CUTTING TOOL CONDITION MONITORING OF THE TURNING PROCESS USING ARTIFICIAL INTELLIGENCE

R.G. Silva

PhD Thesis

1997

CUTTING TOOL CONDITION MONITORING OF THE TURNING PROCESS USING ARTIFICIAL INTELLIGENCE

by

Rui Gabriel Araújo de Azevedo Silva

A thesis submitted in partial fulfilment of the
requirements of the University of Glamorgan/Prifysgol Morgannwg
for the degree of Doctor of Philosophy

June 1997

The University of Glamorgan

To

My Parents, Brothers

and Girlfriend

Abstract

This thesis relates to the application of Artificial Intelligence to tool wear monitoring. The main objective is to develop an intelligent condition monitoring system able to detect when a cutting tool is worn out. To accomplish this objective it is proposed to use a combined Expert System and Neural Network able to process data coming from external sensors and combine this with information from the knowledge base and thereafter estimate the wear state of the tool.

The novelty of this work is mainly associated with the configuration of the proposed system. With the combination of sensor-based information and inference rules, the result is an on-line system that can learn from experience and can update the knowledge base pertaining to information associated with different cutting conditions. Two neural networks resolve the problem of interpreting the complex sensor inputs while the Expert System, keeping track of previous success, estimates which of the two neural networks is more reliable. Also, mis-classifications are filtered out through the use of a rough but approximate estimator, the Taylor's tool life equation.

In this study an on-line tool wear monitoring system for turning processes has been developed which can reliably estimate the tool wear under common workshop conditions. The system's modular structure makes it easy to update as required by different machines and/or processes. The use of Taylor's tool life equation, although weak as a tool life estimator, proved to be crucial in achieving higher performance levels. The application of the Self Organizing Map to tool wear monitoring is, in itself, new and proved to be slightly more reliable then the Adaptive Resonance Theory neural network.

Table of Contents

List of Illustrations

List of Tables

Acknowledgements

Throughout this research I have called upon the assistance of a large number of people both within and outside the University. Firstly I would like to express my gratitude for the guidance and encouragement offered by my Director of Studies, Dr. Steven Wilcox who has excelled as a supervisor, both as a constant source of advice, support and encouragement throughout the project, and more specifically in the reading of countless draft chapters which has helped enormously in the preparation of this thesis. Also, I would like to express my gratitude to Professor Robert Reuben and Mr. Keith Baker for their support and valuable ideas which helped on the development of the Thesis.

I would like to acknowledge the financial support provided by the Department of Mechanical and Manufacturing Engineering - University of Glamorgan, and in particular the Head of Department, Professor John Ward.

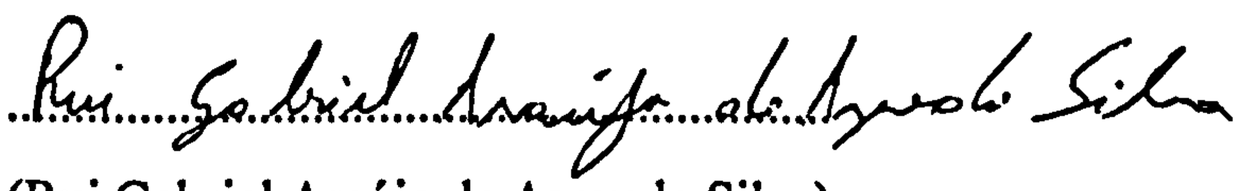
Further thanks are due to Clive Monks, Higher Technical Officer, for his assistance during experiments and valuable contribution towards my better understanding of practical machining. The enthusiastic support of Gareth Betteney and Gregory Jameson, Senior Technical Officers, was vital towards the completion of the experimental set-up.

The help and support of colleagues and members of staff at the University of Glamorgan is also gratefully acknowledged.

R.G. Silva 1997

Author's Declaration

This Thesis has not been nor is currently being submitted for the award of any other degree or similar qualification.



(Rui Gabriel Araújo de Azevedo Silva)

Nomenclature

b	Slope of linear regression equation	
b_{ij}	Resulting pattern of activation, ART2	
C	Constant in Taylor's tool life equation	
C_w	Constant for machining parameters	
d	Depth of cut	mm
d_{ijk}	Euclidean distance between input i and output node (j,k)	
e	Natural logarithmic base	
E	Modulus of elasticity	
f	Feed rate	mm/rev
f_n	Natural frequency of vibration	Hz
F_0	Initial machining force	N
F_1, F_2	Processing stages of the ART2	
F	Cutting force	N
F_f	Force in the feed direction	N
F_{jk}	Winning frequency of SOM output neurone	
F_r	Force in the radial direction	N
F_t	Force in the tangential direction	N
$h(t)$	Continuos time series	
h_k	Discrete time series	
$H(t)$	Fast Fourier series	
H_n	Discrete Fast Fourier series	
KT	Crater depth	
l	Length of overhang	m
m	Number of output nodes for the ART2	
M^2	Number of output nodes for the SOM	
n	Exponent in Taylor's tool life equation	
N	Number of input features	
$N^{1/2}$	Initial value for ART2 weights	
Np	Number of previous samples of historical data	
P_{ART2}	ART2 outlier counter	
P_{NN}	Number of miss-classifications to date (general)	
P_{SOM}	SOM outlier counter	

r	Linear regression coefficient	
R	Radius of cut	m
r_f	Linear regression correlation coefficient of feed force	
r_h	Linear regression correlation coefficient of previous P_{NN} samples	
r_t	Linear regression correlation coefficient of tangential force	
S_i	Sample reference number	
t	Time	s
t_1	Undeformed chip thickness	mm
t_2	Chip thickness	mm
t_{ji}	Threshold of weights between F_1 and F_2 stages, ART2	
T	Tool life	min
T_{jk}	Input pattern to F_2, from F_1, *ART2*	
T_y	Threshold value	
V	Cutting speed	m/min
Vc	Speed between chip and tool	m/min
VB_B	Average flank wear	mm
VB_i	VB_B for sample S_i	mm
VB_N	Length of wear notch	mm
VB_{REF}	Flank wear threshold value	mm
V_{out}	Voltage output	v
X	Input feature vector	
X_i	Input signal feature component	
w_{ijk}	SOM connection weight	
W	Wear land size (general)	
w_i, v_i, u_i, q_i, p_i	Auxiliar variables within the ART2 algorithm	
W_{ijk}	SOM weight between input i and output neurone (j,k)	
α	Rake angle	
$\varepsilon(t)$	Learning rate for the SOM	
π	Pi	3.1415....
ϕ	Shear angle	
ω	Angular speed	radians/s

1. Introduction

Conventional machining systems rely heavily on human operators for monitoring the process, taking the appropriate action in the event of a problem, inspecting the quality of the product, controlling the process and material handling. However, in recent years, the manufacturing industry has been directed towards automated, untended operation with the goal of achieving better product quality and greater overall productivity and reliability. Thus, the implementation of an intelligent machining system which can perform specified machining operations without detailed input from human operators under harsh and unpredictable shop environment becomes increasingly important.

Manufacturing industries and their customers are now demanding substantial increases in flexibility, productivity, and reliability from process machines as well as increased quality and value of their products. One important strategy to support this goal is sensor-based, real-time control of key characteristics of both machines and products, throughout the manufacturing process. Also, according to the US 1993 National Critical Technologies Panel, intelligent processing equipment has an essential role in the achievement of world-class manufacturing capabilities. The fundamental concept is that the manufacturing process include the ability to sense the desired characteristics or properties of a product and has enough local intelligence to control those properties.

The successful automation of machining operations relies, to a great extent, on the ability of artificial systems to recognise process abnormalities and initiate corrective action. In the absence of human operators, this function has to be performed with intelligent decision-making systems which are able to interpret incoming sensor information and decide on the appropriate control action. Intelligent decision-making systems are expected to replace the knowledge, experience, and the combined sensory and pattern recognition abilities of human operators. Successful implementation of these different tasks depends on two factors; first, the quality of information obtained from the monitoring sensor, and second, the techniques used to process this information in order to make decisions. Sensing strategies for unmanned machining should aim at integrating both these factors, thereby allowing for a system that can successfully replace the abilities of the human operator. Hence, the performance of the sensor system is a crucial factor in an intelligent machining system. Thus, the development of a sensor system which is sensitive enough to produce accurate information on the machining process and can operate under harsh and unpredictable shop environments is essential.

To implement intelligent tool wear sensing systems in an automated manufacturing environment, it is helpful to examine how such a function is performed by a human machinist. Human operators probably detect the occurrence of tool wear by observing the machining operation and evaluating the resulting sensory information (for example, visual; such as the colour of the machined chip, audio; such as the sound generated by the cutting action of the tool and olfactory; such as the smell of the hot workpiece and coolant). The sensory information is associated with experience-based memory triggers and also experience of tool life in similar situations. A decision is then made as to whether the tool wear level warrants interruption of the process, checking of the tool condition and then, if necessary, initiation of tool changing procedures. The process is one of pattern recognition in which disparate, noisy and incomplete sensory data patterns are used to make a decision as to the amount of tool wear. Human pattern recognition is a highly developed and a poorly understood characteristic, and the task of emulating it on a computer is a formidable one.

The performance of the cutting process deteriorates seriously as the cutting tool wears out, requiring a tool change. However, tool life is very difficult to predict and has a very widely scattered distribution (Oxley, 1989), making it impossible to set-up generally acceptable tool change policies. Thus, an adaptable system that learns on-line upon experiments and which can identify the state of the tool during the machining operation should be incorporated into the machine tool to improve productivity.

1.1 Motivations for the Development of a Monitoring System

There is limited industrial use of monitoring systems as human skills are still necessary help to manage the complex machining process. Existing systems for monitoring and control are tailor-made and one possible reason for the failure of such systems is the application of unsophisticated models and in many cases the lack of correlation between the measured variable and the process variable of interest (Tonshoff *et al.*, 1988).

Since the advent of numerically controlled (NC) machine tools, which provided the physical means to control the cutting process, the goal of complete automation of machine tools has seemed much more promising. The development of computer numerically controlled (CNC) machine tools has made it feasible to apply additional levels of control to the cutting process by utilising readily available, inexpensive, and reliable computing power. However, control of any

process requires sensing capabilities to provide essential feedback data. One of the most significant pieces of feedback data in a cutting process is the state of tool wear (Danay and Ulsoy, 1987).

Tool wear data could be used for two major objectives:-

1. Detection of tool failure due to excessive wear.
2. Control of the rate of tool wear. First, tool failure detection can be achieved by the on-line monitoring of the tool, thereby, eliminating the function of the machine operator as the tool monitor. Second, the rate of tool wear can be controlled by the manipulation of the cutting conditions, thereby providing:
   i. Better co-ordination in the production line by planning the machine shut down for tool changes.
   ii. More efficient use of the tool.
   iii. Significant savings of time that is usually spent on the more frequent tool changes caused by unreliable estimates of tool life (Zhou and Wysk, 1992).

A technique which is currently being investigated which may contribute to the improvement of tool wear monitoring is the hybrid system. Hybrid systems, comprising Neural Networks (NNs) and Expert Systems (ESs), are an active area of research in the field of AI (Laffey *et al.*, 1988; Medsker, 1994). In these hybrid systems, a neural network is used either for knowledge expression or for knowledge acquisition, that is, to make conclusions from input data or to learn from experience and may be dynamically modified. The Expert System comprises a set of rules that allow it to infer results from acquired information, but cannot be dynamically modified. NNs are more complex and the information encoded by them is difficult to interpret and does not always provide results that are easy to understand. Results from NNs are not predictable to the same extent as those obtained with ESs and they are more likely to fail than ESs (Medsker, 1996). The combination of both methodologies may find uses in an enormous number of fields as an artificial intelligence tool. The primary difference between an automated machining system and intelligent one is that the intelligent system is capable of making decisions based on significant information about the state of the system

The concept of using condition monitoring for automated small batch production equipment is now rapidly gaining ground in manufacturing industry (Davies, 1994), due to the wide availability of powerful and highly reliable electronic sensing systems, together with advances in computer technology and its associated software.

1.1.1 Economic Factors

Since metal cutting operations constitute a large percentage of current manufacturing activity, it is important to consider economic factors.

The costs associated with running a machining centre can be sub-divided into the following categories:-

1. Unscheduled down-time 11% (Machinery[Editor], 1989) of which insert failure accounts for 7% (Dan and Mathew, 1990).
2. Tooling costs 5-30% (Cser *et al.*, 1993).

To reduce the costs that are associated with cutting tool inserts it will be necessary to monitor the level of wear and then schedule insert changes that optimise the life of the tool. It has been estimated that by the use of monitoring systems it is possible to increase efficiency by 10-65% (Tonshoff *et al.*, 1988).

As a result of these factors, there is a strong research effort directed at automating the machining process. Present efforts are aimed at developing reliable sensor technology for detecting factors such as chip form, tool condition, workpiece roughness, machine vibration and bearing failure.

1.2 Existing Approaches

The problem of on-line tool wear measurement has been investigated by numerous researchers (e.g. Taglia *et al.*, 1976; Choi *et al.*, 1990; Dan and Mathew, 1990; Wang and Dornfeld, 1992). The methods proposed can be categorised into two groups: direct and indirect. Direct methods, as the name implies, measure the tool wear by either evaluating the worn surface by visual inspection (Pedersen, 1990) or the material loss of the tool by radiometric techniques (Micheletti *et al.*, 1976). The main difficulty with using optical methods is their limited application to cases where the surface of the tool is visually accessible during the operation (Pedersen, 1990). Radiometric techniques have been proposed and demonstrated in the laboratory, these techniques have not been implemented in production due to requirements for special preparation of the tool and potential hazards due to radioactivity (Dan and Mathew, 1990).

Indirect methods, on the other hand, are based on utilising signals such as force or torque, temperature, tool vibration, acoustic emission, or noise emission (König *et al.*, 1972; Dornfeld *et al.*, 1993). These techniques determine the relationship between the measured parameters and tool wear, thus enabling prediction. Some approaches rely on a detailed mechanistic model of the cutting process (Kannatey-Asibu, 1985) while others use empirical relationships between the measured variable and tool wear (Danai *et al.*, 1992). The mechanistic approach has contributed greatly to the basic understanding of the cutting process, while the empirical approach has been useful for specific tool-workpiece combinations and constant cutting conditions. However, both the mechanistic and empirical approach have certain limitations when applied to on-line tool wear estimation.

The mechanistic approach, which relies on the mathematical description of the physics of cutting, assumes certain wear mechanisms such as diffusion, abrasion, and adhesion as being responsible for tool wear. Due to the inherent complexity of the cutting process, and our incomplete understanding of it, this approach is limited in applicability at the present time. Moreover, since the coefficients and exponents of these models change with tool-workpiece combinations and cutting conditions (Fenton and Oxley, 1976; Lin *et al.*, 1982; Koren *et al.*, 1986; Danai *et al.*, 1992), extensive off-line testing is required for each case. Another limitation in the utilisation of the mechanistic approach is the lack of appropriate sensors. For example, most models developed by this approach (mechanistic models) emphasise in the relationship between temperature and tool wear (Kannatey-Asibu, 1985). However, in the absence of a practical temperature sensor these models are limited in applicability.

The empirical approach, on the other hand, relies on the observed relationship to estimate tool wear. Although new approaches based on a multi-sensor strategy have recently been proposed (e.g. Rangwala and Dornfeld, 1990), most practical methods based on the empirical approach rely on a single sensor to detect tool failure or estimate tool wear. The empirical methods for tool wear estimation usually consider the relationship between two variables such as flank wear and force (Danai *et al.*, 1992), and therefore fail to separate the effect of other variables involved in the process, for example the effect of feed rate on force. This usually results in an erroneous estimation of tool wear as other process variables will affect the measured signal. The empirical approach, like the mechanistic approach, requires accurate information and therefore, extensive off-line testing is required for each tool-workpiece combination and each specific set of cutting conditions.

Artificial intelligence is a tool that can be used to aid the automatic interpretation of data obtained from a tool during the wear process. Neural networks and Expert Systems have been in use for the last two decades in the field of process monitoring (Rangwala and Dornfeld, 1987; Burke, 1990; Ezugwu *et al.*, 1995), and already proved successful in fields such as ultrasonic image interpretation (Hopgood *et al.*, 1993) and in medicine (Anthony, 1993). The use of such techniques in the field of tool wear monitoring (Au *et al.*, 1989; Jantunen *et al.*, 1995; Du *et al.*, 1995) is being investigated but, in common with the mechanistic and empirical approaches, the methods used so far do not separate the effect of other variables such as cutting conditions.

Finally, most of the monitoring systems proposed to date lack the "intelligence" to deal with sporadic outlying data that may arise from workshop conditions or sensor faults.

1.3 Appropriateness of a Hybrid System for Tool Wear Monitoring

In view of the foregoing discussion, it is felt appropriate to use a hybrid system for the following reasons:-

1. Heuristic rules are conventionally used to monitor the cutting process, and these often change because of changes in tool or workpiece material. Therefore, a hybrid system that can easily be modified is appropriate.
2. The monitoring task may not be difficult for a "trained expert", but the constant turnover of personnel makes it very difficult to maintain staff expertise. Also, because there are so many variables, and the "rules" are subject to interpretation by the individual monitoring the cutting process, results are not consistent.
3. Because the problem might not be reduced to a small number of rules, it is appropriate for development and installation in a microcomputer through the use of a hybrid neural network - Expert System.

1.4 Justification for the Development of a Hybrid System

For the reasons given in the previous sections it is now possible to highlight some of the most important achievements through the use of a hybrid system:-

1. Improved performance: Since the system will apply the rules in a more consistent manner than most of the machine operators, especially new ones, the results obtained

should be less likely to omit requirements and should provide for more consistent and efficient tool wear monitoring.

2. Faster configurations: The system will produce a recommended configuration much faster than "field experts". Since the person in charge of configurations will have more confidence in the configuration recommendations, less time should be spend revising the system set-up before approval.
3. Reduced requirements for training personnel: Since the system can be operated by non-experts, who can use the explanation capability to learn on the job, machine operators' training requirements are significantly reduced.
4. Easy maintenance: It is easy to modify the rules and the goals in an Expert System based hybrid system.
5. Increased productivity: Since less time is spent on setting the machine for a particular batch job, production becomes more efficient. Increasing the accuracy of tool wear monitoring enables tools to be used throughout their entire life. Also, significant savings will be made by reducing defects in the machined pieces.

1.5 Scope of the Thesis

In this study, an architecture for an on-line tool wear monitoring system based on two neural networks and an Expert System are developed and evaluated. The system is developed and tested using a particular set of sensors and machining conditions but the method is extensible to other sensors and/or operations. The classification ability of the neural networks is combined with knowledge embedded in an expert system in order to make reliable and accurate estimates of tool wear. The approach presented in this thesis is similar to that which might be used implicitly by a human operator as it combines empirical knowledge of tool wear with classification of sensor based information.

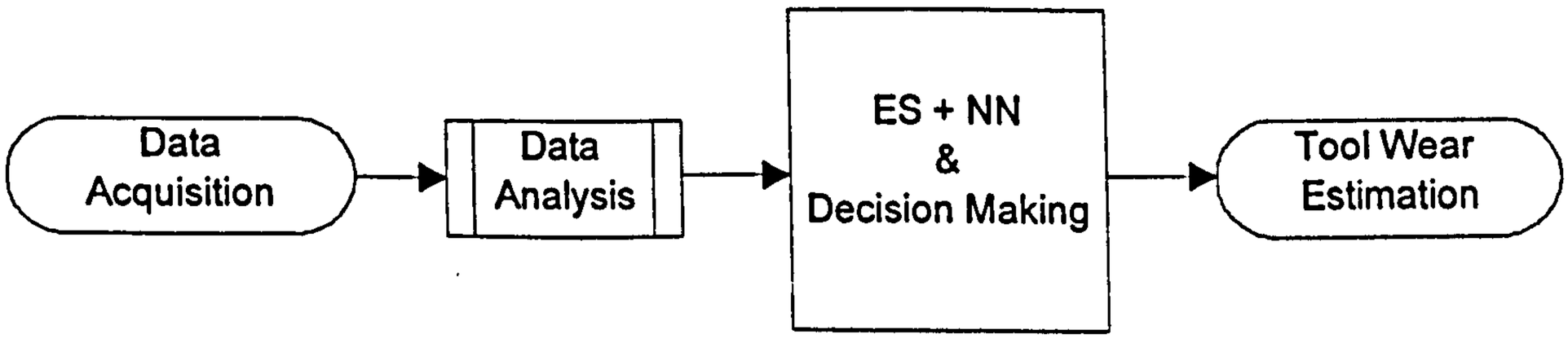


Figure 1: Decision making scheme

The signal processing, neural network interpretation and decision making scheme is shown in Figure 1. The system employs sensors that measure the sound emission, vibration, force and spindle current from the machining operation, with the signals being processed in the frequency and time domains. The resulting features are then presented to two previously trained neural networks, a Self Organising Map and a network based on Adaptive Resonance Theory, to estimate the wear state of the cutting tool in conjunction with the Expert System. At the decision-making stage, an Expert System encoded with empirical tool life data is combined with the networks outputs to reach an estimation which is consistent with the sensed outputs and machining experience.

1.6 Thesis Structure

This Thesis is organised into eight chapters and 4 appendices. Chapter 1 briefly reviews the context of the problem of tool wear monitoring and presents the motivations for the current work.

Chapter 2 details the approach in the light of previous research in the field of tool wear monitoring and related subjects. The chapter starts with a mechanical description of the process of single point cutting and then reviews the most important processing and data interpretation techniques which will be used later. Throughout the chapter, the current approach is compared to other published work on tool wear monitoring.

Chapter 3 gives details of the experimental design as well as experimental procedures. First, the turning centre and related equipment are described, then the positioning of the sensors is indicated and justified. Finally, the procedures adapted in the two main sets of experiments are given, these being the experiments pertaining to the behaviour of the machine/tool and the analysis of signal length effect on feature performance.

Chapter 4 contains a description of the proposed method of tool wear monitoring. Topics such as signal processing techniques and neural network implementation are laid out. The last section describes the various components of the expert system, including knowledge bases, rule base and procedural functions. Finally, a description of the user interface is presented.

Chapter 5 presents all the results obtained through experimental work. In Chapter 6, as well as the results, performance measurements are given for each stage of the hybrid monitoring system.

Chapter 7 presents the achievements of this work giving details on the success of the applied techniques towards tool wear monitoring. Also, the relationship between the different tool wear sensors is brought together to attempt an explanation of the behaviour of such sensors through the understanding of the underlying mechanisms of wear.

Chapter 8 gives the conclusions on findings and contributions related to tool wear monitoring. Also, in this chapter, several recommendations are made which are thought would enhance the success of tool wear monitoring.

Appendix A gives a description of Adaptive Resonance Theory which enabled the construction of this neural network. Appendix B gives the theory which was used for feature evaluation based on the SOM neural network. Appendix C presents the 'C' code used to built the Expert System modules. Appendix D presents the most important functions embedded in the Expert System. Finally, appended to the Thesis are the author's publications to date.

R.G. Silva 1997

2. Background Knowledge

This chapter aims to provide a review of the physical process and the current state of the art in tool wear monitoring, and defines the development of the tool wear monitoring system which is the subject of this work.

2.1 Single Point Metal Cutting - Turning

This basic operation is also the one most commonly employed in experimental work on metal cutting. The work material is held in the chuck of a lathe and rotated. The tool is rigidly held in a tool post and moved at a constant rate along the axis of the bar, cutting away a layer of metal to form a cylinder or a surface with a more complex profile. This is shown diagrammatically in Figure 2.

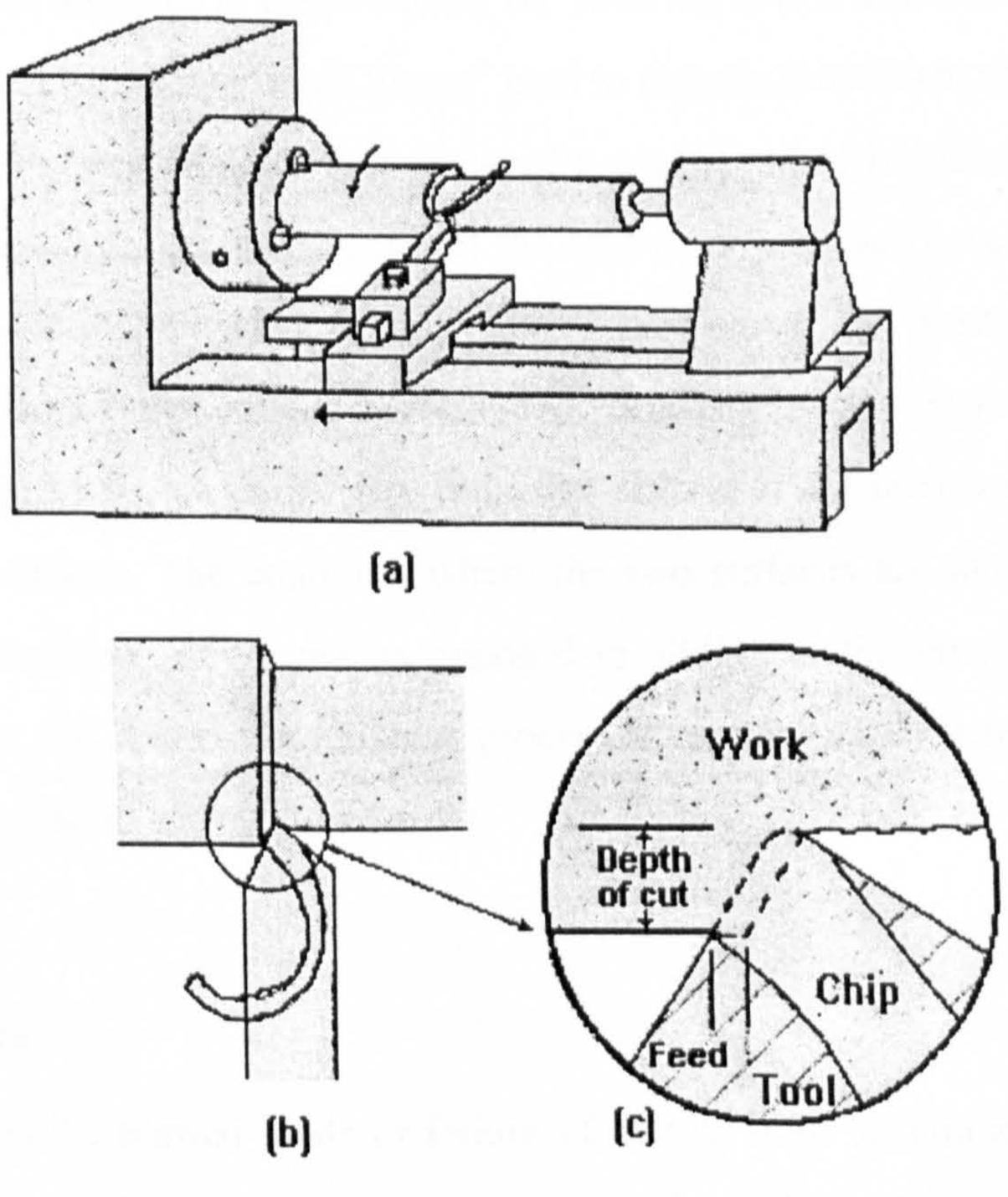


Figure 2: Lathe turning

The cutting speed, feed rate and depth of cut are three most important parameters which can be adjusted by the operator to achieve optimum cutting conditions with the depth of cut being occasionally fixed by the initial size of the bar and the product. The cutting speed is maintained constant by the CNC machine, hence changes in diameter will affect the rotational speed

(RPM) of the spindle. At the nose of the tool the speed is always lower than at the outer surface of the bar, but the difference is usually small and the cutting speed can be considered to be constant along the tool edge.

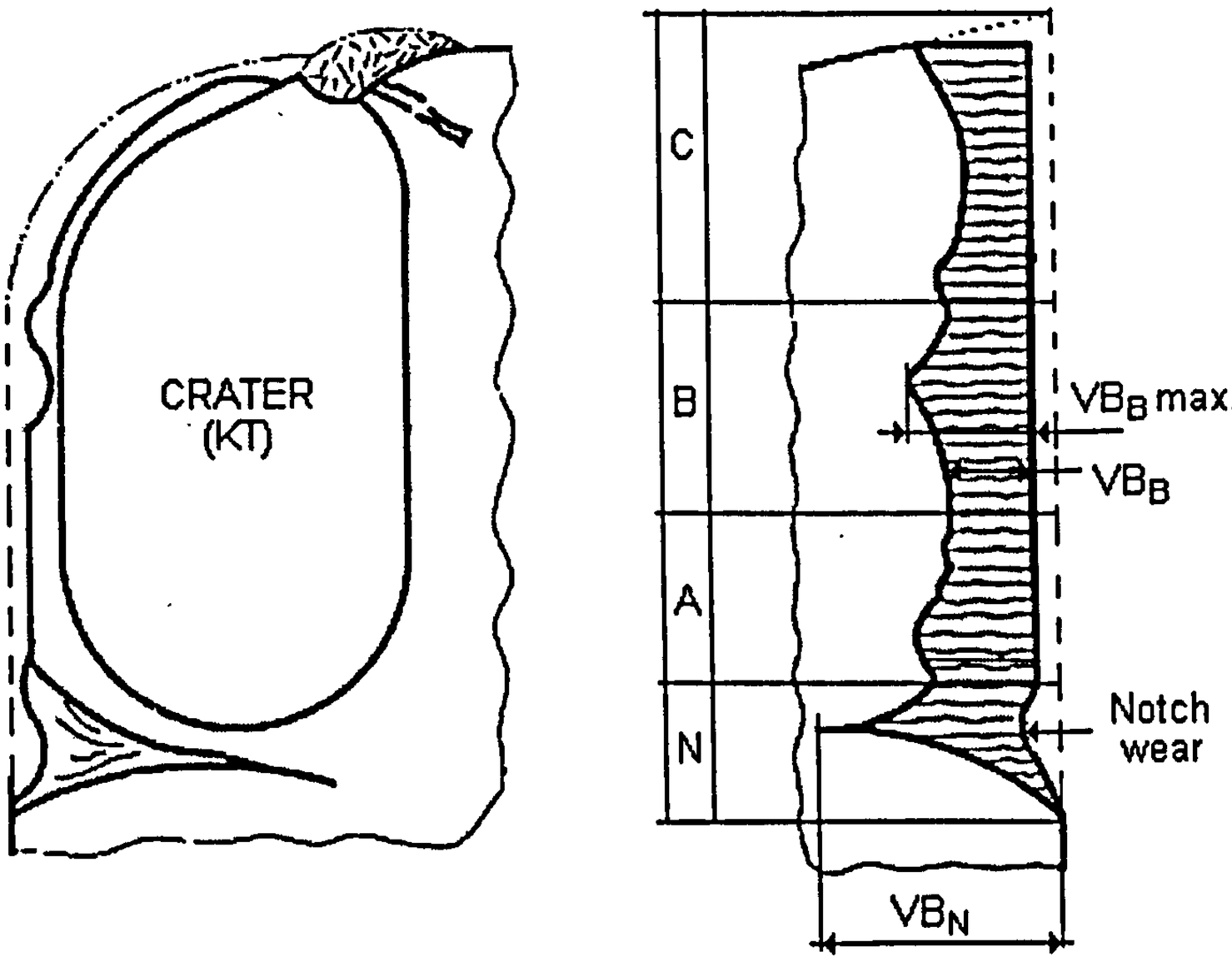
Although in machining processes a large amount of energy is required to form the chip and to move it across the tool face, the theoretical minimum energy required to form the new surfaces is an insignificant proportion of that required to deform plastically the whole of the metal removed (Trent, 1991). Thus, metal cutting is a high energy density process where very local intense deformation forces are supported by a small area of the cutting tool causing it to wear. The aim of tool wear monitoring is to sense these various manifestations of these local forces (through a selected set of sensors) to infer the condition of the cutting edge.

2.1.1 The Interface Between the Tool and Chip

In most mechanical analyses of metal cutting the shearing action has been treated as a classical friction situation, in which "frictional forces" tend to restrain movement across the tool surface, and the forces have been considered in terms of a coefficient of friction between the tool and work material. However, detailed studies of the tool/work interface have shown (Trent, 1991) that this approach is inappropriate to most cutting conditions. The most important conclusion from the observations is that contact between tool and work surfaces is so nearly complete over a large part of the total area of the interface, that sliding at the interface is impossible under most cutting conditions. The condition where the two surfaces are interlocked or bonded is referred to as conditions of seizure as opposed to sliding at the interface (Trent, 1991). In monitoring terms, this means that the wear process is not necessarily detectable by the types of methods that are used to detect sliding wear.

2.1.2 Tool Wear

Excessive wear is the normal mode of failure of cutting tools so knowledge of the tool wear level and the rate at which it wears is necessary for determining the residual tool life. Several approaches, depending on the type of wear, can be used to quantify the wear on a cutting tool (Figure 3) but generally some dimension of the tool wear land is used.



VB_N - Length of wear notch

VB_B - Average flank wear land

KT - Crater depth

Figure 3: Some types of wear (ISO3685, 1993)

Wear results in a modification of the cutting tool geometry so that metal removal gradually becomes more inefficient and, eventually, the quality and accuracy of the cut surface are compromised. Before this happens, sufficient change in the mechanics of the tool/workpiece interaction should have occurred to allow the detection of impending tool failure due to excessive wear. The very high contact forces and metal deformation rates in cutting can give rise to a variety of distinct mechanisms by which material can be removed from the cutting tool, e.g. Shaw (1989):-

- Adhesive wear - This occurs when two surfaces come close enough together to form strong bonds. If these are stronger than the local bonds of the material a particle may transfer from one material to another.
- Abrasive wear - This involves the loss of material by the action of hard constituents as they are swept over the tool surface.
- Diffusion wear - If the localised temperature of the contact surfaces is high enough, interstitial diffusion can occur across the tool-chip interface, softening the tool material (Solid State Diffusion is the mechanism by which atoms in a metallic crystal shift from one lattice point to another causing a transfer of the element in the direction of the

concentration gradient. Diffusion is a time and temperature dependent process and also depends on bonding affinity of the pair and the degree of atomic mobility).

• Fatigue - Fatigue of the tool material can result from the fluctuation of cutting forces.

Diffusive wear is known to occur as the cobalt binder on the rake face of the tool diffuses towards the workpiece at temperatures between 700 and 900°C, thus weakening the surface layer of the tool. Since diffusive wear is a function of temperature, it is commonly affected by cutting speed and feed rate. Micheletti *et al.* (1976) and Dan and Mathew (1990) report that significant changes in tool condition can occur with small changes in cutting temperature. In addition, Dan and Mathew (1990) also note that there is a sudden drop in cutting temperature which is coincident with a drop in force when cutting stops and that this increases both in magnitude and duration as wear progresses. In practice it is difficult to measure the temperature at the cutting edge with most reported work using a thermocouple located near to the tool-workpiece interface or infrared systems.

Practical wear situations rarely involve only one of these mechanisms and this depends on the workpiece and tool materials, the cutting geometry and the cutting conditions (Shaw, 1989). In addition to the wear-related sources of tool failure, the following also may occur; microchipping, gross fracture and plastic deformation. These, however, are readily identified and the solution is apparent, for example when plastic flow occurs at the tool tip, tool clearance is lost, the temperature rises abruptly, and the total tool failure occurs rather rapidly. The obvious solution to the latter difficulty is to use a lower cutting speed or a tool material that is more refractory.

The most important types of tool wear for a carbide insert are wear-land formation and crater formation. Crater formation tends to be more important than wear-land formation in situations where cutting temperatures are very high. As already discussed, there are several interacting mechanisms responsible for tool wear in cutting operations such as turning and the challenge is to increase the accuracy of the wear estimate and, at the same time, improve adaptability in the face of changes in parameters such as the machining conditions and workpiece material.

2.1.2.1 The Mechanisms of Wear and Tool Degradation

There is general agreement, e.g. Bhattacharya and Gosh (1964); Trent (1991), that an increase of cutting temperature due to increased cutting speed and increased wear causes diffusional

metal transfer to occur. It appears that diffusion induces two distinct wear processes: direct metal transfer through diffusion and macroscopic metal removal by the adhering chip through breaking away of the surface layers already weakened by the structural transformation undergone at the interface due to diffusion of various constituents of the tool material into the chip and reverse diffusion of iron into the carbide. Although weaker, the iron carbide contributes to the "re-strength" of the tool reducing somewhat the contribution of diffusion to tool wear, Trent (1991). Therefore diffusion wear is only a fraction of the total wear volume for a given time. The difference between the diffusion wear and the total wear is due to a considerable amount of abrasive wear taking place along with the diffuso-adhesive wear, a suggestion which is also supported by Chubb and Billingham (1980).

According to Opitz and König (1967) the penetration of iron into cobalt at higher temperatures is much deeper than that of cobalt into steel. Also, carbides containing high contents of TiC (Titanium Carbide) reduce oxidation and diffusion tendencies. Therefore, the coating on the flank face wears mostly due to abrasion (Chubb and Billigham, 1980). Once the coating is removed from the flank, abrasion and diffusion take over as the major wear mechanisms. This explains the higher wear rate when the tertiary wear stage is reached; in the beginning the tool is protected by the coating reducing both abrasive wear (due to its hardness) and diffusion wear (lower temperatures due to TiC defence barrier). Once the coating has worn out, diffusion takes place much faster (also due to high temperatures associated with increased wear) and the iron diffusion into the tool weakens the tool's structure by the creation of iron carbides that are weaker than the tungsten carbides.

Trent (1991) in his review suggests that temperature is the dominant factor in the machining of carbon and alloy steels and also suggests that the effect of friction (especially when cutting takes place under severe cutting conditions) is limited, with plastic flow of the chip over the rake face being the mechanism by which material is removed from the workpiece. According to Ya *et al.* (1991) tool wear is caused by extrusion and shearing of the cutter with respect to the material in the separating process of metallic materials as well as the rubbing of chips on the cutter surface.

Concluding the above discussion, and since abrasion might be expected to produce force fluctuations, it is possible to anticipate how the amplitude of vibration of the tool shank might change with tool wear. An initial increase in the vibration might be expected as abrasive wear increases for a while and then a drop should occur on reaching the tertiary wear stage. In the

final wear stage, temperature increases much more and adhesive wear ceases giving way to diffusion wear and a further increase in abrasive wear. Obviously, such an evolution is a complex one and will be sensitive to the tool material, tool cutting conditions and the workpiece material.

2.1.3 Flank Wear

The tool is in contact with the chip under conditions of very high stress and temperature (Trent, 1967) and tool wear occurs on both the rake and flank faces. As the cutting tool wears the cutting edge gradually changes shape so that in time it becomes dull. Typical wear of a cutting tool is shown in Figure 3, where the flank face is usually worn to form an approximately flat surface extending from the cutting edge, the flank wear land. On the rake face, a crater may form a short distance away from the cutting edge and as tool wear progresses, these two wear lands eventually impinge upon each other causing a substantial change in local edge geometry.

Measurement of flank wear is perhaps the most convenient means of quantifying the overall wear level of a cutting tool and so this measure is most commonly used. The width of the flank wear land, VB_B, is a suitable dimension and the attainment of a predetermined value of VB_B is generally regarded as an acceptable wear-out criterion, ISO3685 (1993). According to this standard, when the average width of the flank wear land is 0.3 mm (i.e. $VB_B = 0.3$ mm) the tool insert is worn out, provided that the tool is uniformly worn in Zone B, Figure 3. To provide a quantitative measure of tool wear and a comparison with Taylor's tool life equation, VB_B was adopted as the measure of tool wear in this work.

2.1.4 Empirical Tool Life Equation

One of the most important considerations in machining process optimisation is tool life and its relationship with process parameters. The traditional approach to determining tool life is to determine experimentally a relationship between the cutting conditions affecting tool life for a given tool-workpiece combination. However, a number of other factors can affect tool life and these are not always evident due to the complexity associated with the cutting process. There are many empirical relationships for determining tool life (Colding and König, 1971). Some of these relationships are briefly reviewed here but they are not particularly accurate unless developed for a very specific set of conditions, whereas Shaw *et al.* (1961) use a variation of Taylor tool life equation, others adopt different empirical relationships. Nevertheless, the

limitations of these relationships are widely recognised, especially with regard to factors such as cutting fluid, temperature and tool geometry.

Usui *et al.* (1984) developed a method of predicting the crater and flank wear of tungsten carbide tools cutting a 0.25% carbon steel for a wide variety of tool shapes and cutting conditions in practical operations based only on orthogonal cutting data from machining and two constants related to wear. It has been found that this approach is still limited and required refinement for general applicability. It should also be pointed out that these findings were obtained under controlled laboratory conditions. Hastings and Oxley (1976) have proposed a method of estimating tool life values for a wider range of cutting conditions using a XC45 steel (BS - 080M46, High carbon steel). This has proved encouraging but, as with the Taylor equation, enormous amounts of data have to be collected in order to determine the constants.

Most of the research carried out in developing reliable tool life criteria have concentrated on establishing a relationship based on a specific set of parameters affecting the life of a cutting tool. The complexity of the cutting process associated with the wide variety of material makes it very difficult to establish a global criterion that is reliable. For example, while the presence of manganese sulphide in steels is generally found to extend tool life, certain combinations of speed and feed yield results that indicate the reverse effect (Shaw *et al.*, 1961). Also, several factors such as clearance angle and thermal diffusivity changes might affect the tool life constants, in which case slight changes in predicted tool life may occur (Rubenstein, 1976).

In summary, tool life may be affected by the cutting tool material, workpiece material and machining conditions (Trent, 1991). Today, HSS tools, carbide and coated carbide inserts are commonly used in the metal cutting industry. For these tool materials and under common cutting conditions, diffusion, deformation and abrasion are the main causes of wear. When tool wear is dominated by these mechanisms, Taylor's tool life equation [1] offers a good estimate for expected tool life. In fact, the long term absolute machinability standard ISO3685 (1993), is defined by Taylor's tool life equation; this relates tool life in minutes, T, to cutting speed, V, in m/min:

$$VT^n = C \qquad [1]$$

where n and C are constants. The value C in equation [1] is the speed corresponding to a 1 min tool life, and the exponent n is a constant depending on the tool material. The tool life equation

can also be extended to include feed, depth of cut, or other factors such as tool geometry (Lau and Rubenstein, 1978).

Taylor's relationship was first developed for HSS tools. Carbide and coated carbide inserts were later introduced (1960s) and are now the industry standard in the metal cutting industry. Although other carbides and nitrides have been used as coatings only three coating materials are presently in wide commercial use; TiC, TiN, and Al_2O_3 with TiN appearing to offer the lowest tool friction.

Taylor's tool life equation only provides an expected tool life whereas, in reality, tools can break or can be worn out before or after the expected time (Rangwala and Dornfeld, 1987). This breakage or premature wear can be costly in modern facilities where capital cost is high and system attendance by operators is low. A conservative strategy can be developed to replace the tools more frequently, but, if tools are replaced too frequently, the costs associated with replacements can be unnecessarily high. This is often understressed due to the apparent low cost of inserts, but it should be noted that the use of insert cost as replacement cost is misleading because the cost of indexing and machine down-time can be significant.

Taylor's tool life equation is useful to establish a preliminary tool life and thus establish intervals of confidence whereby it is possible to classify a tool as worn or not. The analysis of data given in the literature indicate that this equation can give estimates that are within a close range of the actual tool life (Colding and König, 1971; Shaw, 1989), however, it should be used with caution. This can help solve problems where illogical classification of wear due to sporadic signal distortion, to a prescribed level, results in misclassification.

Taylor's tool life equation applies reasonably well over a restricted range of cutting speeds. The approximate variation of the tool life exponent n and constant C for different tool materials for the turning process is given in Table 1, according to different authors, regardless of workpiece material.

Table 1: Ranges of values n for a variety of tool materials

Tool Material	n (Juneja, 1986)	n (Pilafidis, 1971)
HSS	0.08 - 0.20	0.09 - 0.55
Carbide	0.20 - 0.49	0.10 - 0.80
Ceramics	0.48 - 0.70	0.30 - 0.65

2.2 Tool Wear Sensing

A sensor is a device which detects a physical process and converts energy associated with this process into an electrical signal which can be recorded and analysed. Due to the difficulties involved in direct measurement of tool wear, most of the proposed techniques use indirect methods in which a certain relationship is established between tool wear and some other, measurable, quantities such as vibration. According to Brunn (1981) tool wear sensors should, as far as possible, be simple to operate, robust and easily read and interpreted, since classification of tool condition is often undertaken in difficult environments. Tool wear sensing is important as a predictor of tool failure due to excessive wear and, in long production cycle manufacturing operations, tool wear detection may be important for surface roughness considerations.

Wear monitoring has been performed using many different types of sensors. The most commonly used measurements are temperature, feed and spindle currents, Acoustic Emission (AE), audible emissions, workpiece and machine tool vibration and cutting force. Reviews of metal-cutting analysis techniques have been carried out by Finnie (1956) and Jetly (1984), who reported that all of the above approaches have been successfully demonstrated under laboratory conditions although there are few successful industrial applications. Other reviews, such as those of Micheletti *et al.* (1976) and Shiraishi (1988), made a comparative analysis of the applicability of sensors, that is, their suitability for monitoring tool wear in a variety of cutting processes. They have found that the most promising tool wear sensors are those which measure force and current.

Commercially available systems for wear and breakage detection typically set limits for force or power based on empirical data (Novak and Ossabhr, 1986). When the measured force or spindle current falls outside these predetermined fixed limits the tool is assumed to have failed due to excessive wear or breakage. The disadvantage of the fixed force limit method is that all the machining conditions must remain nearly identical throughout the whole cutting operation, and therefore this is applicable only in very simple cases.

Taking into account previous studies (e.g. Dan and Mathew, 1990; Guinea *et al.*, 1990; Dornfeld *et al.*, 1993; Ruiz *et al.*, 1993; Nagy and Szalay, 1993) it would seem appropriate to integrate some of the above sensors in order to extract the largest amount of information from the cutting process. Using the criteria of reliability and ease of use, force, vibration and sound

emission transducers were selected as having industrial potential among the indirect methods and these were consequently used in this work.

2.2.1 Audible Emissions

It has long been claimed that an experienced operator with a keen ear can predict with reasonable accuracy the state of deterioration of a cutting tool (Lee, 1986). Audible emissions from the metal cutting operation can be divided in two main components; structural noise and cutting noise. The former arises from excitation of the machine tool by the cutting operation and the latter is caused by interactions between tool and workpiece during the cutting process (Ya *et al.*, 1991). Previous work has investigated the relationship between audible emissions and tool wear and some of the most significant results are described below.

An early study (McNulty and Popplewell, 1977) which investigated changes in the sound spectra during the life of drills, hacksaw blades and single point turning tools revealed that there were significant frequency bands associated with tool wear in certain cutting processes (e.g. a lathe cutting tool exhibits a significant fall in sound pressure level (10dB) over its life). It was suggested that a knowledge of the evolution of both spectra and wear bands would provide a monitoring medium for tool life history. Lee (1986) found that the sound emitted from the turning process exhibited a change of Sound Pressure Level (SPL) that was related to tool wear in the 4 to 6 kHz frequency range and that this occurred for several material/workpiece combinations; HSS and Tungsten Carbide (P3) tools, AISI 1045 carbon steel and AISI 304 stainless steel. When turning with carbide tools the SPL changes were found to be more pronounced than those with HSS tools, this being manifested by a drop in the SPL before the tertiary zone (third and last stage of wear) was reached. Similar results were found by Sadat and Raman (1987) and Trabelsi and Kannatey-Asibu (1991), but these authors noted slightly different frequency bands, the latter authors reporting 100% successful classification of the tool when the spectral components in the 0 to 10 kHz range were used as features.

Experiments carried out by Ya *et al.* (1991) using two different types of turning tool showed that neither the tool rake angle nor the cutting speed exerted any significant influence on the cutting sound. In most cases, the sound emitted whilst cutting was 2-3 dB higher than the idle running noise.

2.2.2 Vibration monitoring

The early work of Weller *et al.* (1969) investigated the possibility of relating tool wear and vibration and found that the high-frequency energy, in the 4 to 8 kHz band, increased with tool wear. This discovery helped in the development of the first tool wear monitoring systems (e.g. Martin *et al.*, 1974; Petrie *et al.*, 1989a). Jiang *et al.* (1987) proposed that the variation of the cutting vibration signal with tool wear is not an accidental and isolated phenomenon but that it is closely related to the cutting process. The frequency composition of the energy of the signal varies regularly with the development of tool wear and at specific frequencies the signal follows a constant pattern with changes in tool wear, i.e. typically increases with wear. Similar results were obtained by Pandit and Kashou (1982) and Taglia *et al.* (1976), who have shown that all the vibration signal power is sustained in frequencies up to 10 kHz, and that a very small percentage of the total signal power varying with tool wear is contained in the frequencies up to 2.5 kHz. Pandit and Kashou suggested that the modes of vibration most sensitive to tool wear occur in the frequency band [4.2;4.7] kHz.

Martin *et al.* (1974) and Bonifacio and Diniz (1994) found that the vibration in the cutting plane was the only component that exhibited significant changes with tool wear. Martin *et al.* (1974) concluded upon experimental work that the vertical vibrations of a lathe tool in the course of stable machining are almost sinusoidal, with frequency perceptibly equal to the natural frequency of the tool, the power of the acceleration signal determined by spectral analysis is a linear function of the cutting speed and of the tool wear, the signal increasing in the ratio of 1:10 between a new and a worn tool.

Several methods for feature extraction from the vibration signal have been proposed in the literature (e.g. Colwell, 1971; Petrie *et al.*, 1989b; Dan and Mathew, 1990). The Fast Fourier Transform (FFT) is common among techniques applied for the estimation of the power spectrum. The use of statistical measurements such as kurtosis can be applied to the original time series in an attempt to quantify the closeness of the series to a normal distribution (Petrie *et al.*, 1989b; Wilcox *et al.*, 1993) and may be used to describe impacts or transients. Other methods for tool wear sensing based on vibration monitoring have also been proposed, such as the one based on Data Dependent Systems (DDS) modelling of vibration (Pandit and Kashou, 1982) and the Group Method of Data Handling (GMDH) (Ravindra *et al.*, 1994).

2.2.3 Force monitoring

The cutting forces experienced by the tool are an important aspect of machining for those concerned with the manufacture of machine tools and also for tool wear monitoring purposes. The component of the force acting on the rake face of the tool, normal to the cutting edge, is called the tangential cutting force F_t and this is usually the largest of the three components, acting in the direction of the cutting velocity. The component of the cutting force acting parallel to the direction of feed is referred to as the feed force F_f. The final component, which tends to push the tool away from the work in a radial direction F_r, is the smallest of the force components in semi-orthogonal cutting and, for purposes of analysis of cutting forces in simple turning, is usually ignored and not even measured (Trent, 1991).

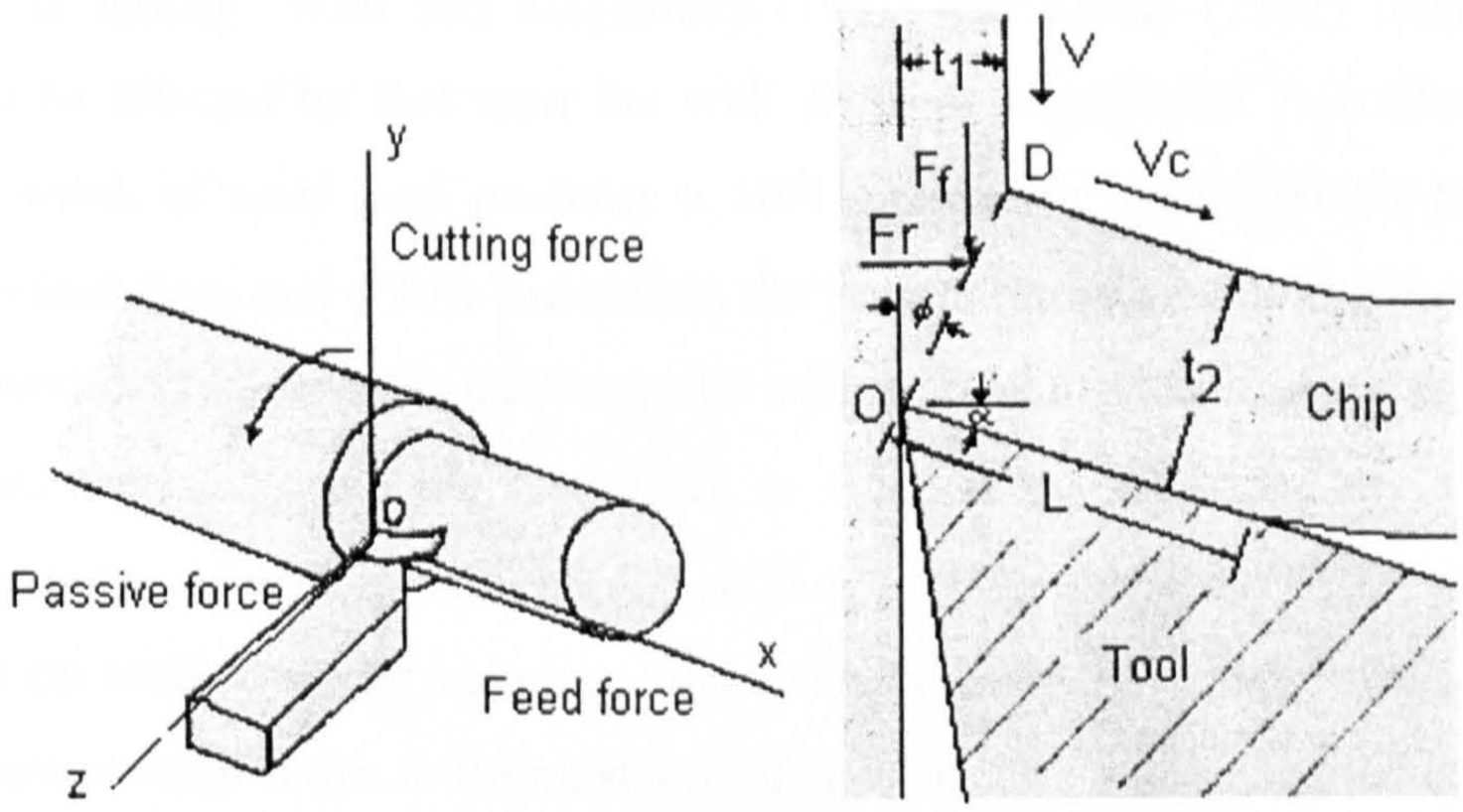


Figure 4: Cutting force components in turning

Among the measurements used for indirect flank wear estimation, the cutting force signal has been the most successful due to its sensitivity to tool wear and ease of measurement (Nair *et al.*, 1992; Danai *et al.*, 1992; Lee *et al.*, 1996). The cutting force generally increases with flank wear due to an increase in the contact area of the wear land with the workpiece (Trent, 1991), but it is also dependent on many other factors such as cutting conditions. Filippi and Ippolito (1969) were among the first to demonstrate the direct effect of flank wear on the cutting force. These methods provide relatively accurate estimates of flank wear as long as the cutting variables (feed rate, cutting speed, and depth of cut) remain unchanged. However, when the cutting variables change, due to factors such as geometric requirements or cutting conditions the results are more difficult to predict (Nair *et al.*, 1992; Lee *et al.*, 1996).

Lately there have been many attempts to use force changes as an indication of tool wear for the turning process (e.g. Mackinnon *et al.*, 1986; Lee *et al.*, 1996; Das *et al.*, 1996). Colwell (1971) and Shiraishi (1988) suggested that cutting forces and associated power spectra appear to be more generally applicable and closer to accomplishment than temperature and vibration. Analysis of the frequency spectrum has shown a distinct peak frequency which is consistent for a wide range of machining conditions and, for different lathes, a good correlation was found between the dynamic cutting force and flank wear (Lee *et al.*, 1989). However, the mechanisms which bring about tool wear and failure have a relatively complex relationship on cutting forces.

Dan and Mathew (1990) showed that the force components are influenced by tool wear in a linear manner in turning. Wolf and Magadomy (1981) and Ridley (1982) found the force components to be affected by tool wear but with different magnitudes. According to Ridley each 0.1 mm width of wear land produces a 10% increase in the tangential force, a 25% increase in the feed force and a 30% increase in the passive (thrust) force. However, Wolf and Magadomy found the difference in the tangential cutting force between a sharp edge and worn ones to be less than 10% and the feed force to increase by approximately 150%. These contradictory results, as well as others e.g. Lee *et al.*, 1989; Lin *et al.*, 1982, prove that no consensus has yet been achieved regarding force measurements. According to Mackinon *et al.* (1986) this inconsistency is due to the existence of many factors influencing the cutting forces, e.g. tool geometry, depth of cut, feed, cutting speed, workpiece hardness.

Some attempts have been made to relate tool forces with tool wear by a mathematical model, such as the one mentioned in Stern and Pellini (1993):

$$F = F_0 + dC_w W \text{ (Koren } et\ al.\text{, 1986)} \qquad [2]$$

Where F is the machining force, F_0 the initial machining force, d the depth of cut, C_w a constant for the machining parameters, and W a size of the wear land. However, the constant C_w cannot be obtained easily because it includes many other interactions with the cutting parameters that may not display linear behaviour (Stern and Pellini, 1993), and the process of obtaining it is essentially experimental and limited in use. Others, such as Lin *et al.* (1982) and Nair *et al.* (1992) have also attempted to model the effect of tool wear on cutting force, the latter demonstrating that, although fairly accurate, results could only be used in a limited environment, i.e. for a specific feed rate and cutting speed. One of the difficulties in trying to

correlate a component of the cutting force with tool wear is that the cutting forces also change with cutting conditions such as cutting speed, depth of cut and feed rate. Oraby and Hayhurst (1991), have also derived an equation to predict tool wear under different cutting conditions based essentially on experimental results. Again, such a method relies on extensive experimental work.

2.2.4 Current Monitoring

Several interesting techniques, most of them more sensitive to tool breakage than to the evolution of tool wear, have recently been developed by monitoring the motor load by measuring the current drawn by the motor, e.g. Shiraishi (1988). The motor current can be related to the torque exerted and hence is related to the cutting force, Agogino *et al.* (1988). Since force is a good measure of tool wear it would be logical to conclude that current is also likely to be an indicator of tool wear (Wilcox *et al.*, 1993). Because of the sharp changes in the motor load during the tool entry and exit, the current sensor is an excellent detector of cutting and also an easy one to implement since a meter is normally provided on the machine tool.

From the above literature there is sufficient evidence to suggest that further study of at least four sensors could prove to be useful in the construction of an intelligent tool wear monitoring system:-

1. Cutting forces (feed and tangential)
2. Spindle current
3. Audible emissions
4. Machine vibration

Although this is not an exhaustive sensor set, it was chosen as being easy to implement and on the past empirical success of other researchers.

2.3 Sensor Integration, Feature Processing and Reduction

Many mechanical systems are sufficiently complex that it is impractical to describe their dynamics by exact mathematical models. In the absence of good or acceptably simple models to predict the processes over the great diversity of operating conditions, sensing devices will play a crucial role in truly automated factories. This section will focus primarily on the pre-processing of data and its importance, following an examination of the signal generating

processes as the cutting tool wears. The use of multiple sensors and their integration will conclude this section.

The decision-making stage of a monitoring system involves making qualitative inferences from large amounts of numeric data obtained from different sensors each carrying a weighting according to its relative importance. There are several approaches to solve this problem but most are based on probability theory and statistics, which unfortunately makes them slow and possibly ineffective. Because of the inherent complexity and variability, underlying distributions are unknown, parameter estimations are unsuccessful and explicit rules, if any, are not well understood. Reddy (1992) presented a survey of a number of different approaches to the problem of multi-sensor integration that have emerged in recent years, these being the application of probability methods, fuzzy logic, Expert Systems, and neural networks. In cases where the sensor data is noisy and not very clustered, the classification performance benefits greatly through the use of multi-layered neural networks. After training these can be used to classify, and thus recognise, new instances of similar patterns (Venkatasubramanian and Vaidyanathan, 1991).

Since neural networks can perform essentially arbitrary non-linear functional mappings between sets of variables (Chitra, 1993), a single neural network could, in principle, be used to map the raw input data directly onto the required final output. In practice, for all but simple problems, such an approach will generally give poor results due to the fact that real data often suffers from a number of deficiencies such as missing input values or incorrect target values. For most applications it is necessary first to transform the data into some new representation before training the neural network (Qin and Rajagopal, 1993). To some extent, the general purpose of a neural network mapping means that less emphasis has to be placed on careful optimisation of this pre-processing than would be the case with simple linear techniques. Nevertheless in many practical applications of ANNs the choice of pre-processing will be one of the most significant factors in determining the performance of the final system (Rahman *et al.*, 1995). In the simplest case, pre-processing may take the form of a linear transformation of the input data, and possibly also of the output data. The fact that such dimensionality reduction can lead to improved performance (Wu and Du, 1996) may at first appear somewhat paradoxical, since it cannot increase the information content of the input data, and in most cases will reduce it. For monitoring of machining processes the sensor signals typically contain information and noise, therefore, it is desirable to extract the features that represent the characteristics of the process (information) and to separate the features from various noise

disturbances. Care has to be taken since some features in sensor signals are correlated with certain levels of tool wear but not with others (Leem *et al.*, 1995).

Numerous experiments (e.g. Rubenstein, 1976; Weller *et al.*, 1969) have shown there are many parameters which influence the cutting process for any given cutting-tool workpiece combination; cutting speed, feed rate, tool overhang, cutting edge condition (sharp, dull, cratered, etc.), tool material, workpiece material, and workpiece configuration. For our purpose, it is sufficient to assume (Lim, 1995) that the degree of wear on the tool's cutting edge is one of the major causes of system performance.

The tool wear signal (without noise) consists of four main components (Arnold, 1946; Heck, 1993):-

1. A slowly varying response of the tool to quasi-periodic excitations - this arises from the high speed rotation of the workpiece.
2. Randomly occurring transients - this includes chipping of the tool or workpiece.
3. Transients due to the modes of vibration of the tool holder.
4. The complex interaction of the material with respect to the cutter.

The noise results from three main sources; mechanical noise, electrical noise, and fluid noise (only applies in the case of coolant use). As the tool wears, cutting forces increase, particularly in the direction of feed, resulting in horizontal vibration and leading to variations in the vertical force experienced by the tool (Oraby and Hayhurst, 1991). This can be compared to the effect obtained by bearing down harder on the bow of a violin, the pitch of sound, or vibration, is the same but the volume of sound generated is greater. Changes in workpiece size or location of the cutting tool along the axis of the workpiece can also have an effect.

Measurable tool wear related signals typically have a very low Signal to Noise Ratio (SNR) (Heck, 1993) because of the variety of noise sources on the machine tool. However, relatively little work has been done on the enhancement of the signal related to tool wear and noise reduction. For tool wear classification, most monitoring systems either use the noisy signals directly without pre-processing, or simply low-pass filter the signal to eliminate the corrupting noise sources (Okafor *et al.*, 1991). While relatively easy to implement, these techniques have proven to be generally ineffective at reducing the noise and tend to remove information necessary for proper tool wear classification (Heck, 1993). It would seem therefore that noise reduction is a particularly difficult task for this process, and it would be unwise to eliminate

components of the signal that could be useful and perhaps it would be more prudent to concentrate on signal processing and feature reduction, such as using basic statistical analysis, skew and kurtosis, as well as mean and standard deviation as additional ways of characterising the signals.

One of the simplest techniques for dimensionality reduction is to select a subset of the inputs, and to discard the remainder. This approach can be useful if there are inputs which carry little useful information for the solution of the problem, or if there are very strong correlations between some of the inputs, i.e. the same information can be obtained from several signals. Any procedure for feature selection should be based on two components. First, a criterion must be defined by which it is possible to judge whether one subset of features is better than another and, secondly, a systematic procedure must be found for searching through candidate subsets of features (Flachs *et al.*, 1990; Trabelsi and Kannatey-Asibu, 1991). Ideally the selection criterion should be the same as that to be used to assess the complete system. The search procedure could simply consist of an exhaustive search of all possible subsets of features since this is the only approach which is guaranteed to find the optimal subset. However, this could be time consuming and so a simplified selection criterion based around a non-exhaustive search procedure could be used to limit the computational complexity of the search process.

A large number of monitoring methods have been developed to classify process condition. The simplest one is to identify two process conditions (normal and abnormal) using a sensor signal. It can be described by the condition statement, "*if* $y<T_y$ *Then normal Else abnormal*", where y is the sensor signal and T_y is a threshold value. However, in most applications, this simple decision-making strategy will not perform satisfactorily as more graduations in the classification are often required.

In general, monitoring methods are model-based or feature-based. Model-based methods have two significant limitations; the first is that many manufacturing processes are non-linear time-variant systems. A typical example is a machining process where the non linearity is caused by the regenerative interaction between the structural vibration and the cutting forces. Also, sensor signals are dependent on process working conditions. It is often difficult to identify whether a change in sensor signal is due to the change of process working condition or to deterioration of the process.

Feature-based monitoring methods use suitable features of an appropriate sensor signal to identify the process conditions. Given a sensor signal, $\{Y_i, i=1,2,3,...\}$, the feature(s) of the sensor signal, called monitoring indices, can be represented by:

$$X = P(y_i) \qquad [3]$$

where, P is an operator. These features could be time and/or frequency domain features of the sensor signal such as mean, variance, skewness, kurtosis, crest factor or power in a specific frequency band (Du *et al.*, 1995).

Feature-based methods consist of two phases; learning and classification. To monitor manufacturing processes, learning from samples is usually more effective than model-based methods since precise instructions are typically unavailable or rather limited. According to Du *et al.* (1995), a large number of methods have been developed for the monitoring of manufacturing processes, but, it is unclear which method performs best and, in fact, most of the literature only shows that a specific method works for a specific application.

The basic pattern recognition classification scheme shown in Figure 5 has a structure general to most classifying schemes. Some of the requirements which are associated with this scheme can be listed *a priori* as follows.

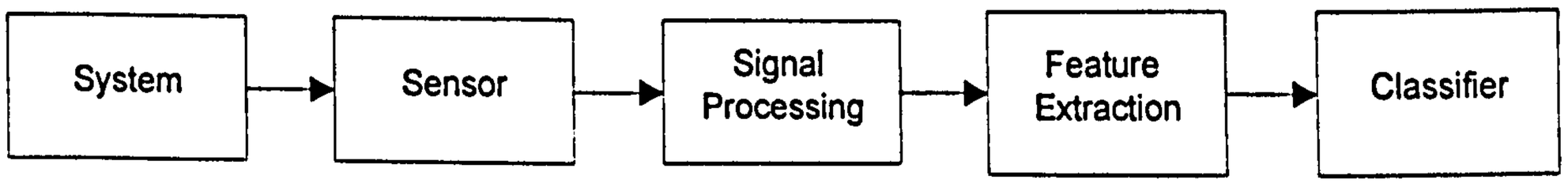


Figure 5: Pattern Recognition Classifying Scheme

1. Improved Signal to Noise Ratio (SNR)
2. Maximum data reduction to enable on-line monitoring
3. Minimisation of the effects of the specific machine structure on the classification performance
4. The need for a reduced, minimal database for the "fine-tuning" of the monitoring scheme

There is obviously some interdependence between the choice of the signal processing, features and classifier and the resulting performance of the scheme. The most critical choice is that of the feature vector, which should respond to the effects being monitored but be minimally sensitive to any other disturbance. If possible, the choice of features should reflect some of the knowledge concerning the specific behaviour of the monitored system as such specifically chosen features will often perform better than those based on very general considerations (Braun *et al.*, 1987).

The use of multiple sensors has been shown to enhance the performance of tool wear monitoring systems (e.g. Chryssolouris and Domroese, 1989; Rangwala and Dornfeld, 1990), because different sensor types can provide independent information related to the tool wear level. Such an approach also reduces the sensitivity of the system to any particular sensor's drawbacks, requiring less precision than with a single sensor potentially leading to less sophisticated signal processing (Agogino *et al.*, 1988). Through appropriate analysis, the dependence of each feature on changes in process conditions can be analysed (Rangwala and Dornfeld, 1987; Ruiz *et al.*, 1993). This provides improved reliability in making decisions as to the state of tool wear in the face of (perhaps minor) changes in machining conditions.

Several approaches exist for integrating information from multiple sensors, e.g.:-

1. Neural networks (Rangwala and Dornfeld, 1990).
2. Multivariate autoregression (Yao and Fang, 1992).
3. The Group Method of Data Handling (GMDH) (Ravindra *et al.*, 1994).
4. Entropy based procedures (Ruiz *et al.*, 1993)

Each of these approaches requires an initial training phase in which appropriate relationships are learned between the input and output signals. Generally, this training phase requires sets of data consisting of input signals for which the correct output signals are known. Among the classifiers studied, Ruiz *et al.* (1993) found that neural networks could achieve as high as 90% correct classification. The support for the use of neural networks for sensor integration is also acknowledged by Chryssolouris and Domroese (1989) through a comparative evaluation of several methods (neural networks, regression analysis, inverse least-square, and GMDH) for the integration of sensor information in machining.

2.3.1 Neural Networks

Connectionist networks, neural networks, or parallel distributed processing models as they are variously called are relative newcomers to the computational modelling scene. Previous techniques were marked by the need to program explicitly all aspects of the model, and by the use of explicit symbols to represent concepts. Connectionist networks, on the other hand, can to some extent program themselves, in that they can "learn" to produce specific outputs when certain inputs are presented.

Early theoretical proposals about the feasibility of learning in neural-like networks were made by McCullock and Pitts (1943) and Hebb (1949). However, the first neural network models, called perceptrons, were developed by authors such as Rosenblatt (1959) and Rumelhart *et al.* (1986). More recent neural networks such as the Hopfield (1982) network still rely on the basic concepts introduced by McCullock and Pitts.

Artificial neural networks have received renewed interest over the last few years, with more than 16,000 neural network development tools sold to date and 20% annual sales growth projected (Bailey and Thompson, 1990). Scores of applications, primarily stand alone, have been developed in several major industries, including finance, manufacturing, defence, aerospace, and business with corresponding applications in a huge variety of engineering problems such as; machine vision, speech recognition and control. Success stories are becoming more common in private industry as well as in academia with neural networks being found to be effective in a wide variety of medical (Anthony, 1993), technical and even marketing problems (Martin-del-Brio and Serrano-Cinca, 1993). An article by Kohonen (1988) showed a brief survey of the motivations, fundamentals, and applications of artificial neural networks, as well as some detailed analytical expressions for their theory. Abstraction of hardly accessible knowledge and generalisation from distorted sensor signals are some of the most attractive features of neural networks when applied to sensor fusion and classification in tool wear monitoring (Rangwala and Dornfeld, 1990).

Despite the current popularity of back-propagation as a supervised algorithm, its need for a correct estimate of tool condition in every training sample limits its successful application to on-line tool wear monitoring systems. The implication of requiring correct tool condition is that the machining operation must be interrupted so as to acquire information about tool condition and, as there are numerous combinations of tools, work materials, and cutting conditions (e.g. cutting speed and feed rate), which the eventual monitoring system should handle, a supervised

learning procedure like backpropagation is undesirable. For a practical and reliable on-line monitoring system, it is desirable to have a neural network utilise "unsupervised" training samples without tool wear information, thereby allowing the interpretation of the resulting self-organisation with the fewest number of "supervised" samples.

Several classes of applications are amenable to a neural network approach, with most involving either pattern recognition or statistical mapping. Conversely, other applications do not lend themselves to a neural network approach, for example mathematically accurate and precise applications. From a survey (Bailey and Thompson, 1990) of successful neural-network applications developers, several preliminary criteria for selecting applications were cited:-

- Conventional computer technology is inadequate.
- Problem requires qualitative or complex quantitative reasoning.
- Solution is derived from highly interdependent parameters that have no precise quantification.
- Data is readily available but multivariate and intrinsically noisy or error-prone.

Based on the research by Bailey and Thompson it can be seen that factors such as; type of memory, training method, type of output, training time, execution time, decision information, information content, and utilisation, some neural networks show higher suitability for the present application then others. These are the ART2 introduced by Carpenter and Grossberg (1987) and the Self Organising Map (SOM) introduced by Kohonen (1984). Kohonen's Feature Map requires many fewer samples with the correct levels of wear because the interpretation of an output node can give information useful to the interpretation of its neighbouring nodes.

The next three sections will detail some of the applications as well as the basic theory of backpropagation, ART2 and SOM neural networks encountered in the literature.

2.3.1.1 Backpropagation

The back-propagation training algorithm is an iterative gradient algorithm designed to minimise the mean square error between the actual output of a multi-layer feed-forward perceptron and the desired output, Lippmann (1987). Its architecture generally resembles layers of successive elements with each processing element producing an output that represents the weighted sum of the previous layers' output. Most existing connectionist learning procedures are slow, particularly procedures that construct complicated internal representations. One way

to speed them up is to use optimisation methods such as recursive least squares that allow faster convergence. If the second derivative of the activation function can be computed or estimated it can be used to pick a direction for the weight change vector that yields faster convergence than the direction of steepest descent, Chitra (1993).

