Appendix A.1. All the above information about the bearing is needed in the monitoring and diagnosis. If the user can not provide all the bearing details but only information on the number of rollers, the characteristic frequencies can still be estimated using Eqs. A.1.19 to A.1.22 (Schiltz, 1989) of Appendix A.1. The above information about each and all of the bearings to be monitored, is stored in a database file named BEARINGS.DBF.

When a subsystem has been defined, the SET-UP module requires information about the mounted sensors. The window and display for the acquisition of information about the sensors are both shown in Figure 7.6. If the end-user does not know the definition of the sensor locations in the RMD-KBS, he can ask for help from the RMD-KBS, by selecting the button labelled "Help". The graph shown in Figure 7.7 which defines the likely locations of the sensors on a machine system, will appear on a smaller window.

After the acquisition of all the information mentioned above, the SET-UP module reasons out the values of the settings for signal processing and feature extraction, based on both the information available up to this point, and the knowledge already encoded in it. Further, it requires the threshold values of the monitoring indices. Another display as shown in Figure 7.8 appears on the screen at this moment. All the values of the settings as well as, the thresholds displayed on the screen corresponding to a single sensor can be modified by the user. Some of the settings for signal processing, regarding the sensor in horizontal direction at the free-end of the air compressor, are listed below as determined by the SET-UP module.

Data file name:	D:\data\u6p2\AH*
Data length:	1,024
Sampling frequency:	5,120 (samples/sec)
Analysis frequency:	2,500 (Hz)
Rotating frequency:	390 (Hz)
Natural frequency:	245.8 (Hz)
The 2nd harmonic:	780 (Hz)
The 3rd harmonic:	1,170 (Hz)
42% of rotating frequency:	163.8 (Hz)

At the end of this step, all the values of both the settings and the thresholds are stored into an information file named SETTING.DBF, which will be accessed during the diagnostic reasoning. The SET-UP module actually provides the users with a tool to access the information files of the RMD-KBS. Whenever a new machine system is to be monitored by the RMD-KBS, all the above initialization steps require to be performed. The existing information stored in the RMD-KBS may be modified as appropriate.

In order to provide the end-users with a utility to view and manage the data files, the Notepad (a Windows text file editor) is employed. If the button on the Central Panel of MAIN-KBS labelled "Browse Data File" is selected, the Notepad will be invoked. As shown in Figure 7.9, the Notepad presents the values of data sample stored in a data file. It can also be used to manage the data files.

7.2 Performance of Monitoring and Diagnosis Tasks

The tasks of monitoring the current condition of a machine system and diagnosing incipient faults are performed by the MONITOR and MAIN-KBS modules, respectively. The MONITOR module provides the facilities needed to display the current values of the monitoring indices, signal, spectrum, orbits etc. Before invoking the MONITOR module from the Central Panel of the MAIN-KBS, a machine subsystem must be selected as the subject to monitor. By selecting the button on the Central Panel labelled "Initiate Diagnosis", the BROWSER module is invoked. The window of the BROWSER module is shown in Figure 7.10, in which the information about a monitored machine system is presented. If the button labelled "View The Machine" is selected, a drawing of the machine will appear on a small window. As an example, Figure 7.11 shows a gearbox. When a machine is selected, by the button labelled "Central Panel", the system goes back to the MAIN-KBS module. At this time, since a machine has been selected as the unit to be monitored or diagnosed, the Central Panel shows the selected machine on the display, as illustrated in Figure 7.12. Simultaneously, the information about the currently selected machine is passed to the MAIN-KBS module. Further, the name of the machine system, the name of the selected subsystem, sensor types and locations, names of data files, and so on, can be transferred to the MONITOR module, by selecting the button labelled "Condition Monitoring".

The first window of the MONITOR module is illustrated in Figure 7.13, and it displays the current values of several diagnostic indices using a bar chart. These values are extracted from a data file named d:\data\u6p1\AH1, which is the currently-selected data file and contains the data samples from a sensor in the horizontal direction. There are several buttons in the display as shown in Figure 7.13, that are designed for the user to select a sensor and a data file that contains the corresponding data samples. When a selection is made, the values of the corresponding set of diagnostic indices are automatically calculated and shown in the bar chart.

In Figure 7.13, it can be seen that the Peak-to-Peak value (PP) and the Maximum x value ($|x|$) are high and both of them attain a value of 0.6. Values of several other indices such as the Absolute Mean value (AX), the Root Mean Square value (RMS), and the magnitude at the machine rotational frequency ($F1$), are also relatively high. The magnitudes in the FFT spectrum at the frequency around 42% of the rotational frequency, i.e. Fh, are significant. From the above observations, an experienced diagnostician can arrive at an impression that the current condition of the compressor is not "good". Further, by selecting the button labelled "Signal/Spectrum", a C++ program (ANALYSIS.EXE) will be invoked to display both the data samples in the file d:\data\u6p1\AH1 and the corresponding spectrum. Then, a window with the graphical display appears as illustrated in Figure 7.14. The time domain vibration signal is shown in the upper plot, while the spectrum that corresponds to the frequency domain components of the signal is shown in the lower plot. The related information about the machine and sensor is also presented. It can be seen in Figure 7.14 that the time domain signal is typical of oil whirl defects. In the spectrum, the second highest peak at the frequency that is a little bit less than 0.5 times the rotational frequency, is the prime symptom of oil whirl. It can also be observed that the magnitude of the amplitude at the rotational frequency is also high. A diagnostician can easily conclude that an oil whirl problem is present, and also, the rotor is not well balanced. Furthermore, the MONITOR module observes that (i) there are two sensors on this compressor that measure simultaneously the vibration signal in both the horizontal and vertical directions, (ii) the two sensors are displacement sensors (proximity eddy current transducers) which measure the relative motion of the rotor with respect to the bearing. Hence, the MONITOR module can plot two sets of signal samples from the two sensors on the x-y plane. This plot is known as the rotor-centre orbit, which shows the actual movement of the centre line of the shaft with respect to the bearing. This orbit is also called the Lissajous pattern. The orbits constituted by the signal in the current file and the signal in the corresponding data file, named d:\data\u6p1\AV1, that comes from the other sensor mounted in the vertical direction, are depicted in the Figure 7.15. In this figure, two ellipses, one bigger and one smaller, are present. This pattern of orbits is typically observed when oil whirl occurs (Eshleman and Jackson, 1992). An experienced diagnostician can tell, based on an examination of the shape of the orbits, that oil whirl is present.

The above illustrations of the signal, spectrum and orbits can now be compared with other monitoring results. The data samples stored in two other files, d:\data\u6p2\AH3 and d:\data\u6p2\AV3, correspond to the same sensors, but they were measured at the time when oil whirl had not occurred. Figure 7.16 illustrates the data samples in d:\data\u6p2\AH3 and the corresponding spectrum. It can be seen that the wave form of the time domain signal is quite close to a pure sine wave. In the corresponding spectrum, there is no significant peak at the 42% of the rotating frequency. Further, the corresponding orbits of the rotor, as displayed in Figure 7.17, have the shape of a repeated plot of a single ellipse.

The above cases can now be analyzed by the MAIN-KBS module based on the diagnostic knowledge it contains. By selecting the button labelled "Begin Analysis" on the Central Panel of the MAIN-KBS module, the diagnosis process will begin. At this point, the subsystem of a machine system must have been selected. Then, a sensor and a data file corresponding to the machine can be selected by the user. For example, the input data is now selected as the data file d:\data\u6p1\AH1, that contains the data samples from the compressor when the oil whirl problem occurred. By selecting the button labelled "Start Diagnosis"on the current display shown in Figure 7.18, the MAIN-KBS module will generate the hypotheses based on information about this machine and the sensors, and the input data file. As shown in Figure 7.18, two hypotheses, namely "Rotor Unbalance" and "Journal Bearing Oil Whirl", are generated by the MAIN-KBS module. Further, by reasoning through the rules, the MAIN-KBS module will find the other data file, i.e. d:\data\u6p1\AV1, which is related to the current diagnosis. The KBS will calculate the values of certain indices from both the data files according to the hypotheses, pass the values of the indices to the neural networks, and then output the final diagnostic results on the screen. For the selected compressor of the unit of interest, it is confirmed that both the two hypotheses, i.e. "Rotor Unbalance" and "Journal Bearing Oil Whirl", are true. The result of the diagnostics is shown in Figure 7.18.

When the data file, d:\data\u6p2\AH3, is selected for a diagnosis, only one hypothesis, that is "Rotor Unbalance", is generated by the MAIN-KBS module, as shown in Figure 7.19. The diagnosis of this input file and another file named d:\data\u6p2\AV3 yields a diagnostic result, that confirms the hypothesis "Rotor Unbalance". Some other examples of diagnosis performed by the RMD-KBS will be demonstrated later in this chapter.

7.3 Training the Neural Networks

The training of the neural networks is performed by the MENTOR module of the RMD-KBS. The trained neural networks are then used by the MAIN-KBS module to perform the diagnosis of machine faults. The information flow in both the training process and the diagnostic process using neural networks is shown in Figure 7.20.

By selecting the button labelled "Refine Knowledge" on the Central Panel of the MAIN-KBS module, the knowledge-base of the MENTOR module is loaded onto the LEVEL5 environment and then it begins to function. Its window and the first display are shown in Figure 7.21. The training of a particular type of neural network begins by selecting the corresponding button. For example, consider that a SOM network is to be trained for machine fault classification and identification. The "SOM" button can be selected, and a small window appears at this time to query for some specifications. The display on the screen at this moment is illustrated in Figure 7.22. The query for user input asks the name of the data file, the number of samples in the data file, the number of nodes and the number of times the set of training data will be repeatedly presented to the network. When the above information has been provided to the MENTOR module, by selecting the button labelled "Done", the corresponding C++ program is invoked to train the network. The data and the initial positions of the network nodes are both shown on the window, that is illustrated in Figure 7.23. By selecting the option named "Train" from the drop-down window of the option named "View", the training process begins and this will be performed in a few seconds. After the training, the nodes of the SOM network are moved to their final locations, as shown in Figure 7.24. If the user wants to know the weight values of the network nodes, the option named "Weight" from the drop-down window of the option named "View" must be selected, and then the weight values of the just-trained SOM network are presented on the screen as illustrated in Figure 7.25. With similar procedures easy and straight forward, the user can train the BP neural networks, or the SOM networks of multiple-index based trend analysis. Correspondingly, the MENTOR module will display windows as illustrated in Figures 7.26, 7.27 and 7.28.

After the training is completed, the weights of the trained neural network will be saved into a database file. When the MAIN-KBS module diagnoses machine faults, it will call the corresponding C++ routines that can read the values of the weights and use them to evaluate the current indices to determine machine condition.

7.4 Applications to Industrial Machine Systems

There are more than 200 data files in the database of the RMD-KBS. Three examples of rotating machinery monitoring and diagnosis are given in this section in order to demonstrate the performance of the RMD-KBS.

Example 1: The specifications of a sump pump and motor unit are stored in the RMD-KBS in a set of database files. The vibration signatures were recorded from the motor in service, that had a rotor unbalance problem (El-Karmalawy, 1993). These signatures are representative of the vibratory behaviour of the system at both smooth operation and unbalance stages. The physical characteristics of both the motor and the pump are presented in Table 7.1 given below. The structural and operational parameters of the motor are given in Figure 7.29.

Two velocity sensors, with a sensitivity of 50 mV/IPS, were mounted on the motor bearing housing, in both the vertical and horizontal directions to pick up the generated vibration signals. The signals from the two pick-ups were recorded on a two-channel tape recorder, and were digitized via a high speed analogue-to-digital converter, using a sampling frequency of 512 Hz. The data samples are stored in 12 data files of the RMD-KBS.

Table 7.1 Motor and pump specifications

Motor			Pump	
Service	HP	RPM	Capacity	Type
Sump Pump Motor	100	1,770	2,200 GPM	Vertical

Figure 7.30 presents the hypotheses and the diagnostic results achieved by the RMD-KBS, by selecting the data file named d:\data\u1p1\UH3. This data file contains the data samples that were obtained from the sensor in the horizontal direction. In the present diagnosis, another data file named d:\data\u1p1\UH3 obtained from the sensor in the vertical direction is also used. The condition of the motor is diagnosed and further, it was noted that rotor unbalance was present. Figure 7.31 shows the vibration signal obtained from the motor and the corresponding spectrum. In the spectrum, the second highest peak is at a frequency about 42% of the rotor rotational frequency. The RMD-KBS recognized this symptom, which is the prime symptom of oil whirl. Since the type of the bearing of this motor is a rolling element bearing, oil whirl cannot be a malfunction of the rotor. Hence, in Figure 7.30 it can be seen that the index *Fh* does not show a rotor problem in the motor. Further, there is no fault that has been detected from the vibration signal which was measured in the vertical direction.

Example 2: Vibration data were acquired from a type 308E ball-bearing that has 8 rolling elements. The bearing was rotated at 1,470 rpm and loaded with a circumferentially-symmetric radial force of 20.8 kN, in a test machine at a bearing manufacturing plant. An accelerometer was mounted on the housing and its output was linked to a computer-based monitoring system. A sampling frequency of 5,000 samples per second was used and the digitized raw data were stored in more than 100 data files. The information about the bearing test machine stored in the RMD-KBS is shown in Figure 7.32.

As mentioned in Chapter 4, several different types of bearing faults have been identified through the previous analysis of the data, such as pitting problem and the inner ring of the bearing that slid on the shaft. The diagnostic result yielded by the RMD-KBS is demonstrated in Figure 7.33. Two hypotheses, namely "pitting problem" and "bearing wear problem", have been posted. The bearing under test is a faulty bearing with pitting defects in its rolling elements. Since the RMD-KBS contains a SOM network with 8-units, which has been trained previously, for the trend analysis of the bearing service life time, it can perform the multiple-index based trend analysis on the input data. The diagnostic result is posted in the display that is demonstrated in Figure 7.33.

Example 3: Vibration signals from a boiler feed pump in service that had a misalignment problem, were collected. The pump specifications are given in Table 7.2 below.

Two velocity sensors were mounted on the pump bearing housing, to measure vibration in both the horizontal and vertical directions. The signals from the pick-up were recorded on a two-channel recorder, and were digitized via a high speed analogue-to-digital converter, using a sampling frequency of 1,024 Hz. The digital data was stored in 16 data files, with each file having 2,048 data points. Figure 7.34 shows the display of the BROWSER module which presents information about the pump. Since information about other subsystems of the boiler feed pump system is not available, the pump is defined as the only subsystem of the machine unit.

Table 7.2 Pump specifications

Service	Manufacturer	Size	Capacity	Rotating Speed
Boiler Feed Pump	INGE	8x4x11 in.	1,000 GPM	3,600 rpm

In this example, the file that contains the data samples obtained from the sensor in the horizontal direction, is selected for the diagnosis. Since no obvious symptom of any fault from among the list of faults contained in the RMD-KBS has been found, the hypothesis named "Fault Free ?" has been posted as shown in Figure 7.35. While using the neural network to evaluate the indices, that are in turn obtained from both the selected data file and the file containing the data from the sensor in the vertical direction, however, two types of faults have been detected. They are unbalance and misalignment. In the diagnostic results posted on the screen (see Figure 7.35), that no fault has been identified from the data which had been obtained from the horizontal sensor, is also declared. The data samples that were measured in the vertical direction, and the corresponding spectrum are both presented in Figure 7.36. In the spectrum, it can be seen that the magnitudes at both the pump rotational frequency and the second harmonic are high. This observation confirms the diagnostic results yielded by the RMD-KBS.

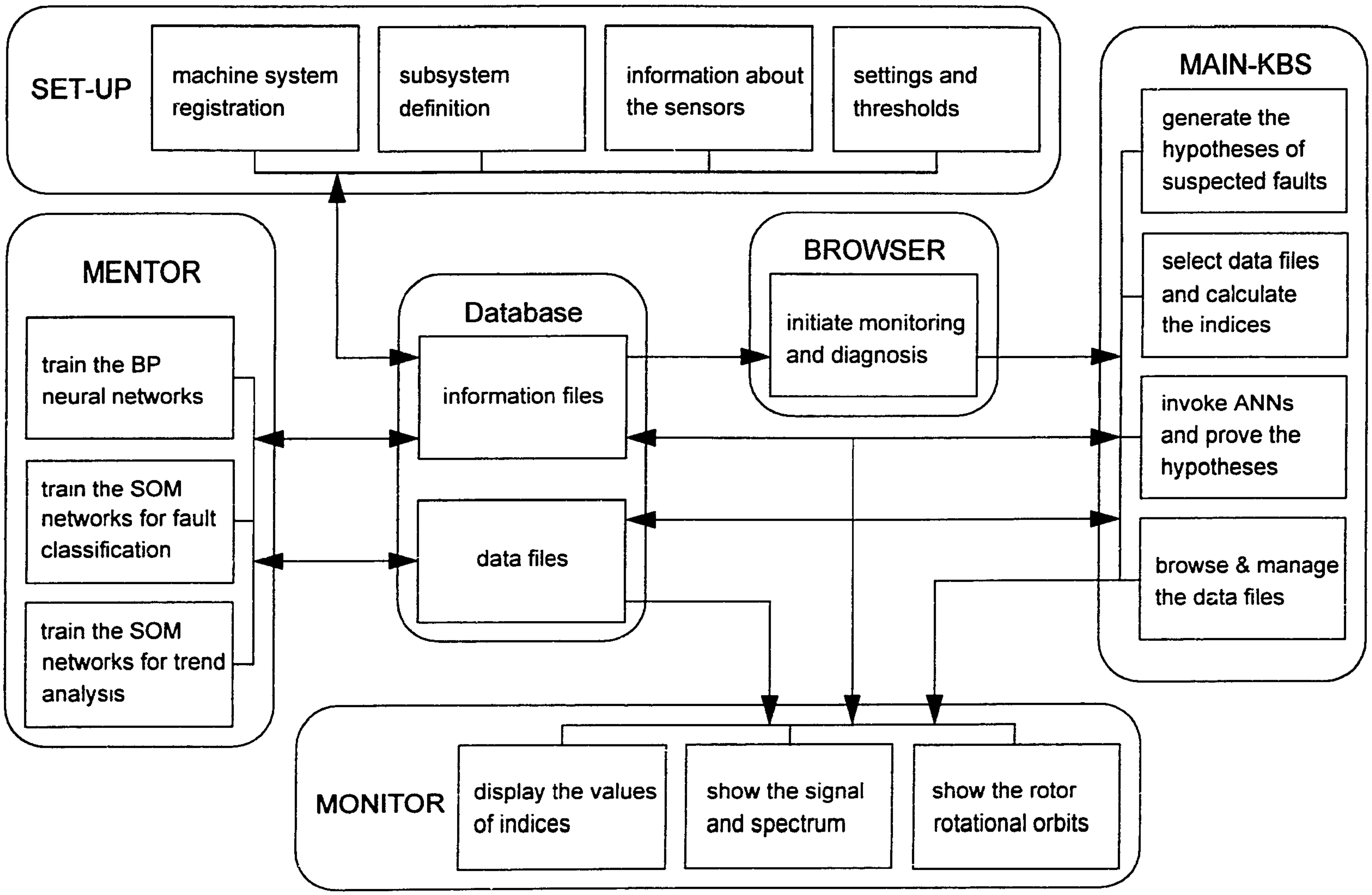


Fig. 7.1 The information flow within the RMD-KBS.

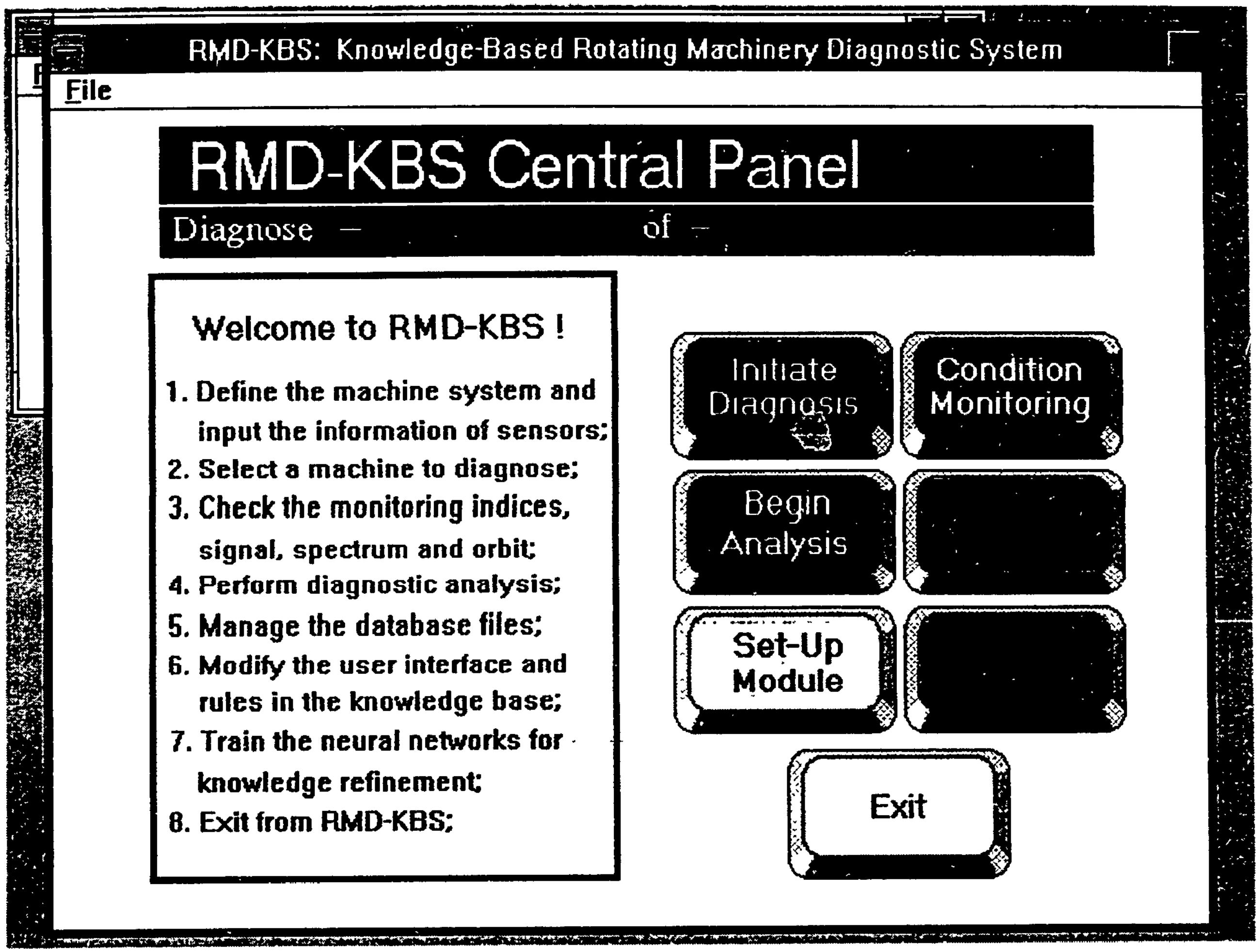


Fig 7.2 The Central Panel of RMD-KBS.

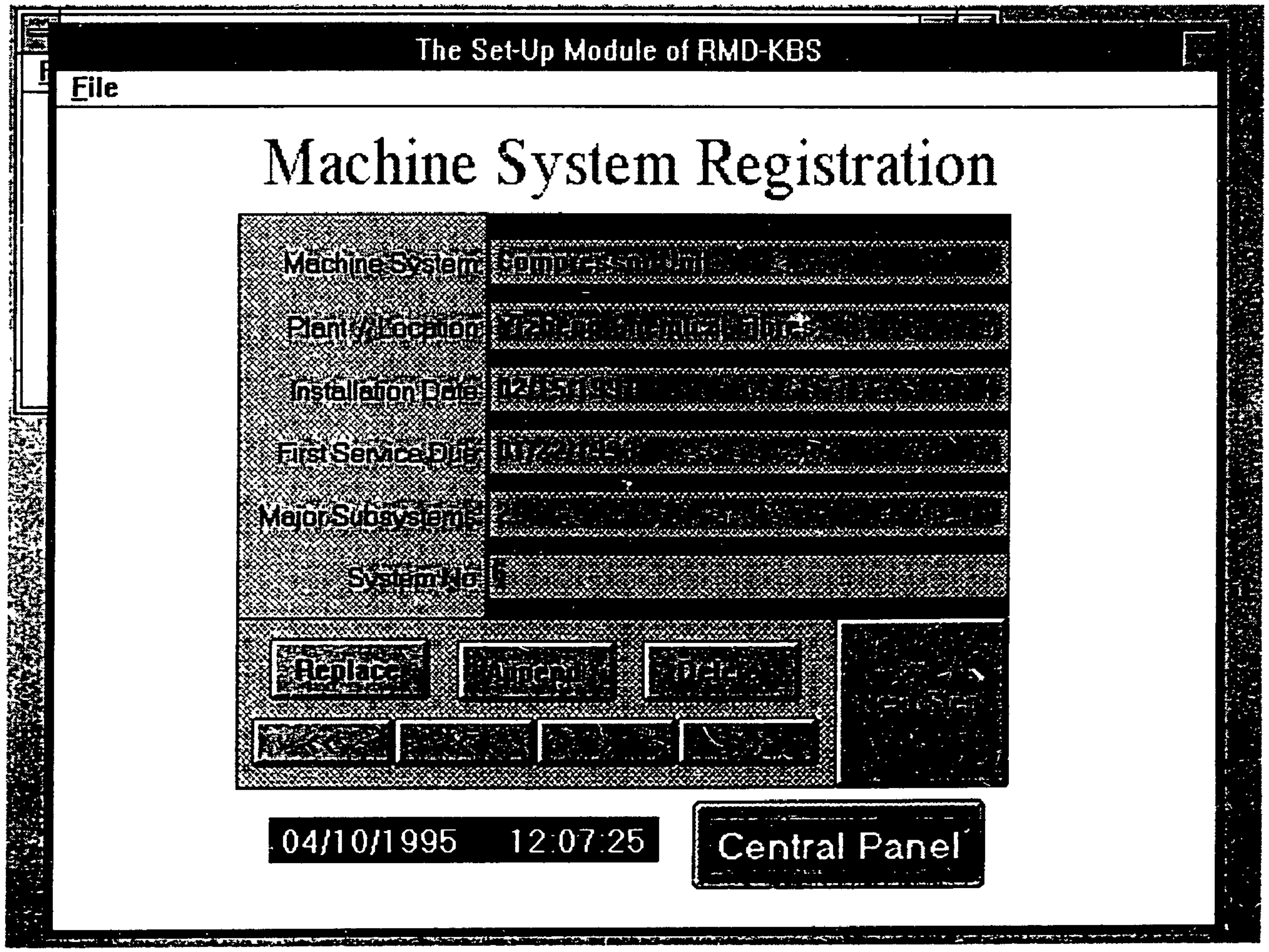


Fig 7.3 The window of SET-UP module for machine system registration.

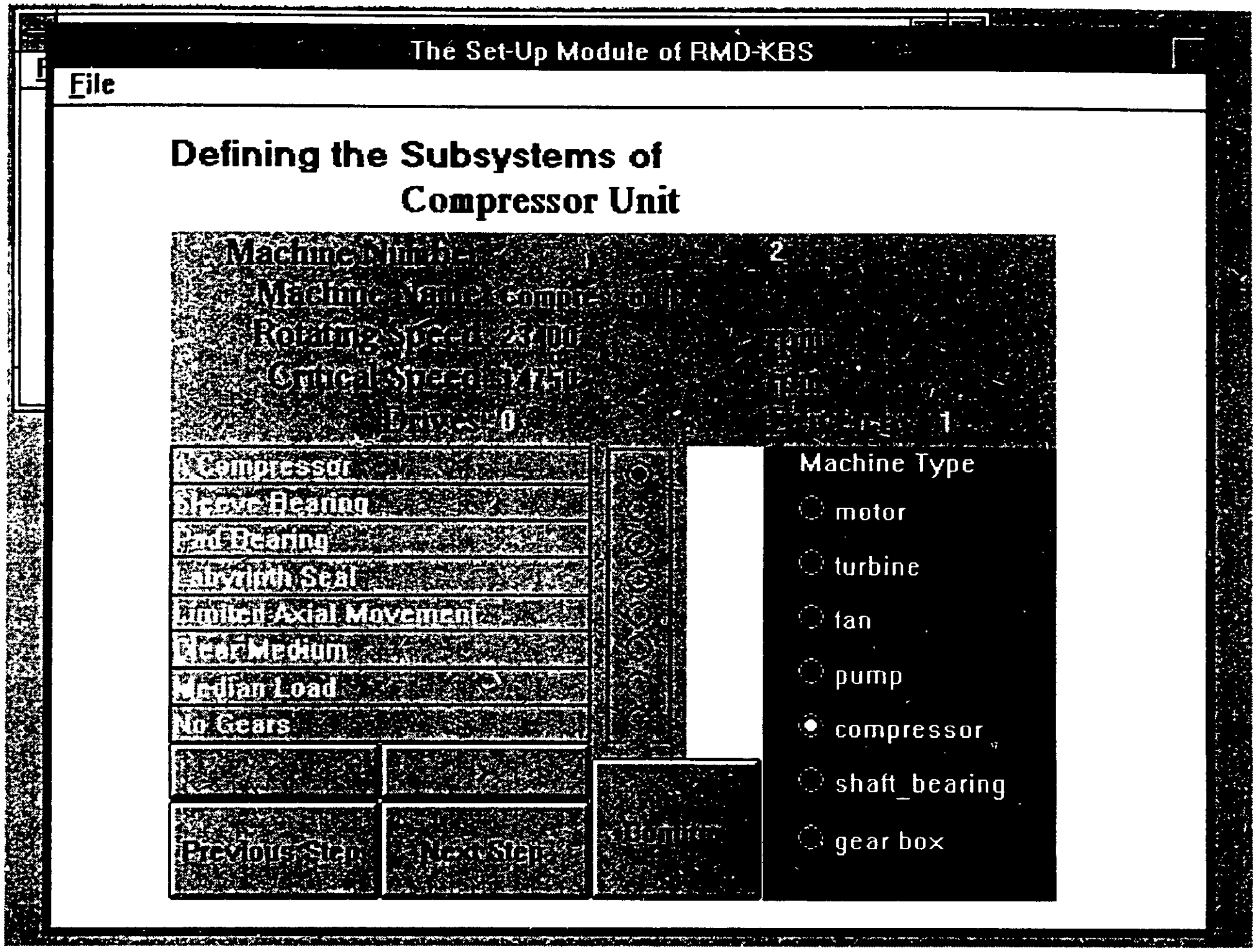


Fig 7.4 The window of SET-UP module for defining machine subsystems.