The largest drawback of the backpropagation method is the long learning process, especially when using large training sets, or large networks (Denoeux and Lengelle, 1993). Another drawback of this method is that if data sets are not presented in a certain order it may "forget" previous learning. For example, if the network is learning to recognise the alphabet, there is little use in learning *B* if, in so doing, it forgets *A*. A process is needed for teaching the network to learn an entire training set without disrupting what it has already learned. Sima (1996) suggests that backpropagation is generally not an efficient algorithm; and also demonstrated that training the sigmoid feedforward network, with a single hidden layer and with zero threshold of output neurone, is not easily controlled.

2.3.1.2 The Self-Organising Map

The Self-Organising Map (SOM), also known as the Kohonen Map, is an unsupervised neural model of widespread use in areas such as pattern recognition and robotics (Kohonen *et al.*, 1996). The SOM is a neural network model that projects a high dimensional input space onto a usually one or two dimensional output space by using unsupervised training. This output space is represented by a discrete lattice of neurones, usually arranged in a rectangular manner. The idea of such a model is to generate topology mappings, where a low-dimensional image of the high dimensional input space is built into a rectangular array. Neighbourhood neurones on the map tune to similar features of the sensory or input space, following a self-organising competitive learning process. This simple neural model has a strong similarity with the maps existing in several parts of the brain, such as in the somatosensory cortex, where sensory information is represented in topological form (Kohonen, 1993).

Kohonen's algorithm (Kohonen, 1984) creates a vector quantiser by adjusting weights from common input nodes (X_i) to output nodes arranged in a two dimensional grid as shown in Figure 6, essentially seeking models which minimise the quantisation error (Sabourine and Mitiche, 1993). The map is generated by establishing a correspondence between inputs and neurones such that the topological (neighbourhood) relationship among the inputs is reflected as faithfully as possible in the arrangement of the corresponding neurones in the lattice (Ritter and Schulten,

1988). This renders a "non-linearly flattened" two-dimensional version of the input space which for many tasks constitutes a very useful data structure. Models are adjusted incrementally as new data is presented and an interesting aspect of the SOM is that some ordering takes place as adjacent models in the pattern space are near each other model space, that is, similar input patterns will be allocated to nearby neurones.

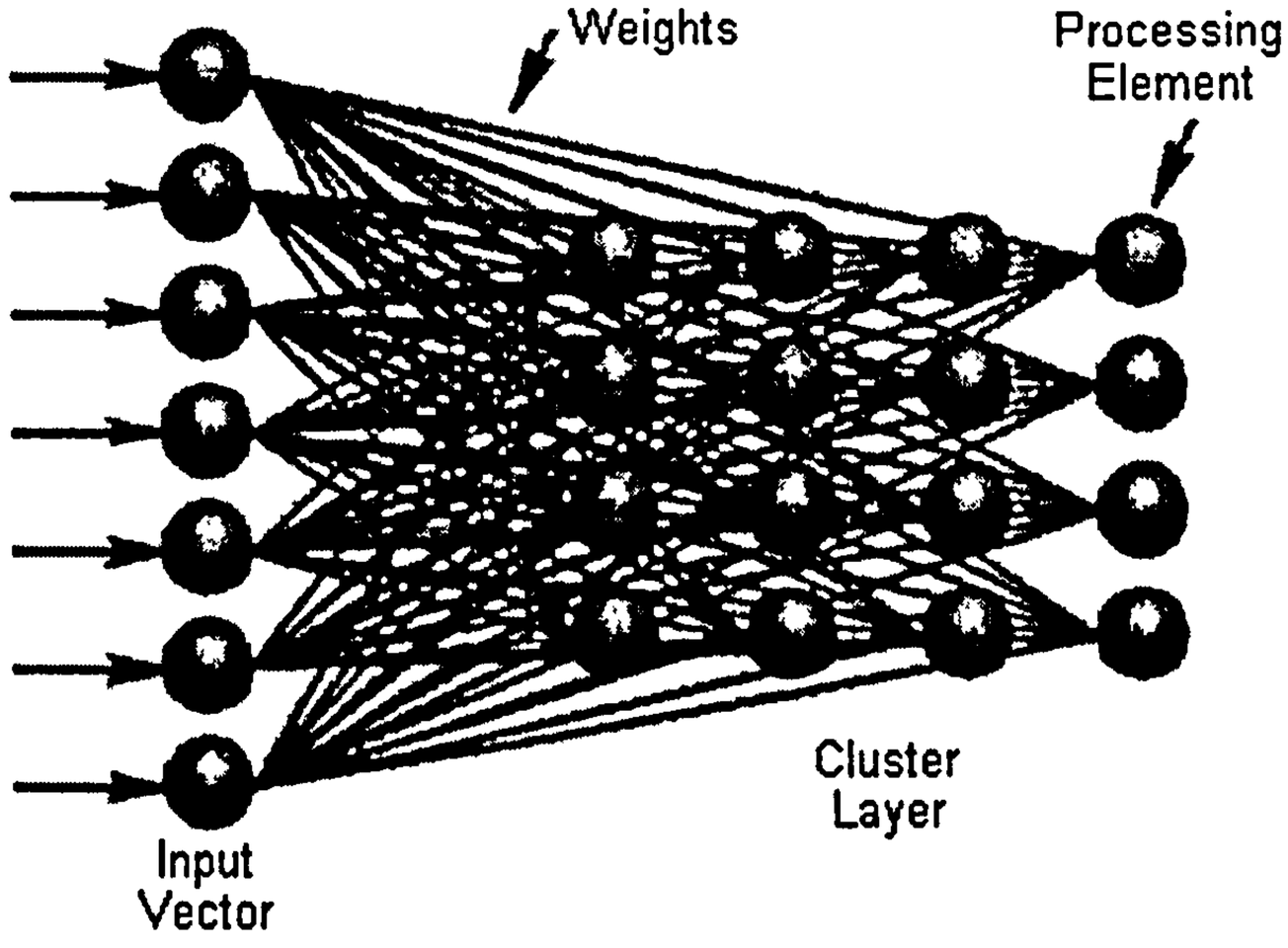


Figure 6: SOM basic structure

The correspondence between input(s) and output(s) is obtained interactively by a sequence of training steps, which can be formulated in terms of synaptic modification laws (Kohonen, 1990). Continuous-valued input vectors are presented sequentially in time without specifying the desired output. After enough input vectors have been presented, weights will specify clusters or vector centres that sample the input space such that the point density function of the vector centres tends to approximate the probability density function of the input vectors (Kohonen, 1988). In addition, the weights will be organised such that topologically close nodes are sensitive to inputs that are physically similar. However, results may depend on the presentation order of input data for small amounts of training data and therefore data sets should be specifically tested for this influence. Analysis of the synaptic weights can be used to investigate the effect of different input variables on the network operation and this enables delimitation of the different regions of the map.

The temporal topographic map can lead to a classification of sequences by means of the winner at the last stage of the sequence. Tracking the winning nodes during the presentation of a sequence gives greater sensitivity (Chappel and Taylor, 1993), leading in particular to context

distinction. The self-organising solvency map has been useful in predicting the time evolution of the financial situation of a firm, by using a time sequence of data as input patterns, e.g. data for several consecutive years from a particular bank (Martin-del-Brio and Serrano-Sinca, 1993). Other applications of the Kohonen's feature map include statistical pattern recognition (Kohonen, 1990), image processing (Sabourine and Mitiche, 1993), syntactical analysis (Chappell and Taylor, 1993) and economy issues (Martin-del-Brio and Serrano-Sinca, 1993).

2.3.1.3 Adaptive Resonance Theory

The adaptive resonance theory (ART) architecture, creates and organises categories for features and has the ability to respond immediately to experience. It was developed by Carpenter and Grossberg (1987) for its ability to continue to store information without loss. ART2 networks self-organise stable recognition categories in response to arbitrary sequences of analogue input patterns, as well as binary input patterns. Figure 7 oversimplifies the true ART2 structure given in Appendix A, it aids in understanding ART2 as a relative of competitive learning. The current purpose lies in summarising the operation of ART2 without becoming involved in biological or psychological details.

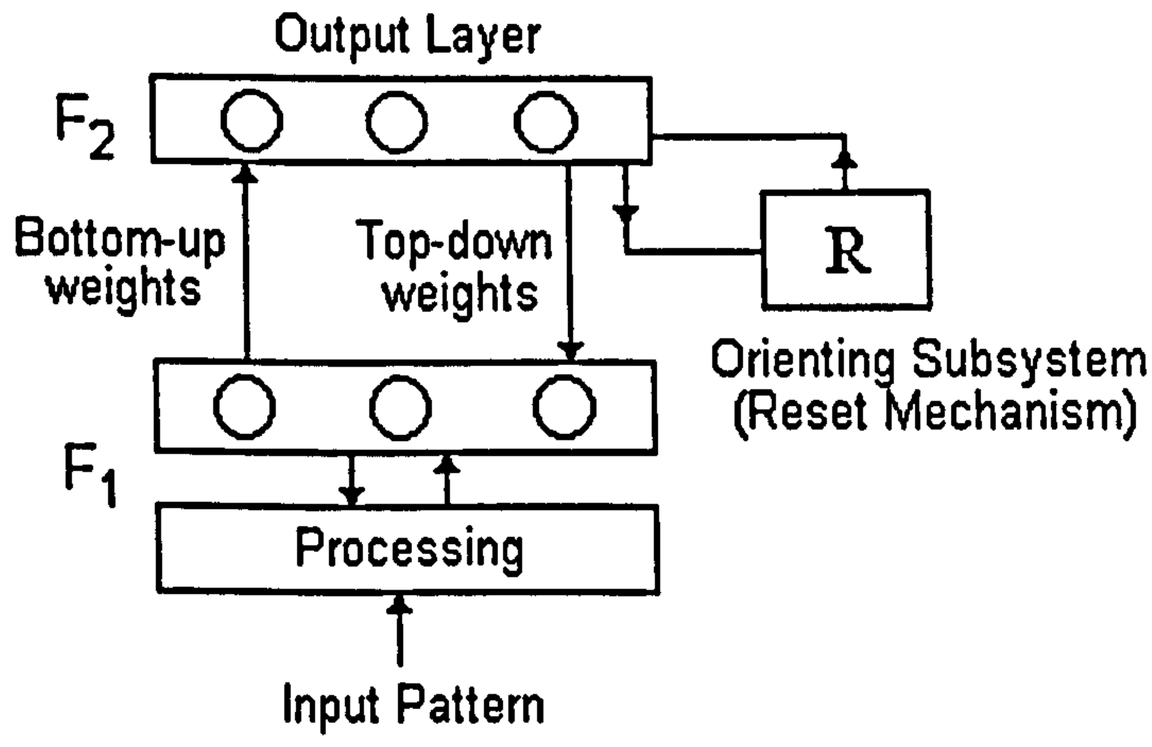


Figure 7: Simplified Representation of the ART2 network

The heart of an ART network consists of two interconnected layers of neurones, F_1 and F_2, which comprise the attentional system. The pre-processed input pattern is received at the stage F_1 of an attentional subsystem. This input leads to activity in the feature detector neurones in F_1. This activity passes through connections (synapses) to the neurones in F_2. Each F_2 neurone adds together its input from all the F_1 neurones and responds (Figure 7). The resulting pattern

of activation across F_2 is a new pattern. The contrast-enhanced pattern rather than the input pattern is stored in the Short Time Memory (STM) by F_2.

After learning and self stabilising, the search process is automatically disengaged after which input patterns directly access their recognition codes without any search. Thus recognition time for familiar inputs does not increase with the complexity of the learned codes and a novel input pattern can directly access a category. A parameter called the attentional vigilance determines how fine the categories will be, if vigilance increases (decreases) due to environmental feedback, then the system automatically searches for, and learns, finer (coarser) recognition categories. A gain control parameter enables the architecture to suppress noise up to a prescribed level, also its global design enables it to learn effectively despite the high degree of non linearity of such mechanisms (Carpenter and Grossberg, 1987).

One of the key computational ideas rigorously demonstrated within adaptive resonance theory (Carpenter and Grossberg, 1987) is that features that have been learned are protected from being 'washed away' by new learning, which enables learning to be automatically incorporated into the total knowledge base of the system in a globally self-consistent way. In summary, it selects new nodes for initiating learning of novel recognition categories, or defends its fully committed memory capacity against being 'washed away' by the incessant flux of new input patterns.

In conclusion, backpropagation neural networks tend to be slow and not applicable to on-line monitoring. The amount of time spent on the training stage is long and therefore inefficient. The use of a SOM is suitable because less time is required for the learning procedure although the interpretation of results tends to be difficult due to the "topographic map" type output structure. The ART2 is a network that consolidates speed and easy interpretation. Unlike the Carpenter and Grossberg classifier, the SOM can perform better in the face of noise because the number of classes is fixed, weights adapt slowly, and adaptation stops after training (Lippman, 1987). The SOM is thus a viable sequential vector quantiser when the number of clusters desired can be specified before use and the amount of training data is large relative to the number of clusters desired. The SOM and ART2 have been implemented for further investigation in this work.

2.3.2 Clustering Methods

Clustering of numerical data forms the basis of many classification and system modelling algorithms. The purpose of clustering is to distil natural groupings of data from large data sets, producing a concise representation of the system's behaviour. In order to interpret the SOM results we need such a method. In the next two sections two methods that generate computationally fast and robust algorithms for data classification are presented.

2.3.2.1 Mapping and Fit Data

As discussed previously, the SOM results in a form suitable for computer interpretation. The topological format of the map resembles a surface where two dimensions are given by the neurones arrangement and the third dimension by the activation of each neurone upon classification with the minimum value representing a data cluster associated to the classified data. In order to characterise the SOM results after training it is necessary to map the results into a grid where clusters are associated with similar test data patterns.

Interpolation between sets can be achieved using several methods such as triangulation, trigonometric interpolation and the weighting method (Crain, 1970; McLain, 1976). For increased simplicity, accuracy and execution speed, it was decided to use the Kriging method (Davis, 1986). This method can be used to make contour maps but, unlike conventional contouring algorithms, it has certain statistically optimal properties (Davis, 1986). The Kriging method uses the information from the degree of spatial dependence between samples along a specific direction to find an optimal set of weights that are used in the estimation of the surface at unsampled locations (Davis, 1986). This method is particularly efficient at estimations along the "borders" of the topological map where data is scarce and sometimes absent.

2.3.2.2 Fuzzy Logic

Fuzzy logic starts with an uncertain idea and aims to transform it into something clear and useful. Fuzzy sets are a generalisation of conventional set theory and were introduced by Zadeh (1965) as a new way of representing vagueness in everyday life. Fuzzy interpretations of data structures are a natural and intuitively plausible way of formulating and solving various problems in pattern recognition (Bezdek, 1993). Fuzzy logic, simply put, transforms vague concepts such as "a little hot", "almost correct", and "very fast" into a mathematical form which can then be used by a system to perform problem solving actions (OMROM, 1992).

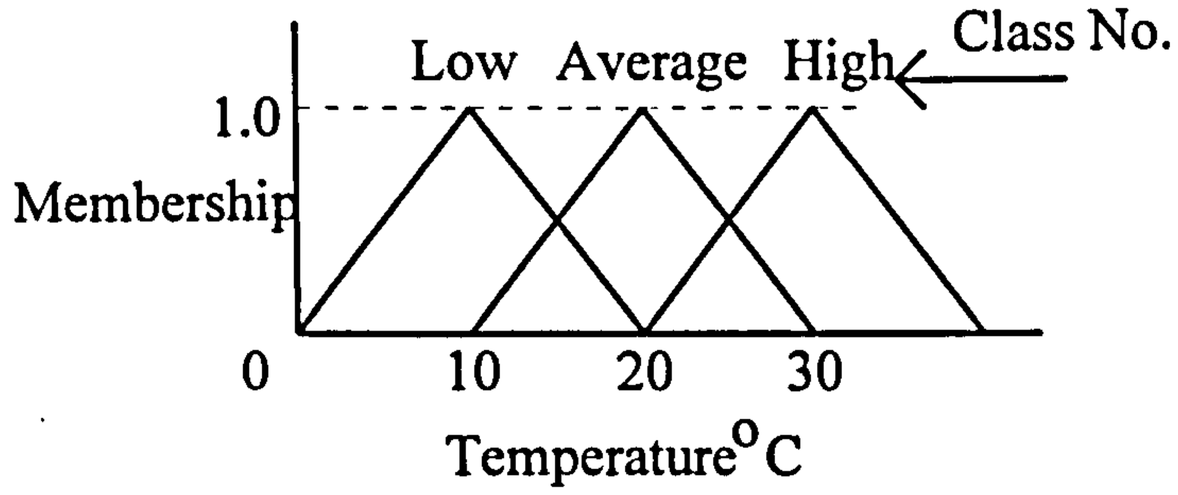


Figure 8: Example: "What is The Temperature?"

Fuzzy logic starts with the concept of a fuzzy set which is a set without a crisp, clearly defined boundary and can contain elements with only a partial degree of membership. The membership function is the basic idea in fuzzy set theory and its value measures the degree to which objects satisfy imprecisely defined properties. Fuzzy logic theory has been applied to many different areas such as; cluster estimation of numerical data (Chiu, 1994), subway control, household appliances (Kosko and Isaka, 1993), and several others (Zimmermann, 1991). Combinations of NN and fuzzy logic have been proposed as a way of achieving better pattern recognition algorithms (Halgamuge *et al.*, 1994).

2.3.3 Signal Processing

The sensor outputs from this work are acquired as a discrete time series of measurements and some methods are required to generate a set of features characteristic of these time series. The time domain techniques are well enough known to require a special presentation here, but a brief description of frequency domain techniques is included below.

Spectral analysis is the partitioning of the variation in a time series into components according to the duration or length of the intervals within which the variation occurs. This is achieved by considering the time series $h(t)$ to be the sum of many simpler time series that have the form of regular sinusoids of differing amplitudes, wavelengths, and phases.

$$H(t) = \int_{-\infty}^{\infty} h(t) e^{-2\pi i f t} dt \qquad [4]$$

If t is measured in seconds, then f is in cycles per second or Hertz. In most common situations, the function $h(t)$ is sampled at evenly spaced intervals. The estimate of the Fourier transform from a finite number of sampled points is called the Discrete Fourier Transform (DFT),

$$H_n = \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N} \qquad [5]$$

The Discrete Fourier Transform maps N numbers (the h_k) into N complex numbers (the H_n), N being a power of two. A fuller documentation can be found in the established literature, as well as DFT algorithms that significantly reduce the computation needed to obtain the power spectrum (Press *et al.*, 1992). These have been used widely to analyse such signals as vibration (Martin *et al.*, 1974; Petrie *et al.*, 1989b; Bonifacio and Diniz, 1994), force (Lee *et al.*, 1989), and sound emission (Lee, 1986; Sadat and Raman, 1987). DFTs are normally referred to as FFTs, although the latter term strictly applies only to continuos signals.

Like the Fast Fourier Transform (FFT), the discrete wavelet transform (DWT) is a fast, linear operation that operates on a data vector whose length is an integer power of two. Both FFT and DWT can be viewed as a rotation in function space, from the input space (time) domain, where the basis functions are the unit vector e^i to a different domain (Press *et al.*, 1992). For the FFT, this new domain has basis functions that are familiar sines and cosines. In the wavelet domain, the basis functions are somewhat more complicated and have the name of wavelets. The wavelet transforms work as a data compressor with the wavelet coefficients representing the signal.

The use of wavelet transforms was investigated (Lin and Ewins, 1993) by comparing their original thrust force data with recomposition of wavelet transformation where it was found that use of zeroth level wavelet transformation coefficients was enough to enable detection of severe tool problems before failure takes place. Chatter monitoring in turning has also been successfully demonstrated using wavelet transforms (Wu and Du, 1996).

2.4 Expert Systems

The principal difference between a knowledge-based system and a conventional program lies in its structure. In a conventional program knowledge is intimately intertwined with software for controlling the application of that knowledge. In a knowledge-based system the two roles are

explicitly separated, the knowledge module is called the knowledge base and the control module is called the inference engine. This explicit separation makes it easier to add new knowledge either during program development, or in the light of experience during the program's lifetime. A rule-based system is a knowledge-based system in which the knowledge is represented as a set of rules known as the rule base (Figure 9).

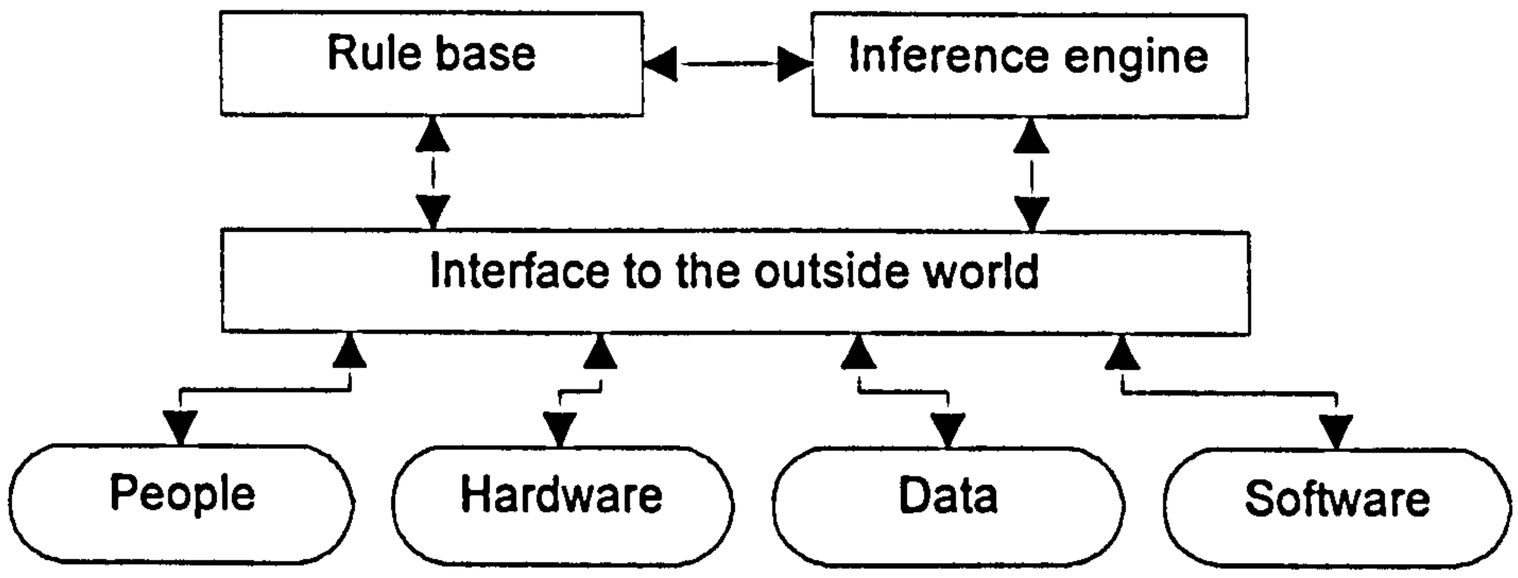


Figure 9: The components of a rule-based system

In an introductory handbook on artificial intelligence, Harmon and King (1985) defined Expert Systems or knowledge-based systems as computer programs that contain both declarative knowledge (facts about objects, events, and situations) and procedural knowledge (information about courses of action) to emulate the reasoning processes of human experts in a particular domain area of expertise. Typically Expert Systems use integral domain knowledge and reasoning strategies to solve problems normally requiring expertise. They usually consist of:-

1. The data base which contains data structures from which conclusions can be drawn. This can be static (unchanging) or dynamic (capable of being updated).
2. Knowledge applied to the data.
3. The inference engine in order to reach the desired conclusion.

The Expert System approach to signal processing attempts to address the apparent limitation of statistical pattern recognition systems by utilising a higher level of information to classify the signal. The approach emulates a human expert's decision-making process by applying expert knowledge (knowledge base) in the form of queries and rules. Expert Systems (ES) can be constructed via 'productions', where a production is an "IF...THEN...ELSE..." rule. These rules can take many forms, but an example we might want children to have is, "IF someone smiles at you, THEN smile back". In a typical production system model long-term memory is built up from a large set of these "IF...THEN...ELSE..." rules. There is also a working memory, i.e. a system holding information that is currently being processed (Eysenck and Keane, 1995).

Expert diagnosticians and equipment operators, whose knowledge is not easily encoded using numerically based algorithms, are often more efficient than currently available automated systems, Agogino *et al.* (1988) and Milne (1988). Yet due to human limitations and to the technical complexities of modern tests it is no longer possible for a field expert to make the best judgement or take the appropriate action in the face of increasing amounts of information. The need for automated diagnostic tools is even more acute in manufacturing applications because of the requirement to process data and respond effectively within a short time (Betancourt *et al.*, 1990). According to Dodhiawala *et al.* (1989), speed alone is insufficient for real-time performance. There are four aspects of real-time processing to be taken into account, these are; speed, or the rate of execution of tasks, responsiveness, or the ability of the system to stay alert to incoming events, timeliness, or the ability of the system to react to meet deadlines, and graceful adaptation, or the ability of the system to reset task priorities according to changes in work based on resource ability.

Knowledge based systems have proved to be particularly effective for performing diagnostic tasks in a variety of domains (e.g. Fikes and Kehler, 1985; Au *et al.*, 1989; Lieslehto *et al.*, 1993). Such tasks involve determining a description of a given process in terms of the parameters the system knows about. An integrated monitoring system, when linked to an expert diagnostic system, is usually required to perform an "overseeing" function and make qualitative judgements about the machining operation in the FMS system. Monitoring involves the evaluation and analysis of a large number of features. For this reason, Expert Systems have a growing importance, interface and operational potentials (Lieslehto and Koivo, 1991; Medsker, 1996).

2.4.1 Potential for the Use of Expert Systems

As mentioned earlier an important advantage of Expert Systems is the ease with which knowledge bases can be modified. As a result, changing the knowledge base does not require programming but can be done via word processing or an editor. Another advantage of expert systems is the number and variety of commercial development systems that have become available over the last few years. These and other aspects of Expert Systems functionality have been previously addressed in the literature (Hayes-Roth *et al.*, 1983; McGraw and Cliffs, 1989).

Expert systems have been used for the monitoring and control of processes in real-time (Laffey *et al.*, 1988; Agogino *et al.*, 1988). The use of sensors gives rise to important differences from the pioneering applications of the expert systems. Here, when the system is running, data must be gathered directly from sensors, instead of being obtained from the operator. This numerical data must then be integrated into the symbolic processing module. Until recently the use of Expert Systems has presented some difficulties in this kind of application, because they were initially designed for domains where static data and non-time-critical responses were required (Barrios *et al.*, 1994). However, recent applications have overcome this difficulty (Laffey *et al.*, 1988) by the application of new techniques, e.g. parallel processing, flexible architecture design, progressive deepening and reasoning.

It is normal to build the knowledge base of an Expert System so that the experience of the specialist is encoded. In extremely complex applications neither do adequate mathematical models exist nor is it feasible to obtain all the knowledge of the experts. This happens, for example, in the real-time estimation of the tool condition in machining processes as the enormous number of variables in the process, some of them difficult to observe and control, makes the knowledge of the best machinists too vague. Therefore, other knowledge is required in order to achieve a diagnosis.

Multi-sensor information is so rich, but at the same time so complex, that it is not easily interpreted. Synthesising inference rules directly from sensor information through the use of neural networks can avoid the above inconveniences.

2.4.2 Knowledge Acquisition

Knowledge acquisition is the most important aspect of Expert System development and also the most problematic. The idea of acquiring knowledge from an expert in a given field for the purpose of designing a specific representation of the acquired information is not new. Reporters, journalists, writers, and industrial designers have been practising "knowledge acquisition" for years.

There are a number of approaches to knowledge acquisition for Expert Systems development, Medsker (1994). One requires the "knowledge engineer" to extract domain knowledge from a human expert and then program it into the knowledge base which at best is very tedious and requires that the knowledge engineer have a multitude of skills (e.g. communication and

conceptualisation) (Stacey, 1994). Nisbett and Wilson (1977) conducted a review of studies to compare retrospective verbalisation with behavioural reports and concluded that inconsistency was common. They found that when asked questions about their cognitive processes, subjects frequently did not base their answers on specific points, but theorised about their processes. When subjects are asked for information that may not be immediately accessible, they are inclined to guess and may offer responses that are inconsistent with previous answers.

Another approach to knowledge acquisition involves analysing examples and previous cases, inductive learning. This mode depicts the source information residing in literature, which could include training manuals, well established knowledge, and other such material. This method has several advantages as the knowledge engineer does not have to be versed in communication skills, the knowledge is ready for implementation and has been proved to be correct for implementation. An example of this is the case of medical diagnoses (Anthony, 1993).

2.4.3 Real-Time Issues

Laffey *et al.* (1988) formulated a definition of real-time systems as: The system's ability to guarantee a response after a fixed time has elapsed. This definition refers to the interaction with an external environment in a timely fashion which is the fundamental feature that distinguishes real-time processing from non real-time processing.

The symbolic processing, characteristic of Expert Systems, influences the design of the system because an effective mapping of the Expert System's operation in the computer system will reduce the semantic gap and improve system performance. In order to take some of the load from the Expert System's inference engine, tedious and time consuming tasks should be organised into separate processing units with autonomous execution; only rules should be allowed into the Expert System environment.

2.5 On-Line Condition Monitoring With Expert Systems and Neural Networks

Rule-based Expert Systems, perhaps the most successful application of AI, have some significant limitations, amongst which are those listed below:-

- The method of acquiring knowledge and translating it into a knowledge base comprising a set of self-consistent If-Then-Else kind of rules is tedious (Milne, 1988).

- The solutions developed for a restricted domain do not easily scale up to handle more complex domains as the system becomes unable to solve interesting problems within acceptable time limits (Heck, 1993).

Fortunately, the properties of neural networks listed below complement the above-cited "brittleness" of Expert Systems.

- As neural networks are trained, not programmed, their behaviour can be dynamically modified (Burke and Rangwala, 1991).
- Neural networks have an ability to generalise, thus permitting training by specific examples (Sorsa and Koivo, 1993).
- As neural networks use distributed representation of the external world, they exhibit only gradual degradation in performance, both in the case of system failure and when the network encounters a problem outside the range of experience.

On the debit side, neural networks cannot explain the reasoning behind their processing as knowledge acquired by the network is encoded in the synaptic weights and it is not easy to decipher the "meaning" of the weights. Also, ANNs are not designed for the exploitation of existing expertise. The development of neural network interfaces to Expert Systems is an active area of research (e.g. Opitz and Shavlik, 1993; Hopgood *et al.*, 1993; Medsker, 1994). In these hybrid systems, a neural network can be used either for knowledge expression or for knowledge acquisition. In the former, the knowledge acquired by an ANN is incorporated into an ES whilst, in the latter, inference rules are derived from the ANN.

Neural network process modelling involves several steps including data acquisition, data pre-processing, variable selection, network training and testing, and network validation. An Expert System can be effectively used to guide each step involved in building a network. The expert knowledge can also be used to improve the techniques such as in variable selection (Qin and Rajagopal, 1993). With the integration of Expert Systems with neural network training, it is possible to hide the detailed neural network technology with the interface. The user can deal with the network training without fully understanding the underlying techniques involved. In the case where the user would like to know more about the technology, the Expert System can provide advisory knowledge (Qin and Rajagopal, 1993). For the above reasons, hybrid systems, comprising neural networks and expert systems, appear to hold promise.

Therefore, neural networks relying on training data to "program" themselves can be useful for hybrid systems, when appropriately trained they enable the system to generalise for operation on future input data. Neural networks can be useful when rules are not known, either because the topic is too complex or no human expert is available, also it has the benefit of easy modification by retraining with an updated data set. Another advantage is speed of operation after the network has trained. Where knowledge is precise and well defined expert systems are most commonly used given their logical behaviour. Finally, since neural networks results are crude and non intelligible expert systems may provide the best interfacing capabilities.

In summary, Expert Systems and artificial neural networks have unique and complementary features that can be successively brought together for use in the challenging problem of tool wear monitoring.

3. Experimental Apparatus and Procedure

The development of the hybrid system was based on a series of machining experiments using the sensor set identified in Chapter 2. In order to perform the required experiments certain details such as sensor positioning and calibration, workpiece/tool material, data acquisition and experimental procedure have to be decided. These preliminary procedures provide a better understanding for the experiments and enable the evaluation of sources of possible inconsistency such as those related to the equipment or mounting. Therefore, it is intended in this Chapter to give detailed information on all the aspects of the experimental environment used to develop the tool wear monitoring system.

3.1 Experimental Apparatus

This section consists of two parts, the first concerning the machine tool and the second describing the sensors and their positioning. Some of the considerations regarding sensor selection and positioning were given in the literature review and therefore will be given less emphasis here.

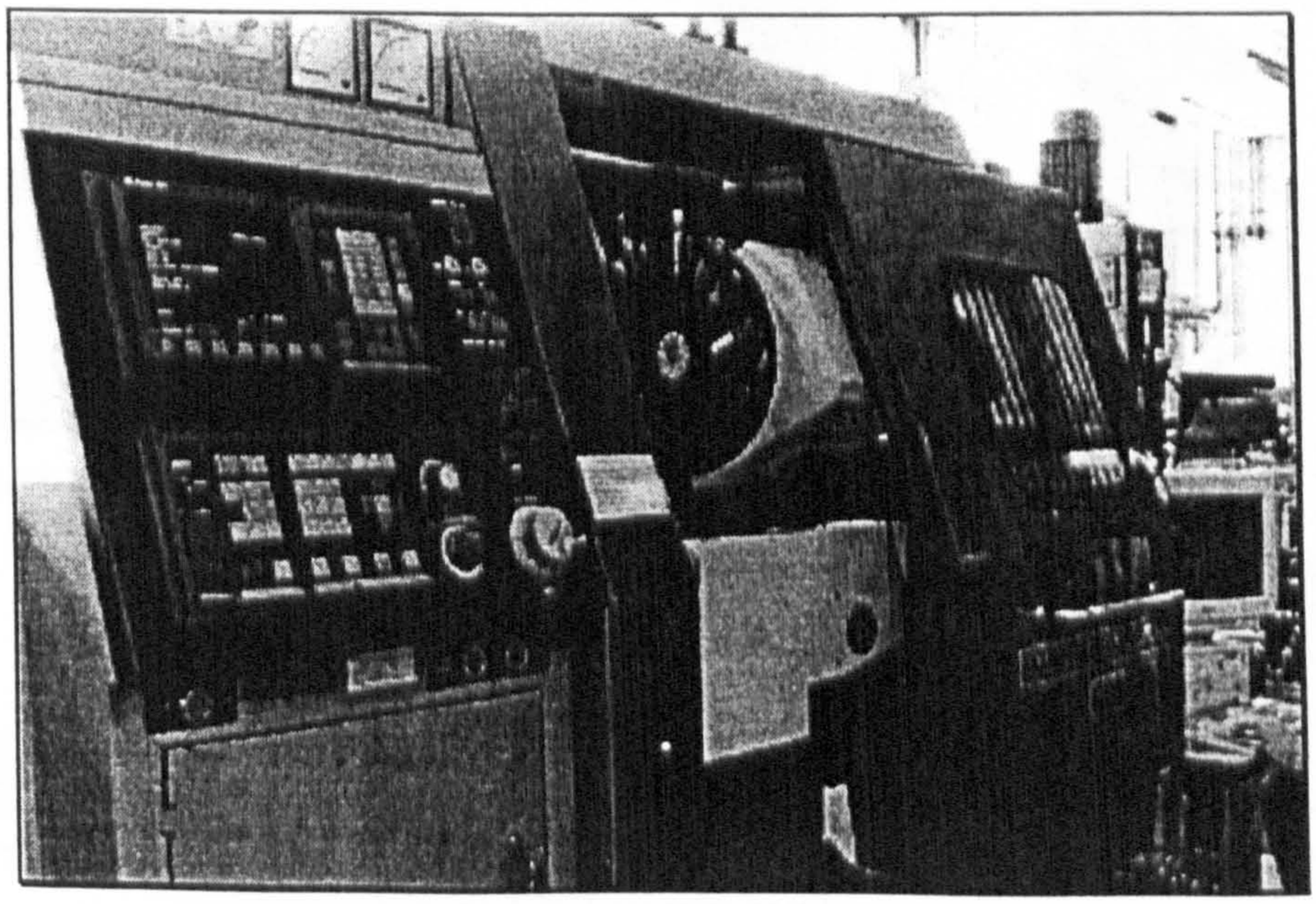


Figure 10: MHP Moog-Turn 50 Slant Bed Turning Centre

All the experimental work was carried on an MHP Model Moog-Turn 50 (MT50) Slant Bed Turning Centre (MHP Machines Ltd.), with standard CNC control. The effective bed size is 500 mm with a DC servo motor of 18 kW driving the spindle. This machine can provide a constant power of 34 kW between 1000 and 3000 RPM (Figure 11), and the range of

admissible cutting parameters is limited by the maximum 4000 RPM imposed by the chuck capacity. The turning centre has the following programme resolution: feed rate 0.001 mm/rev; cutting speed 1 m/min; and depth of cut resolution 0.001 mm.

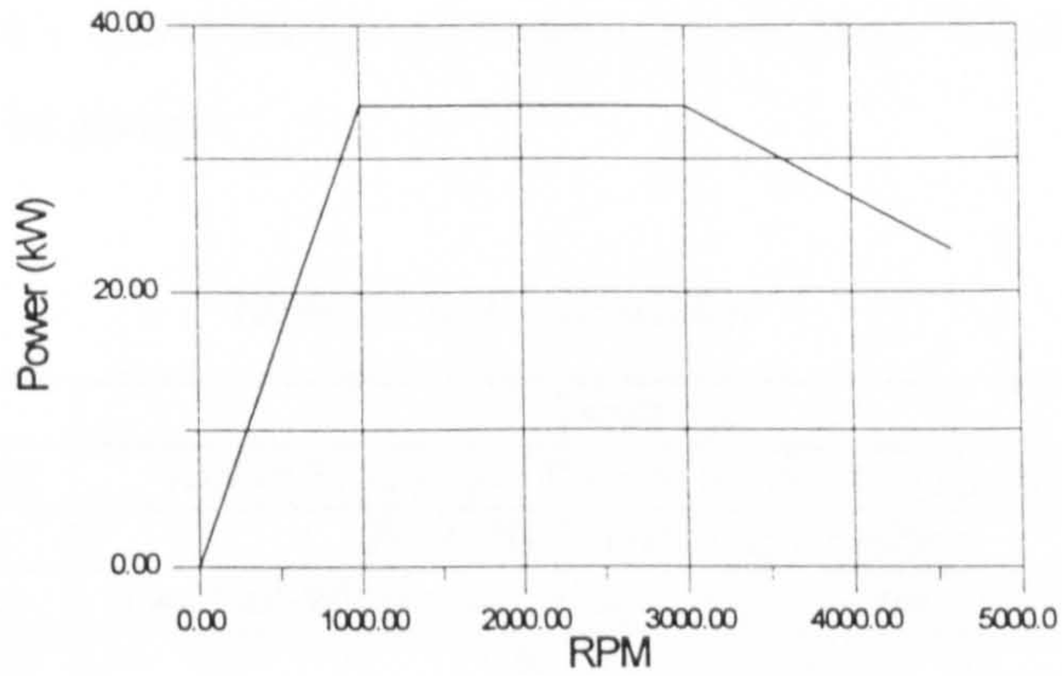


Figure 11: Moog Turn 50 - Power versus speed

The experimental arrangement can be seen in Figure 12, which provides a general idea of equipment layout as well as the data handling approach.

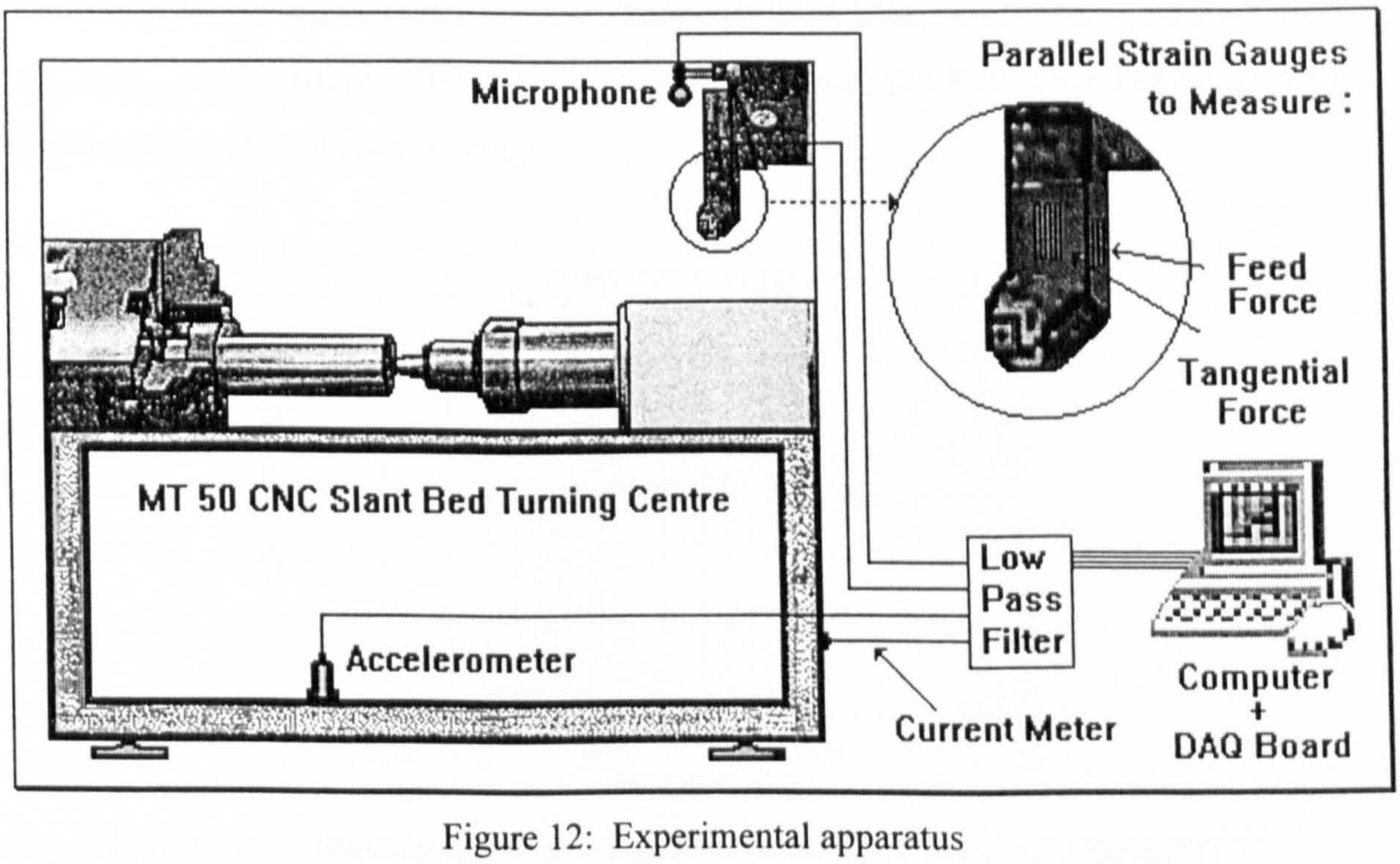


Figure 12: Experimental apparatus

Establishing the optimum sensor position is not an easy task and a balance has to be drawn between proximity to the signal source and intrusion on the cutting process. One of the aims of the work was to utilise sensors in the least intrusive way, so as to facilitate the potential use of the monitoring system in practical situations. Four measurements were selected to monitor tool wear evolution; vibration, sound emission, cutting forces, and spindle current.

3.1.1 Technical Information on Sensors

Table 2 gives the list of sensors used for the present study. This section provides a detailed description of each sensor's characteristics relevant to their purpose. Where possible static and dynamic calibration will be given.

Table 2: Instrumentation

Sensor	Description
Accelerometer	Kistler 8752A50 & Piezotron Coupler - Kistler 5108
Microphone	ECM-1028, matching amplifier
Strain gauges	Two half Wheatstone bridge (amplification - RS 435-692)
Current Meter	CNC built in sensor

3.1.1.1 Accelerometer Technical Data

The accelerometer was calibrated at the origin by Kistler Instrument Corporation using a back-to-back comparison technique against a Kistler Working Standard. The calibration information for this accelerometer is; mounted resonant frequency 32.6 kHz, transverse sensitivity 1.6%, range ± 50 g, and sensitivity 100.2 mV/g. Figure 13 shows the frequency response of this accelerometer as calibrated at the origin.

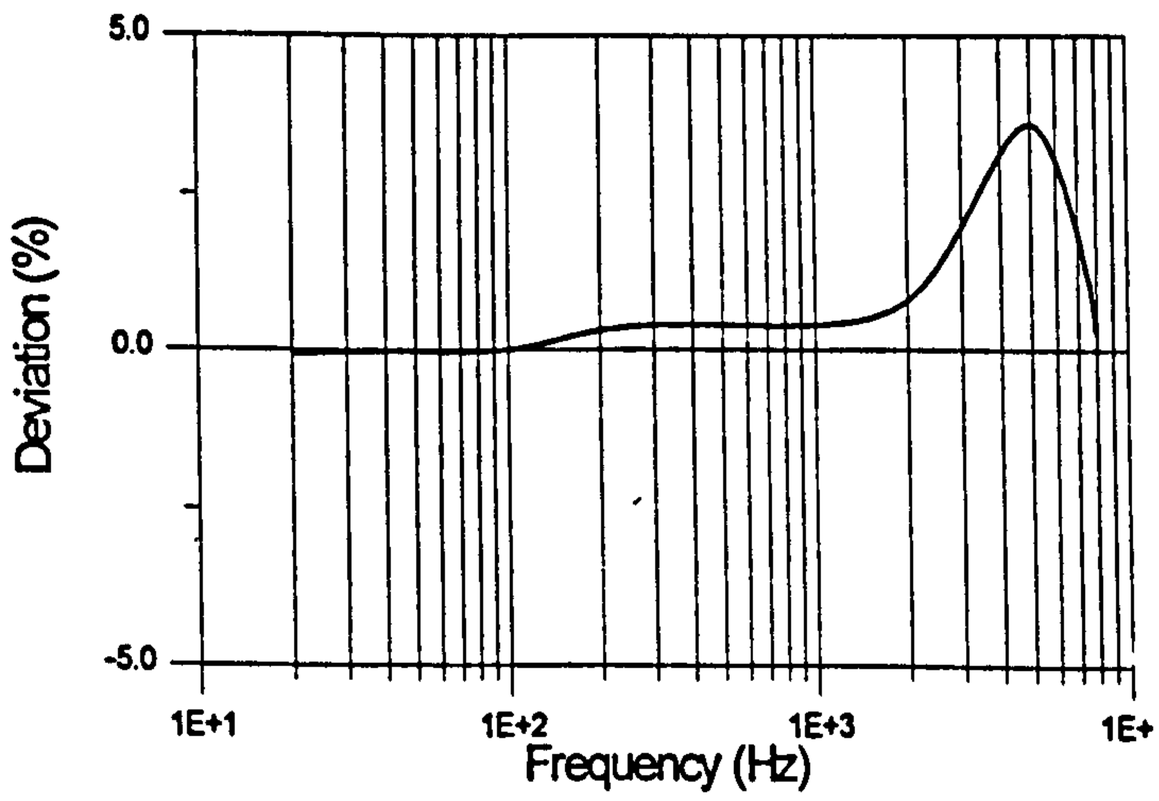


Figure 13: Frequency response of accelerometer (data from manufacturer)

3.1.1.2 Strain Gauge Calibration

Static calibration of both sets of strain gauges was conducted by applying forces of between 600 and 900 N to the feed and tangential directions respectively. The relationships derived from the calibration graphs are:

- Feed direction, $F_f\,(N) = 210.62\,V_{out} + 632.85,\ r_f = 0.99996$
- Tangential direction, $F_t\,(N) = 154.58\,V_{out} + 463.68,\ r_t = 0.999978$

where V_{out} is the measured voltage in volts (v) and r is the linear correlation coefficient.

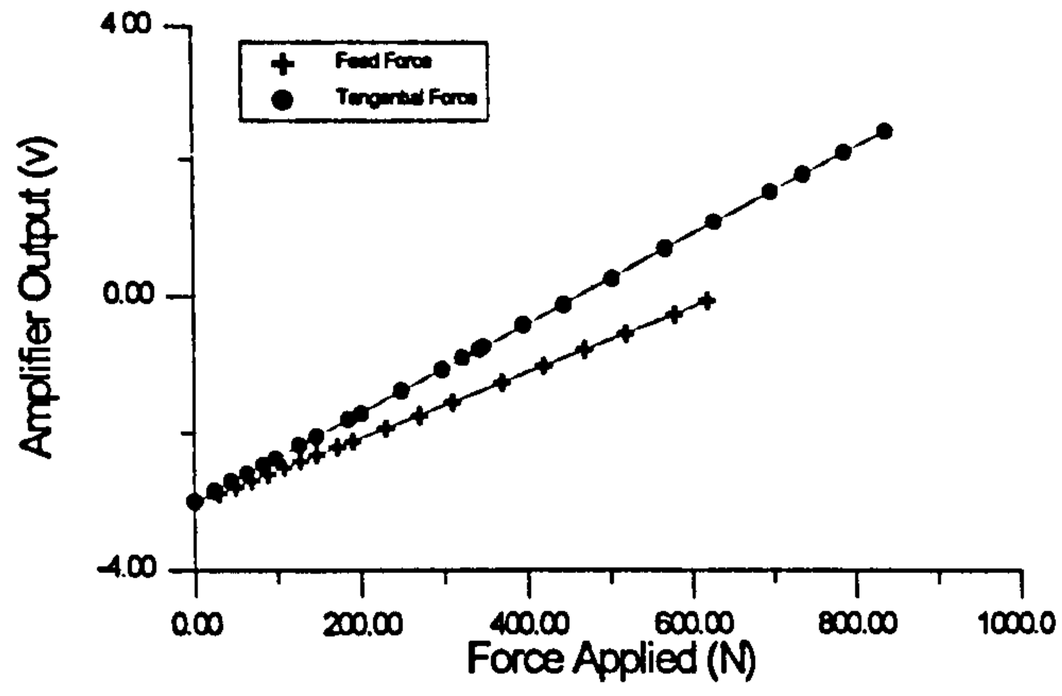


Figure 14: Calibration graph of force transducer

Both the accelerometer and microphone output voltage were set to zero. Given their oscillatory character, amplification was set for an output voltage within [-3,3] v, according to previously acquired data under worn tool cutting conditions. Feed and tangential force output voltage were amplified to give values within [-3,3] v, the lower value (not cutting) was set to -3 v and the upper value (cutting with a worn tool) to be less then 3 v.

3.1.1.3 Spindle Current Measurement

The spindle current was measured directly from the CNC machine. The lathe is equipped with a current meter which is driven internally by the current from the spindle motor. A connection was made to this meter and fed trough an RC filter. At the source this signal is half-wave rectified and can simply be fully rectified to give a constant unidirectional voltage signal.

Figure 15 gives a typical 512 sample acquisition of the voltage meter measuring spindle current, as provided by the machine interface. Spindle current was averaged digitally after acquisition and a sample size of 512 is adequate given the nature of the signal. Each rectified half wave has a frequency of 300 Hz, the original sine wave should have a frequency of 150 Hz due to the fact that the spindle is driven by a three-phase motor

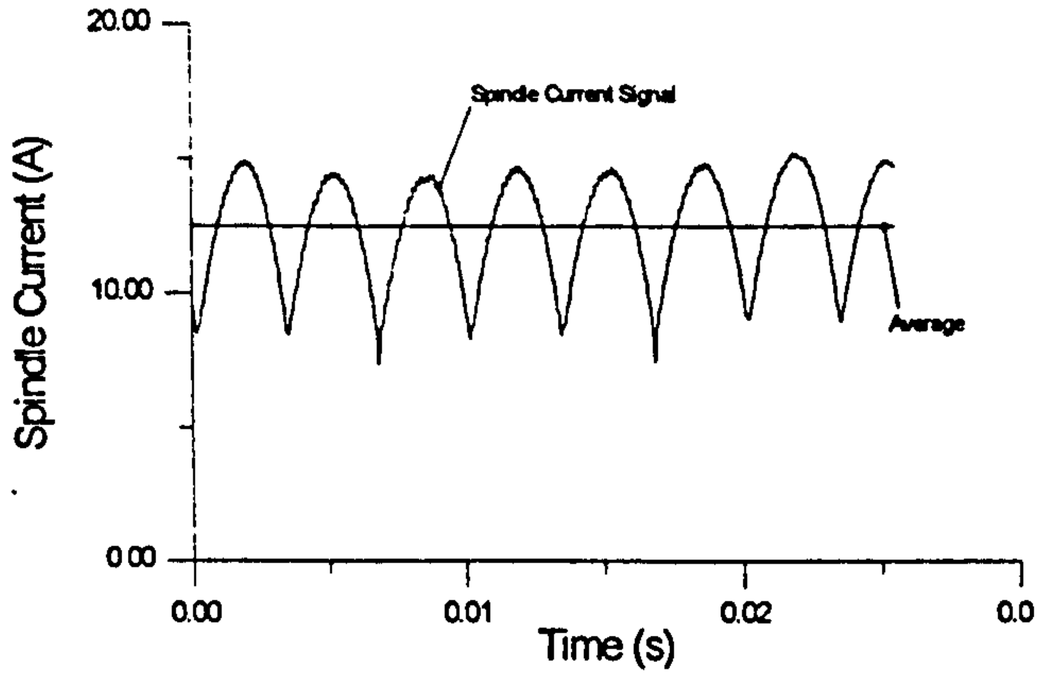


Figure 15: Typical spindle current raw signal

3.1.2 Sensor Mounting and Positioning

A brief description of the motivations for the proposed sensor mounting and positioning will now be given as well as relevant information regarding the latter.

- Microphone - Sound may be defined as any pressure variation that the human ear can detect, in this case corresponding perhaps to the rubbing between the cutting tool and workpiece, and vibration of the tool shank. The microphone was mounted in rubber pads in one turret of the tool handling system so that the signal path remained at 20 cm.
- Accelerometer - The accelerometer was mounted on the base of the CNC machine in a vertical orientation. A permanent magnet was used as the attachment method, which gave a resonant frequency of 7 kHz sufficient for vibration levels up to 2000 m/s^2, which was felt to be adequate for the expected frequencies.
- Force - Strain gauges were attached to the tool holder with epoxy resin and cured at a constant temperature of 120°C for 8 hours before being left to cool to room temperature. They were placed 5.5 cm away from the tool/workpiece interface and arranged to form two full Wheatstone bridges.

3.1.3 Tool Shank and Workpiece Material Details

The tool shank was of rectangular section 2x2.5 cm, having the 2 cm side facing the feed direction. The shank was fixed to the tool holder with two screws overhanging by 6.5 cm. One observation of significance in later discussion was that chatter occurred occasionally at the extreme end of a tool's life typically for $VB_B > 0.35$ mm.

The workpiece, a bar initially 75 mm in diameter and 173 mm long was held at one end by a special 250 mm diameter 3 jaw type 87 international power chuck and supported at the other

end by a motorised programmable tailstock unit. This arrangement provided a relatively stable and rigid workpiece arrangement.

The material used was a cylindrical bar of EN1A[1] (BS 970: Part 1: 1983: 220M07/230M07) machined to a final diameter of 30 mm with 135 mm cut lengths. An insert type HC-P25 grade (WALTER designation of grade WTN 43) coated carbide (CNMG 120408) was used throughout all the experiments. The tool coating is made in two layers by Chemical Vapour Deposition (CVD); an inside one of titanium carbonitride (Ti(C,N)) and an outside one of titanium nitride (TiN), to a total thickness of 12 μm. The angles of the insert seating relative to the tool holder were: cutting rake -6°, and back rake -6°.

3.1.4 Equipment Set-Up

Preparing the equipment for experiments involved calibration and signal level adjustment to ensure the maximum signal to noise ratio.

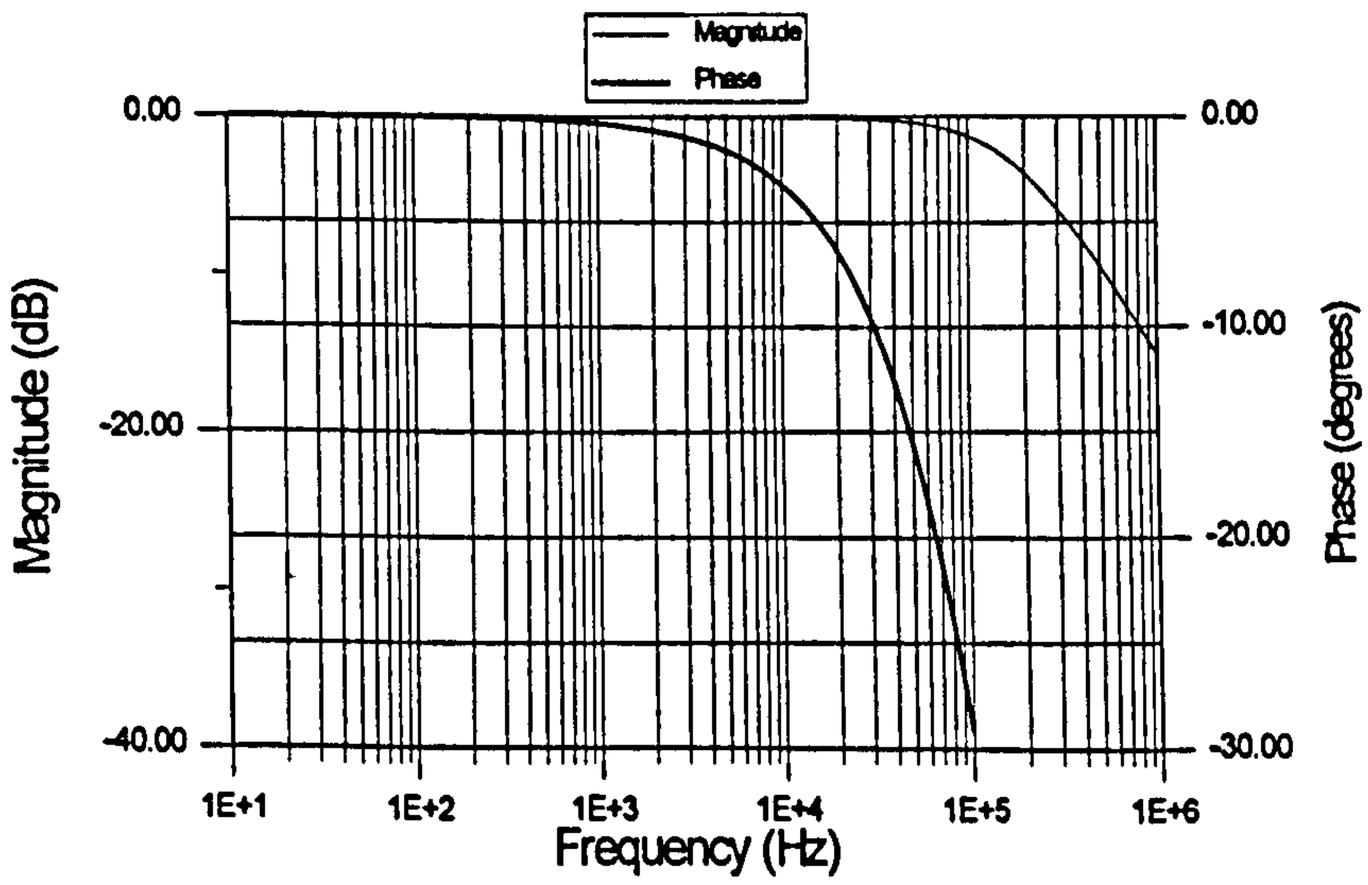


Figure 16: Bode diagram for RC circuit

In order to prepare the experimental apparatus the following procedure was adopted. All amplifiers were arranged to provide an output voltage within the range [-5;5] v, and were filtered by an analogue low pass (RC) filter with dynamic characteristics as in Figure 16. To reduce noise interference due to environmental constraints, shielded cables were employed.

[1] The EN1A specification gives: C 0.15% max., Mn 0.9/1.3%, P 0.07 max., S 0.25/0.35%.

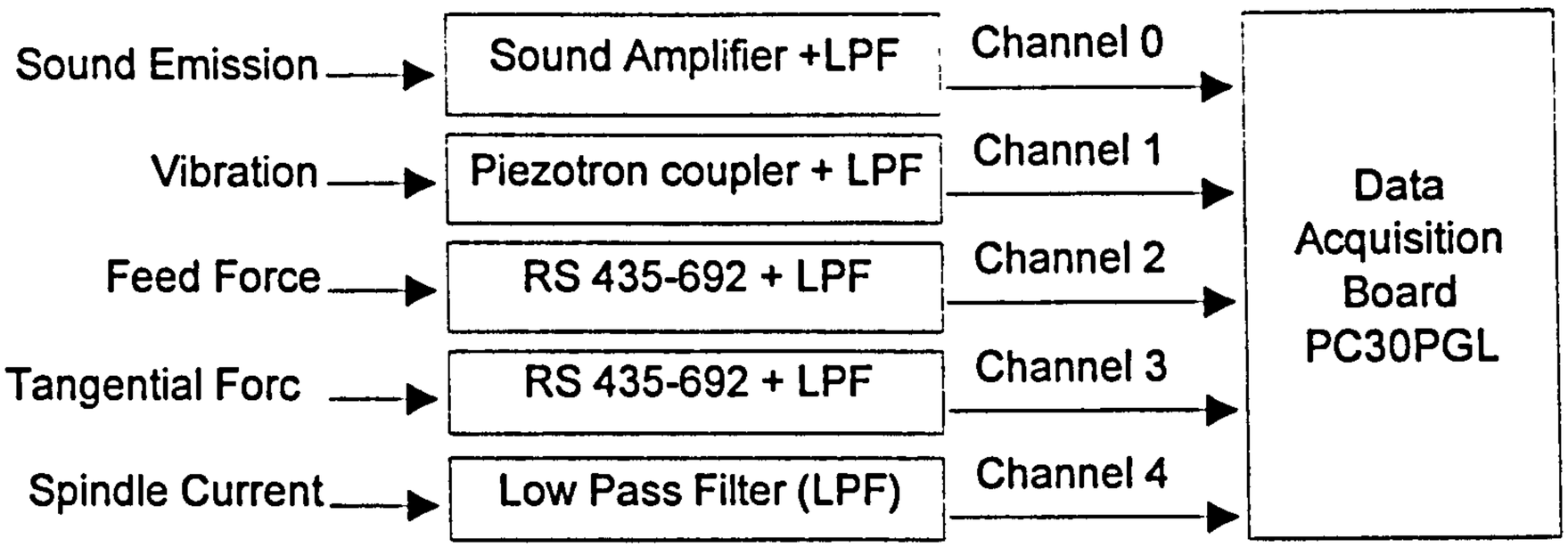


Figure 17: Sensor coupling and channel connection

The following list gives details concerning each sensor circuitry and signal amplification.

- The accelerometer was coupled with the charge amplifier in an arrangement shown in Figure 17.
- A simple circuit gave an amplification ratio of 10/1 for the audible sound (Figure 18).

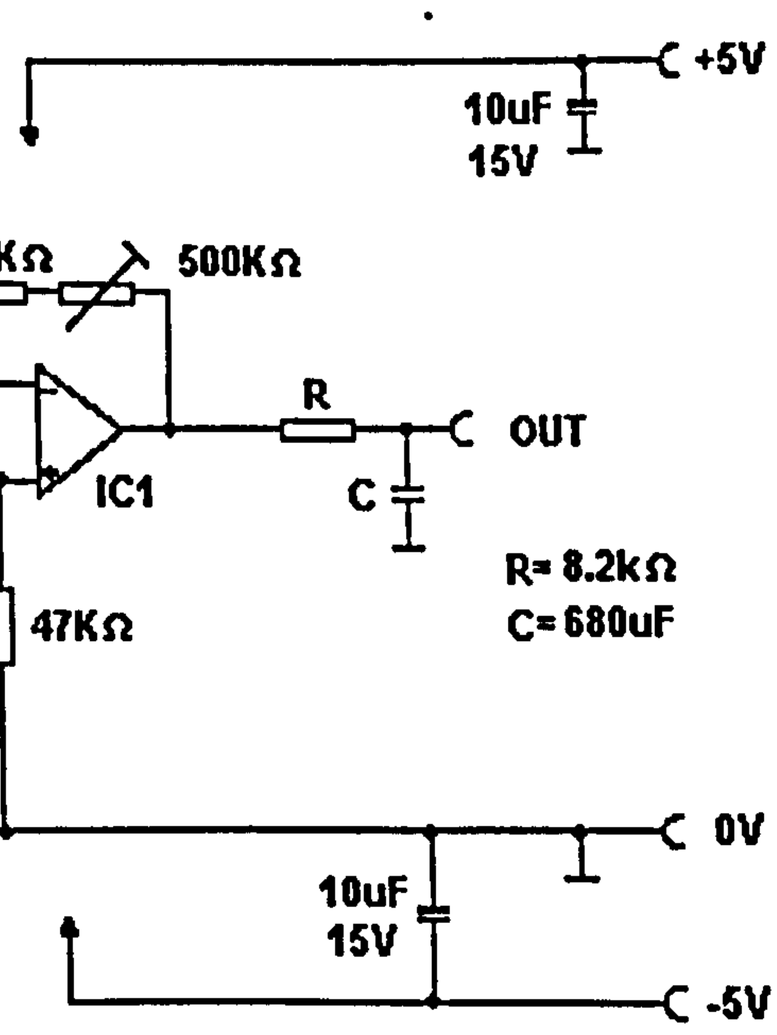


Figure 18: Sound amplification circuit diagram

- The spindle current was rectified (for reasons given earlier) and used without amplification given its adequate voltage level.
- The strain gauges were set-up in a full bridge arrangement (Figure 19) for temperature compensation and accuracy. A Wheatstone bridge integrated circuit (RS 435-692) was used for amplification of the strain signal.

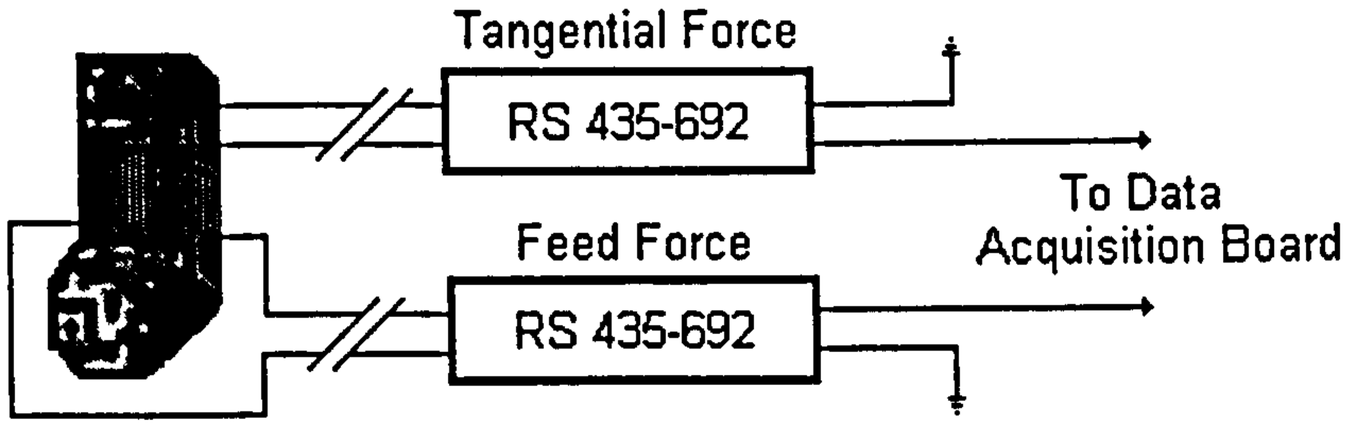


Figure 19: Schematic of strain gauge arrangement

3.1.4.1 Data Acquisition System

The computer used to handle all the data processing and event scheduling was a 486 DX2/66 with 16 Mb of RAM. The computer was fitted with a data acquisition board (Amplicon PC30 PGL) with a maximum sampling rate of 200 kHz, a typical conversion time of 10 microseconds and a 16 way multiplexed input for which Direct Memory Access (DMA) control was available. A sampling frequency of 20 kHz per channel in DMA mode was used and controlled by the on board clock provided by the data acquisition board, this being set via software. The first 5 A/D channels were connected in bipolar mode for noise reduction using a voltage range of ±5 v and a resolution of 12 bits or 0.0024 v. The gain on the board was set to unity. Data gathering while turning consisted of recording at the mid-point of the machined workpiece. After each recording, VB_B was measured with an engineering microscope with a resolution of 0.01 mm. Data was acquired every 2 minutes over the tool life of around 15 minutes (experimentally determined) along with the respective measurements of flank wear.

Calibration of the data acquisition board was done using a three digit multimeter according to the manufacturer's recommendations which assured consistent readings throughout experimental tests.

3.2 Experiments to Determine the Influence of Sample Size on Features

Preliminary experiments were carried out to determine the influence of sample size on the different statistical parameters as well as the frequency spectrum. The effects of sampling frequency and sample length were investigated in order to establish a suitable set of parameters avoiding misinterpretation of data and also the acquisition of excessive amounts of duplicate information. The influence of sample size on data acquired with the microphone can be seen in Figure 21 to Figure 24 (obtained from a full record of sample data - 8192 data points), this

sample data was acquired during normal cutting for a flank wear value of 0.22 mm. It can be observed that a variation in the value of kurtosis and skewness (Figure 23 and Figure 24) occurs with increasing sample size whereas the mean and absolute deviation showed little variation with sample size. The vibration signals showed approximately the same characteristics. In the present study a 512 sample size was chosen to enable processing of the 15 features in real-time even though this may not be the optimum in terms of feature stability.

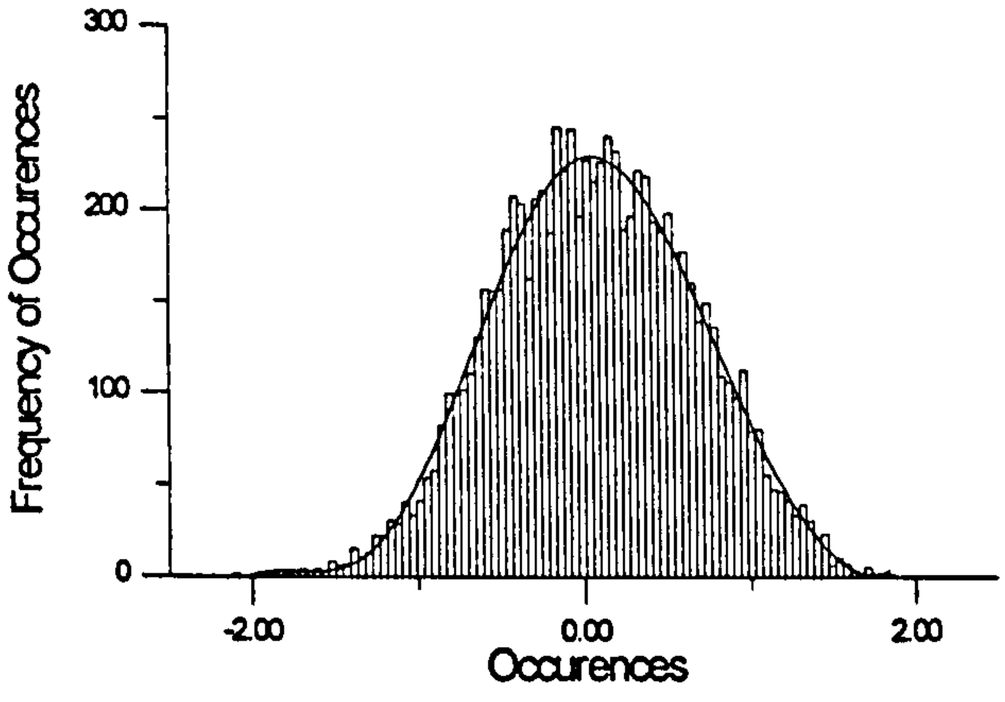


Figure 20: Histogram of occurrences on a 8192 samples of sound

Figure 20 shows that this distribution is slightly left tailed, that is, it would be expected a negative skewness. However, this distribution is not far from a normal (Gaussian) distribution therefore a value slightly lower then zero would be expected. A value for the Kurtosis is more difficult to determine visually since the distribution does not appear to exhibit any tendencies for "flatness" or "peakedness" therefore a value around zero is expected.

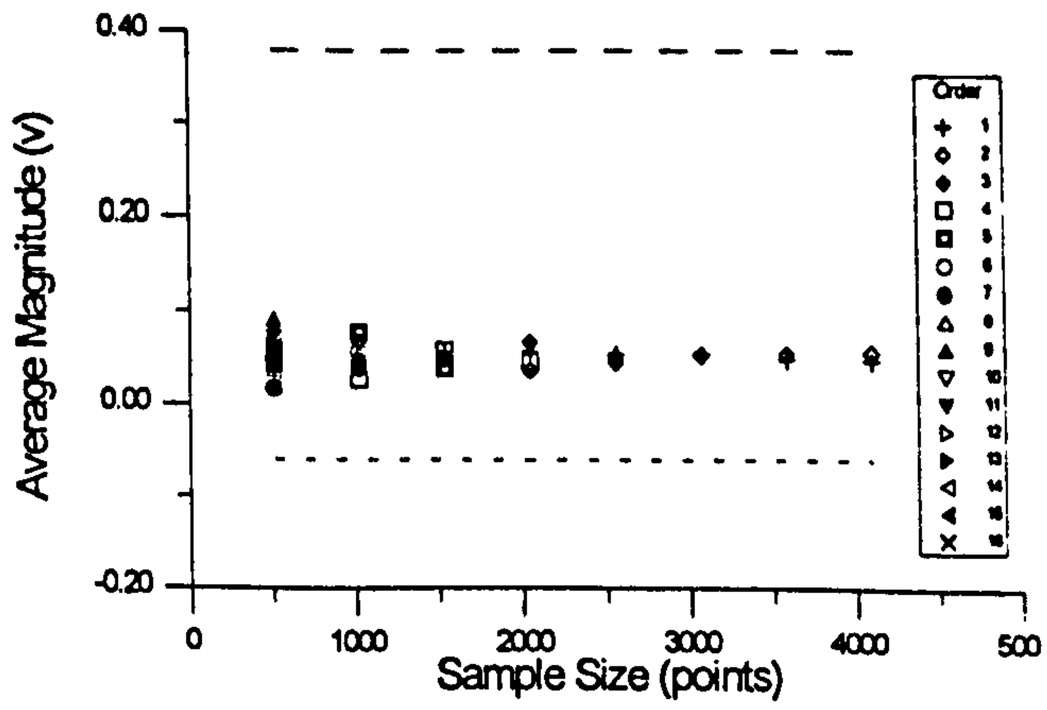


Figure 21: Average of sound for different sample size, $VB_B = 0.22$ mm

Figure 21 to Figure 24 complement the discussion above. The horizontal axis represents the number of points taken into account for the evaluation of a determined parameter and the

vertical axis the value calculated for it. The upper and lower bands were established from the overall data as being respectively the obtained maximum and minimum.

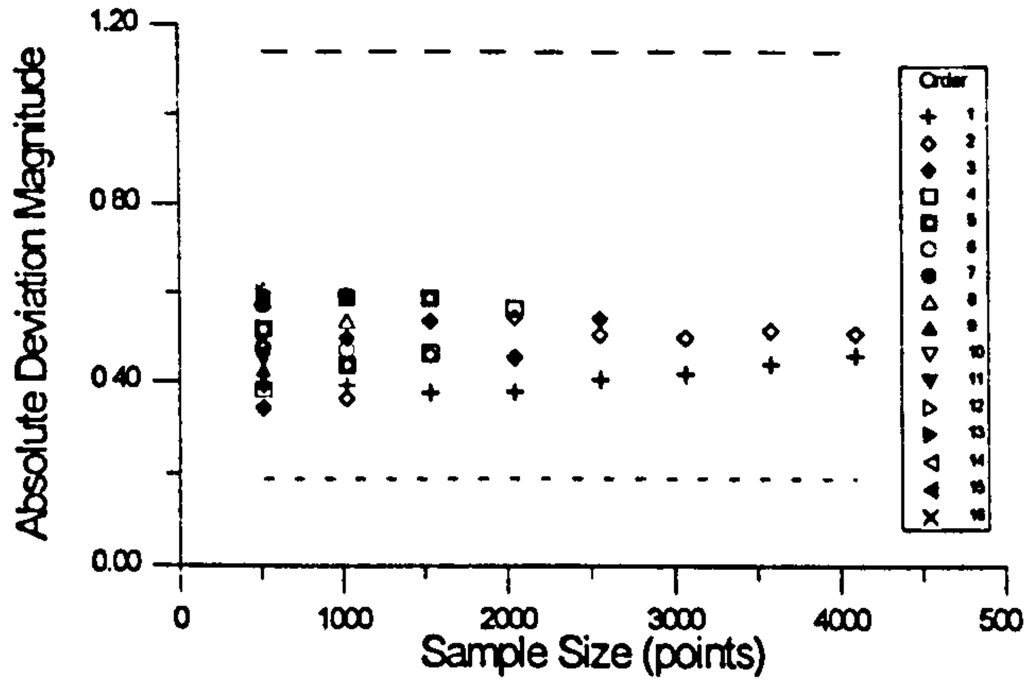


Figure 22: Absolute deviation of sound for different sample size, $VB_B = 0.22$ mm

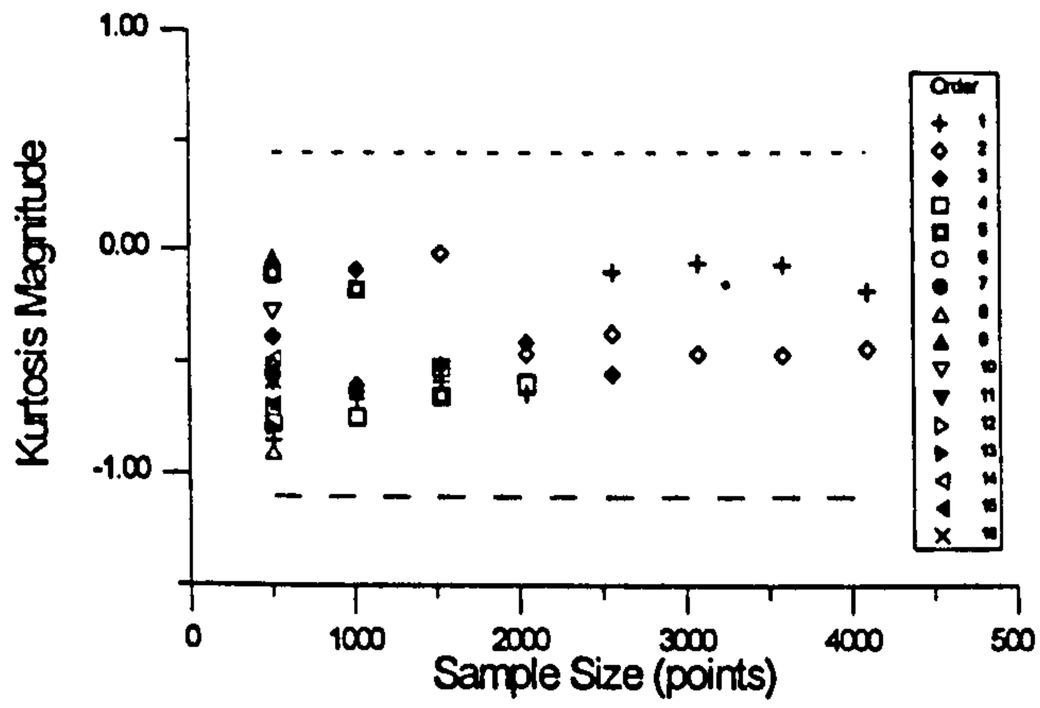


Figure 23: Kurtosis of sound for different sample size, $VB_B = 0.22$ mm

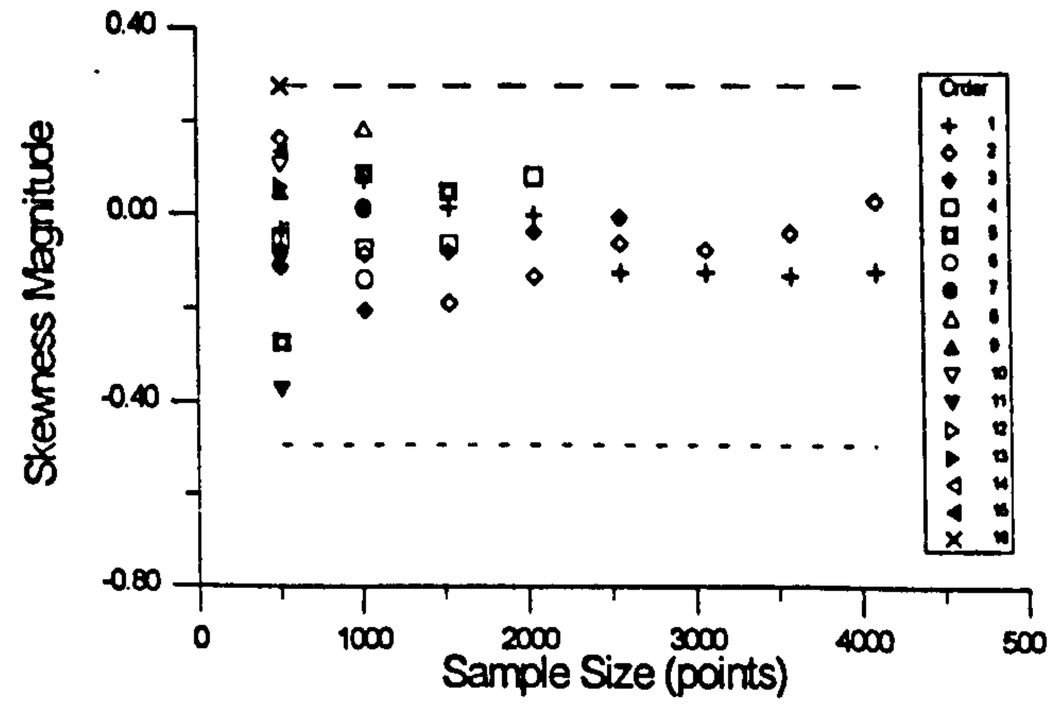


Figure 24: Skewness of sound for different sample size, $VB_B = 0.22$ mm

The power spectrum experienced little change with changes in sample length, and as processing of the spectrum was restricted to estimating the power in a certain band, it was felt that a 512 sample size was acceptable.

3.3 Experimental Method

All the tests were performed under CNC control to ensure repeatability between the experiments. The CNC was programmed to give the desired cutting conditions. As already mentioned, the criterion used to determine tool life was based on ISO3685 (1993), whereby a tool is considered to be worn out if the flank wear, VB_B exceeds 0.3 mm. All the tests were carried out in the absence of coolant.

3.3.1 Feature Extraction

Data was acquired with a sample length of 512 points for each signal over which data analysis was performed using different time and frequency analysis techniques, Table 3. The relevant formulae can be found in literature (e.g. Press *et al*, 1992). Statistical analysis was performed for the first four moments, frequency analysis used the commonly known Fast Fourier Transform (FFT) algorithm using a rectangular window. The power spectrum of each signal was scaled linearly and its resolution was of 39 Hz per division.

Table 3: Data analysis applied to each sensor signal

Signal	Data Analysis
Sound and vibration	FFT, average, absolute deviation, kurtosis, skewness
Cutting forces	Average
Spindle current	Average

Table 4: Feature description

Feature	Processing
Average	Mean value of 512 points
Absolute deviation	Absolute deviation of 512 points
Skewness	Skewness of 512 points
Kurtosis	Kurtosis of 512 points
FFT	2 bands selected from a 512 points FFT

For both sound and vibration signals, the width of the frequency band was chosen to be 200 Hz centred around 2.3 and 4.5 kHz. Signals from both time and frequency domains represent the feature space and are taken as the input to the neural networks.

3.3.2 Experimental Procedure

The experimental work was divided into two main parts, one with a fixed set of cutting conditions, and the second over a range of different cutting conditions. The CNC was stopped

immediately after sample data was acquired to enable the measurement of flank wear. The value of flank wear was then typed into the file containing the relevant sensor data and this was labelled for off-line tests.

3.3.2.1 Experiments With Fixed Cutting Conditions

The aim of this first set of experiments was to gather data to allow training and testing of the monitoring system. The cutting conditions investigated were selected so that the tool would experience realistic production conditions and so that the tool wore out in a reasonable time (cutting conditions: cutting speed 350 m/min; feed rate 0.25 rev/min; and depth of cut of 1 mm), SECO TOOLS AB (1993). These tests were carried out for 6 insert tips giving a total of 52 different wear levels. The 6 inserts gave a standard deviation of 2.1 min (14% of average tool life) justifying once again the use of a monitoring system.

3.3.2.2 Experiments With a Range of Different Cutting Conditions

In order to assess the range of applicability of the tool wear monitoring system it was necessary to collect data from the complete range of cutting conditions applicable to this tool-workpiece combination (SECO TOOLS AB, 1993). The experiment systematically varied cutting speed, feed rate and depth of cut in a coarse manner and a fine manner around the cutting conditions of the experiment in section 3.2.2.1 as the time involved in completely wearing out tools at each point would have been prohibitive it was decided to collect data from three tool states, new; $VB_B \approx 0$, worn; $VB_B \approx 0.15$ mm and worn; $VB_B \approx 0.3$ mm.

According to the range of cutting conditions allowed by such a tool the following limits were established for the cutting conditions; feed rate 0.2 to 0.5 mm/rev, cutting speed 200 to 350 m/min, and depth of cut up to 5 mm. To keep the number of experiments within reasonable limits, tests were conducted with the neural networks on-line to establish the range of adaptability of a network trained for a given set of cutting conditions. Following the determination of the range of depth of cut, feed rate, cutting speed under which the neural networks could still perform tool wear monitoring accurately, finer variations were introduced within the area of tolerance for each of the cutting conditions.

4. Hybrid System Approach to Tool Wear Monitoring

In order to implement intelligent tool wear sensing systems in an automated manufacturing environment, it is helpful to examine how such a function is performed by a human machinist. Human operators detect the occurrence of tool wear by observing the machining operation and evaluating the resulting sensory information as described earlier. The sensory information is probably associated with experience-based memory triggers and a decision is then made as to whether the tool wear level warrants interruption of the process and investigation of a tool changing procedure. The process is one of pattern recognition in which disparate, noisy and incomplete sensory data patterns are used to make a decision as to the level of tool wear.

This chapter discusses the design and implementation of a system for tool wear monitoring. It begins with a discussion of data normalisation, the linking of the neural networks and Expert System and finally the architecture and implementation issues of the unsupervised system.

4.1 Formulation of the Problem

The development of tool condition monitoring systems has attracted a large research effort in the past two decades. Machining conditions of interest include chatter, tool breakage and different states of tool wear. A number of studies have focused on the development of effective monitoring indices, which are sensitive to tool condition but insensitive to cutting conditions, Table 5 summarises some of the results when monitoring tool wear.

Table 5: Application of condition monitoring using sensor fusion

Monitoring Indices	Reference
Ratio of the force amplitude at first natural frequency of tool-holder and the vibration amplitude at the same frequency	Rao, 1986
Frequency analysis of dynamic forces	Choi et al., 1990
Frequency band energy of the tool holder vibration	Sokolowski et al., 1992
Spectral components of the sound radiated during cutting	Lee, 1986
Motor Current mean value	Agogino et al. 1988

Several approaches have been described for the development of monitoring systems by the use of multiple sensors. Table 6 summarises several examples from recent studies applied to the turning process.

Table 6: A list of monitoring indices for tool condition monitoring in turning

Sensor Signal	Monitoring Indices	Classification Method	Reference
Force and AE	Power Spectrum	Neural networks	Burke and Rangwala, 1991
Sound Emission	Power spectrum	Least-squares minimum-distance	Trabelsi and Kannatey-Asibu, 1991
Forces	Force ratio	Neural network	Lee et al., 1996
AE, forces and spindle current	AR models and power spectrum	Neural network	Dornfeld, 1990
Vibration	Power spectrum	Data Dependent System(DDS) modelling	Pandit and Kashou, 1982
Vibration and forces	Ratio between force & vibration amplitude	Wear index	Rao, 1986

According to published results, these monitoring methods may achieve a classification success of 90% or higher in controlled laboratory experiments using the same data for analysis and testing. This assumes that all samples carry the same information which may not be true. This means that the sensitivity and robustness of monitoring systems has not been widely investigated. This section develops a design for a tool condition monitoring system for the turning process with an emphasis on the development of a robust system.

In order to increase robustness two neural networks have been used to interpret the sensor information. An expert system acts as a mediator, synthesising information from different sources, namely; the neural networks, awareness of cutting conditions, workpiece material and cutting tool, cutting time, and empirical knowledge in the form of rules. By combining all this information intelligently it was expected that the system would be robust whilst retaining a high success rate on classifying tool wear.

In summary, this work involves the development of a hybrid neural network/knowledge-based strategy for intelligent tool wear monitoring using multiple sensor information for cutting tool monitoring applications. The neural networks focus on determining tool wear state under normal and off-normal conditions, and this requires an investigation of the temporal signatures of the various sensor measurements. The hybrid strategy focuses on integrating the ANN results with expert assessment and prediction knowledge in order to provide the machine operator with intelligent and consistent recommendations.

4.2 Identification of Neural Network Strategy

This section proposes a system for tool wear classification which employs clustering type neural networks as an efficient alternative to modelling. The 'pure' neural network perspective

might hold that the so called unsupervised networks operate without any labelled training data; further, that their assumed continuously plastic mode of operation requires proof of stability. The pure engineering perspective might mistrust a system for determining tool wear which has no basis in theory or empirical history of materials. The next sections address these reservations and proposes and justifies a practical design.

The advantage of a system which requires no 'teacher' and no labelled training data is obvious. At the very least, unlabelled data costs less (labelled data implies stopping the machine and making measurements of flank wear) and at best the system requires less time and human participation to train. The ideal unsupervised system runs in a continuously plastic mode; that is, weights continue to change as long as incoming patterns are fed to the network. However, plasticity may lead to instability, so the system must possess the ability to stabilise.

A completely unsupervised system, when confronted with an input pattern, will yield one or more integers that indicate to which class it believes the input belongs. If data vectors were to be submitted to such a system, the results would have no meaning unless the correct classification were known *a priori*. For example, a system might assign a pattern to a particular class, but it does not understand what that class represents. In order to assess system performance it is necessary to know the number of classes required and be able to confirm the classification made by the neural network. Thus the so-called unsupervised system for this application at least requires labelled data, as does a supervised system. This unsupervised system still possesses the ability to 'learn' without intervention; i.e. to cluster data. It can adapt and self-organise, without incorporating the external data of the correct classification. The unsupervised system seeks to minimise the error between the input and weight vector whereas the supervised system minimises error between the response of the network to a given input and the expected response to that input.

Although an ideal system might not possess a training period (Burke, 1989) in practice this is not possible as some assessment of the network's performance has to be made. For the ART2, Carpenter and Grossberg (1987) provided in-built stability whilst learning. The SOM achieves the same results by the introduction of decreasing learning rates, which on their own prevent the neural network from becoming unstable. This results in a reduced learning capacity with time in contrast with the ART2's ability to overcome stability without losing plasticity.

Once the system has clustered the training set with sufficient accuracy as indicated by its stability, or some other external criterion, then it must respond to new inputs. If the system weights remain plastic, then the difficulty of interpreting clusters may reappear. The training set definition may lose integrity, since it remains in a learning mode and can gradually redefine previously well understood clusters. Practical considerations in this application necessitate the loss of plasticity, hence the use of a limited training period. In order to achieve the best performance it is necessary to make a compromise between stability and performance; an optimum training period has to be achieved for minimal weight change.

4.3 Neural Network Implementation

When using neural networks it is important to be aware of the effect of computational limitations. Relevant features have to be determined externally and the task of the neural network is to determine the relationship between the incoming data and the tool wear classes.

In the application to tool wear monitoring it is necessary to know the tool wear rate in order to take corrective actions. The following two networks generate approximate values of wear for each set of input features. The objective is to be able to classify the present wear stage to take appropriate action in the future regarding tool change. The previous considerations are not addressed in other research (e.g. Dornfeld, 1990; Burke and Rangwala, 1991; Lee *et al.*, 1996), where a limited number of classes exists for the evaluation of performance.

4.3.1 Self-Organising Feature Map

The SOM typically has two layers. The input layer is fully connected to a two-dimensional Kohonen layer, which is a different structure to multiple hidden layers. In the SOM layer, none of the neurones are connected to each other, regardless of relative position.

The Kohonen layer neurones each measure the Euclidean distance of its weights to the incoming input vector. During recall, the Kohonen neurone with the minimum distance is called the winner. Thus the winning neurone is, in a measurable way, the closest to the input value and thus represents the input value. During training, the Kohonen neurone with the smallest distance adjusts its weights to be closer to the values of the input vector. The neighbours of the winning neurone also adjust their weights to be closer to the same input data vector.

One problem that could arise is that, by chance, one neurone can end up representing too much of the input data. For example, this could appear due to the initial randomisation of the Kohonen neurone weights, where one neurone might end up representing all the data and little information about the clusters of the input data would be found in the Kohonen layer. To solve the problem of a neurone winning too frequently, the mechanism of "conscience" is introduced (DeSieno, 1988). The conscience mechanism depends on keeping a record of how often each Kohonen neurone wins and this information is then used during training to 'bias' the distance measurement. If a neurone has won more than average, then its distance is adjusted upwards to decrease its chance of winning. The adjustment is proportional to how much more frequently than average the neurone has won. Also, if a neurone has won less than average, then its distance is decreased to make it more likely to win. The conscience mechanism helps the Kohonen layer achieve another benefit: The neurones naturally represent approximately equal information, that is, where the input space has high density the representative neurones spread out to allow finer discrimination.

Taking into account what was previously discussed, and based on the above indications, the algorithm will be presented step by step as it was built for the SOM working module (Appendix B gives further details).

Step 1 Initialise weights from the N inputs to the total output nodes M^2 to small random values. Set the initial neighbourhood radius. Initialise neurone winning frequency F_{jk} to $1/M^2$ for all output nodes.

Step 2 Present an input randomly selected from the training set.

Step 3 Compute each distance d_{ijk} between the input X_i and each output node (j,k) using,

$$d_{ijk} = \sum_{i=0}^{N-1} \left(x_i(t) - w_{ijk}(t) \right)^2,$$

where $x_i(t)$ is the input feature to node i at time t and $w_{ijk}(t)$ is the weight from input node i to output node (j,k) at time t.

Step 4 Adjust distance according to neurone winning frequency,

$$d_{ijk\,new} = d_{ijk\,old} + \gamma \left(M^2 . F_{jk} - 1 \right)$$

where $0 < \gamma < 1$ represents the weight of the conscience effect, M^2 the number of neurones in the output layer.

Step 5 Select node (j_0,k_0) as that output node with minimum distance d_{ijk}.

Step 6 Update of neurone winning frequency,

for the winning neurone,

$$F_{jknew} = F_{jkold} + \beta \left(1.0 - F_{jkold}\right)$$

for all other neurones,

$$F_{jknew} = F_{jkold} + \beta \left(0.0 - F_{jkold}\right)$$

where $0 < \beta < 1$ corresponds to the coefficient of distance adjustment, which should be picked so that F_{jknew} does not reflect the random fluctuations in the data.

Step 7 Weights are updated for node (j_0,k_0) and all nodes in the neighbourhood defined by $R(t)$. The new weights are,

$$w_{ijk}(t+1) = w_{ijk}(t) + \varepsilon(t).h(t).\left[x_i(t) - w_{ijk}(t)\right],$$

The term $\varepsilon(t)$ is the learning rate ($0 < \varepsilon(t) < 1$) that decreases linearly in time, the function $h(t)$ provides the lateral interaction between the neurones.

Step 8 Repeat by going to Step 2

The SOM implementation consists of three major components; input vector normalisation, training, and map interpretation. The SOM relies on the normalisation in order to avoid overloading of variables and is carried out by determining the maximum and minimum for each feature so that the relative magnitude and variation is preserved. The training period was set experimentally by inspecting the results from different number of epochs using the guidelines given by Kohonen (1990). Upon training, the weights start to stabilise until there is no significant change in their value, at this stage care has to be taken that the network does not become over-trained. If over-training occurs, generalisation upon new data is impossible and the network becomes inaccurate in its classification ability when presented with new data.

In order to evaluate the performance of the proposed network it was necessary to interpret the SOM topological output. This was achieved by using the Kriging method for surface meshing (Davis, 1986). This meshed surface is stored in a file ready for classification of a test sample.

4.3.2 Adaptive Resonance Theory Module

ART networks are most easily understood as devices for classifying input patterns. The goal is to present a sequence of patterns to such a network and have each "appropriately" classified by this network. An ART algorithm can classify and recognise input patterns without ever having a teacher present. The heart of an ART network consists of two interconnected layers of

neurones, F_1 and F_2, which comprise the attentional system. The input leads to activity in the feature detector neurones in F_1. This activity passes through connections (synapses) to the neurones in F_2. Each F_2 neurone adds together its input from all the F_1 neurones and responds. Furthermore, in one version of ART, presented here, the neurones in F_2 compete with each other, so that at any instance, at most one neurone is active. This simple overview of the ART2 algorithm is extended in Appendix A.

The detailed equations of the ART2 dynamics are provided in Appendix A but a summary would be:

Step 1 Initialise the following parameters: a, b, c, d, θ, ρ.

$$t_{ji}(0) = 0,$$

$$b_{ij}(0) = \frac{1}{\sqrt{N}},$$

$$0 \le i \le N-1, 0 \le j \le n,$$

where N is the dimension of the input vector and n is the number of F_2 nodes,

$$w_i = x_i = v_i = u_i = q_i = p_i = 0.$$

Step 2 Apply a new input vector.

Step 3 Calculate the F_1 activities using Equations [A.1] - [A.8].

Step 4 Compute the F_2 matching scores according to Equation [A.9].

Step 5 Choose the active F_2 node using Equation [A.10].

Step 6 Activate the Jth F_2 node.

Step 7 Perform the vigilance test according to Equations [A.12] and [A.13].

Step 8 If

$$\frac{\rho}{\|r\|} < 1,$$

then deactivate the selected F_2 node and go to Step 4.

Step 9 Adapt bottom-up and top-down weights according to Equations [A.14] and [A.15].

Step 10 Repeat by going to Step 2.

For the ART2, the three major design issues are; normalisation of the input vector, training, and tuning the network. For tool wear identification, the magnitude of tool wear related features has been found to be an indicator of tool wear. However the ART2 algorithm relies on an explicitly normalised input vector and automatically normalises the vectors during processing. It was

therefore decided to use the normalised inputs taking into account their limit values, thus adopting a fixed scale for each feature .

Training of the ART2 starts with uniform initial weight vectors, all equal to $N^{-1/2}$ (N is the number of features). The duration of the training period is set according to the network's performance, during which weights adapt and after which weights are frozen, and this is based on tests of stability. This policy relies on the understanding that the coarse encoding produced by a limited training period will not significantly differ from that obtained after more cycles.

Network tuning is probably one of the most important points in the implementation of the ART2 network. In order to achieve a good resolution, that is, a suitable number of representative wear clusters, the vigilance parameter has to be chosen appropriately. This means that the number of clusters has to be large enough to provide a suitable number of wear classifications without increasing mis-classifications and therefore has to be set experimentally.

The first step in using unsupervised networks is to interpret clusters and use them for classification. It was felt that, in order to associate different wear levels to the clusters created, it would be necessary to use "supervised" samples. The approach here uses a classification concept similar to a hit ratio where the average of wear values associated with the "supervised" samples is calculated for all the samples falling into each of the clusters. An assessment of performance was made by plotting the classification against measured values, the linear correlation coefficient giving a measure of performance.

4.3.3 Neural Network Policy Development

From the design issues discussed above and the practical requirement of tool wear classification, the best policy will emerge after testing. However, based on the abilities and limitations of the unsupervised system and on preliminary observations the policy outline will be described here.

In real-time, the only available information concerning a configuration's success will reside in its training performance. The ideal policy will recommend employing a neural network exhibiting "good" sample set classification. The testing to be performed will assess the validity of such a policy for competitive learning, i.e. it will observe its generalisation ability. In addition, testing will identify the configurations which typically yield good results, and mark

them as good candidates for the application. In other words, good choices for the training period and vigilance parameter are hoped to be found. Two policies exist for training pertaining to weight update. In this work the policy dictating that weights freeze after "sufficient" training is followed because this provides better control over test classification.

The ART2 and SOM neural networks were implemented in the 'C' language, based on the respective algorithms. Both modules have two modes of operation, one for training and another for classification which is controlled by the expert system.

4.4 Expert System Development

The ANN interpretation knowledge base maps the outputs of the ANN into specific detector states. This is done by determining their membership value which measures how worn the tools are in terms of a threshold value selected according to tool wear criteria or finish requirements. The final result is then selected based on a reliability measure for the neural networks which takes into account their performance.

Knowledge is encoded in the form of rules which enable the expert system to perform any reasoning required for tool wear prediction. The priority of a rule allows the determination of the order of precedence in the reasoning path when more than one rule applies.

4.4.1 Improving the Effectiveness of the Stand-Alone Neural Networks

In order to reduce the effects of mis-classification by the two ANNs rules have been encoded which encapsulate Taylor's tool life equation. The prediction made by Taylor's tool life equation ($VT^n=C$) is used to establish preliminary wear intervals with mis-classifications being removed if they fall outside these intervals. The set of rules devised to establish classification confidence limits are shown in Table 7 and exemplified in Figure 25.

Table 7: Rules based on cutting time

Rule No	IF	THEN	Priority
1	VB_B (NN) > VB_B (Taylor) + 0.15	Exclude	2
2	VB_B (NN) < VB_B (Taylor) - 0.15	Exclude	2

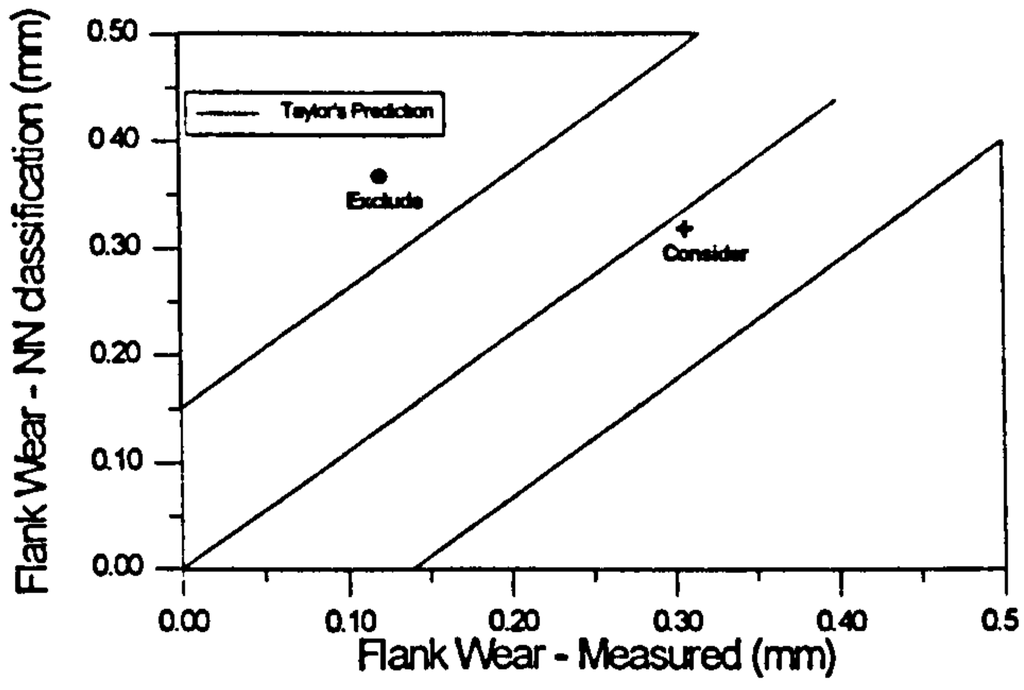


Figure 25: Outlier detection example

The parametric values of Taylor's tool life equation for the workpiece and tool material, that is *n* and *C*, have to be known in order to determine the above rules, and for a given set of working materials and cutting tools a data base could be constructed to cover a typical workshop. For the workpiece material and cutting tool used in this study, these values were set to C = 823, n = 0.33 (Shaw, 1989).

4.4.2 Interpreting Uncertainty Using Fuzzy Rules

At this stage tool wear level is classified in order to establish the degree of wear, that is, the interpretation made by the ANNs combined with the outlier detector is mapped onto different levels using a characteristic function which returns a value in the interval [0;1], thus allowing a continuous grading of set membership between 0 and 1. The membership function [6] determines how worn is the tool (Figure 26).

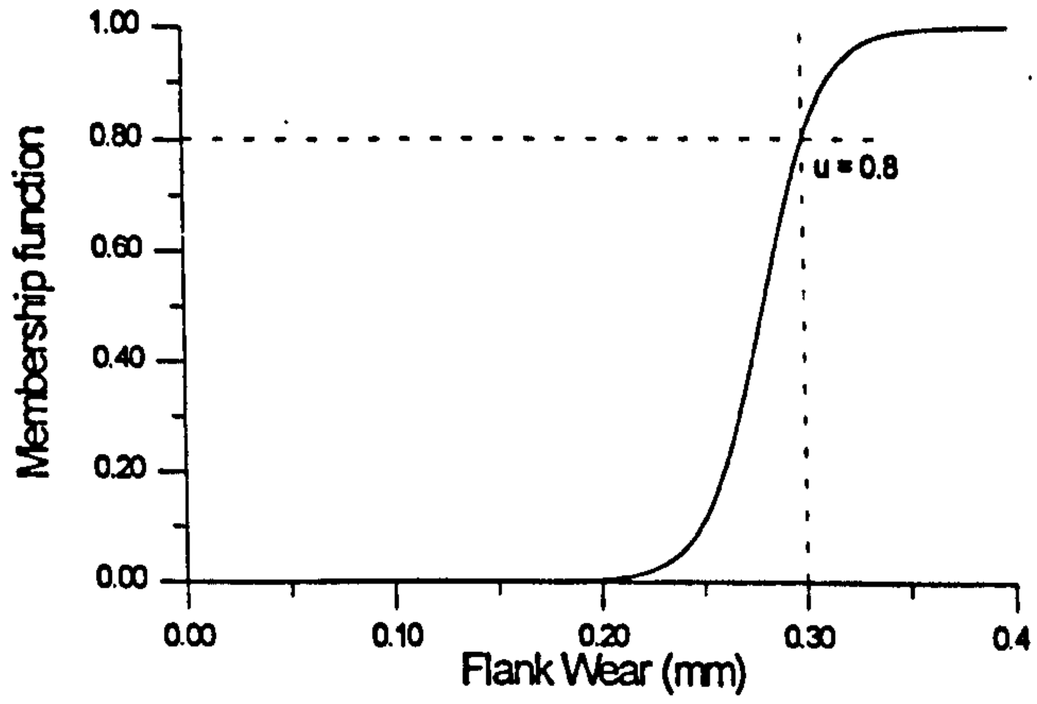


Figure 26: Worn out - Membership function

$$\mu(VB_B) = \frac{1}{1 + e^{-70(VB_B - VB_{REF})}} \qquad [6]$$

Membership of a class is defined by the sigmoid function which uses VB_{REF} as an exponent constant. This value might be selected to account for surface finish, machine stability, or different wear criteria. It is also important to account for small deviations from the predicted results, this may be referred to as a safety margin. For this work a value of $VB_{REF} = 0.28\ mm$ was found to be suitable, giving $\mu(0.3)=0.8$. A fuzzy membership function avoids the hard barrier created by crisp sets (worn or not), thus for each classification a grade is given which specifies how worn is the tool. The corresponding rules implemented in the current system are shown in Table 8, these, effectively, account for outliers or calculate their membership value according to the previously achieved by rules 1 and 2.

Table 8: List of rules to determine membership of non-outliers

Rule No	IF	THEN	ELSE	Priority
3	VB_B (ART2) ≠ Unknown	μ (VB_B (ART2))	P_{ART2} + 1	1
4	VB_B (SOM) ≠ Unknown	μ (VB_B (SOM))	P_{SOM} + 1	1

Neural network performance is taken into account by keeping track of previous failures recorded by the outlier detector (P_{NN} value). Membership values are only calculated for predictions that are found to pass the outlier detector, otherwise they are accounted for the reinforcement of a neural network's ineffectiveness towards tool wear classification (P_{NN} value).

4.4.3 The Use of Historical Data

In order to increase the classification confidence, each prediction is compared with up to *Np* previous classifications which allows the expert system to establish the final decision of the neural networks (rules 5 and 6). The analysis consists in obtaining the linear correlation coefficient for the last *Np* samples.

$$r_h = \frac{Np\sum_{i=1}^{Np}(VB_i.S_i).-\sum_{i=1}^{Np}VB_i.\sum_{i-1}^{Np}S_i}{\sqrt{\left[Np\sum_{i=1}^{Np}(VB_i)^2-\left(\sum_{i=1}^{Np}VB_i\right)^2\right].\left[Np\sum_{i=1}^{Np}(S_i)^2-\left(\sum_{i=1}^{Np}S_i\right)^2\right]}} \quad [7]$$

Table 9: List of rules to account for historical weighting

Rule No	IF	THEN	Priority
5	VB_B (ART2) ≠ Unknown	r_h (ART2)	1
6	VB_B (SOM) ≠ Unknown	r_h (SOM)	1

where S_i are the previous *Np* samples from previous predictions. If the last *Np* predictions are well correlated a value near *1* is achieved for r_h. Although the evolution of wear with time may not be linear it is possible to assume a linear relationship in small intervals of time, corresponding to the *Np* previous samples. Due to sparcity of data obtained during experimental work an *Np* value of 3 was used.

4.4.4 Tool Wear Diagnosis

Finally, to obtain the overall assessment of tool state it is necessary to examine the results of the combined rules 3 to 6. Since the aim is to determine whether the tool has failed due to excessive wear, a *Goal* has to be established within the expert system to ask, "In What State is the Tool?", this goal triggers all the process of rule interpretation. If the ANNs agree on the state of the tool, or one reports an *unknown* state, a solution is possible, otherwise tool state will be resolved from a reliability perspective. Reliability is determined using the NN wear prediction correlation (determined from previous data) and the present prediction. Thus, a poorly correlated prediction history reflects the inability of the neural network to perform tool wear classification effectively.

The diagnosis consists on the integration of both neural network predictions using the weights provided from the analysis of their performance, based on the evolution of historical data, r_h, and successful classifications obtained, P_{NN}. The prediction is assumed based on their reliability, therefore the most reliable neural network controls the final result.

Table 10: Rule to resolve conflicts between predictions

Rule No	IF	THEN	Priority
7	P_{NN}(ART2)/ r_h(ART2) < P_{NN}(SOM)/r_h (SOM)	ART2	0
8	P_{NN}(ART2)/ r_h(ART2) >= P_{NN}(SOM)/r_h (SOM)	SOM	0
9	Both Unknown	Unknown	0

Table 10 shows the rules which resolve the conflict between neural networks by comparing their reliability and historical success based on up to date information processed each time a verdict is triggered by a new data sample.

4.5 Hybrid System Development

The prototype tool wear monitoring system is a hybrid integration of the neural networks and the expert system knowledge base described in the previous sections. Within this hybrid architecture, the neural networks are employed to detect the state of the tool as it changes with time. The knowledge based expert system is used to determine confidence limits of tool wear stages via the Taylor's tool life equation, interpret the ANN results, and provide an overall monitoring assessment.

A proprietary environment (KAPPA-PC) was used to combine the neural network and expert system technologies and to develop the tool wear monitoring system. Although all of the software framework and most of the knowledge base was embedded in KAPPA-PC some numerically intensive processing was performed by external 'C' programs (Appendix C), for example the neural networks.

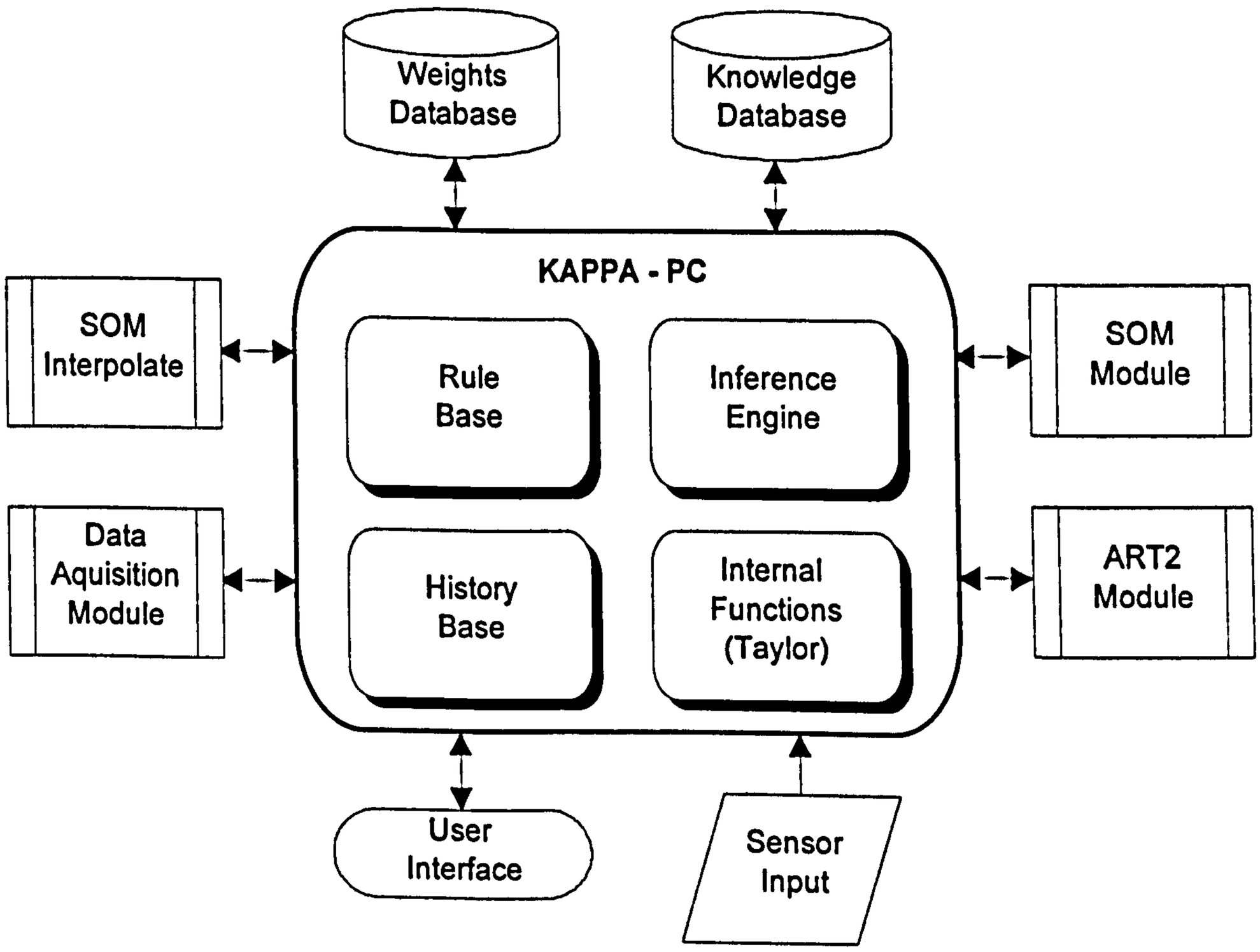


Figure 27: Expert system structure

KAPPA-PC allows the rapid development of applications in a high-level graphical environment, generates standard ANSI C code and provides a wide range of tools for constructing and using applications. In the KAPPA-PC system, the components of our domain are represented by structures called objects. Objects can be either classes or instances within classes (e.g. *Tool:State* where *Tool* is the class and *State* one of the instances). The relationships among the objects in a model can be represented by linking them together into a structure called a hierarchy, as illustrated in Figure 28. This figure gives a representation of the hierarchical structure used to build the present monitoring system.

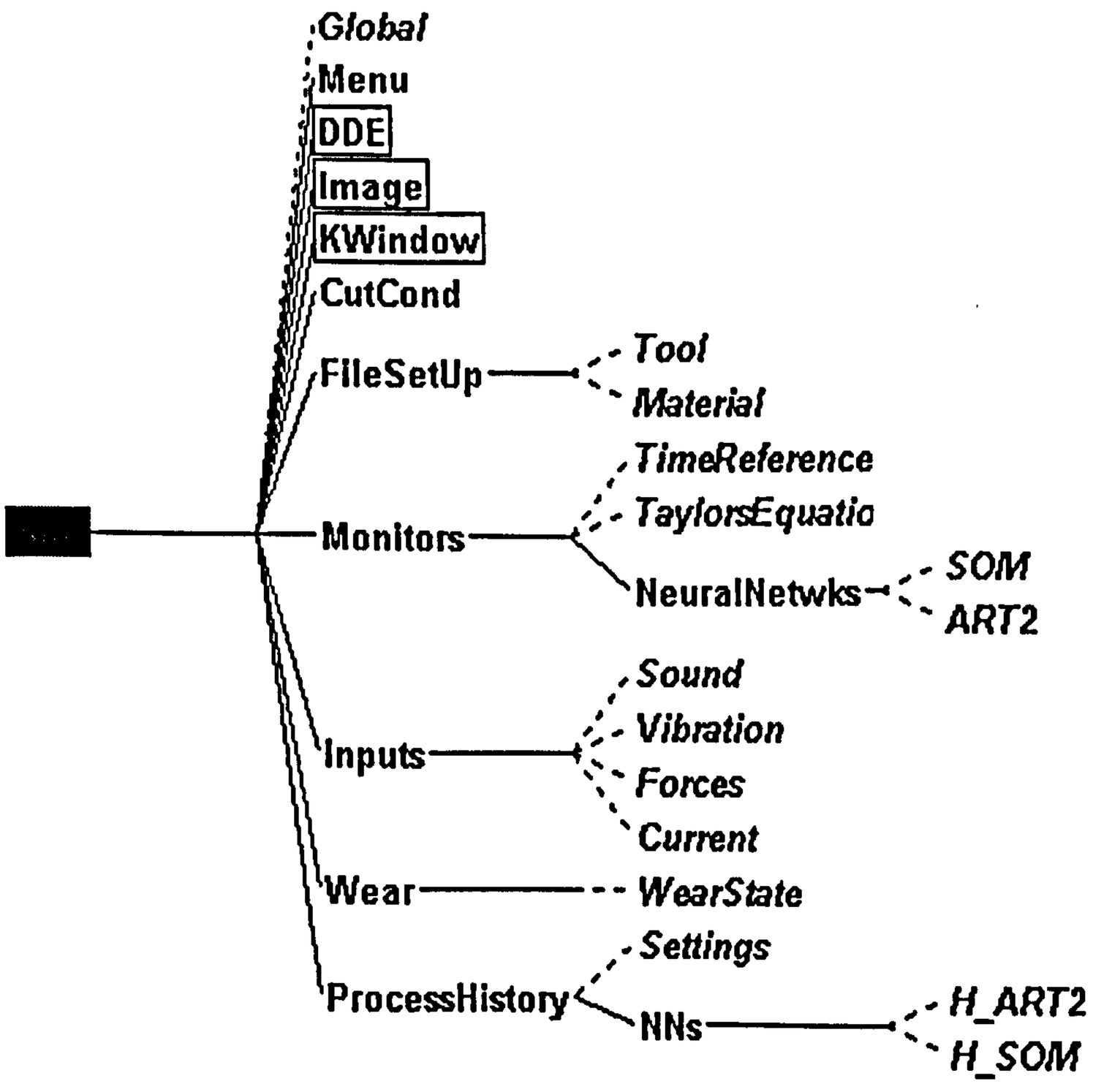


Figure 28: Hybrid system frame-based representation

Figure 28 shows how the knowledge is organised into classes and subclasses in the Expert System. The "Global", "Menu", "DDE", "Image" and "Kwindow" classes are standard in the KAPPA-PC shell Expert System. The classes "CutCond", FileSetup", "Monitors", "Inputs", "Wear" and "ProcessHistory" are used for the Expert System of interest here.

The class "Inputs" has four subclasses corresponding to the four types of sensor used in the overall system. Each subclass contains information relative to feature extraction on each of the

sensors. The "Monitors" class has three subclasses that hold different tool life related information. Information related to the process history is held on the subclasses of "ProcessHistory", these contain the settings for the processing of historical data as well as lists of data built from previous predictions. The last two classes, "CutCond" and "Wear", are responsible, respectively, for holding the cutting conditions and the predicted wear value reported to the operator.

Once the objects and methods for the knowledge base were constructed, the rules were implemented to specify how objects should behave and reason using the objects. Each rule specifies a set of conditions and a set of conclusions to be made if the conditions are true. The conclusions represent logical deductions about the knowledge base and neural network specifications with changes over time. Each rule is a relatively independent module, so the system could be built gradually, rule by rule. A *Goal* was used to halt backward chaining.

4.5.1 Blackboard System

Knowledge of the application domain is divided into modules referred to as "knowledge sources" (KSs), each of which contains code for performing a particular sub-task. KSs are independent and may communicate only by reading from or writing on the "blackboard", a globally accessible portion of working memory where evolving information about the problem in hand is stored (Figure 29). This framework allows small bodies of knowledge to be represented in the most suitable form and facilitate modification of the knowledge.

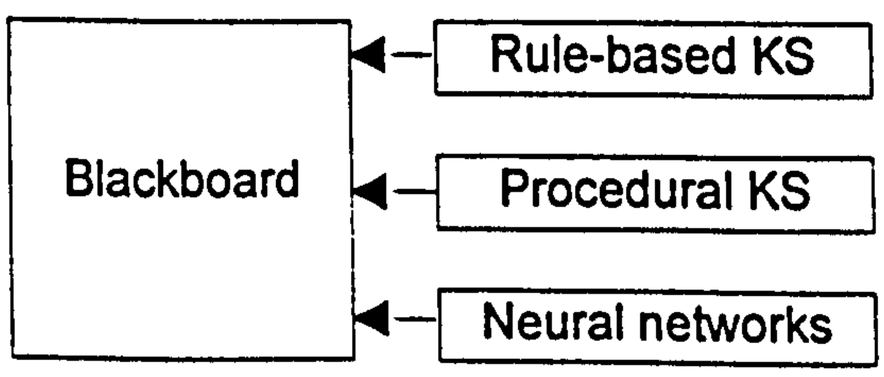


Figure 29: Modular blackboard architecture

Knowledge is made accessible to the expert system in two forms; static and dynamic. The static knowledge is related to the information provided via the user interface, such as cutting conditions, the dynamic knowledge is provided by keeping track of past classifications in order to reason upon the robustness of sequential evaluations.

4.5.2 Rule-Based Reasoning

There are two types of rule-based reasoning, forward and backward chaining, available in KAPPA-PC. Generally, forward reasoning is most appropriate when it makes sense to enter new facts and find their consequences. Given that the monitoring procedure always starts with a data acquisition stage it was decided to use forward reasoning. The selective evaluation strategy is the default for the KAPPA-PC forward chainer and this is the most efficient strategy as it follows only one successful path of reasoning, eliminating all other possible paths as it goes.

Sub-tasks required for the interpretation of tool wear on related information are shown in Figure 30. The figure reflects the way in which a solution is gradually built up on the blackboard. First data is acquired (data acquisition module) and then processed to extract the relevant features. After feature extraction, the feature vector is presented to the neural networks which give their predictions based solely on sensory information. Tool life knowledge based on Taylor's equation predictions is then applied in order to eliminate most classifications that fall outside a confidence band around the empirical formulae prediction. Both neural networks are tested for tool wear evolution coherence and finally combined to give the final tool wear level prediction. The most relevant functions are presented in Appendix D.

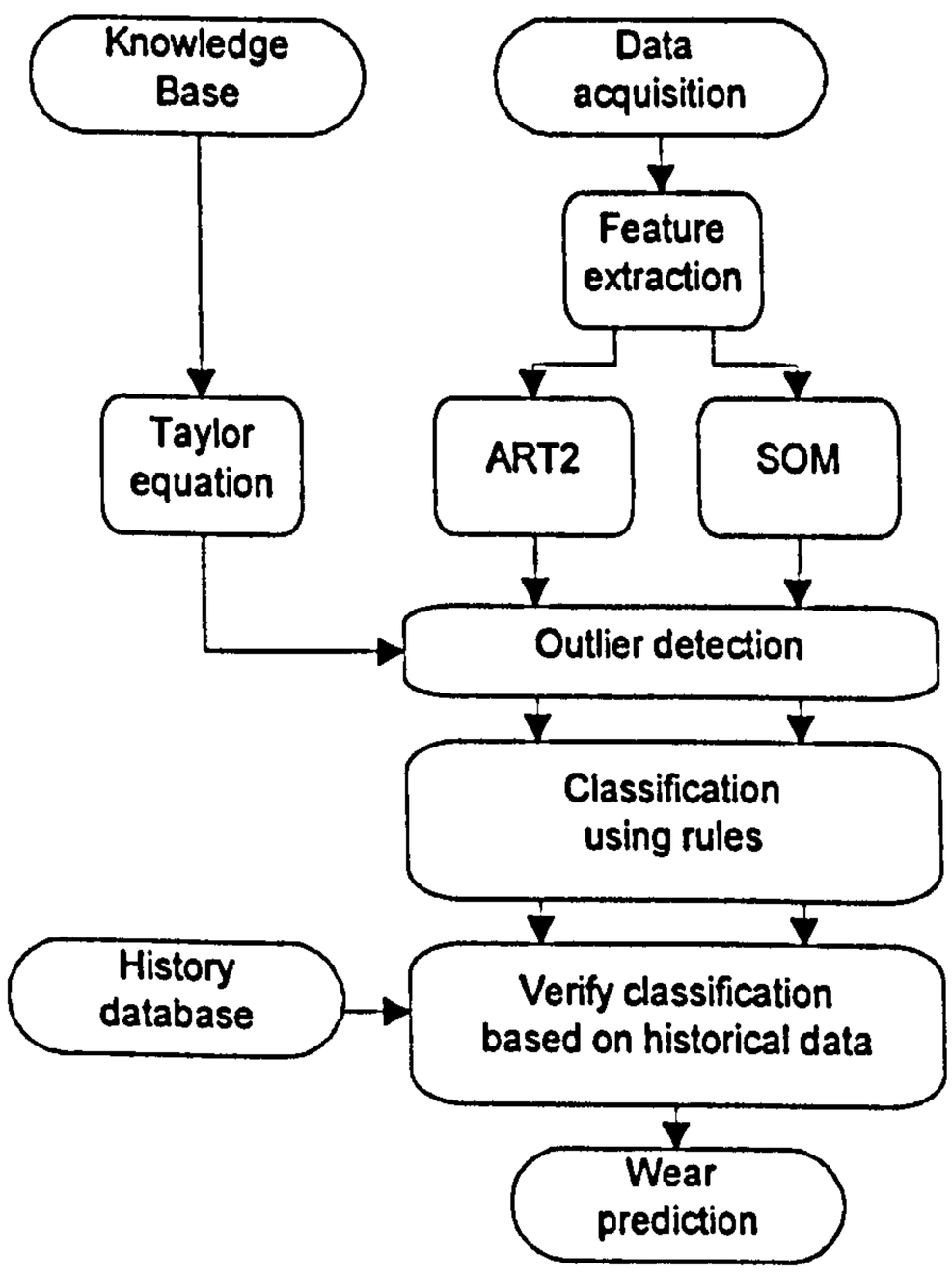


Figure 30: Stages in tool wear estimation

4.5.3 User Interface

The system's main interface is shown in Figure 31. Several system parameters can be set by the user namely; cutting conditions for weight reference (creates links to previously trained data), and sampling interval for real-time monitoring. Links to sub-interfaces are made via labelled buttons and these are; the monitoring interface, the neural network training interface, and data file selection. The tool state can be viewed on the monitoring interface.

Tool and material selection is done via the user interface (Figure 31) and this sets the values of Taylor's tool life equation pre-defined in the tool/material database. The monitoring screen (Figure 32) is responsible for the display of the tool wear prediction. From the monitoring screen it is also possible to adjust the threshold values for tool wear criteria as well as setting the number of samples for historical reasoning. At the moment the system is triggered manually although it is possible to trigger it via spindle current threshold since this is a good indicator of when the machine is cutting.

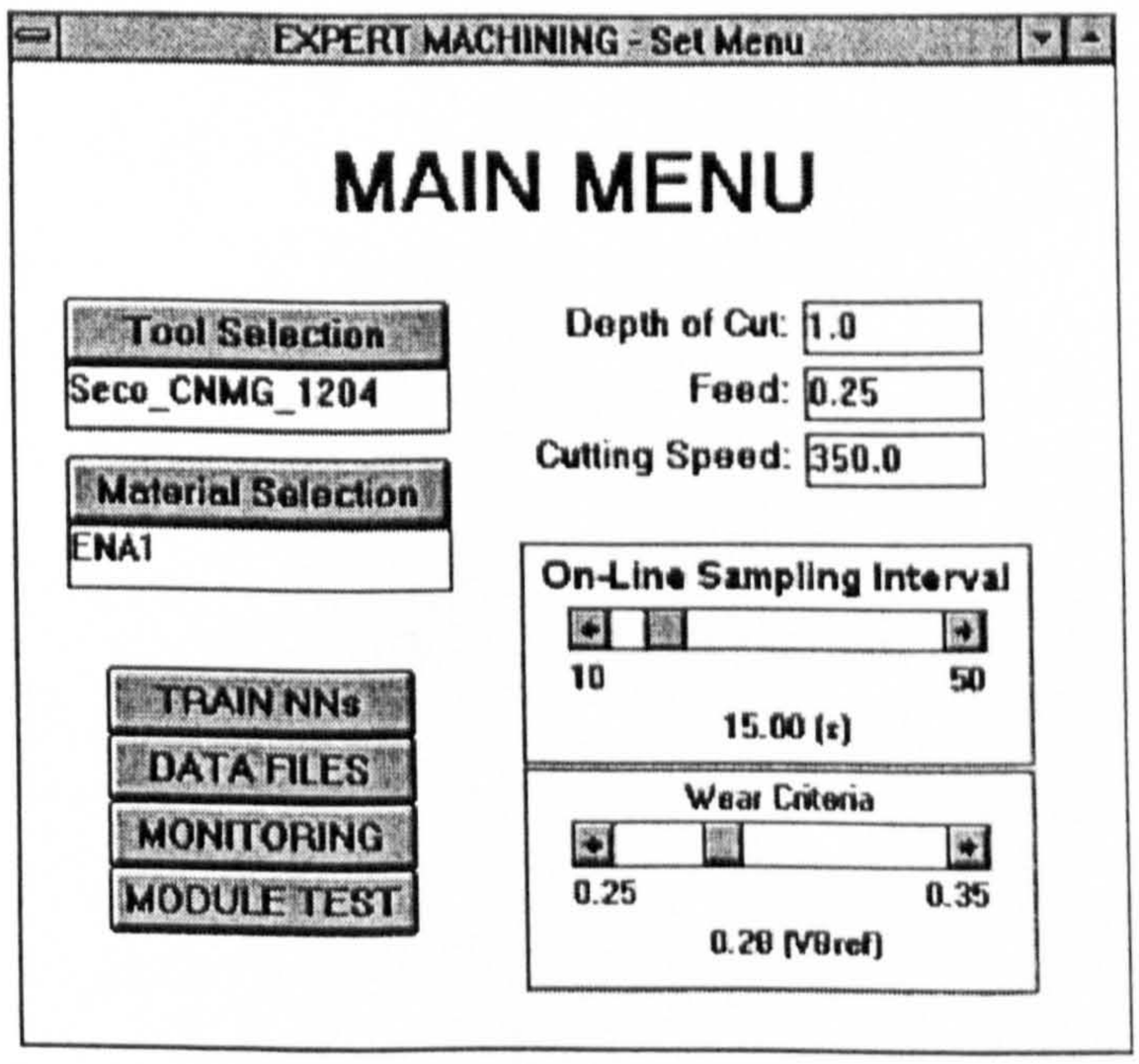


Figure 31: Main Menu Screen

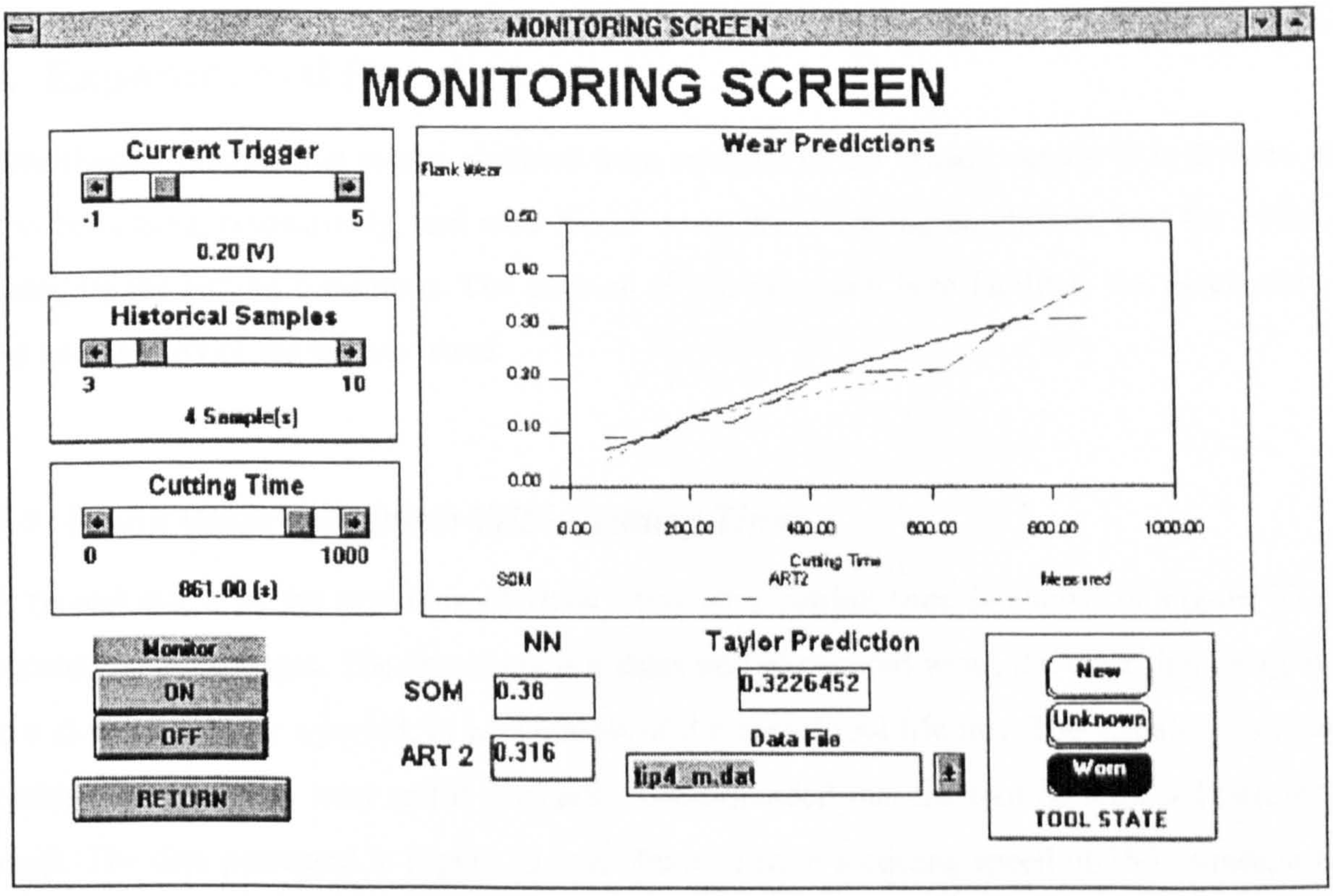


Figure 32: Monitoring Menu

5. Experimental Results

This chapter presents the results obtained from analysis of the sensor signals. It is divided into two describing, respectively, tool wear based on constant cutting conditions, and the effect of changing the cutting conditions. The purpose of this approach is to facilitate the assessment of the adaptability of the system based.

5.1 Flank Wear Evolution With Cutting Time

A typical graph of the evolution of flank wear with cutting time is shown in Figure 33 and consists of three stages. The first stage is a short period of rapid wear, the wear then progresses at a slower rate over a period, in which most of the useful tool life lies. The last stage is a rapid period of accelerated wear and it is usually recommended that the tool be replaced before this stage. The data presented in Figure 33 was obtained from a cutting speed of 350 m/min, a feed rate of 0.25 mm/rev, and a 1 mm depth of cut. For these cutting conditions the first stage can be observed to end at approximately 3.5 min after the start of cutting, which corresponds to VB_B = 0.09 mm, the second stage lies in the interval between 3.5 min and 14.7 min ($0.09 < VB_B < 0.3$ mm), and the third stage starts after 14.8 min of cutting time. The beginning of the third stage coincides with a value of flank wear of 0.3 mm which is the tool life criterion established in the ISO3685 (1993) standard.

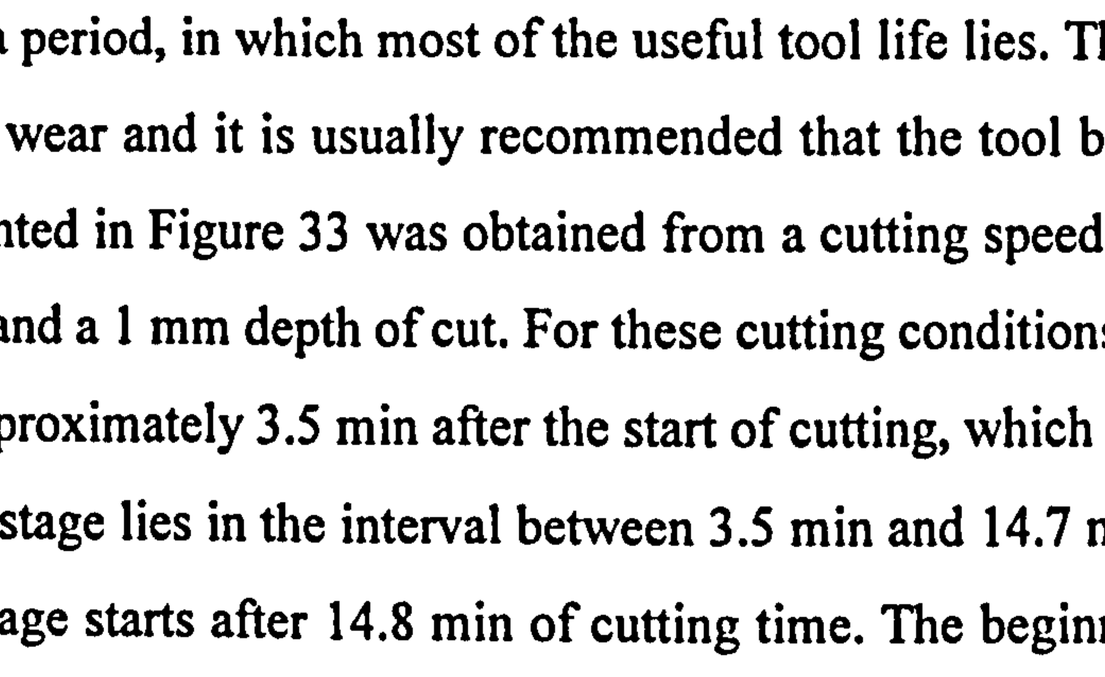


Figure 33: Flank wear evolution with time: 350 m/min, 0.25 mm/rev and 1 mm depth of cut

5.2 Results Obtained From Machining With Constant Cutting Conditions

The following results were obtained with fixed cutting conditions to investigate the influence of flank wear on the evolution of sensor outputs independently of the cutting conditions and therefore assess the suitability of each sensor for tool wear classification. The cutting

conditions were those shown in Table 11, chosen so as to lie in the upper range of severity recommended for the particular tool/workpiece combination.

Table 11: Tests at fixed cutting conditions

Depth of Cut (mm)	1.0
Feed Rate (mm/rev)	0.25
Cutting Speed (m/min)	350

5.2.1 Sensor/Feature Analysis

The following sections present the results obtained from each of the sensors used. The features for which it is easy to ascertain a correlation to the tool wear evolution are; the frequency spectrum of sound and vibration; and the feed and tangential forces. More detail is given in the following sections regarding the evolution of these features with tool wear.

5.2.1.1 Sound Emission and Machine Vibration

An examination of the power spectrum of both sound and vibration reveals some features centred around the frequencies of 2.3 and 4.5 kHz that vary with tool wear (Figure 34 and Figure 35) with the feature at 2.3 kHz being the most visible. These figures are typical over the different tools examined.

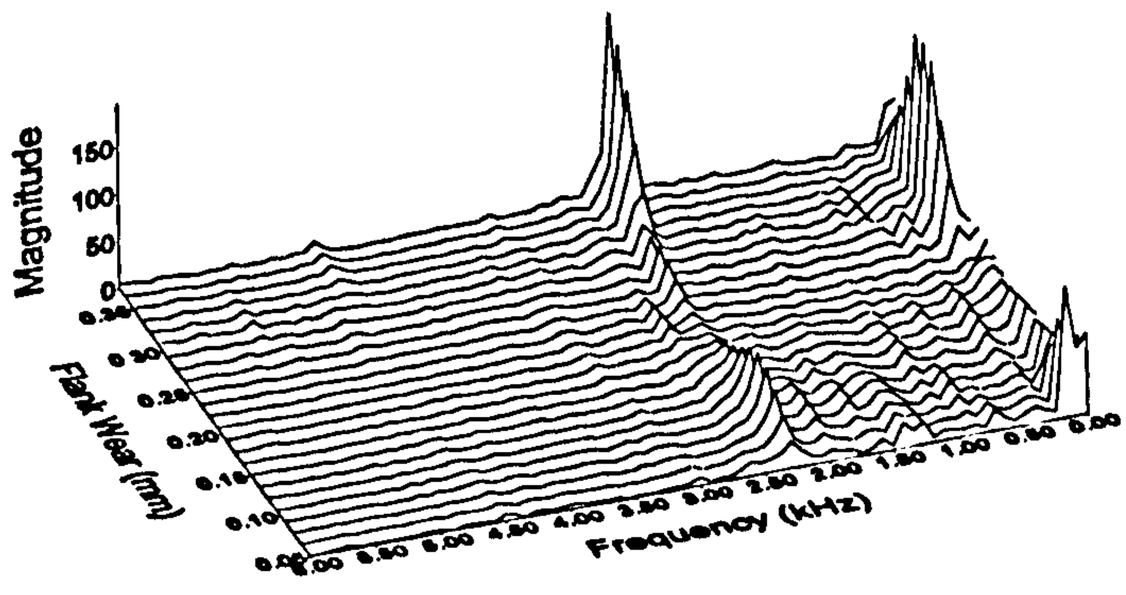


Figure 34: Sound spectrum versus flank wear

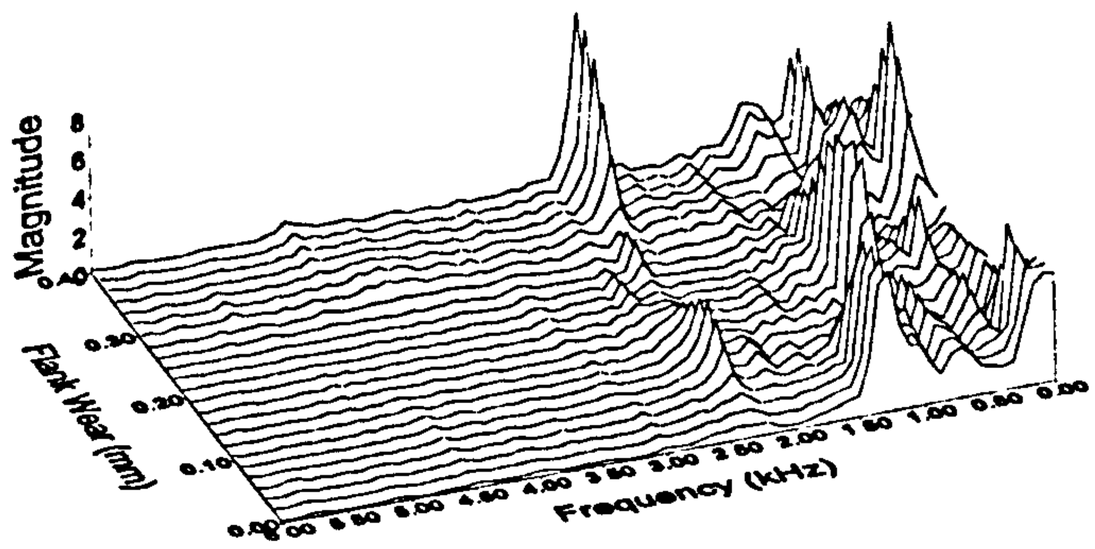


Figure 35: Vibration spectrum versus flank wear

Typically, for flank wear levels up to 0.1 mm the frequency peak rises in amplitude, then up to 0.25 mm the signal drops in amplitude before rising sharply for flank wear levels greater than 0.25 mm.