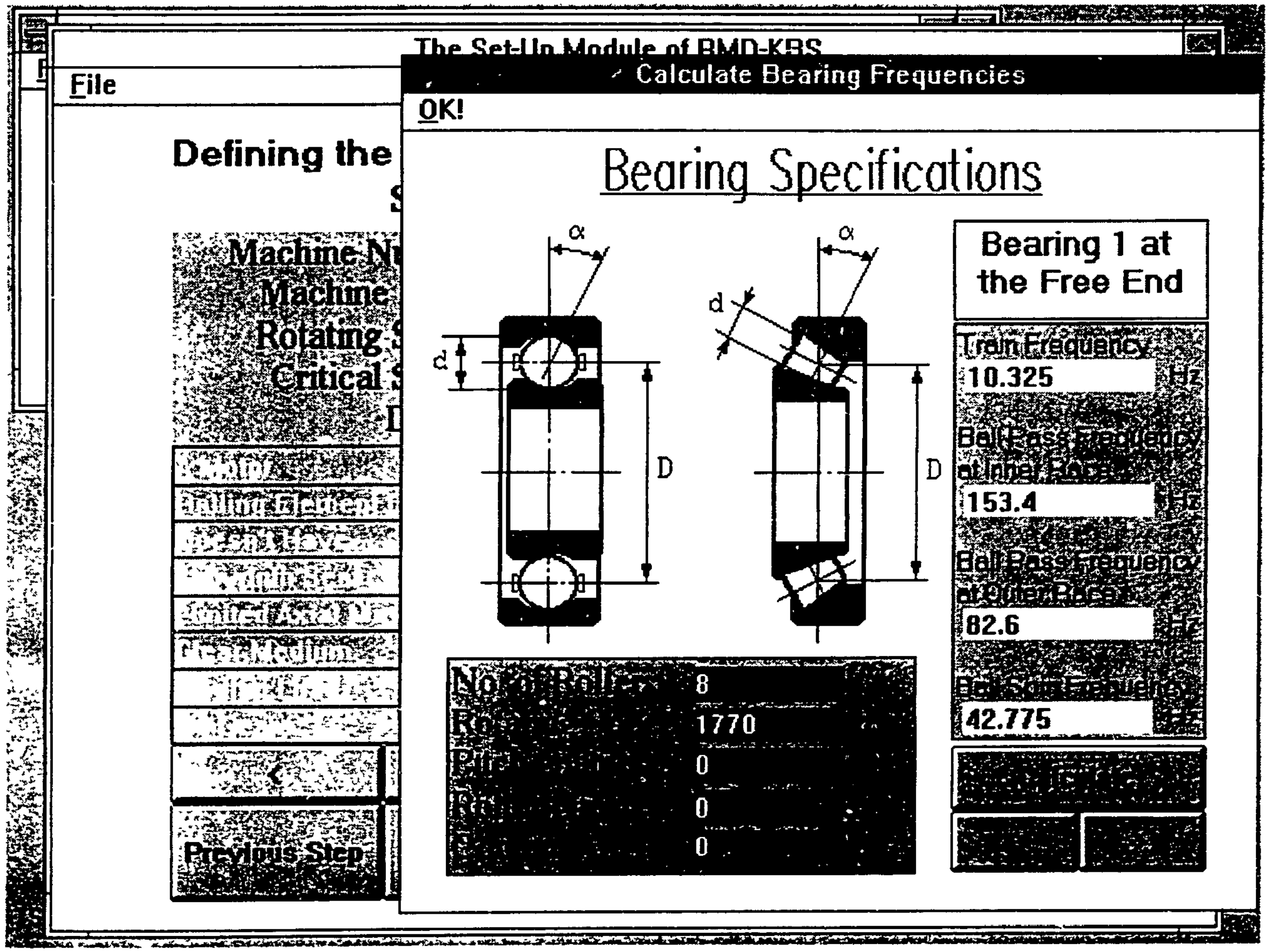


Fig. 7.5 Acquisition of the bearing specifications using SET-UP module.

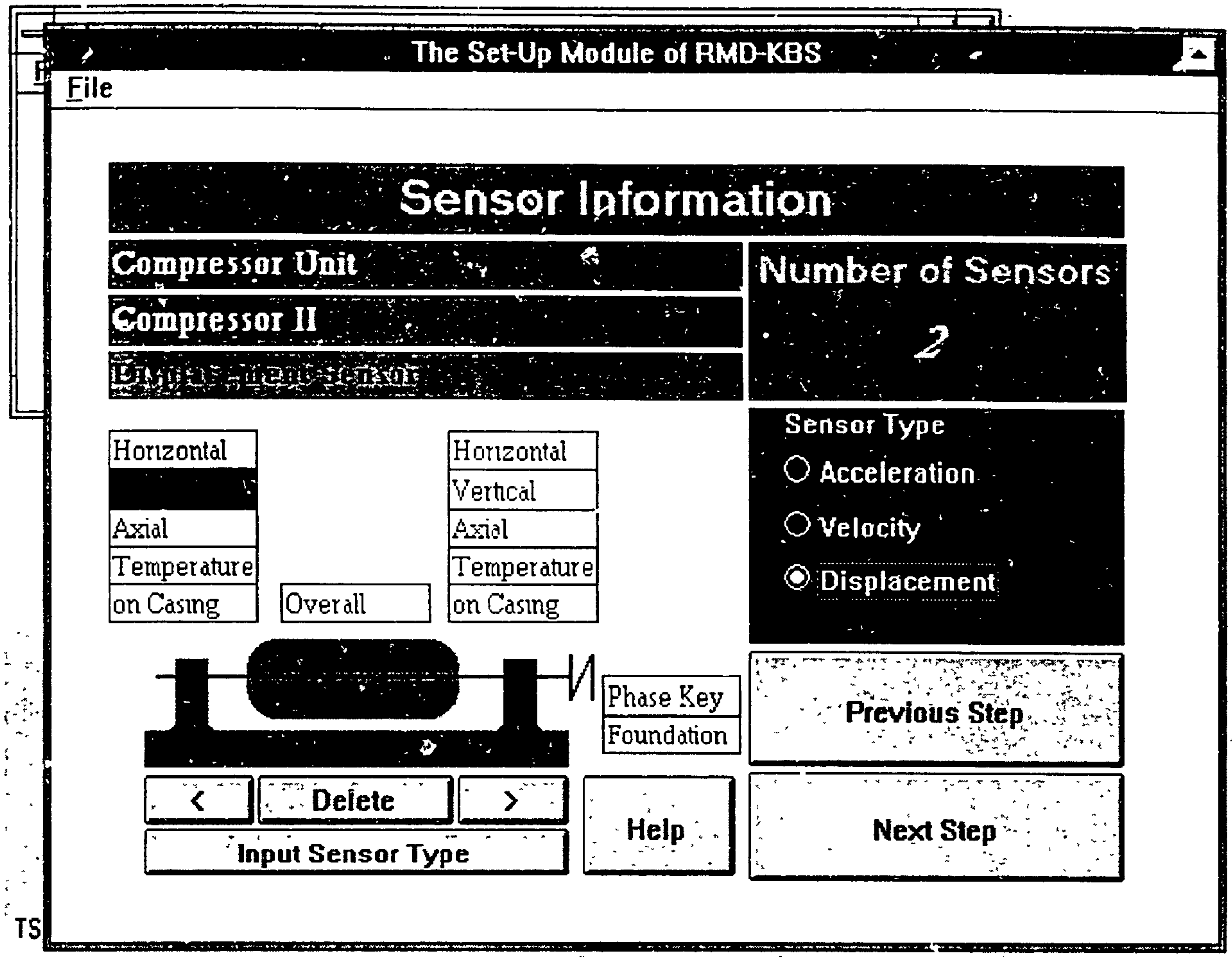


Fig. 7.6 Acquisition of the information about sensors using SET-UP module.

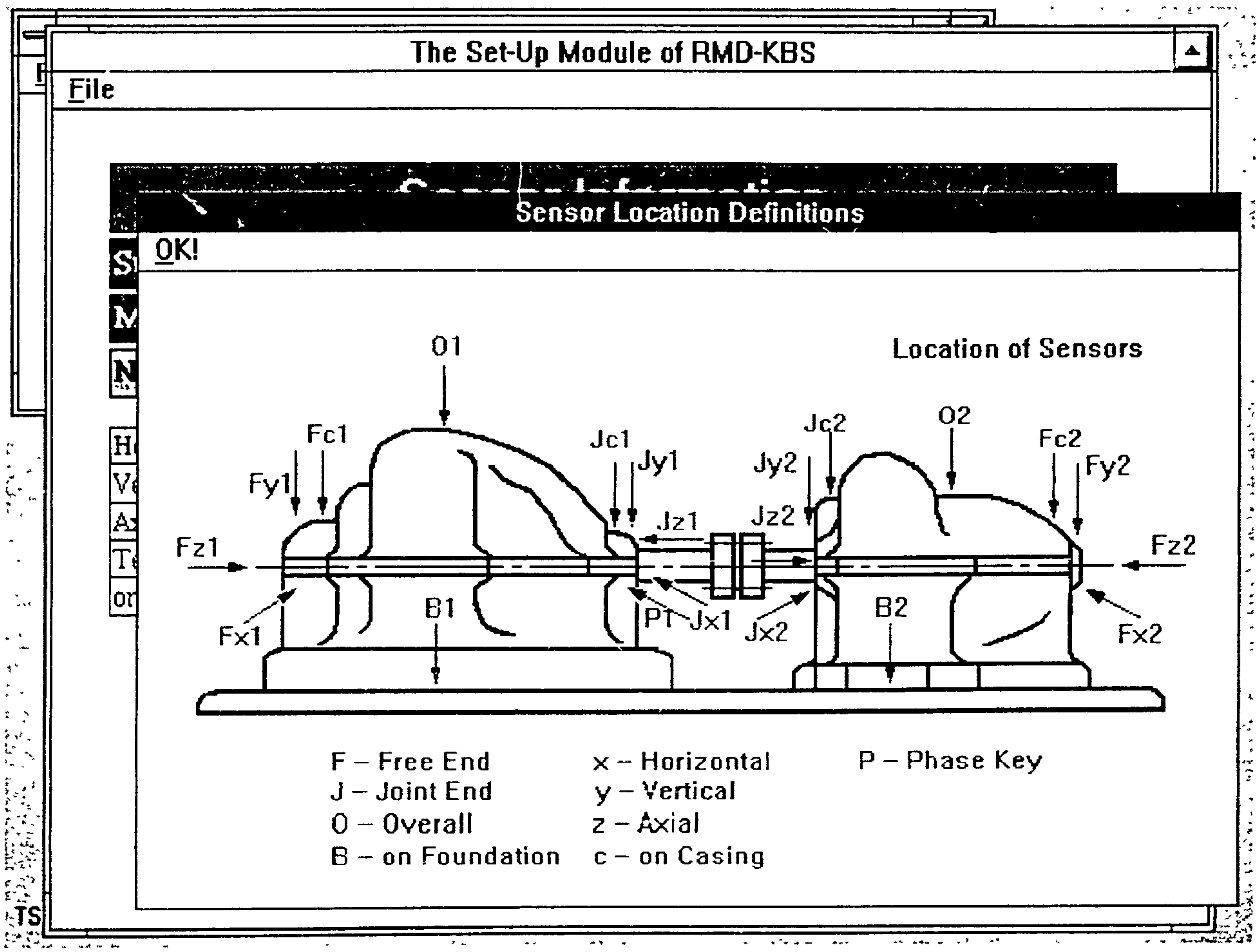


Fig. 7.7 The definition of sensor locations for helping the information acquisition.

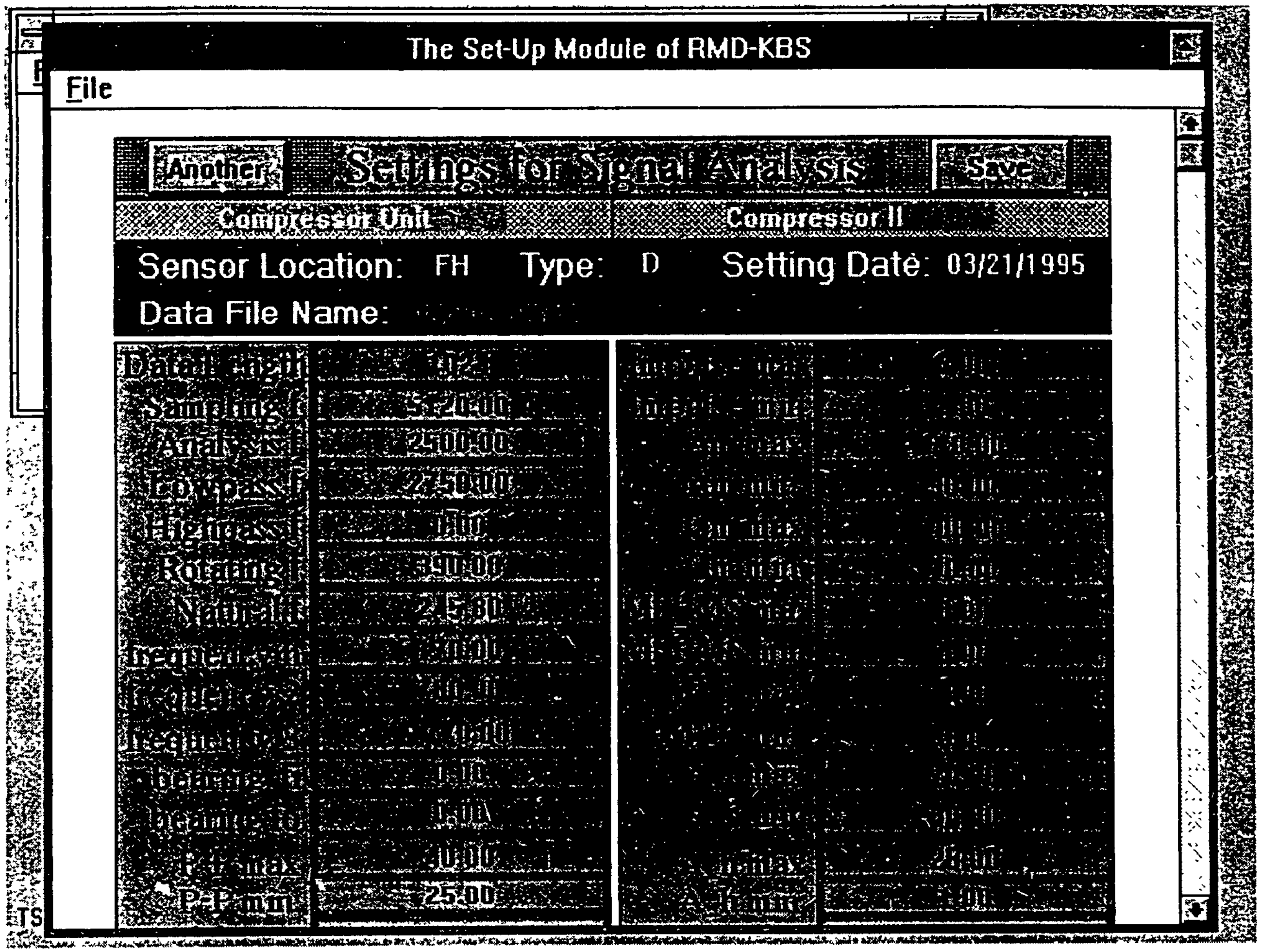


Fig. 7.8 List of the settings for signal processing and the threshold values of indices.

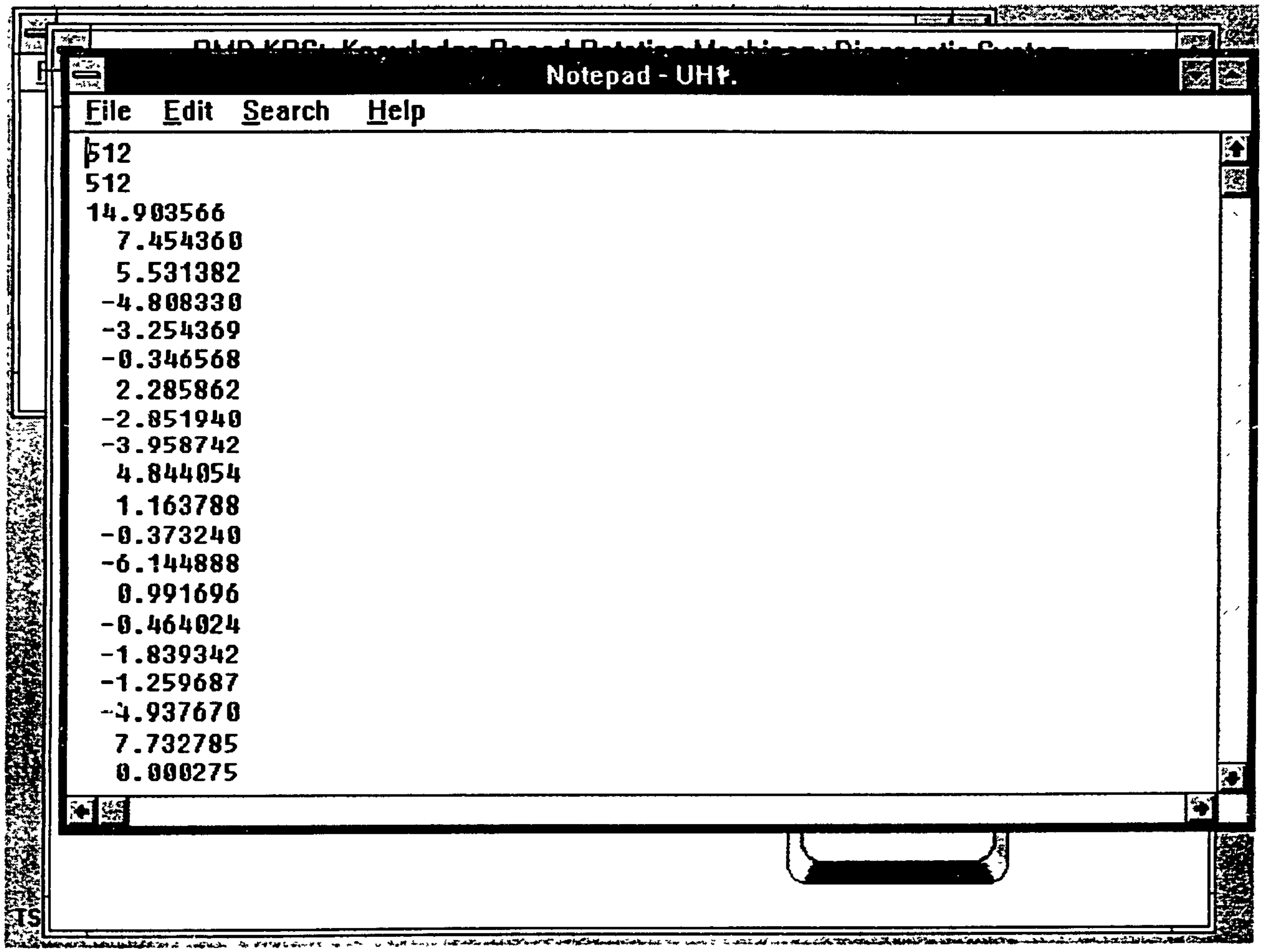


Fig. 7.9 The window for browsing and managing the data files.

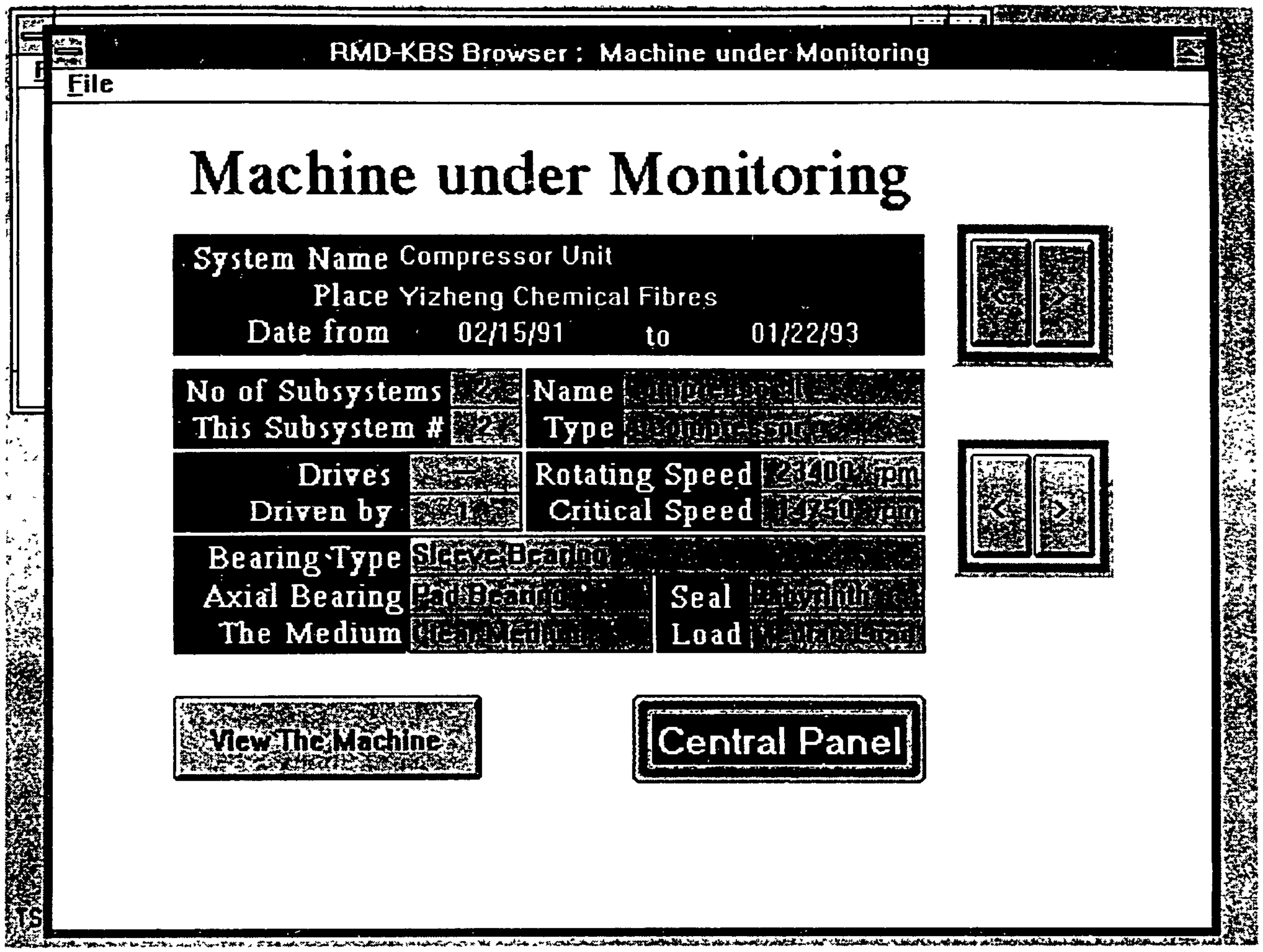


Fig. 7.10 The BROWSER module for initiating the monitoring and diagnosis.

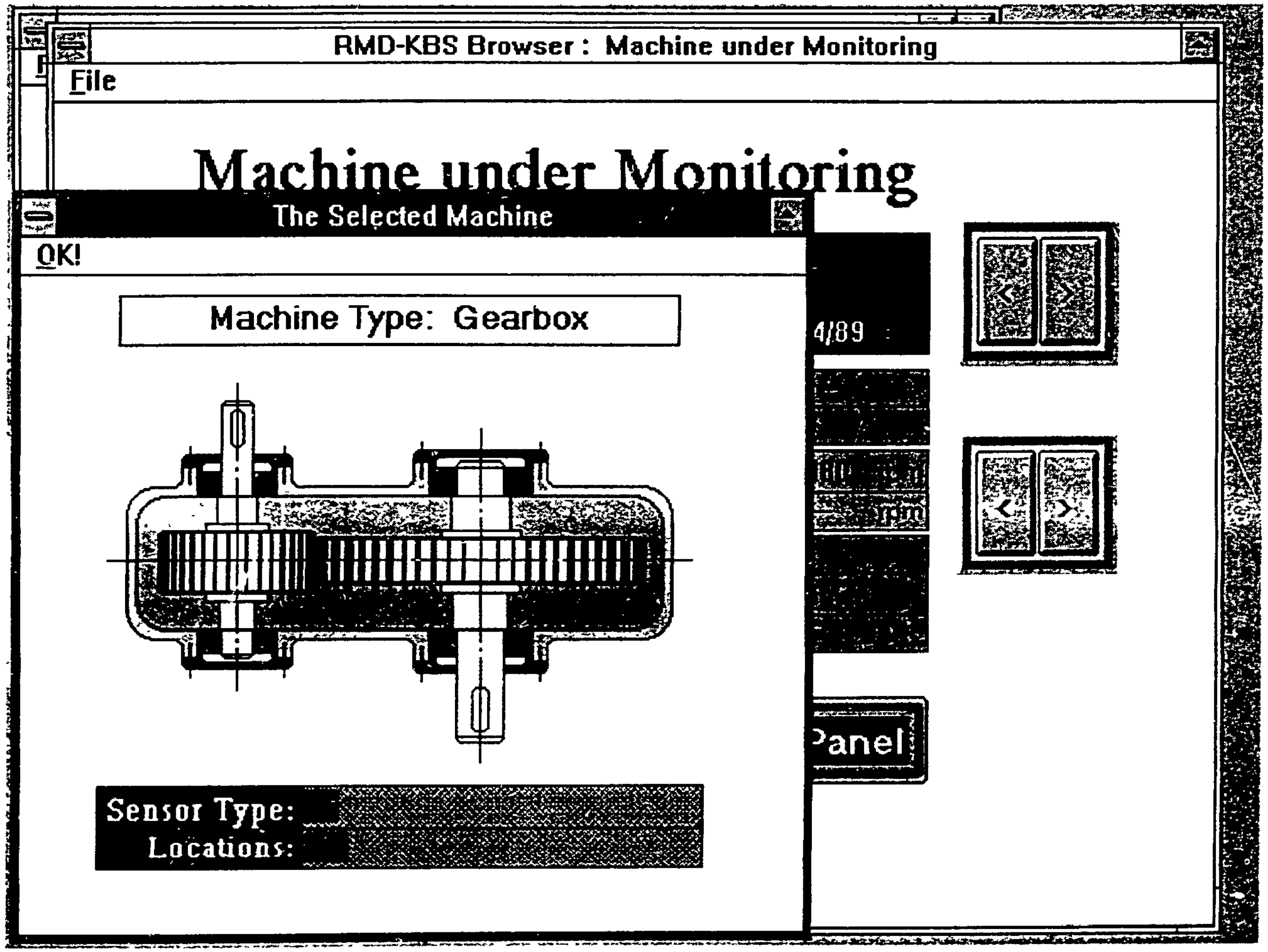


Fig. 7.11 A graph presenting the machine system to be diagnosed.

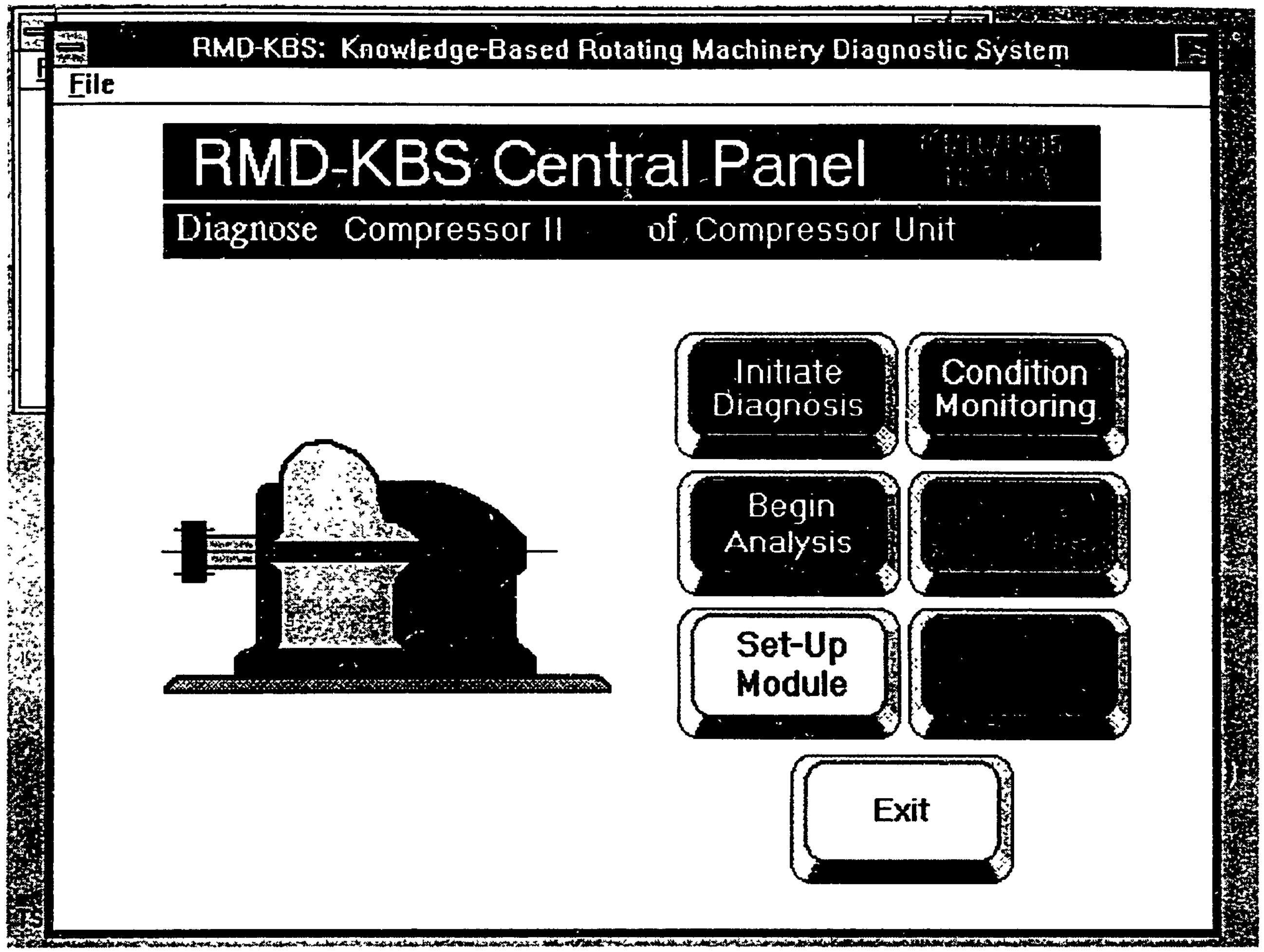


Fig. 7.12 The display of Central Panel when a machine has been selected.