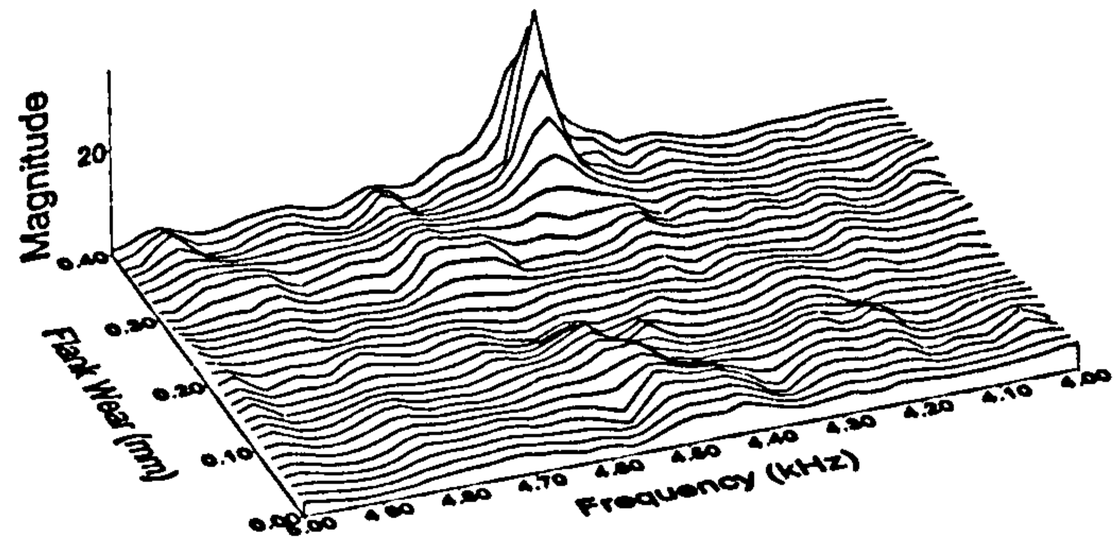


Figure 36: Detail of Figure 34

A statistical feature analysis for sound and vibration of the raw time series failed to exhibit any clear correlation with flank wear (Figure 37 to Figure 44).

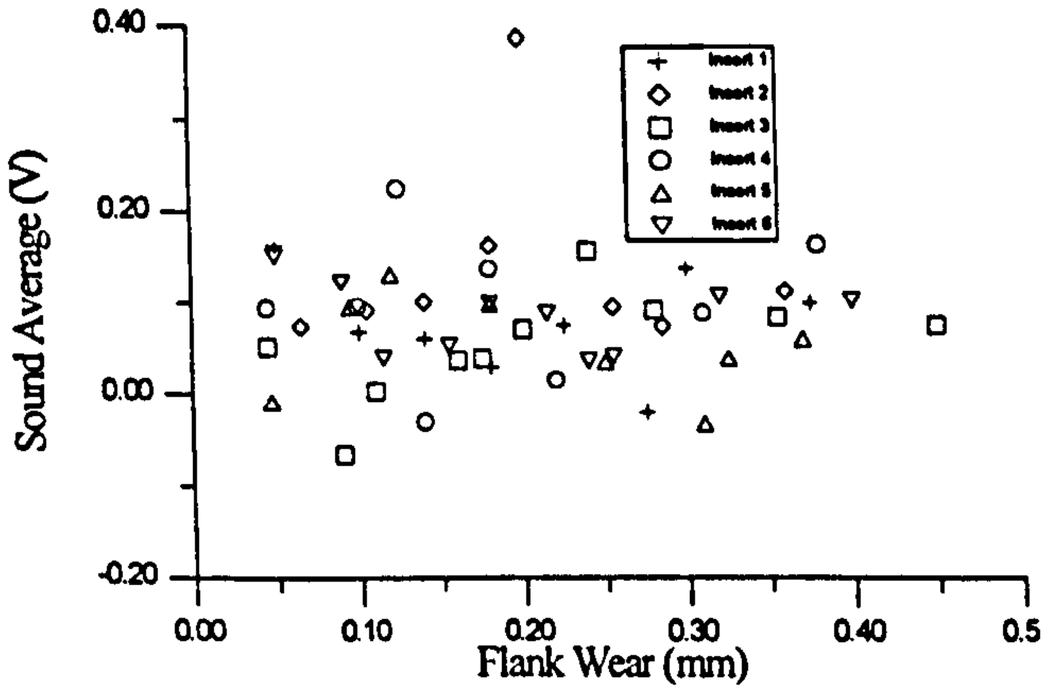


Figure 37: Sound average versus flank wear

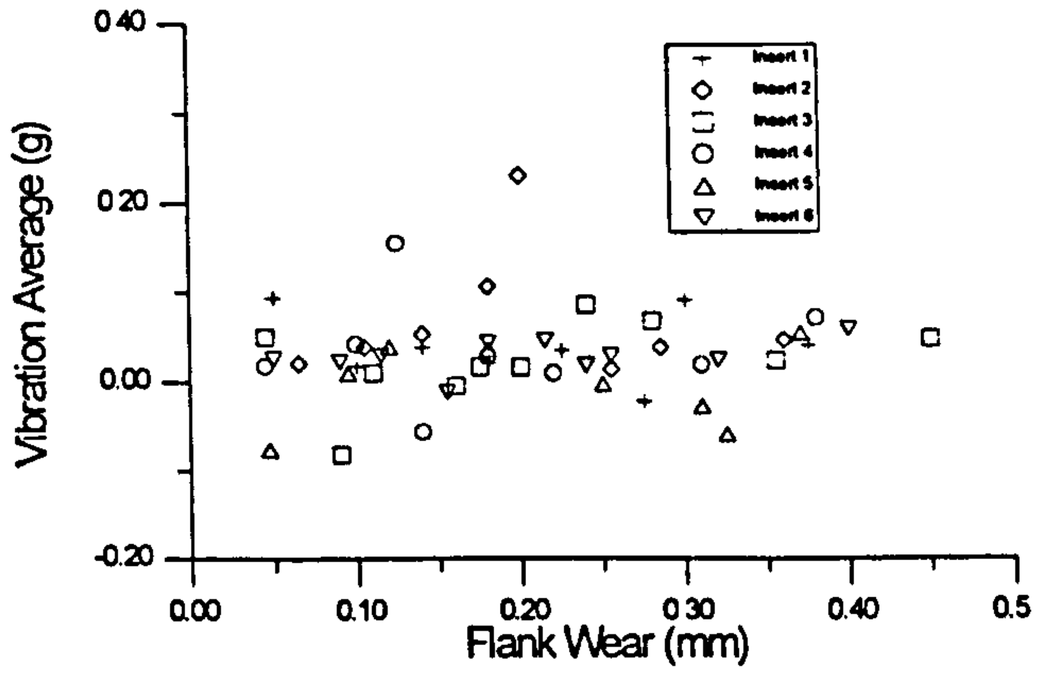


Figure 38: Vibration average versus flank wear

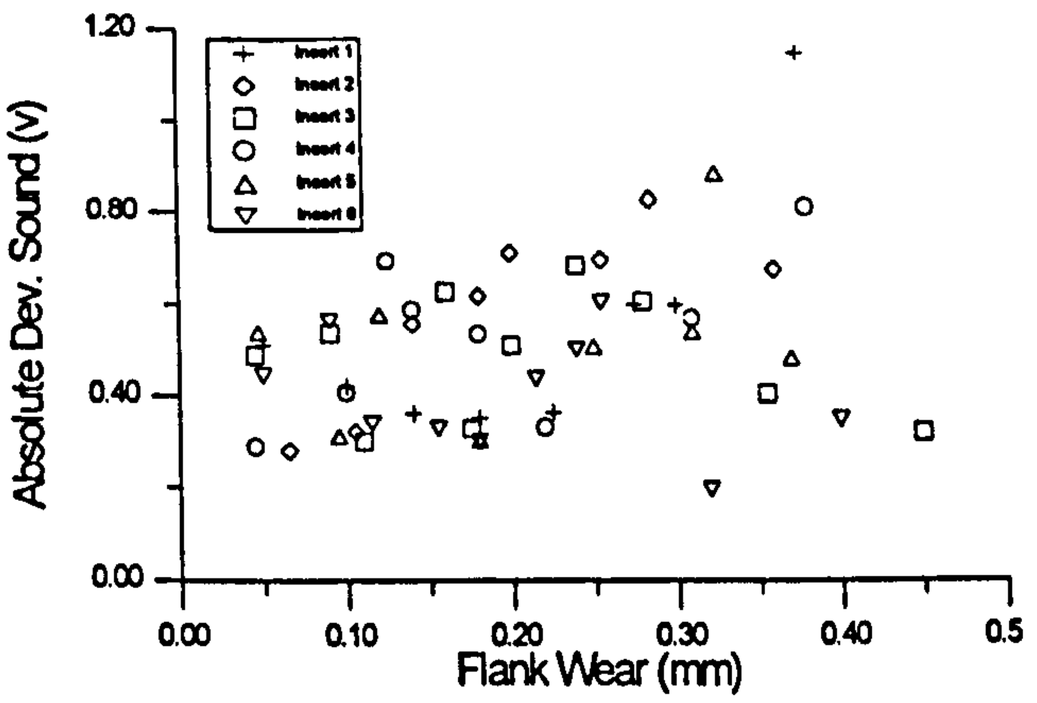


Figure 39: Sound absolute deviation versus flank wear

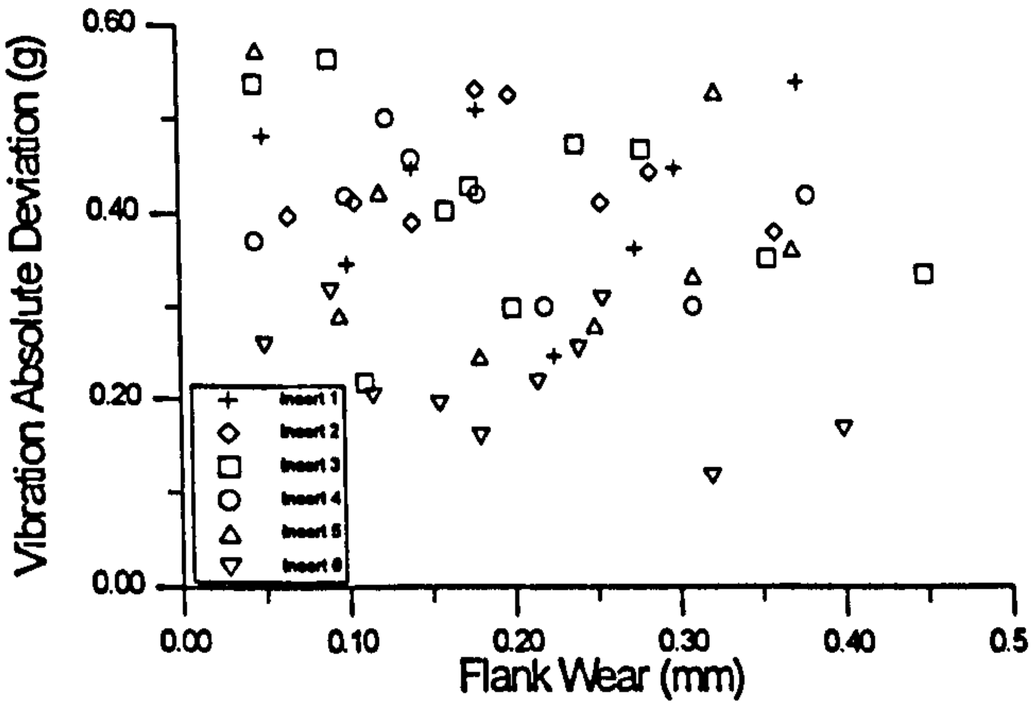


Figure 40: Vibration absolute deviation

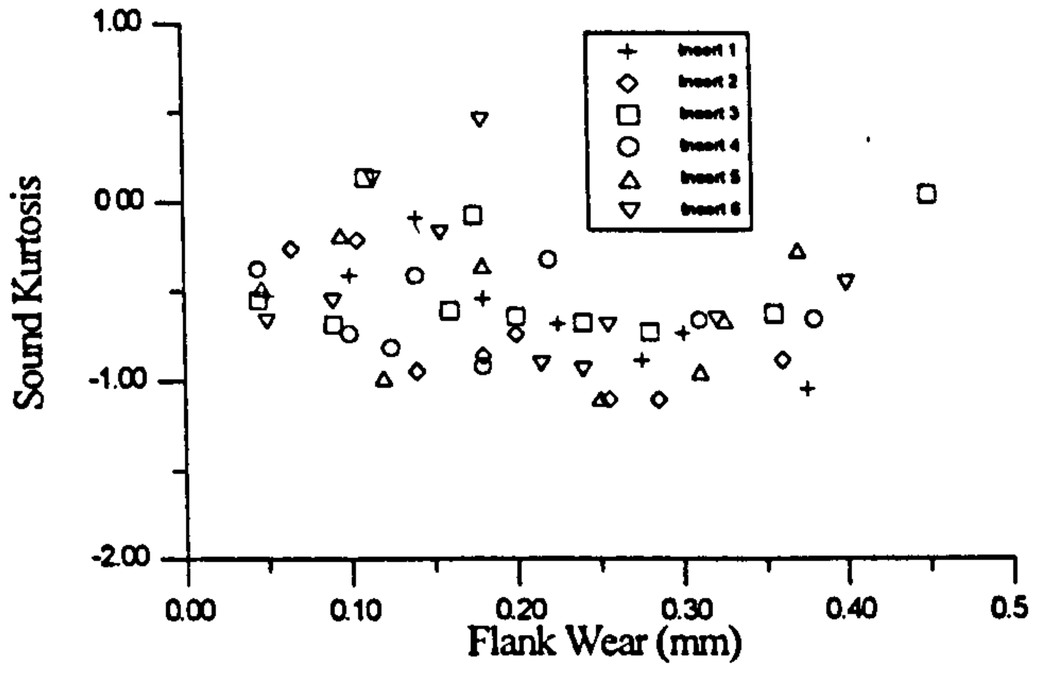


Figure 41: Sound kurtosis versus flank wear

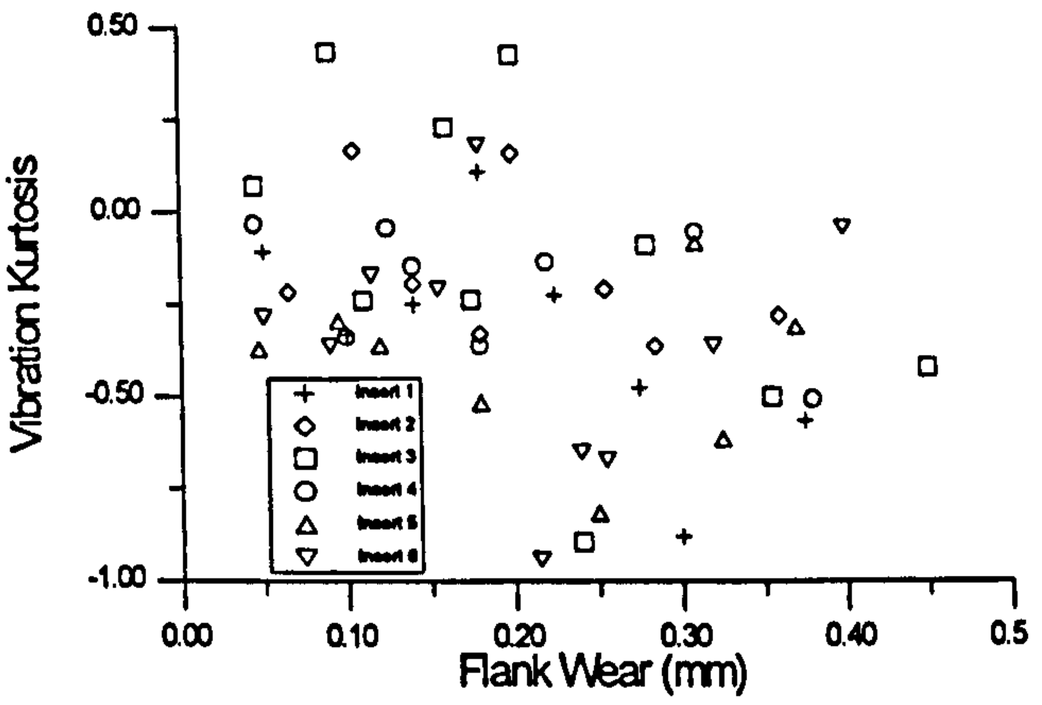


Figure 42: Vibration kurtosis versus flank wear

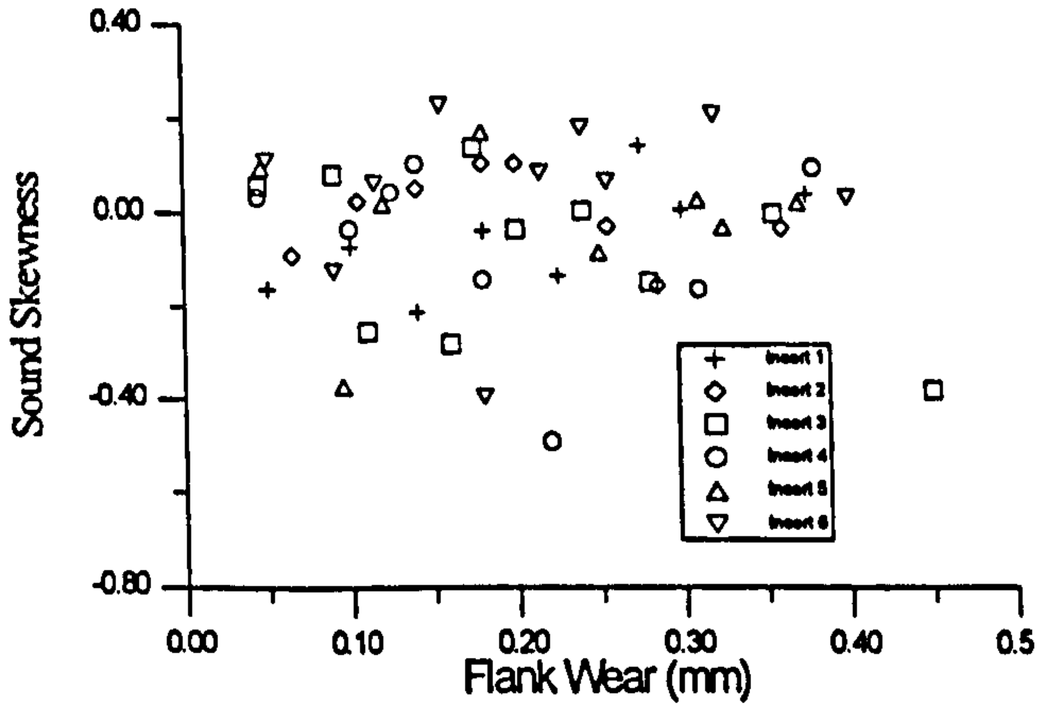


Figure 43: Sound skewness versus flank wear

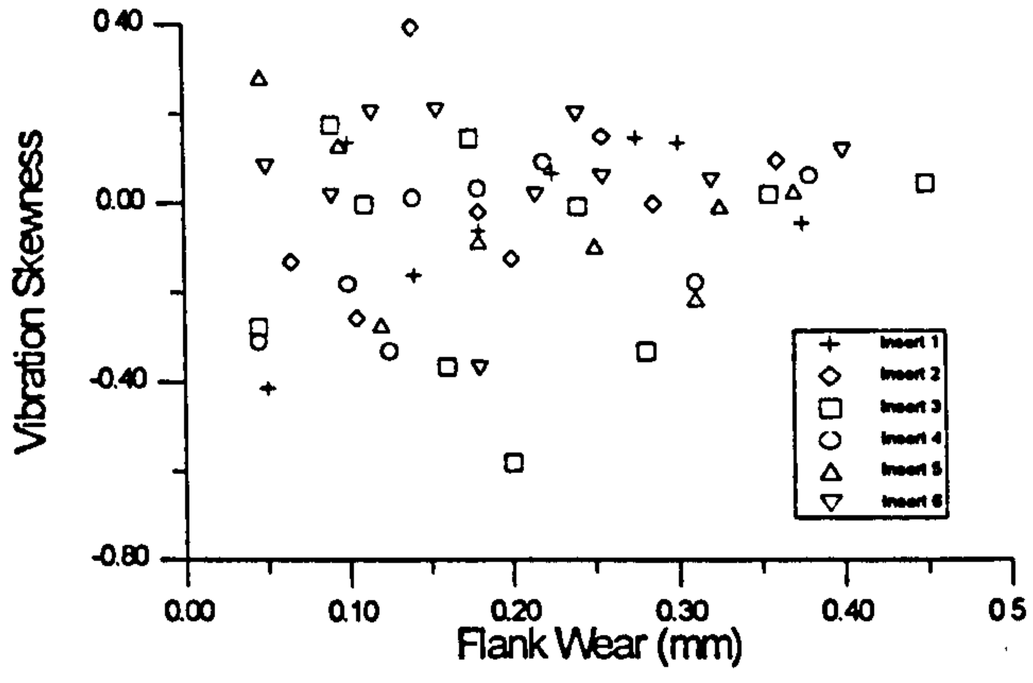


Figure 44: Vibration skewness versus flank wear

5.2.1.2 Tangential and Feed Forces

Under fixed cutting conditions the tangential and feed components of the cutting force showed a good linear correlation with increasing flank wear (Figure 45 and Figure 46) with the feed force increasing by 30% ($F_f \cong 300.VB_B + 300$, $r_f = 0.604$) compared to 21% ($F_t \cong 300.VB_B + 500$, $r_t = 0.673$) for the tangential force. If individual evolutions are compared the tangential forces ($r > 0.85$) shows better correlation coefficients than the feed forces ($r > 0.65$). Each point on Figure 45 and Figure 46 is the average of 512 samples taken from the force signal along with the other sensor signals.

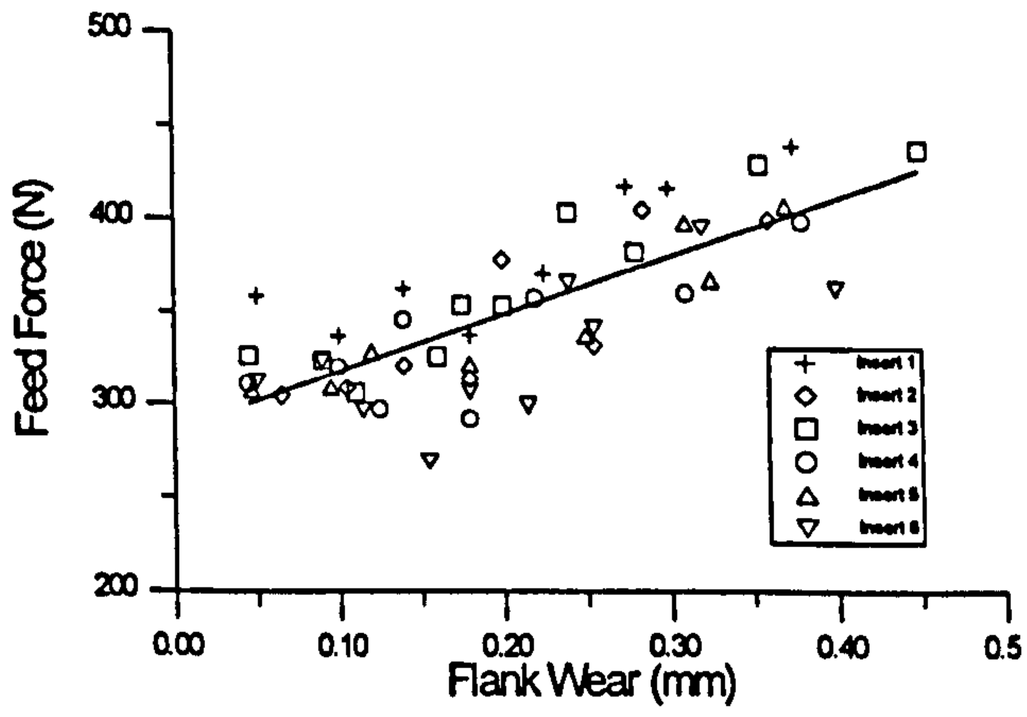


Figure 45: Feed force versus flank wear

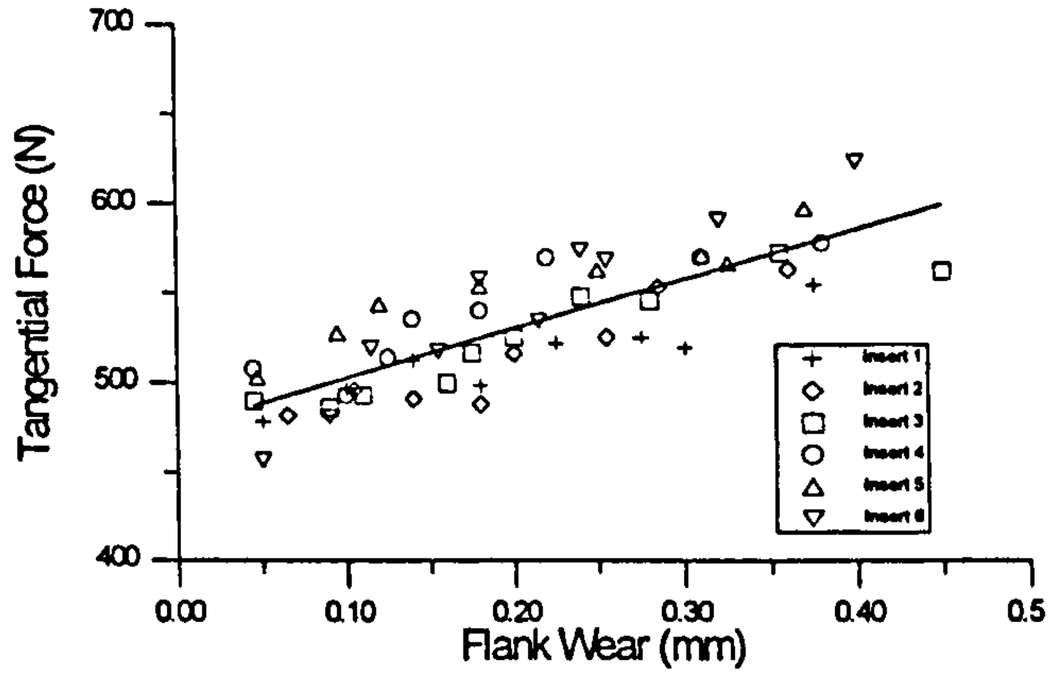


Figure 46: Tangential force versus flank wear

Frequency analysis of the force components reveals some important characteristics of the dynamic behaviour of the tool shank when cutting, these were not used as features and are only presented here due to their interesting behaviour. The feed component of the force shows (Figure 47) two frequencies that appear consistently, 3.4 and 6.6 kHz and these correspond to the modes of vibration of the tool shank in the feed direction under forced vibration.

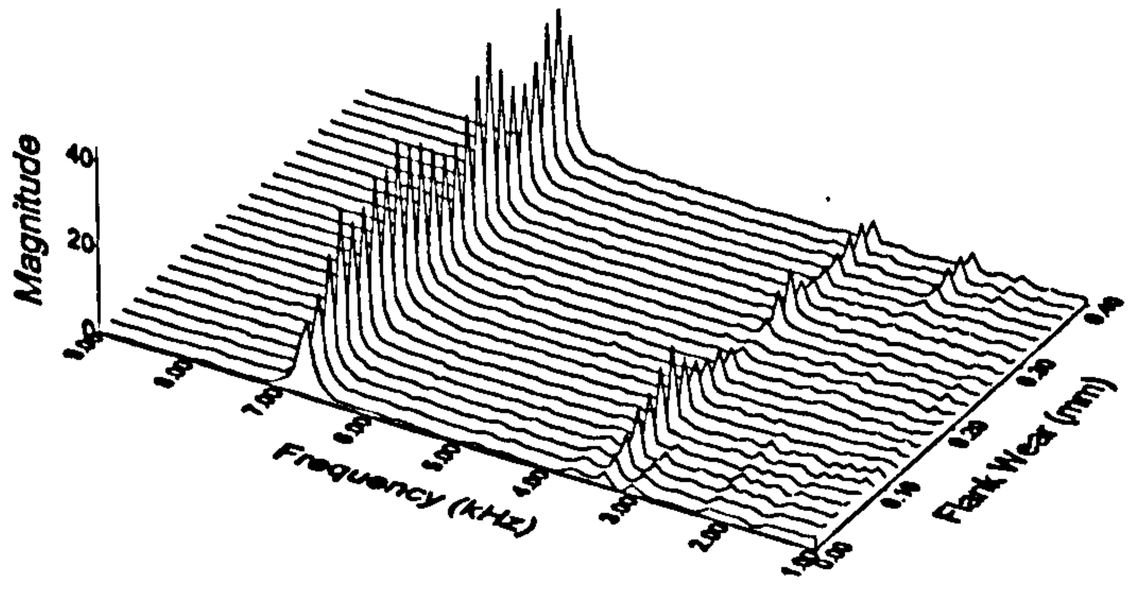


Figure 47: Feed force spectrum versus flank wear

In Figure 48 a typical evolution of the frequency spectrum for the tangential force is shown for increasing tool wear. The two frequency components that stand out from the spectra and which seem to have a repeatable behaviour are those due to forced vibration of the system tool/workpiece, at 2.3 and 6.6 kHz. Pattern repeatability of the power spectrum in the feed direction is weaker than in the tangential direction, both Figure 47 and Figure 48 are representative of the behaviour for all 6 inserts, these are the results for Tool 1.

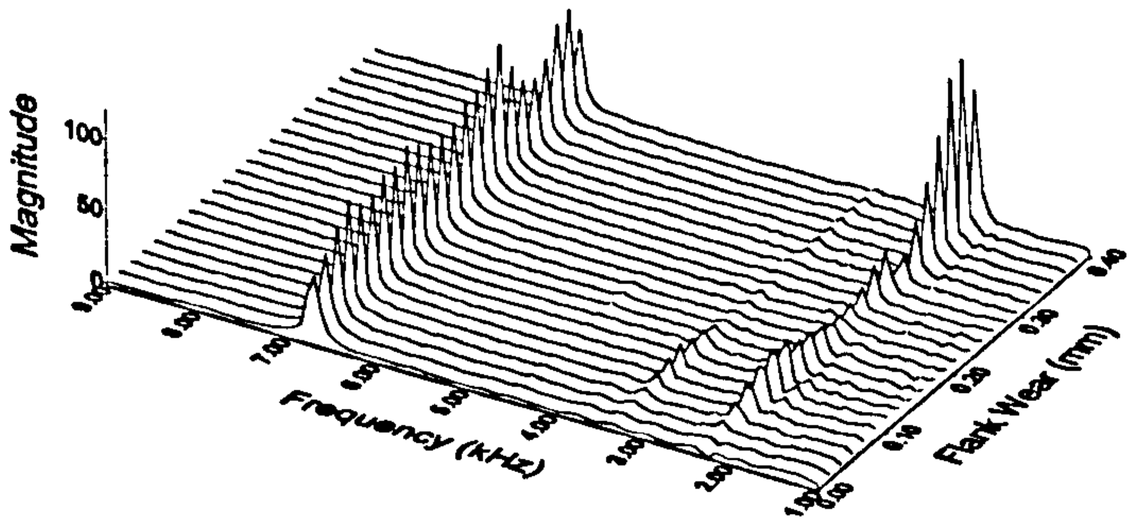


Figure 48: Tangential force spectrum versus flank wear

Analysis carried with different data sets varying feed, depth of cut and cutting speed did not affect the position of previously identified frequencies.

5.2.1.3 Spindle Current

Figure 49 shows the variation in spindle current with tool wear. As can be seen there is little variation although there are some interesting points which will be raised in the discussion. What seems to happen is an increase in spindle current with reduced workpiece diameter, that corresponds to an increased RPM.

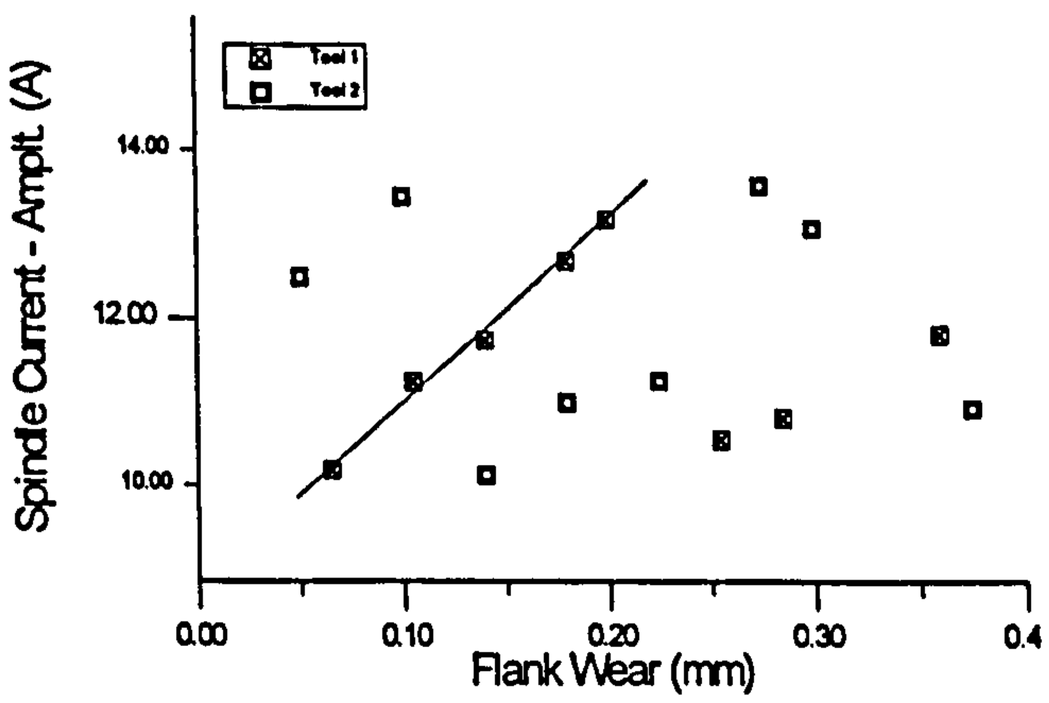


Figure 49: Spindle current versus flank wear

5.3 Test Results With Variable Cutting Conditions

In order to assess the adaptability of the monitoring system a systematic set of experiments were conducted where the cutting conditions were varied, both around a region close to the training set of conditions and widely across the range of applicable conditions for the tool/workpiece combination.

Results obtained from machining with variable cutting conditions provide the basis for future research regarding system adaptability. This is regarded as very important for industrial applications given the slight changes in cutting conditions required when cutting different workpiece profiles and with different surface finish. The changes in system performance have been analysed for these different cutting conditions. Since tool wear monitoring depends mainly on sensor information, despite the "last word" being given by the Expert System, an analysis of the influence of sensor features will be given in this chapter. The neural networks' generalisation capabilities will be dealt with in Chapter 6.

The results presented next are based on samples taken at three different wear levels for each set of cutting conditions; new, $VB_B \approx 0.15$ mm, and $VB_B \approx 0.3$ mm. At each wear level two consecutive samples were acquired and saved for processing. The following graphs were built from the features extracted for consecutive samples, both values are presented as error bands. The frequencies examined here are the ones identified in section 5.2.1.1 to behave in a typical way with wear, 2.3±0.1 and 4.5±0.1 kHz.

5.3.1 Variation in the Depth of Cut

Changing the depth of cut has an obvious effect on the cutting forces (Figure 50 and Figure 51), which increase linearly with the depth of cut for the tested range. The maximum feed force, corresponding to a worn tool, increased by 12% for each 0.1 mm increase in depth of cut ($F_f \cong 400.d + 200$) and the tangential force by 7% ($F_t \cong 400.d$). The difference between a new and a half-worn tool, for both force components, are small and vary slightly with depth of cut.

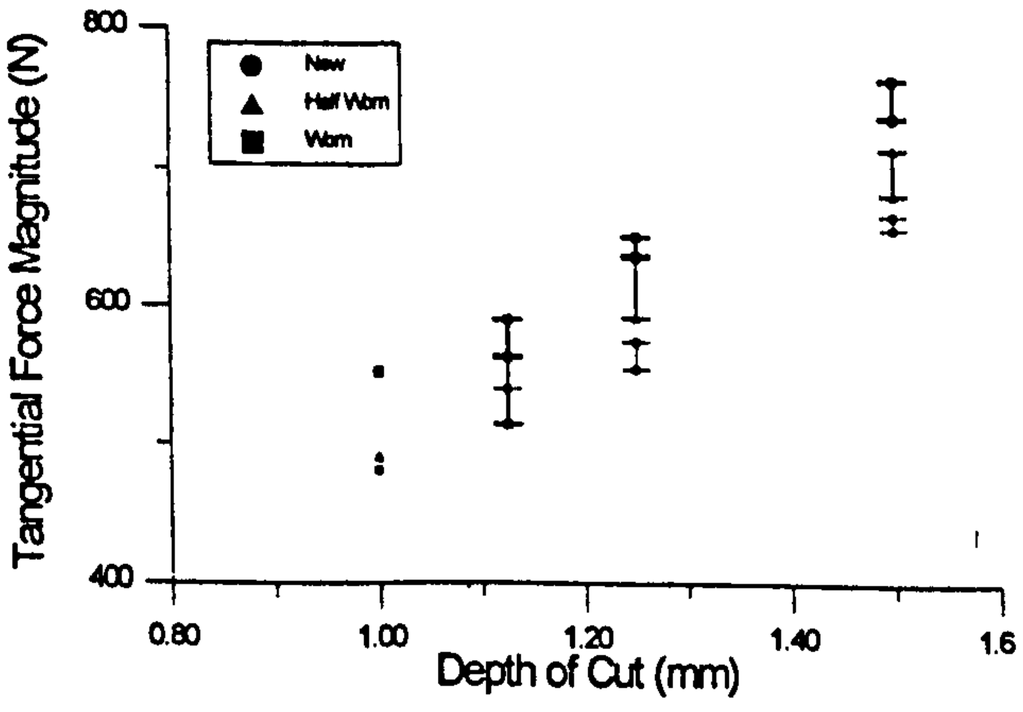


Figure 50: Tangential force versus depth of cut

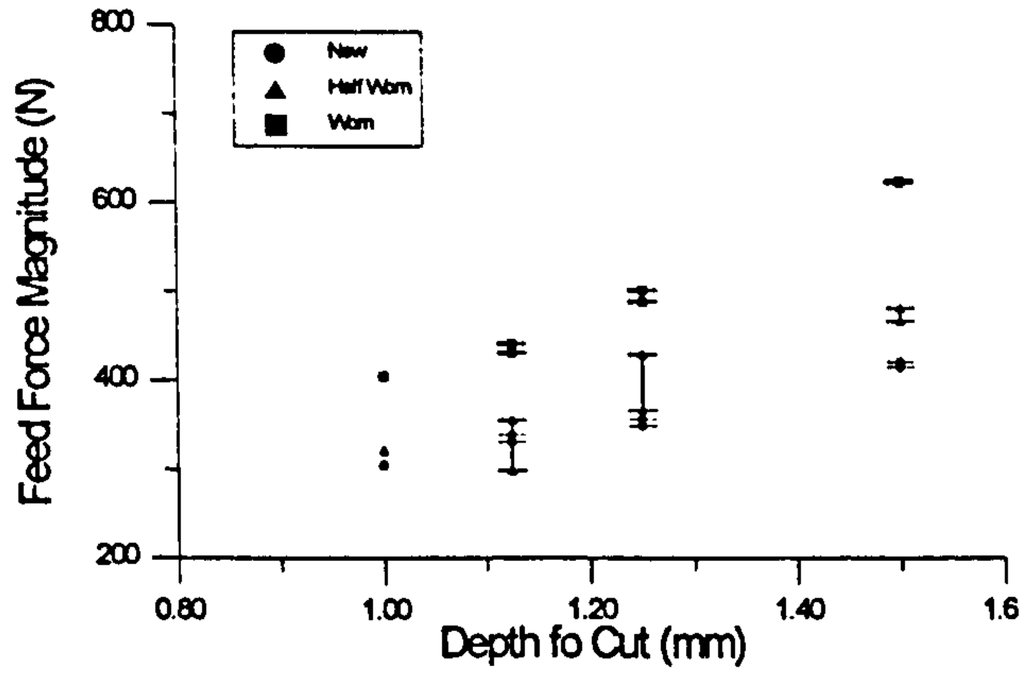


Figure 51: Feed force versus depth of cut

The tangential force maintained the same ratio of increase for a worn tool as well as new, for the feed force a slight increase occurred between a new and worn tool.

An increase in depth of cut reduced the magnitude of the absolute deviation of both sound and vibration (Figure 52 and Figure 53). In addition, as the depth of cut increased so the order of new/half-worn/worn changed so that a new tool went from the smallest amplitude to the largest, for sound as an example.

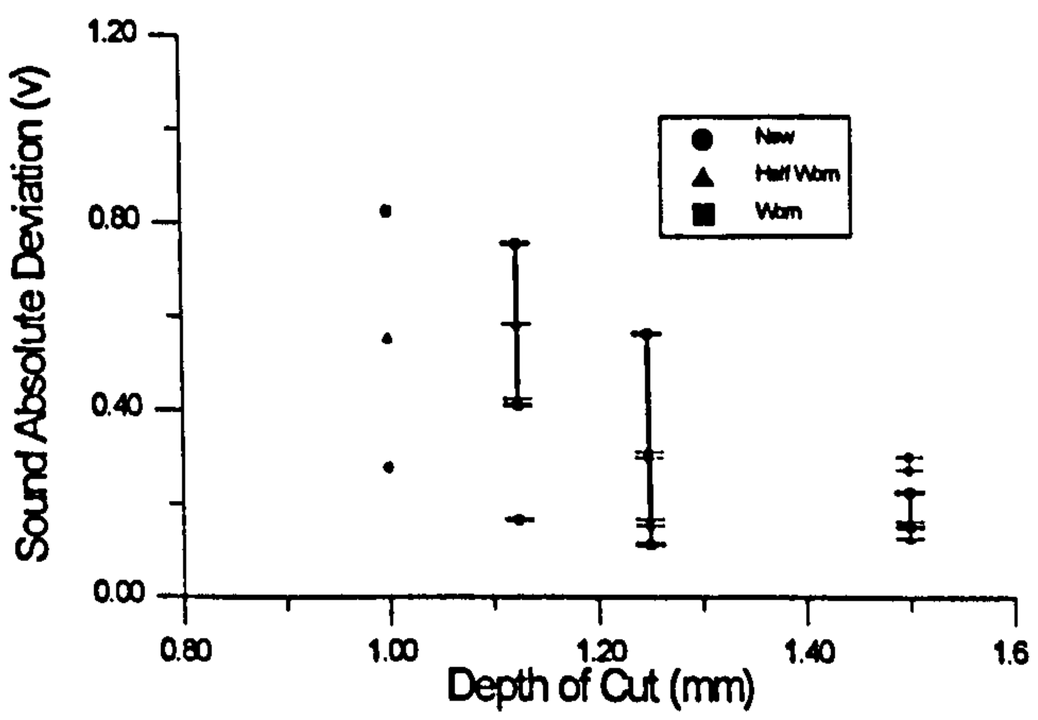


Figure 52: Sound absolute deviation versus depth of cut

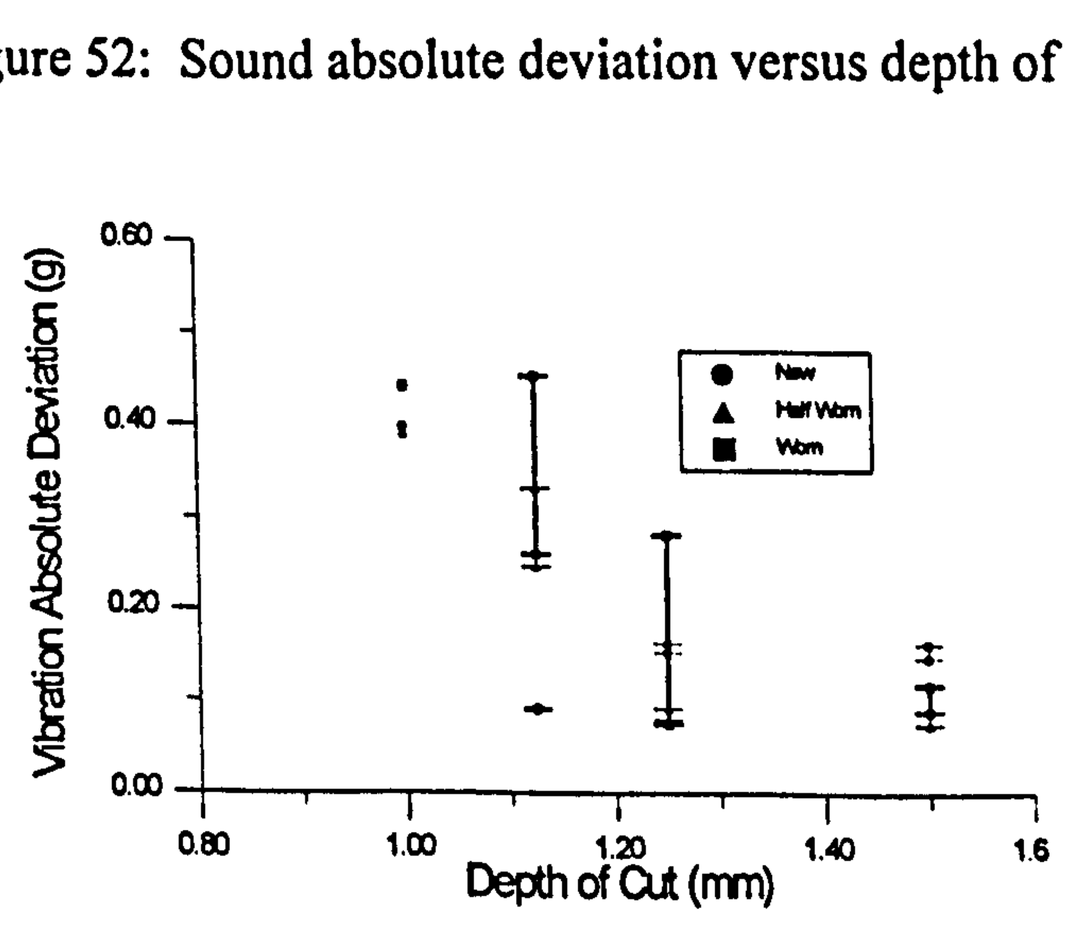


Figure 53: Vibration absolute deviation versus depth of cut

The frequency spectrum of both sound and vibration (Figure 54 to Figure 56) suffers a reduction in amplitude as a result of increased depth of cut.

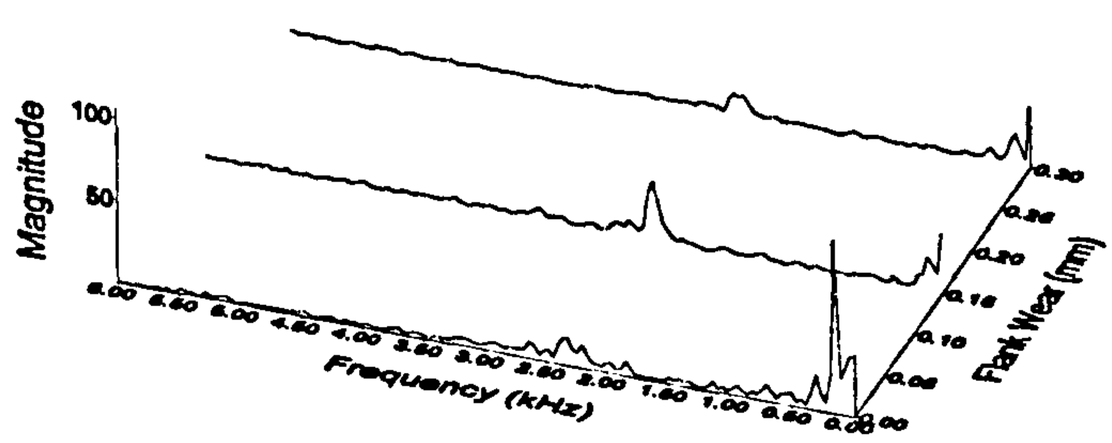


Figure 54: Sound spectrum versus flank wear (depth of cut 1.25 mm)

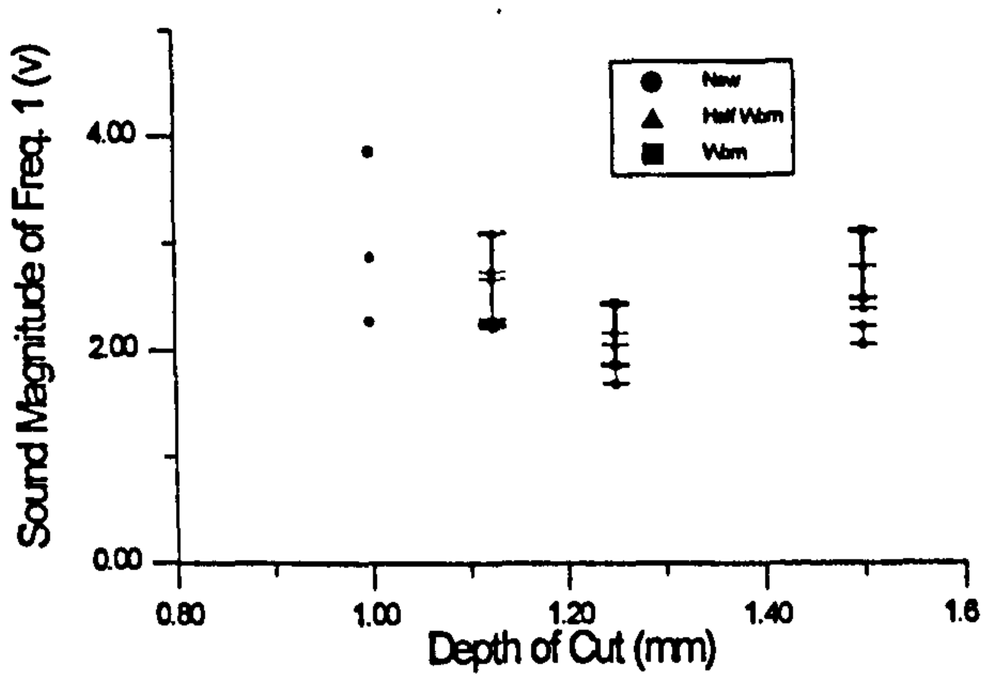


Figure 55: Sound magnitude at frequency band 2.3±0.1 kHz versus depth of cut

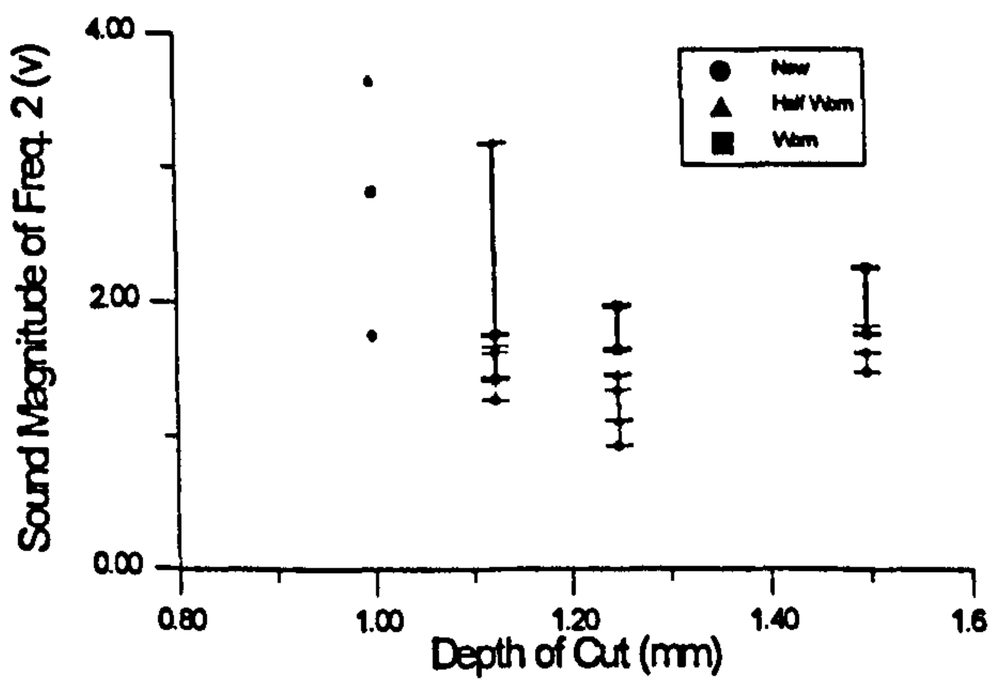


Figure 56: Sound magnitude at frequency band 4.5±0.1 kHz versus depth of cut

Frequency band 2.3±0.1 kHz on the sound spectrum (Figure 55) appears to be more sensitive and easily related to the dynamic behaviour of the machine/tool. The magnitude of this frequency band appears to reduce up to a depth of cut of 1.25 mm, thereafter increasing slightly.

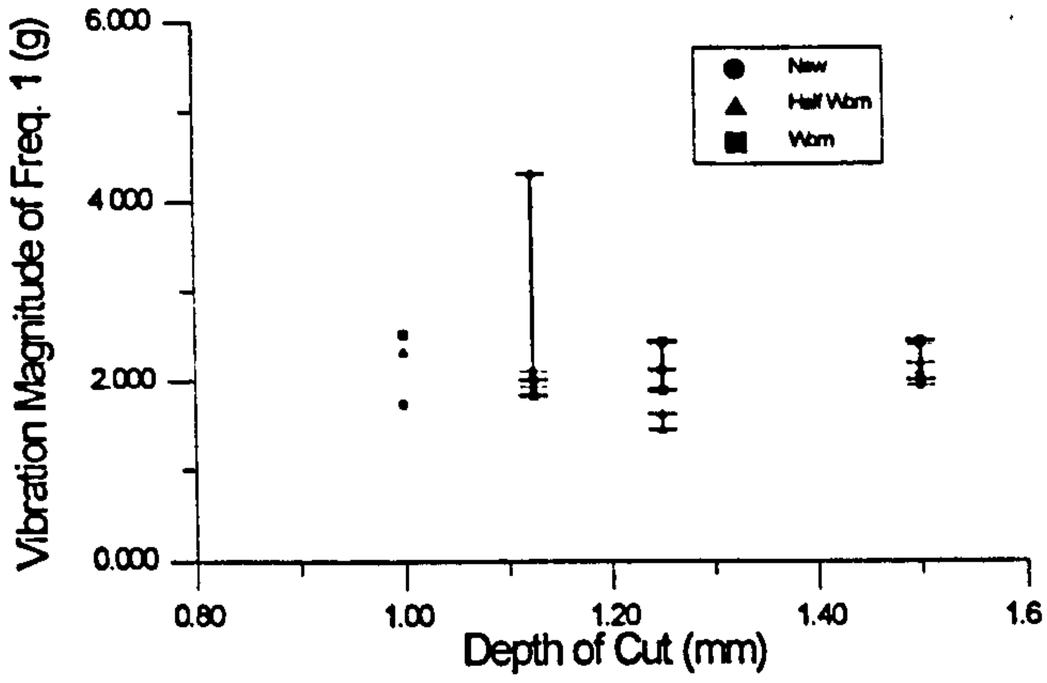


Figure 57: Vibration magnitude at frequency band 2.3±0.1 kHz versus depth of cut

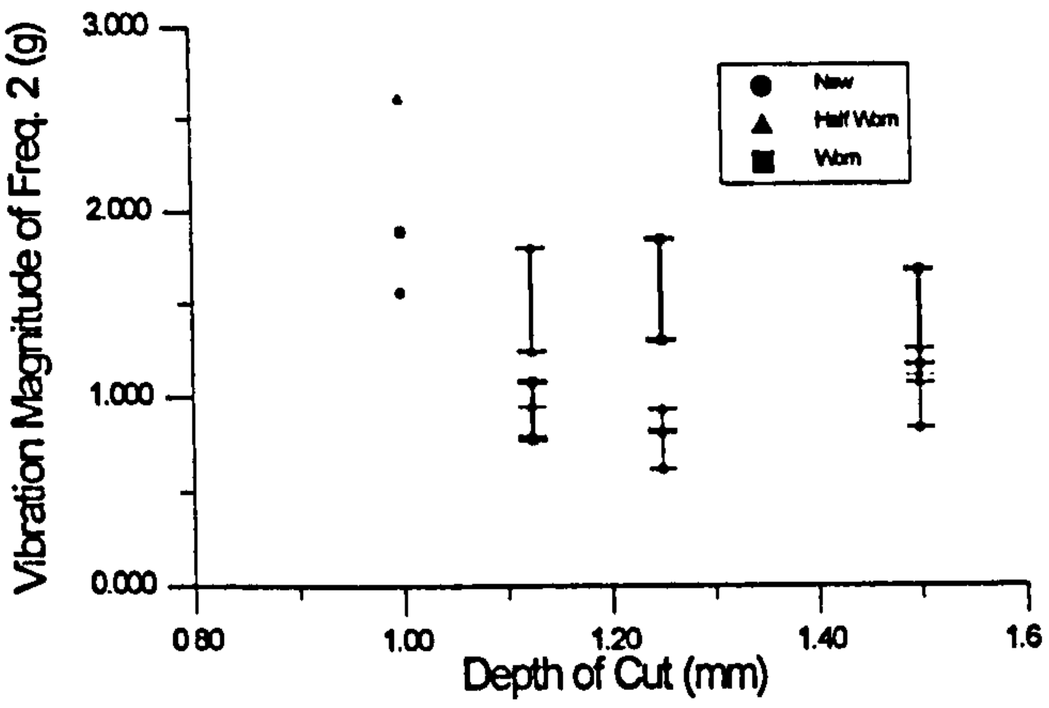


Figure 58: Vibration magnitude at frequency band 4.5±0.1 kHz versus depth of cut

The changes in magnitude of frequency band 2.3±0.1 kHz of the vibration spectrum seem to be small, only one half-worn point seems to stand out from the rest, this being due to random impacts which caused the error to increase. The magnitude of frequency band 4.5±0.1 kHz shows a larger spread of data points.

5.3.2 Variation in the Feed Rate

As feed rate increases so does the metal removal rate, the contact area between the tool and workpiece increases and therefore the forces required to remove the workpiece material would also be expected to increase. Figure 59 and Figure 60 illustrate this trend with the increase in feed ($F_f \cong 2000.f$) and tangential ($F_t \cong 2000.f$) force following an almost linear pattern, increasing by 5% for each 0.01 mm/rev increase in the feed rate. The forces corresponding to new and half worn stages increase approximately with the same rate up to 0.25 mm/rev and then seem to decrease, giving an overall final percentile increase between new and worn tools of 64% and 23% respectively for the feed and tangential forces. The difference between a new

and half-worn tool remains constant for both feed and tangential forces. It appears that as feed rate increases so do the forces separation between new and worn tools.

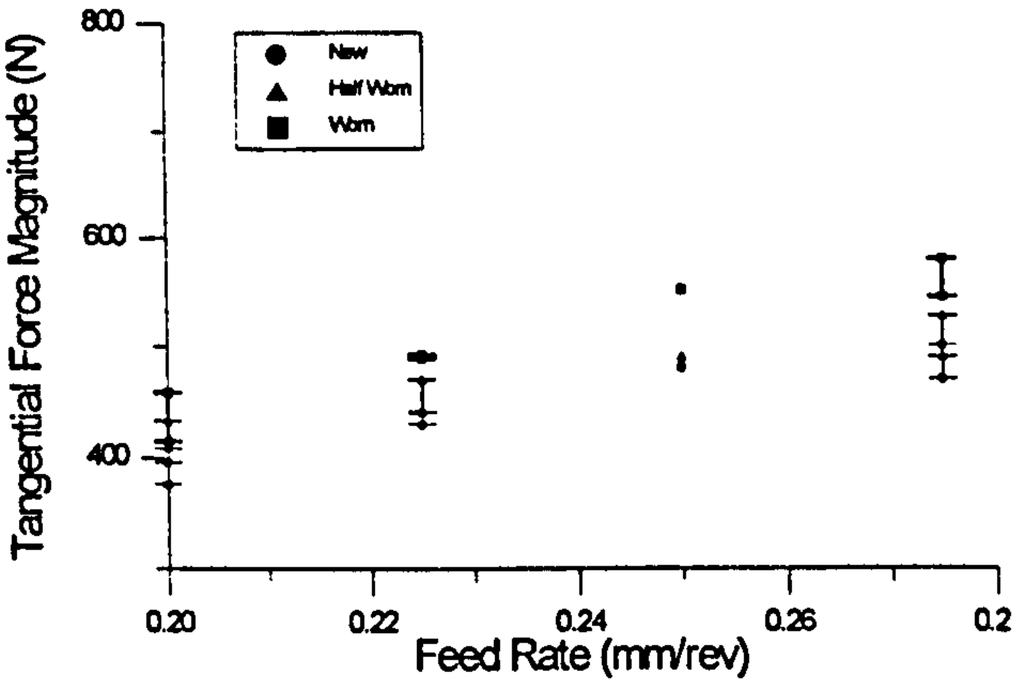


Figure 59: Tangential force versus feed rate

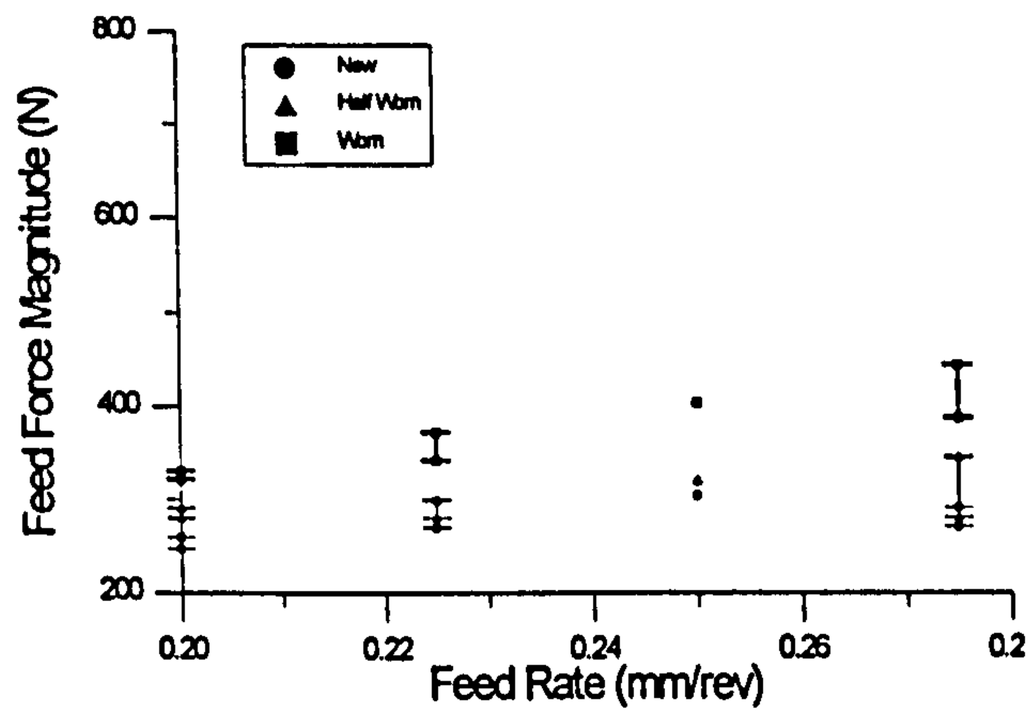


Figure 60: Feed force versus feed rate

For variations in the absolute deviation of both sound and vibration with feed rate there does not appear to be any consistent variation (Figure 61 and Figure 62). Both graphs show scattered data points which seem to be affected in different ways by the feed rate, these figures also show that there are larger variations with wear than feed rate for both sound and vibration.

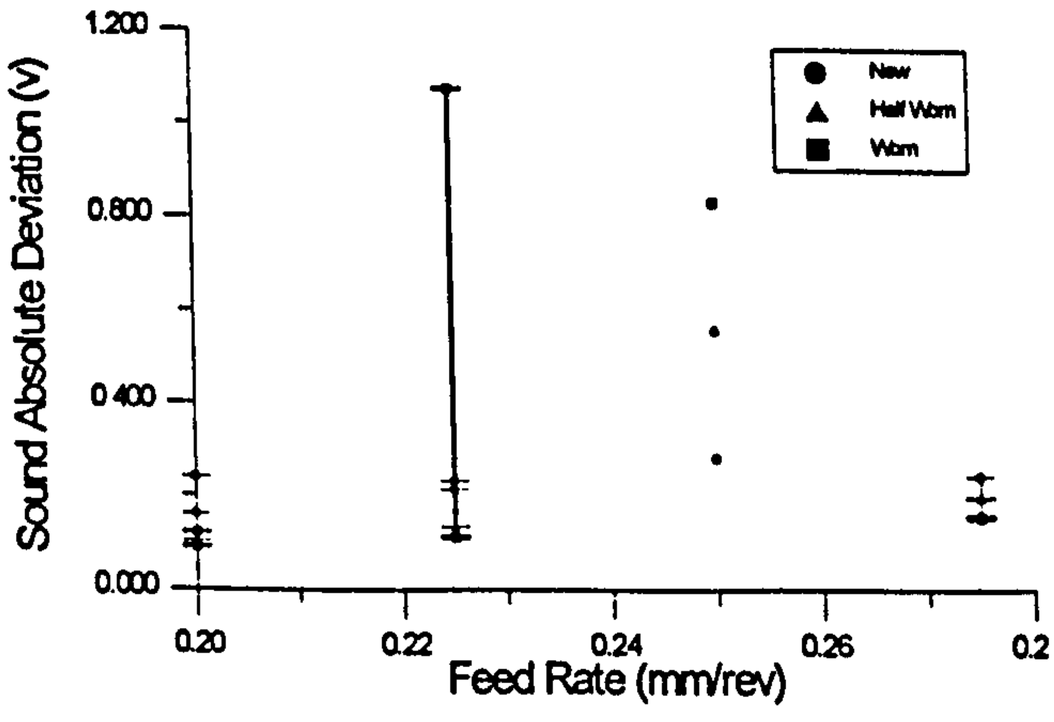


Figure 61: Sound absolute deviation versus feed rate

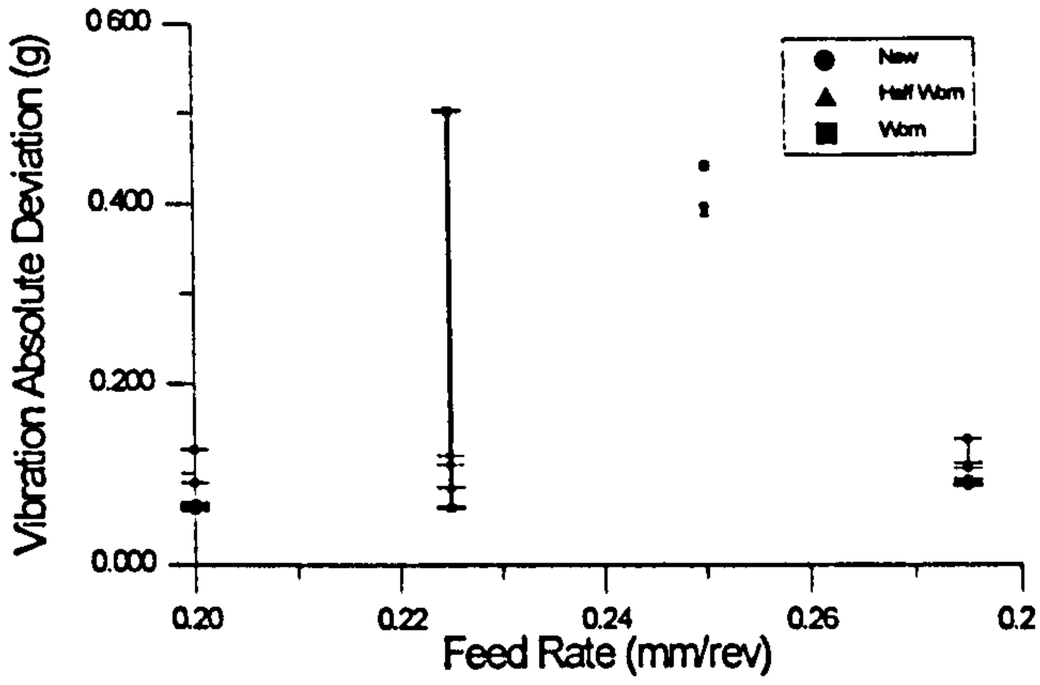


Figure 62: Vibration absolute deviation versus feed rate

The amplitude of the frequency spectrum for both sound and vibration over both frequency bands increased up to a value of 0.25 mm/rev feed rate, and from there on decreased (Figure 64 and Figure 65). An inversion on the amplitude scale of the frequency for both sound and vibration can be observed at the initial tested feed rate (0.2 mm/rev) and at 0.28 mm/rev. In fact, the last point on these graphs seem different for all signal processing.

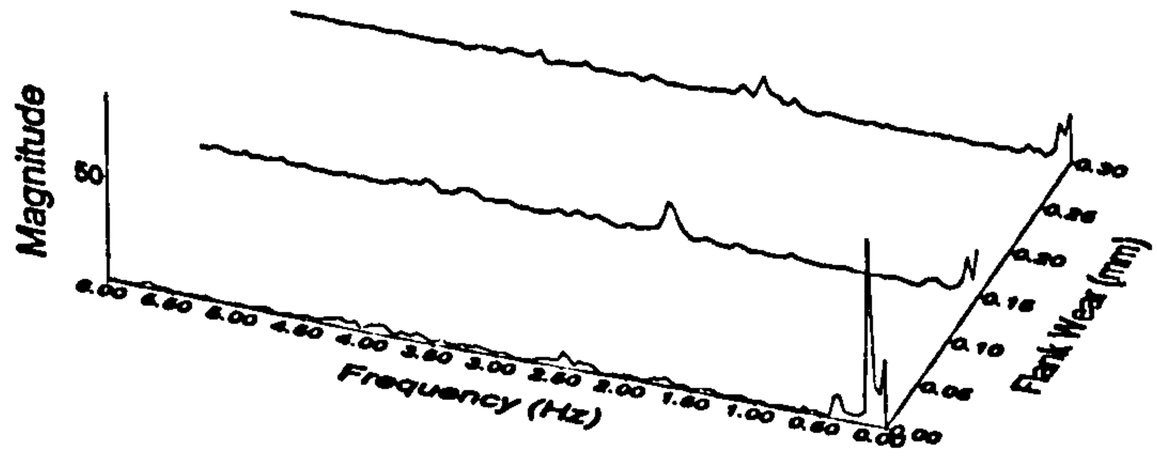


Figure 63: Sound spectrum versus flank wear (feed rate 0.2 mm/rev)

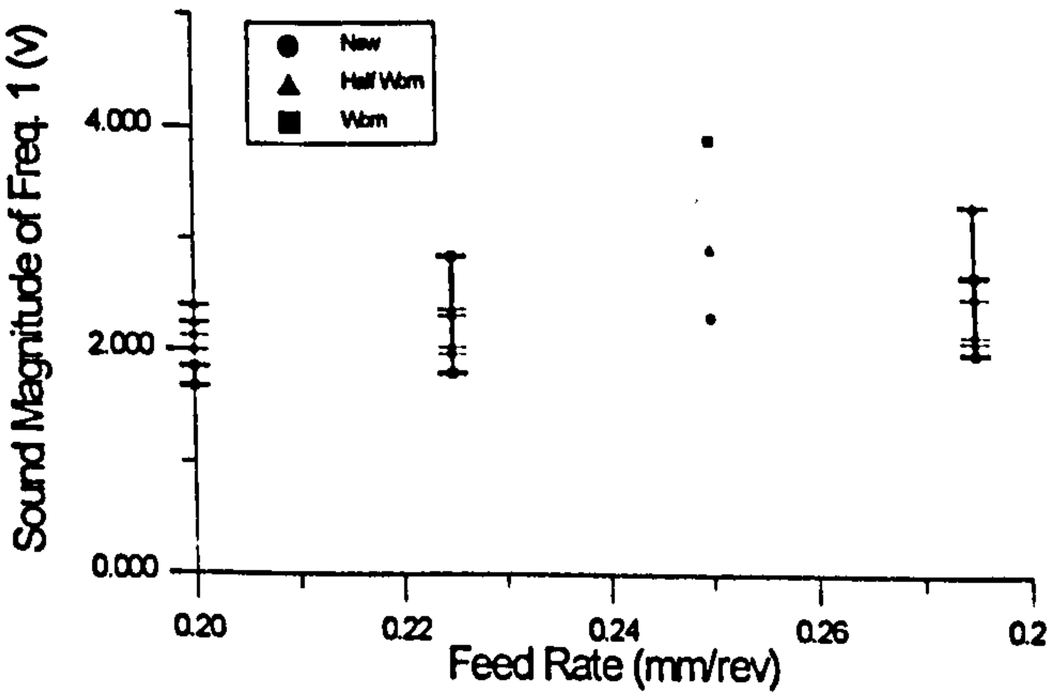


Figure 64: Sound amplitude at frequency band 2.3±0.1 kHz versus feed rate

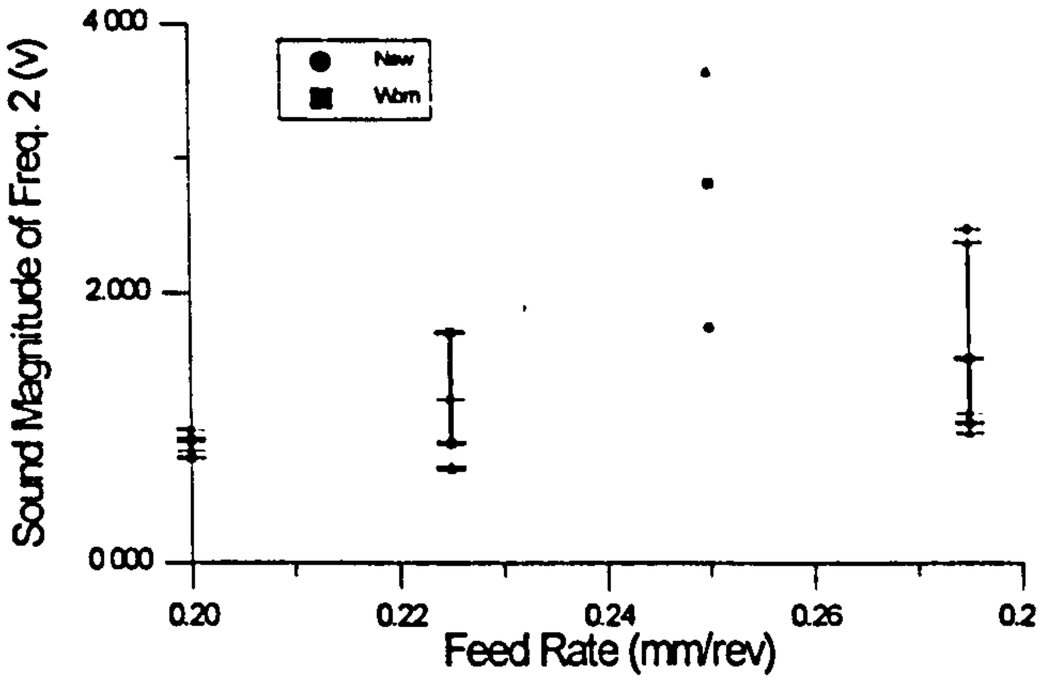


Figure 65: Sound magnitude at frequency band 4.5±0.1 kHz versus feed rate

The sound frequency band 2.3±0.1 kHz (Figure 64) shows a slightly more consistent evolution then band 4.5±0.1 kHz (Figure 65), a maximum on the frequency magnitude corresponds to a maximum value of wear. Similar trends are exhibited by both sound (Figure 64 and Figure 65) and vibration (Figure 66 and Figure 67) spectra.

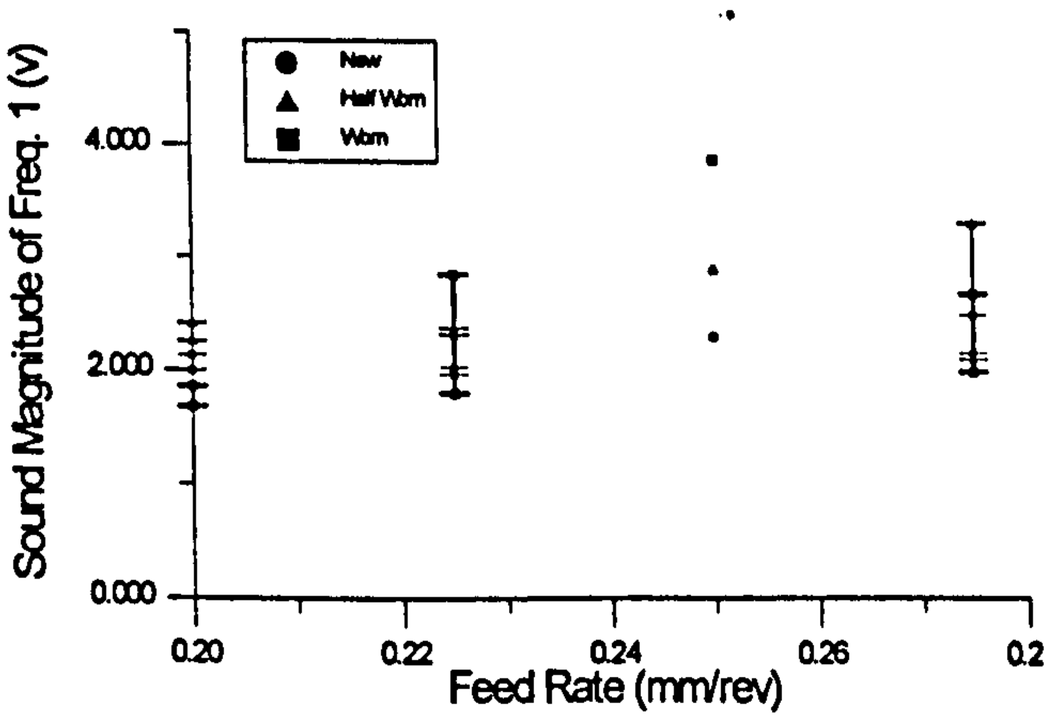


Figure 66: Vibration magnitude at frequency band 2.3±0.1 kHz versus feed rate

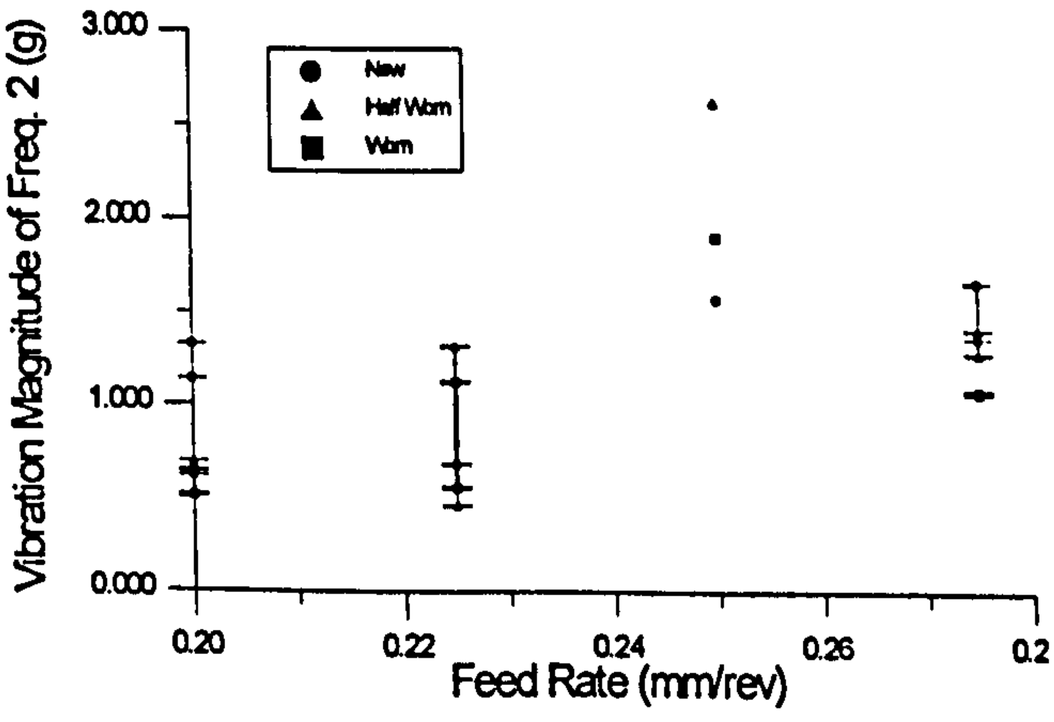


Figure 67: Vibration magnitude at frequency band 4.5±0.1 versus feed rate

5.3.3 Variation in the Cutting Speed

The cutting forces, feed and tangential, exhibit a very similar evolution with changes in cutting speed (Figure 68 and Figure 69). For worn tools, both forces seem to decrease up to a speed of 344 m/min and increase thereafter. The tangential force has an initial step increase between new and worn tool, falling afterwards, and then increasing again - the feed force as an identical progression. The relative behaviour of both force components is not seen as substantial compared to the effect of feed rate and depth of cut. Also, it can be seen that the relative increase and decrease in both forces does not follow a smooth evolution as that found for feed rate and depth of cut.

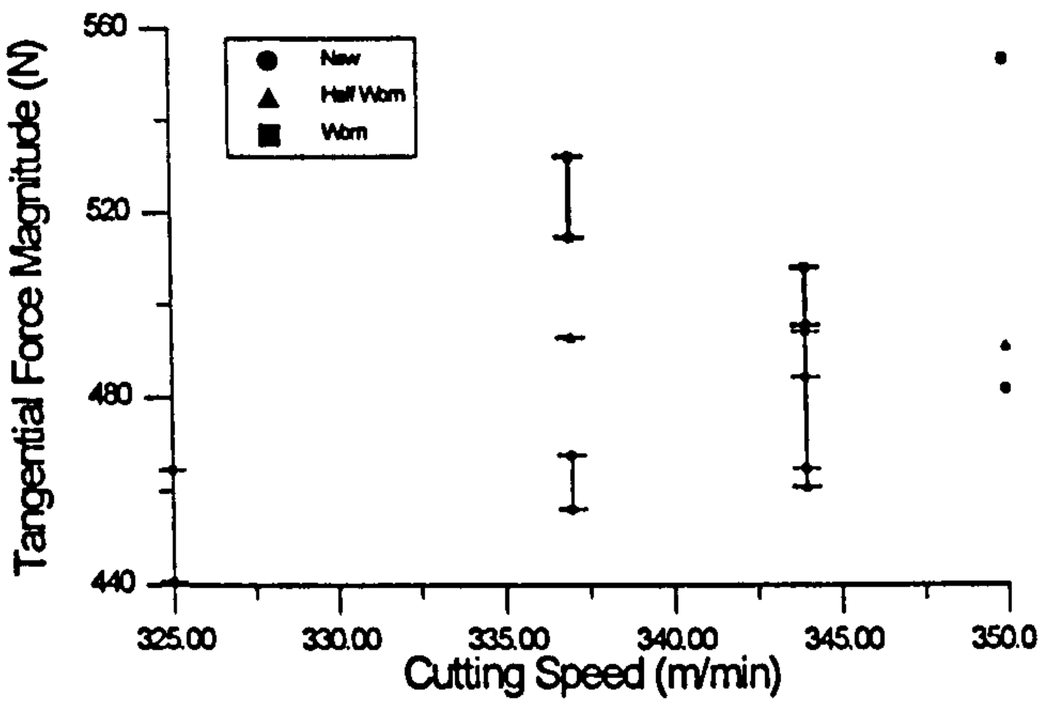


Figure 68: Tangential force versus cutting speed

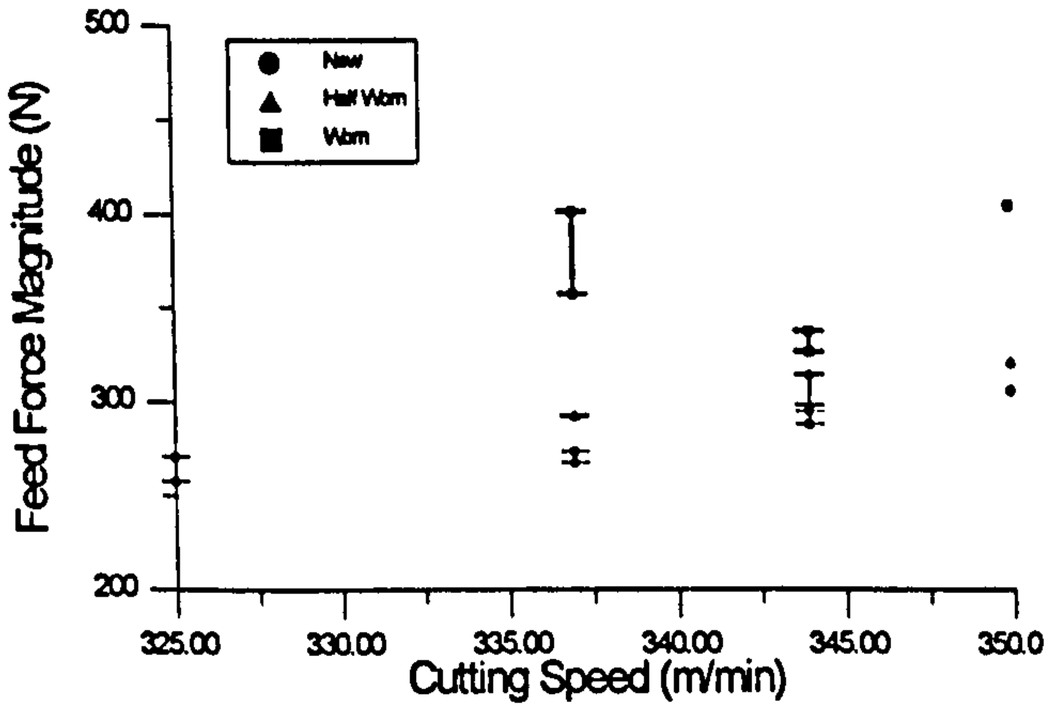


Figure 69: Feed force versus cutting speed

Although no significant increase in either force could be observed with wear, an increase in the absolute deviation for both sound and vibration signals (Figure 70 and Figure 71) proved to be correlated with the cutting speed. An increase in cutting speed resulted in an increase of the absolute deviation which was substantial for the worn tools and followed an almost linear pattern. For a new tool the sound absolute deviation remained constant independent of changes

in cutting speed. Also, an increase in the cutting speed resulted in an increased scatter for the absolute deviation of sound.

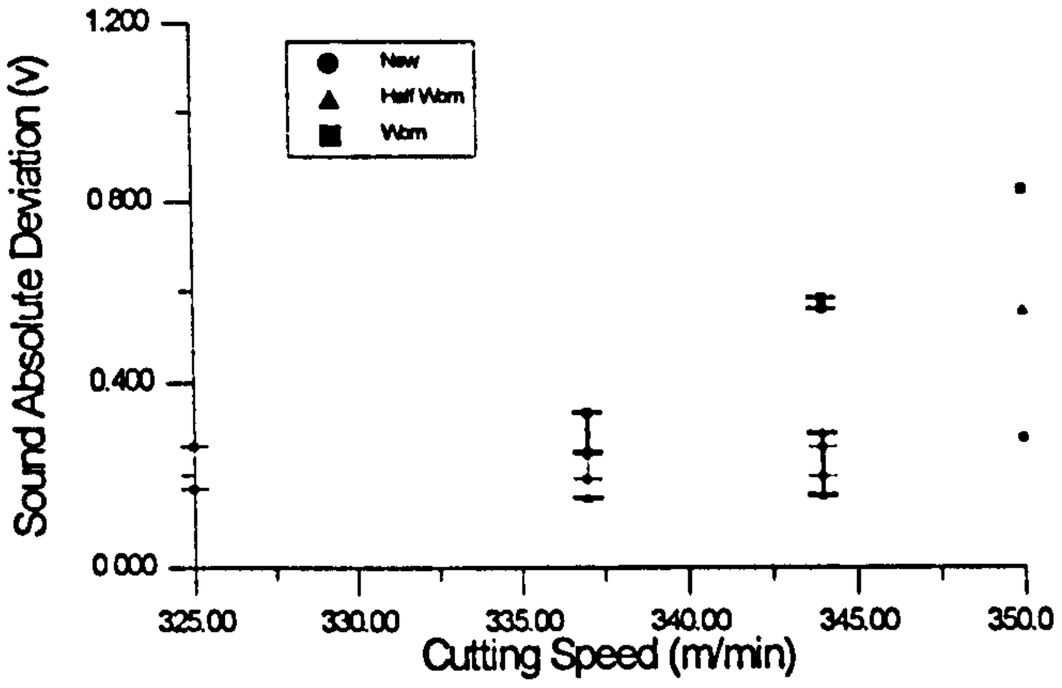


Figure 70: Sound absolute deviation versus cutting speed

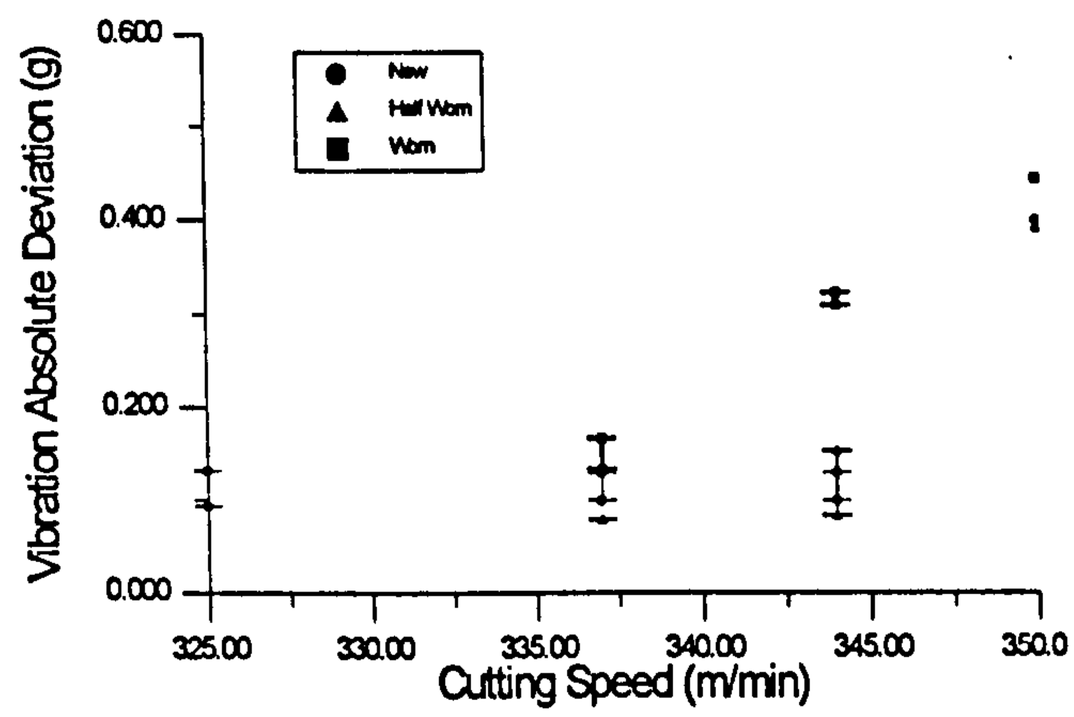


Figure 71: Vibration absolute deviation versus cutting speed

Figure 72 shows the spectrum evolution for a cutting speed of 344 m/min, the first frequency band 2.3±0.1 kHz increases in magnitude with tool wear. For the frequency bands of the sound and vibration (Figure 73 and Figure 74) a general increase with cutting speed occurred associated with an increase in the difference between a new and worn tools. Generally, worn tools report an increase in their power spectrum while new tools seem to remain constant with changes in cutting speed.

R.G. Silva 1997

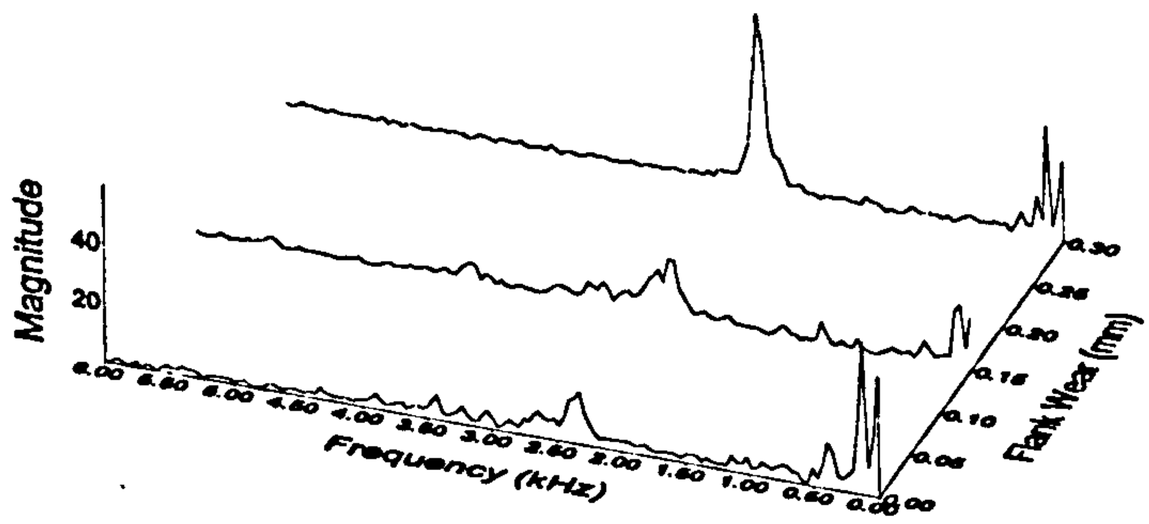


Figure 72: Sound spectrum versus flank wear (cutting speed 344 m/min)

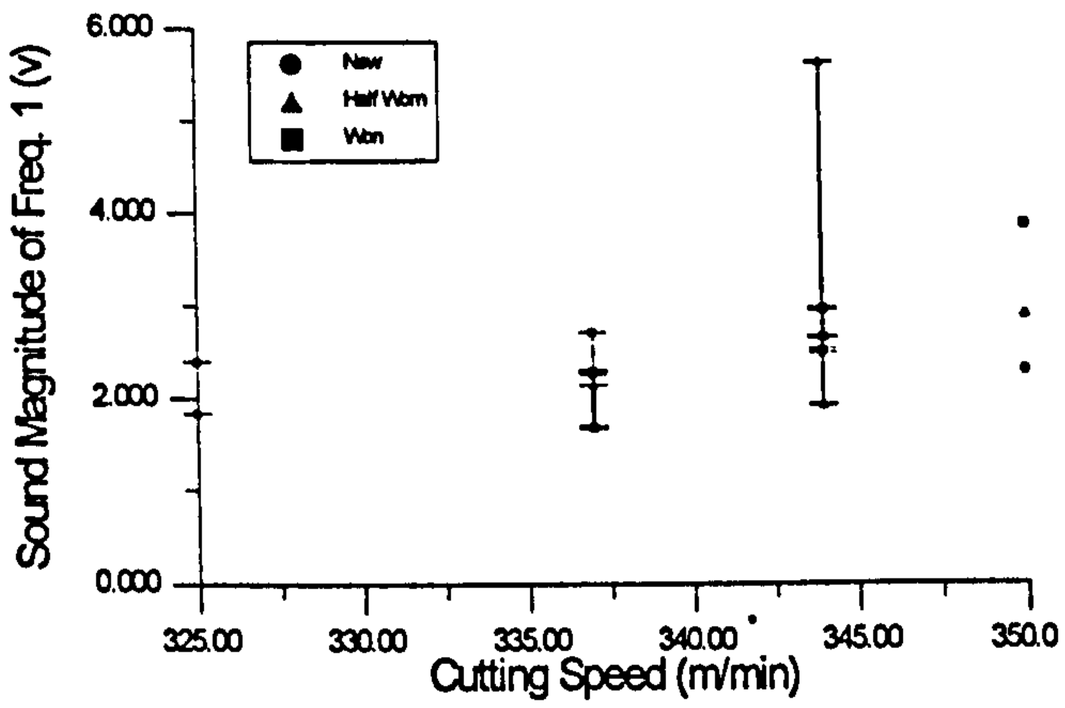


Figure 73: Sound magnitude at frequency band 2.3±0.1 kHz versus cutting speed

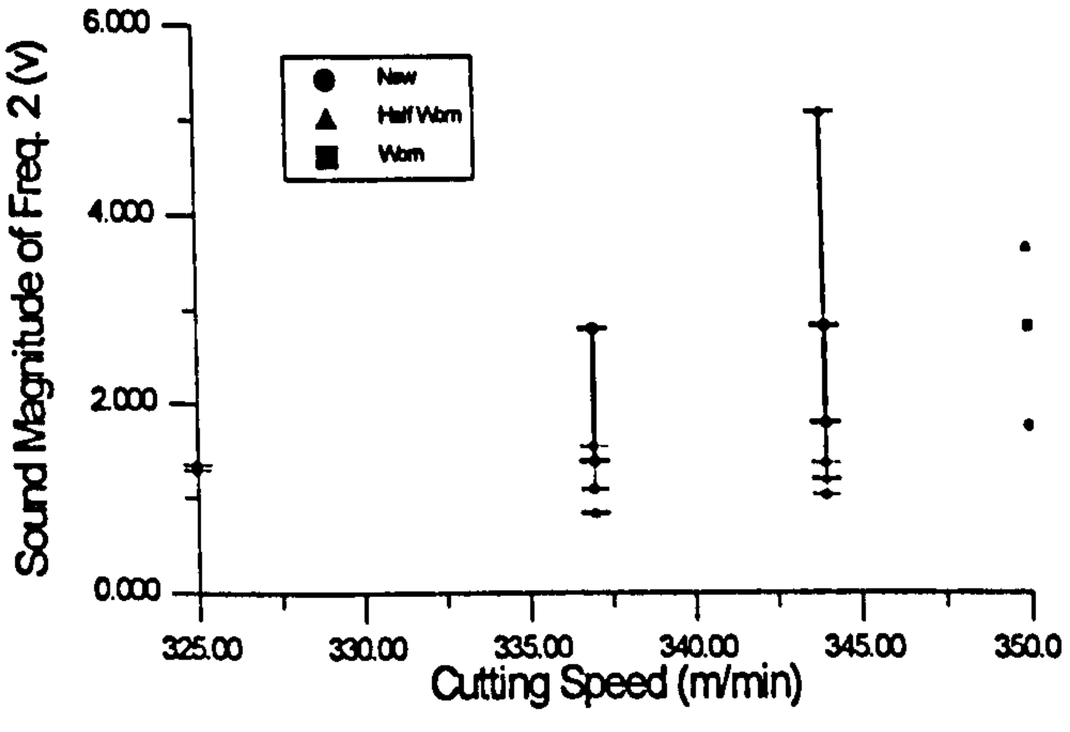


Figure 74: Sound magnitude at frequency band 4.5±0.1 kHz versus cutting speed

The vibration amplitude at frequency band 2.3±0.1 kHz (Figure 75) does not exhibit any substantial changes with cutting speed, with the values only changing a small amount with wear. On the other hand, frequency band 4.5±0.1 kHz (Figure 76) shows a substantial increase on its power spectrum with an increase in cutting speed.

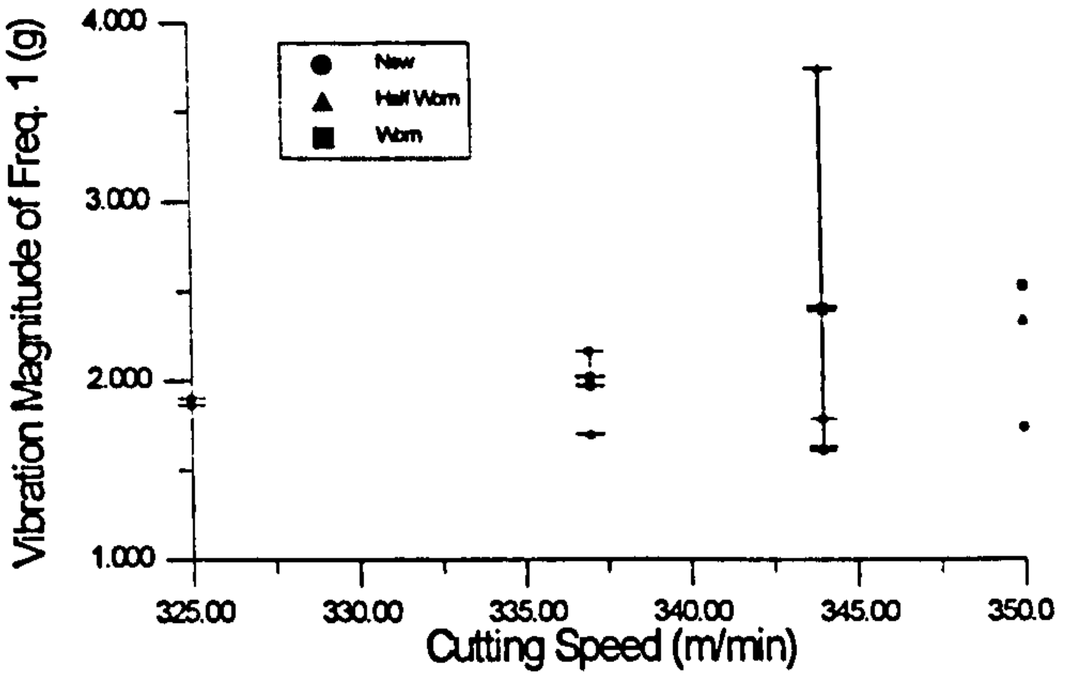


Figure 75: Vibration magnitude of frequency band 2.3±0.1 kHz versus cutting speed

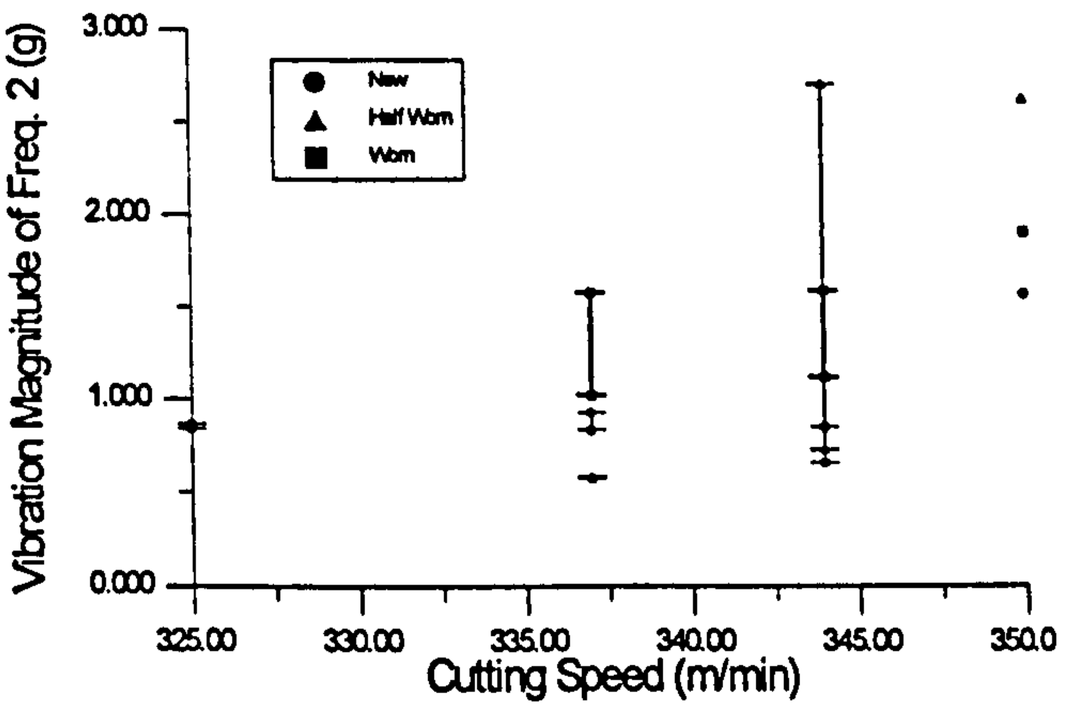


Figure 76: Vibration magnitude of frequency band 4.5±0.1 kHz versus cutting speed

5.4 Summary of Results

The results obtained from the analysis carried on the features extracted is rather complex and is now summarised recurring to a simple set of indicators. These indicators will show simply if a particular feature increases, decreases or behaves randomly with an increase in tool wear and due to changes in cutting conditions.

Table 12 illustrates these variations in each feature recurring to symbols referring to changes with an increase in a determined parameter; "↑" increase, "↓" decrease, "≈" constant, "C" complex", and "R" random behaviour.

Table 12: Summary results on feature analysis

	Feature	Flank Wear	Depth of Cut	Feed Rate	Cutting Speed
Sound	Average	R	R	R	R
	Absolute deviation	R	↓	R	↑
	Skewness	R	R	R	R
	Kurtosis	R	R	R	R
	Frequency band 2.3±0.1 kHz	C	↓	C	↑
	Frequency band 4.5±0.1 kHz	C	↓	C	↑
Vibration	Average	R	R	R	R
	Absolute deviation	R	↓	R	↑
	Skewness	R	R	R	R
	Kurtosis	R	R	R	R
	Frequency band 2.3±0.1 kHz	C	≈	C	≈
	Frequency band 4.5±0.1 kHz	C	↓	C	↑
Forces	Feed force	↑	↑	↑	R
	Tangential force	↑	↑	↑	R
Current	Spindle current	C	C	C	C

Generally, it can be seen that only the forces and the frequency bands appear to be correlated to tool wear, the other features behaving apparently in a random fashion. However, as it will be discussed later on, these features are not absent of significance relative to the cutting process.

6. Neural Network and Expert System Results

The results described in the previous chapter are used here to test the performance of the neural networks towards tool wear classification. Under fixed cutting conditions, sample data from 6 cutting inserts was acquired producing 52 different feature vectors representative of 7 different wear levels. The first 4 cutting inserts (34 feature vectors, each composed of 15 features) were used to train the neural networks and the remaining 2 for testing (18 feature vectors). The tests carried out under different cutting conditions consisted of a different set of feature vectors and these are presented later in this chapter. 50 different cutting conditions were tested of which 25 are presented to assist the discussion.

6.1 Neural Network Results With Fixed Cutting Conditions

The features described in Chapter 5 were used to train both the SOM and ART2 neural networks and results are presented here. Plotting classification results against measured values of wear generates a useful characteristic of the neural network performance, correct classification giving a straight line fit of unit gradient. Therefore, the criterion used in this work for performance evaluation of the neural networks is based on the correlation coefficient, a value near unity indicates good classification (correlation) and one near 0 means poor correlation. The linear regression(s) studied here are the best fits through the origin, comparisons of performance use the best fit through origin because it is coherent to use the origin as an anchor point, and therefore compare the results with the ideal fit. Additionally, to investigate the neural network capacity to perform classification with reduced input information, a comparative analyses was carried out using a reduced input feature vector where the forces and the spindle current were eliminated from the original set of features.

6.1.1 Self Organising Map

The self organising map was trained with data from 4 inserts obtained from fixed cutting conditions for 30,000 epochs and, after this period, organised areas, representative of different wear levels, were created on a 6 by 6 neurone output layer. To determine the optimum number of epochs several tests were undertaken in order to achieve good classification results. An example of how these areas were organised for a new and worn tool is shown in Figure 77 and Figure 78.

In order to facilitate the interpretation of each neurone's wear representation a contour map was created where all the test conditions are mapped, this provides a means for the interpretation of the input vector weight results. The construction of this map is possible given that the representation of similar wear levels occupy adjacent locations on the output neurone grid, therefore allowing the use of the Kriging method for surface meshing. For example, a minimum value on the output map for a new tool feature vector (Figure 77) gives the neurone corresponding to a new tool classification, and similarly for a worn tool (Figure 78). Figure 79 shows the resultant map built from the data obtained after the presentation of all the test inputs, the so called "supervised" sample. The "supervised" sample is a collection of feature vectors representative of the full wear range which are tested via the SOM with the wear level along with the respective neurone activation for subsequent gridding, the result is a "look up table" where each neurone corresponds to a determined value of wear.

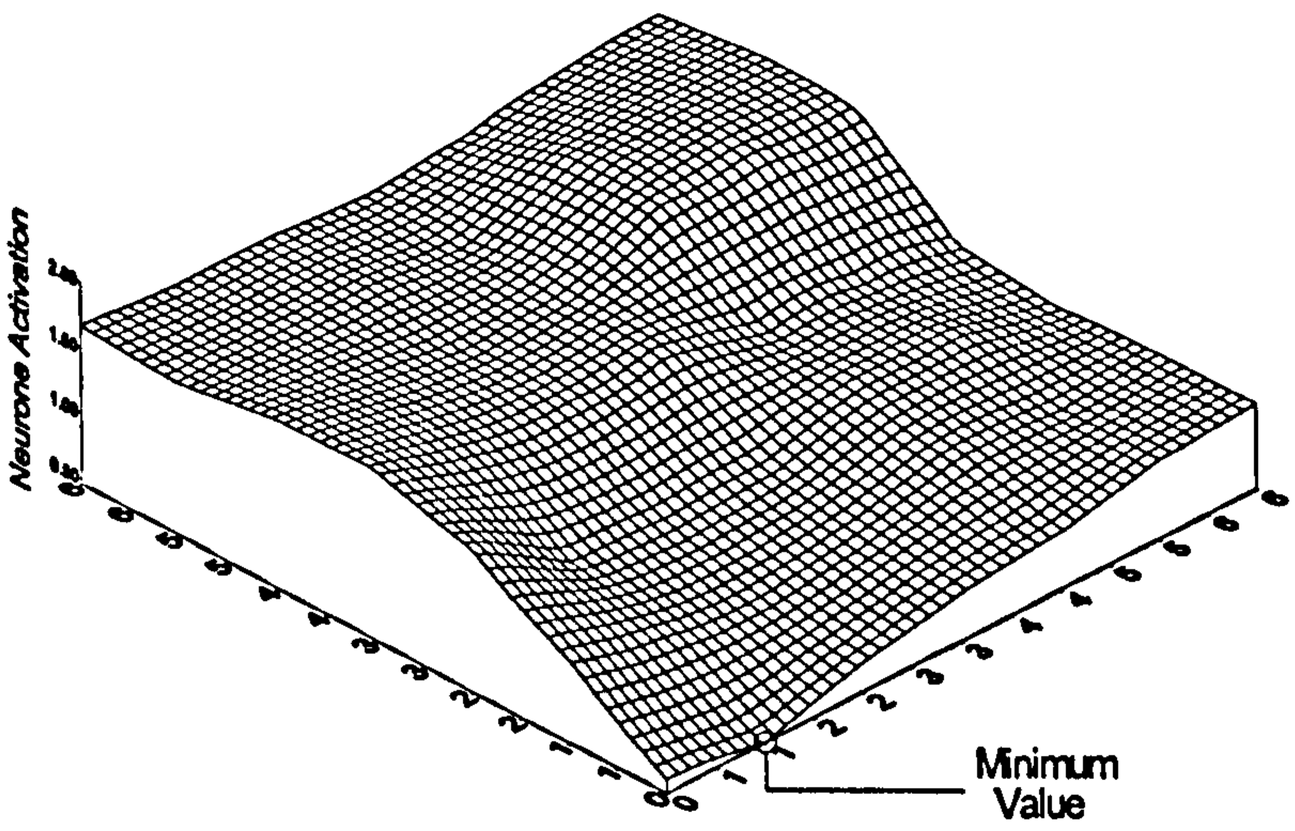


Figure 77: Topological visualisation of neurone activation for a new tool

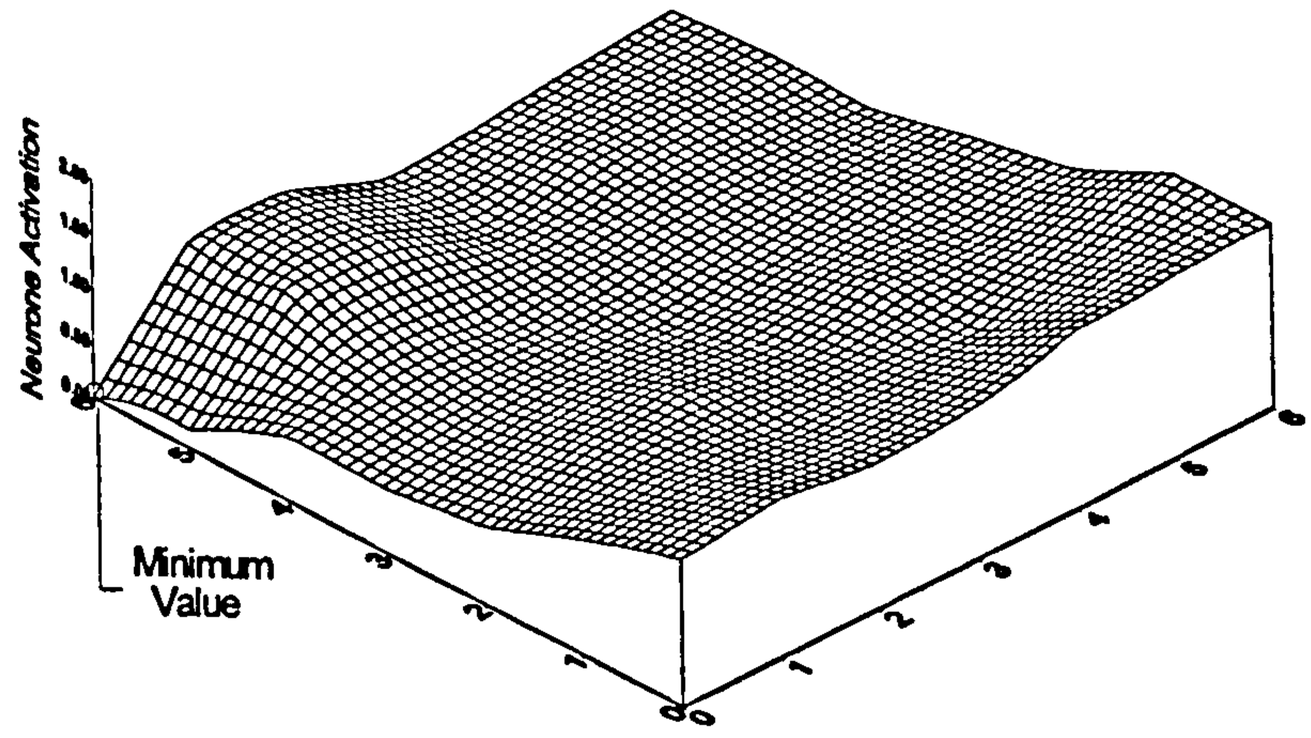


Figure 78: Topological visualisation of neurone activation for a worn tool

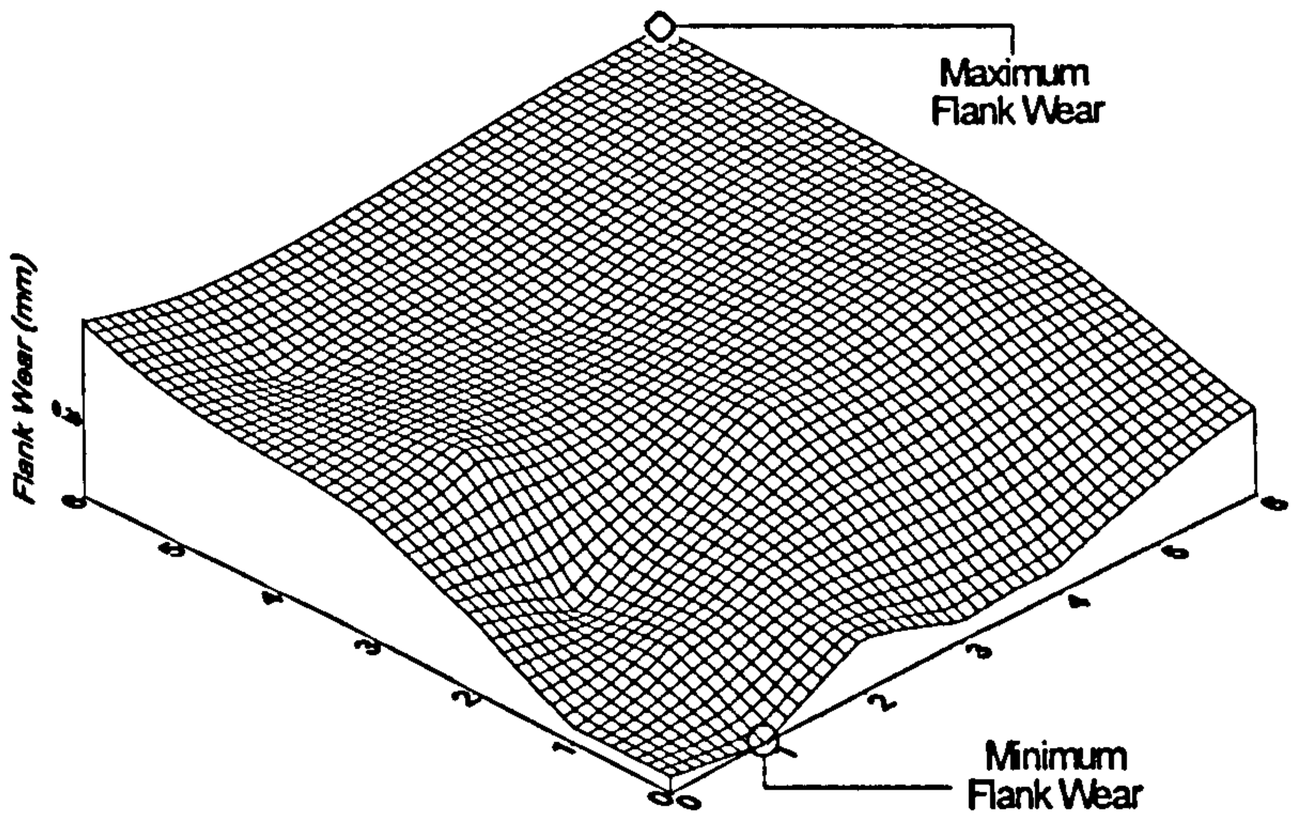


Figure 79: Map of wear states associated with each neurone, "look-up table"

Now that the methodology for obtaining a wear classification with the SOM has been presented the following figures show the success of the SOM at classifying tool wear. The measurement of performance for the self organising map consists simply of plotting the classification results against the measured ones, the straight line representing the ideal fit and the dashed the data fit through the origin. A value of performance is obtained by determining the correlation coefficient of the linear fit through origin, only accounting for the test data. Figure 80 shows the classification results of the SOM given the presentation of all the data using the full feature vector.

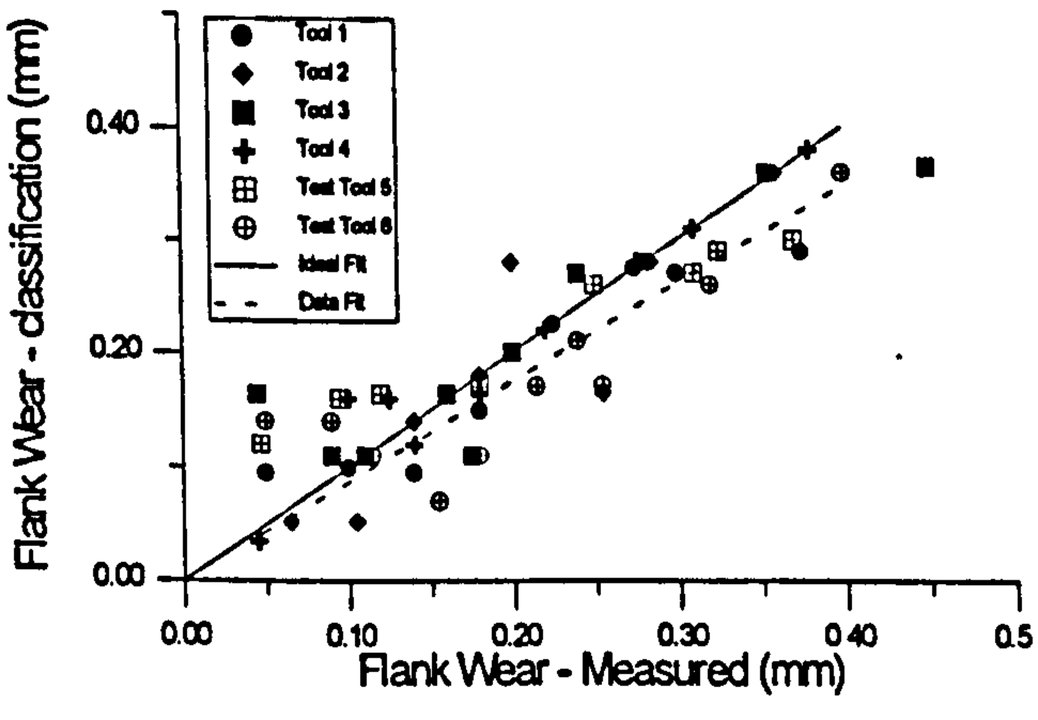


Figure 80: SOM classification using the full feature vector

Figure 80 shows that classification is possible with great accuracy with a correlation coefficient of $r=0.946$, and linear gradient of $b=0.871$. Reducing the feature vector set by removing the forces and spindle current features gives decreased performance but nevertheless the SOM can classify the different wear levels with a reduced accuracy ($r=0.782$, Figure 81).

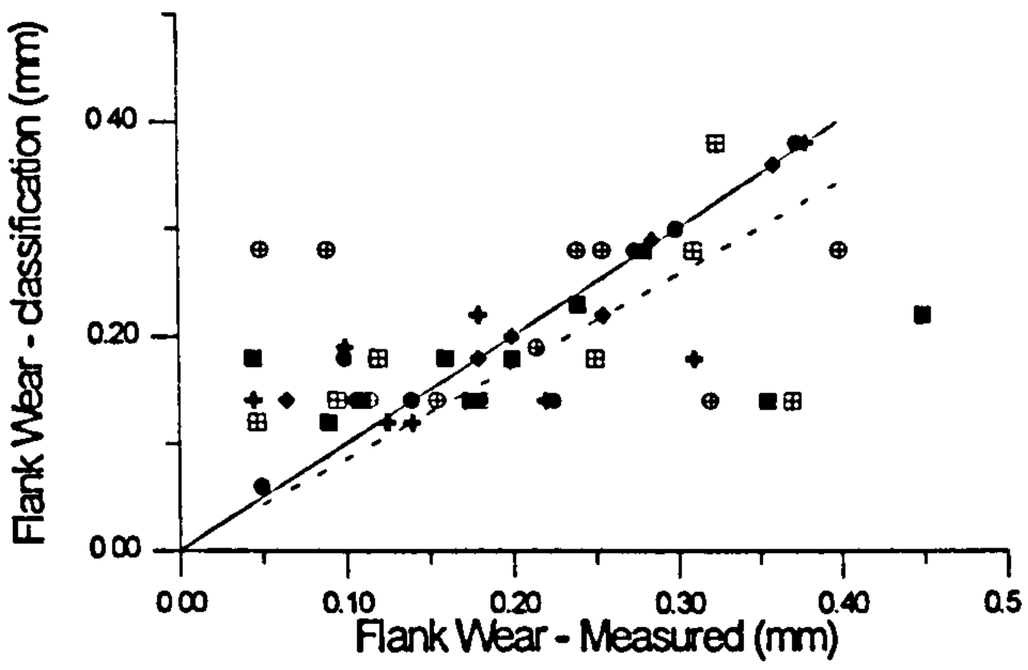


Figure 81: SOM classification with the reduced feature vector

Although the reduced feature vector performance is not as good as the one including the forces and spindle current, the SOM shows some ability to resolve complex information based on poorly correlated features.

The following table summarises the results obtained from analysing the classification results using the SOM. Both the Standard Deviation (SD) and the confidence limits are established from the test data set relative to the ideal fit.

Table 13: Statistical analysis of SOM results

NN / Feature Set	SD (mm)	95% Confidence Limits	
Full feature set	0.0480	0.0058	-0.0209
Reduced feature set	0.0880	0.0198	-0.0296

6.1.2 Adaptive Resonance Theory

The adaptive resonance theory neural network gave its best performance when trained for 1,000 epochs with a vigilance parameter equal to 0.996, these were determined after successive tests from which the best configuration was chosen. The parameters were set to give a reliable training time as well as a minimum of 10 data clusters. The number of data presentations was determined by the magnitude with which the weights changed, as for small weight changes clusters cease to be created.

The same criteria for performance evaluation, as the one used for the SOM, was applied to the ART2 network, consisting of the measurement of the correlation coefficient of linear regression. Figure 82 and Figure 83 show the graphical results respectively for the full and reduced feature vectors.

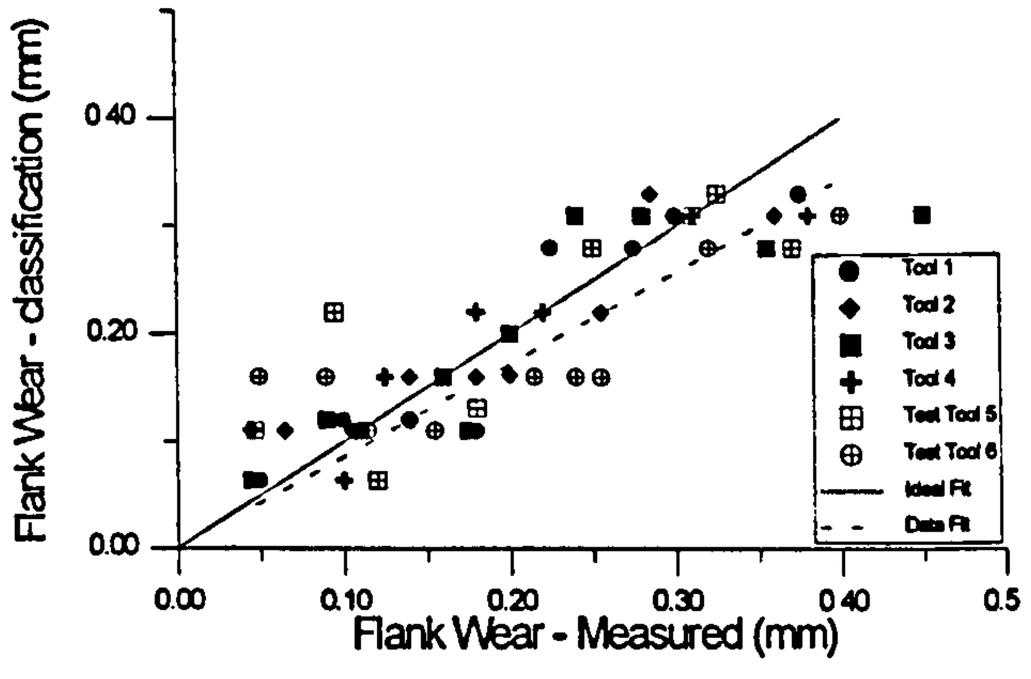


Figure 82: ART2 classification using the full feature vector

A value of $r=0.914$ (gradient $b=0.858$) was obtained for the case where all the features were used and $r=0.691$ (gradient $b=0.689$) for the reduced feature test where a significant gradient change occurred. The ART2 showed a reduction in performance compared to the SOM.

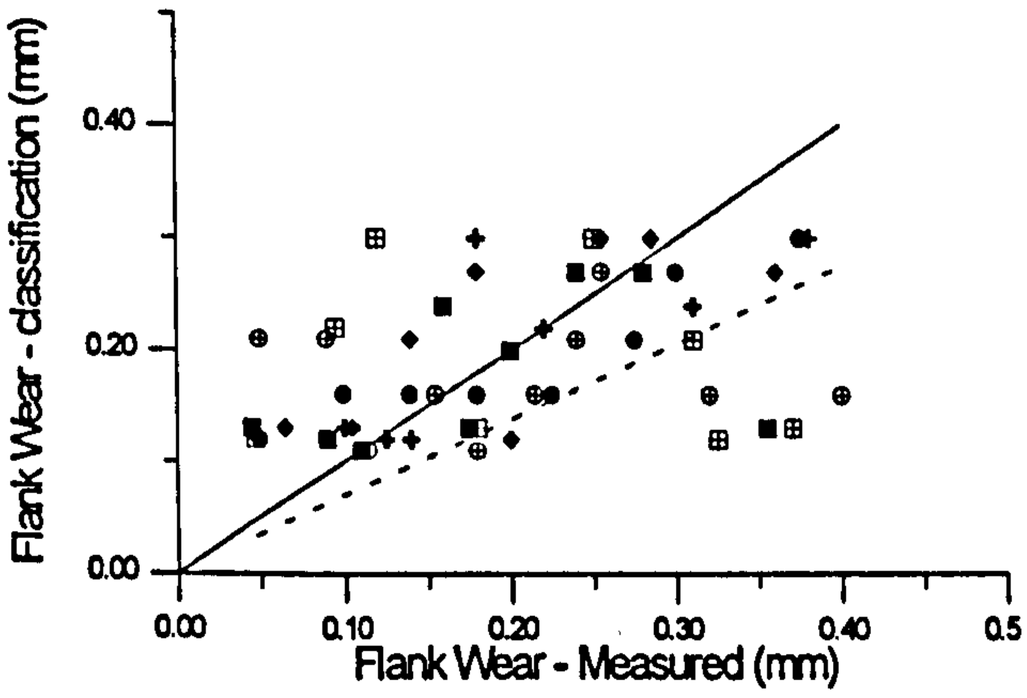


Figure 83: ART2 classification using the reduced feature vector

The degradation in performance for the reduced feature case is associated with the appearance of outliers. These data input vectors do not carry enough information to be classified, or may be contaminated by sporadic environmental noise.

A statistical measure comparing tool wear classification using the ART2 algorithm to the ideal fit is given in Table 14. From this table, as it was seen on Figure 82 and Figure 83, it can be noted that the deviation from the ideal values results in a gradient less than unity.

Table 14: ART2 statistical analysis of results

NN / Feature set	SD (mm)	95% Confidence Limits	
Full feature set	0.0549	0.0067	-0.0239
Reduced feature set	0.0958	0.0194	-0.0345

A summary of the results for both neural networks is given in Table 15. This table presents both correlation coefficient (*r*) and linear slope (*b*). Two results are presented for each neural network configuration, one for each test and training feature vectors (overall), and one for the test set alone (test set).

Table 15: Performance results for NNs

TESTS	b	r
ART2 - Full feature set, overall	0.930	0.960
--, test set	0.858	0.914
ART2 - Reduced feature set, overall	0.902	0.896
--, test set	0.689	0.691
SOM - Full feature set, overall	0.946	0.964
--, test set	0.871	0.946
SOM - Reduced feature set, overall	0.868	0.894
--, test set	0.836	0.782

6.1.3 Expert System Results With Fixed Cutting Conditions

In order to illustrate how the final results were obtained a stepped approach is presented in order to simulate the decisions taken by the Expert System to arrive at the final classification. The first step towards tool wear prediction consists of outlier elimination. This has been based on the tool life estimation given by the Taylor's tool life equation and is implemented in the form of rules (section 4.4.1). The following figures show the results achieved for both neural networks when combined with the Expert System and outlier removal for the reduced feature set, since the full feature set required minimal changes from the network classification.

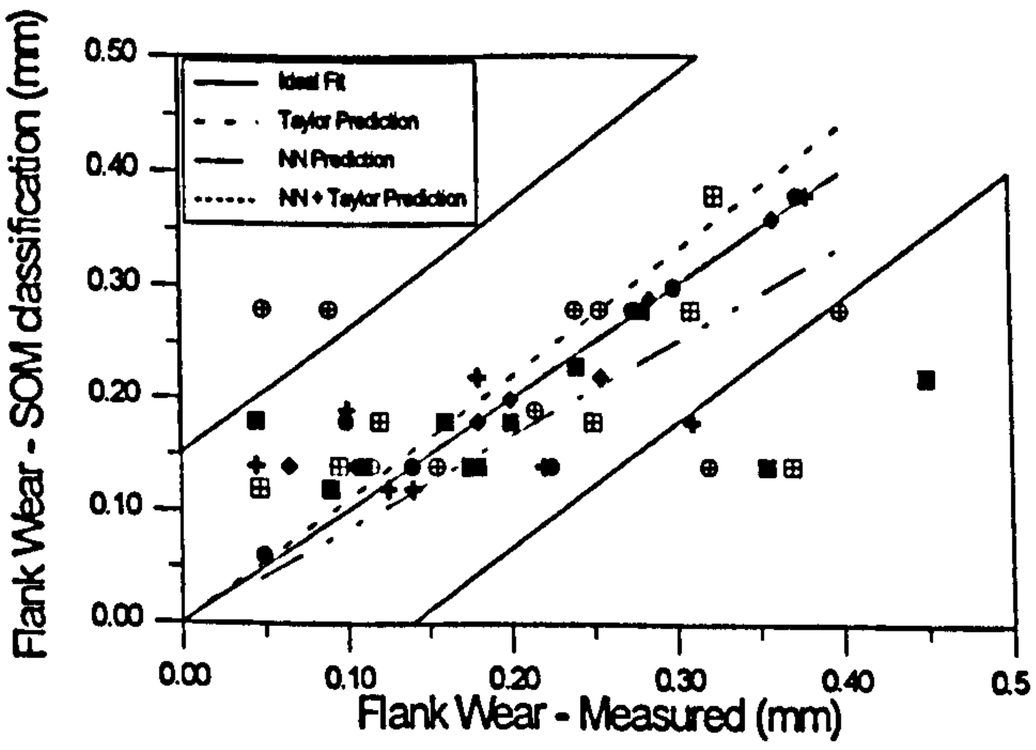


Figure 84: SOM outlier detection

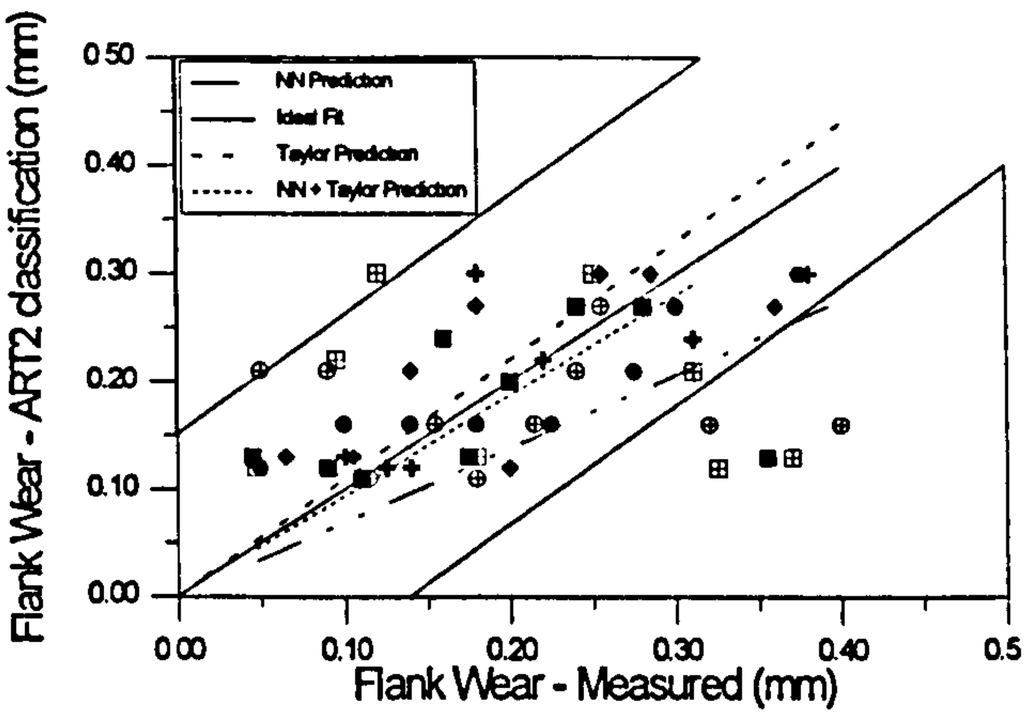


Figure 85: ART2 outlier detection

For the parameters of the Taylor's tool life equation the values obtained were, as mentioned before, $C = 823$ and $n = 0.33$ (Shaw, 1989), this giving a tool life of *13.34* min for a cutting speed of 350 m/min. The dashed lines on Figure 84 and Figure 85 represent the prediction given by the Taylor's equation estimates and the bottom scale corresponds to the actual tool life.

The values obtained for the linear correlation coefficient of linear regression of the reduced feature tests combined with outlier detection were; SOM, $r=0.955$; ART2, $r=0.872$. These results show a marked improvement over those obtained from the neural networks straight after input vector presentation. Clearly, this method can only be successfully used as long as Taylor's tool life equation coefficients are available to the Expert System for the tool/workpiece in use.

Table 16 shows a comparative analyses of the results obtained after the elimination of outliers through the use of Taylor's tool life equation. It is easily seen that not only was there a major improvement in the correlation coefficient but also the gradient, this means that the predictions after outlier detection become more accurate, as well as reliable from the point of consistency. It is opportune to mention that the use of Taylor tool life equation on its own would give overall conservative results.

Table 16: Improvements in NN Performance using Taylor outlier detector

TEST SET	b		r	
	NNs	NNs + Taylor	NNs	NNs + Taylor
ART2 - Reduced feature set	0.689	0.935	0.691	0.872
SOM - Reduced feature set	0.836	1.005	0.782	0.955

6.1.3.1 Expert System Output to the User Interface

Although a value of tool flank wear is shown on the user interface for guidance, the Expert System's judgement has to be based on both neural network tool wear predictions and process history. The current configuration uses only a total of three samples for correlation purposes due to the scarcity of data but this could be improved by increasing the sampling resolution relative to rate of wear.

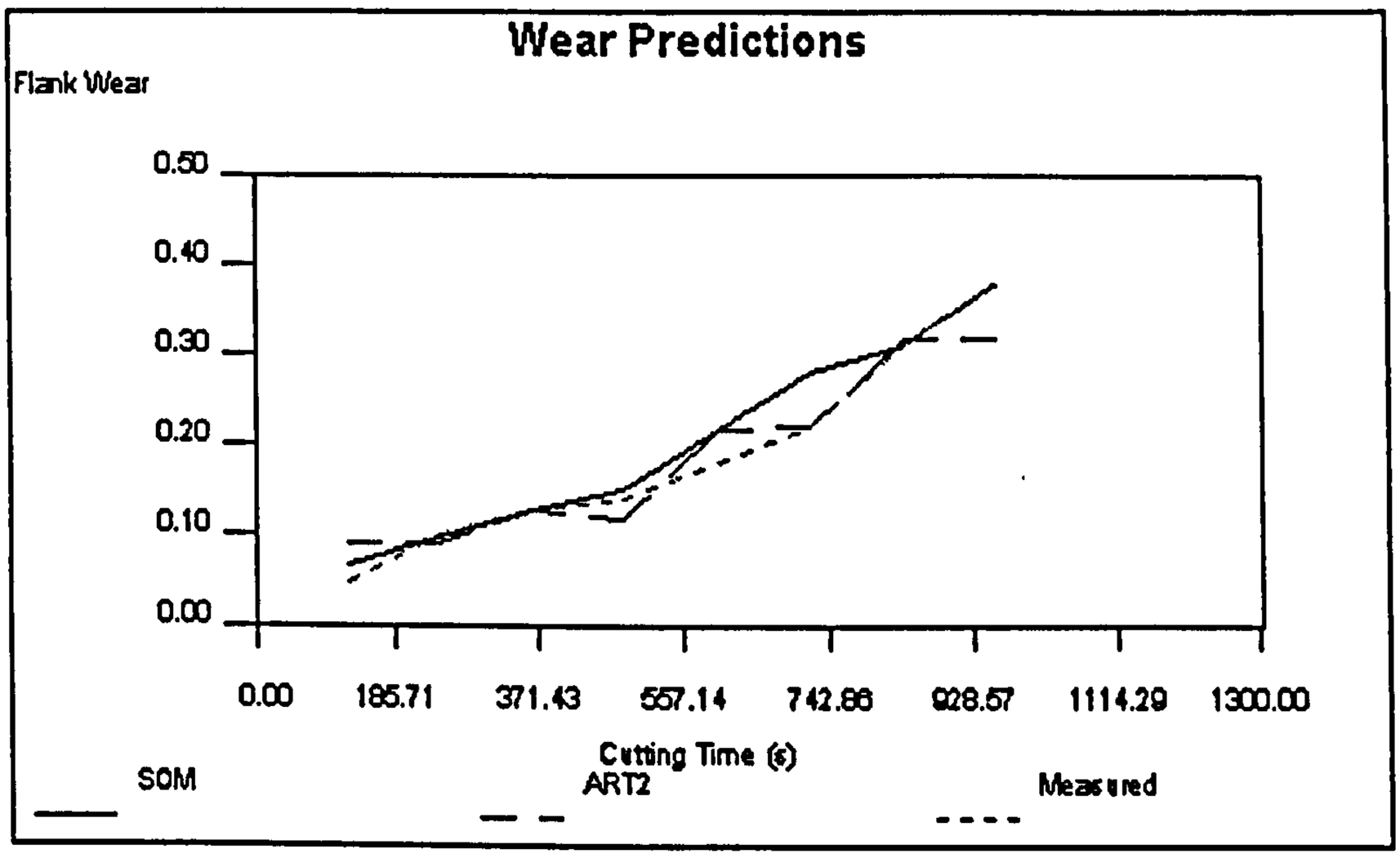


Figure 86: Monitoring screen for NNs: Training sample, tool 4

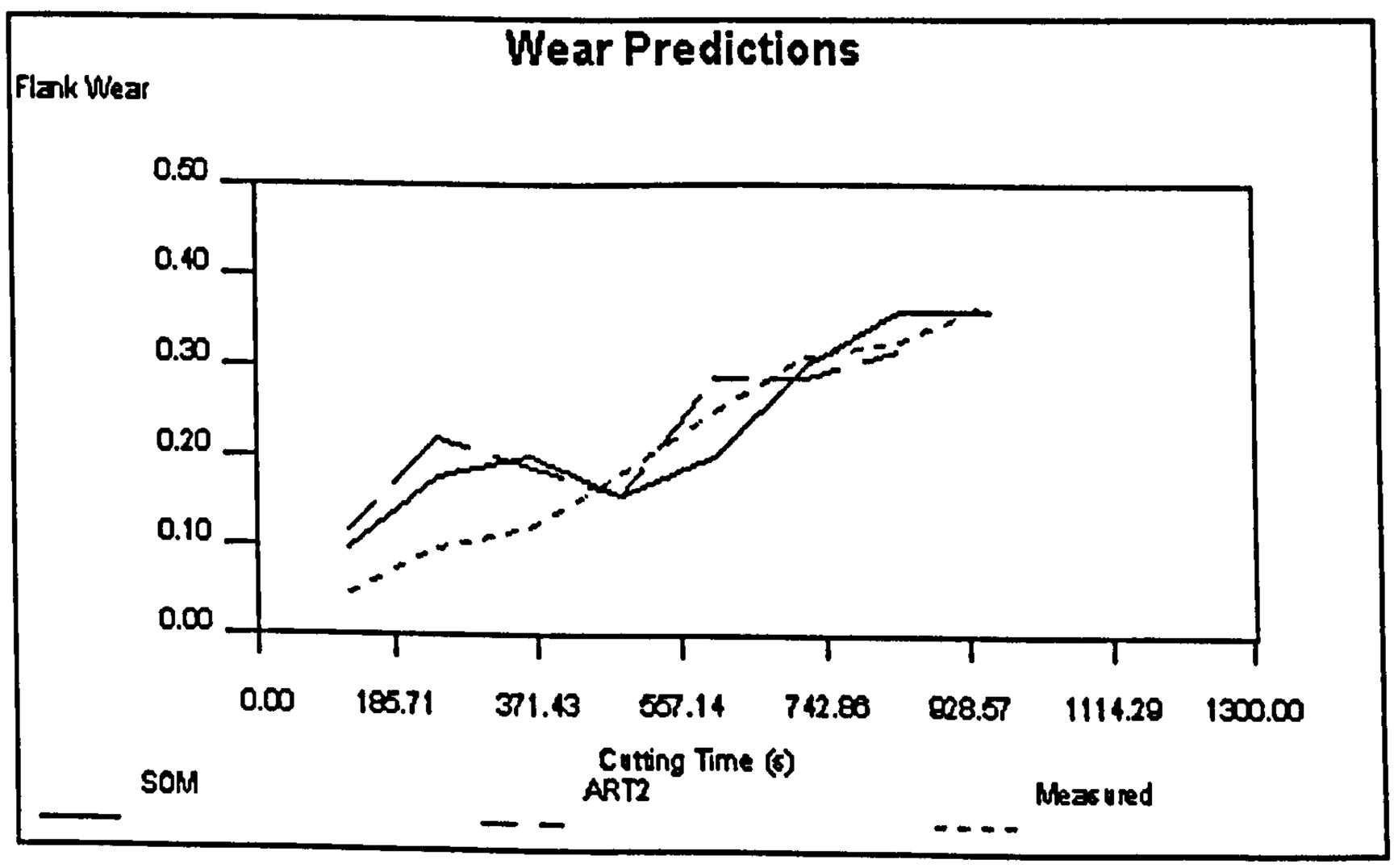


Figure 87: Monitoring screen for NNs: Test sample, tool 5

Figure 86 shows part of the screen on the demonstration system and illustrates the system monitoring wear for *Tool 4* and reporting both neural network results. It can be seen that the system achieves a good accuracy at predicting tool wear. Worn tools were identified successfully for all the training sets by both neural networks, according to the wear criteria. For the test set, Figure 87, consisting of data on the two remaining tools, performance was substantially reduced. Nevertheless the *worn* tool state is recognised accurately. A greater scatter can be observed reflecting less accuracy from the neural networks. Figure 86 and Figure 87 were generated using full feature vectors.