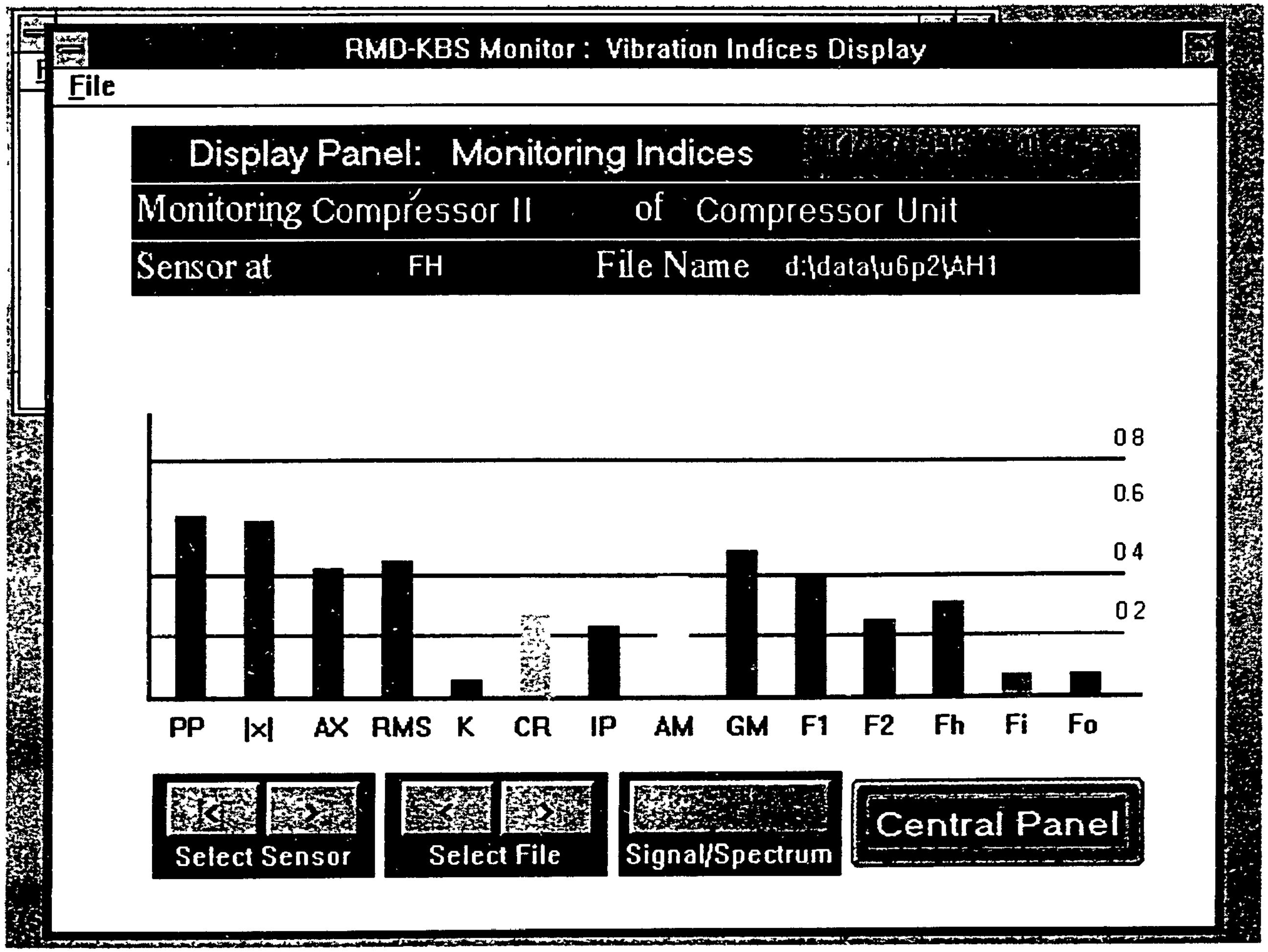


Fig. 7.13 The graphical display of the current values of the monitoring indices.

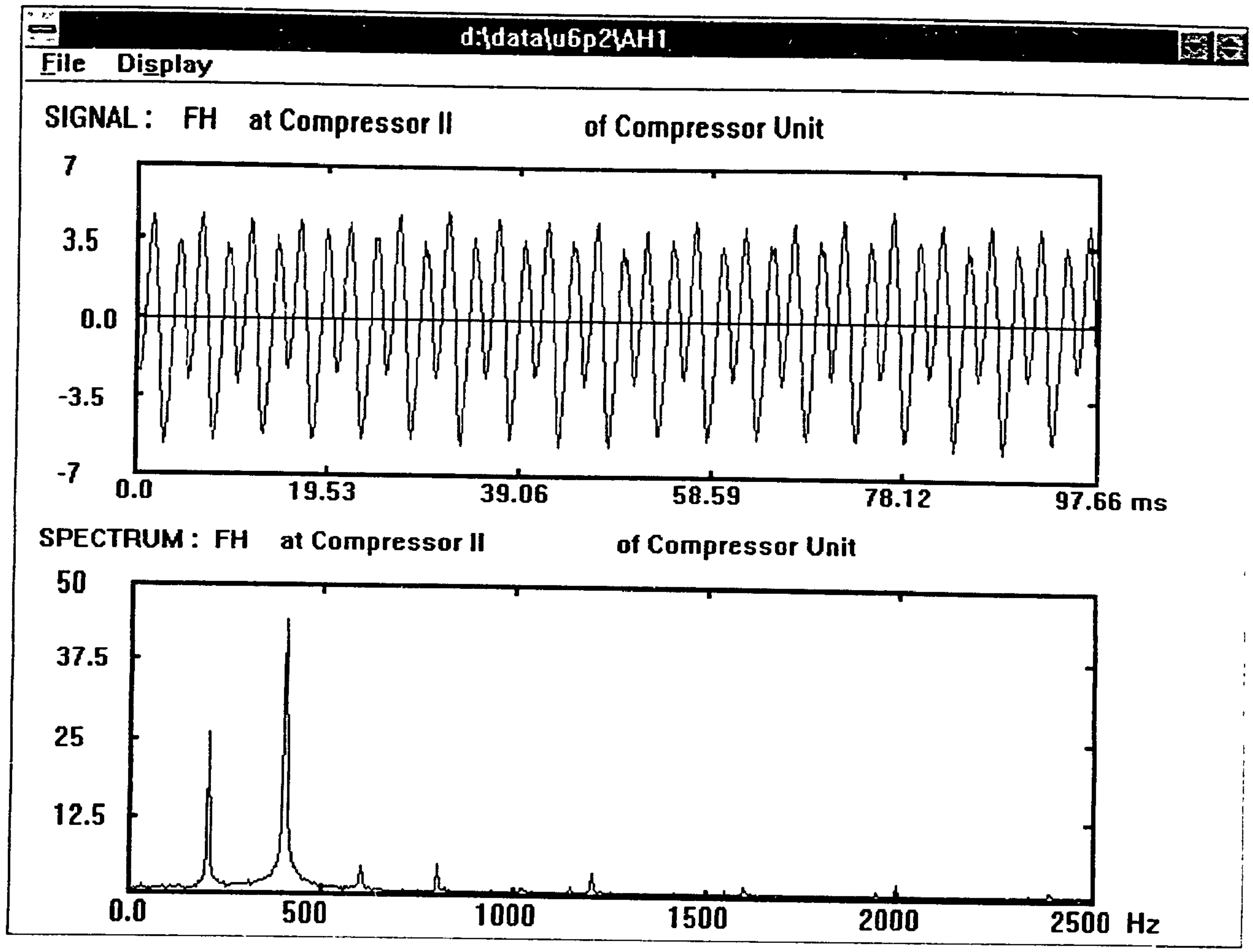


Fig. 7.14 The display of MONITOR module that shows the data samples and spectrum.

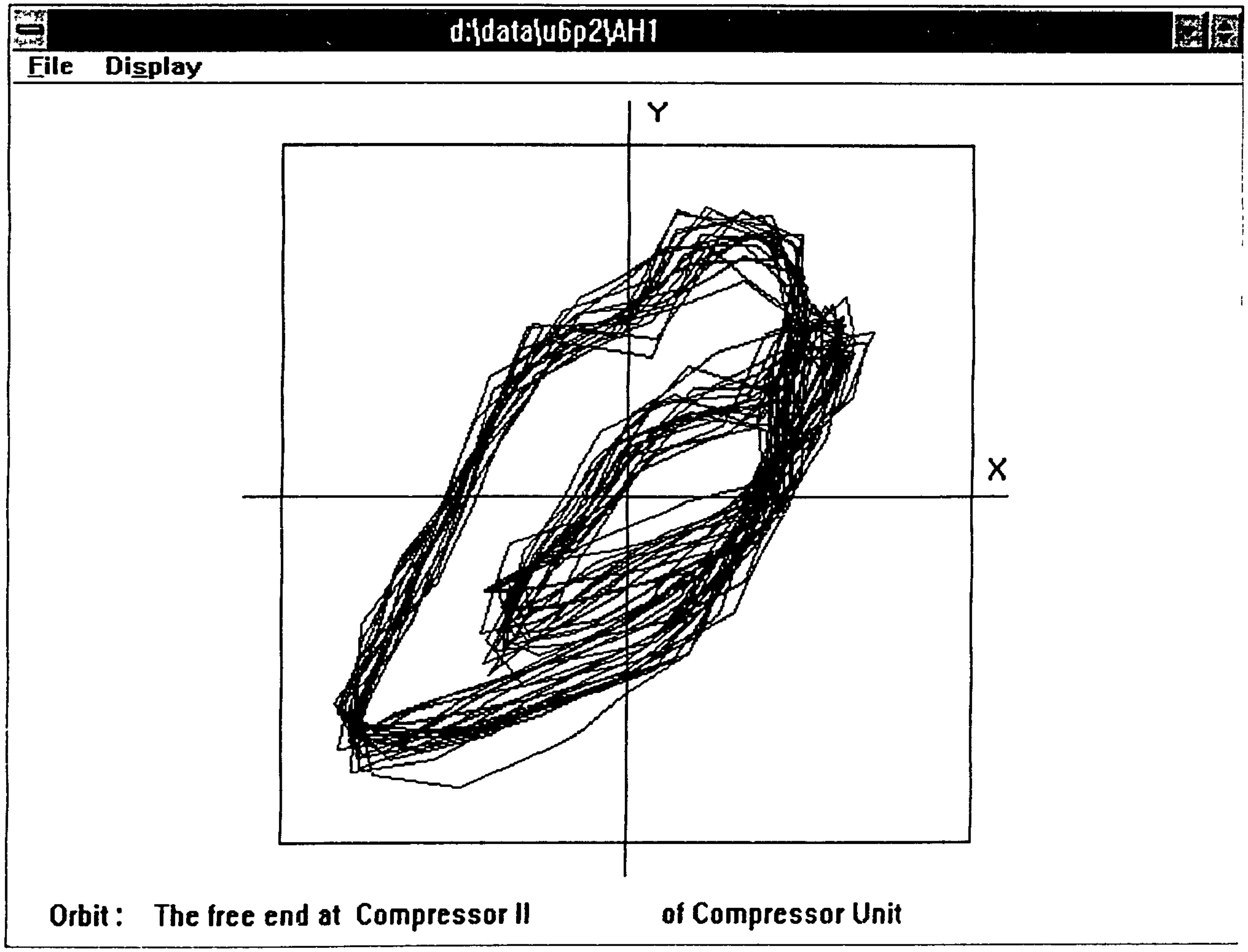


Fig. 7.15 The display of MONITOR module that shows the orbit of the centre of rotor shaft.

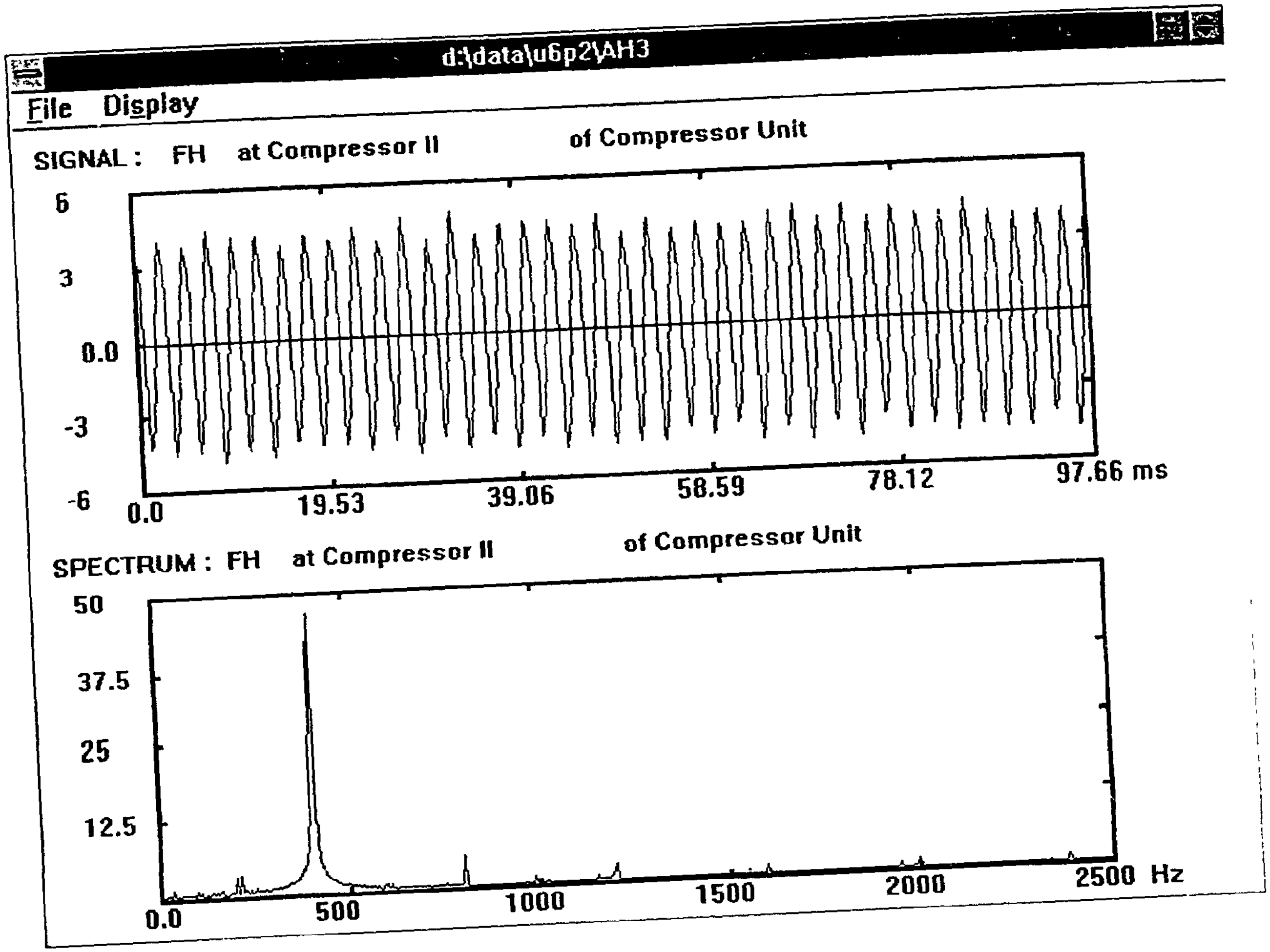


Fig. 7.16 The display of the data samples and spectrum from the file AH3.

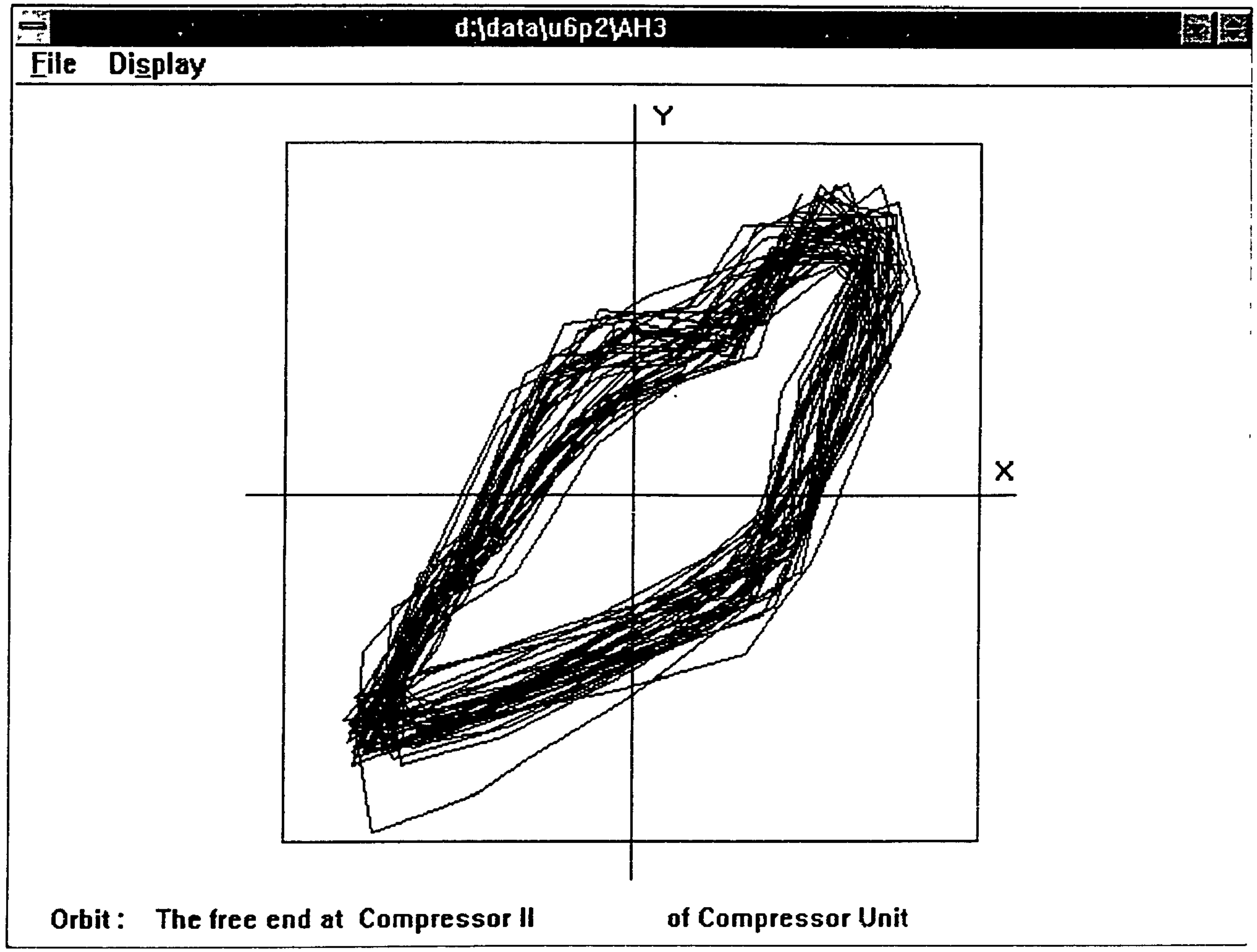


Fig. 7.17 The display of the orbit from files AH3 and AV3 using MONITOR module.

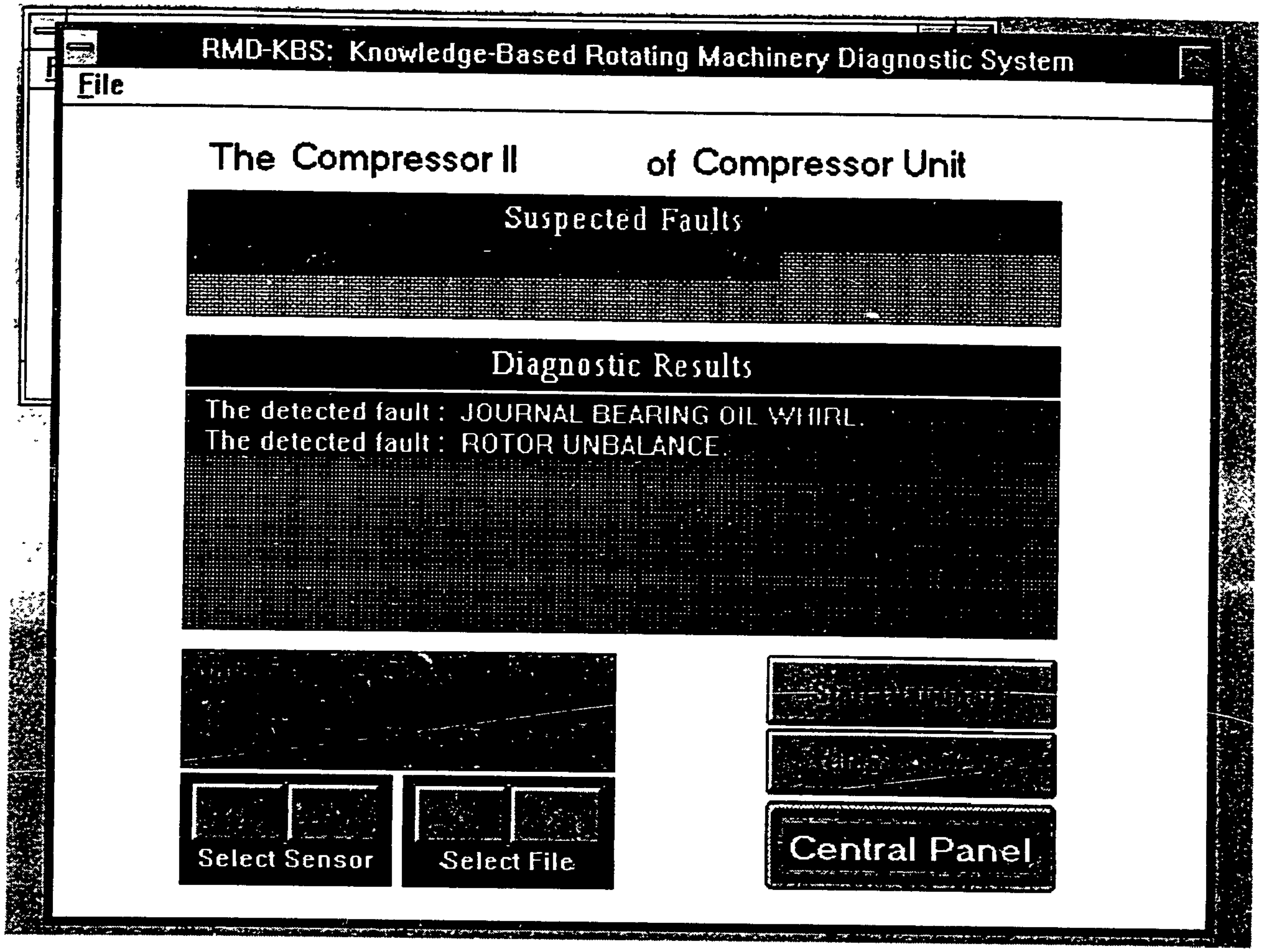


Fig. 7.18 The display of both the hypotheses and the diagnostic results for the compressor unit.

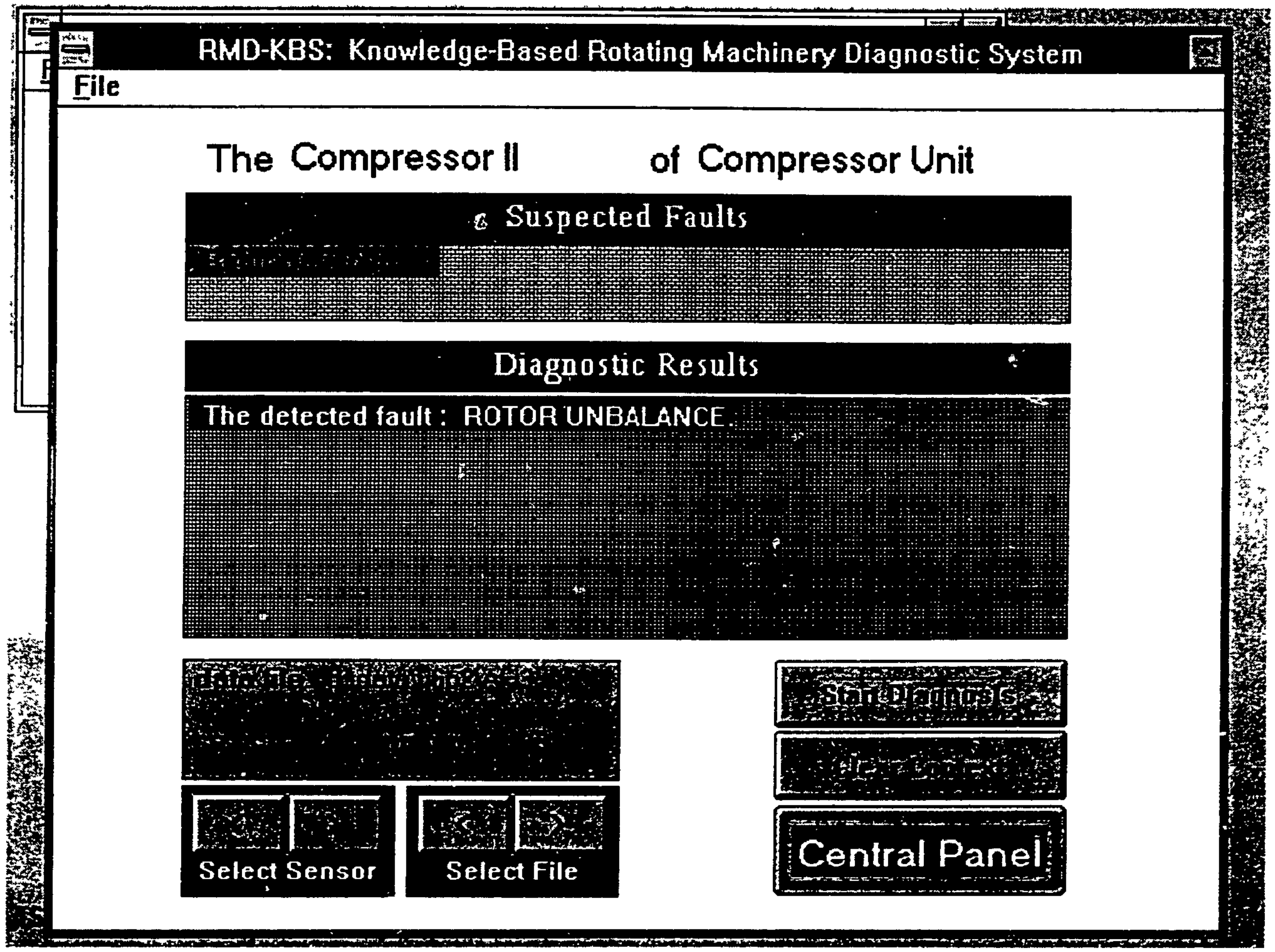


Fig. 7.19 The hypotheses and the diagnostic results for a compressor with unbalance problem.

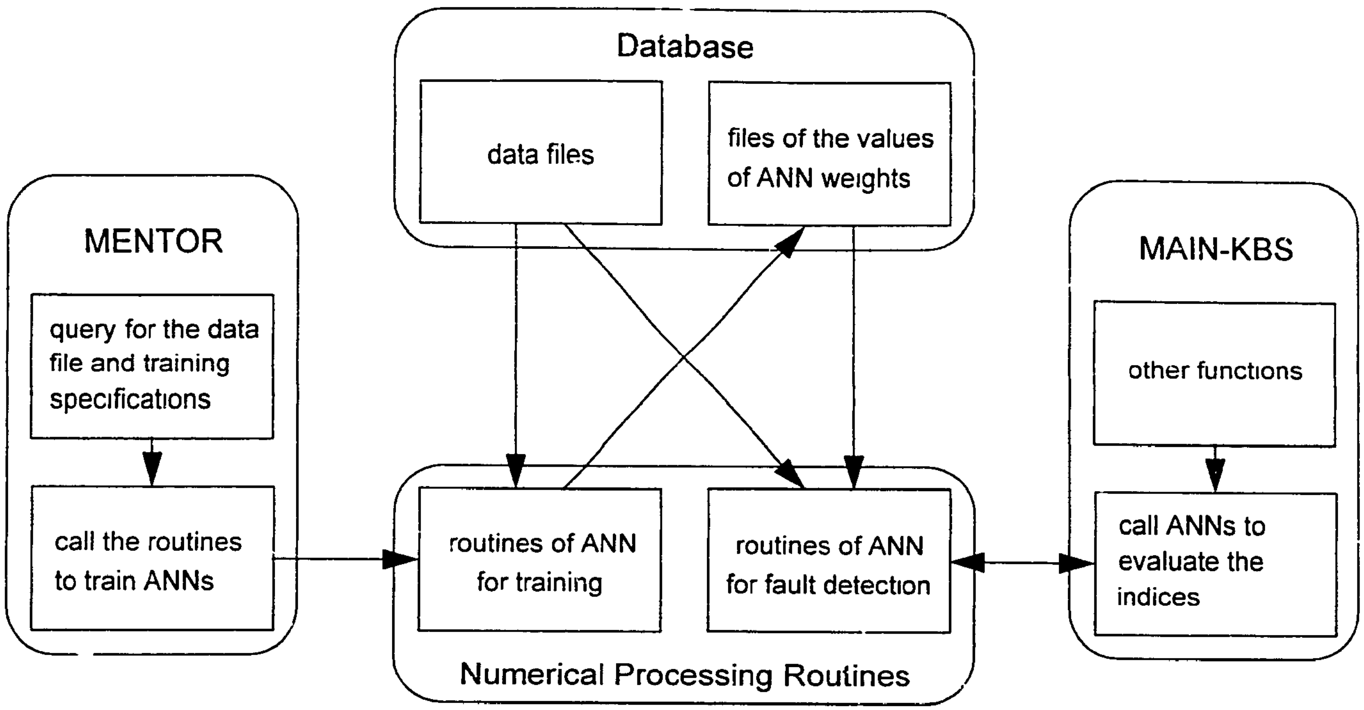


Fig. 7.20 The information flow during the training and utilization of ANNs for fault detection.