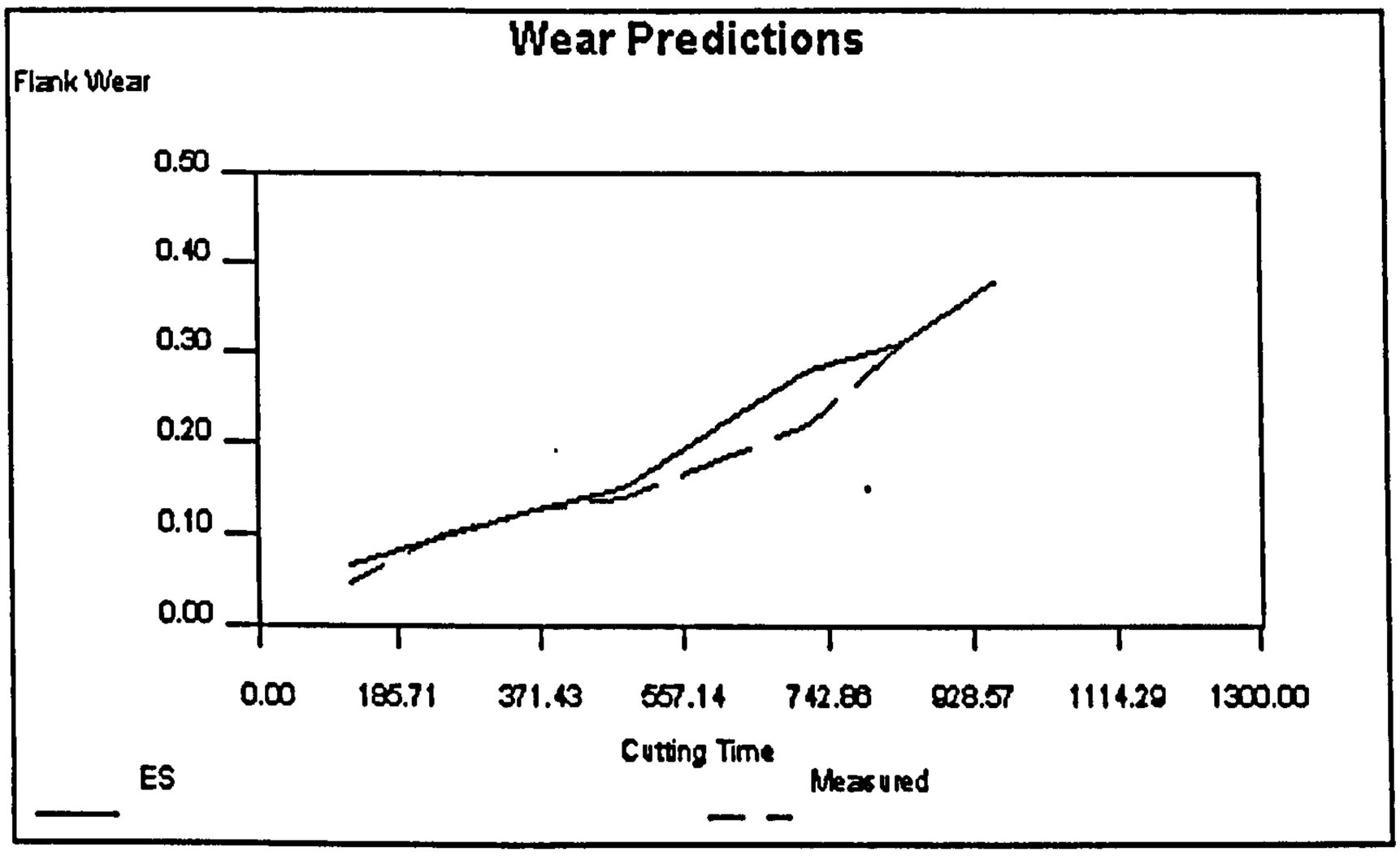


Figure 88: Monitoring screen for ES: Training sample, tool 4

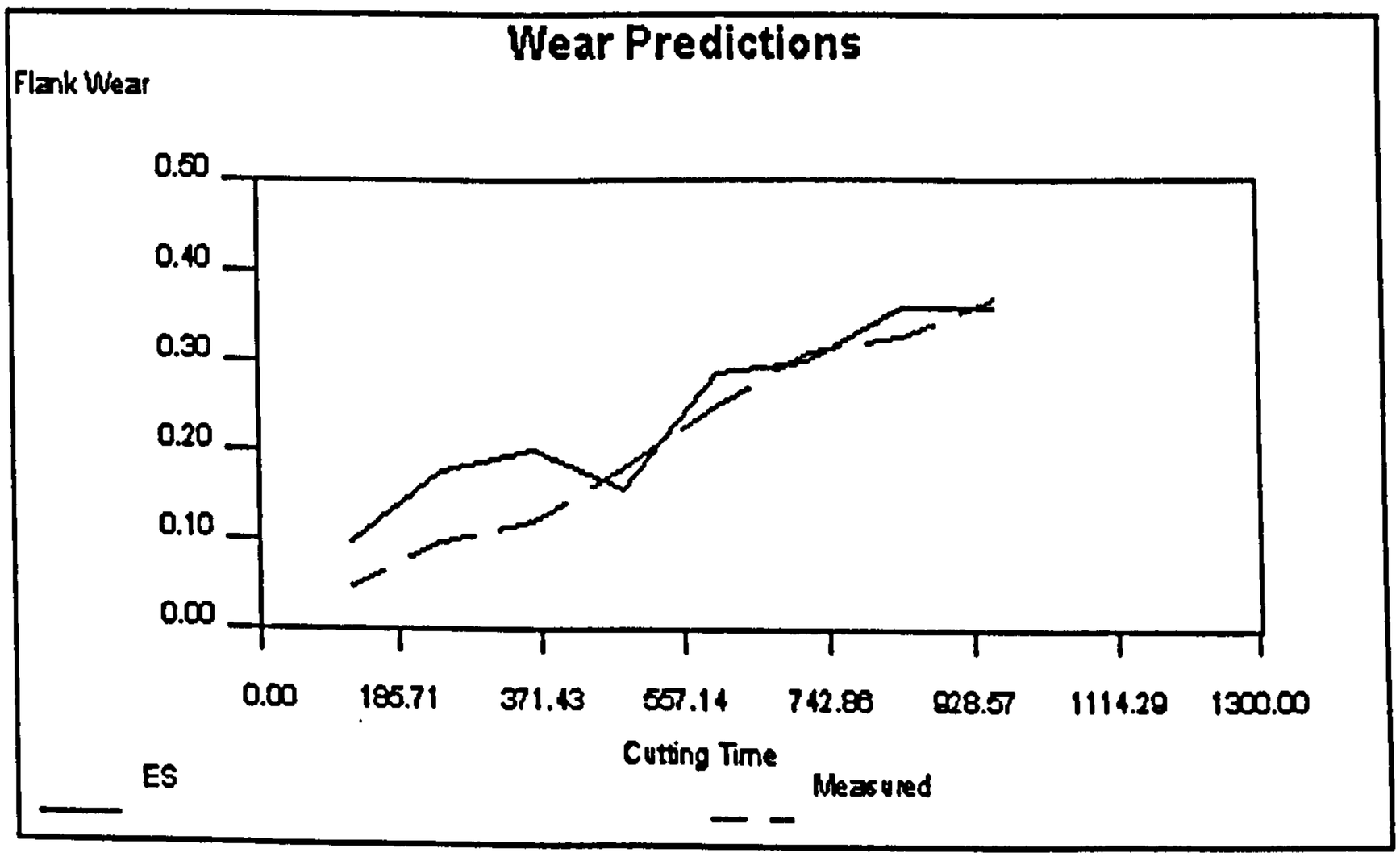


Figure 89: Monitoring screen for ES: Test sample, tool 5

The final result is obtained by combining both neural networks using historical data, these is shown in Figure 88 and Figure 89 for the full feature vector case. It can be seen that an improved classification has been obtained by the use of historical data.

6.2 Neural Network Generalisation With Variable Cutting Conditions

In order to test the system's ability to generalise under different cutting conditions tests were conducted as summarised in Table 17. The present cutting conditions were selected according to recommended values for this tool/workpiece configuration given by SECO TOOLS (1988) guide where values were chosen to account for small changes in tool life which are liable to affect neural network performance. Variations in cutting speed are expected to alter a tool's life, e.g. altering the cutting speed by a factor of 0.86 might double the life of the tool (SECO TOOLS, 1988).

Table 17: List of cutting conditions tested

Test No*	Feed (mm/rev)	Speed (m/min)	Depth (mm)
0	0.250	350	1.000
1	0.275	350	1.000
2	0.250	350	1.500
3	0.200	350	1.000
4	0.225	350	1.000
5	0.250	350	1.250
6	0.250	337	1.000
7	0.250	350	1.125
8	0.300	350	1.125
9	0.200	350	1.125
10	0.250	344	1.000
11	0.300	344	1.000
12	0.250	344	1.125
13	0.300	344	1.125
14	0.275	344	1.000
15	0.225	344	1.000
16	0.225	350	1.125
17	0.275	350	1.125
18	0.225	344	1.125
19	0.275	344	1.125
20	0.300	350	1.000
21	0.250	325	1.000
22	0.250	344	1.000
23	0.200	344	1.125
24	0.175	350	1.000
25	0.325	350	1.000

* used to assist discussion

The depth of cut was always increased in these tests to safeguard the integrity of the equipment since at a lower depth of cut strings of metal chip would form damaging the instrumentation and jeopardising the experiments. The limit imposed on the cutting speed is justified by the fact that a 350 m/min cutting speed was already at the upper end of the recommended value (SECO TOOLS, 1988).

Performance measurements were obtained by averaging two consecutive samples from each test condition (Table 17) with individual sample performance being calculated as the percentage error of the prediction compared to the actual wear value. The maximum percentage error at each wear level was chosen as the final performance measure. The results under these cutting conditions proved that the system is capable of generalising under a certain range of cutting conditions. This capacity differed slightly between the two neural networks as will be shown in the following sections.

6.2.1 Self Organising Map Generalisation For Variable Cutting Conditions

Figure 90 shows the results of the ability of the SOM to generalise when tested at different cutting conditions than those for which it had been trained.

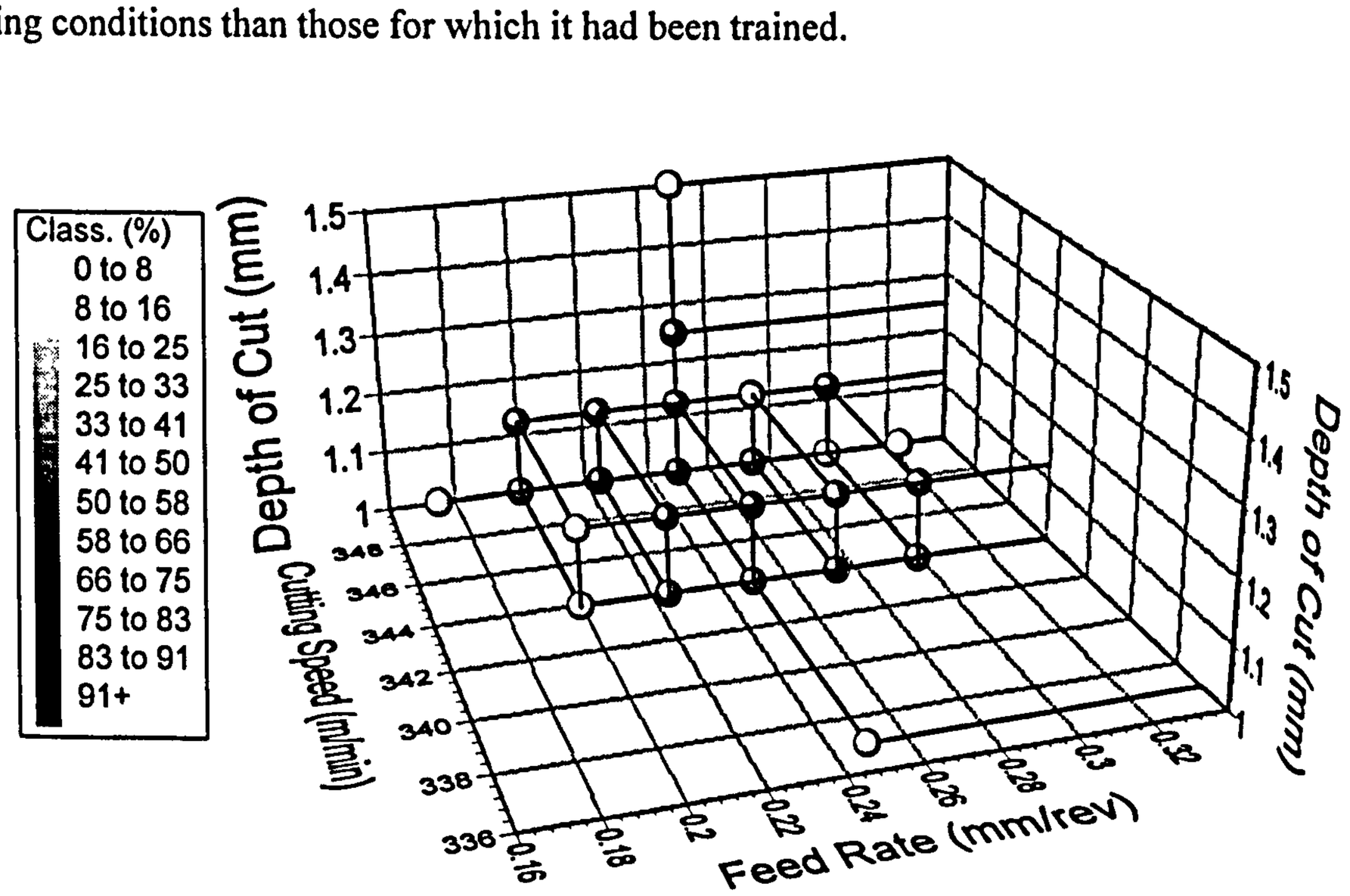


Figure 90: Map of capacity of SOM to generalise for varied cutting conditions

As can be seen, classification was successful (>70%) for variations in feed between 0.2 and 0.275 mm/rev with the other cutting conditions kept constant. As soon as the depth of cut was increased the performance deteriorated and for values after 1.175 mm classification decreased

sharply. Cutting speed changes also resulted in reduced performance of the SOM, only one set of cutting conditions have been shown to achieve 60% classification with speeds of 344 m/min (Test 14).

Generally the further away from the reference cutting condition the worse the classification result becomes, these are summarised in Figure 91 where the cutting speed, feed rate and depth of cut demonstrate this effect. The deterioration rates for individual cutting conditions are; cutting speed 7%/(m/min), feed rate 10%/(0.01 mm/rev), and depth of cut 20%/(0.1 mm).

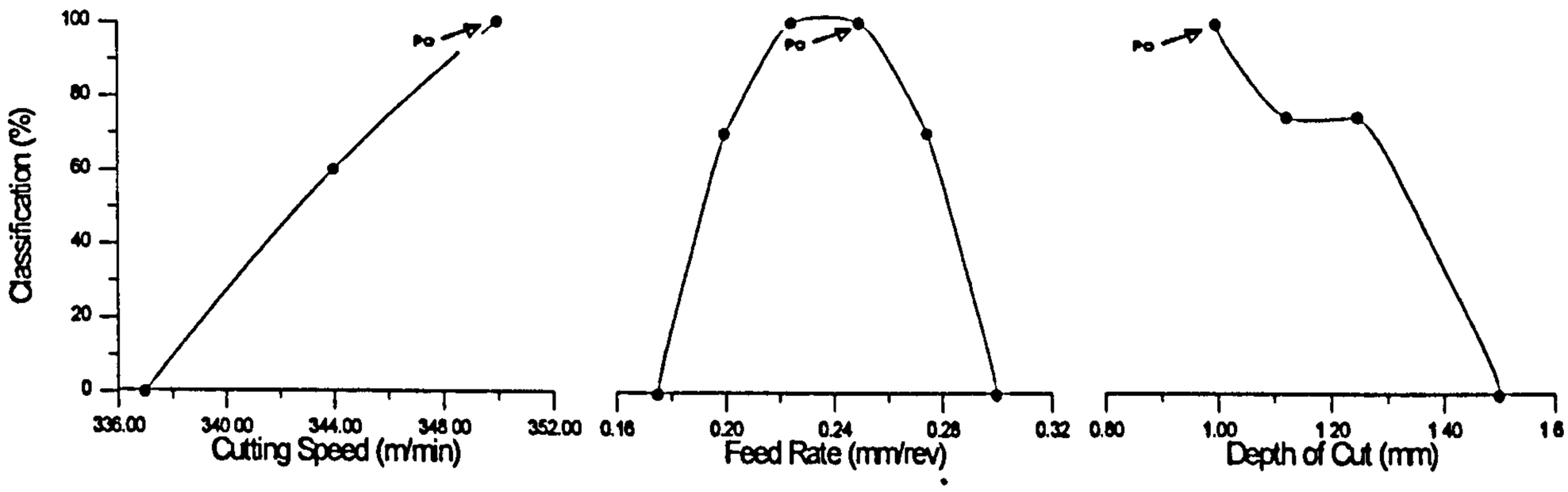


Figure 91: SOM capacity to generalise by cutting condition

6.2.2 Adaptive Resonance Theory Generalisation for Variable Cutting Conditions

The results for the ART2 after testing under different cutting conditions than the ones under which it was trained are shown in Figure 92.

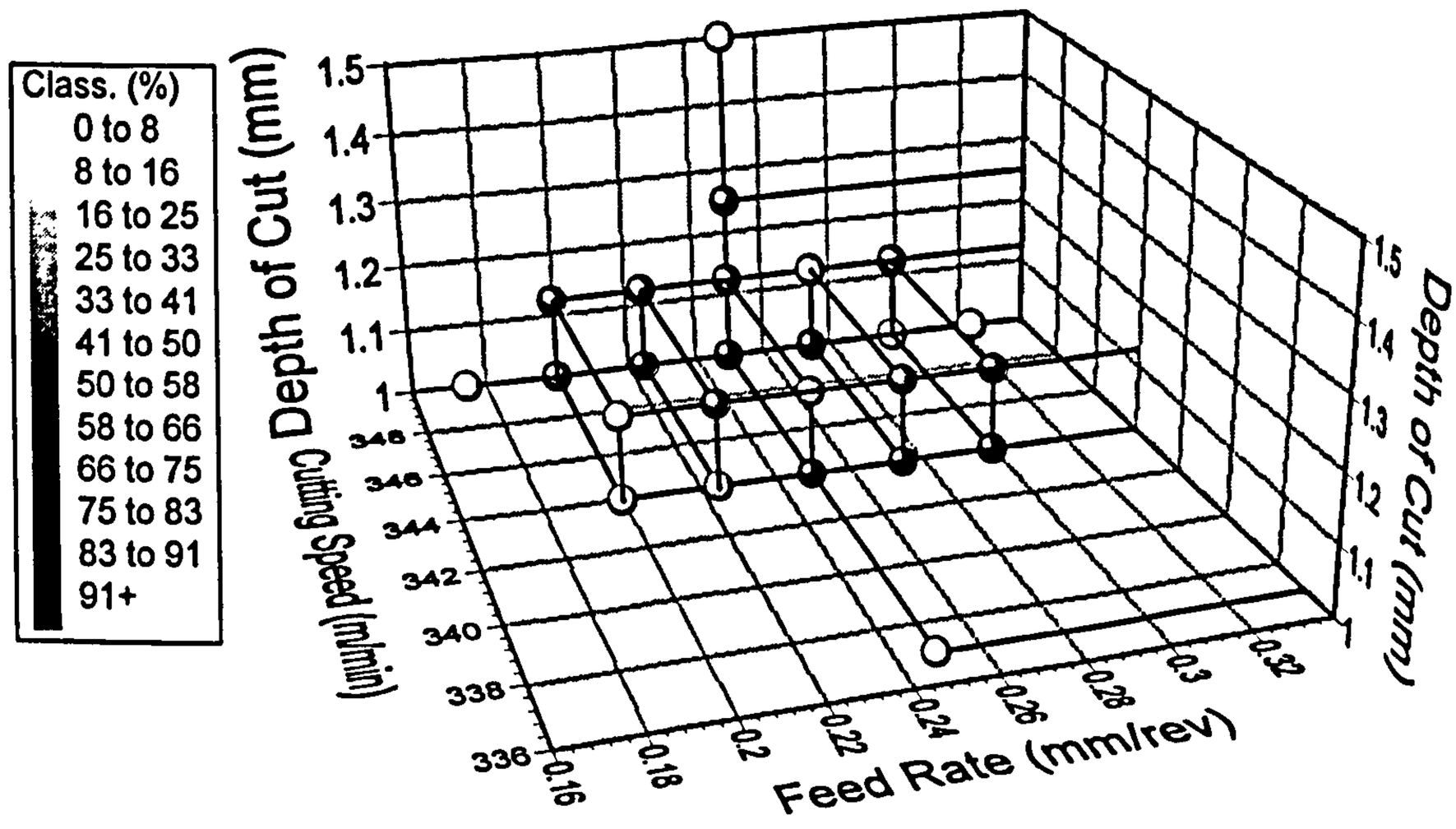


Figure 92: Map of capacity of ART2 to generalise for varied cutting conditions

Variations in feed rate between 0.225 and 0.275 mm/rev result in a classification success grater than 70%. The ART2 performance was successful in classifying patterns with depths of cut up to 1.225 mm, but a further increase in the depth of cut did not produce further successful results. Cutting speed add a significant influence on the ART2 performance. This neural network was generally unable to generalise when presented with data acquired at cutting speeds lower than 344 m/min. At some isolated cutting conditions (e.g. Test 5) the ART2 succeeded in classifying the test samples, but its successful performance was constrained to a limited zone of influence.

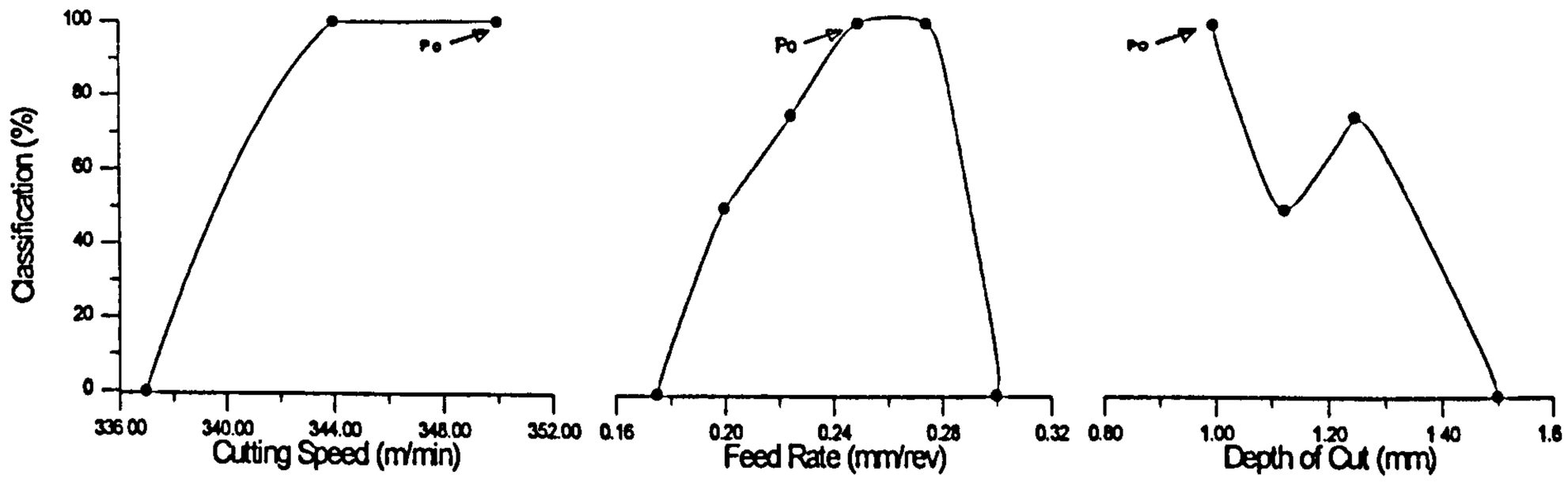


Figure 93: ART2 capacity to generalise by cutting condition

Figure 93 illustrates the description above. The deterioration rates for the ART2 consulting each of the cutting conditions are approximately; cutting speed 100% after 344 m/min, feed rate asymmetrical for feed > 0.275 100% and for feed < 0.25 10%/(0.01 mm/rev), depth of cut 20%/(0.1 mm) average.

In summary, both algorithms achieved similar generalisation capabilities which resulted in successful classification within a common zone of influence.

6.3 Feature Evaluation Based on SOM Weight Interpretation

As mentioned before, the SOM output map is generated by establishing a correspondence between inputs and neurones such that the topological (neighbourhood) relationship among the inputs is reflected as faithfully as possible in the arrangement of the corresponding neurones in the lattice. Therefore, it is possible to establish how different features are contributing towards the final arrangement of the map by analysing the weight distribution associated with each feature.

6.3.1 Criterion for Feature Evaluation

In order to apply the above concept towards the understanding of feature significance the network output is displayed as a two-dimensional distance image, in which a grey scale gives the interpretation of the distances with darker shades corresponding to higher weights associated with a particular zone on the map. Also, higher weights have a higher influence on the final activation of a particular neurone. Therefore, in order to attempt an interpretation of such maps, it is necessary to take into account both shade variation and shade 'darkness'. The distance map is presented individually for each of the features X_i and then its contours computed for visual interpretation.

There are a few key points worth remembering which are essential for the understanding of the following progression. From the algorithm presented earlier (Chapter 4) it can be seen that for a neurone to be selected as representative of a determined input feature vector, its distance to the neurone's weight vector as to be a minimum among all neurones. Therefore, the contour map presented will resemble the evolution of the feature vector, in this case for individual features. It is important to note that the weights were adjusted upon training bearing in mind all the features and not only one at a time, therefore the interaction of two poorly correlated features may resolve into very significant help towards pattern identification. Considering now one feature at a time will not demonstrate which feature combination is the best but will surely show which ones are playing an important role in the organisation of the output map of weights. It is also important to note that the distance calculated is a sum of all the weighted features. Finally, it is also important to note that since the SOM is an unsupervised network it will train based on the data provided prior to knowing the intended purpose of the mapping, that is, the output should reflect the evolution of wear since this is presumably the only variable affecting the sensing devices. This last point will be taken further in the discussion.

Supposing that a particular feature was very weak towards pattern recognition of a determined function, the distance map built with the distance to each of the output neurones would result in a similar activation of all the output neurones. Similar activations are mapped with similar shades on the 2D visualisation map, with strong features creating several well defined shaded areas. The higher the difference between darker and lighter areas the stronger the effect this particular feature has towards feature vector identification.

6.3.2 Feature Visualisation and Interpretation

Figure 94 shows two of the features visualised in this way. The different shaded areas represent the distribution of the weights associated with each feature, the stronger the feature towards tool wear identification the higher the difference between the lighter and darker shades on the map. Therefore, the absolute deviation of sound (a) is of greater use for tool wear monitoring than the average sound (b), Figure 94.

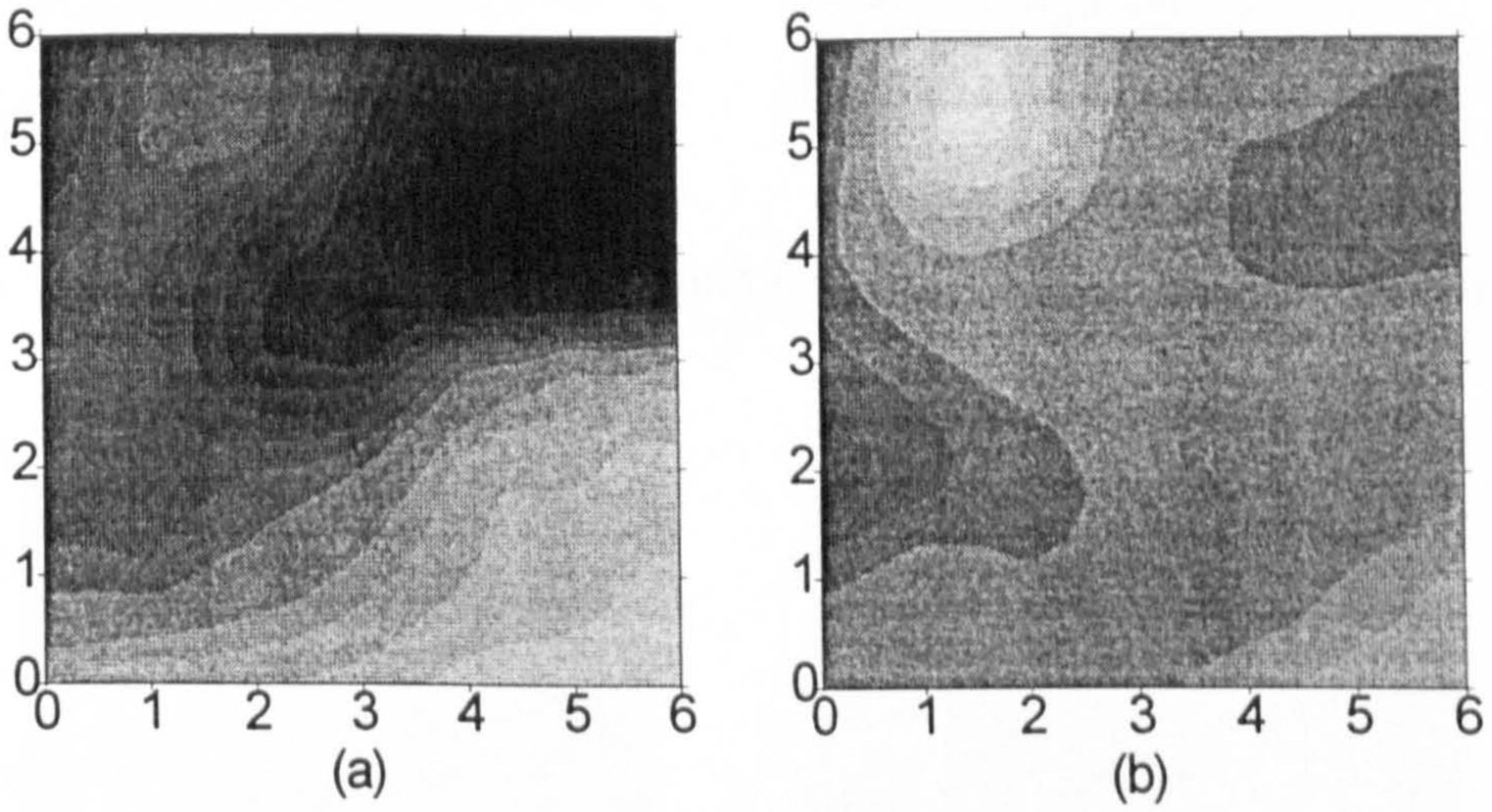


Figure 94: Feature visualisation: (a) Absolute deviation of sound (b) Average of sound

Figure 95 to Figure 101 show the remaining features and the contribution each has towards tool wear prediction. It can be seen that most of the features contribute, in different degrees, towards the classification of the input feature vector. For those features where there is a linear relationship with wear it is possible to distinguish among the different shades which ones correspond to a worn tool and which to a new tool, because they are directly related to the way in which each neurone weight is tuned (e.g. Figure 100 (a)).

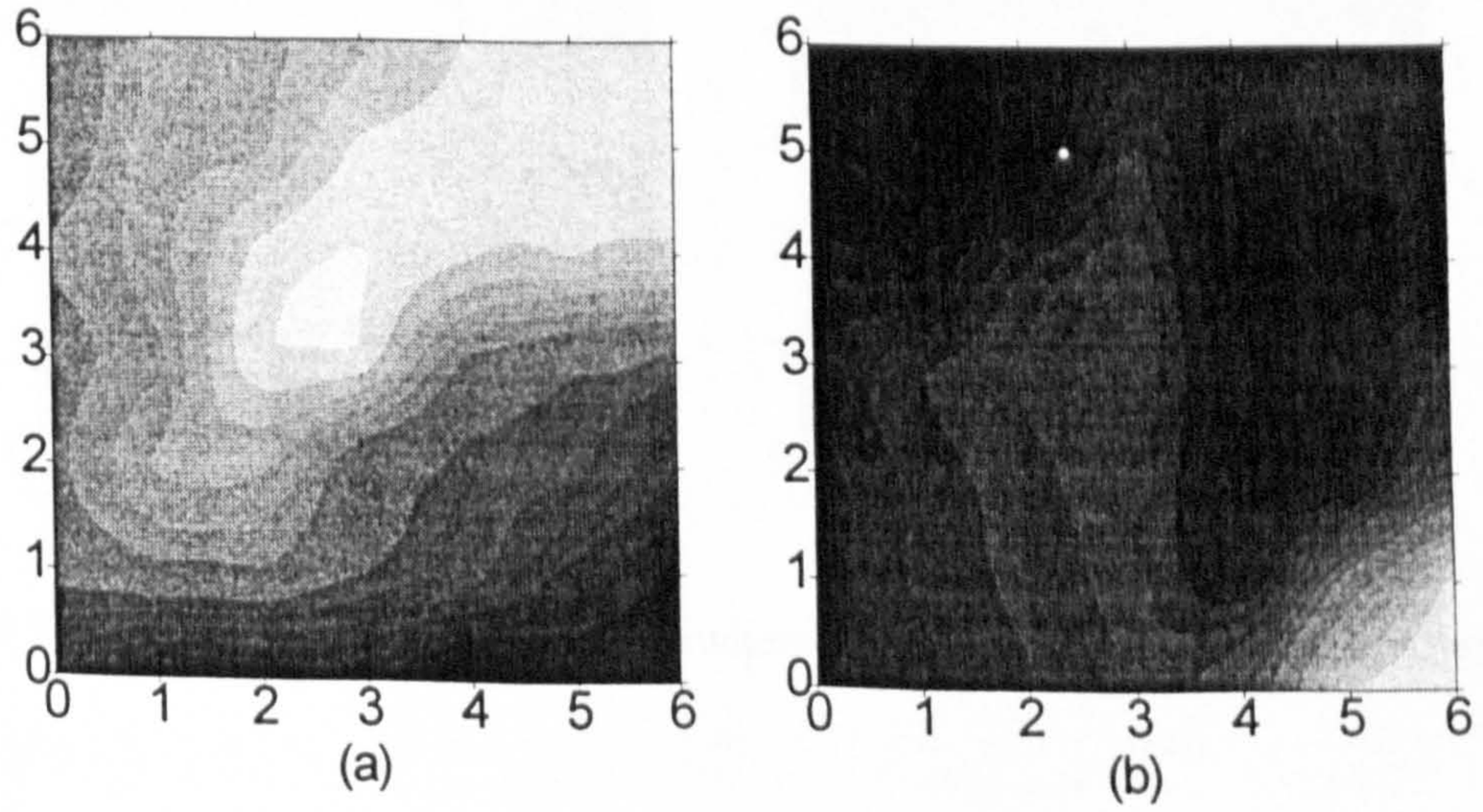


Figure 95: Feature visualisation: (a) Skewness of sound (b) Kurtosis of sound

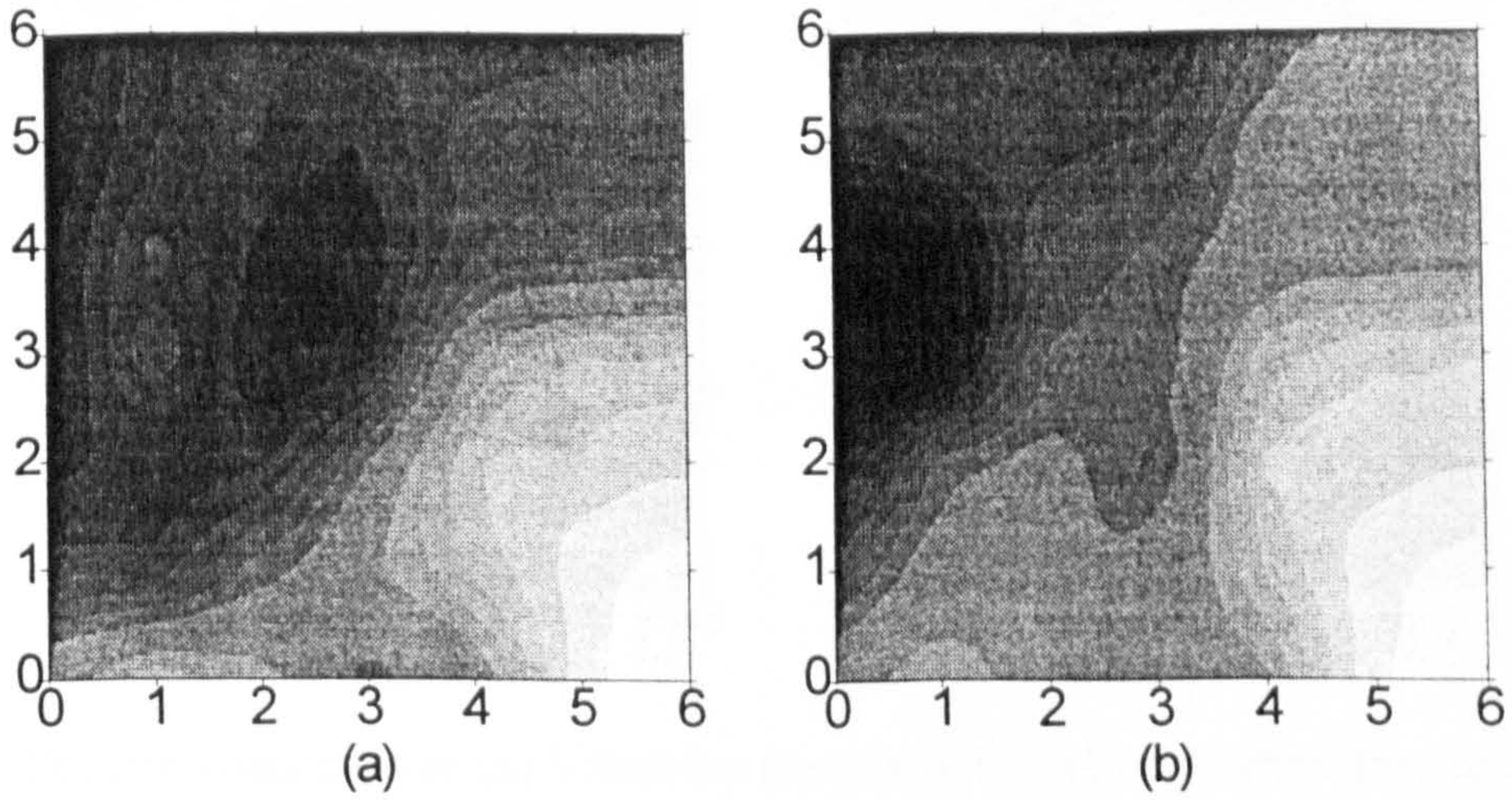


Figure 96: Feature visualisation: (a) Sound frequency b. 1 (b) Sound frequency b. 2

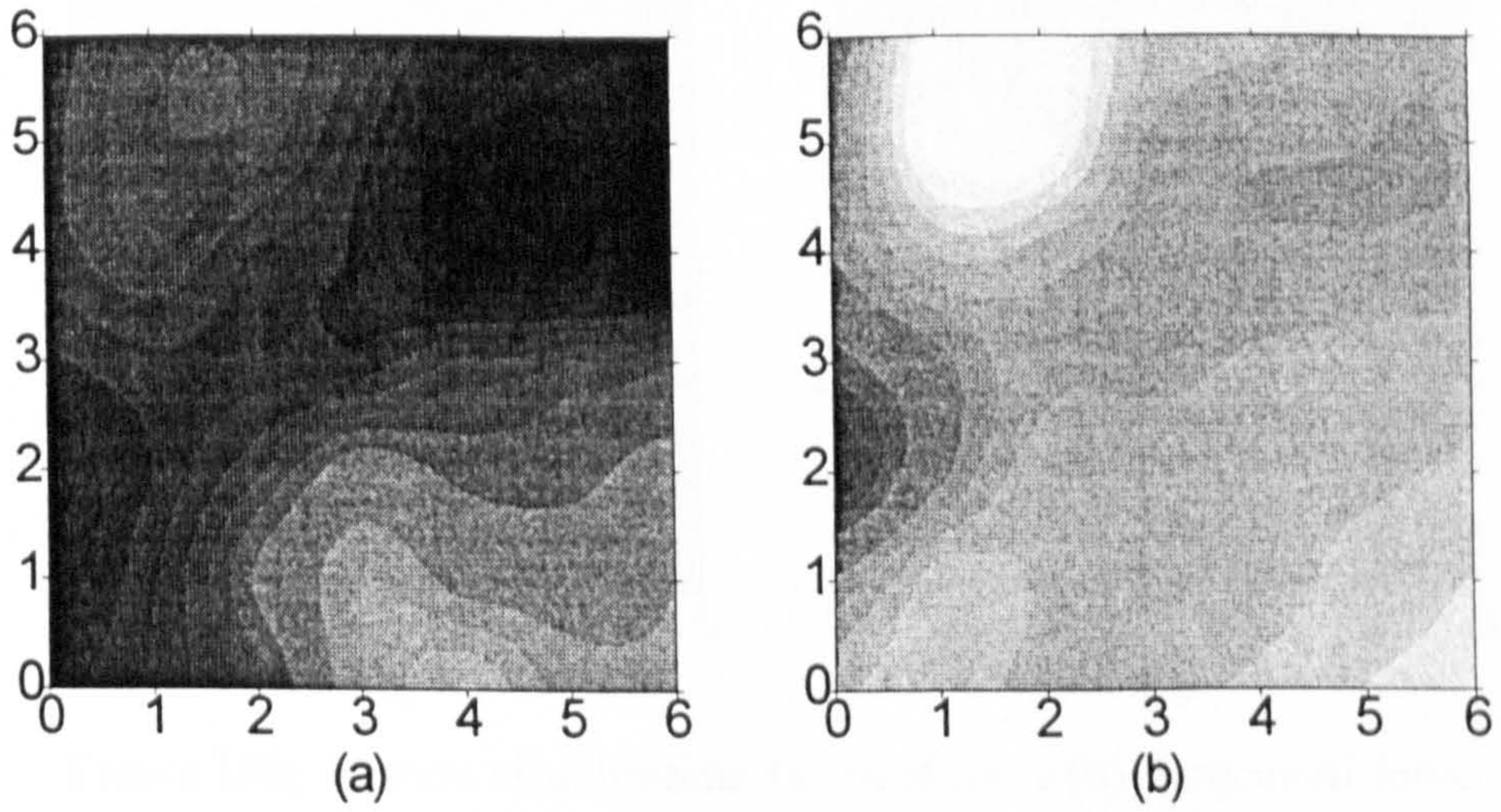


Figure 97: Feature visualisation: (a) Absolute deviation of vibration (b) Average of vibration

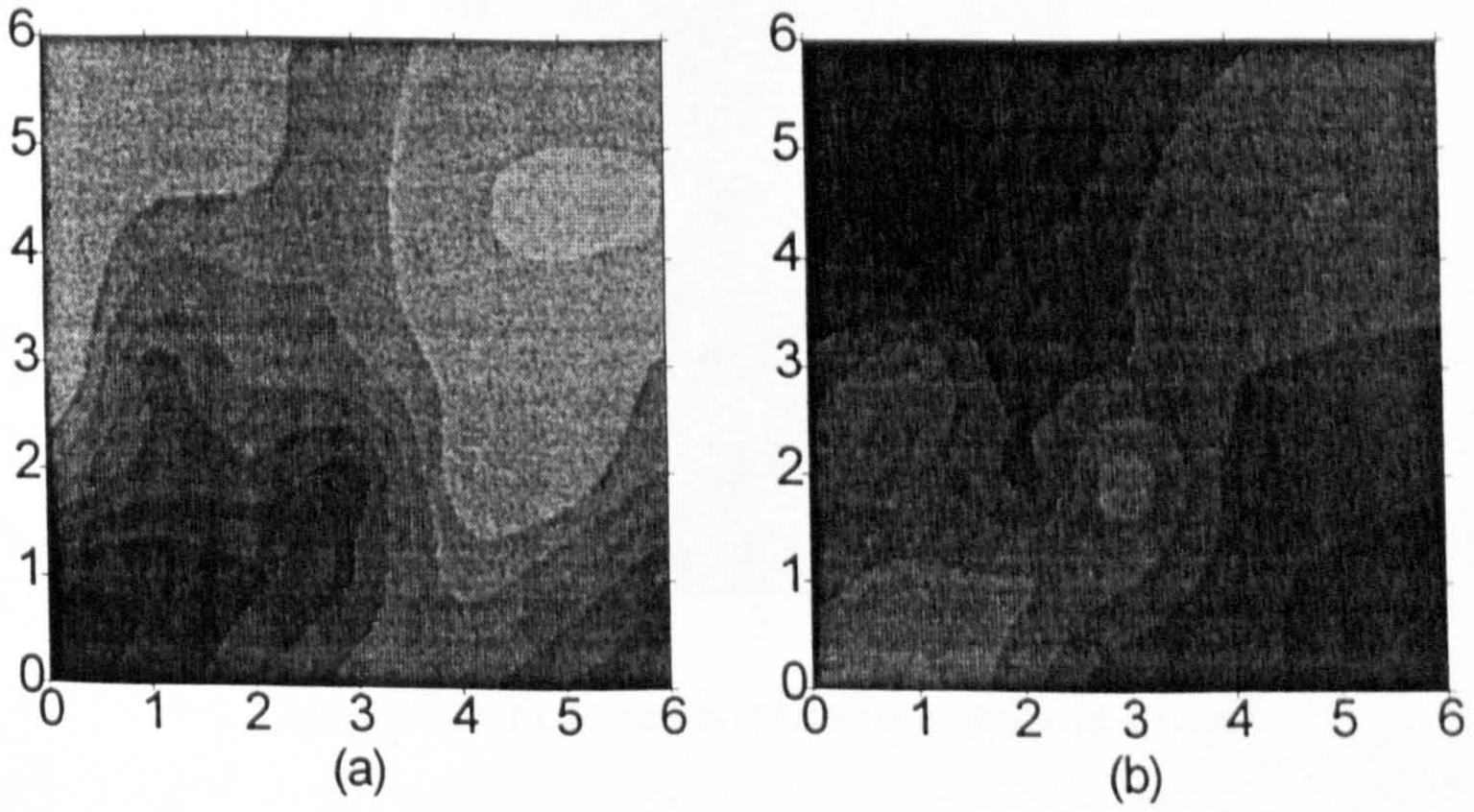


Figure 98: Feature visualisation: (a) Skewness of vibration (b) Kurtosis of vibration

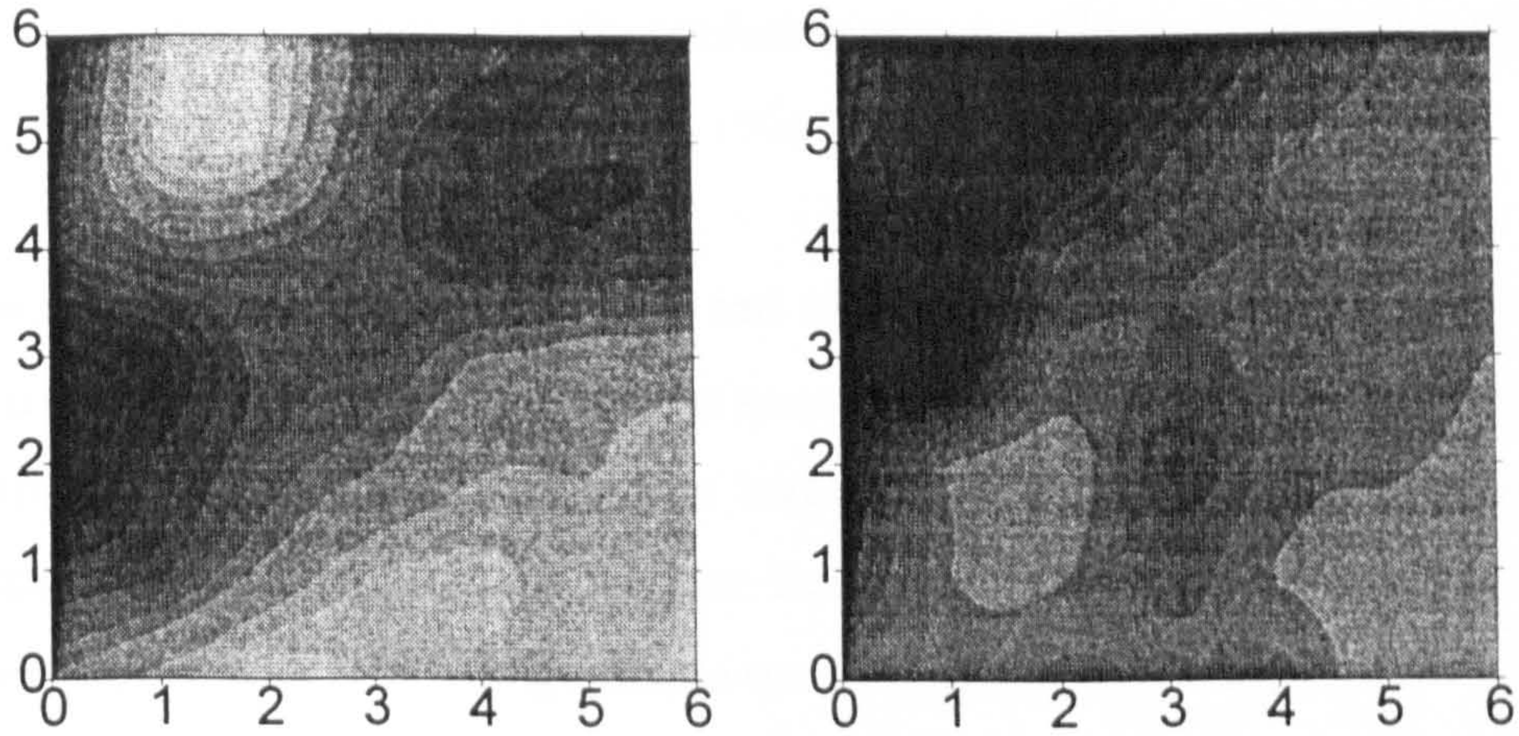


Figure 99: Feature visualisation: (a) Vibration frequency b. 1 (b) Vibration frequency b. 2

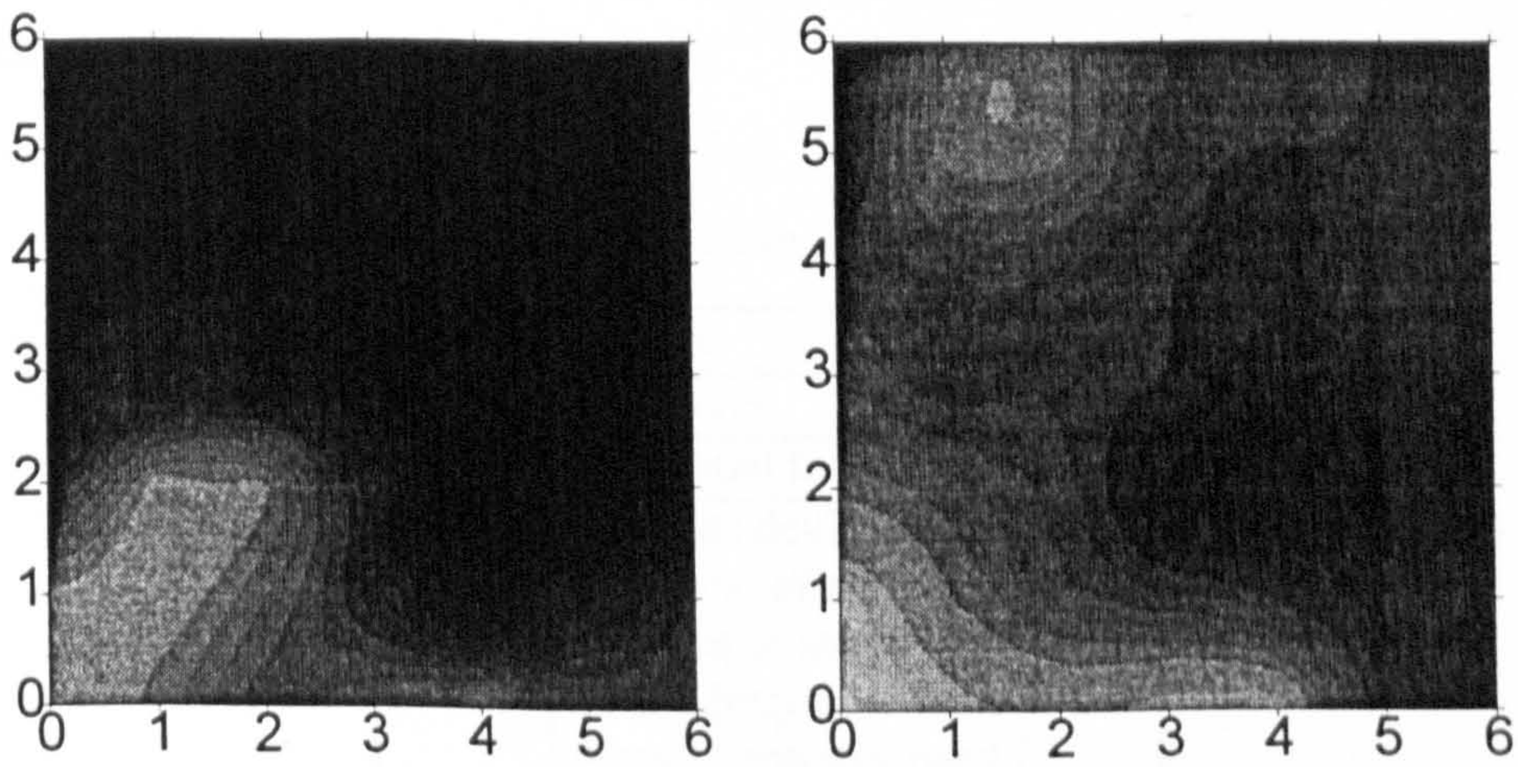


Figure 100: Feature visualisation: (a) Feed force (b) Tangential force

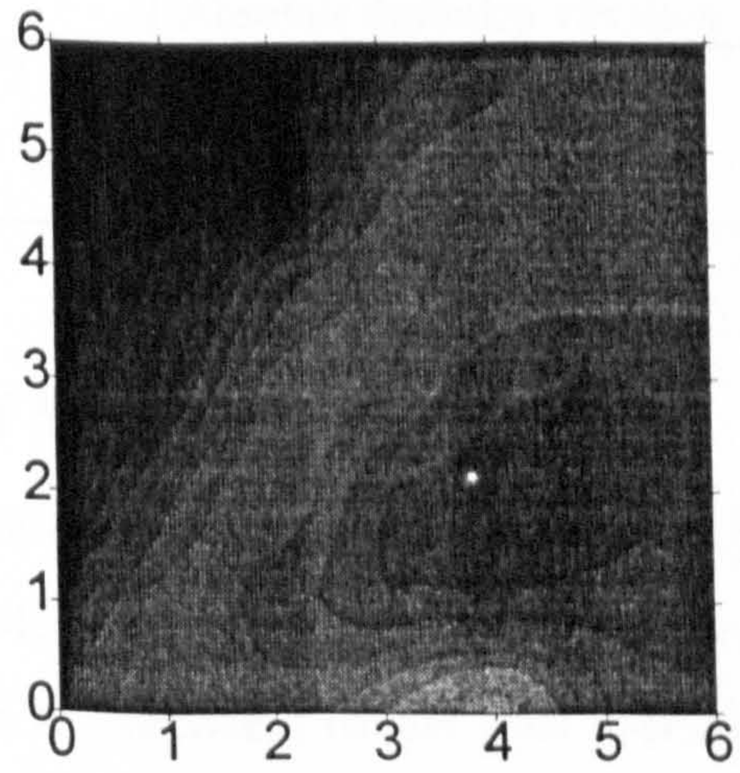


Figure 101: Feature visualisation: Spindle current

The feed and tangential forces show a linear increase with tool wear, and their feature-associated weights arrange themselves so as to increase with increased force and therefore their

activation is always directly linked to the evolution of wear. This behaviour is replicated in the way the map of weights is organised (Figure 100).

The absolute deviation, average, skewness and kurtosis of both sound and vibration did not reveal any obvious relationship with wear, as seen in Chapter 5. Therefore it is expected that it will be difficult to associate different wear levels to the different shades. Nevertheless, the different shades seen on their maps suggest that these contribute strongly towards feature vector identification, otherwise a single shade would cover the entire map.

According to the criterion described in section 6.3.1 it is possible to rate the features according to their contribution towards feature vector identification. Table 18 ranks the features in the order of their strength.

Table 18: Ranked features by strength

Rank	Feature
1	Feed force
2	Tangential force
3	Absolute deviation of sound Skewness of sound Kurtosis of sound Sound frequency band 1 Sound frequency band 2
4	Spindle current Skewness vibration Vibration frequency band 1 Vibration frequency band 2 Absolute deviation vibration
5	Average sound Average vibration Kurtosis vibration

6.4 Summary of Neural Network results

From the results obtained with the neural networks and Expert System it can be seen that tool wear classification is possible with apparently weak features, these resolve the complex interrelation between features to produce a robust wear classification. It has been shown that improved performance can be obtained by the use of the Taylor tool life equation as an outlier detector, which when conjugated with process history converges to good classification results. Also, a zone of influence has been defined around which the system can perform tool wear monitoring successfully.

7. Discussion of Results

In this chapter the results obtained from processing the sensory data, neural network and Expert System, will be discussed. The discussion is tied up in with the objective of this Thesis which is to devise a system of tool wear monitoring which is sensor-based but which includes an empirical ground to minimise mis-classification and human intervention in its set-up and running. The devised system proved to be very efficient for tool wear classification under a fixed set of cutting conditions, moving away from the reference cutting conditions showed a gradual deterioration in the success of classification. This limitation could be overcome by retraining the neural networks at different cutting conditions, which would aid in the construction of a reference database of neural network weights for different cutting conditions. It is also advisable in future developments to increase the training sample as this should increase accuracy, that is, reduce neural network sensitivity to learn from erroneous data.

To aid the discussion, the chapter is divided into four main sections; the first consists of a study of the feature/sensor relationship with tool wear evolution, the second analyses the effect of cutting condition changes, the third section deals essentially with the study of the neural networks' ability to perform tool wear pattern recognition upon sensor inputs, and the final section presents an overview of the expert system's role in this tool wear monitoring system.

7.1 Tool Wear Versus Sensor Signal Analysis

The different mechanisms of tool wear and, in particular, the way in which it is manifested in cutting mechanics is fundamental to an understanding of sensor signal behaviour and interpretation. The tool behaviour has to be analysed in order to understand the importance of the proposed features for tool wear prediction. One recurring theme that will be highlighted in the following sub-sections, is the relationship of a large number of features to the dynamic behaviour of the tool.

7.1.1 Tool Wear Evolution

The three phases shown in Figure 102 correspond to the three wear stages present. In the first stage the wear rate decreases which is perhaps due to the abrasion of the uneven tool tip, the tool tip then becomes more "smooth" and abrasion reduces slightly. During the second stage of wear the coating is progressively worn away allowing direct contact between the workpiece and

the tool tip substrate (tungsten carbide), which is less resistant to wear; this corresponds to a slowing of the wear rate followed by an increase in the third stage, as denoted from the average wear rate calculated from the data points. The first stage is less critical towards tool wear monitoring, it is during the second stage, and mainly in the transition from the second to the third stage, that the tool has to be monitored more carefully.

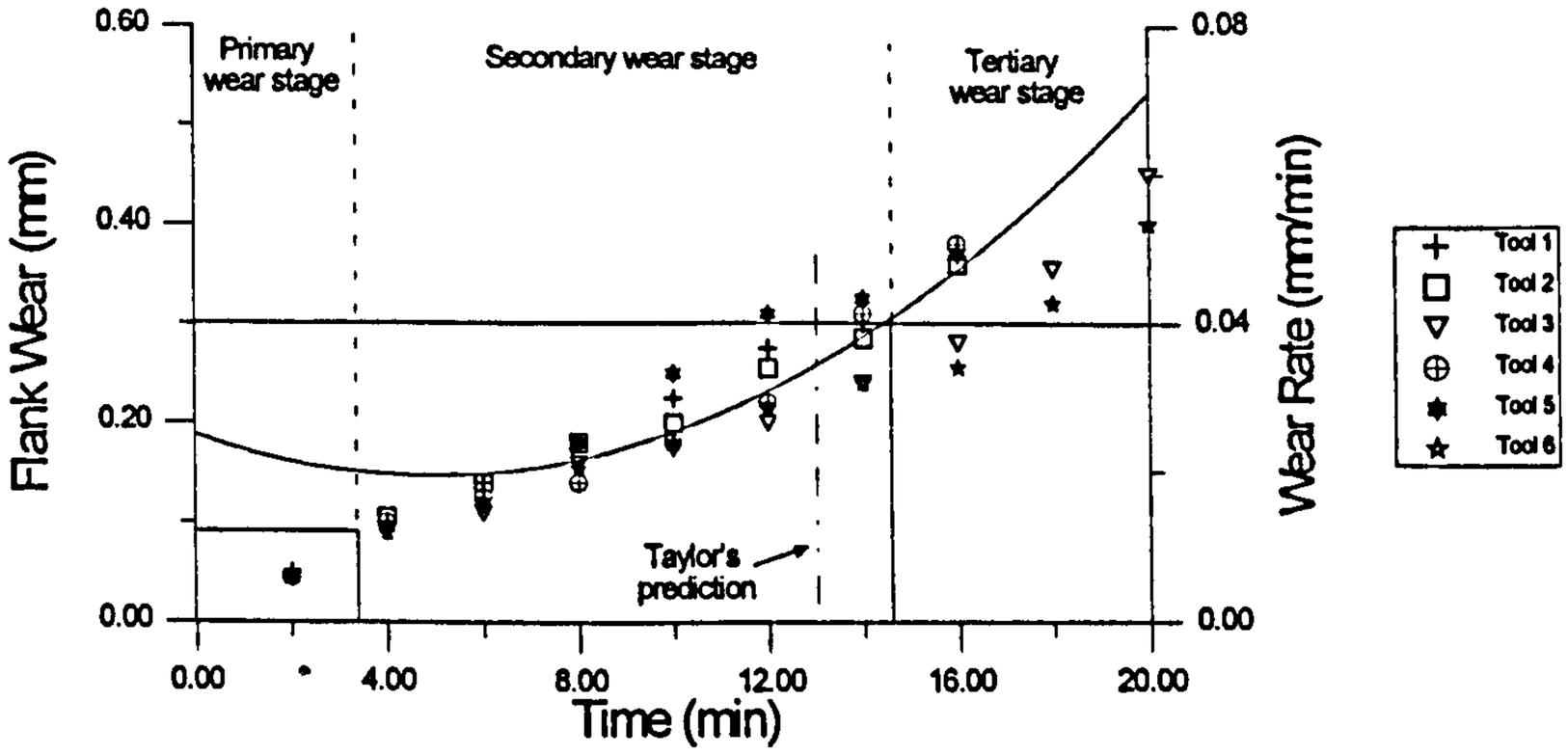


Figure 102: Typical flank wear and wear rate (reference conditions)

Based on the criteria of a tool life limit for VB_B=0.3 mm, the average tool life was calculated to be 14.7 min with a standard deviation of 2.1 min (14% of total tool life). The behaviour of a tool under a specific set of cutting conditions seems reproducible to a certain extent, however the identifications of a limit value for the tool life is always dependent on many factors that may lead the tool to fail prematurely or be underestimated. For example, Tool 5 fails prematurely and Tool 6 is underestimated according to Taylor's tool life equation, in this particular case one of the tools (17%) would fail prematurely and two (33%) would retain 14% of its useful life. In particular, towards the end of a tool's life the wear rate increases so fast that on-line identification is necessary to adequately monitor its state instantaneously.

7.1.2 Changes in Average Cutting Forces Due to Tool Wear

As the tool wears it can be seen that the average feed and tangential forces increase proportionally (Figure 103), accompanied by an increase in tool/workpiece contact area. The tangential force is generally higher than the feed force throughout tool life. This is understandable since higher loads are imposed on the rake face as a consequence of most of the shear strain taking place on this plane. Crater wear did not develop under this set of cutting conditions, the fact that there is no crater wear and the slopes of both forces are similar might lead to the suggestion that the changes are due to friction on both rake and flank faces.

The feed and tangential forces have been shown to be affected in a similar manner with increasing wear. Figure 103 shows that although the feed force suffers a greater percentage increase, higher than the tangential force, both show approximately the same gradient. This leads to the conclusion that both force components are suitable for tool wear monitoring, the feed component being more suitable given its greater percentual increase (more sensitive). The static components of the forces do not seem to be affected by the dynamic characteristics of the machine tool and this enables the use of average forces in the absence of detailed knowledge of the dynamic aspects of machine tool behaviour.

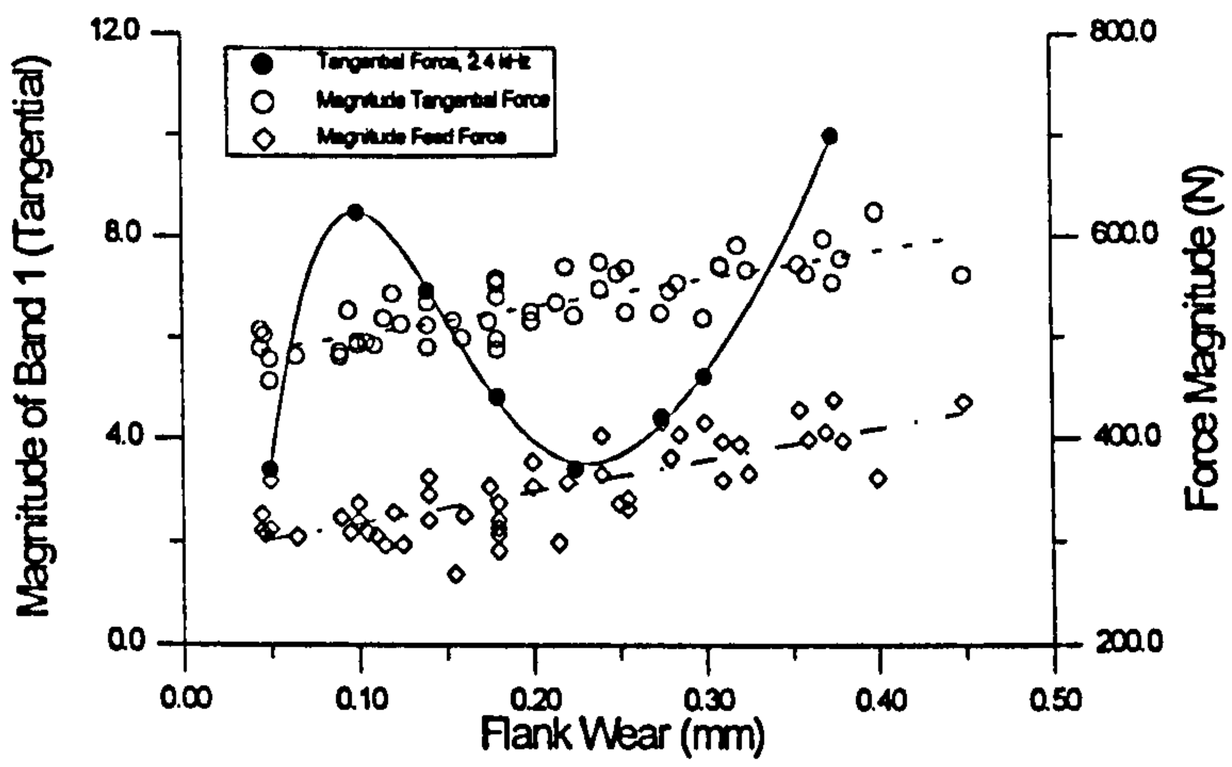


Figure 103: Static and dynamic forces

Whereas the static components of the forces can be isolated from the dynamic behaviour of the cutting tool while cutting, this does not apply to the sound and vibration. Sound and vibration show similar variations in power spectrum to the ones observed in the dynamic behaviour of the cutting forces and vary with dynamic behaviour of the tool.

7.1.3 The Influence of Tool Vibration on the Sensor Space

The vibration of the tool in the feed direction occurs at two distinct frequencies; 3.4 and 6.6 kHz. None of these is equal to the calculated first unconstrained mode of vibration of the tool in the feed direction (9.8 kHz) as the tool is constrained by its interaction with the workpiece. Although significant changes can be identified in the power spectrum at 3.4 kHz this does not seem to have a simple relationship with tool wear.

The power spectrum of the tangential force revealed two distinct peaks, the first occurring at 2.4 kHz and the second at 6.6 kHz. These are not directly related to the calculated natural

frequency of the tool in the tangential direction which was determined to be 7.8 kHz. Tool vibration in the direction of the cutting velocity has a typical evolution with wear which is reproducible throughout inserts, these were obtained from an analysis of the two main characteristic frequencies, 2.3 and 4.5 kHz. The second frequency component on both force spectra, 6.6 kHz, is thought to be associated with torsional vibration of the tool given the torsion effect caused by both forces. Figure 104 shows the variation in magnitude of frequency 6.6 kHz for both force components, these show similar patterns of evolution, the feed component being smaller possibly due to the higher stiffness associated with the tool in this direction. These frequencies did not show repeatability throughout inserts as obtained from the forces spectra, furthermore, they do not appear in the sound and vibration spectra. Because this frequency is not air-born or manifested in the vertical direction may explain this.

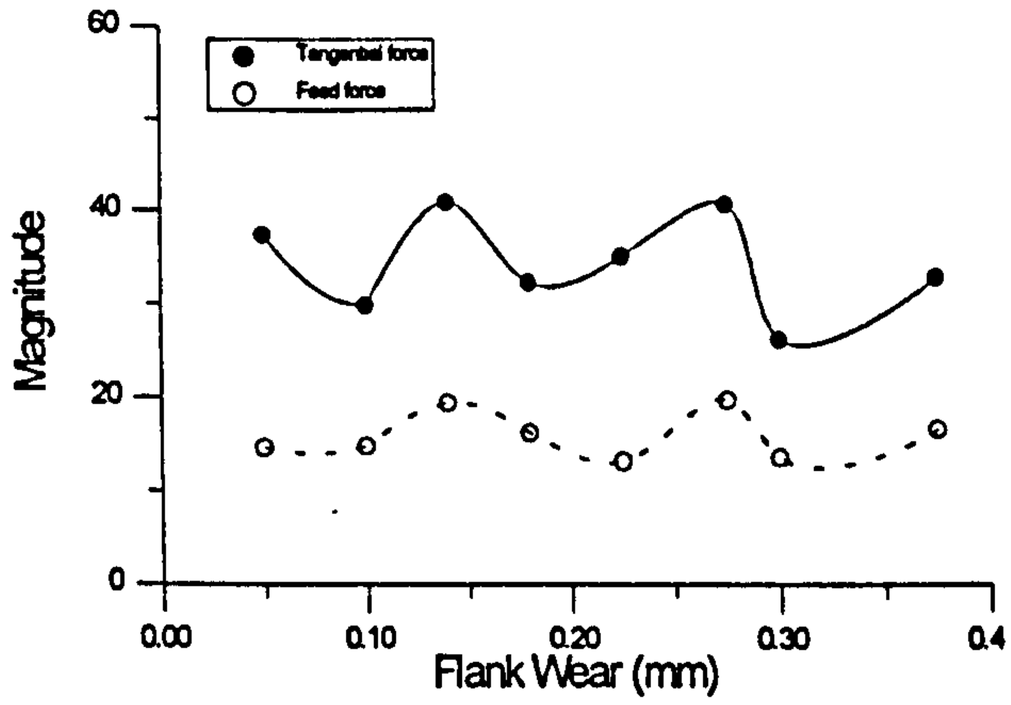


Figure 104: Variation of frequency 6.6 kHz for Tool 1

Given the rectangular section of the tool shank the difference in both force frequencies is understandable, these possibly being related to the natural frequencies of each cutting direction. There is obviously a complex relationship associated with the behaviour of the tool when under load condition which will not be dealt with as it is outside the scope of this work. Nevertheless, end conditions, not taken into account in the theoretical calculations, reduce significantly the value for the natural frequencies of vibration in both directions (Tobias, 1965), also, the tool fixation to the tool holder affects differently the feed and tangential direction. Once again, the latter suggest that frequencies 2.4 and 3.4 kHz are associated with the natural frequencies on the tool. Both factors are due to affect the behaviour of the tool in a complex way, therefore a theoretical model based on cutting process dynamics would prove to be impracticable.