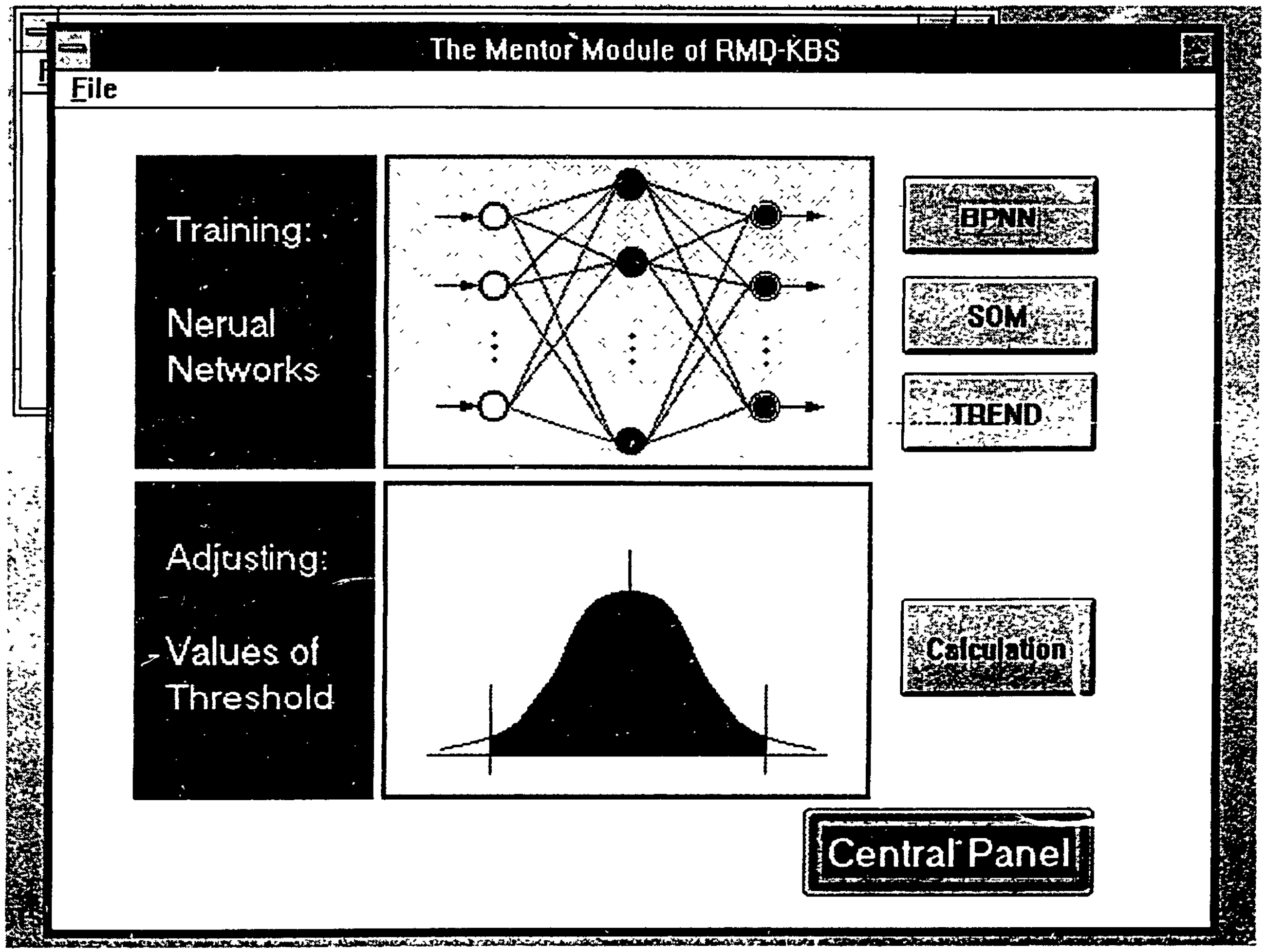


Fig. 7.21 The MEMTOR module of RMD-KBS which performs the training of neural networks.

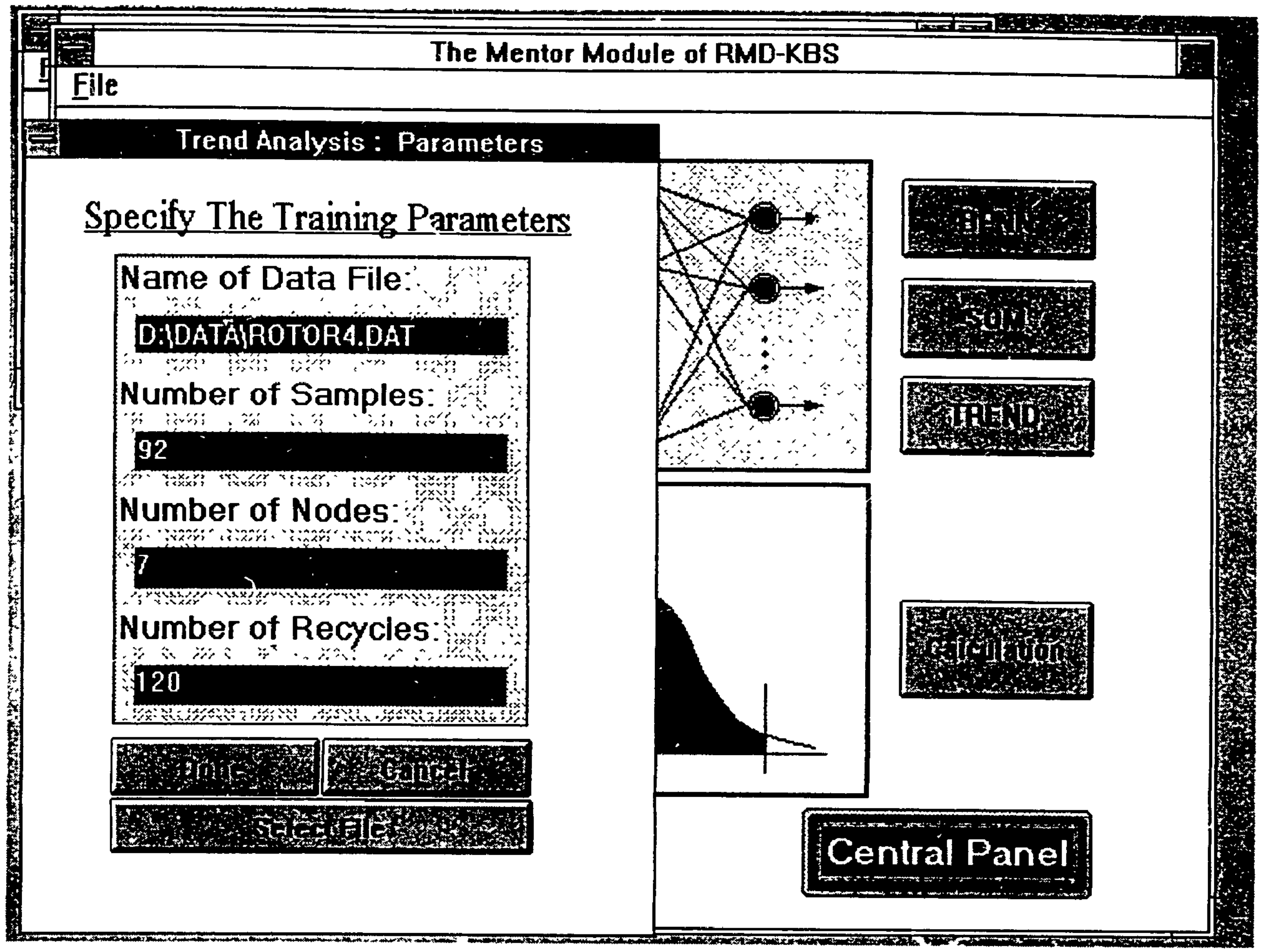


Fig. 7.22 The query for the specifications of ANN training.

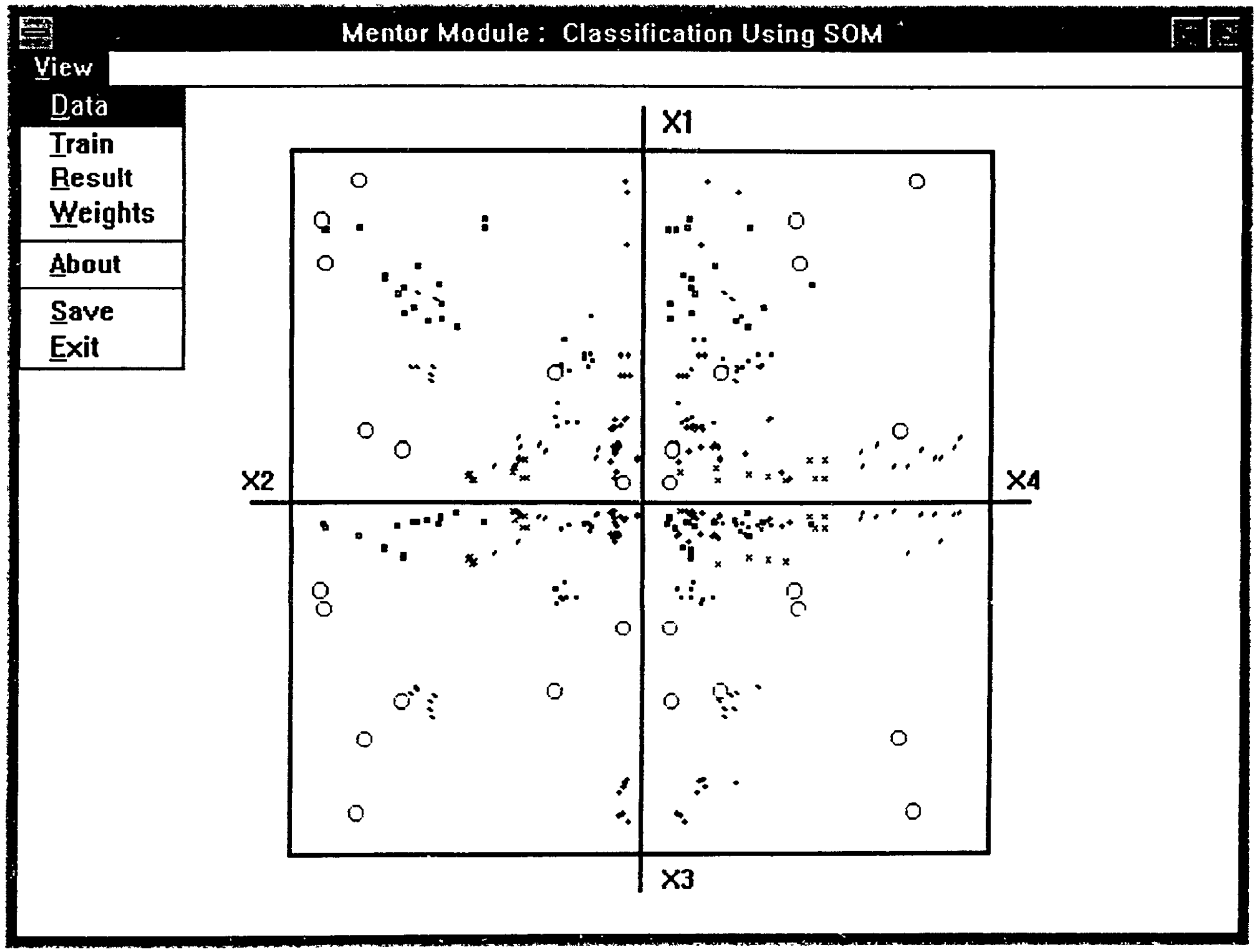


Fig. 7.23 A window displaying the training data samples and the initial positions of the seven weights.

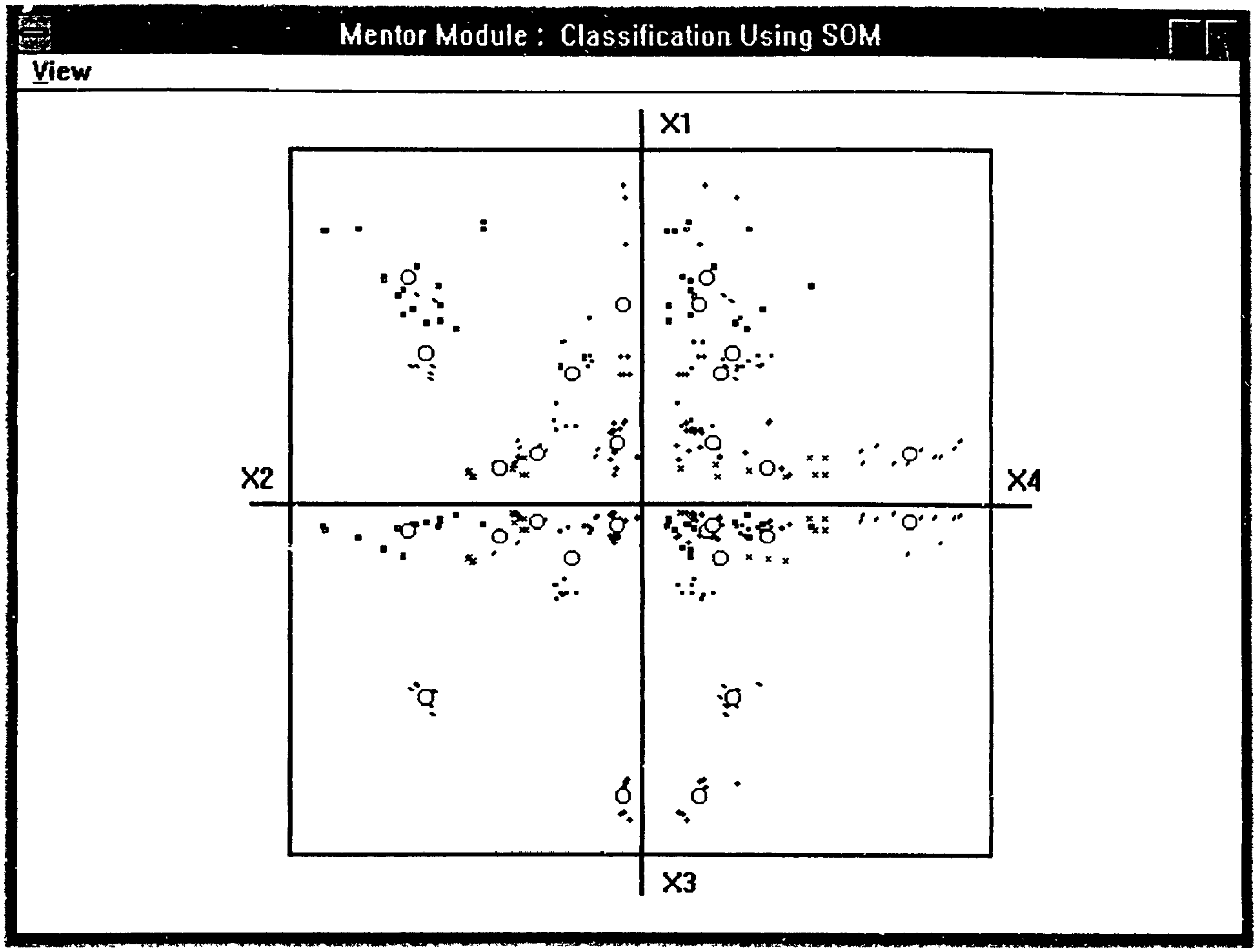


Fig. 7.24 Results of training the SOM network with seven weights.

Mentor Module : Classification Using SOM

View

	X1	X2	X3	X4
1	0.643408	0.662653	0.078920	0.186262
2	0.425122	0.615568	0.555968	0.256026
3	0.561795	0.051197	0.833994	0.161882
4	0.362018	0.197331	0.163566	0.225457
5	0.171066	0.071063	0.061768	0.205403
6	0.089228	0.404598	0.097671	0.359006
7	0.137864	0.295710	0.054517	0.765751

Fig. 7.25 Values of the seven weights of the SOM network for machine fault identification.

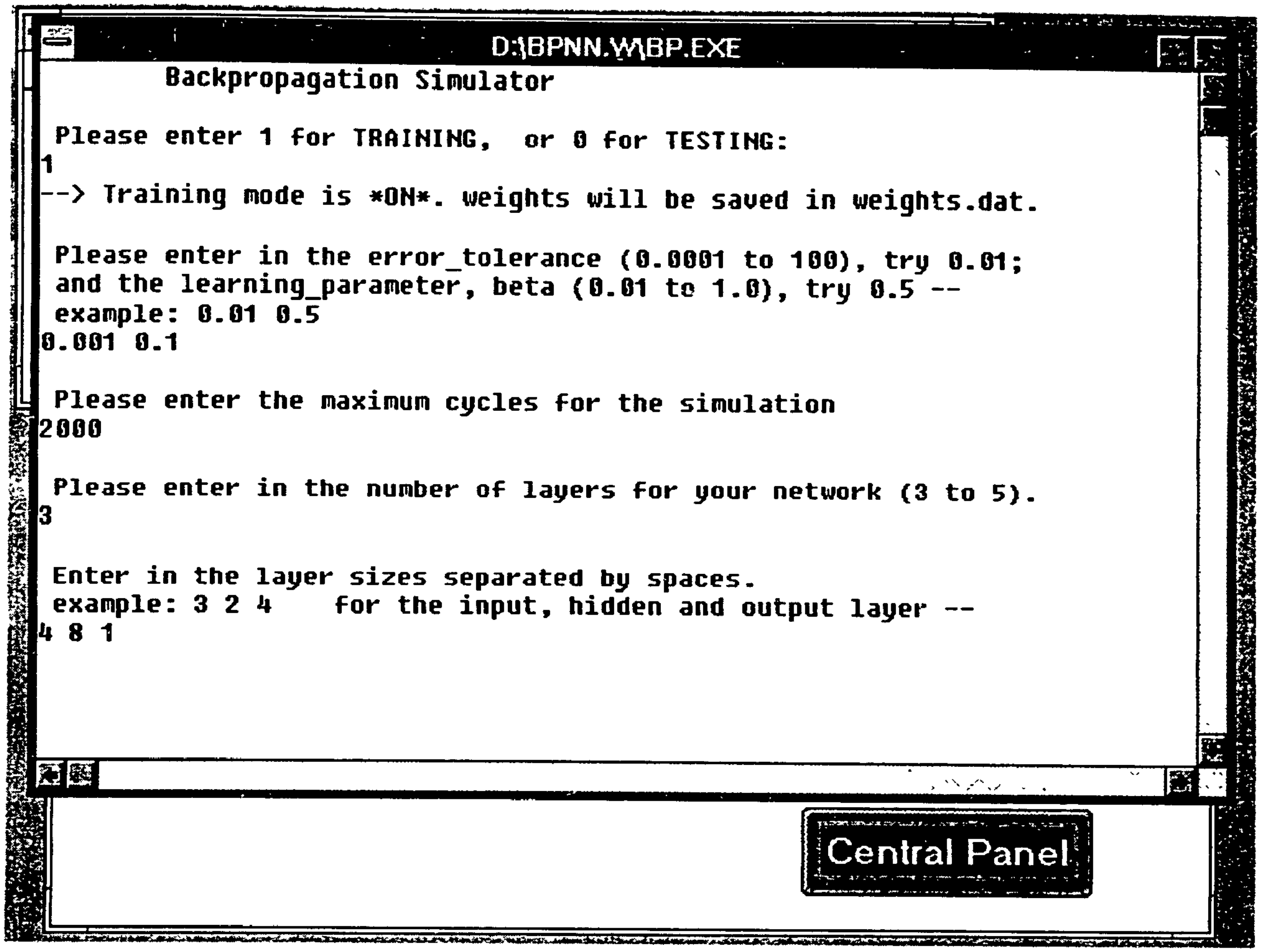


Fig. 7.26 Windows of the routine for training Back-Propagation neural networks.

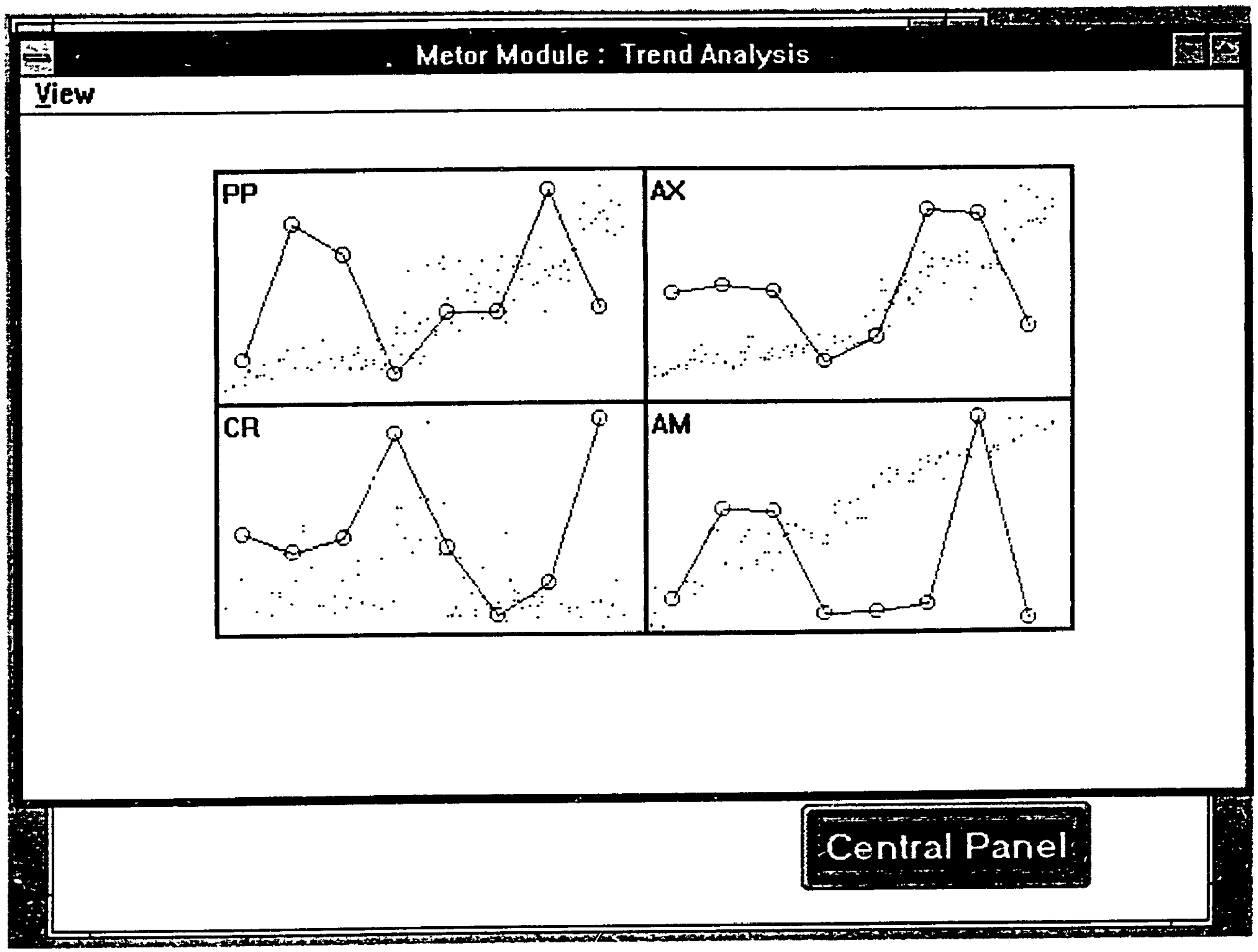


Fig. 7.27 The display of the data samples and the initial weights of the SOM network for trend analysis.

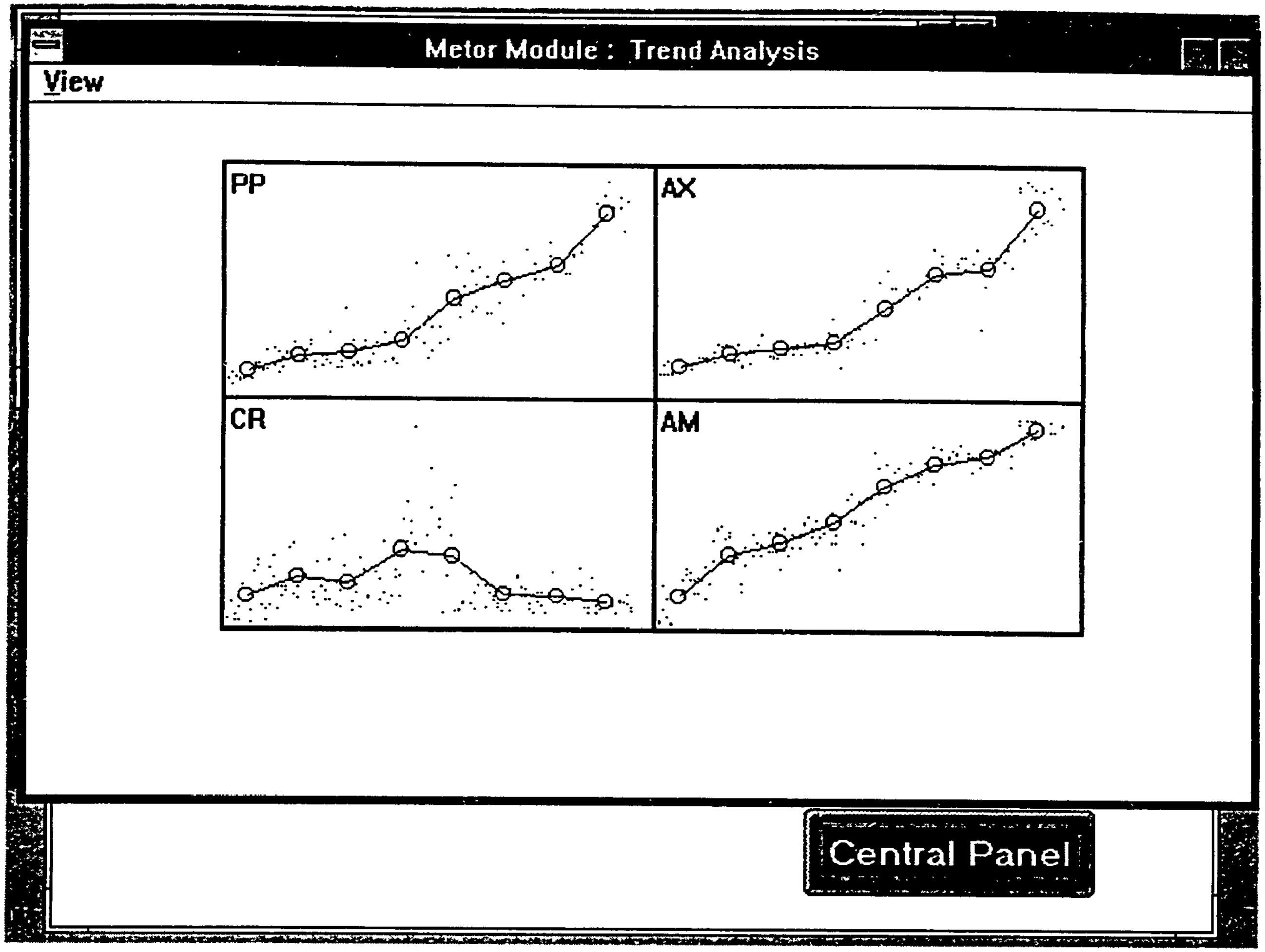


Fig. 7.28 The display of the results of training the SOM network for trend analysis.

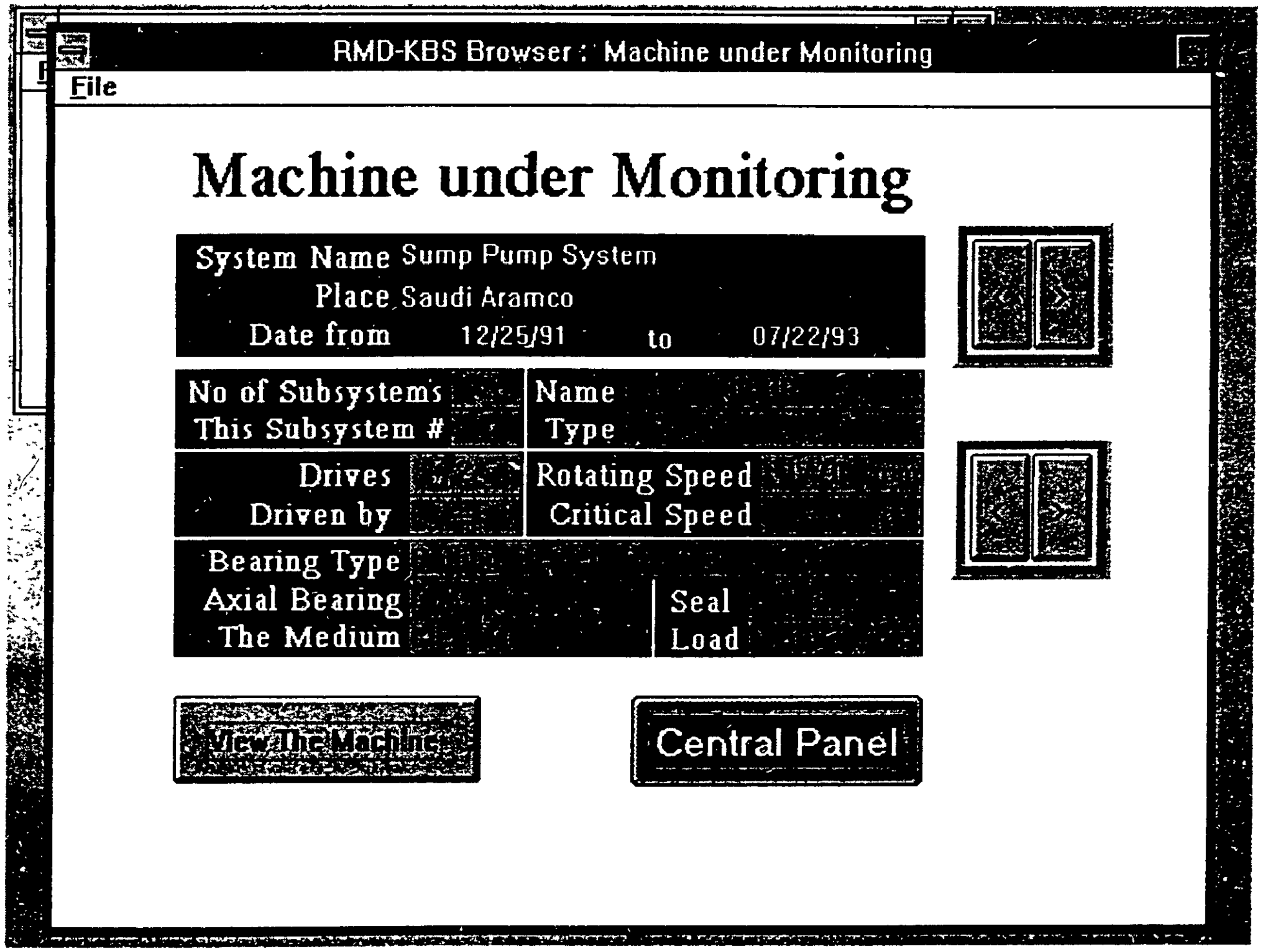


Fig. 7.29 The information about the sump pump system stored in the RMD-KBS.

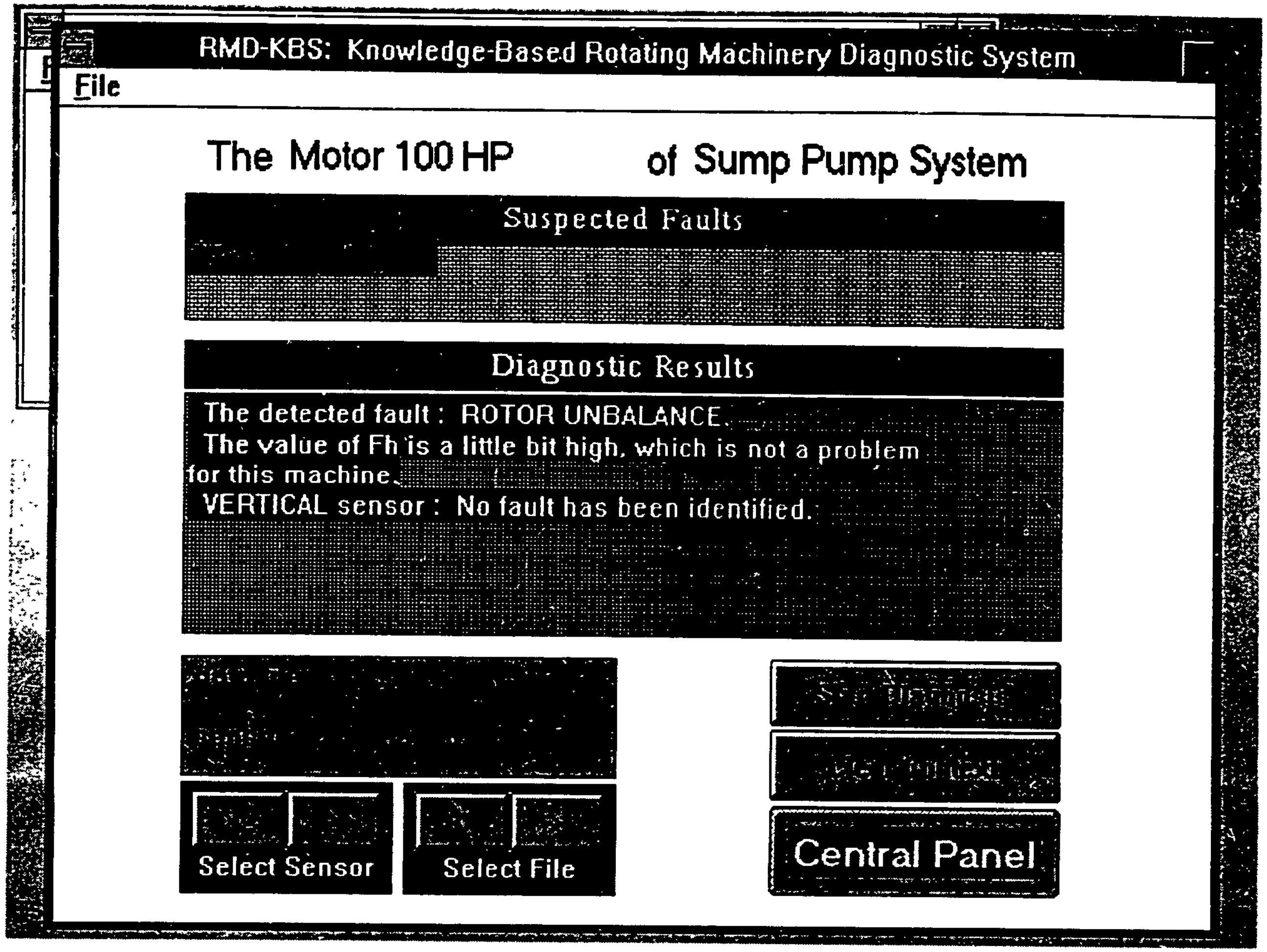


Fig. 7.30 The diagnostic results about the condition of the motor of the sump pump system.

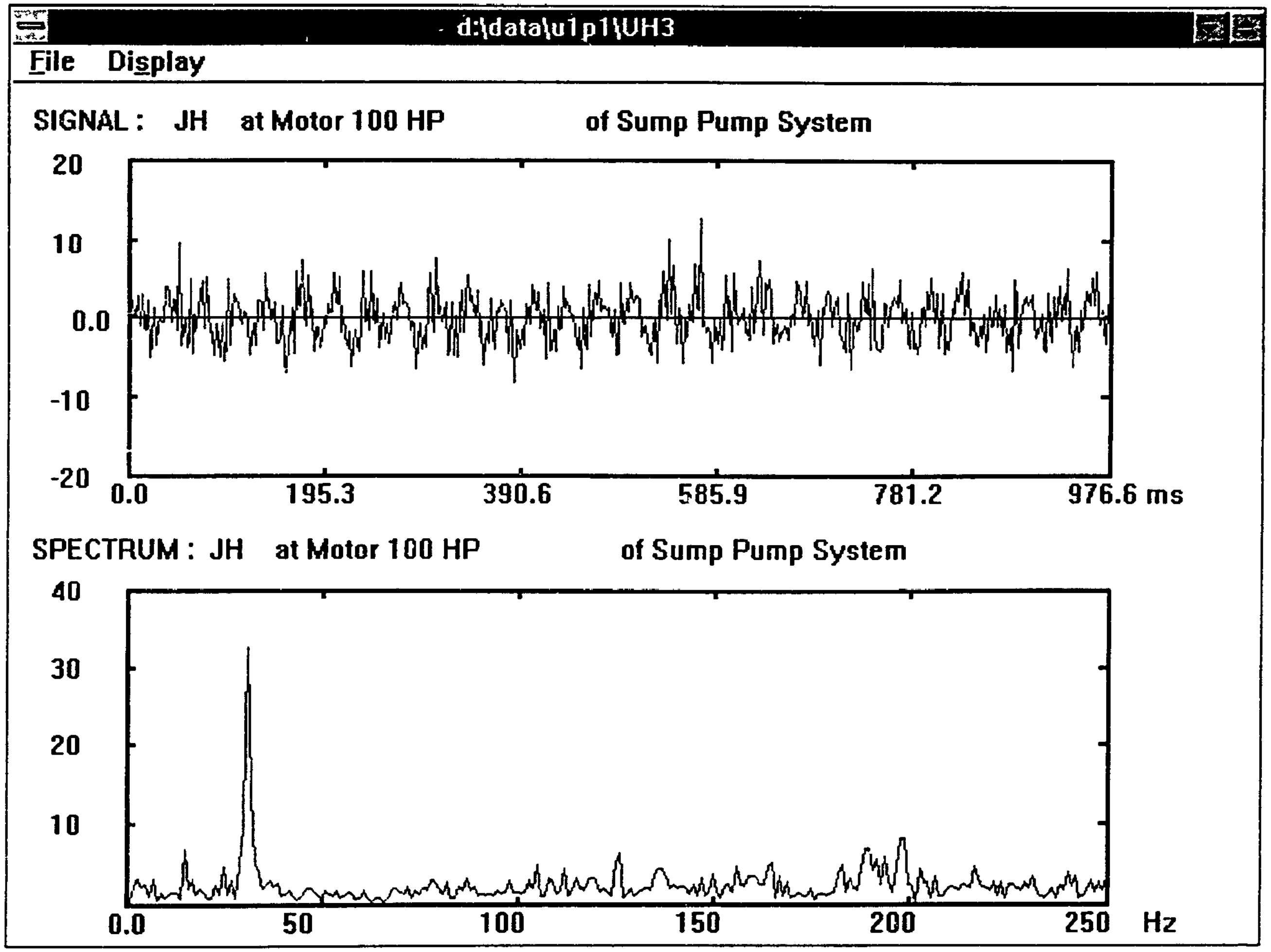


Fig. 7.31 The vibration signal from the motor of the sump pump system and the corresponding spectrum.

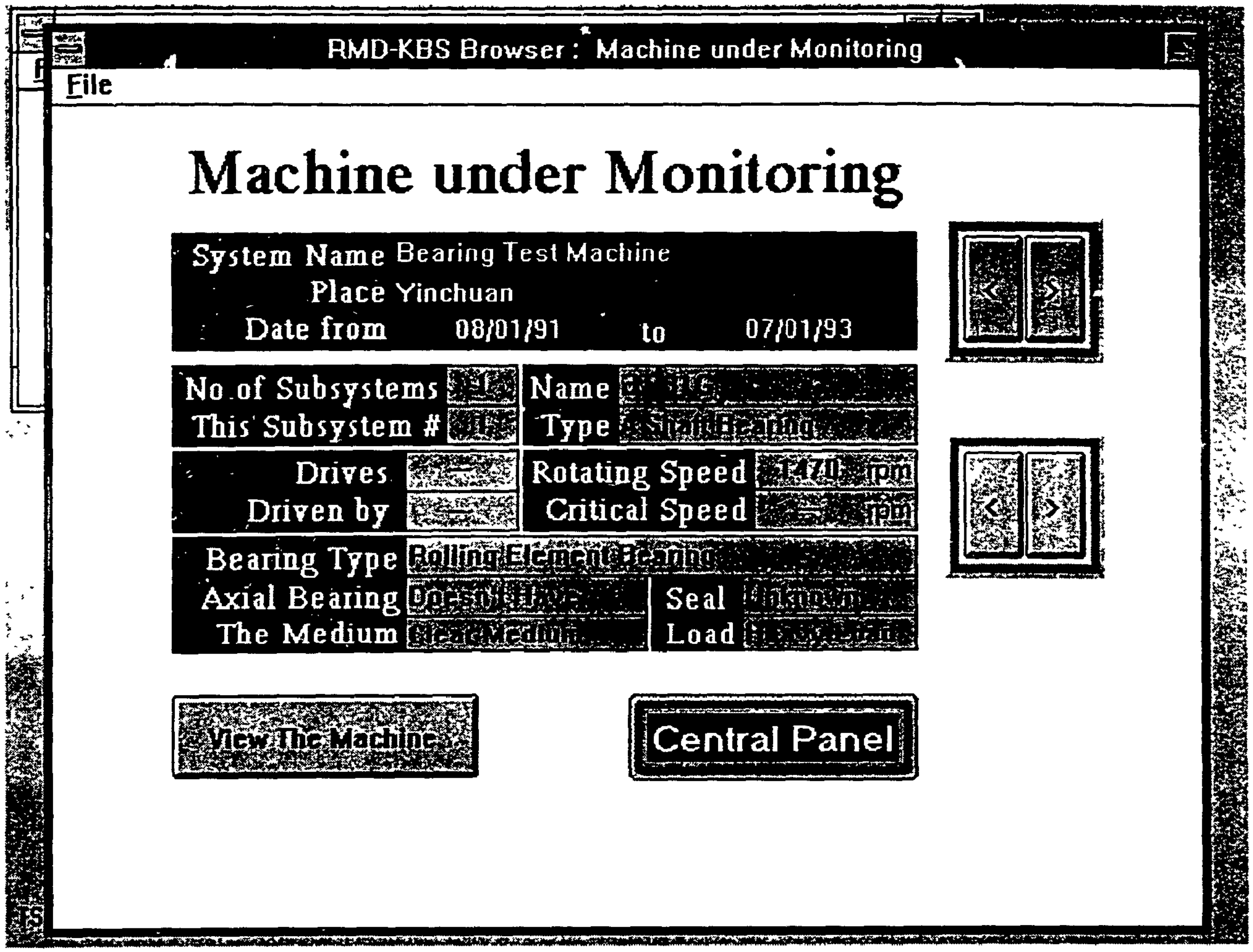


Fig. 7.32 The information about the bearing test machine stored in the RMD-KBS.

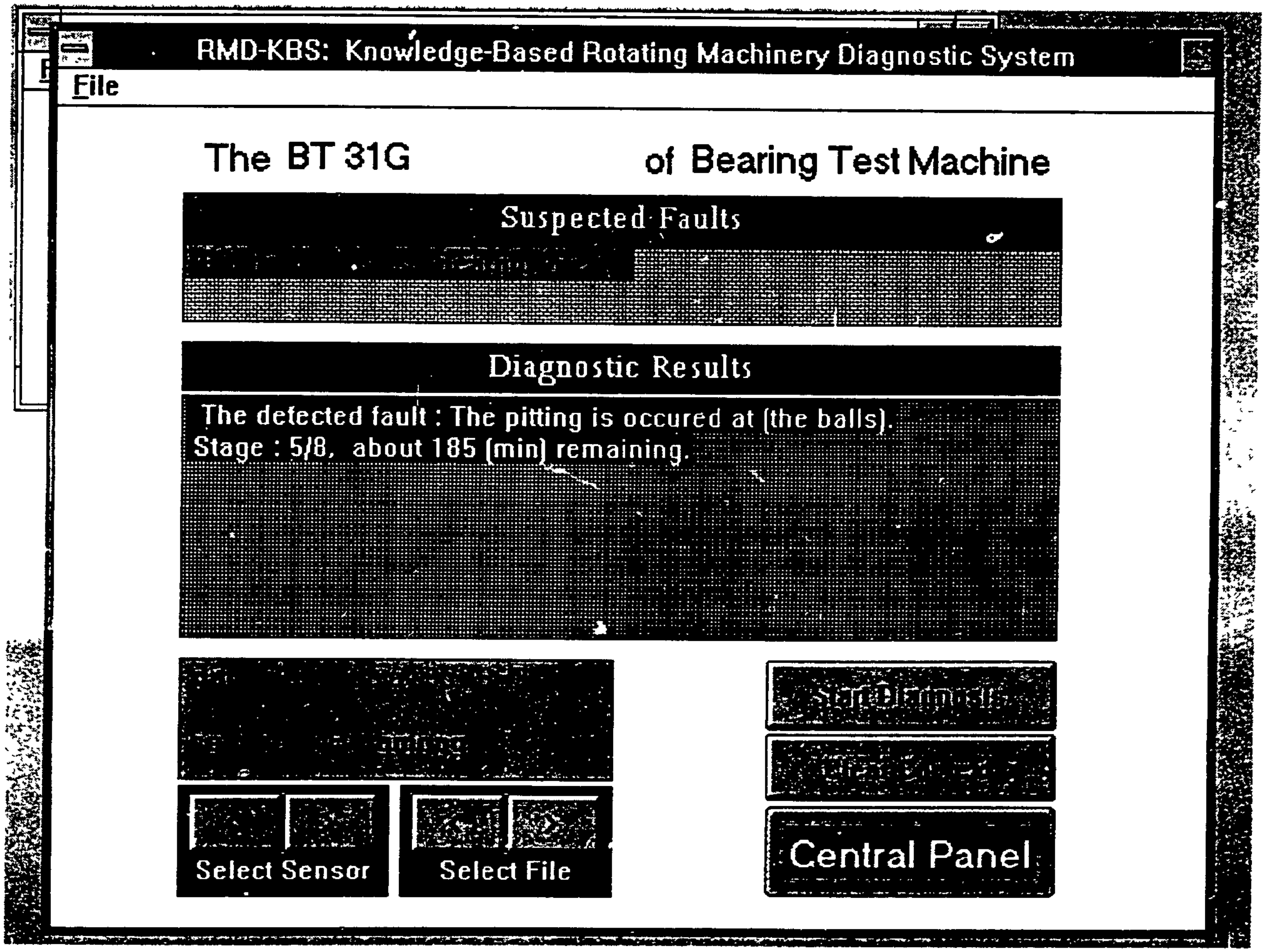


Fig. 7.33 The diagnostic results of the bearing condition.

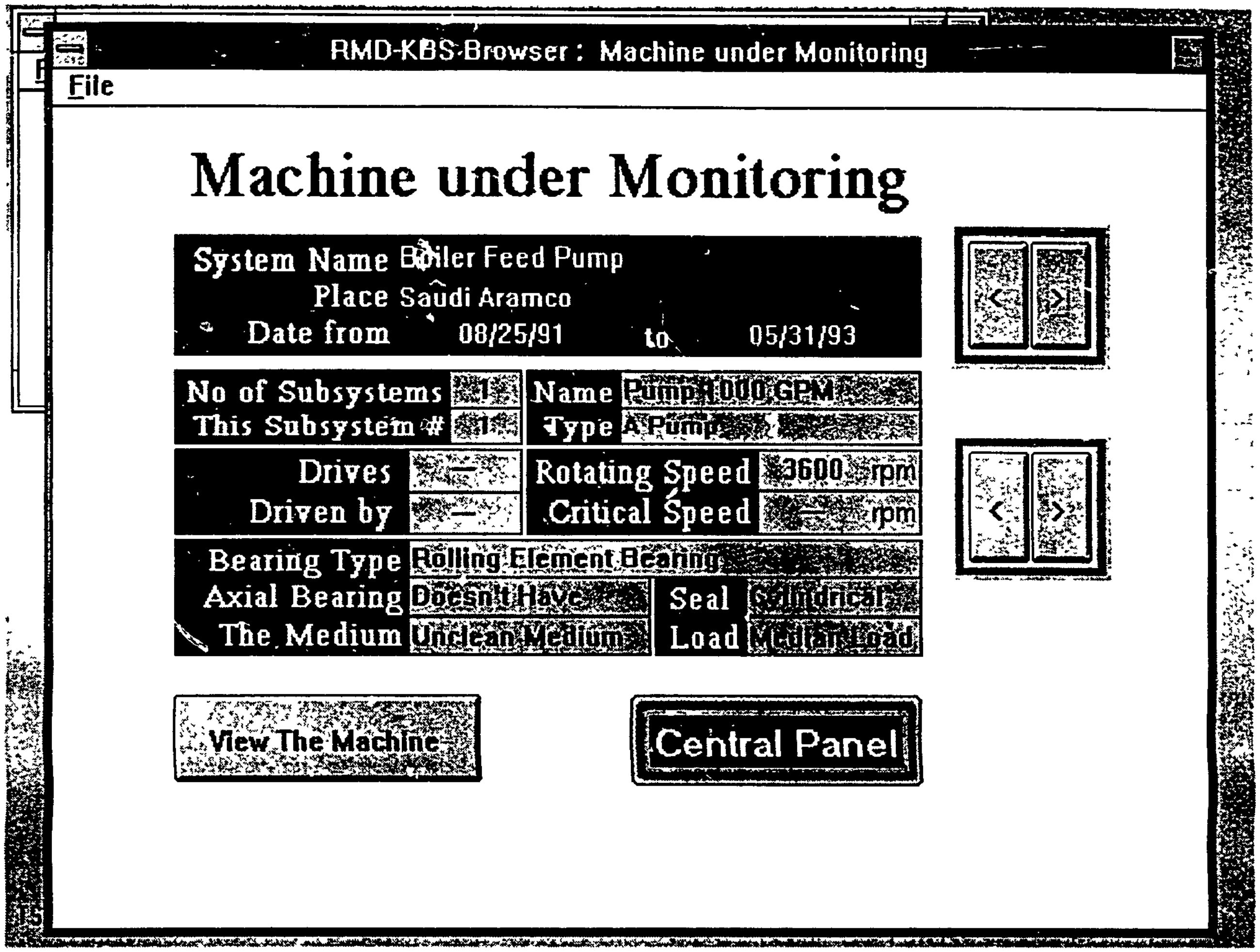


Fig. 7.34 The information about the pump of the boiler feed pump system.

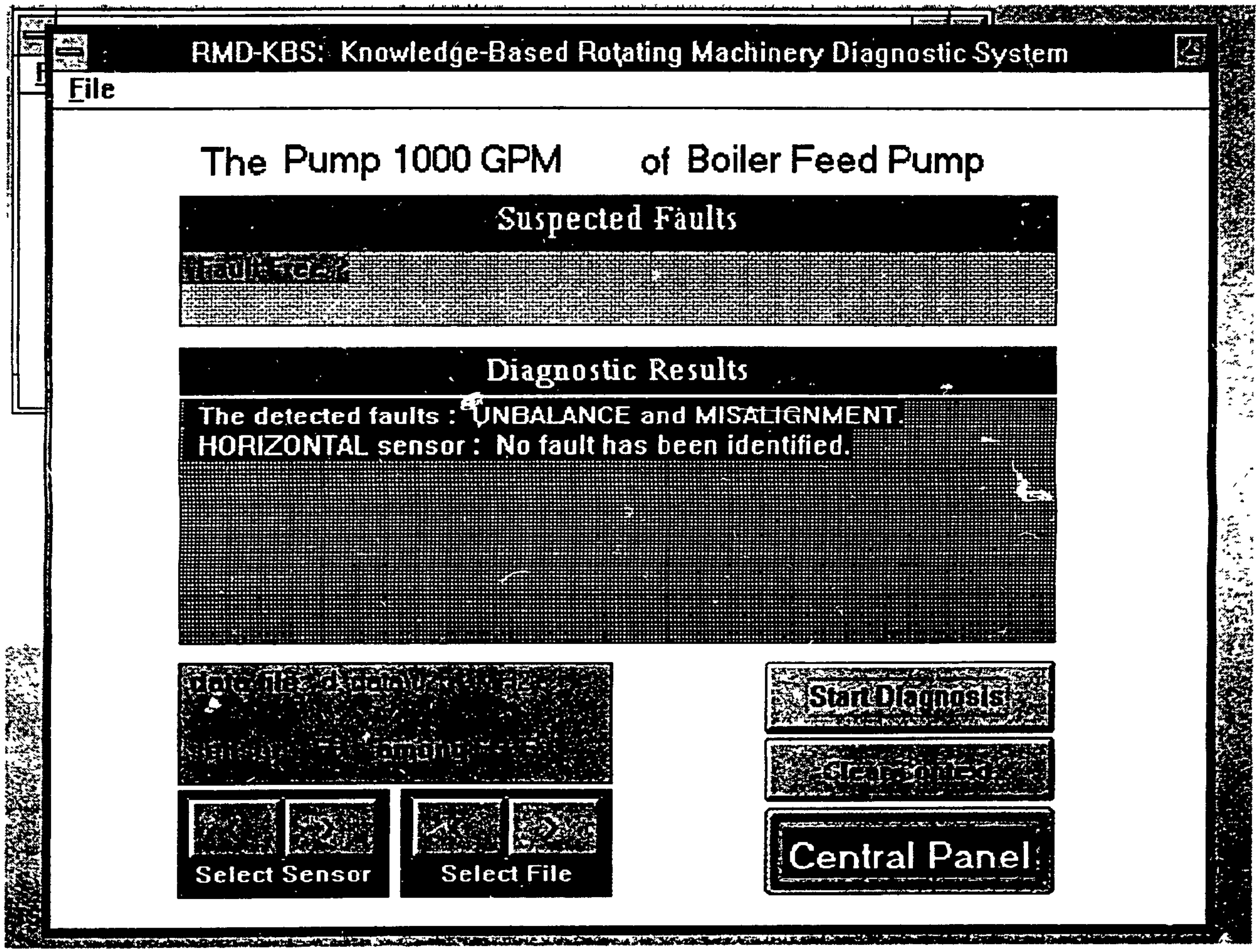


Fig. 7.35 The diagnostic results of the pump condition.

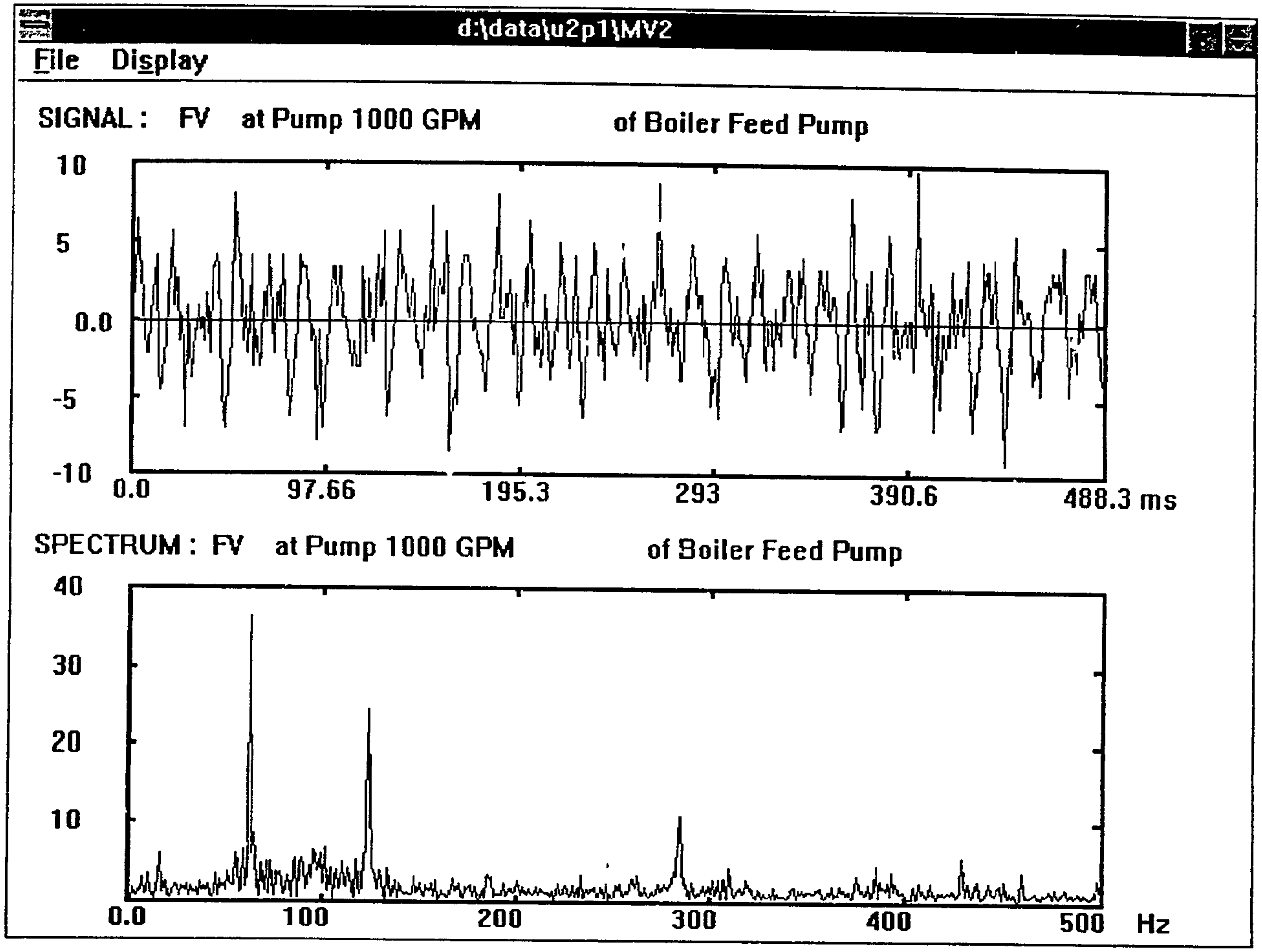


Fig. 7.36 The vibration signal from the boiler feed pump and its spectrum.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

A new approach to design and construct a knowledge-based diagnostic system to monitor and predict the condition of rotating machinery systems, has been developed. This approach is in fact an efficient and judicious combination of available techniques, and utilizes recent advances in computational techniques such as neural networks in conjunction with pertinent expert system programming features. Further, quantitative fault-symptom relationships are obtained based on a number of diagnostic indices that are extracted from on-line vibration signals. It is well known that different indices are sensitive to different types of rotating machinery faults or malfunctions. In order to increase the overall reliability and performance of the KBS, multiple condition related indices are considered in diagnosing the condition of rotating equipment. An on-line mode of operation has been considered and an on-line expert diagnostic system suitable for real-life industrial applications has been constructed. Comparing both the advantages and disadvantages of consultative KBSs and on-line diagnostic KBSs, it is believed that the later are more suitable for industrial applications. Also, the diagnostic strategy that has been implemented in the present approach is very close to the procedure with which human experts perform the diagnosis.

In this work, the diagnostic KBS is designed to contain a deeply-coupled numerical and symbolic processing system, which can perform on its own, most of the tasks involved in rotating machinery diagnosis. The deeply-coupled system makes use of numerical data values and diagnostic indices directly, and thus the loss of data accuracy typically resulting through a number of symbolic transformations, is avoided. To process both the numerical and symbolic information in a better way, the new approach employs hybrid technologies, i.e. the artificial neural networks and the rule-based system, to represent the diagnostic knowledge and to perform the knowledge-based diagnosis. For the quantification of fault development, a new methodology, the Multiple-Index Based Trend Analysis, is employed. It may be noted here that usage of single-index based trending of the fault has so far been used in the MMD of rotating machinery. The usage of multiple indices in the present approach ensures the establishment of a more accurate and precise quantification of the fault as well as, quantitative fault-symptom relationships. Further, the existing SOM algorithms when used to solve the above Multiple-Index Based Trend Analysis, have been found to be inefficient and to lead to erroneous results. A new SOM has been developed from first principles to solve the above multiple-index based trend analysis problem. The new algorithm has been demonstrated to be superior to and more efficient than the existing SOM algorithms.

The two fundamental problems of MMD viz., Condition Identification and Classification, and Quantifying and Trending of the fault development, have been mathematically formulated into problems of Clustering and Pattern Recognition, and Multiple-Index Based Trend Analysis, respectively. The formulation has been oriented in a such a way that neural network solutions, particularly, the solutions using Self-Organizing Mapping algorithms can be obtained. In the design of the RMD-KBS, a numerical processing unit consisting of a group of routines in C++ codes, are invoked during the reasoning. The self-organizing neural networks required to solve the above fundamental problems of MMD, have been embedded in the numerical processing unit.

The establishment of these self-organizing networks is much easier than implementing the pattern recognition, fuzzy reasoning and expert system approaches of previous knowledge-based diagnostic systems. Further, the training of such computationally-effective neural networks requires only a very short CPU time and so they are highly suitable for real-time on-line diagnostic applications.

The symbolic processing unit, which has knowledge about signal processing and machine fault detection, controls the diagnostic process. It has the capability of heuristically selecting the key features corresponding to a diagnostic problem, calling signal processing routines to calculate the current values of the features of interest, and invoking a proper neural network to reason out based on the obtained values of the features. In this manner, the self-organizing networks in the numerical processing unit are invoked by the symbolic processing unit of the RMD-KBS, and further, the parameters required by the neural networks are provided by the symbolic processing unit. Communications between the neural networks and the symbolic processing unit are established in a way similar to that between numerical analysis routines, thus they are easier, simpler and less time consuming. The neural networks become components of the diagnostic system itself, and a deep coupling between the symbolic processing unit and the neural networks component exists. The quantitative relationships between faults and symptoms have been represented by these neural networks in the numerical processing unit. In conventional and existing diagnostic KBSs for rotating machinery monitoring, a rule-based knowledge representation has been used to represent the fault-symptom relationships. The advantages of modern symbolic and numerical processing technologies have thus been utilized in the hybrid system developed in this thesis.

Any updating of the knowledge contained in the RMD-KBS, unlike the existing diagnostic KBSs, depends on the type of knowledge to be added or modified. If new quantitative fault-symptom relationships have to be added, or the existing ones are to be modified, it can be done through training the neural networks, which is a relatively much simpler, easier and less time consuming task, compared to the knowledge updating in previous diagnostic KBSs. This way, the updating of the RMD-KBS is made fairly simple and easy and further, its hybrid architecture has been so designed that network learning can be easily performed by the user.

Based on this new approach, a prototype diagnostic KBS for condition monitoring and diagnosis of rotating machinery, the RMD-KBS has been developed. This system contains more assortments of diagnostic knowledge than existing systems, so that less work is left to its end-user. The user interface has been efficiently designed with high level programming techniques. An efficient and full usage of the data base has been made. The OOP technology has been utilized for structuring the information and knowledge, and sharing the data between different modules.

However, the types and the capability of the neural networks employed in this work, are limited. In the present design, neural networks have been used in the determination of machine condition. It is also possible to use ANNs to perform feature extraction directly from the raw data. It is also possible to employ other types of ANNs that can learn new knowledge in the on-line mode, i.e. at the same time when diagnosis is performed. Moreover, fuzzy computing can be introduced into ANNs, so as to yield the diagnostic results with a certain measure of the possibility of the occurrence of each and all hypotheses, as expressed through fuzzy sets. The establishment of the neural networks has been pre-designed. For example, a SOM can identify four types of bearing conditions based on the values of four diagnostic indices. This may be viewed as a restriction to the symbolic reasoning before achieving the stage of calling this neural network for the quantification of fault development. Moreover, in the present design, a change in the structure (i.e. number of inputs, number of outputs etc.) of an ANN will cause the modification of the corresponding symbolic statements in the KB.