Such self-excited vibrations marked by changes in the power spectra at certain frequencies are normally classified into two types; one due to the flexibility of the cutting tool and the other to

deflection of the main spindle of the lathe or the workpiece. Further, these self-excited vibrations are divided into two groups depending on the circumstances in which they occur, namely primary and regenerative chatter (Marui *et al.*, 1988). Primary chatter takes place during the primary and secondary stages of wear, whereas the regenerative type appears when the tool becomes seriously worn resulting in "chatter" marks. With primary chatter the tool does not loose contact with the workpiece whereas during regenerative chatter the tool frequently looses contact with the workpiece generating undulations on the surface of the work. Primary chatter and regenerative chatter can be observed in Figure 105. The following results prove that this behaviour is not a sporadic one, but a characteristic of the turning processes.

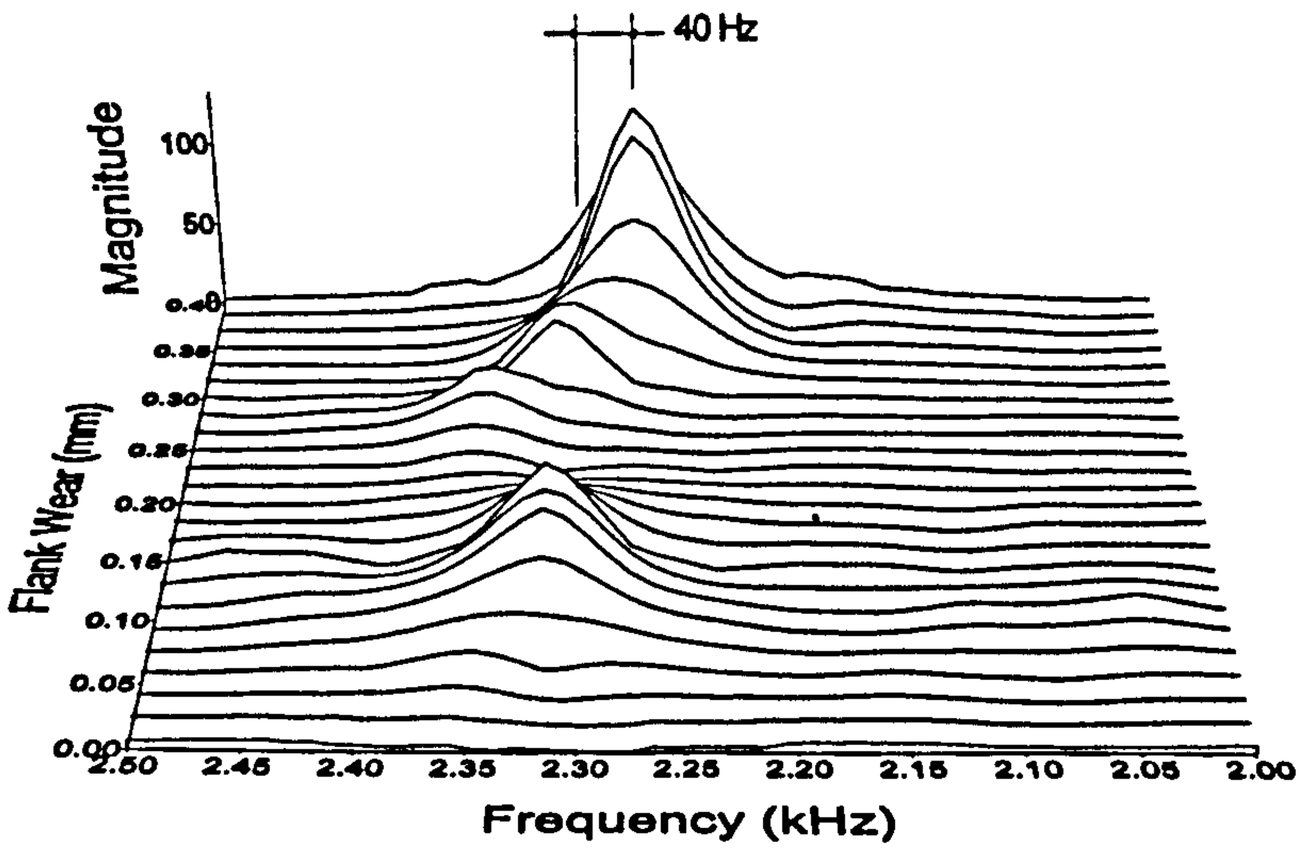


Figure 105: Evidence of two types of chatter, from tangential force spectra

The regenerative chatter frequency is nearly equal to the natural frequency of the system in the tangential direction (Figure 105), and is a little lower (40 Hz) than the frequency of the primary chatter vibration (2.3 kHz). This occurs possibly because a higher load is imposed on the cutting tool as it becomes dull, also, the regenerative effect becomes evident by the "chatter marks" left on the workpiece. The drop in frequency is possibly due to the increased load which causes the tool to become more stiff, moreover, the fact that the tool looses contact with the workpiece makes the problem non-linear and therefore more difficult to interpret using conventional formulae.

The following two figures (Figure 106 and Figure 107) show the comparative evaluation of sound, vibration and dynamic force spectra behaviour. These show that the tangential force dynamic behaviour is closely related to that of the sound and vibration sensors as the tool wears for the frequency band 2.3±0.1 kHz.

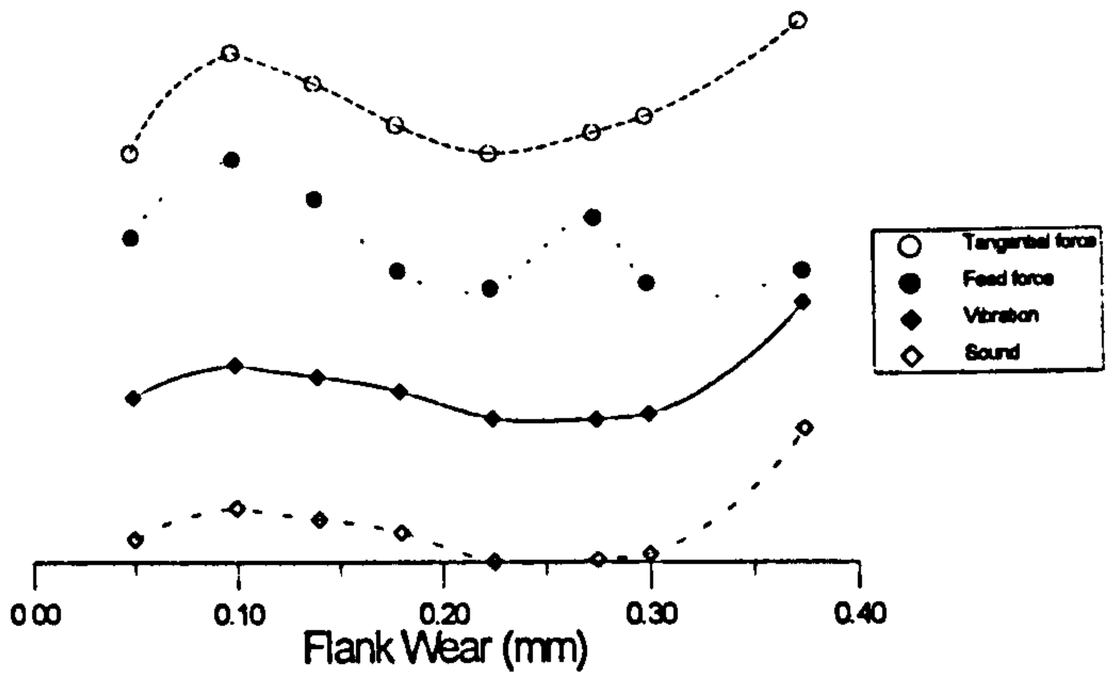


Figure 106: Magnitude variation of frequency 2.3±0.1 kHz (relative scales)

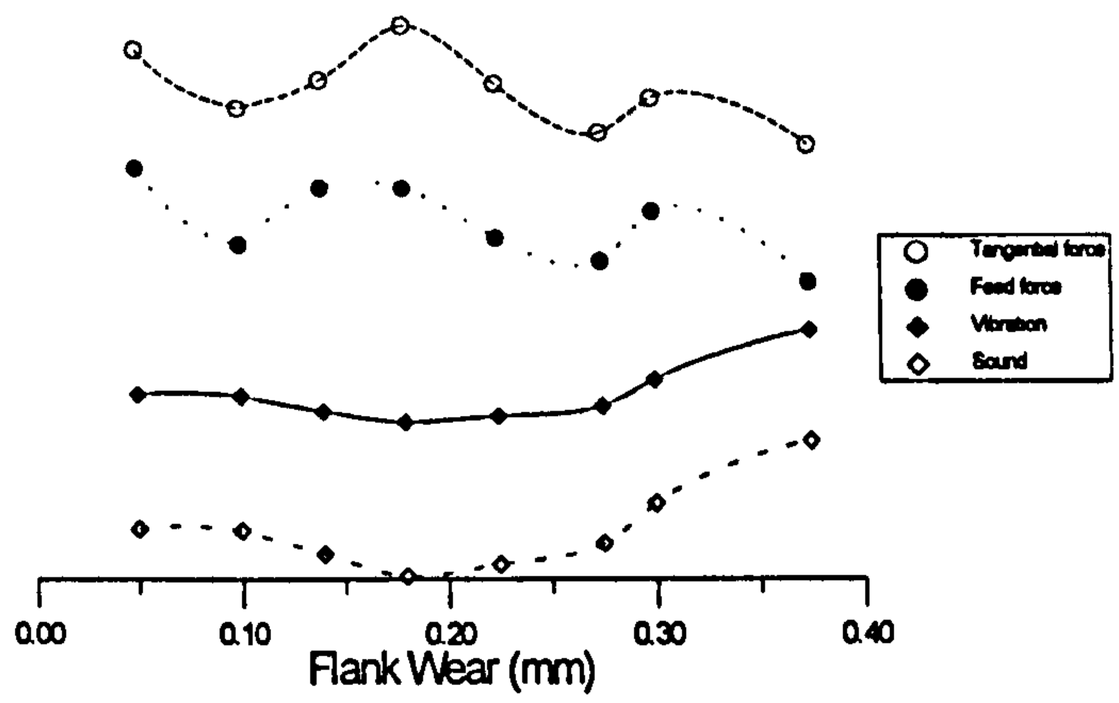


Figure 107: Magnitude variation of frequency 4.5±0.1 kHz (relative scales)

The sensed signals show a characteristic behaviour around 2.3 kHz (frequency band 1), except for the feed force component. The latter tends to increase during the first stage of wear becoming smaller during the second stage of wear, and finally increasing sharply as the tool reaches the end of its life. It is notable that both the feed force and its superimposed dynamic components are significantly smaller than both the magnitude and dynamic components of the tangential force, this is partially due to increased stiffness in the feed direction (higher second moment of area), nevertheless it does not justify it being much smaller than the torsional vibration experienced by similar strain gauges. The reduced magnitude associated with higher stiffness may contribute to the apparent inconsistency of the dynamic behaviour of the feed force frequency band 1.

Examining the vertical vibration of the machine tool, as sensed by the accelerometer positioned on the base of the machine, it can be seen that vibration evolution of the tool in band 1 is related to the tangential force dynamic behaviour. As might be expected, the vibration of the tool originate from small changes in depth of cut which resulted in a variation of the thrust force, these variations are transmitted to the machine in the vertical direction due to the

orientation of the cutting tool. Also, according to Arnold (1946), the magnitude of this influence is dependent on the extent of wear on the flank face. A similar behaviour would be expected in the feed direction since, as shown by the feed force, there are vibrations occurring in the feed direction.

Magnitude of frequency band 2 (4.5±0.1 kHz) is significantly smaller then band 1 (2.3±0.1 kHz), in the order of 4 times smaller, also it does not appear to be associated directly with the cutting forces (Figure 107). Since the force spectrum is expected to reflect the dynamic characteristics of the cutting process whereas the vibrations, and eventually the sound, are a response to the forcing function, this possibly means that frequency band 2 has more to do with machine flexibility than the dynamics of the cutting process.

The dynamic cutting force induces vibration on the relevant parts of the machine tool therefore the cutting force should be closely related to the cutting noise. Figure 106 shows the evolution of the cutting force frequency band 1 of vibration and the evolution of the power spectrum of sound at the same frequency. If only the shapes are considered, the two lines are extremely alike. Thus, for the cutting force, cutting noise and vibration, not only are their spectral shapes alike, but the spectral level evolution with wear corresponds to each other. Again, as for the vibration, frequency band 2 is indistinguishable from the force spectrum possible because this arises indirectly due to machine flexibility and not directly from the cutting process.

As the tool gradually wears, the flank surface becomes larger, causing the contact surface between tool and workpiece to increase, and resulting in higher frictional forces. This causes vibration of the tool which, in turn, produces sonic vibrational waves that are transmitted through the air. The sound picked up by the microphone arises not only from the tool vibration, but also from the vibration of the lathe, although the latter is substantially weakened because of the orientation of the microphone. Because of the higher magnitude of vibration in the tangential direction and lower tool stiffness, it would be expected that the greater part of the sound picked up by the microphone would be associated with this component of vibration.

In summary, it has been shown that the dynamic behaviour of the tangential force is closely related to the behaviour of the sound and vibration as the tool wears and, this is probably due to the fact that the main source of sound is the vibration of the cutting tool under dynamic cutting forces.

7.1.4 Limitations of Spindle Current as a Condition Monitoring Sensor

Motor torque is intrinsically correlated with the cutting process. The spindle current is in turn related to the drive RPM, but this is complicated slightly for the turning process since, as the diameter reduces the RPM of the workpiece has to be increased to keep the cutting speed constant and this causes quite dramatic changes in spindle current. In order to maintain the cutting speed and feed rate constant the rotational speed has to be changed so that the rate of metal removal remains constant, the spindle speed will vary according to the equation,

$$v = 2\pi R\omega\,, \qquad [6]$$

where v is the tangential speed of a point at a radius r with an angular speed ω. So, as the radius reduces the angular velocity (or RPM) will have to increase to keep the cutting speed constant. The above is confirmed by the experimental results obtained. The points labelled "New bar" in **Figure** represent readings taken at the beginning of a new bar (maximum radius), the straight line shows the increase in spindle current with a decrease in workpiece diameter for a single bar. As can be seen, the spindle current is more influenced by the motor speed (RPM) than by the level of wear on the tool. For insert 2, the spindle current increases 4% due to an increase of tool wear of 0.19 mm, while it increases 30% due mainly to the difference in workpiece diameter for an increase of tool wear of 0.14 mm.

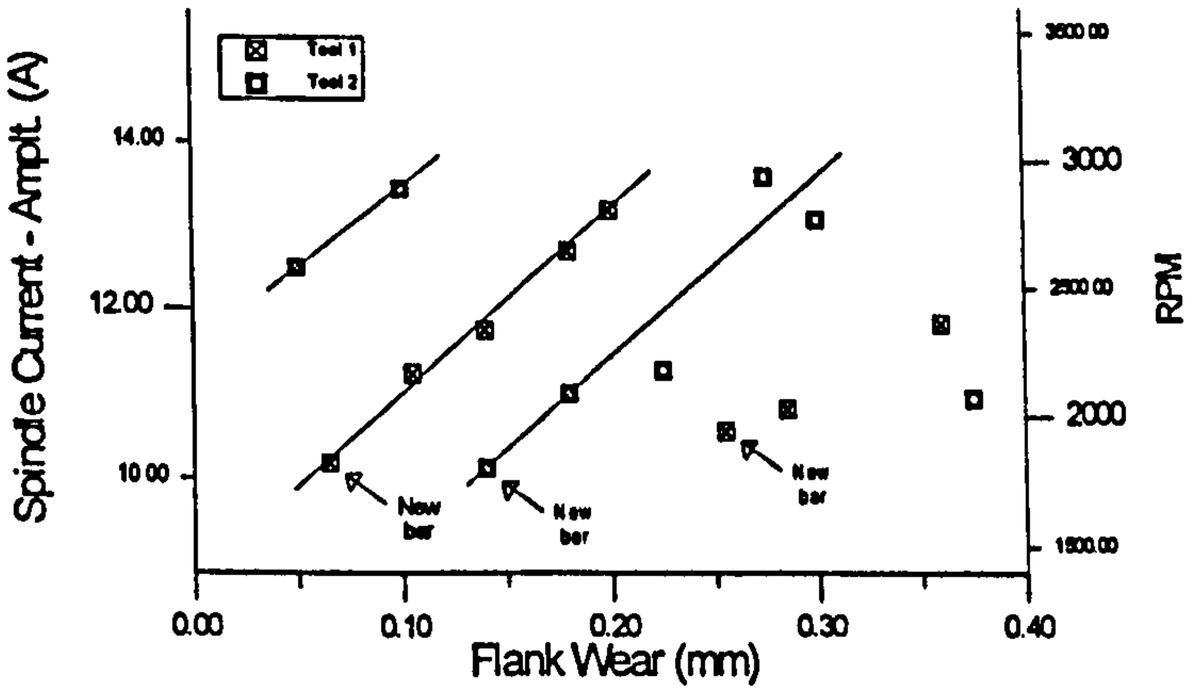


Figure 108: Spindle current versus flank wear

On its own the spindle current adds little towards the task of monitoring tool wear of the turning tool. Nevertheless, since the power provided by the machine is constant (Chapter 3) the angular velocity may help to unravel the previous limitations, $P = T.\omega$. If the angular speed increases as the workpiece diameter decreases the torque has to increase, therefore the torque will rise and consequently the current to drive it. Having this influence in mind and making use

of the workpiece diameter it would be possible to use the spindle current as a condition monitoring sensor.

7.1.5 The Effect of Workpiece Diameter on Sensor Features

Given that the spindle current has the behaviour described previously, and that the torque changes with workpiece diameter, raises the question of whether any of the other features are influenced by bar diameter. Since the torque increases it would be expected for some changes in other features reflecting the machine behaviour along with changes due to tool wear to occur. To answer this question a comparative analysis will be given between spindle current, given its direct relationship with bar size, and the other features. Figure 109 to Figure 116 show a comparative analysis of this effect.

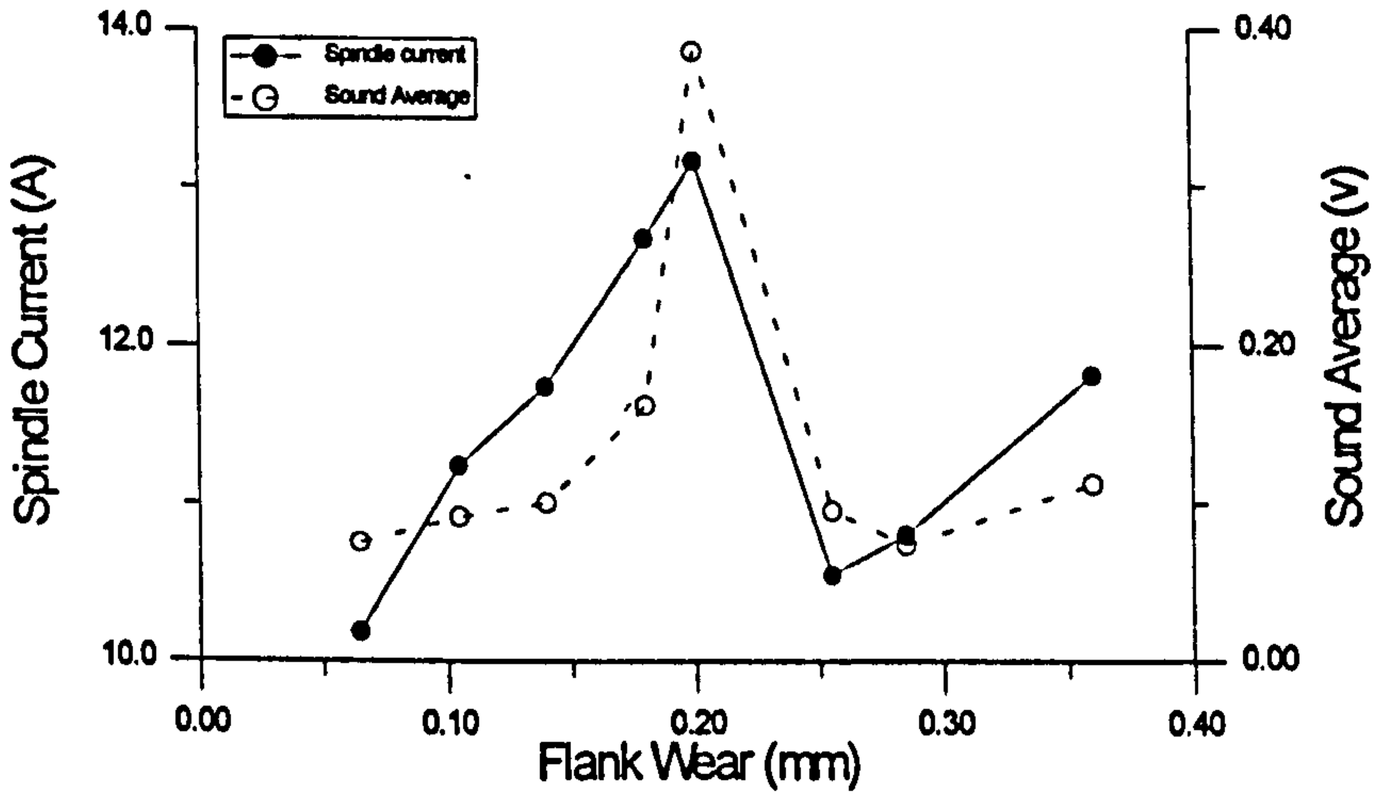


Figure 109: Bar size versus sound average

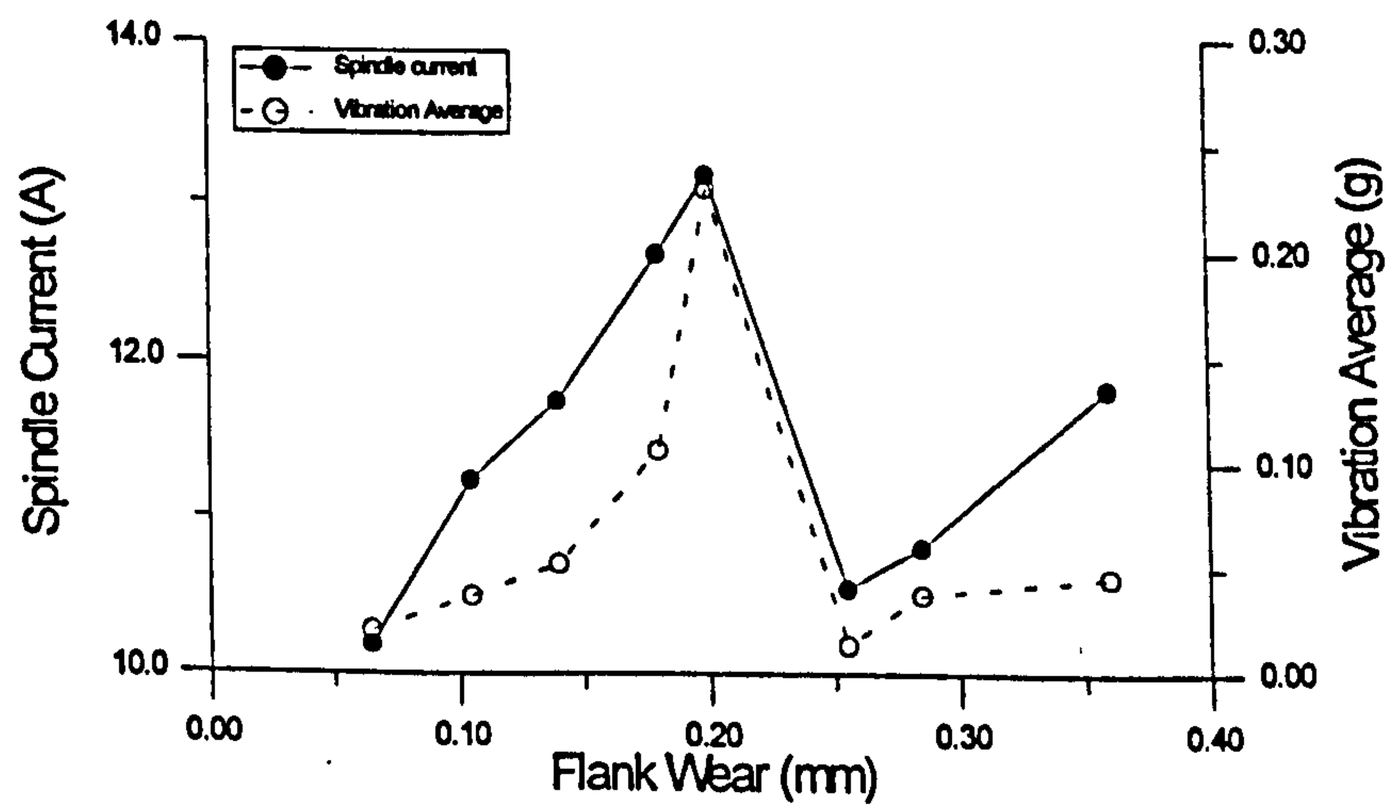


Figure 110: Bar size versus vibration average

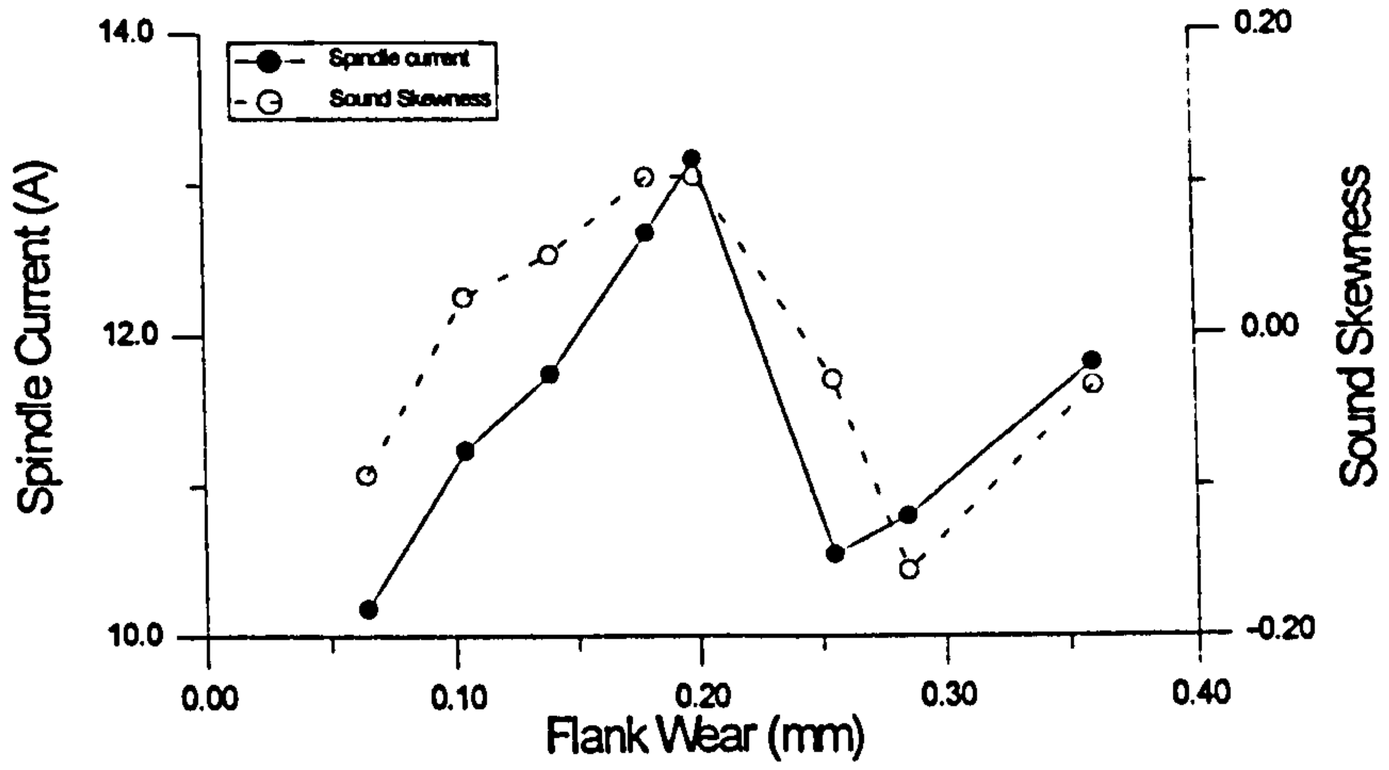


Figure 111: Bar size versus sound skewness

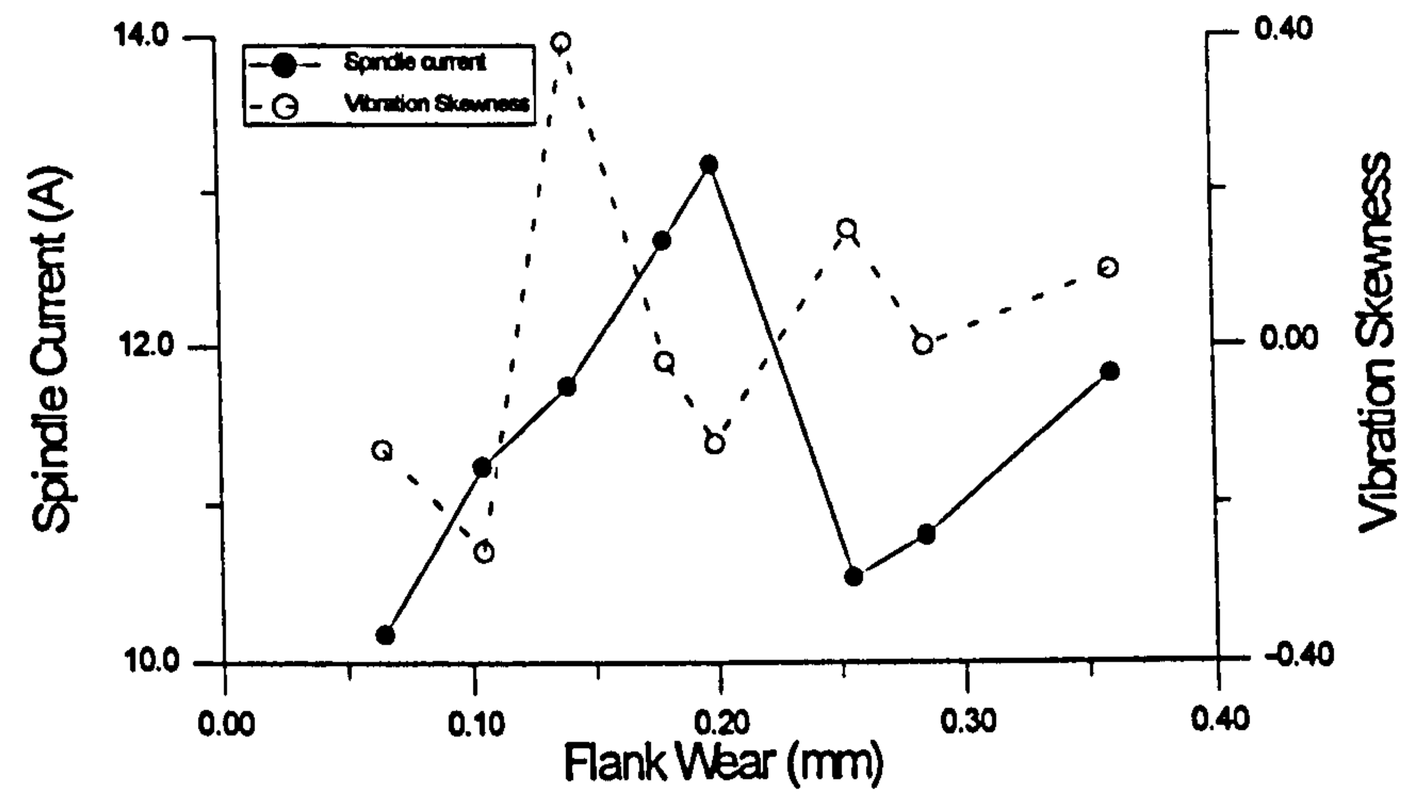


Figure 112: Bar size versus vibration skewness

Despite the wide scatter shown in Chapter 5 for the average and skewness of sound and vibration relative to flank wear evolution, Figure 109 to Figure 112 show an obvious relationship to the spindle current, and indirectly to the size of the workpiece, which is consistent throughout inserts. The average reveals a strong correlation with bar size, this is possibly associated with an increase of RPM which in turn causes the machine related sound pitch to rise, as observed. The skewness is also affected by the bar size giving rise to an increase with reduced workpiece diameter, this is better illustrated for the skewness of sound (Figure 111). An increase in the skewness value means that the distribution becomes more asymmetric leading to a positive tailed distribution, this is possibly due to an increase in RPM which leads to an increased number of random impacts per sample. The absolute deviation and kurtosis did not reveal any obvious relationship with spindle current although some complex interaction may exist between these and other variables inherent to the cutting process.

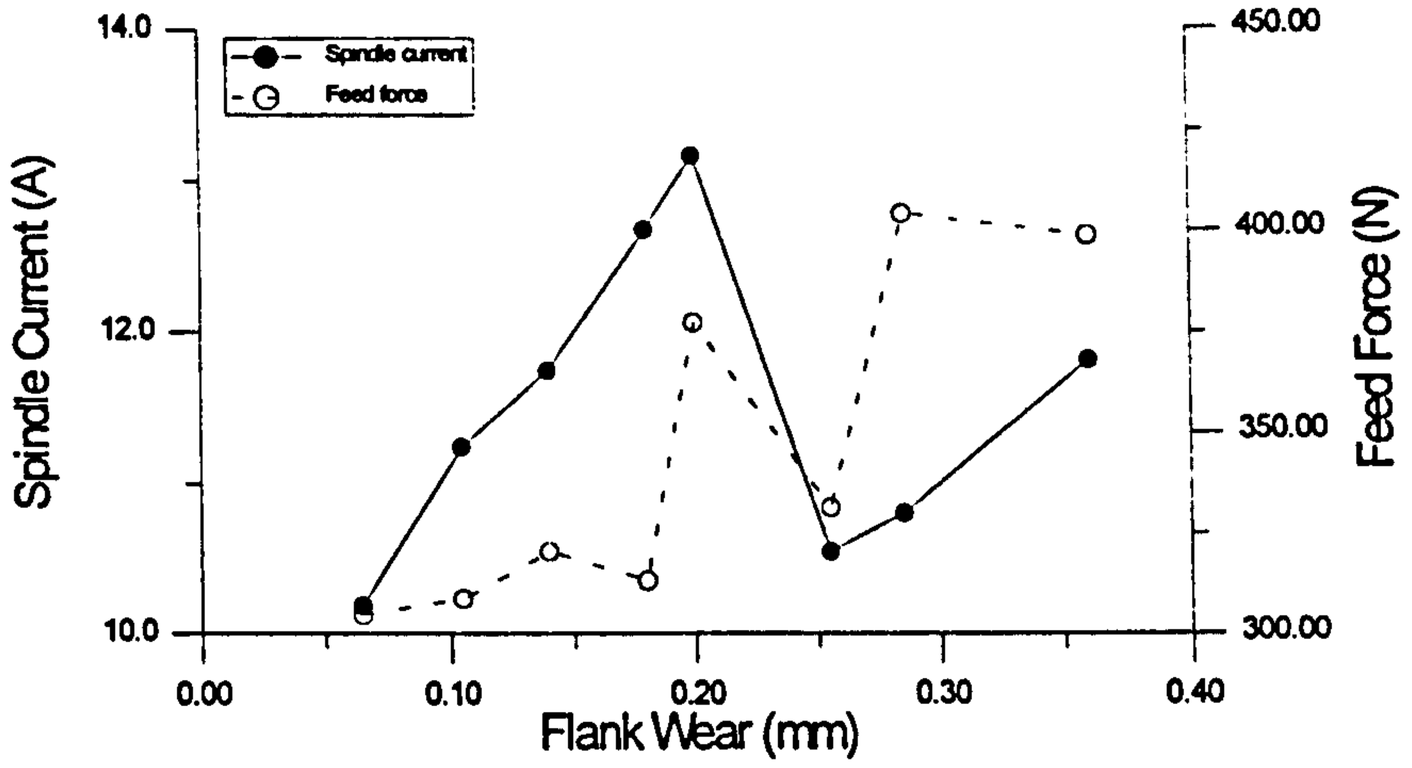


Figure 113: Bar size versus feed force

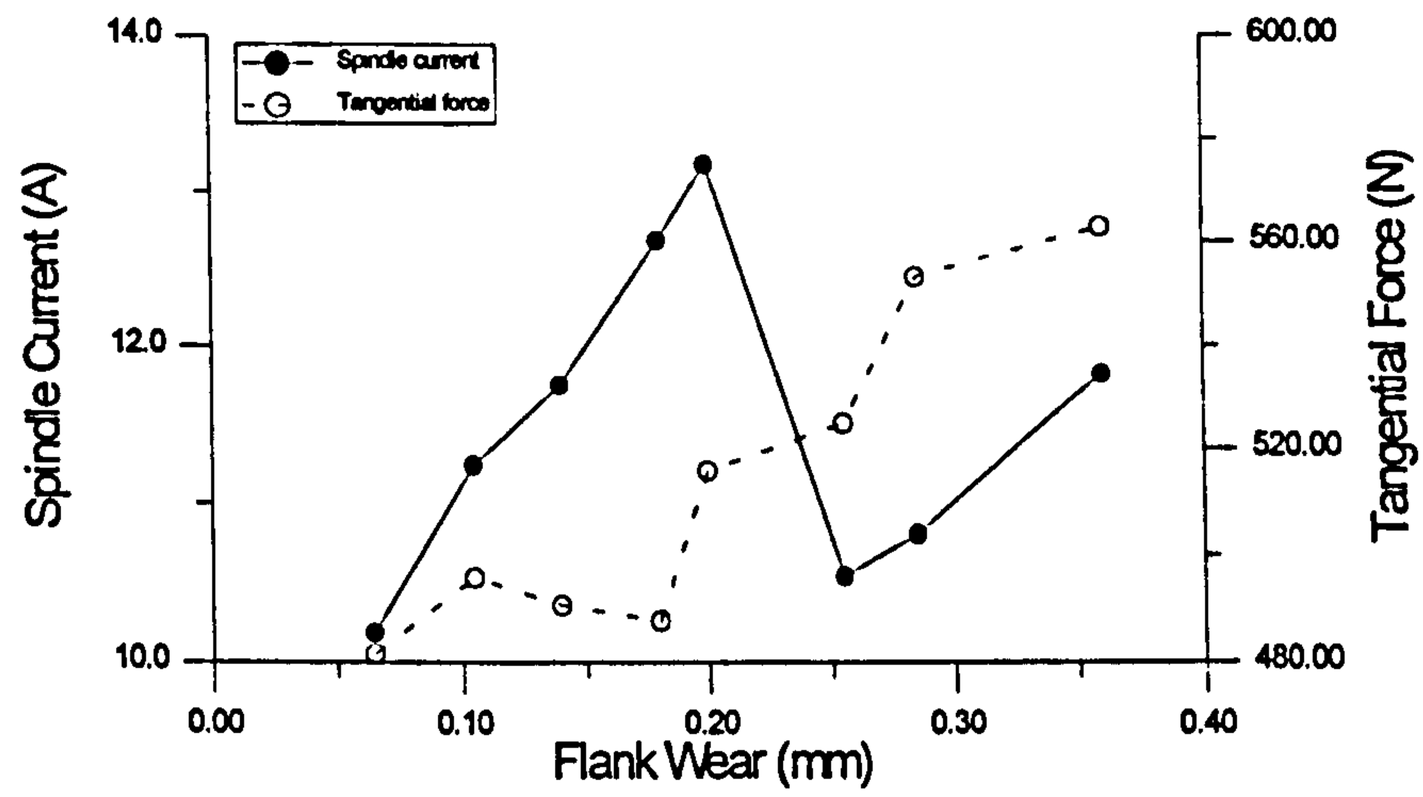


Figure 114: Bar size versus tangential force

Although a general increase could be observed in the cutting forces with flank wear (Chapter 5), some relation with spindle current can be observed (Figure 114 and Figure 115). This would possibly account for some part of the scatter observed in the evolution of the forces with flank wear. Again, these figures are representative of the entire set of cutting tools.

Figure 115 and Figure 116 plot the evolution of both frequency bands used as features. Apparently no straightforward relationship exists between the dynamics of the process and the bar size. Eventually, the dynamic behaviour would be affected by changes in workpiece size due to changes in mass which can not be confirmed with the present data. It is also of note that the weight of the workpiece (≈ 6 kg) only accounted for a small part of the total weight of the combined workpiece/tailstock/spindle system. Thus, it seems that the vibrations are more likely to come from the cutting tool/machine dynamic characteristics than from the workpiece. Also, even under conditions of relatively stable machining, the presence of vibration is manifested by saw-tooth marks on the workpiece and serrations on the metal chip (Lee *et al.*, 1989).

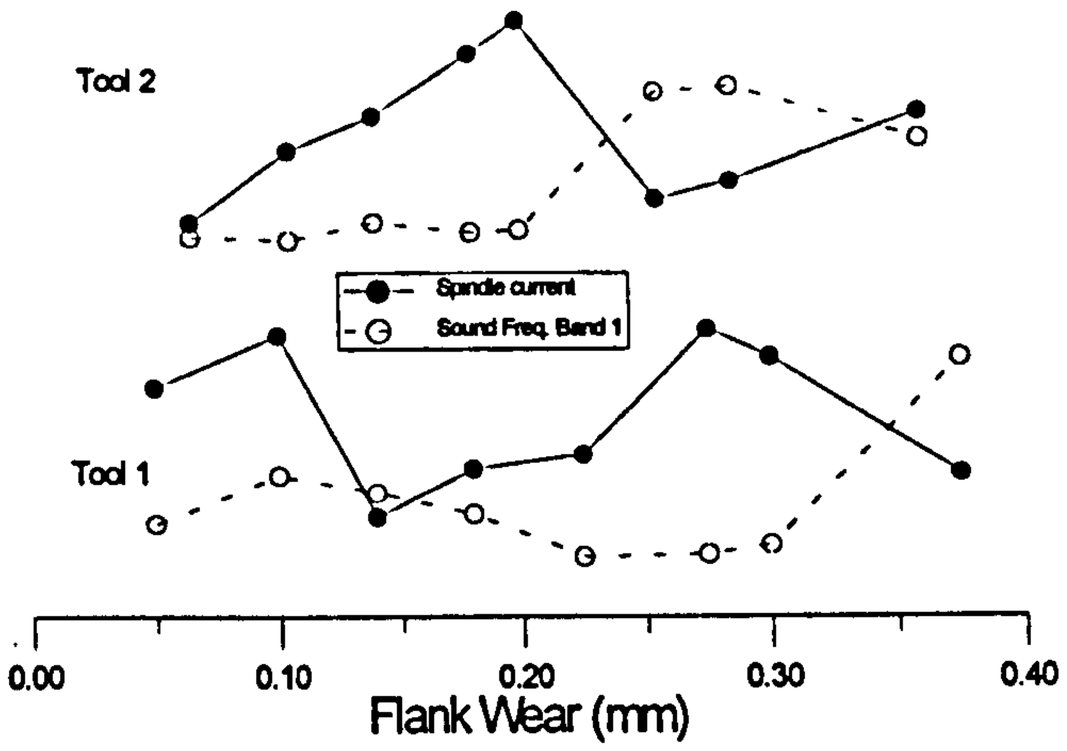


Figure 115: Bar size versus sound frequency 2.3±0.1 kHz

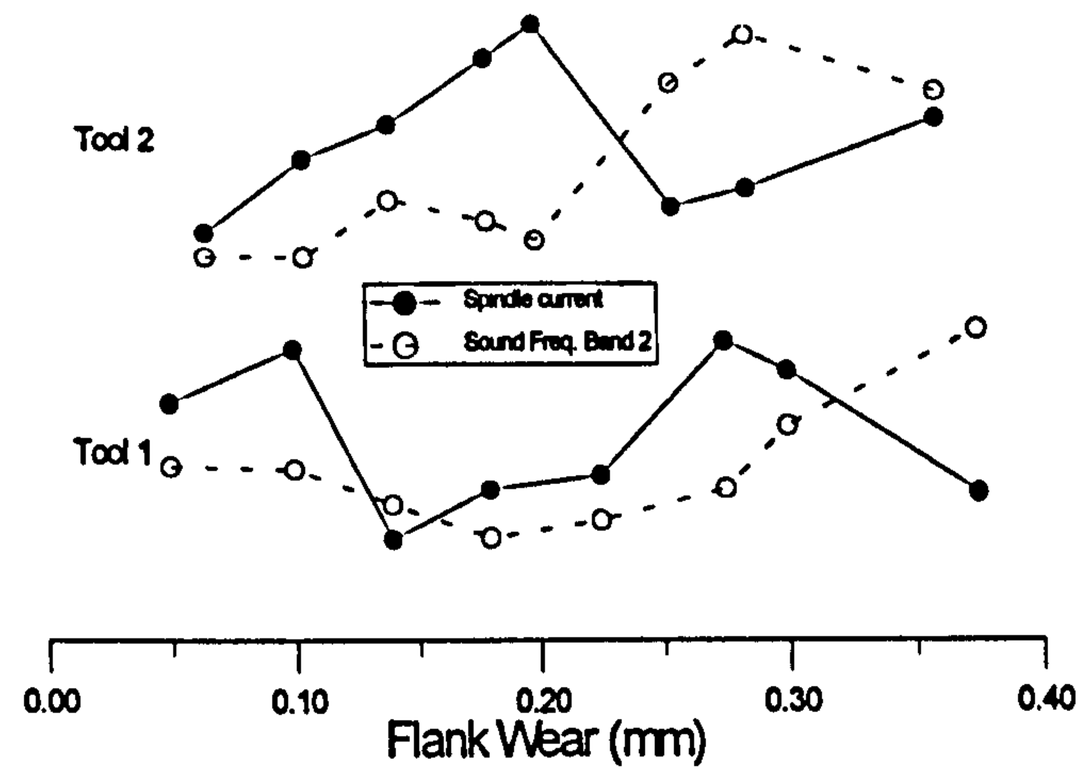


Figure 116: Bar size versus sound frequency 4.5±0.1 kHz

7.1.6 Overall Sensor Analysis

It can be concluded from the previous discussion that wear monitoring using indirect sensing methods such as vibration, sound emission and cutting force components is dependent on the vibrational behaviour of the machine tool. Also, workpiece dimension constitutes another variable subject to machine constraints which may be overcome since the working diameter is known at all times. This results in a monitoring system that is highly tuned to a particular machine and even perhaps to a particular range of cutting conditions, tool and workpiece material. A methodology for coping with some of these limitations will be discussed later in this chapter, whereas the following section will quantify the effect of cutting conditions.

7.2 The Influence Of Cutting Conditions on Sensor Features

This section discusses the effect of changing the cutting conditions on the sensors and derived features used here. The effect of changing the cutting parameters (depth of cut, feed rate and cutting speed) will be analysed taking each of the cutting parameters separately. It is important to note that the results in this section are based on two samples taken at the same workpiece diameter, therefore and according to what was discussed in section 7.1.5, the effect of depth of cut is eliminated which leads to more representative values for comparison.

7.2.1 The Effect of Depth of Cut on Sensor/Feature

If the depth of cut is increased, the area of the chip-tool contact will increase approximately in proportion to the change in depth of cut, and this causes the components of cutting force to increase. It might be possible to remove the effect of depth of cut by use of a model, but most models are very specific and usually only estimate cutting force for a new tool and neglect the rate at which it increases with wear. It would be simpler to normalise the forces dividing it by the depth of cut.

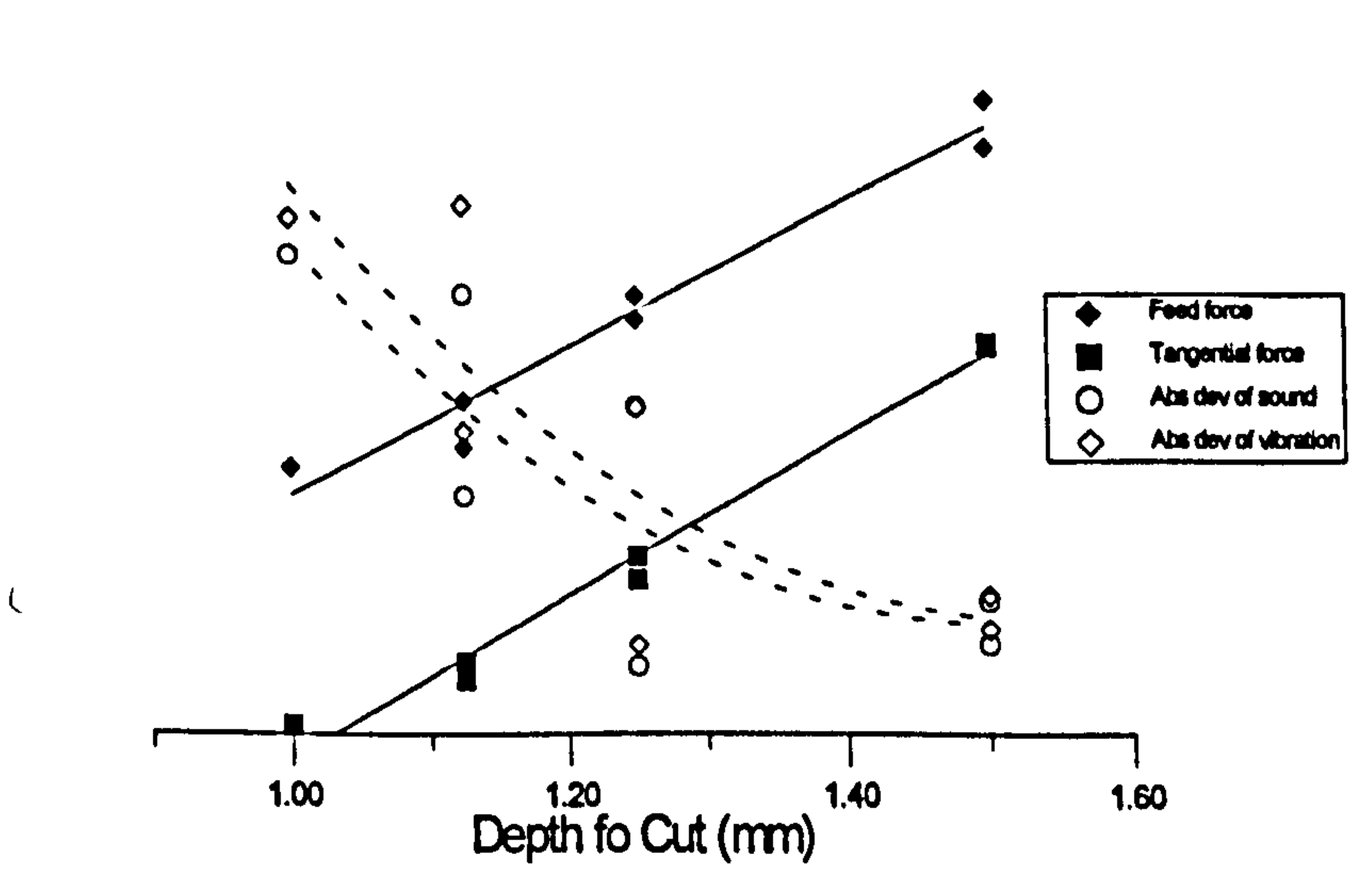


Figure 117: Effect of depth of cut on force magnitude and absolute deviation of sound and vibration, worn tool (relative scales)

For machine vibration, sound and dynamic components of force it appears that the vibrational modes of the machine tool dominate so that different forcing functions (from different depths of cut) result in different evolutions. However, as the depth of cut increases the vibration appears to reduce, as shown by the evolution of both the absolute deviation and frequency spectra of sound and vibration (Figure 117). This effect is possibly due to an increase of the cutting forces leading to increased stability of the cutting process.

7.2.2 The Effect of Feed Rate on Sensor/Feature

A feed rate increase results in higher metal removal rate, higher forces at the tool rake as well as an increase in compressive stress thus giving rise to an increase in the cutting forces. The force gradient changes slightly at a feed rate of around 0.275 mm/rev for both tangential and feed forces. At this feed rate the tool appears to gain stability (Figure 118) which considerably contributes to a reduction of the wear rate and which reflects the slight reduction of both feed and tangential forces. This linear increase possibly reflects the increase in metal removal rate.

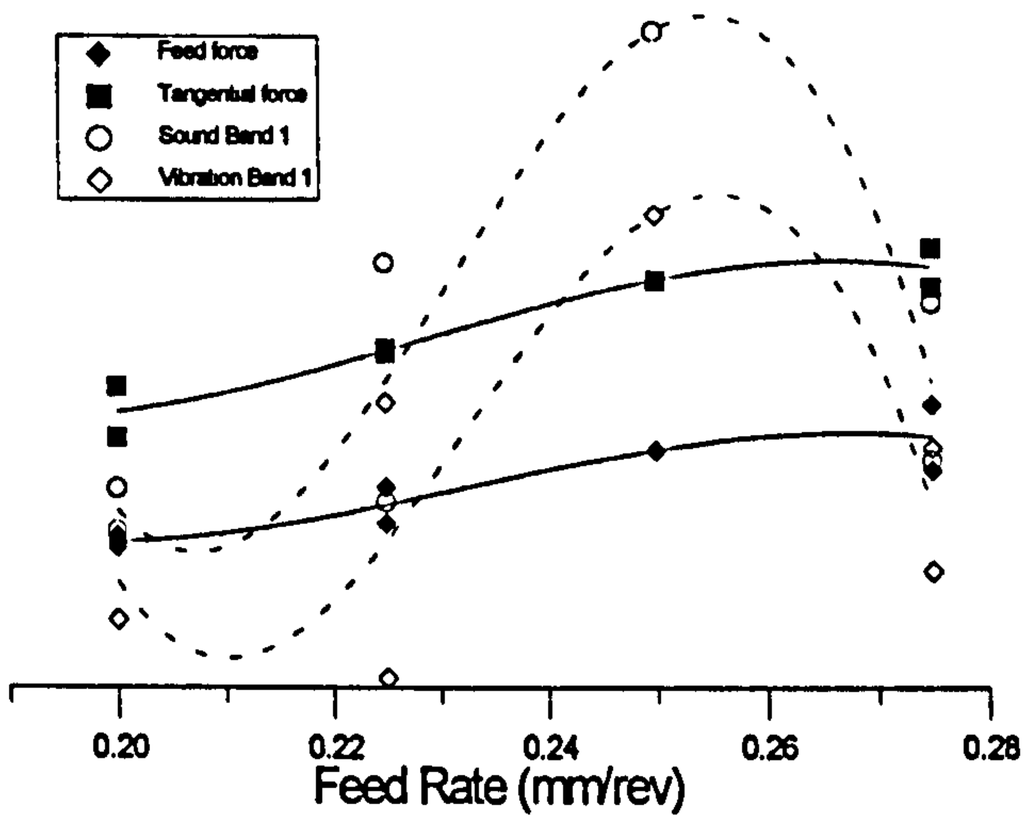


Figure 118: Force magnitude and frequency 2.3±0.1 kHz, worn tool

All sensors seem to be affected in different ways with changes in the feed rate. The spectra of both sound and vibration have shown non-monotonic relationships with tool wear as well as the other statistical features in general.

The change in behaviour of the force as well as the other features at a feed rate of 0.275 mm/rev may be due to a change in metal removal rate (Figure 119). An increase in the feed rate eventually leads to what is seen in Figure 119, part of the material supposed to be removed is left on the workpiece. With a further increase of the feed rate it would be expected for the forces to remain constant, with a gradual reduction in the gradient of the force increase with feed rate. This effect happens around this particular point because for a feed value higher then 0.3 mm per revolution, the same as the flank wear for a worn tool, $VB_B = 0.3$ mm, the tool is not able to cover the area intended by increasing the feed per revolution. This also explains why *new* tools are affected, by this effect, prior to the *worn* tools.

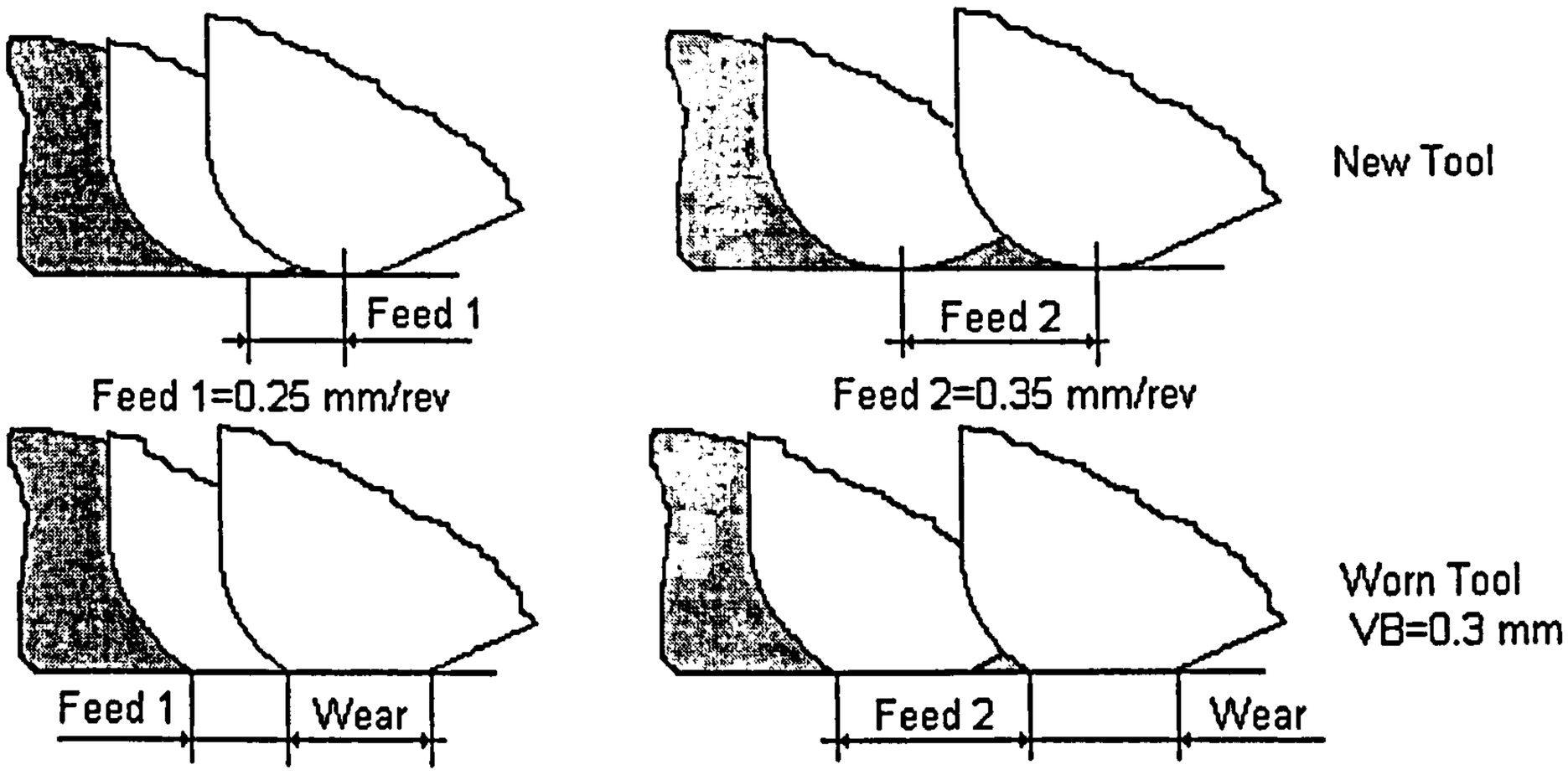


Figure 119: Material removal rate changes with increased feed rate (not to scale)

The above effect is responsible for most of the behaviour associated to a reduction in the expected force values. Possible it has an effect on other features, however these are not perceptible to such an extent.

7.2.3 The Effect of Cutting Speed on Sensor/Feature

Changes in the average cutting forces due to cutting speed seem to be affected differently with different cutting speeds, showing no obvious evolution with cutting speed. However, the dynamic changes with cutting speed show a clearer evolution. The absolute deviation of sound and vibration signals as well as changes in their first frequency band have good correlation with cutting speeds and seem to increase. It is of note that for frequency band 2 only the vibrational signal was affected significantly, increasing with the cutting speed. Again, frequency band 2 is possibly reflecting the changes in flexibility, more with the machine than with the cutting process directly.

7.2.4 Overall Assessment of the Effect of Varying Cutting Conditions

It has been demonstrated that the vibrational characteristics of the machine/tool affect the performance of features, as their vibrational characteristics are influenced by the cutting conditions and tool shank properties (e.g. dimension, tool post stiffness). In many cases the changes in features resulting from changing cutting conditions were much larger then those occurring with wear making it difficult to define a set of signal processing routines that would isolate the tool wear changes independently. This has significant implications for a tool wear

monitoring system probably resulting in a monitoring system that is specific to a small working set of conditions. However, this might not be as limiting as one might expect since many flexible cells use only a small set of conditions.

To overcome some of the limitations necessary to extend the adaptability of the system to cutting condition variation it would be worth implementing the following suggestions:-

- Normalise the cutting forces, e.g. feed force normalised for depth of cut.
- Account for workpiece diameter, which is readily available from the CNC machine.
- Care in the use of feed rate values as for large feed rates the behaviour of the forces is due to change (as it would be recommended, depending on the insert radius a limit value for the feed rate is recommended).

The Expert System can be adapted to cope with the above suggestions by making it aware of cutting conditions and workpiece size. Otherwise adaptability could be built into the system to the extent of detecting this changes by itself.

7.3 Generalisation Capabilities of the Neural Networks

In this section the capabilities of both neural networks to perform tool wear classification are discussed. Several aspects will be raised, which address the unsupervised character of the neural networks along with training issues.

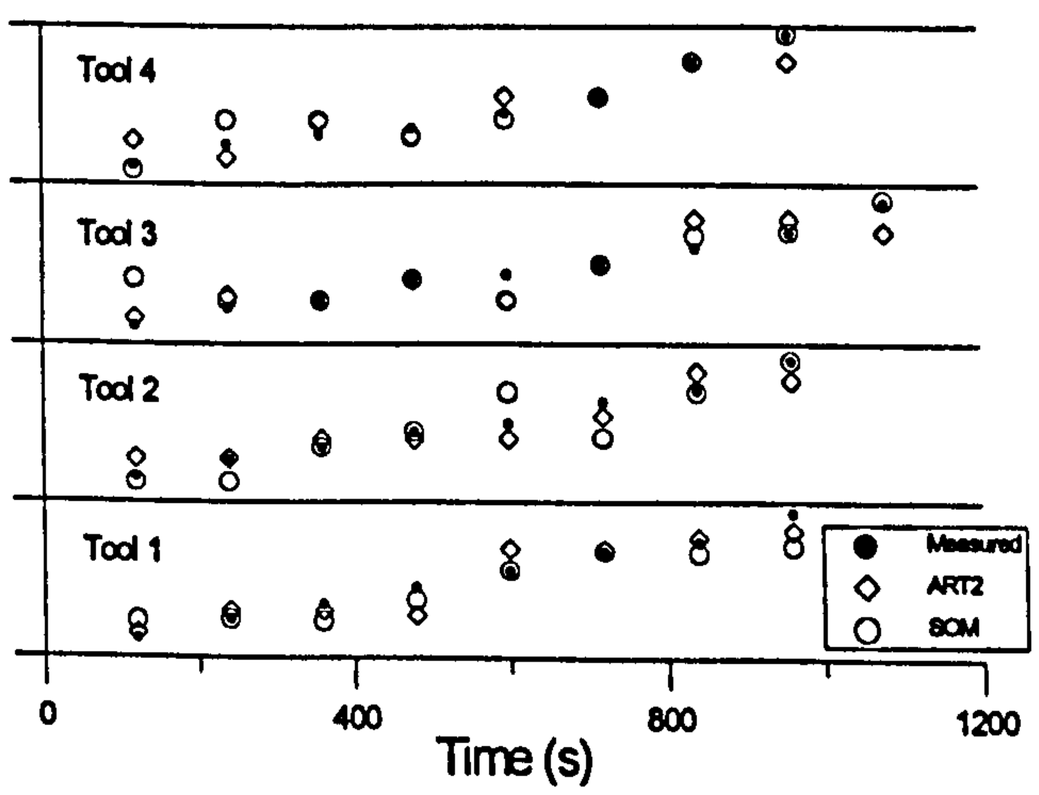


Figure 120: Neural networks performance for training data

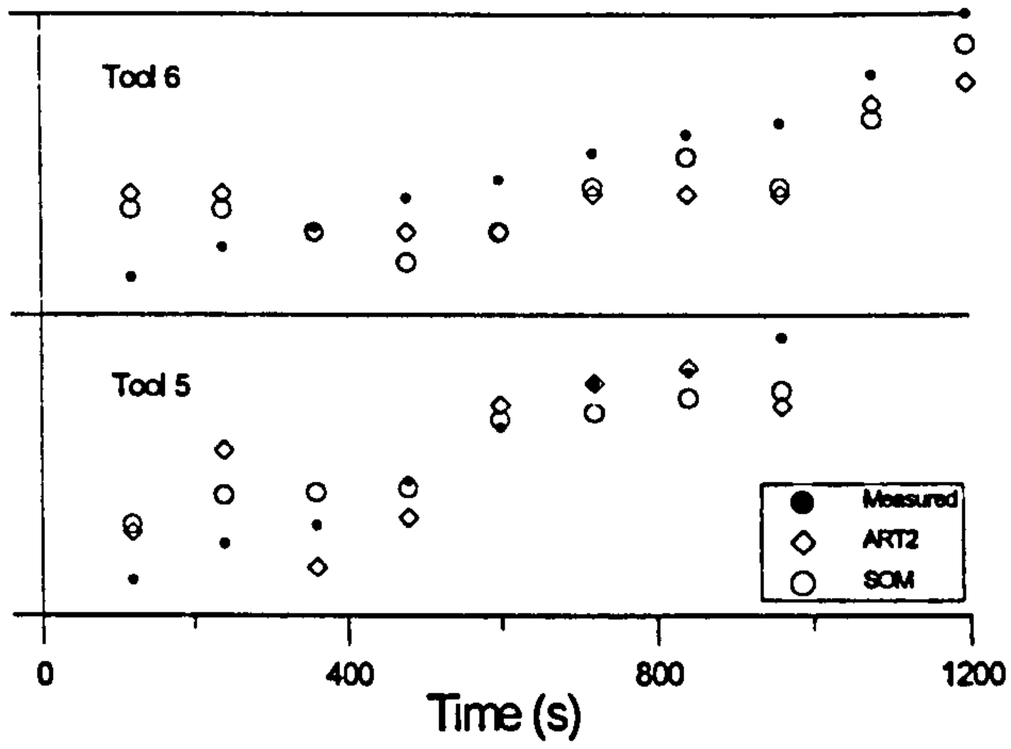


Figure 121: Neural network performance for test data

Figure 120 and Figure 121 demonstrates that both the self organising map and ART2, each acting alone, have a large capacity to categorise the different wear stages. The training period had a large effect on the performance of the SOM and more training time was required for the SOM than the ART2, although at the interpretation stage both have the same speed since the basic calculations are relatively simple.

Of the two networks the SOM, comparatively to the ART2, was better able to extract the complex relationship between tool wear and the selected features, it was less prone to the influence of noise and was able to generalise more completely. This is due to the fact that with the SOM more graduations on the wear scale are possible (6x6) given that each neurone can tune to a different wear level, whereas the ART2 is subject to the number of classes created during training. To increase the accuracy of the ART2 it is necessary to reduce the vigilance parameter which controls how finer the classes are.

Both neural networks have demonstrated an ability to classify tool wear under a fixed set of cutting conditions. When asked to classify tool wear under different cutting conditions their performance was radically reduced, although there was a zone that surrounded the training conditions where the neural networks worked effectively. As might be expected the strongest features (feed and tangential force) were the factors that influenced network performance the most. Adding to the suggestions made previously, it would possibly increase adaptability if a selection of points at various cutting conditions were added to the training set, allowing the neural network to learn the combined effect of tool wear and cutting conditions. The latter might become too extensive because a useful training set would have to be large enough to reflect reliably a range of cutting conditions, it would imply a large number of experiments.

7.4 Feature Evaluation and the SOM

In order to evaluate the individual contributions made by each feature obtained from the different sensors the SOM neural network was used as described in Chapter 6. The results have shown that certain features seem stronger than others towards tool wear monitoring. The process of analysing the weights associated with each feature, although difficult due to the complexity associated with the training process, has revealed that the forces are very efficient towards tool wear monitoring, as it would be expected. Also the absolute deviation of both sound and vibration has been shown to contribute greatly for tool wear diagnosis, the least efficient features appear to be the average of sound and vibration.

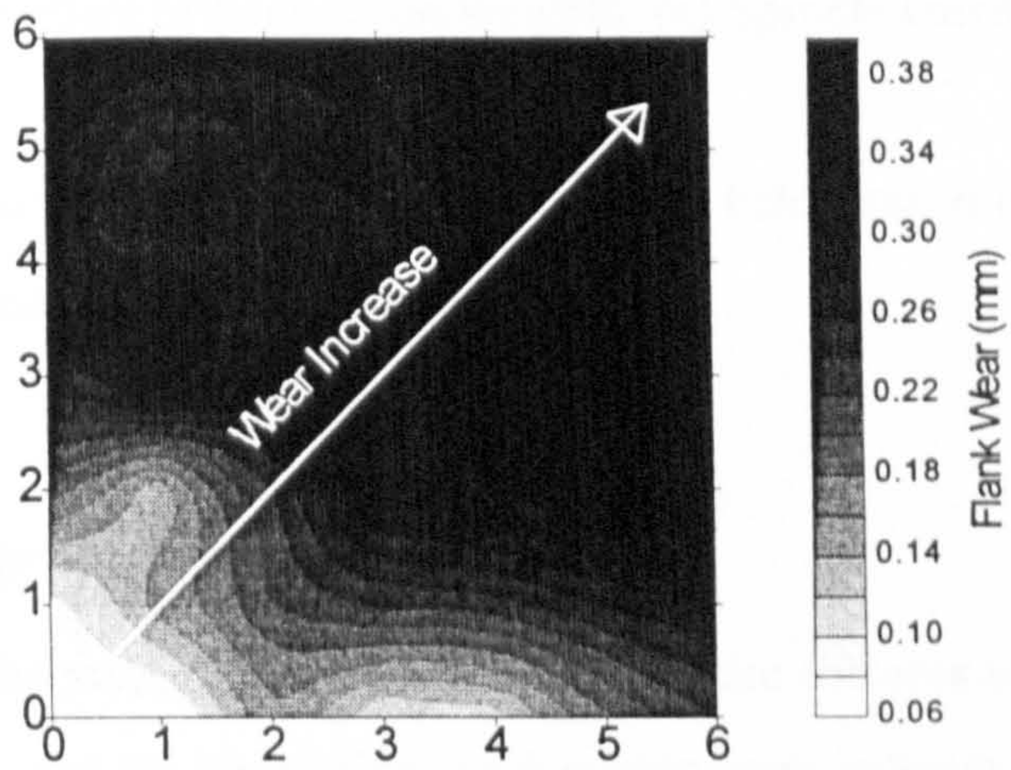


Figure 122: Contour map of wear states associated with each neurone

Figure 122 shows the wear states associated with each neurone upon training. The way in which the map was built reflects the association of input feature vectors representative of different wear levels. Given the unsupervised character of the Self Organising Map the associated neurone weights are expected to differentiate all variables to which the map has tuned to, that is, if for example surface roughness was monitored a map of surface roughness could be built in the same way as the map of wear states. In fact, monitoring two conditions using the same training feature set is very appealing.

As shown before the bar diameter has a strong effect on some of the features, namely the averages of sound and vibration. Since the higher contribution towards spindle speed is made by changes in workpiece diameter it should be possible to identify some areas on the individual feature weight maps revealing this kind of behaviour (Figure 123 (a)). An increase in the spindle current represents a reduction in the bar size, and possibly an added contribution from an increase in tool wear.