The RMD-KBS is not a complete system. Since symbolic representation of diagnostic knowledge has been relatively well studied, the present research did not attempt to improve rule-based reasoning. It is possible that some new techniques of symbolic representation that are helpful to solve rotating machinery diagnostic problems, could be implemented so as to further improve the performance of the diagnostic system. In cases where the data available for training the neural networks are insufficient, it might be possible to train the neural networks by using the artificial or simulated data sets that are given by domain experts. The neural networks can thus be trained by the expert's experience without requiring data from real-world systems. Then, a more complete knowledge can be manipulated in the neural networks during the implementation. Further, during on-line use of the system, it is quite possible that the training of the neural networks will only focus on the updating of the knowledge.

In general, if the human knowledge can be sorted out as

IF	index 1 IS high
AND	index 2 IS low
AND	index 3 IS not very high
THEN	fault 1 IS occurring
AND	fault 2 IS not occurring

and the quantitative measures, i.e. the interval between the values, information about the high and low values of the indices and so on, are provided by human experts, the artificial simulated data can be given as a uniform or normally distributed data set within the interval corresponding to "high", "low" and "not very high" descriptors. The data can then be used to train ANNs. If the data for network training can not be provided by any means (either real data from a case study, or generated by the expert), then at least the fuzzy measure can not be established, and the fuzzy set involved in the KBSs will fail. For the conventional rule-based system, the rules can be written as given in the above example. However, the machine operator has to know the quantitative measures regarding the machine faults or malfunctions. If so, a proper normalization of the "estimated" data that are obtained from different cases, can be used to train the neural networks.

A study is needed to focus on the proper methods to generate data for network training and to test the diagnostic results, i.e. the relative performance of the ANNs that are trained by simulated data and the data from real-world machine systems, should be established. Moreover, there are some ANN algorithms that have been newly developed. Fuzzy Neural Networks constitute one such category, and they have recently received considerable attention from AI researchers. Fuzzy sets and fuzzy logic, developed by Zadeh in 1965, have recently been proven (Kosco, 1992) to be very successful in solving problems in many areas, but it is difficult to tune the membership functions and adjust the rules before a solution is obtained (Khan and Venkatapuram, 1993). Fuzzy neural networks (FNNs) can learn system behaviour and accordingly, can generate fuzzy rules and membership functions, that are normally given by human experts only (Nakayama et al, 1992; Ishibuchi et al, 1993). Also, FNNs can use qualitative and linguistic knowledge, and perform approximate reasoning under the framework of neural networks with parallel processing (Nie and Linkens, 1992; Narazaki and Ralescu, 1993). This capability makes FNNs distinct from ANNs. Several models of FNNs have quite recently been proposed, such as fuzzy min-max neural networks (Simpson, 1992; 1993), fuzzy ARTMAP (Carpenter et al, 1992), fuzzy multilayer perceptron (Sankar and Mitra, 1992), CFC (Kao and Kuo, 1992; Kou et al, 1993) etc. Their performance and suitability in diagnostic problems are yet to be completely established and further, various aspects of computing are yet to be adequately addressed. Once these issues are resolved, it may be feasible to implement FNNs in diagnostic systems such as the RMD-KBS developed in this thesis.

REFERENCES

Adamovits, P. J., and Pagurek, B., 1993, "Simulation (Model) Based Fault Detection and diagnosis of A Spacecraft Electrical Power System," in *Proc. of IEEE 9th Conf. on AI for Applications*, Orlando, March 1-5, pp. 422-428.

Ali, M. et al, 1986, "A Flight Expert System (FLES) for On-Board Fault Monitoring and Diagnosis," in *Proc. of USAF Workshop on AI Applications*, Colodado, pp. 264-267.

Allard, J. R., and Faemmerer, W. F., 1987, "The Goal/Subgoal Knowledge Representation for Real-Time Process Monitoring," in *Proc. of AAAI-87*, Seattle, Washington, July, pp. 394-398.

Altrock, C. V., Krause, B., and Zimmermann, H. J., 1991, "FIT - A Shell for Fuzzy Expert Systems and Fuzzy Control," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 2748-2758.

Artiba, A., 1991, "Rules-Based Planning System for Parallel Multiproduct Manufacturing Lines," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 1930-1991.

Bacchus, F., 1990, "Lp, A Logic for Representing and Reasoning with Statistical Knowledge," *Computer Intelligence*, Vol. 6, pp. 209-231.

Barr, A., 1981-1989, *The Handbook of Artificial Intelligence*, William Kaufmann Inc., Vol. I-IV, Los Altos, California.

Bernhard, D. L., 1989, "A Knowledge-Based Automatic Machinery Diagnostic System," in *Proc. of the 1st IMMDC (Inter. Machinery Monitoring and Diagnosis Conf.)*, Las Vegas, Nevada, Sept., pp. 651-654.

Bhandarkar, S., and Suk, M., 1991, "Coupling Symbolic and Numerical Computing in Knowledge-Based Systems," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 1483-1490.

Boyce, M. P., Bhargave, R. K., and Chinoy, R., 1989, "On-Line Condition Monitoring of Turbomachinery," in *Proc. of 12th Biennial ASME Conf. on Mechanical Vibration and Noise*, Montreal, Oct., pp. 419-427.

Brawley, G. H., 1989, "Diagnostic Health Condition Performance Monitoring - Does This Make Sense?" in *Proc. of 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 215-221.

Brawley, G. H. et al, 1989b, "Performance Monitoring of Large, Stationary Combustion Turbines," in *Proc. of 12th Biennial ASME Conf. on Mechanical Vibration and Noise*, Montreal, Oct., pp. 428-438.

Buchanan, B. G., and Shortliffe, E. H., 1984, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addion-Wesley, Reading, Mass..

Buckley, J. J., Siler, W., and Tucker, D., 1986, "A Fuzzy Expert System," *Fuzzy Set and Systems*, North-Holland, Vol. 20, pp. 1-16.

Butler, C. W., Hodil, E. D., and Richardson, G. L., 1988, "Building Knowledge-Based Systems with Procedural Languages," *IEEE Expert*, pp. 47-59, Summer.

Cantu-Ortiz, F. J., 1991, "Expert Systems in Manufacturing: An Experience in Mexico," *Expert Systems with Applications*, Pergamon Press, Vol. 3, pp. 445-455.

Carbonell, J. G., Michalski, R. S., and Mitchel, T. M., 1983, "An Overview of Machine Learning," *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, Tioga Publishing Company, Palo Alto, California.

Carlson, G. J. et al, 1988, "An Expert System On-Line Rotor Crack Monitoring and Diagnostic System," in *Proc. of The ASME Inter. Computer in Eng. Conf.*, California, July 31-Aug 4, pp. 361-366.

Carpenter, G. A. et al, 1992, "Fuzzy ARTMAT: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 698-713, Sept..

Cempel, C., 1988, "Vibroacoustical Diagnostics of Machinery: An Outline," *Journal of Mechanical Systems and Signal Processing*, **2**(2), pp. 135-151.

Chao, S. K. et al, 1986, "An Application of Expert System Techniques to Radar Fault Diagnosis," in *Proc. of IEEE Western Conf. on Knowledge-Based Eng. and Expert Systems*, California, June, pp. 127-135.

Chen, K., and Li, C., (Eds.), 1991, *Machinery Monitoring and Diagnostic Techniques*, in Chinese, Science and Technology Press, P. R. China.

Cherkassky, V., and Lari-Najafi, H., 1991, "Constrained Topological Mapping for Nonparametric Regression Analysis," Neural Networks, Vol. 4, pp. 27-40.

Cherkassky, V., and Lari-Najafi, H., 1992, "Nonparametric Regression Using Self-Organizing Topological Maps", *Neural Networks for Human and Machine Perception*, H. Wechsler, ed., Academic Press, Vol. 2, pp. 40-64.

Chiang, T. W., and Wang, K. W., 1992, "A Hybrid AI-Based Approach for Mechanical

System Design Problems," *ISA Trans. on Artificial Intelligence*, Vol. 31, No. 2, pp. 159-164.

Chitoor, S. S., Murty, M. N., and Bhandaru, M. K., 1991, "A New Data Structure HC-Expression for Learning from Examples," *Pattern Recognition*, Pergamon Press, Vol.24, No. 1, pp. 19-29.

Clark, J. D., 1992, *Windows Programmer's Guide to OLE/DDE*, SAMS Printice Hall Computer Publishing, Indiana.

Collacott, R. A., 1977, *Mechanical Fault Diagnosis and Condition Monitoring*, Chapman and Hall, London.

Collacott, R. A., 1979, *Vibration Monitoring and Diagnosis*, George Godwin Limited, London.

Cooper, G. F , 1986, "A Diagnostic Method That Uses Causal Knowledge and Linear Programming in The Applications of Bayes' Formula," *Computer Methods and Programmes in Biomedicine*, Elsevier Science Publishers B.V., Vol. 22, pp. 223-237.

Danai, K., and Chin, H., 1991, "Fault Diagnosis with Process Uncertainty," *ASME Trans. J. of Dynamic Systems, Measurement, and Control*, Vol. 113, pp. 339-343, Sept..

Dempster, A. P., 1967, "Upper and Lower Probabilities Induced by A Multivalued Mapping," *Ann. Math. Stat.*, Vol. 38, pp. 325-339.

Diaz, A. C., 1990, *An Overview of Real Time Expert Systems*, NRC Report, ERA-380, No. 31759, Canada, April.

Dillon, T. S., and Laughton, M. A., 1990, *Expert System Applications in Power Systems*, Prentice Hall International Series in Power Systems Computation, Prentice Hall Inter. UK Ltd, London.

Dimarogonas, A. D., 1992, "Fuzzy Input-Neural Net Adoptive Expert Systems for Rotor Diagnosis and Prognosis," in *Proc. of The Inter. Conf. on Rotating Machine Dynamics*, Venice, April 28-30, pp. 107-113.

Doherty, N. F. et al, 1994, "Knowledge-Based Approaches to Fault Diagnosis: a Practical Evaluation of The Relative Merits of Deep and Shallow Knowledge," *Proc. Inter. Mech. Eng., IMechE*, Vol 208, pp. 39-45.

Eddy, W. F., and Pei, G. P., 1986, "Structures of Rule-Based Belief Functions," *IBM J. Research Development*, Vol. 30, No. 1, pp. 93-101.

El-Karmalawy, M., 1993, "Machinery Monitoring and Diagnostics Using Vibration Signal Analysis", Master's Thesis, Concordia University, Montreal.

Elkordy, M. F., Chang, K. C., and Lee, G. C., 1994, "Application of Neural Networks in Vibrational Signature Analysis," ASCE *Journal of Engineering Mechanics*, Vol. 120, No. 2, pp. 250-265, Feb..

Elman, J. L., 1991, "Distributed Representations, Simple Recurrent Networks, and Grammatical Structure," *Machine Learning*, Vol. 7, No. S2/3, pp. 195-225.

El-Shakweer, A., 1989, "Experimental Analysis Technique to Diagnose Sleeve Bearing Problems and Verify Its Design," in *Proc. of 12th Biennial ASME Conf. on Mechanical Vibration and Noise*, Montreal, Oct., pp. 134-138.

Eshleman, R. L., and Jackson, C., 1992, "Performance Verification, Diagnostics, Parameter Identification, and Condition Monitoring of Rotating Machinery," *Handbook of Rotordynamics*, F. F. Ehrich, ed., McGill-Hill, USA, Chapter 4.

Feldman, J. A., and Ballard, D. H., 1987, "Connectionist Models and Their Properties," *Cognitive Science*, Vol. 6, pp 205-254.

Feldman, R., Segre, S., and Koppel, M., 1991, "Refinement of Approximate Rule Base," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 615-622.

Feret, M. P., and Glasgow, J. H., 1992, "Case-Based Reasoning in Model-Based Diagnosis," *in Proc. of AIENG 92*, Waterloo, July.

Freeman, R. A., 1991, "Use of Computers for Non Destructive Testing and Condition Monitoring in Aerospace," *Expert Systems with Applications*, Pergamon Press, Vol. 3, pp. 67-83.

Friedman, J. F., and Silverman, B. W., 1989, "Flexible Parsimonious Smoothing and Additive Modelling", *Technometrics*, Vol. 31, No. 1, pp. 3-21.

Fu, L. M., 1989, "Building Expert Systems on Neural Architecture," in *Proc. of 1st IEE Inter. Conf. on ANN*, London, Oct. 16-18, pp.221-225.

Gaines, B., 1988, "Knowledge Acquisition Systems for Rapid Prototyping of Expert Systems," *INFOR*, Vol. 26, No. 4, pp. 256-285.

Garcia, O. N., and Chien, Y., 1991, *Knowledge-Based Systems: Fundamentals and Tools*, IEEE Computer Society Press, Los Alamitos, California.

Gauger, R. H., and Smee, A. B., 1989, "Data Acquisition and Monitoring with Expert System Diagnostics," in *Proc. of the 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 528-533.

Geman, S., Bienenstock, E., and Doursat, R., 1992, "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, Vol. 4, pp. 1-58.

Good, M. et al, 1989, "A Multi-Site Power Plant Vibration Monitoring and Analysis

System." in *Proc. of 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 534-540.

Grace, M. C., 1990, "Thermodynamic Performance Analysis and Trending Techniques for Single Section Centrifugal Compressors," in *Proc. of 2nd Inter. Machinery Monitoring and Diagnosis Conf. (IMMDC)*, Los Angeles, CA, pp. 242-248.

Grossberg, S., 1976, "Adaptive Pattern Classification and Universal Recording, I: Parallel Development and Codding of Neural Feature Detectors," *Biolog. Cybernetics*, Vol. 23, pp. 121-134.

Grossberg, S., 1986, "The Adaptive Brain I: Cognition Learning, Reinforcement, and Rhythm," and "The Adaptive Brain II: Vision, Speech, Language, and Motor Control," Elsevier/North-Holland, Amsterdam.

Guth, M. A. S., 1987, "An Expert System Design Incorporating Fuzzy Logic for Diagnosing Heat Imbalances in A Nuclear Power Plant," in *Proc. of 3rd Annual Expert Systems in Government Symposium*, Washington, D.C., Oct., pp. 210-216.

Haddad, S. D., 1991, "Condition Monitoring and Fault Diagnosis for Predictive Maintenance and Quality Control," in *Proc. of 3rd Inter. Machinery Monitoring and Diagnosis Conf.*, Las Vegas, Nevada, Dec. 9-12, pp. 1-13.

Haramis, G. T., Karberg, W. E., and Berris, R. E., 1989, "Ultrasonic On-Line Bearing Condition Monitoring Technology," in *Proc. of 1st International Machinery Monitoring and Diagnosis Conf. (IMMDC)*, Las Vegas, Nevada, Sept., pp. 127-133.

Harmon, P., Maus, R., and Morrissey, W., 1988, *Expert Systems: Tools and Applications*, Wiley, New York, N.Y..

Hart, P., 1986, "Interview: Peter Hart Talks about Expert Systems," *IEEE Expert*, pp. 96-99, Spring.

Heinze, D. T., 1989, "A Signal Analysis Expert System for Signal Noise Reduction," in *Proc. of 1989 Annual AI Systems in Government Symposium*, Washington, D.C., March 27-31, pp. 93-99.

Hill, J. W., and Baines, N. C., 1988, "Application of an Expert System to Rotating Machinery Health Monitoring", in *Proc. of Inter. Conf. on Vibration in Rotating Machinery*, IMechE, C307/88, pp. 449-454.

Hopfield, J. J., 1982, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. Natl. Acad. Sci.*, USA, Vol. 79 pp. 2544-2558.

Hopfield, J. J., 1984, "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," *Proc. of Natl. Acad. Sci. USA*, Vol. 81, pp 3088-3092, May.

Hopfield. J. J.. and Tank. D. W.. 1986. "Computing with Neural Circuits: A model," *Science*, Vol. 233, pp. 625-633. August.

Irgon, A. et al, 1990, "Expert System Development," *IEEE Expert*, pp. 25-39, June.

Irwin, J. A., and Orden, H. M. V., 1987, "Expert System Interface for B-1B CEPS," in *Proc. of Western Conf. on Expert Systems*, California, June, pp. 46-49.

Ishibuchi, H., Fujioka, R., and Tanaka, H., 1993, "Neural Networks That Learn from Fuzzy If-Then Rules," *IEEE Trans. on Fuzzy Systems*, Vol. 1, No. 2, pp 85-97, May.

Ishii, K., 1988, "Expert Systems in Japan," *IEEE Expert*, pp. 65-74, Summer.

Iwamasa, M. et al, 1992, "Model-Based Diagnosis Using Qualitative Causal Model and Set-Covering," in *Proc. of The IEEE 8th Conf. on AI for Applications*, Monterey, March 2-6, pp. 60-66.

Jiaa, C. L. and Dornfeld, D., 1992, "Detection of Tool Wear Using Gradient Adaptive Lattice and Pattern Recognition Analysis," *Journal of Mechanical Systems and Signal Processing*, Vol. 6, No. 2, pp. 97-120.

Kadesch, R. R., 1986, "Subjective Inference with Multiple Evidence," *Artificial Intelligence*, Elsevier Science Publishers B.V., North-Holland, Vol. 28, pp. 333-341.

Kadushin, D., 1991, "Roller Element Bearing Fault Analysis Using Envelope Detection During an Experiment Case Study," in *Proc. of 3rd Inter. Machinery Monitoring and Diagnosis Conf. (IMMDC)*, Las Vegas, Nevada, Dec. 9-12, pp. 132-136.

Kanal, L., and Raghavan, S., 1992, "Hybrid Systems - A Key to Intelligent Pattern Recognition," in *Proc. of IEEE Inter. Joint Conf. on Neural Networks*, Baltimore, June 7-11, pp. IV-177-183.

Kandel, A., 1992, *Fuzzy Expert Systems*, CRC Press, Boca Raton, Florida.

Kao, C., and Kuo, Y. H., 1992, "A Neural Network Model Based Fuzzy Classification Concept," in *Proc. of IEEE Inter. Joint Conf. on Neural Networks*, Baltimore, June 7-11, pp. II727-732.

Karel, G., and Kenner, M., 1988, "KLUE: A Diagnostic Expert System Tool for Manufacturing," in *Proc. of the ASME Inter. Computer in Eng. Conf.*, California, July 31-Aug 4, pp. 331-336.

Kato, M. et al, 1990, "Turbine Generator Monitoring and Diagnostic Expert System," in *Proc. of 2nd IMMDC*, Los Angeles, pp. 13-18.

Keim, M., and Nordmann, R., 1989, "Application of An Expert System for Diagnosis of Rotordynamic Problems in Turbomachinery," in *Proc of 12th Biennial ASME Conf. on*

Mechanical Vibration and Noise, Montreal, Oct., pp. 85-91.

Khan, E., and Venkatapuram, P., 1993, "Neufuz: Neural Network Based Fuzzy Logic Design Algorithms," in *Proc. of 2nd IEEE Inter. Conf on Fuzzy Systems*, San Francisco, Mar.28 - April 1, pp. 647-654.

Khosla, R., and Dillon, T., 1993, "Task Decomposition and Competing Expert System-Artificial Neural Net Objects for Reliable and Real Time Inference," in *Proc. of IEEE Inter. Conf. on Neural Networks*, San Francisco, CA, Mar. 28- April 1, pp. 794-800.

Kim, P. Y., and Lowe, I. R. G., 1983, "A Review of Rolling Element Bearing Health Monitoring," in *Proceedings of Machinery Vibration Monitoring and Analysis Meeting*, Houston, Texs, April, pp. 145-154.

Kim, J., and Courtney, J. F., 1988, "A Survey of Knowledge Acquisition Techniques and Their Relevance to Managerial Problem Domains," *Decision Support Systems*, Elsevier Science Publishers B.V., North-Holland, Vol. 4, pp. 269-284.

Kim, D. S., Shin, Y. S., and Carlson, D. K., 1991, "Machinery Diagnostics for Rotating Machinery Using Backpropagation Neural Networks," in *Proc. of 3rd Inter. Machinery Monitoring and Diagnosis Conf.*, Las Vegas, Nevada, Dec.9-12, pp. 309-320.

Kirk, R. G., Hoglund, J., and Mondy, R. E., 1989, "Development of A PC-Based Off-Line Expert System for Evaluation of Turbomachinery Response," in *Proc. of 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 439-444.

Kohonen, T., 1984, *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.

Kohonen, T., 1988, "Self-Organization and Associative Memory," *Springer Series in Information Sciences*, Springer Verlag, New York.

Kohonen, T., 1989, "The Self-Organizing Map", *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1464-1480.

Kolodner, J. L., 1991, "Improving Human Decision Making through Case-Based Decision Aiding," *AI Magazine*, Vol. 12(2), pp. 52-68.

Kosco, B., 1988, "Bidirectional Associative Memories," *IEEE Trans. System, Man, Cybernetics*, Vol. SMC-L8, pp. 46-60, Jan./Feb..

Kosco, B., 1992, *Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, Englewood Cliffs, NJ.

Kuo, Y. H., Kao, C., and Chen, J., 1993, "A Fuzzy Neural Network Model and Its Hardware Implementation," *IEEE Trans. on Fuzzy Systems*, Vol. 1, No. 3, pp. 171-183, Aug..

Kuczewski, R. M., and Eames, D. R., 1992, "Helicopter Fault Detection and Classification with Neural Networks," in *Proc. of Inter. Joint Conf of Neural Networks*, Baltimore, June 7-11, pp. II-947-956.

Laffey, T. J. et al, 1986, "Reasoning about Fault Diagnosis with LES," *IEEE Expert*, pp. 13-20, Spring.

Lee, J., and Kim, B. M., 1993, "Analysis of Machine Degradation Using A Neural Network Based Pattern Discrimination Model," *Journal of Manufacturing Systems*, Vol. 12, No. 5, pp. 379-387.

Lee, L., 1991, "Review of Artificial Intelligence and Expert Systems Applications Within BHP Australia," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 685-697.

Leonard, J. A., and Kramer, M. A., 1993, "Diagnosing Dynamic Faults Using Modular Neural Nets," *IEEE Expert*, pp. 44-53, April.

Leung, P. S., 1992, "The Effects of a Transverse Crack on the Dynamics of a Circular Shaft," *Rotordynamics, Proc. of the Inter. Conf. on Rotating Machine Dynamics*, Venice, April, pp. 59-67.

--, 1990a, *LEVEL5 OBJECT™ User's Guide*, Information Builders, Inc., New York.

--, 1990b, *LEVEL5 OBJECT™ Reference Guide*, Information Builders, Inc., New York.

Liddle, I., and Reilly, S., 1993, "Diagnosing Vibration Problems with an Expert System," *Mechanical Engineering*, pp. 54-55, April.

Lipovszky, G., Sólyomvári, K., and Varga, G., 1990, *Vibration Testing of Machines and Their Maintenance*, Elsevier Science Publishing Co., Amsterdam.

Lippmann, R. P., 1987, "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, pp. 4-22, April.

Liu, Y., Chen, K., and Zhang, S., 1991, "Early Detection of Gearbox Fault Using Frequency Demodulation," in *Proc. of 3rd IMMDC*, Las Vegas, Dec. 9-12, pp. 72-76.

Liu, T. I., and Mengel, J. M., 1992, "Intelligent Monitoring of Ball Bearing Conditions," *Mechanical Systems and Signal Processing*, Vol. 6, No.5, pp 419-431.

Liu, T. I., and Anantharaman, K. S., 1993, "On-Line Sensing of Drill Wear Using Neural Network Approach," in *Proc. of 1993 IEEE Inter. Conf. on Neural Networks*, San Francisco, Mar 28- April 1, pp. 690-694.

Lukas, M., and Anderson, D. P., 1989 "Developments in Instrumentation and Automation in Techniques for Machine Condition Monitoring Through Oil Analysis," in *Proc. of 1st*

IMMDC, Las Vegas, Nevada, Sept., pp. 479-487.

Lyon, R. H., and Dejong, R. G., 1984, "Design of a High-Level Diagnostic System," *J. of Vibration, Acoustics, Stress, and Reliability in Design, Trans. of the ASME*, Vol. 106, pp. 17-21, Jan..

Mahalingam, S., and Sharma, D. D., 1986, "ROMAD - An Expert System for Vibration Analysis," in *Proc. of the 41st Meeting of The Mechanical Failures Prevention Group*, Ed. by Shives, T.R., Naval Air Test Centre, Patuxent River, Maryland, Oct. 28-30, pp. 391-401.

Majstorović, D. V., 1990, "Expert Systems for Diagnosis and Maintenance: The State-of the-Art," *Computer in Industry*, Vol 15, pp. 43-68.

Maile, A. J. K. et al, 1991, "ESR - A Large Knowledge Based System Project of European Power Generation Industry," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 522-529.

Martin, M. R., Ismail, F., and Sakuta, A., 1990, "New Statistical Approach for Gear Damage Detection," in *Proc. of 2nd Inter. Machinery Monitoring and Diagnosis Conf.*, Los Angeles, CA, pp. 329-334.

Mathew, J., and Alfredson, R. J., 1984, "The Condition Monitoring of Rolling Element Bearing Using Vibration Analysis," *ASME Trans., Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 106, pp. 447-453, July.

Matteson, M. A., Morris, R., and Tate, R., 1992, "Real Time GMAW Quality Classification Using An Artificial Neural Networks with Airborne Acoustic Signals as Inputs," in *Proc. of Inter. Conf. on Computerization of Welding Information*, Orlando, FL, Oct..

McFadden, P. D., 1989, "Interpolation Techniques for Time Domain Averaging of Gear Vibration," *J. of Mechanical Systems and Signal Processing*, Vol. 3, No. 1, pp. 87-97.

Mechefske C. K., and Mathew, J., 1992, "Fault Detection and Diagnosis in Low Speed Rolling Element Bearings Part I: The Use of Parametric Spectra," *Journal of Mechanical Systems and Signal Processing*, Vol. 6, No. 4, pp. 135-151.

Mielnik, M. J., 1990, "Reasoning about The Quality of Data in On-line Expert Systems," in *Proc. of 2nd Inter. Machinery Monitoring and Diagnosis Conf.*, Las Vegas, Nevada, Oct., pp. 1-7.

Milne, R., 1990, "AMETHYST: An Expert System for The Diagnosis of Rotating Machinery," in *Proc. of COMADEM 90: The 2nd Inter. Congr.*, Brunel Univ., UK, 16-18 July, pp. 287-293.

Minami, E., and Hirata, T., 1991, "An Expert System for Large Scale Fault Diagnosis in

Steel Manufacturing," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 1311-1318.

Minsky, M., 1991, "Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy," *AI Magazine*, Vol. 12, No. 2, pp. 35-51.

Mussi, S., and Morpurgo, R., 1990, "Acquiring and Representing Strategic Knowledge in The Diagnosis Domain," *Expert Systems*, Vol.7, No.3, pp. 157-165, August.

Nakayama, S. et al, 1992, "Knowledge Acquisition of Strategy and Tactics Using Fuzzy Neural Networks," in *Proc. of IEEE Inter. Joint Conf. on Neural Networks*, Baltimore, June 7-11, pp. II751-756.

Narazaki, H., and Raiescu, A. L., 1993, "An Improved Synthesis Method for Multilayered Neural Networks Using Qualitative Knowledge," *IEEE Trans. on Fuzzy Systems*, Vol. 1, No. 2, pp. 125-137, May.

Nie, J., and Linkens, D. A., 1992, "Fuzzy Reasoning Implemented by Neural Networks," in *Proc. of IEEE Inter. Joint Conf. on Neural Networks*, Baltimore, June 7-11, pp. II702-707.

Nielsen, R. H., 1987, "Counterpropagation Networks," in *Proc. of The IEEE 1st Inter. Conf. on Neural Networks*, San Diago, CA, June.

Nigrin, A., 1993, *Neural Networks for Pattern Recognition*, The MIT Press, Cambridge, Massachusetts.

Parsaye, K., and Lin, K. Y., 1987, "An Expert System Structure for Automatic Fault Tree Generation fro Emergency Feedwater System for Nuclear Power Plants," in *Proc. of Western Conf. on Expert Systems*, California, June, pp. 25-30.

Pattipati, K. R. et al, 1986, "A Heuristic Search/Information Theory Approach to Near-Optimal Diagnostic Test Sequencing," in *Proc. of the 1986 IEEE Inter. Conf. on System, Man and Cybernetics*, Geogia, Oct. 14-17, pp. 230-235.

Pao, Y. H., 1989, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, New York.

Pau, L. F., 1981, "Failure Diagnosis and Performance Monitoring," *Control and System Theory Series*, Vol. 11, Marcel Dekker, New York.

Pazzani, M. J., 1989, "Explanation-Based Learning of Diagnostic Heuristics: A Comparison of Learning from Success and Failure," in *Proc. of 1989 Annual AI Systems in Government Symposium*, Washington, D.C., March, pp. 164-168.

Pedrycz, W., 1988, *Fuzzy Control and Fuzzy System*, Research Studies Press, Taunton, Somerset, England.

Persad, P., and Wei-Muddeen, H., 1991, "Knowledge-Based Expert System for Training in Diagnostics and Repair of Consumer Electronics," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 202-207.

Petersen, D. G., 1990, "Development of An Integral PC-Based Predictive Maintenance Expert System for Automated Machine Diagnostics," in *Proc. of 2nd IMMDC*, Los Angeles, pp. 8-12.

Piety, K., and Corley, J. E., 1989, "Development of an Expert System to Diagnose Machinery Vibration Problems," in *Proc. of 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 523-527.

Poggio, T., and Girosi, F., 1990, "Networks for Approximation and Learning", *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1481-1497.

Ramakrishna, K. et al, 1992, "Development of Computerized On-Line Condition Monitoring of Rotating Machinery - A Custom Based Approach", in *Proc. of The Inter. Conf. on Rotating Machine Dynamics*, Venice, April, pp. 76-83.

Ramamoorthy, C. V., and Sheu, P. C., 1988, "Object-Oriented Systems," *IEEE Expert*, pp. 9-15, Fall.

Ramu, S. A., 1991, *Lecture Notes for Knowledge-Based Systems in Dynamics and Design*, Concordia University.

Randall, R. B., 1982, "A New Method of Monitoring Gear Faults," *J. of Mechanical Design*, Trans. of The ASME, Vol. 4, pp. 259-267, April.

Rangwala, S., and Dornfeld, D., 1990, "Sensor Integration Using Neural Networks for Intelligent Toll Condition Monitoring," ASME *Journal of Engineering for Industry*, Vol. 112, No. 3, pp. 219-228.

Rao, R., Au, J., and Griffiths, B., (Eds.), 1990, *Condition Monitoring and Diagnostic Engineering Management, Proc. of COMADEM 90*, July 16-18, T.J. Press Ltd, Padstow, GB.

Rao, V. B., and Rao, H. V., 1993, *C++ Neural Networks and Fussy Logic*, MIS: Press, New York.

Reiter, R., 1987, "A Theory of Diagnosis from The First Principle," *Artificial Intelligence*, Vol. 32, pp. 57-95.

Rich, E., 1990, "Expert Systems and Neural Networks Can Work Together," *IEEE Expert*, pp. 5-7, Oct..

Ritter, H., and Schulten, K., 1989, "Combining Self-Organizing Maps", in *Proc. of Inter. Joint Conference on Neural Networks*, Wash., DC, June, pp. 499-502.

Rosenman, M. A. et al, 1986, "Expert System Applications in Computer-aided Design," *Computer-Aided Design*, Vol. 18, No. 10, pp. 546-551.

Rumelhart, D. E., and McClelland, J. L., 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, "Learning Representations by Back-propagating Errors," *Nature*, No. 323, pp. 533-536.

Saavedra, P., Cabeza, C., and Juan, L. S., 1990, "Experience from Fault Diagnosis of Critical Rotatory Machines," in *Proc. of 2nd Inter. Machinery Monitoring and Diagnosis Conf. (IMMDC)*, 'os Angeles, CA, pp. 538-542.

SAE E-32 Committee, 1981, *Aircraft Gas Turbine Engine Monitoring Systems*, SP-478, Society of Automotive Engineers, Inc., Warrendale, PA, May.

Sankar, K. P., and Mitra, S., 1992, "Multilayer Perceptron, Fuzzy Set, and Classification," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 683-697, Sept..

Sasaki, Y., and Tomita, T., 1984, "A Study on Oil Whip for a Shaft Supported by Multi-Bearings," in *Proc. of the 3rd Inter. Conf. on Vibrations in Rotating Machinery*, IMechE, C292/84, pp. 17-21.

Schiltz, R. L., 1989, "Accuracy in Estimating Forcing Element Forcing Frequencies," in *Proc. of 1st Inter. Machinery Monitoring and Diagnosis Conf.*, Las Vegas, Nevada, Sept., pp. 139-144.

Schindler, P. et al, 1989, "Vibrations Monitoring: A Real Time Expert System for Complex Machinery Management under Operational Constraints," in *Proc. of 12th Biennial ASME Conf. on Mechanical Vibration and Noise*, Montreal, Oct., pp. 77-83.

Schönwälder, J., Hofmann, M., and Langendörfer, H., 1991, "Fault Diagnosis of a Swage Plant," in *Proc. of the IEEE 7th Conf on AI for Applications*, Miami Beach, Feb. 24-28, pp. 120-123.

Sejnowski, T., and Rosenberg, C. R., 1986, "NETtalk: A Parallel Network That Learns to Read Aloud," *Johns Hopkins Univ. Technical Report JHU/EECS-86/01*.

Senjen, R. et al, 1993, "Hybrid Expert Systems for Monitoring and Fault Diagnosis," in *Proc. of IEEE 9th Conf. on AI for Applications*, Orlando, March 1-5, pp 235-241.

Serridge, M., 1989, "Ten Crucial Concepts Behind Trustworthy Fault Detection in Machine Condition Monitoring," in *Proc. of 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 722-727.

Sharfer, G., 1976, *Mathematical Theory of Evidence*, Princeton Univ. Press, Princeton, NJ.

Shi, X. Z. et al, 1990, "The Inference Recognition of Signal Sources and Neural Network Model for The Fault Diagnosis," in *Proc. of The 2nd IMMDC*, Los Angeles, CA, pp. 215-219.

Shives, T. R. and Mertaugh, L. J., (Eds.), 1986, *Detection, Diagnosis and Prognosis of Rotating Machinery to Improve Reliability, Maintainability, and Readiness Through the Application of New and Innovative Techniques*, Cambridge Univ. Press, GB.

Simpson, P. K., 1992, "Fuzzy Min-Max Neural Networks - Part 1: Classification," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 776-786, Sept..

Simpson, P. K., 1993, "Fuzzy Min-Max Neural Networks - Part 2: Clustering," *IEEE Trans. on Fuzzy Systems*, Vol. 1, No. 1, pp. 32-45, Feb..

Smets, P. 1990, "The Combination of Evidence in The Transferable Belief Model," *IEEE Trans. on Pattern Analysis and Machine Intelligent.* Vol. 12, No. 5, pp. 447-458, May.

Sohre, J. S., 1980, "Turbomachinery Problems and Their Correction", *Sawyer's Turbomachinery Maintenance Handbook.* Vol. 2, Chapter 7, Connecticut USA.

Sood, A. K., Friedlander, C. B., and Fahs, A. A., 1985, "Engine Fault Analysis: Part I - Statistical Methods," *IEEE Trans. on Industrial Electronics*, Vol. IE-32, No. 4, pp. 294-300, Nov..

Specht, D. F., 1991, "A General Regression Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 2, No. 6, pp. 568-576, Nov..

Stephanou, H. E., and Sage, A. P., 1987, "Perspectives on Imperfect Information Processing," *IEEE Trans. on System, Man, and Cybernetics*, Vol. SMC-17, No. 5, pp. 780-798, Sept/Oct..

Stronach, A. F. et al, 1984, "Condition Monitoring of Rolling Element Bearings," *Condition Monitoring '84*, Pineridge Press Ltd, UK.

Stuart, J. D., and Vinson, J. W., 1985, "TURBOMAC: An Expert System to Aid in The Diagnosis of Causes of Vibration-Producing Problems in Large Turbomachinery," in *Proc. of ASME Conf. on Computers in Engineering*, Boston, MA, Aug. 4-8, pp. 319-325.

Tailor, A., 1988, "MXA - A Blackboard Expert System Shell," *Blackboard Systems*, R. Engelmore and T. Morgan, ed., Addison-Wesley Publishing Co..

Tang, H. et al, 1991, "The Principle of Cepstrum and Its Application in Quantitative Fault Diagnostics of Gears," DE-Vol. 38, *Modal Analysis, Modelling, Diagnostics and Control - Analytical and Experimental.* ASME, pp. 141-144.

__, 1989, *NEXPERT OBJECT Product Description*, TATA Consultancy Services.

Taylor, J. I., 1980, "Identification of Bearing Defects by Spectral Analysis," *Journal of Mechanical Design*, Trans. of the ASME, Vol. 102, pp. 109-204.

Tian, H. and Yun, D., 1991, "A More Complete Knowledge Acquisition Methodology," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp 812-819.

Togai, M. and Watanabe, H., 1986, "Expert Systems on a Chip: An Engine for Real-Time Approximate Reasoning," *IEEE Expert*, Fall, pp. 55-62.

Tranter, J., 1989, "The Fundamentals of, and The Application of Computers to, Condition Monitoring and Predictive Maintenance," in *Proc. of 1st IMMDC*, Las Vegas, Nevada, Sept., pp. 394-401.

Ueno H. et al, 1991, "Knowledge Modelling of Digital System and Model-Based Reasoning for Trouble Shooting," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 2162-2171.

VerDuin, W., 1990, "Solving Manufacturing Problems with Neural Nets, Neural Nets + Expert Systems = Manufacturing Solutions," *Automation*, pp. 54-58, July.

Wang, S. S., 1989, "AI and Expert Systems for Diagnosis," in *Proc. of 1st Inter. Machinery Monitoring and Diagnosis Conf.*, Las Vegas, Nevada, Sept., pp. 516-522.

Wang, S. S., 1990, "Diagnostic Expert Systems for Industry", in *Proc. of The 2nd Inter. Machinery Monitoring and Diagnostic Conf.*, Los Angeles, Oct., pp. xvii-xxxi.

Wang, L., Porter, A. L., and Cunningham, S., 1991, "Expert Systems: Present and Future," *Expert Systems with Applications*, Pergamon Press, Vol. 3, pp. 383-396.

Waterman, D. A., 1986, *A Guide to Expert Systems*, Addison-Wesley, The USA.

Watts, W. A., and VanDyke, D. J., 1989, "A Portable, Automated Machine Condition Diagnostic Program Using Order Normalized Vibration Data," in *Proc. of 1st Inter. Machinery Monitoring and Diagnosis Conf.*, Las Vegas, Nevada, Sept., pp. 637-643.

Weiskopf, F. B., Arcella, F. G., Lin, J. S., and Newman, R. W., 1990, "A Hybrid Approach to Machinery Monitoring and Diagnosis," in *Proc. of The 2nd Inter. Machinery Monitoring and Diagnostic Conf.*, Los Angeles, Oct., pp. 25-30.

Whitehall, B. L., 1989, "Knowledge Intensive Empirical Learning Using Multiple Levels of Background Knowledge," in *Proc. of 1989 Annual AI Systems in Government Symposium*, Washington, D.C., March, pp. 157-163.

Wright, P. K., and Bourne, D. A., 1988, *Manufacturing Intelligence*, Reading, MA: Addison-Wesley.

Wu, H. J., 1991, "Expert Systems for Tool Wear Detection via Neural Network

Approach," in *The World Congress on Expert Systems Proc.*, Orlando, Florida, Dec., pp. 2180-2185.

Xistris, G. D., 1979, *Technical Report on The Reliability of Mechanical Systems Subjected to Random Loading Environment*, EP79-R-9, École Polytechnic de Montréal, Feb..

Xistris, G. D., and Giangrande, I., 1991, "A Microprocessor-Based Vibration Collector to Monitor Condition of Shipboard Machinery," in *Proc. of 3rd Inter. Machinery Monitoring and Diagnosis Conf. (IMMDC)*, Las Vegas, Nevada, Dec. 9-12, pp. 114-118.

Xu, J. and Zhang, S., 1990, "Fuzzy Diagnosis Method in Machinery Condition Monitoring and Failure Diagnosis," in *Proc. of The 2nd Inter. Machinery Monitoring and Diagnostic Conf.*, USA, Oct., pp. 93-96.

Xu, J., Liu, Y., and Zhang, S., 1991, "A Distribute Monitoring and Diagnostic System with AI," in *Proc. of The 3rd Inter. Machinery Monitoring and Diagnostic Conf.*, Las Vegas, Dec. 9-12, pp. 81-85.

Yan, Y. and Shimogo, T., 1990, "Application of the Impulse Index in Rolling-Element Bearing Fault Diagnosis," *J. of Mechanical Systems and Signal Processing*, Vol. 6, No. 2, pp. 167-177.

Yu, X. and Biswas, G., 1992, "A Multi-level Diagnosis Methodology for Complex Systems," in *Proc. of the IEEE 8th Conf. on AI for Applications*, Monterey, March 2-6, pp. 81-87.

Zadeh, L. A., 1988, "Fuzzy Logic," *IEEE Computer*, pp. 83-92, April.

Zador, P. L., 1982, "Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension", *IEEE Transactions of Information Theory*, Vol. IT-28, pp. 139-149.

Zhang, S., Liu, Y. and Chen, K., 1986, "A New Approach of Monitoring Gear Malfunctions by Microprocessor," in *Proc. of the 4th IMECO Inter. Symposium on Technical Diagnostics*, Yugoslavia, Oct., pp. 3.20-3.23.

Zhang, S., 1990, "Rules for Design The Hardware and Software of Machinery Monitoring and Diagnostic Systems," in *Proc. of The 10th National Conf. on Measurement Tech.*, Tianjin, China, Oct. 8-10, pp. 283-293.

Zhang, S. and Sankar, T. S., 1993, "New Methodology and Architecture for A Machinery Diagnostic Expert System," in *Proc. of 1st CWC ICIA '93*, Beijing, Aug., pp. 135-140.

Zhang, S. and Sankar, T. S., 1994a, "Machine Condition Identification by SOM Algorithm," in *Proc. of 1994 IMACS & IEEE Inter. Symposium on Signal Processing, Robotics and Neural Networks*, Lille, France, April, pp. 88-93.

Zhang, S. and Sankar, T. S., 1994b, "A New Machinery Diagnostic Expert System with Learning Ability," in *Proc. of CSME Forum*, Montreal, June, pp. 683-688.

Zhang, S. and Sankar, T. S., 1994c, "Function Estimation for Multiple Indices Trend Analysis Using Self-Organizing Mapping," in *Proc. of ETFA '94 Seiken/IEEE Symposium on Emerging Technologies & Factory Automation*, Tokyo, Nov. 6 10, pp. 160-165.

Zhang, S, Ganesan, R., and Sankar, T. S., 1995a, "RMD-KBS: A Coupled Numerical and Symbolic Processing System for Machinery Monitoring and Diagnostics," in *Proc. of 8th Inter. Congress on Condition Monitoring and Diagnostic Engineering Management (COMADEM '95)*, Kingston, pp. 289-296.

Zhang, S., Ganesan, R., and Sankar, T. S., 1995b, "A New Self-Organizing Mapping Algorithm for Regression Problems," in *Proc. of World Congress on Neural Networks '95*, Washington, D.C., July 17-21.

Zhang, S., Ganesan, R., and Sankar, T. S., 1995c, "Machinery Condition Identification by Self-Organizing Neural Networks," ASME *Journal of Engineering for Industry*, in press.

Zhang, S., Ganesan, R., and Sankar, T. S., 1995d, "Multivariable Trend Analysis for System Monitoring through Self-Organizing Neural Networks," ASME *Journal of Dynamic Systems, Measurement, and Control*, in press.

Zhang, S., Ganesan, R., and Sankar, T. S., 1995e, "A New Self-Organizing Mapping Algorithm for Regression Problems of System Monitoring and Diagnostics," *IEEE Transactions on Neural Networks*, (under review).

Zhao, M. et al, 1989, "KDS: The Knowledge-Based Diagnostic System Shell," in *Proc. of 1st IMMDC, Las Vegas*, Nevada, Sept., pp. 669-673.

Zhao, M. and Luo, Z., 1992, "An Expert System of Crack Monitoring and Diagnosis for Rotating Machines," in *Proc. of The Inter. Conf. on Rotating Machine Dynamics*, Venice, April 28-30, pp. 85-91.

Zhuge, Q, Lu, Y., and Yang, S., 1990, "Non-Stationary Modelling of Vibration Signals for Monitoring The Condition of Machinery," *Mechanical Systems and Signal Processing*, **4**(5), pp. 355-365.

Zurada, J. M., 1992, *Artificial Neural Systems*, West Publishing Company, St. Pal, Minnisuda.

APPENDIX

A.1 Definition of the Diagnostic Indices

The monitoring indices defined below are used in the RMD-KBS. They have been obtained from the literature during the knowledge acquisition. All of the following expressions have been encoded in C++ as the signal processing routines.

Digitized Vibration Signal

The digitized vibration signal is expressed as a set of data samples, that is called herein the time domain vibration signal. A set of the original data samples is denoted as $\{x_1^*, x_2^*, ..., x_n^*\}$, with n denoting the number of data samples (data length) in the set. the data samples are obtained from analogue signal when the sampling interval Δt is used in the digitizing process. The sampling frequency, f_s, is defined as

$$f_s = \frac{1}{\Delta t} \quad (samples \ / \ second) \tag{A.1.1}$$

The spectrum of vibration data is expressed as a set of magnitudes of the amplitudes at different frequencies (the Fourier coefficients) as $\{a_1, a_2, ..., a_m\}$, with m being the length of the set. The maximum frequency of a spectrum is equal to or less than half of the sampling frequency. The magnitudes at certain frequencies are directly taken as the indices, that show the frequency components of signals that correspond to several machine faults.

Mean Value

The Mean value is calculated from the time domain signal using the expression given below.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{(A.1.2)}$$

In the signal processing, the Mean value is the first index to be calculated. After the Mean value is obtained from a set of data samples, the mean component will be removed from the original data samples before extracting the other indices from the data. Hence, in the definition of the following indices, the set of data samples, i.e. $\{x_i\}$, i=1, 2, ..., n is used to denote the zero-mean data.

Peak-to-Peak Value (*PP*)

$$PP = x_{max} - x_{min} \quad \text{(A.1.3)}$$

where

$$x_{max} = \max\{x_1, x_2, ..., x_n\}$$

$$x_{min} = \min\{x_1, x_2, ..., x_n\}$$

Maximum Absolute Value ($|x|_{max}$)

$$|x|_{max} = \max\{|x_1|, |x_2|, ..., |x_n|\} \quad \text{(A.1.4)}$$

Absolute Mean Value (*AX*)

$$AX = |\bar{x}| = \frac{1}{n}\sum_{i=1}^{n}|x_i| \tag{A.1.5}$$

Variance (*VX*)

$$VX = \frac{1}{n}\sum_{i=1}^{n}(x_i^2 - \bar{x}) \tag{A.1.6}$$

Standard Deviation (*DX*)

$$DX = \sqrt{VX} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^2 - \bar{x})} \tag{A.1.7}$$

Root Mean Square (*RMS*)

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2} \tag{A.1.8}$$

Kurtosis Value (*K*)

$$K = \frac{1}{n}\sum_{i=1}^{n}x_i^4 \;/\; (\frac{1}{n}\sum_{i=1}^{n}x_i^2)^2 \tag{A.1.9}$$

Crest Factor (*CR*)

$$CR = \frac{|x|_{max}}{RMS} \tag{A.1.10}$$

Impulse Factor (*IP*)

$$IP = \frac{|x|_{max}}{\bar{x}} \tag{A.1.11}$$

Arithmetic Mean (*AM*)

$$AM = 20\ \log\{\ (\ \frac{1}{m}\sum_{i=1}^{m} a_i\ )/10^{-5}\ \} \tag{A.1.12}$$

Geometric Mean (*GM*)

$$GM = \frac{1}{m}\sum_{i=1}^{m}\{\ 20\ \log(\ \frac{a_i}{\sqrt{2}\ 10^{-5}})\ \} \tag{A.1.13}$$

Matched Filter Root Mean Square Value (*MM*)

$$MM = 10\ \log\{\ \frac{1}{m}\sum_{i=1}^{m}(\frac{a_i}{a_{i(ref)}})^2\ \} \tag{A.1.14}$$

where $a_{i(ref)}$, i=1, ..., m, are the magnitudes of the reference spectrum.

The magnitudes at certain frequencies of the FFT spectrum are considered as diagnostic indices. The magnitude at the rotating frequency of the machine is denoted as

$F1$. The magnitudes at the second and third harmonics of the machine rotating frequency are denoted by $F2$ and $F3$ respectively. The maximum magnitude at a frequency around 42% of the rotating frequency is denoted as Fh.

To analyze the vibration signal that is generated by rolling element bearings, the magnitudes at the following frequencies are of interest.

1) The frequency of ball passing inner race:

$$f_{inn} = \frac{N}{2} f_1 \left(1 + \frac{Bd}{Pd}\cos\varphi\right) \quad (Hz) \tag{A.1.15}$$

2) The frequency of ball passing outer race:

$$f_{out} = \frac{N}{2} f_1 \left(1 - \frac{Bd}{Pd}\cos\varphi\right) \quad (Hz) \tag{A.1.16}$$

3) The frequency of the ball spin:

$$f_{spin} = \frac{Pd}{2Bd} f_1 \left(1 - \frac{Bd^2}{Pd}\cos^2\varphi\right) \quad (Hz) \tag{A.1.17}$$

where N: the number of balls or rollers,

f_1: the rotational frequency,

Bd: the diameter of the balls or rollers,

Pd: the pitch diameter,

ϕ: the contact angle.

The magnitudes at both the frequencies of ball passing inner race and outer race are used as monitoring indices for the diagnosis of rolling element bearings. They are denoted by Fi and Fo respectively.

In some cases the values of Bd, Pa or ϕ may not be known. The following expressions can then be used to approximate the above frequencies (Schiltz, 1989).

1) The frequency of ball passing inner race:

$$f_{inn} = f_1 \left(\frac{N}{2} + 1.2\right) \quad (Hz) \tag{A.1.18}$$

2) The frequency of ball passing outer race:

$$f_{out} = f_1 \left(\frac{N}{2} - 1.2\right) \quad (Hz) \tag{A.1.19}$$

3) The frequency of the ball spin:

$$f_{spin} = f_1 \left(0.2N - \frac{1.2}{N}\right) \quad (Hz) \tag{A.1.20}$$

where only the value of the bearing parameter N, i.e. the number of balls or rollers, is required to be known.

A.2 The Knowledge Editor and Confidence Factors of LEVEL5

The expert system development tool, LEVEL5 OBJECT™, Version 2.2 (Level5, 1990a; 1990b), has been employed for the implementation of RMD-KBS. This tool has been designed and developed by Information Builders, Inc., 1250 broadway, New York, NY 10001. It is a KBS shell that can be used for the development of rule-based system. It combines the versatility of object-oriented techniques with powerful, multiple interfacing strategies in a flexible windowing environment. The main features of LEVEL5 are listed below.

1) In LEVEL5, the information and knowledge can be represented by objects and their instances, relational database models (for database accessing), agenda for (goal-driven inference), IF-THEN-ELSE rules and demons with confidence factors, grouped rules, WHEN NEEDED and WHEN CHANGED methods.

2) The inference engine supports both forward and backward chaining, and also mixed-mode chaining. It promises certain level controls on the inference procedure, such as *to fire the first* or *fire all*. It provides two commands, ACTIVATE and ESTABLISH, that can be used to call external programs from a rule during the reasoning. The external programs may be specifically written to communicate with LEVEL5, or suitable programs that run under Windows.

3) This tool provides multiple-windows environment for inputting and editing the knowledge, that consists of Object Editor, Rule/Demon/Method Editor, Display/Forms Editor, Agenda Editor, Windows Editor, Database Editor and Interfaces, Graphical Knowledge Tree, Session Monitor/Debugger, Values Report, History Report, External Program Interfaces and so on. They are shown in Figures A.2.1 to A.2.12 respectively.

4) It supports Microsoft® Windows™ type user interface that can be developed using the LEVEL5 Windows Editor and Display Editor.

The Confidence Factors of LEVEL5: In LEVEL5, any attribute can have a confidence (certainty) factor (CF) that quantifies the uncertainty in the information it caries. In addition, any rule or demon can also have a confidence factor that accounts for the uncertainty in the relationships. Confidence factors are used by LEVEL5 as an indicator of the reliability of an attribute's value in the session context. The values of a CF can be any integer number between 0 to 100, with 100 representing the TRUE value and 0 representing the FALSE value. It can also be -2 representing UNKNOWN value, or -1 representing UNDETERMINED value. For example, confidence factors of attributes can be given as: attribute 1 CF 80, attribute 2 CF 75, and attribute 3 CF 90. The CFs are assigned during initialization or in the reasoning session. A rule with a CF may be written as

RULE	Example
IF	attribute 1
AND	attribute 2
OR	attribute 3
THEN	conclusion CF 80

The relationship between the attributes and the conclusion in the above rule is believed to be 80 percent true. At the time of evaluating a rule in the form of IF fact THEN conclusion CF, the final CF of the conclusion is determined by

$$\text{Final CF} = (\text{CF of the fact} \times \text{CF of the rule}) / 100 \qquad \text{(A.2.1)}$$

For example, if a rule is given as IF a (CF 75) THEN b CF 80, then the above calculation results in a conclusion b that has a CF= 60. If the operators AND, OR, NOT are used in a rule, the calculation of the final CF is based on the following rules:

$$\text{IF a AND b: CF} = \min\{\text{CF of a, CF of b}\} \qquad \text{(A.2.2)}$$

$$\text{IF a OR b: CF} = \max\{\text{CF of a, CF of b}\} \qquad \text{(A.2.3)}$$

$$\text{IF NOT a: CF} = 100 - (\text{CF of a}) \qquad \text{(A.2.4)}$$

The confidence factors are used by LEVEL5 as an indicator for determining the order in which the inference engine will evaluate the members of the attribute's rule group. For an attribute in a KBS, there may be a group of rules that can be fired according to the value of the same attribute. In this case, the rule with the highest CF value will be fired first. For example, among the two rules, IF a THEN c CF 80, and IF b THEN c CF 90, the latter will be evaluated first. There is also a confidence threshold using which the inference engine can determine the results of rule evaluation. By default, the confidence threshold is assigned as 50. If the final CF of a conclusion obtained from a rule is less than this threshold, then the verification of that conclusion is taken to have failed.

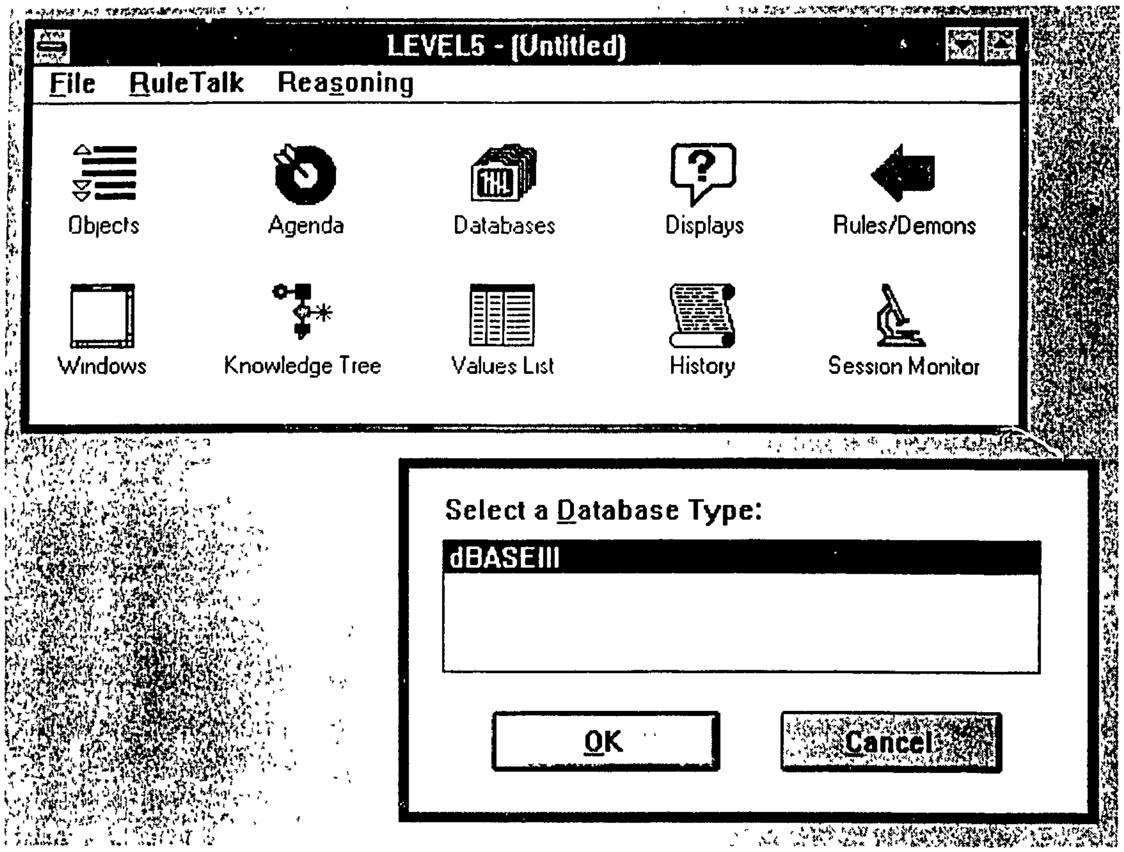


Fig. A.2.1 The Knowledge Editor of LEVEL5 OBJECT shell and its Database Editor.

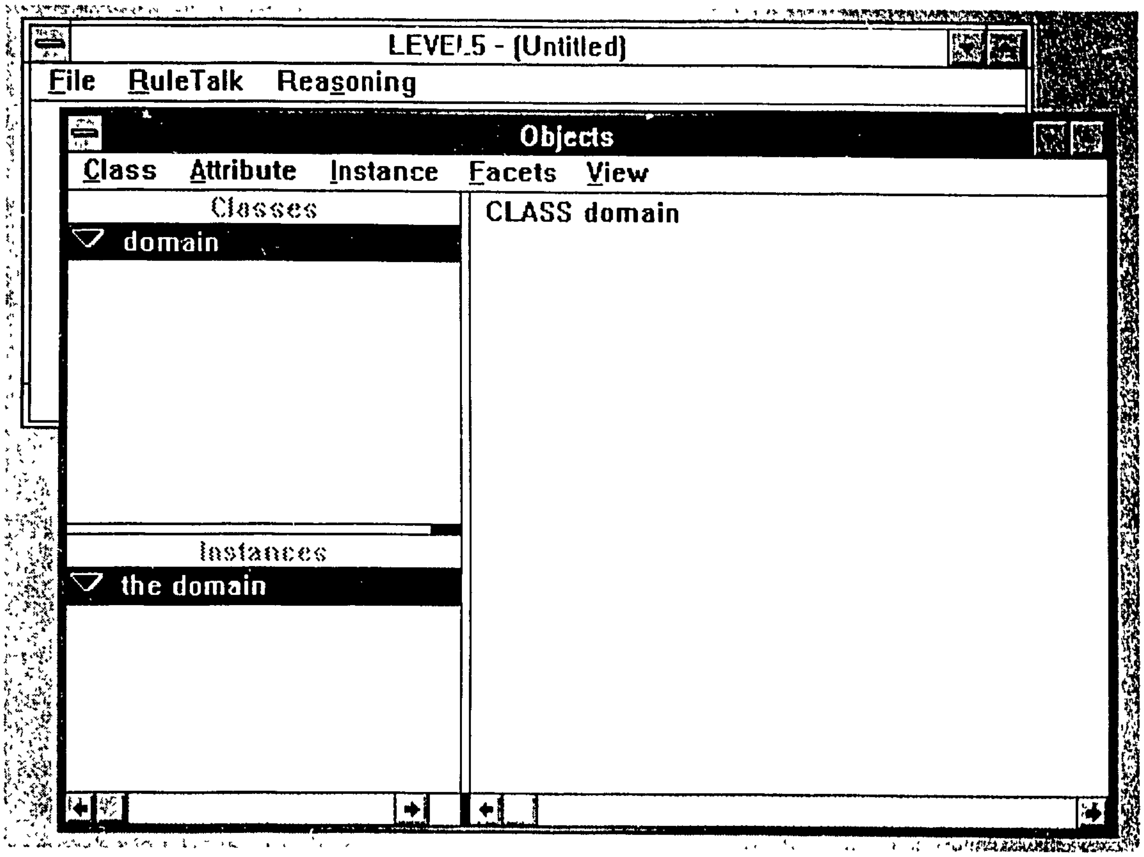


Fig. A.2.2 The Object Editor of LEVEL5, that is used to declare and modify classes and instances.

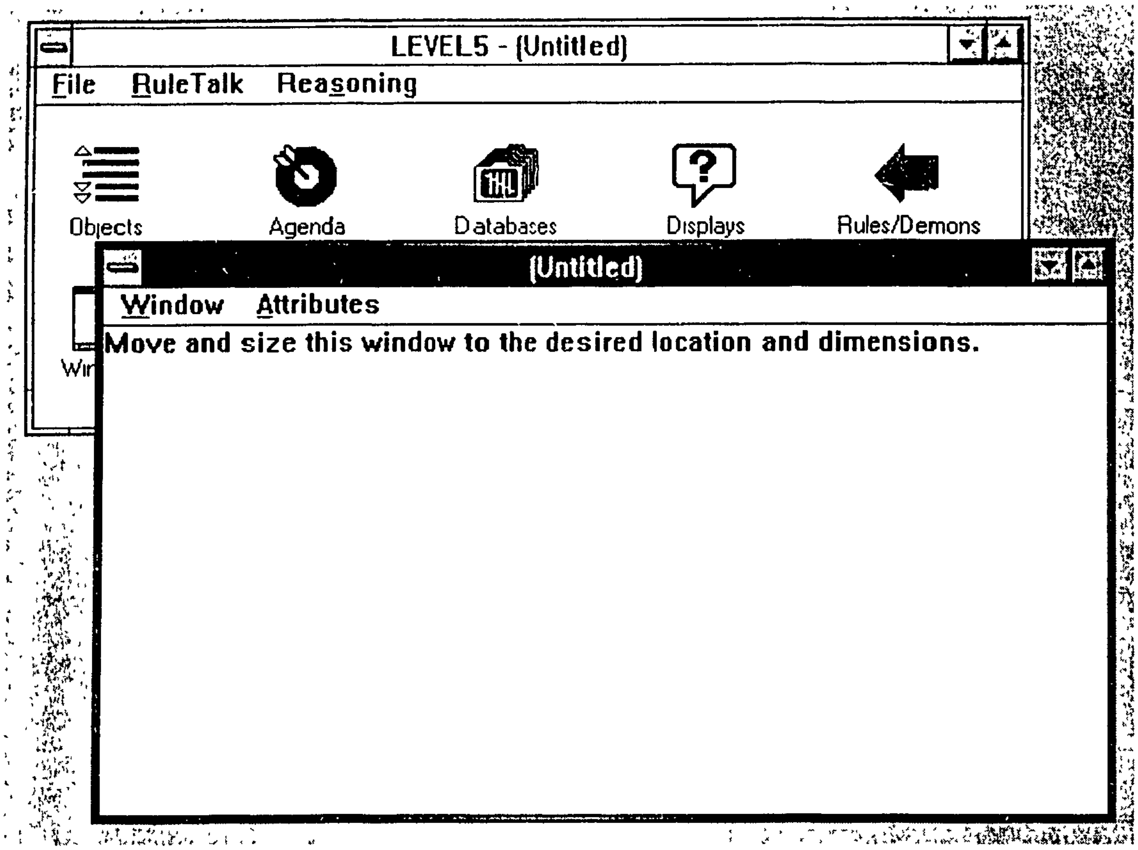


Fig. A.2.3 The Windows Editor of LEVEL5. By using it a KBS developer can specify the style, size and location of a window of the user interface under design.

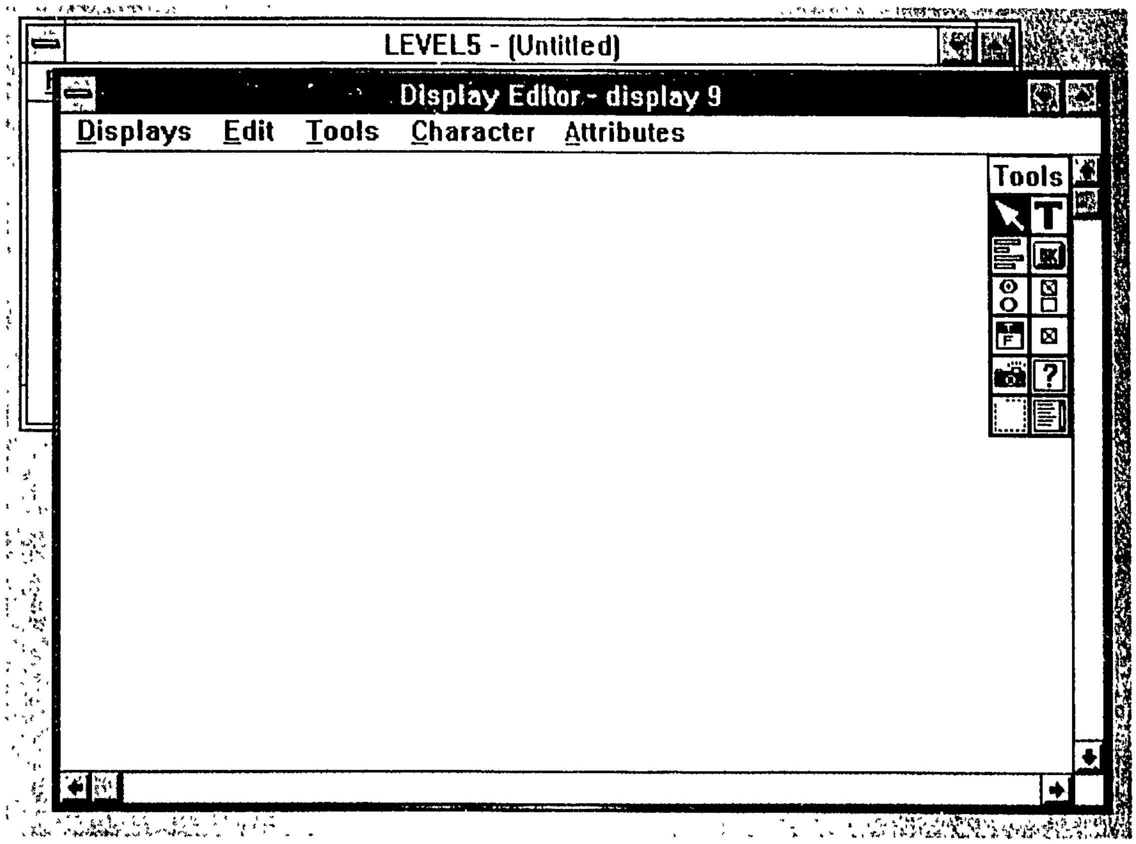


Fig. A.2.4 The Display / Forms Editor of LEVEL5, that is used for the KBS developer to create a new display.

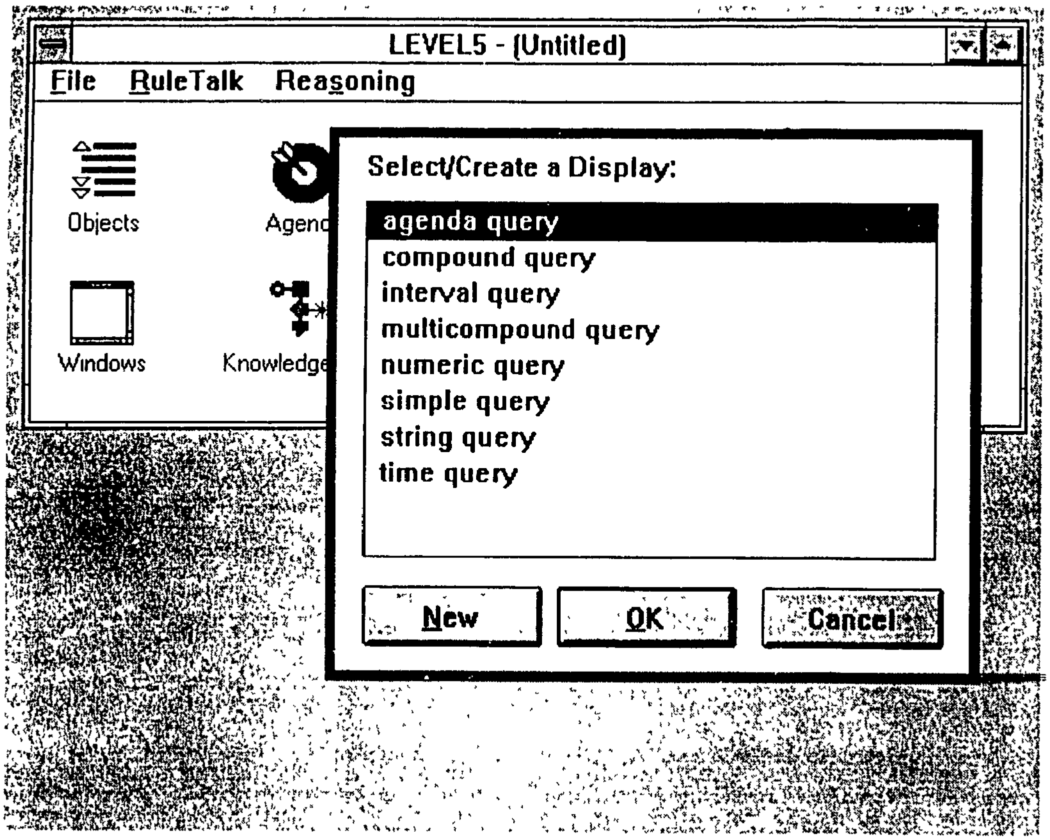


Fig. A.2.5 The first step in using the Display / Forms Editor: to select a pre-designed form in the list.

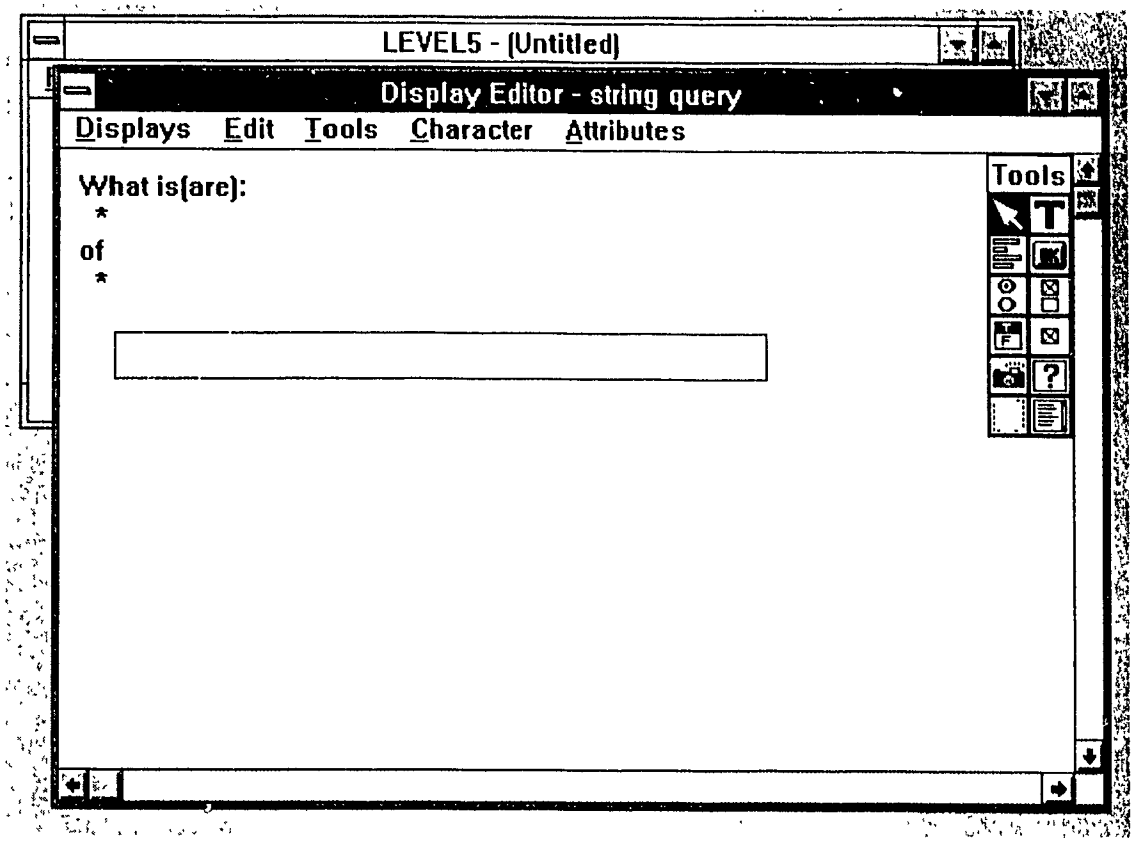


Fig. A.2.6 The second step in using the Display / Forms Editor: to define a system provided form for string query from the end-user of the KBS.

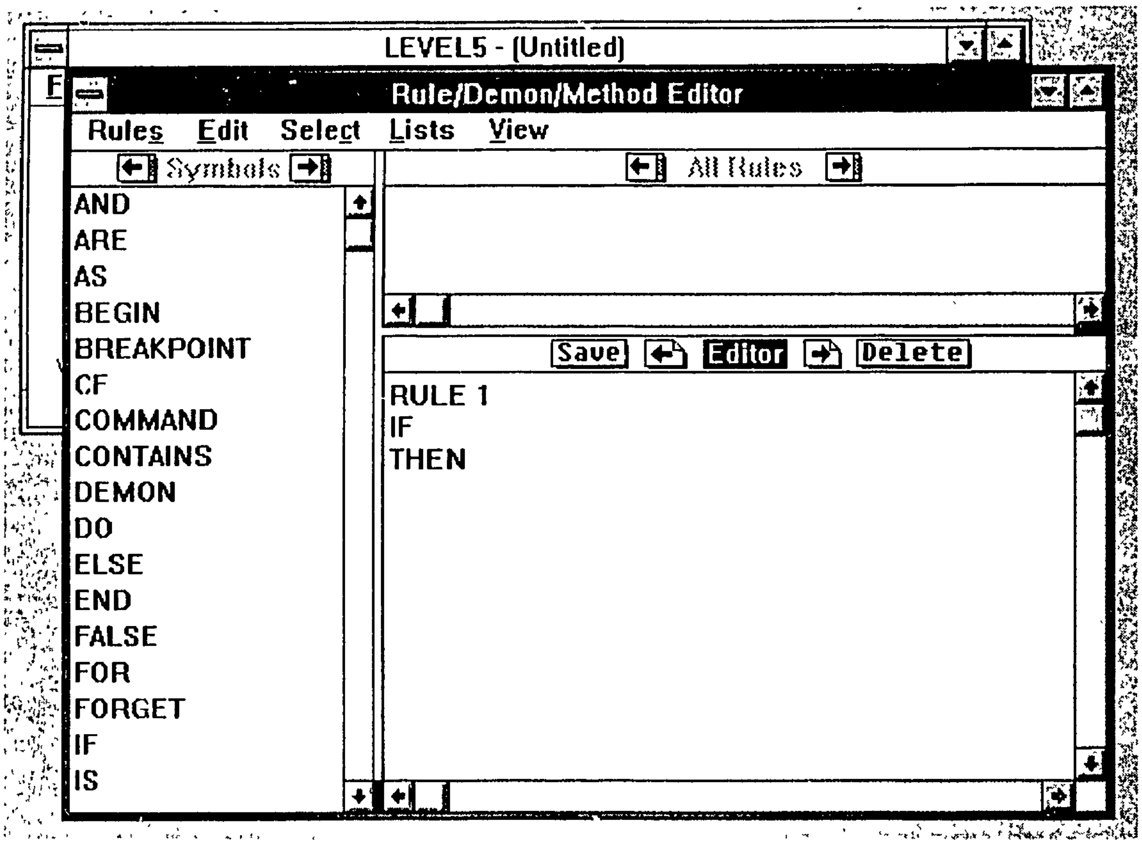


Fig. A.2.7 The Rule / Demon / Method Editor of LEVEL5.

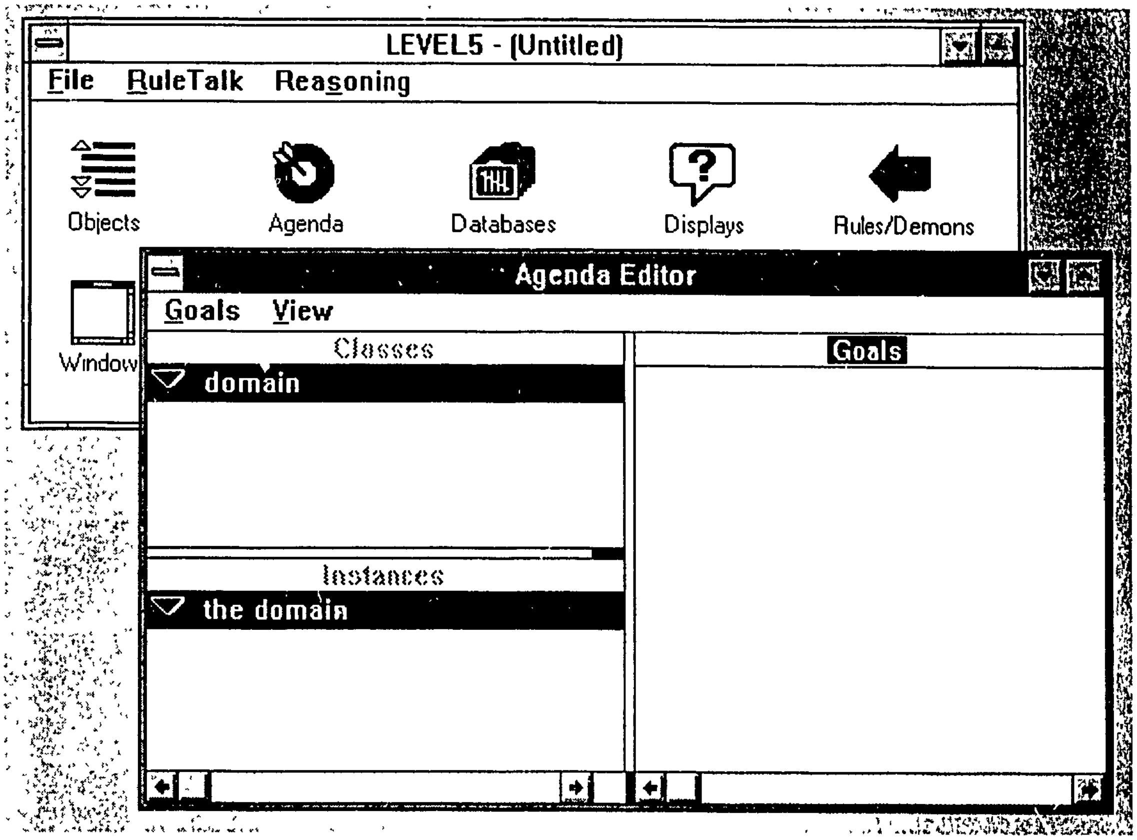


Fig. A.2.8 The Agenda Editor of LEVEL5, that is used to select a number of class attributes as agenda.

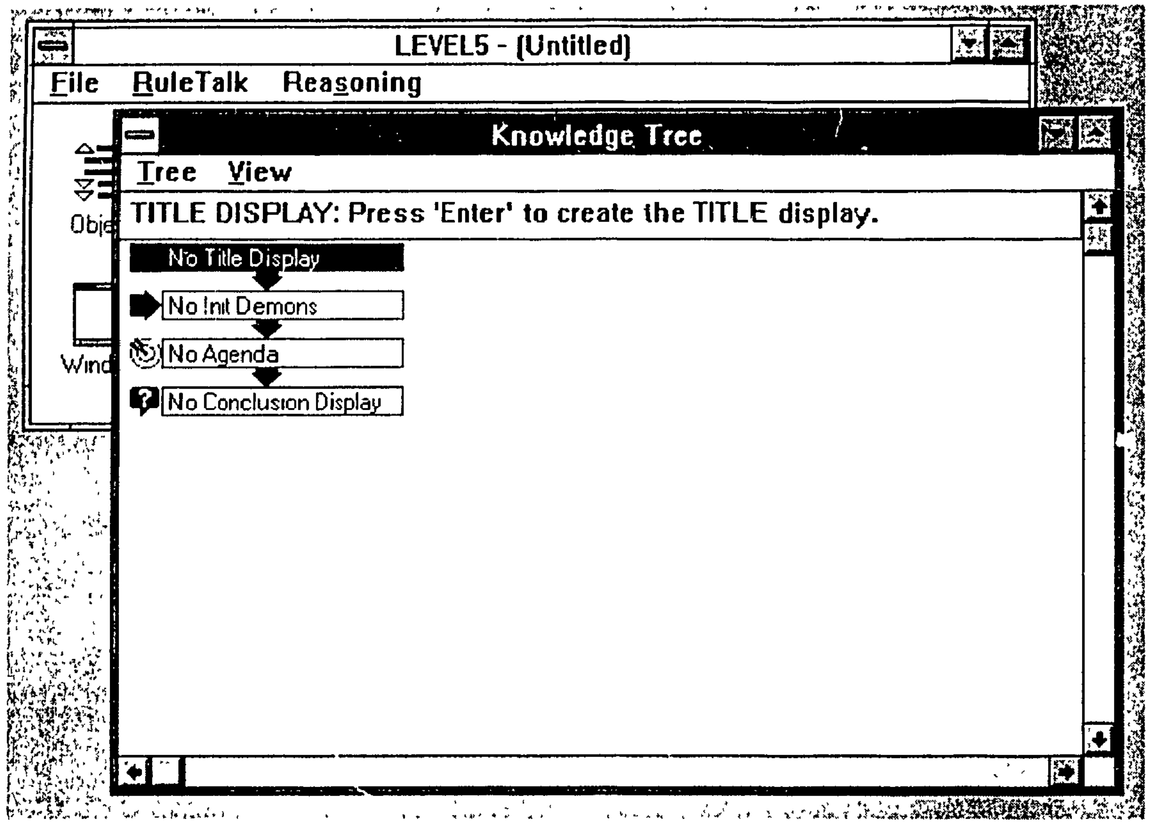


Fig. A.2.9 The Knowledge Tree of LEVEL5 which is an interactive environment for building, debugging, and maintaining the knowledge bases.

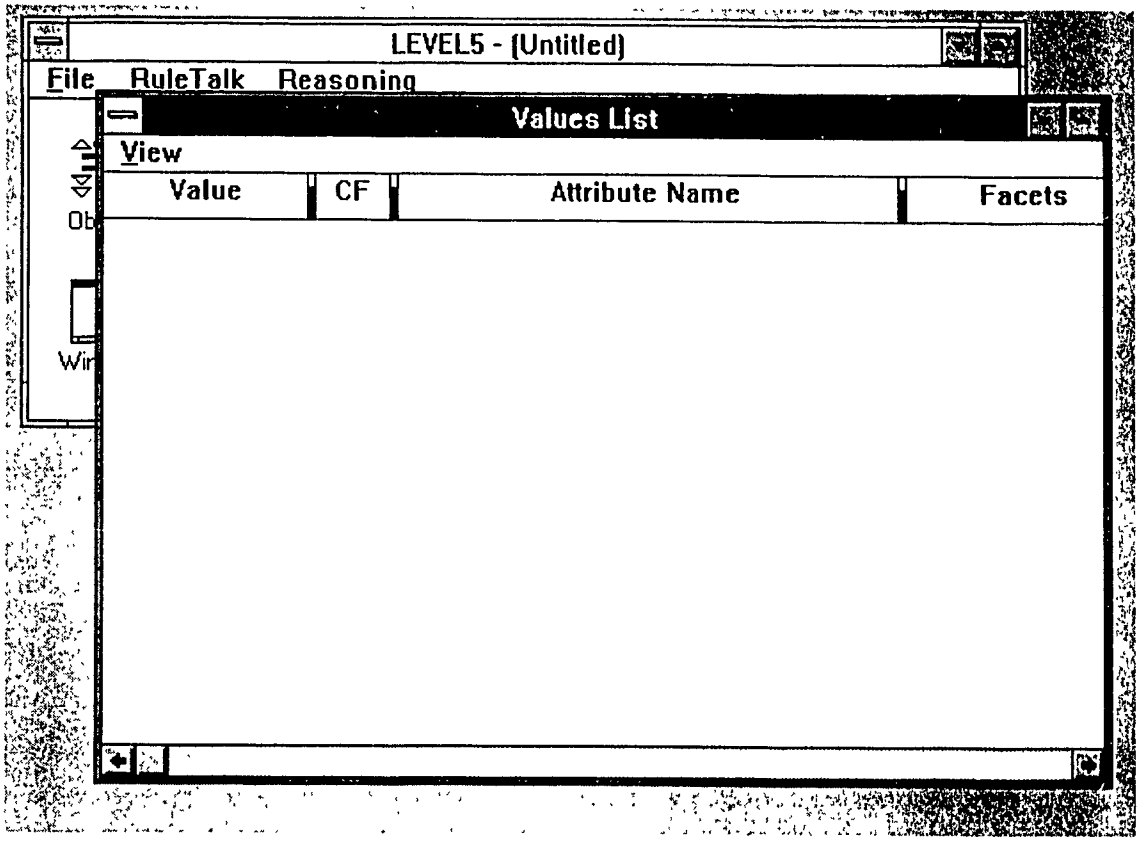


Fig. A.2.10 The Values List of LEVEL5, that catalogues the current values, confidence factors (CF), facets, and attribute names in the knowledge base.

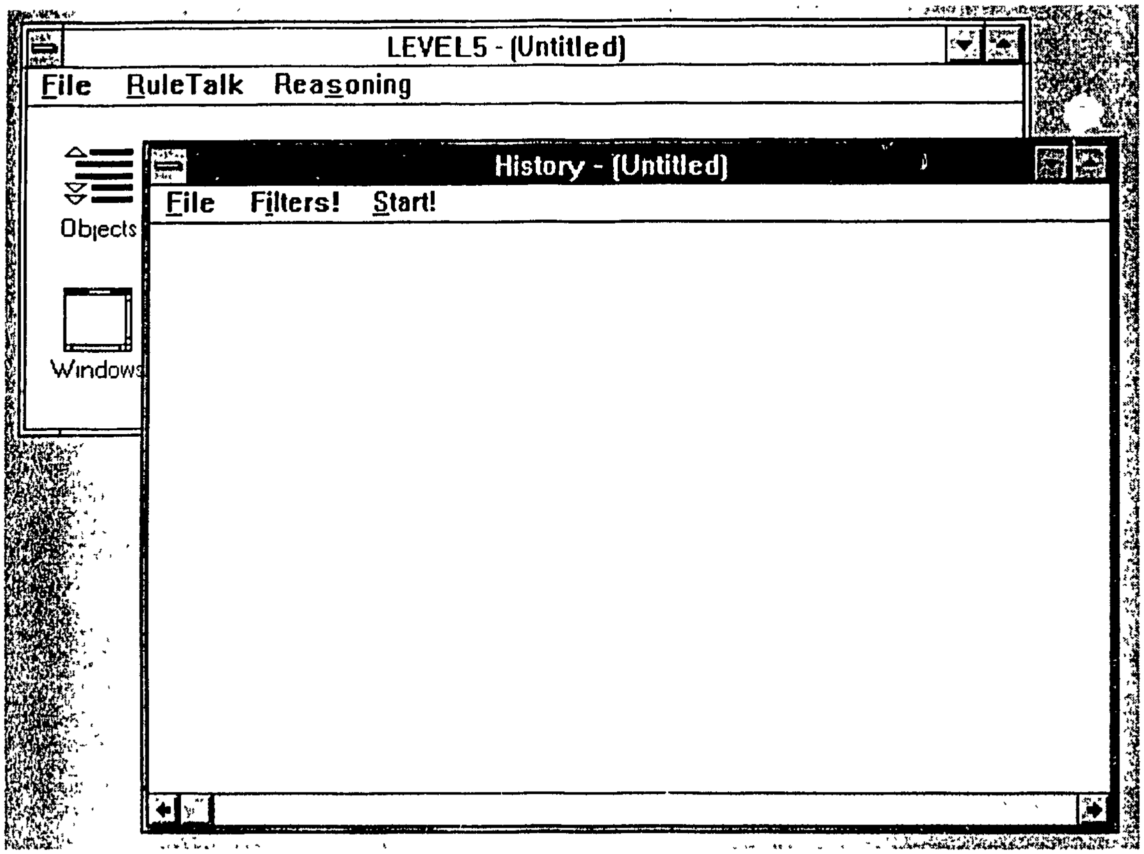


Fig. A.2.11 The History Report of LEVEL5, which creates a text file that is a chronological record of the events of a processing session.

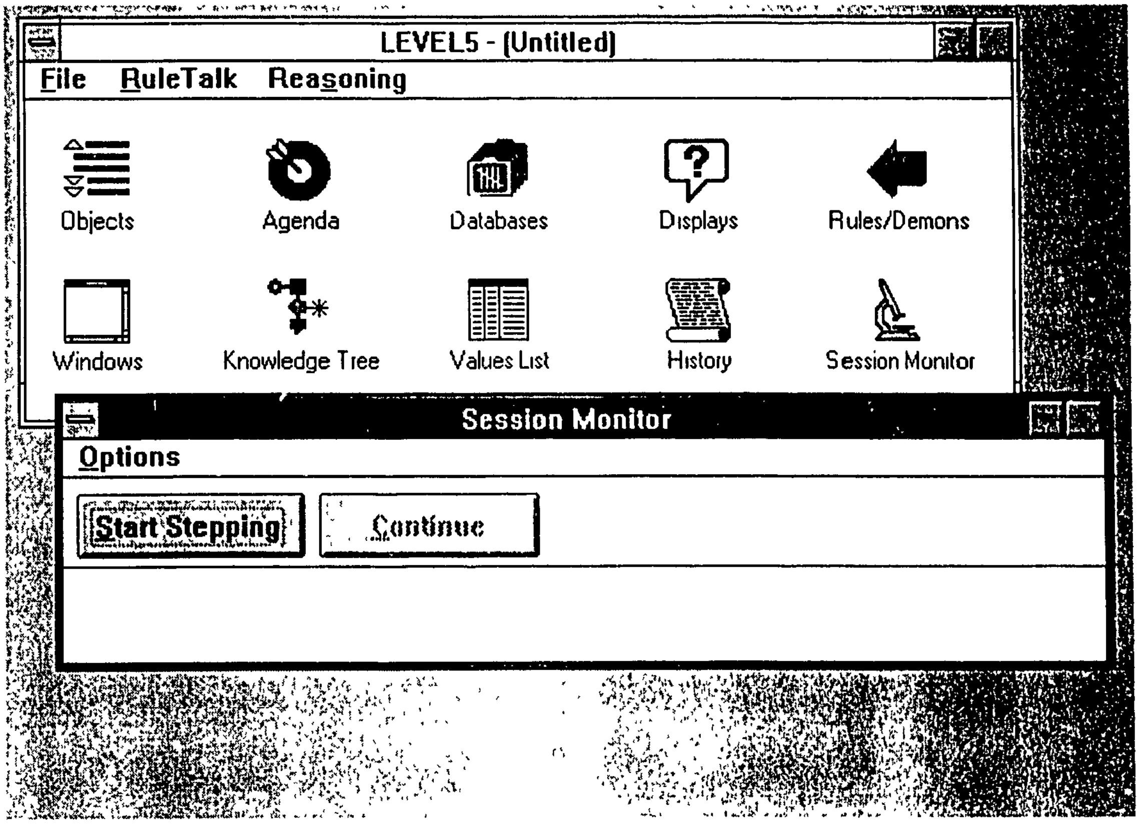


Fig. A.2.12 The Session Monitor of LEVEL5, that follows and displays the current status of the reasoning season at run-time.