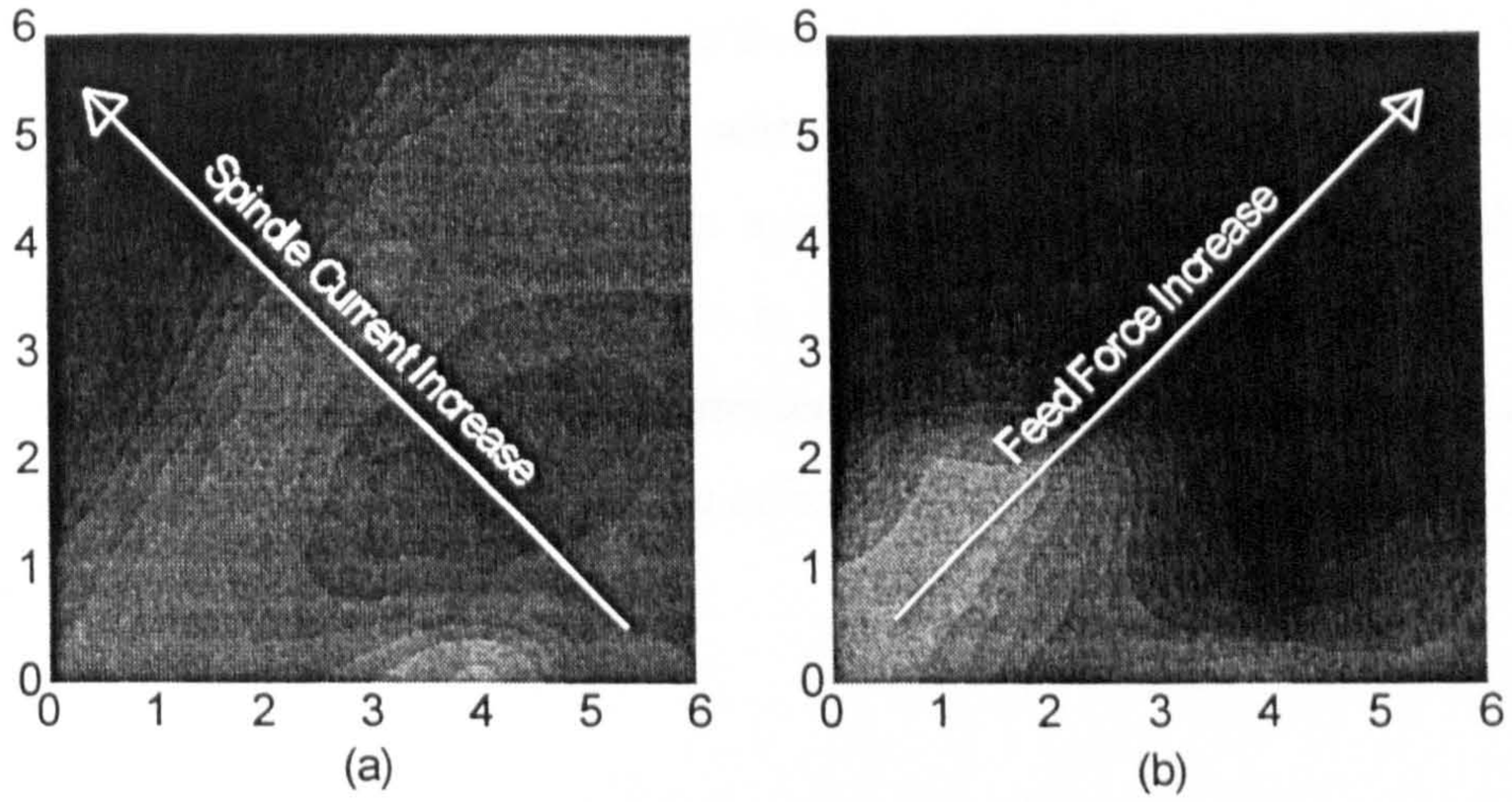


Figure 123: The effect of features on weights: (a) Spindle current (b) Feed force

The fact that some of the features seem to exhibit similar behaviour raises the question whether there is redundancy of features or sensors.

7.5 Feature/Sensor Redundancy

As mentioned earlier, the strong correlation of some of the features to the dynamic behaviour of the cutting tool/lathe, and the generating mechanisms, may indicate that a smaller number of sensors/features would still be able to perform tool wear classification. Surely there is some interrelation between the information obtained from different sensors, however further tests are required to reach such a compromise. Nevertheless, the results obtained so far suggest that the removal of the microphone and accelerometer may be compensated by the extraction of similar features from the force spectra. Also, the average of sound and vibration appear redundant since they imitate the spindle current evolution.

The monitoring architecture built in the Expert System would still be of use since its functionality is not affected by the choice of sensors, or even features. It would be necessary to reconfigure the inputs, nevertheless the monitoring strategy would prevail.

7.6 Hybrid System Assessment

As was presented in Chapter 6 the linking of an Expert System with the neural networks allowed for the removal of obvious mis-classifications, thereby increasing overall system performance.

Generally, the overall system modularity contributed largely to the success of this application, mainly due to the flexibility that it provides when implementing or modifying the embedded knowledge. Modularity becomes essential in a system like this because adaptability is not always possible with a fixed architecture, that is, accuracy may be built into the system with little change to the system according to requirements. Further, knowledge in the form of rules can be updated or added as required with little effort which would not alter the overall structure of the system.

7.6.1 Knowledge-Based Criteria and Performance

The present system was custom built for a specific tool/workpiece combination. However the modular approach to that development has resulted in a system that enables new materials and tool configurations to be incorporated by updating the knowledge base with new parameters for Taylor's tool life equation and/or by training the neural networks under new cutting conditions. The database is easily updated through a user friendly interface that allows the neural networks to be trained with test data acquired in a few tests which can then be stored for future use.

7.6.2 The Importance of Historical Data

In order to improve the performance of the system past experience was taken in to account for each tool life. This consisted of keeping track of classifications and then using a number of classifications to assess the consistency of classification. Through the use of historical data the system becomes aware of previous performance and can judge its reliability, therefore system 'awareness' provides a certain degree of "intelligence" which enables the system to know, according to the "past", what the "present" may or may not be. This methodology resembles, to a certain extent, the cognitive process of the operator when confronted with doubtful information and enables the "machine" to make more reliable decisions.

7.6.3 Efficiency of the Proposed System

Until now only a qualitative judgement has been made regarding system performance and design. This subsection analyses the quantitative aspects and complements the above discussion on system effectiveness and performance.

As the Taylor equation makes a conservative estimate of tool wear, its use allows the elimination of some outliers generated from poor neural network classification or sporadic noisy signals picked up by the sensors. In order to allow the neural networks and Expert System to interact, a ±0.15 mm margin was applied around the optimum prediction so as to prevent the Expert System from removing good classifications which resulted in an improved performance especially for the reduced feature sets where the correlation coefficients increased from 0.691⇒ 0.872 for the ART2, and 0.782 ⇒ 0.955 for the SOM. Overall system results resulted in 100% successful classification of all *worn* states. Thus, the linking of an Expert System based on empirical data and two neural networks enables the monitoring system to achieve consistently better results than either classification technique alone.

It is important to point out that the selected features although sensitive to the tool's behaviour, and indirectly to the state of the tool, should be the subject of more experimental work. Although under the cutting conditions studied, and for the reasons given in other sections, it is possible to classify wear, it is possible that this behaviour will not be reproduced for other cutting conditions. Therefore, it may be necessary to find a feature or features that are insensitive to the cutting conditions, or otherwise estimate the dynamic instability limits.

7.7 Summary of Discussion

It has been demonstrated that, given the correct degree of empiricism, that the neural networks in combination with a supervisory Expert System, can monitor wear under a limited set of cutting conditions. Adding to the previous, some of the changes which would make the proposed system more universal, and based on the suggestions from the previous discussion, would be:-

- Neural network training under the required set of cutting conditions taking into account its zone of influence.
- Enhance the present set of sensor/features, possibly eliminating some of the sensors which show redundancy.

8. Conclusions

It has been demonstrated that the combination of an Expert System and neural networks is an appropriate way to monitor tool wear. Although, it has only been possible to classify tool wear successfully over a limited range of cutting conditions without retraining, the methodology adopted during this work would allow re-training of the monitoring system in a short period of time.

The use of multiple sensors has proved to be of great value towards tool wear evaluation since the noisy character of each sensor alone would lead to certain failure of the monitoring system. The tangential and feed forces proved to be the strongest features of all. The other sensors; spindle current, vibration and sound, have shown to be related to the evolution of wear, although more weakly.

Feature extraction proved to be adequate for the present monitoring system, generating an enormous amount of information which, although very complex, was successfully interpreted by the neural networks.

In this hybrid approach, neural networks classify the off-normal and normal operating states, while the knowledge base interprets the ANN results and classifies the state of the tool with the expert knowledge encoded in it. In particular, this investigation has shown that:-

1. The Self Organizing Map (SOM) and Adaptive Resonance Theory (ART2) neural networks can classify different tool wear levels based on sensory information even in the presence of large amounts of noise.
2. That the Expert System complements the neural networks by removing neural network mis-classifications and increases the overall prediction capacity by the use of process history.
3. The use of multiple neural networks enhances classification by monitoring their reliability and thereafter selecting the one performing better.

A lot of routine testing is required to provide a complete evaluation of the ability of the system to determine whether any occasional deviations from the expected conditions can be identified. Currently, the only flaws identified on-line have been improper clamping and local variations in material quality, no major breakage having occurred. The importance of detecting specific flaws and associated accuracy of such evaluations will be dependent upon the final added value

of the components, and acceptable tolerances on quality. It would be expected, however, that using the framework outlined, the required accuracy could be built into the software for each particular process.

The magnitude of the cutting forces are the single most effective features for monitoring tool wear. Nevertheless these features cannot distinguish between wear and certain changes in cutting conditions. Also heuristic models are very inaccurate and rely extensively on the experimental determination of parameters pertaining to these models. According to the spindle current results the effect of the cutting speed is not obvious, but previous studies suggest that it is possible to exclude cutting speed from affecting the force-dependent features used.

During this study it was found that the dynamic behaviour of the tool reveals an enormous amount of information related to the tool wear state. Nevertheless, this information can only be used reliably if a better understanding of the dynamics under different cutting conditions is achieved. It is therefore suggested that further work should be carried out on the understanding of tool behaviour with changes in cutting conditions.

8.1 Further Work and Recommendations

A number of other possible pieces of work might be undertaken to improve the range of applicability and adaptability of the system.

- Investigate features which are independent of cutting conditions, such as material and/or machine tool. More research is required to find features from the sensor data which are less sensitive to changes in cutting conditions. Such features will have to accommodate stability effects caused by changes in cutting conditions as well as tool degradation, as such, monitoring indices have to date relied on constant cutting conditions, which makes them very limited.

- Extending the knowledge base to allow application in a production environment. Further work is needed to extend the knowledge base to cope with aspects such as optimisation of the cutting conditions, surface finish requirements, and tool change strategies. This could be achieved by testing the monitoring system in a real production environment. Then, with the help of technical specialists, additional rules could be incorporated into the Expert System to take into account specific product requirements.

- Increasing the adaptability of the neural networks. By increasing the training data set evenly over the entire range of cutting conditions for all tool wear states it might be possible to remove the effect of cutting conditions in the neural network classification or even allow the networks to identify a change in cutting conditions.

- Optimise the software for real-time processing. The software developed so far is in an experimental state and could be better achieved through the use of dedicated object oriented structures. On-line implementation will dictate its own necessities regarding speed of execution as well as flexibility, characteristics that were not studied to a significant depth given the broader scope of the project and fields studied.

- Investigate other sensors. Other sensors, such as Acoustic Emission (AE), might prove to be of value in the tool wear monitoring system and should be investigated.

- Investigate the minimum number of sensors so as to reduce the monitoring system cost, as was suggested in the discussion.

Appendix A - ART2 System Dynamics According to Carpenter and Grossberg (1987)

A more detailed description of the ART2 algorithm as used in the monitoring system is presented here, after Carpenter and Grossberg (1987), where all terminology originates. These equations were used in order to build the ART2 module described in Chapter 4.

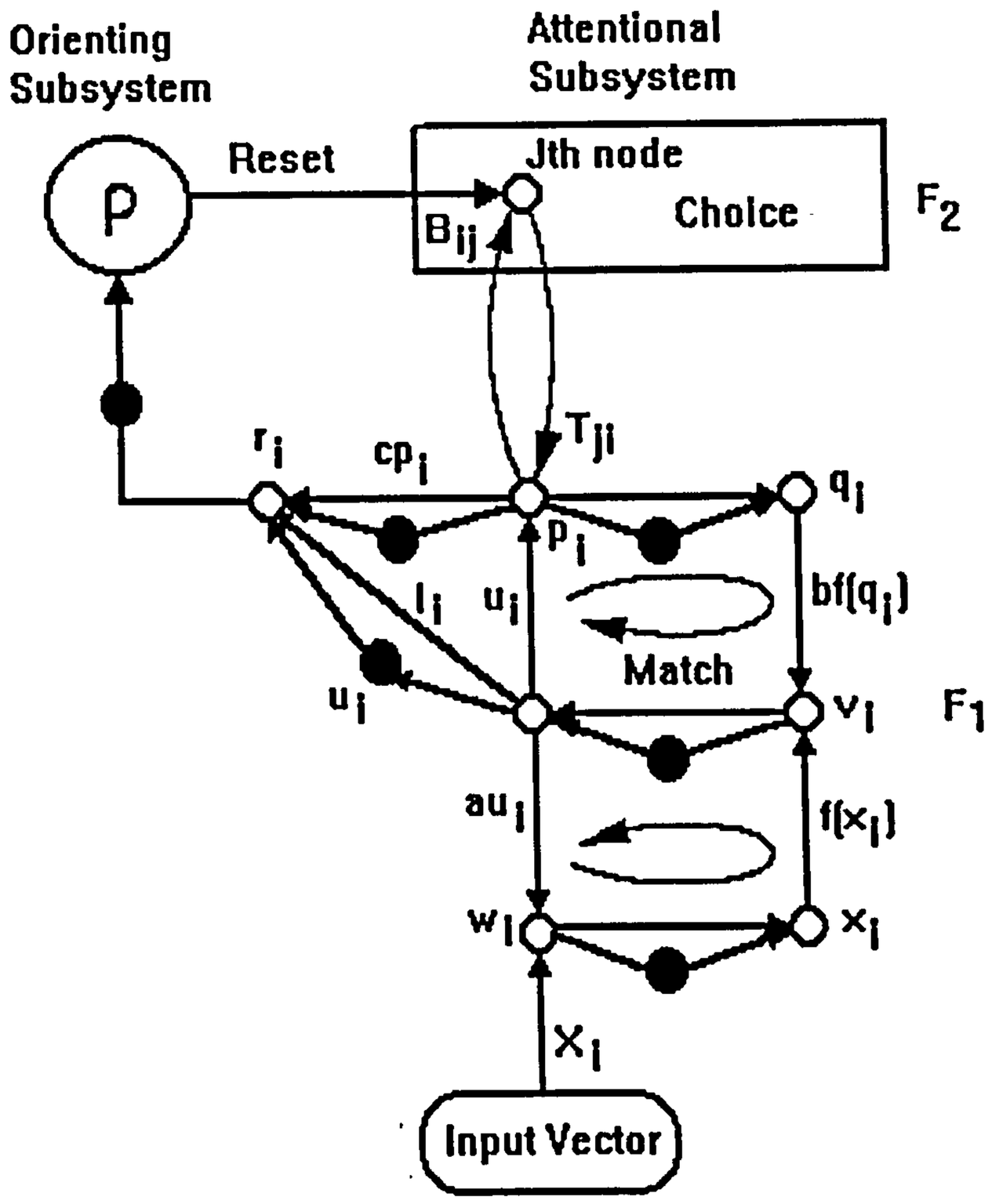


Figure 124: A representation of the ART network of Grossberg (1987).

Figure 124 reproduces the ART2 architecture given by Carpenter and Grossberg. In order to cope with an arbitrary sequence of analogue input patterns, ART2 architectures embody solutions to a number of design principles, such as the stability-plasticity trade-off, the search-direct access, and the match-reset trade-off. A parallel search scheme updates itself adaptively as the learning process unfolds, and realises a form of real-time hypothesis discovery, testing, learning, and recognition.

A.1 The Input Representation Field F_1

An incoming pattern first undergoes extensive processing in ART2. Several parameters and constraints play important role in this processing as well as in operation of ART2. They are as follows:-

1. The bottom-up weight vector for all nodes j, $j=1,...,n$, denoted as b_{ij}, must initially obey $b_{ij} \leq 1/\sqrt{m}$ (where m is the dimension of the input pattern). The top-down weight vectors, denoted z_j, equal 0 at initialisation.
2. $t = 1/\sqrt{m}$
3. $a = b = 10$
4. $(cd)/(1-d) \leq 1$
5. $w_i = x_i = v_i = u_i = q_i = p_i = 0$

The lower (F_1) layer of ART normalises the pattern, then suppresses noise, then re-normalises. A squashing function accomplishes the noise suppression. Carpenter and Grossberg suggest the following:-

$$f(x) = \begin{cases} 0 & if\ 0 \leq x \leq \theta \\ x & if\ x \geq \theta, \end{cases} \tag{A.1}$$

$$0 < \theta < t \tag{A.2}$$

In order to be able to process analogue input patterns, the ART2 network needs to include three processing layers in the F_1 STM field, as shown in Figure 124. The combination of these three processing layers enables the network to separate signal from noise, to enhance the contrast of activation signal and, in addition, to perform the matching function between the F_1 STM and the F_2 STM. The equations below, given originally in Carpenter and Grossberg (1987), shows the values for the labelled nodes of the system pictured in Figure 124.

$$w_i = I_i + au_i, \tag{A.3}$$

$$x_i = \frac{w_i}{\|w\|}, \tag{A.4}$$

$$v_i = f(x_i) + bf(q_i), \tag{A.5}$$

$$u_i = \frac{v_i}{\|v\|} \tag{A.6}$$

$$q_i = \frac{p_i}{\|p\|}, \tag{A.7}$$

where b is a constant and f is the non-linear signal function in equation (A.1). At the top of F_1 layer, vector p sums both the internal F_1 signal vector u and all the $F_2 \rightarrow F_1$ filtered signals:-

$$p_i = u_i + \sum_j g(j) t_{ji} \tag{A.8}$$

In this equation, $g(j)$ is the output of the jth F_2 node and t_{ij} is the top-down weight (LTM trace) between the jth F_2 node and the ith F_1 node.

A.2 Choose in F_2 field and Reset in the Orienting Sub-System

The normalised input pattern next enters the competitive stage of the system. As in competitive learning, a node at the second (F_2) layer receives as input the inner product of its weight vector with the processed input vector. The node having the greatest input 'wins' the competition. Next the network checks that the winning node sufficiently matches the pattern. If the match should prove insufficient, according to the vigilance parameter discussed below, then the winning node is 'reset' by the orienting subsystem, or reset mechanism, of the network. The reset cannot participate in the coding (classifying) of the present pattern. The F_2 node having the next greatest input becomes the winner, its match is checked, and so on.

Match in the F_1 field, choose in the F_2 field and reset in the orienting subsystem are very closely related functions in the ART2 dynamics. The $F_2 \rightarrow F_1$ input is a sum of weighted path signals, as in equation (A.8). The $F_1 \rightarrow F_2$ input is also a sum of weighted path signals which represents the matching score between the current STM vector p and the jth LTM trace in the F_2:-

$$s_j = \sum_i p_i b_{ij}, \tag{A.9}$$

where s_j is the matching score and b_{ij} are the bottom-up weights (LTM trace). After all matching scores are obtained, the one with maximum matching value is chosen as an activated node in the F_2:-

$$s_j = \max\{s_j : j = M+1,\ldots,N \tag{A.10}$$

Then, the *j*th F_2 node is activated. The activation of the F_2 is given as follows:-

$$g(j) = \begin{cases} d & if\ j = J \\ 0 & if\ j \neq J \end{cases} \tag{A.11}$$

To this end, the F_2 activation is propagated back to vector p_i in F_1 field according to equations (A.8) and (A.11), then a vigilance test is carried out to determine whether the top-down signal is matched with the input pattern. The vigilance test is given as follow:-

$$r_i = \frac{u_i + cp_i}{\|u\| + \|cp\|}, \tag{A.12}$$

$$\frac{\rho}{\|r\|} > 1\,? \tag{A.13}$$

where *c* is a constant and $0 < r < 1$. If the match fails to pass the vigilance test, a reset is sent to the F_2 and forces the F_2 to deactivate the selected F_2 node and search for the next best match. Otherwise, the bottom-up and top-down weights are adapted from the following equations:-

$$b_{ij}(t+1) = dt\left[p_i - b_{ij}(t)\right] \tag{A.14}$$

$$t_{ij}(t+1) = dt\left[p_i - t_{ij}(t+1)\right] \tag{A.15}$$

where *dt* is the weight adjusting parameter. The initial values for top-down and bottom-up weights must satisfy the following constraints:-

$$t_{ij}(0) \leq \frac{1}{(1-d)\sqrt{m}}, \tag{A.16}$$

$$b_{ij}(0) \leq \frac{1}{(1-d)\sqrt{m}}, \tag{A.17}$$

$$\frac{cd}{1-d} \leq 1, \tag{A.18}$$

Where *c* is the constant used in equation (A.12), *d* is a constant, and *m* is the dimension of the input vector. The initial bottom-up weights, $b_{ij}(0)$, should be set as large as possible because this helps stabilise the ART2 network by ensuring that the system will form a new category, rather that re-code an established but badly mismatched one.

A.3 The Reset Mechanism

The reset function in ART2 essentially checks that the 'winning' node at the output layer actually matches the input pattern as closely as desired. Reset occurs if the reset function falls below ρ, Carpenter and Grossberg built it in such a way to ensure the following:-

1. When learning has just begun, ART2 will not reset a new uncommitted node; i.e. learning can commence.
2. As learning progresses and nodes become committed, mismatches will trigger resets; i.e. learning increases mismatch sensitivity

A.4 Parameters, Constraints and Initialisation

Several parameters affect ART2 operation, Carpenter and Grossberg suggests values for some of these and provide ranges for others. The parameters *c* and *d* are the least mysterious, they play an important role in the reset equation, suggested values are 0.1 and 0.9, respectively. Parameters and b are, perhaps, most mysterious, in that Carpenter and Grossberg suggest they simply equal '10'. They provide a rough sort of scaling in weight update, the vigilance parameter has a well defined meaning but is difficult to specify a priori, as it defines the level of closeness of patterns within classes.

Appendix B - The Self Organizing Map

Feature extraction can improve the generalisation ability of classifiers and reduce the computational requirements of pattern classification. Kohonen's Self-Organising Map has a very desirable property of topology preserving, which captures an important aspect of feature maps that occur in the human brain. The network architecture is shown in Figure 125 and consists of a two dimensional array of nodes, each of which is connected to all the input nodes.

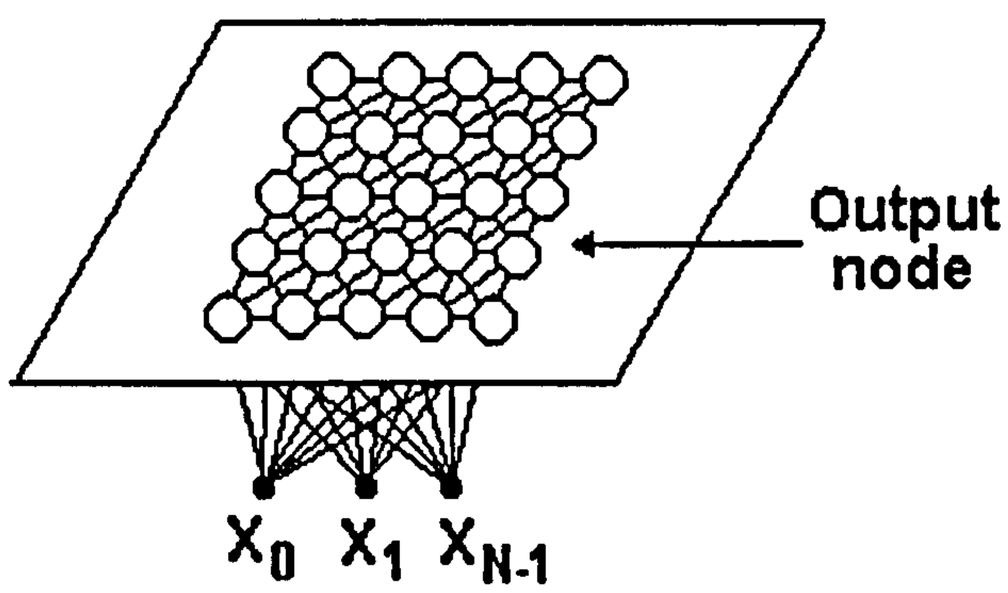


Figure 125: SOM 2D projection Map

Because of the topology-preserving property of the SOM, some insights into the data can be gained by examining the activation pattern of the 2D array of units when each of the features is presented at one time to the network after the training is complete.

B.1 The Self Organising Algorithm

The SOM consists of two layers of neurones, the first is the sensory or input layer, consisting of n neurones, one for every input feature, that receives the information of the sensory space. These neurones work as buffers that distribute the input data to the neurones on the second layer, without performing any computation. The computation is carried out on the second layer, called the map, that also acts as the output layer. The map consists of a rectangular grid of M^2 neurones or processing elements that work in parallel. The N input neurones are labelled by the i index, with $1 \leq i \leq N$, and the M^2 neurones on the map by j and k, with $1 \leq j \leq M$, and $1 \leq k \leq M$ that indicate the location of the processing elements. Every i neurone in the sensory layer is connected to every (j,k) neurone on the map through a synaptic weight. In summary, the map can be seen as a two-dimensional array of identical elements, with each one storing a vector w_{ijk} of synaptic weights.

R.G. Silva 1997

There are two operating modes; learning and recall. The map is developed during learning, in this phase, sample vectors x_i are draw from its training data set. For every iteration, one vector is presented to the network, the similarity between the weight vector of every processing element and the input feature vector is computed by means of a similarity measure, and competition among the nodes start. This is won by the most similar node, which is called the 'winner'. It was defined a symmetric neighbourhood of neurones surrounding the winning neurone, this neurone, and those belonging to its neighbourhood, are updated in such away that their weight vectors approach to the present input vector slightly. The process starts with a wide neighbourhood that allows a global ordering of the synaptic weights, this shrinks with time, until only the winning neurone and its nearest neighbours are updated. The same happens with the learning rate $\varepsilon(t)$, that starts at a value less than 1.0 and decreases until a small value is reached.

The most appealing characteristic of competitive learning algorithms is that they perform vector quantization, dividing the input vector space into discrete, equi-probable regions. Each processing element represents one of these regions in the vector space. If it is possible to have the probability that an input vector falls into a region being the same for each region, then the processing element would generate an optimal 1 out of M^2 code for the input vector space. It has been noted that Kohonen learning does not achieve this result, but is biased in favour of the regions of lower density of input vectors (DeSieno, 1988). Therefore, the goal is to introduce a conscience mechanism, as described in (DeSieno), to bring all the processing elements available into the solution quickly, and to bias the competition process so that each processing element can win the competition with close to the $1/M^2$ probability desired for an optimal vector quatization.

B.1.1 SOM Implementation Algorithm

There is no standard training algorithm, being the final results quite independent of its actual realisation (starting parameters, learning rate, time dependence of the parameters,...). An explanation of the algorithm used in this work will now be presented. Every iteration t is a discrete step in time.

First synaptic weights, w_{ijk}, have to be initialised to small random weights, initialise neurone winning frequency F_{jk} to $1/M^2$ for all output nodes. Presented a new input vector, every (j,k) neurone on the map computes in parallel the similarity between the input vector w_{ijk}, Euclidean

distance is used as a similarity measure (Equation B.1). The present neighbourhood of the winning neurone consists of the winner and the neurones surrounding it.

$$d_{jk} = \sum_{i=0}^{N-1}\left(x_i(t) - w_{ijk}(t)\right)^2 \qquad \text{(B.1)}$$

At this point a bias is introduced to modify the competition, so that the winning node reduces its probability of winning again if selected.

$$d_{jk\,new} = d_{jk\,old} + \gamma\left(M^2.F_{jk} - 1\right) \qquad \text{(B.2)}$$

The constant γ represents the bias factor, it determines the distance a losing processing element can reach in order to enter the solution (used as 0.01). Search then starts for the winning neurone (j_0,k_0), this is the neurone whose distance is minimum according to the Euclidean distance, i.e. that whose weight is most similar to the input vector. A bias is developed for each processing element based on the number of times it has won the above competition (β used as 0.01).

For the winning neurone,

$$F_{jk\,new} = F_{jk\,old} + \beta\left(1.0 - F_{jk\,old}\right) \qquad \text{(B.3)}$$

For all other neurones,

$$F_{jk\,new} = F_{jk\,old} + \beta\left(0.0 - F_{jk\,old}\right) \qquad \text{(B.4)}$$

where β corresponds to the coefficient of distance adjustment

The process starts with a fairly wide neighbourhood that allows for a global ordering of the map, with a starting radius R_0 (2 times M). Thereafter, the radius size slowly decreases with time, shrinking until only the (j_0,k_0) neurone and its six nearest neighbours (final radius $R_f = 1$) are involved in the updating process. For this purpose it was used a linear decreasing for updating $R(t)$:-

$$R(t) = R_0 + (R_f - R_0)\frac{t}{t_{nf}} \qquad \text{(B.5)}$$

where t is the present iteration and t_{rf} is the maximum number of iterations until reaching the final radius R_f. An accurate rule is not important, and other functions for updating $R(t)$ can be used. To modify the weights w_{ijk} of the winning neurone (j_0,k_0) and those of its neighbours to make their weight vectors more similar to the inputs is described next:-

$$w_{ijk}(t+1) = w_{ijk}(t) + \varepsilon(t).h(t).\left[x_i(t) - w_{ijk}(t)\right] \qquad \text{(B.6)}$$

where $\varepsilon(t)$ is the learning rate and $h(t)$ the lateral interaction function (decreasing with the distance to the winning neurone), that defines the present neighbourhood. The learning rate decreases with time, as is the case with $R(t)$:-

$$\varepsilon(t) = \varepsilon_0 e^{\frac{t_{rf}-t}{t_{rf}}} \qquad \text{(B.7)}$$

where ε_0 is the starting learning rate (< 1.0), t is the present iteration and t_{rf} the maximum number of iterations. Again, an accurate rule is not important and other functions can be used.

The function $h(t)$ provides the lateral interaction between the neurones. It depends on the present neighbourhood radius, is symmetric and is centred on the winning neurone:-

$$h(t) = \frac{h_0^{\sqrt{(j_0-j)^2+(k_0-k)^2}}}{R(t)} \qquad \text{(B.8)}$$

where h_0 is a chosen value ($h_0 < 1$), in this case a value of 0.75 proved to be efficient. This enables the central neurone (the winning neurone) to become closer to the input pattern compared with the surrounding neurones that will change according to their distance to the winning neurone. After a sufficient number of iterations the training is stopped, and the network operates in recall mode. In this mode the map responds to an input vector without modifying its weights. Again, every processing element computes the distance between its weight vector and the input vector, and a competition starts that is won by the neurone whose weights are more similar to the input vector.

Appendix C - Neural Networks Software Code

The following programs were built in the 'C' language as external Dynamic Link Librarie (DLL) modules, these allow fast execution of time consuming numerical calculation procedures. They are automatically called upon request, KAPPA-PC handles their execution and scheduling as required by the monitoring strategy..

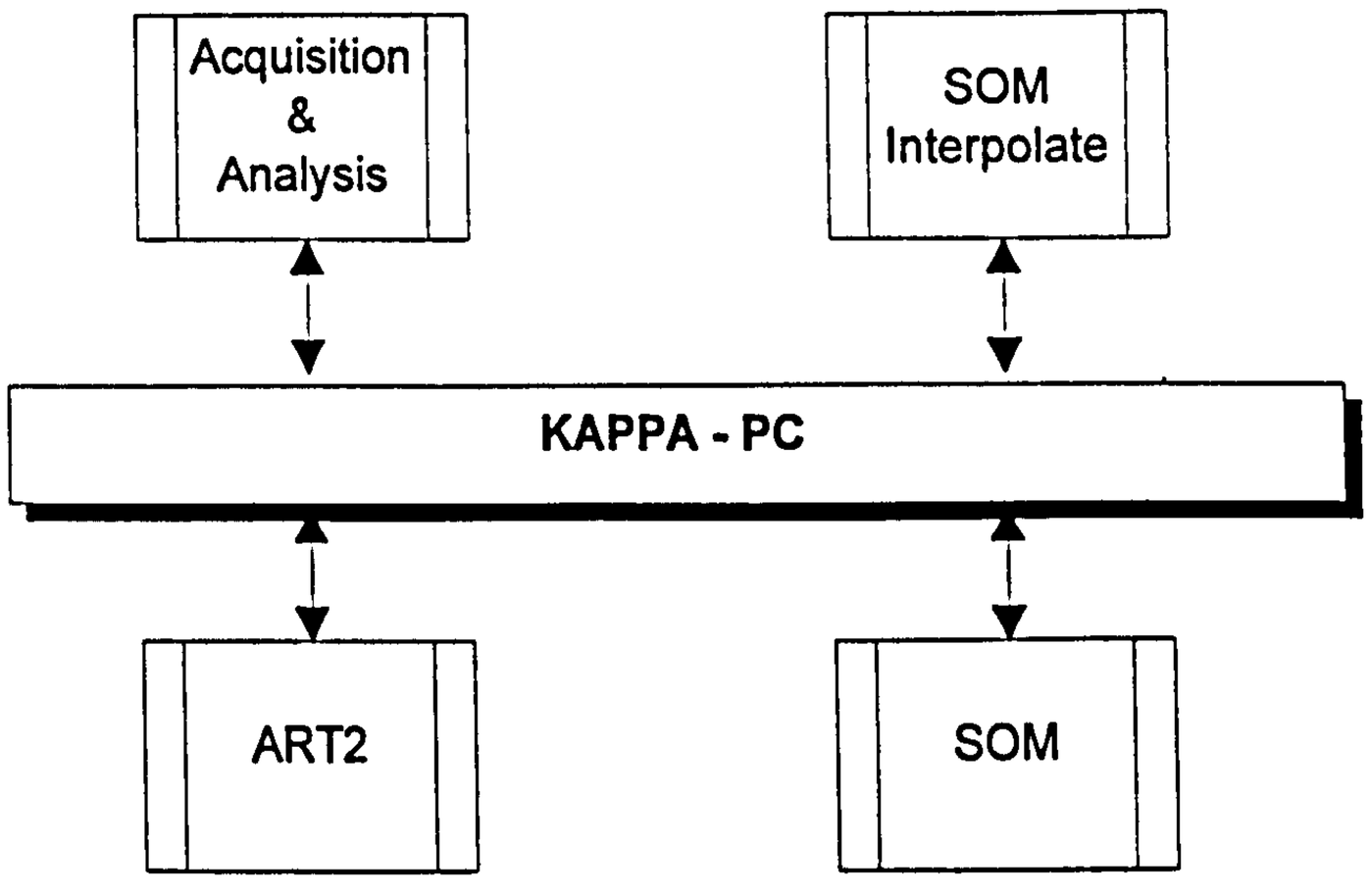


Figure 126: Modules in 'C'

C.1 Data Acquisition and Analysis Module

This module is responsible for the acquisition and analysis of data, the expert system provides the required inputs and scheduling.

Sampling frequency is passed on to the Data AQuisition (DAQ) module as well as frequency bands to be averaged. The data provided by the sensors is acquired for a short period and stored in memory for further processing. Processing for feature extraction is then carried out by the in built functions (moments analysis and FFT) and then transferred to KAPPA-PC. The reason for integrating both data acquisition and signal processing into the same module is to avoid KAPPA from handling enormous amounts of data. Figure 127 shows a simplified step by step flow diagram of the procedures in this module.

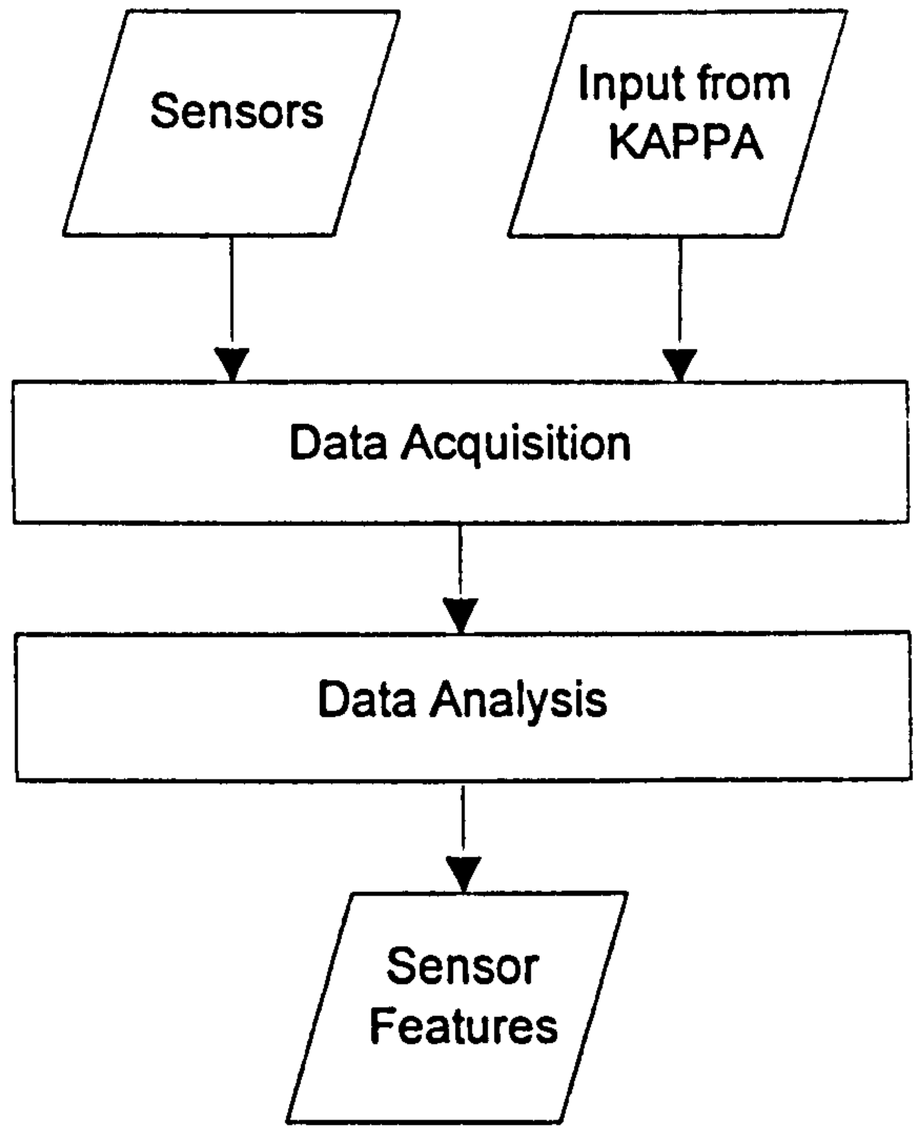


Figure 127: Flow diagram of Data Acquisition Module (DAQ)

More detail of the procedures can be encountered in the following program list, this was built with ANSI C.

```
#include "kappadll.h"
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

#include "edr.h"

#define samples 512
#define sensors 5
#define features 15

double data_matrix[sensors][samples];          // acquired data matrix
double help_matrix[2*samples+1];
int num,ref=1;
int file_save;
double Band_1L,Band_1R,Band_2L,Band_2R;

R.G. Silva 1997
```

```
double Samp_Fr;
LISTID lista;

/****************************** Data Acquisition ******************************/

void data_acquisition(void)
{
    int  baseaddr = 0x700;
    int bh;
    int i,j;
    unsigned bin[sensors*samples*2];
    long uvolts[sensors*samples];

    bh = EDR_AllocBoardHandle();
    EDR_InitBoard(bh,baseaddr);
    EDR_SetADTransferMode(bh,EDR_SINGLEDMA);
    EDR_SetADClockmilliHz(bh,Samp_Fr*1000*sensors);
    EDR_SetADChanListLen(bh,0);
    for (i=0;i<sensors;i++) {
            EDR_AddToADChanList(bh,i);
            EDR_SetADInGain(bh,i,1);
    }
    EDR_ADInBinBackground(bh,sensors*samples,bin);
    while ((num=EDR_BackgroundADInStatus(bh)) < samples*sensors);
    EDR_StopBackgroundADIn(bh);
    EDR_ADInBinToVoltageBlock(bh,uvolts,bin,num,0);

    for (i=0;i<samples;i++)                    // tranfer data to matrix
            for (j=0;j<sensors;j++)
                    data_matrix[j][i] = uvolts[i*sensors+j]/pow10(6);

        EDR_FreeBoardHandle(bh);

}

        /******************  Statistical Analysis  ***********************/
```

R.G. Silva 1997

```
void mean_value(int column)                    // Calculate mean value
{
        int i;
        double s=0.0,ave;

        for (i=0;i<samples;i++)
                s+=data_matrix[column][i];
        ave=s/samples;
        KppInsertNthElem(lista,ref++,_f2a(ave));
        if (file_save) KpcWriteLineF(_f2a(ave));
}

void moments(int column)              // Calculate ave,adev,skew,kurt
{
        int i;
        double s=0.0,p,ep=0.0;
        double ave,adev,skew,curt,var,sdev;

        for (i=0;i<samples;i++) s+=data_matrix[column][i];
ave=s/samples;
adev=var=skew=curt=0.0;
for (i=0;i<samples;i++) {
                adev += fabs(s=data_matrix[column][i]-ave);
                var += (p = s*s);
                skew += (p *= s);
                curt += (p *= s);
 }
 adev /= samples;
 var = (var-ep*ep/samples)/(samples-1);
 sdev = sqrt(var);
 if (var) {
                skew /= (samples*var*sdev);
                curt = curt/(samples*var*var) - 3.0;
 }                                             // Write parameters:

 KppInsertNthElem(lista,ref++,_f2a(adev));          // Absolute deviation
 KppInsertNthElem(lista,ref++,_f2a(ave));           // Average
 KppInsertNthElem(lista,ref++,_f2a(curt));          // Kurtosis
```

```
        KppInsertNthElem(lista,ref++,_f2a(skew));          // Skewness
        if (file_save) {
                KpcWriteLineF(_f2a(adev));
                KpcWriteLineF(_f2a(ave));
                KpcWriteLineF(_f2a(curt));
                KpcWriteLineF(_f2a(skew));
        }
}

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr

void FFT(int column)                                    // Calculate FFT
{
        unsigned long n,mmax,m,j,istep,i;
        double wtemp,wr,wpr,wpi,wi,theta;
        double tempr,tempi;

        for (i=0;i<samples;i++) {
                help_matrix[2*i+1] = data_matrix[column][i];
                help_matrix[2*i+2] = 0.0;
        }

        n=2*samples;                    // FFT algorithm
        j=1;
        for (i=1;i<n;i+=2) {
                if (j > i) {
                        SWAP(help_matrix[j],help_matrix[i]);
                        SWAP(help_matrix[j+1],help_matrix[i+1]);
                }
                m=n >> 1;
                while (m >= 2 && j > m) {
                        j -= m;
                        m >>= 1;
                }
                j += m;
        }
        mmax=2;
        while (n > mmax) {
```

```
istep=mmax << 1;
theta=2*M_PI/mmax;
wtemp=sin(0.5*theta);
wpr = -2.0*wtemp*wtemp;
wpi=sin(theta);
wr=1.0;
wi=0.0;
for (m=1;m<mmax;m+=2) {
        for (i=m;i<=n;i+=istep) {
                j=i+mmax;
                tempr=wr*help_matrix[j]-wi*help_matrix[j+1];
                tempi=wr*help_matrix[j+1]+wi*help_matrix[j];
                help_matrix[j]=help_matrix[i]-tempr;
                help_matrix[j+1]=help_matrix[i+1]-tempi;
                help_matrix[i] += tempr;
                help_matrix[i+1] += tempi;
        }
        wr=(wtemp=wr)*wpr-wi*wpi+wr;
        wi=wi*wpr+wtemp*wpi+wi;
                }
                mmax=istep;
        }
        // End FFT algorithm
}

#undef SWAP

void PSD(void)
{
        int i;
        double tempor;

        // Calculate PSD of Band 1

        tempor=0.0;
        for (i=Band_1L;i<Band_1R;i++)
                tempor += sqrt(help_matrix[2*i+1]*help_matrix[2*i+1] +
        help_matrix[2*i+2]*help_matrix[2*i+2]);
```

```
tempor /= (double) (Band_1R-Band_1L);
KppInsertNthElem(lista,ref++,_f2a(tempor));
if (file_save) KpcWriteLineF(_f2a(tempor));

// Calculate PSD of Band 2

tempor=0.0;
for (i=Band_2L;i<Band_2R;i++)
        tempor += sqrt(help_matrix[2*i+1]*help_matrix[2*i+1] +
    help_matrix[2*i+2]*help_matrix[2*i+2]);
    tempor /= (double) (Band_2R-Band_2L);
    KppInsertNthElem(lista,ref++,_f2a(tempor));
    if (file_save) KpcWriteLineF(_f2a(tempor));
}


short PEXPORT DataHandler (ARGLIST lpArgList)
{
/************************** Read Kappa Inputs ***********************************/
int i,j;
char file_source[12],file_source2[12];

KpaGetArgDouble(lpArgList, 1, Samp_Fr);
KpaGetArgDouble(lpArgList, 2, Band_1L);
KpaGetArgDouble(lpArgList, 3, Band_1R);
KpaGetArgDouble(lpArgList, 4, Band_2L);
KpaGetArgDouble(lpArgList, 5, Band_2R);
KpaGetArgString(lpArgList, 6, file_source, 12);
KpaGetArgString(lpArgList, 7, file_source2, 12);
KpaGetArgInt(lpArgList, 8, file_save);

Band_1L = (int) samples*Band_1L/Samp_Fr;
Band_1R = (int) samples*Band_1R/Samp_Fr;
Band_2L = (int) samples*Band_2L/Samp_Fr;
Band_2R = (int) samples*Band_2R/Samp_Fr;

lista = KppMakeList(0);
```

```
        if (file_save) KpcOpenWriteFile(file_source,_s2a("APPEND"));

/************************** End Kappa Read inputs ******************************/
/************************** Start - Sensor Scanning  ******************************/

        data_acquisition();

/************************** End - Sensor Scanning  ******************************/
/*********** Start - Vector Analysis & Output to Kappa-PC ******************/

        moments(0);             // sound statistics
        FFT(0);                 // FFT of sound
        PSD();                  // PSD of sound
        moments(1);             // vibration statistics
        FFT(1);                 // FFT of vibration
        PSD();                  // PSD of vibration

        mean_value(2);          // feed force
        mean_value(3);          // tang. force
        mean_value(4);          // spindle current

        if (file_save) KpcCloseWriteFile();

        if (file_save) {
                KpcOpenWriteFile(file_source2,_s2a("APPEND"));
                for (i=0;i<samples;i++)                 // tranfer data file
                        for (j=0;j<sensors;j++)
                                KpcWriteLineF(_f2a(data_matrix[j][i]));
                KpcCloseWriteFile();
        }
                KpaReturnList(lista);
                KppDeleteList(lista);

/*************** End - Vector Analysis & Output to Kappa-PC ******************/
}
□


R.G. Silva 1997
```

C.2 Neural Network Code

The following diagram shows the common steps undertaken by the self organising map and adaptive resonance theory network modules. As can be seen both modules can work in two modes, training or recall mode. When training sample data is used to compute the weights for each network, this same data is used to map the classification versus measured flank wear. If the neural network is working on the recall mode it will simply work out the classification of an input feature vector.

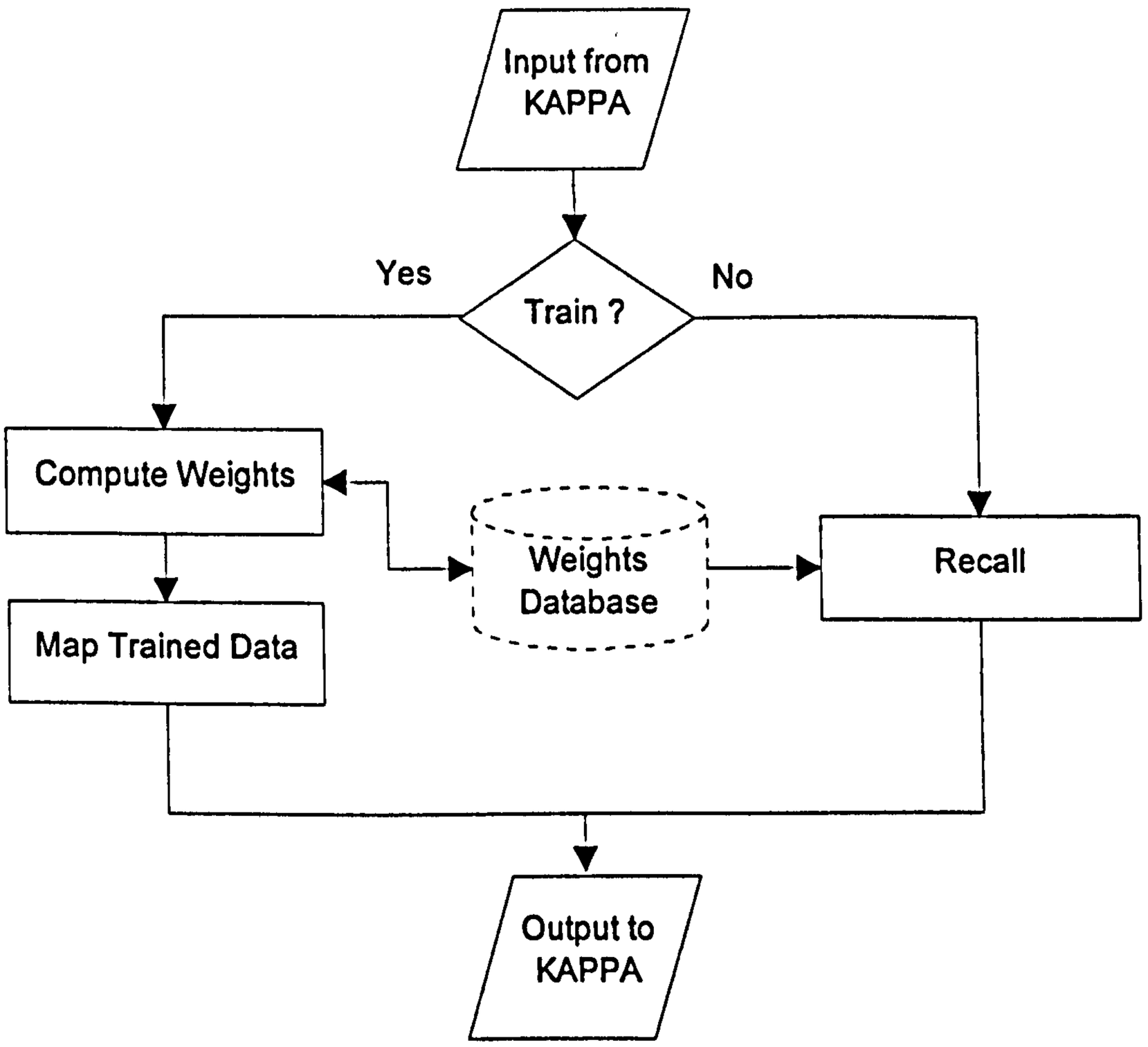


Figure 128: Neural networks flow diagram

The following code was used to built the neural network modules, as presented in Chapter 4.

C.2.1 Self Organising Map

The present self organising map code is used in order to train the network and for the prediction of wear by returning a two dimensional co-ordinate which corresponds to the position of the cluster in the map of wear conditions associated with a certain wear level.

```
#include "kappadll.h"
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <time.h>

#define outputs 10
#define inputs 18
#define n_lines 70

        int tempo;
        double InputVector[inputs];
        double weight[inputs][outputs][outputs];
        char Weights_Source[12],Data_Source[12];
        int train_limit;
        int auxb,auxc;
        double DataVector[n_lines*(inputs+2)];
        double d[outputs][outputs],f[outputs][outputs];
        float norm[]={0.0,1.2, -0.07,0.4, -1.1,0.4, -1.5,1.0, 1.0,5.0, 0.0,4.4, 0.0,0.06, 0.0,0.02, -0.9,0.44,
        -1.5,1.0, 0.1,0.45, 0.0,0.3, -2.7,1.3, 0.8,6.8, 0.8,3.0, 330.0,360.0, 0.1,0.4, 0.9,1.3 ,0.0,1.0};
        int Lines;

        void generate_weights(void)  {
                int k,i,j;

                randomize();
                for (k=0;k<inputs;k++)
                        for (i=0;i<outputs;i++)
                                for (j=0;j<outputs;j++)
                                        weight[k][i][j]=random(30000)/30000.0;
        }

        void ReadWeights(void)  {
                int i,k,j;
                double num;
                char convers[30];
```

```
KpcOpenReadFile(Weights_Source);
for (k=0;k<inputs;k++)
        for (i=0;i<outputs;i++)
                for (j=0;j<outputs;j++) {
                        KpcReadLine(convers,0);
                        weight[k][i][j] =_s2f(convers);
                }
KpcCloseReadFile();
}

void ReadData(void)  {
        int i,j;
        double num;
        char convers[30];

        KpcOpenReadFile(Data_Source);
        i=0;
        while (!(KpcEndOfFile_QSM())) {
                KpcReadLine(convers,0);
                DataVector[i] = _s2f(convers);
                i++;
}
KpcCloseReadFile();
Lines=i-1;

for (i=0;i<(Lines/(inputs+2));i++) {
        for (j=0;j<inputs;j++)
                DataVector[i*(inputs+2)+j]=(DataVector[i*(inputs+2)+j]-
                norm[j*2])/(norm[j*2+1]-norm[j*2]);
}
}

void compute_distance(void) {
        int k,i,j;
        double auxa;

        for (k=0;k<outputs;k++)
                for (i=0;i<outputs;i++) {
```

```
                d[k][i]=pow(InputVector[0]-weight[0][k][i],2.0);
                for (j=1;j<inputs;j++)
                        d[k][i]+=pow(InputVector[j]-weight[j][k][i],2.0);
                d[k][i]=sqrt(d[k][i]);
                }

// Calculate bias and update distances
        for (k=0;k<outputs;k++)
                for (i=0;i<outputs;i++)
                        d[k][i]+=0.1*(outputs*f[k][i]-1);


// calculate the minimum distance
        auxa=d[0][0];
        auxb=auxc=0;
        for (k=0;k<outputs;k++)
                for (i=0;i<outputs;i++)
                        if (auxa>d[k][i]) {
                                auxa=d[k][i];
                                auxb=k;
                                auxc=i;
                        }

// update frequencies of winning nodes
        for (k=0;k<outputs;k++)
                for (i=0;i<outputs;i++)
                        f[k][i]*=1.0-0.1;
        f[auxb][auxc]=f[auxb][auxc]+0.1*(1.0-f[auxb][auxc]);
}

        void update_weights(void) {
        // update the weights according to neighberhood
                int k,i,j;
                int nb;
                double alf;

                nb=20*pow(1.0/20.0,(float) tempo/train_limit); // reducing exponentially with time
```

```
                alf=0.9*pow(0.01/0.9,(float) tempo/train_limit); // decrease with training time,
                    exponentially

        for (k=max(auxb-nb,0);k<min(auxb+nb,outputs);k++)
                for (i=max(auxc-nb,0);i<min(auxc+nb,outputs);i++) {
                        for (j=0;j<inputs;j++)
                                weight[j][k][i]+=alf*exp(-pow(d[k][i]/nb,2.0))
                                        *(InputVector[j]-weight[j][k][i]);
                        }
        }

        void WriteWeights(void) {
                int i,k,j;

                KpcOpenWriteFile(Weights_Source,NULL);
                for (k=0;k<inputs;k++)
                        for (i=0;i<outputs;i++)
                                for (j=0;j<outputs;j++)
                                        KpcWriteLineF(_f2a(weight[k][i][j]));
                KpcCloseWriteFile();
        }

/*************************************************************
                        SOM Handler
*************************************************************/

short PEXPORT SOMHandler(ARGLIST lpArgList)
{

        /* Inport information, from Kappa-PC */
        int i,j,g;
        int line;
        LISTID lista;
        char Train_Source[12];
        ATOMID convers;

        for (i=0;i<outputs;i++)
                for (j=0;j<outputs;j++)
```

```
                f[i][j]=1.0/(outputs*outputs);

        KpaGetArgString(lpArgList, 1, Weights_Source,12);
        KpaGetArgString(lpArgList, 2, Data_Source, 12);
        KpaGetArgString(lpArgList, 3, Train_Source, 12);
        KpaGetArgInt(lpArgList, 4, train_limit);
        for (i=0;i<inputs;i++)
                KpaGetArgDouble(lpArgList, 5+i, InputVector[i]);

        for (i=0;i<inputs;i++)
                InputVector[i]=(InputVector[i]-norm[i*2])/(norm[i*2+1]-norm[i*2]);

        // Program Body

        if (train_limit>0) {
                generate_weights();
                ReadData();
                randomize();
                for (tempo=0;tempo<train_limit;tempo++) for (g=0;g<2;g++) {
                line=random(Lines/(inputs+2));
                for (j=0;j<inputs;j++)
                        InputVector[j]=DataVector[line*(inputs+2)+j];
                compute_distance();
                update_weights();
        }
        WriteWeights();
        KpcOpenWriteFile(Train_Source,NULL);
        for (i=0;i<(Lines/(inputs+2));i++) {
                for (j=0;j<inputs;j++)
                        InputVector[j]=DataVector[i*(inputs+2)+j];
                compute_distance();
                convers = _i2a(auxb);
                KpcWriteLineF(convers);
                convers = _i2a(auxc);
                KpcWriteLineF(convers);
                convers = _f2a(DataVector[i*(inputs+2)+inputs]);
                KpcWriteLineF(convers);
        }
```

R.G. Silva 1997

```
            KpcCloseWriteFile();
            KpaReturnInt(Lines);
    }
    else {
            ReadWeights();
            compute_distance();

            lista = KppMakeList(0);
                KppInsertNthElem(lista,1,_f2a(auxb));
                KppInsertNthElem(lista,2,_f2a(auxc));
                KpaReturnList(lista);
                KppDeleteList(lista);
        }
}
```

□

C.2.2 Adaptive Resonance Theory

The algorithm presented in Chapter 4 and complemented by Appendix A was the base for the production of the following program. This module provided the external capacity of training the ART2 and was also used to recall the cluster association to a particular vector input.

```
#include "kappadll.h"
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <time.h>

#define inputs 15
#define outputs 15
#define n_lines 70

#define t 0.25    // t=1.0/sqrt(inputs);
#define tt 0.01

        int Commitment[outputs];
        double LTM[inputs][outputs];
```

```
double InputVector[inputs];
double P[inputs],b;
double DataVector[n_lines*(inputs+2)],Wear[outputs][2];
double Vigilance,Reference;
char Data_Source[12],Weights_Source[12];
int Train_Limit,Lines;
float norm[]={0.25,1.15, -0.07,0.4, -1.1,0.14, -0.5,0.18, 2.2,4.8, 1.8,4.4, 0.02,0.06,
0.0,0.02, -0.9,0.44, -0.6,0.4, 0.15,0.37, 0.14,0.26, -2.3,-1.19, 2.6,4.2, 1.5,2.5, 0.0,1.0};

void Initialize(void)
{
        int i;

        for(i=0;i<outputs;i++)
                Commitment[i]=0;
}

void ReadWeights(void)
{
int i,k;
double num;
char convers[30];

KpcOpenReadFile(Weights_Source);
for(i=0;i<inputs;i++)
        for(k=0;k<outputs;k++) {
                KpcReadLine(convers,0);
                LTM[i][k] = _a2f(_s2a(convers));
        }
 for(i=0;i<outputs;i++) {
                KpcReadLine(convers,0);
                Commitment[i] = _a2j(_s2a(convers));
        }
        KpcCloseReadFile();
}

        void ReadData(void)
```

```
{
        int i,j;
        double num;
        char convers[30];

        KpcOpenReadFile(Data_Source);
        i=0;
        while (!(KpcEndOfFile_QSM())) {
                KpcReadLine(convers,0);
                DataVector[i] = _a2f(_s2a(convers));
                i++;
}
KpcCloseReadFile();
Lines=i-1;

for (i=0;i<(Lines/(inputs+2));i++) {
        for (j=0;j<inputs;j++)
                DataVector[i*(inputs+2)+j]=(DataVector[i*(inputs+2)+j]-
        norm[j*2])/(norm[j*2+1]-norm[j*2]);
        }
}


void LowerLevelProcessing(void)
{
        double wm=0.0,vm=0.0;
        double W[inputs],V[inputs],X[inputs];
        int i;

        for(i=0;i<inputs;i++) {
                W[i]=InputVector[i];
                wm+=pow(W[i],2.0);
}
wm=sqrt(wm);
for(i=0;i<inputs;i++) {
        X[i]=W[i]/wm;
        if (X[i]>=tt) V[i]=X[i];
        else V[i]=0.0;
```

```
                vm+=pow(V[i],2.0);
        }
        vm=sqrt(vm);
        for(i=0;i<inputs;i++)
                P[i]=V[i]/vm;
}

void Competition(void)
{
        int i,j,k,Ref;
        double T[outputs],swap1,norm[inputs],n,N;

        for(i=0;i<outputs;i++)
                T[i]=0.0;
        for (i=0;i<outputs;i++)
                if (Commitment[i])
                        for(k=0;k<inputs;k++)
                                T[i]+=P[k]*LTM[k][i];
                else {
                        for (k=0;k<inputs;k++)
                                T[i]+=P[k];
                        T[i]*=t;
                }

        Ref=0;
        swap1=T[0];
        for (i=1;i<outputs;i++)
                if (T[i]>swap1) {
                        swap1=T[i];
                        Ref=i;
                }

        // Resonance or reset  & Learning

        if (!Commitment[Ref] || (Commitment[Ref] && T[Ref]>=Vigilance)) {
                Commitment[Ref]=1;
                n=0.0;
                for (i=0;i<inputs;i++) {
```

```
                if (InputVector[i]>tt) norm[i]=InputVector[i];
                else norm[i]=0.0;
                n+=pow(norm[i],2.0);
        }
        N=0.0;
        for (i=0;i<inputs;i++) {
                norm[i]/=n;
                LTM[i][Ref]=b*norm[i]+(1-b)*LTM[i][Ref];
                N+=LTM[i][Ref];
        }
        for (i=0;i<inputs;i++)
                LTM[i][Ref]/=N;
}

if (Commitment[Ref] && T[Ref]<Vigilance) {
        Commitment[Ref]=0;
        for (i=0;i<inputs;i++)
                LTM[i][Ref]=InputVector[i];
}
}

void DetermineResult(void)
{
        int i,k;
        float T[outputs],swap1;

        for(i=0;i<outputs;i++)
                T[i]=0.0;
        for (i=0;i<outputs;i++)
                if (Commitment[i])
                        for(k=0;k<inputs;k++)
                                T[i]+=P[k]*LTM[k][i];

        swap1=-1.0;
        for (i=0;i<outputs;i++)
                if ((T[i]>swap1 || swap1==-1.0) && Commitment[i]) {
                        swap1=T[i];
                        Reference=i;
```

```
                    }

        }

        void WriteWeights(void)
        {
                int i,k;

                KpcOpenWriteFile(Weights_Source,NULL);
                for(i=0;i<inputs;i++)
                        for(k=0;k<outputs;k++)
                                KpcWriteLineF(_f2a(LTM[i][k]));
                for(i=0;i<outputs;i++)
                        KpcWriteLineF(_j2a(Commitment[i]));

                KpcCloseWriteFile();
        }

/*************************************************************
                        ART2 Handler
*************************************************************/

short PEXPORT ART2Handler(ARGLIST lpArgList)
{

        /* Inport information, from Kappa-PC */

        int i,j;
        int line;
        LISTID lista;

        KpaGetArgString(lpArgList, 1, Weights_Source, 12);
        KpaGetArgString(lpArgList, 2, Data_Source, 12);
        KpaGetArgDouble(lpArgList, 3, Vigilance);
        KpaGetArgInt(lpArgList, 4, Train_Limit);
        KpaGetArgDouble(lpArgList, 5, b);
        for(i=0;i<inputs;i++)
                KpaGetArgDouble(lpArgList, 6+i, InputVector[i]);
```

```
for (i=0;i<inputs;i++)
        InputVector[i]=(InputVector[i]-norm[i*2])/(norm[i*2+1]-norm[i*2]);

/* Program Body */

if (Train_Limit>0) {
        Initialize();
        ReadData();
        randomize();
        for (i=0;i<Train_Limit;i++) {
                line=random(Lines/(inputs+2));
                for (j=0;j<inputs;j++)
                        InputVector[j]=DataVector[line*(inputs+2)+j];
                LowerLevelProcessing();
                Competition();
        }
        WriteWeights();
        for (i=0;i<outputs;i++) {
                Wear[i][0]=0.0;
                Wear[i][1]=0.0;
        }
        for (i=0;i<(Lines/(inputs+2));i++) {
                for (j=0;j<inputs;j++)
                        InputVector[j]=DataVector[i*(inputs+2)+j];
                LowerLevelProcessing();
                DetermineResult();
                Wear[Reference][0]+=DataVector[i*(inputs+2)+inputs];
                Wear[Reference][1]+=1.0;
        }
        lista = KppMakeList(0);
        for (i=0;i<outputs;i++)
                if (Wear[i][1]!=0) {
                KppInsertNthElem(lista,2*i+1,_j2a(i));
                KppInsertNthElem(lista,2*i+2,_f2a(Wear[i][0]/Wear[i][1]));
        }
        KpaReturnList(lista);
        KppDeleteList(lista);
```

```
        }
        else {
                ReadWeights();
                LowerLevelProcessing();
                DetermineResult();
                KpaReturnInt(Reference);
        }
}
```

□

C.3 Krigging Method of Interpolation

As described in Chapter 2, this interpolation method yields a good approximation of the SOM map topology and enabled generalisation of nearby unlabeled clusters. It was used after SOM training upon known cluster associations with wear and resulted in a sort of look up table for input vector interpretation.

🖫

```
short PEXPORT InterpolateHandler(ARGLIST lpArgList)
{
#define points 45
#define outputs 7
#define inputs 15

        int i,j;
        double data_in[points][3],data[points][3],matrixa[points][points],matrixb[points];
        double semivariances[points][points];
        char file_in[12],file_out[12],convers[30];
        double averagez,variancez,maxx,maxy,minx,miny,semivariance,radius;
        int Lines,l,m,n;
        double aux1;
        int imax,k,ii,ip,indx[points];
        double big,dum,sum,temp,vv[points],d;

        KpaGetArgString(lpArgList, 1, file_in,12);
        KpaGetArgString(lpArgList, 2, file_out,12);
```

```
/* Read data to fit into surface from file */


KpcOpenReadFile(file_in);
i=0;
while (!(KpcEndOfFile_QSM())) {
        KpcReadLine(convers,0);
        data_in[i][0] = _a2f(_s2a(convers));
        KpcReadLine(convers,0);
        data_in[i][1] = _a2f(_s2a(convers));
        KpcReadLine(convers,0);
        data_in[i][2] = _a2f(_s2a(convers));
        i++;
}
KpcCloseReadFile();
Lines=i-1;
n=Lines;


// Check for points overlaping positions

m=0;
for (i=0;i<n;i++)
    if (data_in[i][2]!=-1.0) {
        data[m][0]=data_in[i][0];
        data[m][1]=data_in[i][1];
        data[m][2]=0.0;
        sum=0.0;
        for (j=i;j<n;j++)
            if (data_in[i][0]==data_in[j][0] &&
            data_in[i][1]==data_in[j][1] && data_in[j][2]!=-1.0) {
                data[m][2]+=data_in[j][2];
                data_in[j][2]=-1.0;
                sum+=1.0;
            }
        data[m][2]/=sum;
        m++;
    }
  n=m;
```

```
// Do calculations for all the grid


averagez=0.0;
maxx=minx=data[0][0];
maxy=miny=data[0][1];
for (i=0;i<n;i++) {
        averagez+=data[i][2];
        if (maxx<data[i][0]) maxx=data[i][0];
        if (minx>data[i][0]) minx=data[i][0];
        if (maxy<data[i][1]) maxy=data[i][1];
        if (miny>data[i][1]) miny=data[i][1];
}
averagez/=n;


variancez=0.0;
for (i=0;i<n;i++)
        variancez+=pow(data[i][2]-averagez,2.0);
variancez/=n;


radius=sqrt(pow(maxx-minx,2.0)+pow(maxy-miny,2.0))/2.0;
semivariance=variancez/radius;


// Calculate distances and Semivariances between known points and location


for (i=0;i<n;i++)
        for (j=i;j<n;j++)
                semivariances[i][j]=semivariance*sqrt(pow(data[i][0]-
                data[j][0],2.0)+pow(data[i][1]-data[j][1],2.0));


 // fill in matrix a with known values of semivariances


for (i=0;i<n;i++)
        for (j=i;j<n;j++)
                matrixa[i][j]=matrixa[j][i]=semivariances[i][j];
for (i=0;i<n;i++)
        matrixa[n][i]=matrixa[i][n]=1.0;
matrixa[n][n]=0.0;
```

```
// Using LUDCMP.C from NRC, calculate LU decomposition of a


d=1.0;
for (i=0;i<=n;i++) {
        big=0.0;
        for (j=0;j<=n;j++)
                if ((temp=fabs(matrixa[i][j]))>big) big=temp;
        vv[i]=1.0/big;
}

for (j=0;j<=n;j++) {
        for (i=0;i<j;i++) {
                sum=matrixa[i][j];
                for (k=0;k<i;k++) sum-=matrixa[i][k]*matrixa[k][j];
                matrixa[i][j]=sum;
        }
        big=0.0;
        for (i=j;i<=n;i++) {
                sum=matrixa[i][j];
                for (k=0;k<j;k++)
                        sum-=matrixa[i][k]*matrixa[k][j];
                matrixa[i][j]=sum;
                if ((dum=vv[i]*fabs(sum))>=big) {
                        big=dum;
                        imax=i;
                }
}
if (j!=imax) {
                for (k=0;k<=n;k++) {
                        dum=matrixa[imax][k];
                        matrixa[imax][k]=matrixa[j][k];
                        matrixa[j][k]=dum;
                }
                d=-d;
                vv[imax]=vv[j];
        }
        indx[j]=imax;
```

R.G. Silva 1997

```
            if (matrixa[j][j]==0.0) matrixa[j][j]=pow10(-20);
            if (j!=n) {
                    dum=1.0/matrixa[j][j];
                    for (i=j+1;i<=n;i++) matrixa[i][j]*=dum;
            }
}

// choosing a point p where to interpolate, starts the interpolation

KpcOpenWriteFile(file_out,0);

for (l=0;l<outputs;l++)
        for (m=0;m<outputs;m++) {

        // distances and semivariances of location p

        for (i=0;i<n;i++)
                semivariances[i][n]=semivariance*sqrt(pow(data[i][0]-
                l,2.0)+pow(data[i][1]-m,2.0));

// matrix b formation with remaining parameters

for (i=0;i<n;i++)
        matrixb[i]=semivariances[i][n];
matrixb[n]=1.0;

// Find the weights Wi by solving the equation A.X=B, using
 // the results from LUDCMP.C in LUBKSB.C (next)

ii=-1;
for (i=0;i<=n;i++) {
        ip=indx[i];
        sum=matrixb[ip];
        matrixb[ip]=matrixb[i];
        if (ii>-1)
                for (j=ii;j<=i-1;j++) sum-=matrixa[i][j]*matrixb[j];
        else if (sum) ii=i;
        matrixb[i]=sum;
```

```
        }
        for (i=n;i>=0;i--) {
                sum=matrixb[i];
                for (j=i+1;j<=n;j++) sum-=matrixa[i][j]*matrixb[j];
                matrixb[i]=sum/matrixa[i][i];
        }

        // Write value into output matrix of surface

        aux1=0.0;
        for (i=0;i<n;i++)
                aux1+=matrixb[i]*data[i][2];
        KpcWriteLineF(_f2a(aux1));
}

// Write map into data file

KpcCloseWriteFile();
KpaReturnDouble(semivariance);
}
```

□

C.3.1 SOM Map Interpretation Algorithm

This module, also implemented as an external DLL, provided the means to recall the SOM wear association with different clusters.

```
short PEXPORT WearInterpolateHandler(ARGLIST lpArgList)
{
        char file_in[12];
        int i,j,out=7;
        double x,y;
        double data[7][7];
        char convers[30];

        KpaGetArgString(lpArgList, 1, file_in,12);
        KpaGetArgDouble(lpArgList, 2, x);
        KpaGetArgDouble(lpArgList, 3, y);
```

```
KpcOpenReadFile(file_in);
for (i=0;i<out;i++)
        for (j=0;j<out;j++) {
                KpcReadLine(convers,0);
                data[i][j] = _a2f(_s2a(convers));
        }
KpcCloseReadFile();
        KpaReturnDouble(data[x][y]);
}
```

Appendix D - Functions Defined Within Kappa

Functions are the most flexible tool in KAL, the language within KAPPA. All the actions necessary in the various windows in KAPPA-PC, from adding an object to activating a method, can be done easier, faster and more efficiently on a larger scale using the KAL functions. The following functions were used to enable the monitoring system to exchange information from and to files as well as performing several actions in the coarse of evaluating tool wear.

```
/*****************************************
**** FUNCTION: DBopenTool
*****************************************/
MakeFunction( DBopenTool, [],
        Let [file SelectFile( "Database Name", "*.wks" )]  {
        Tool:File = file;
        ResetImage( Edit1 );
} );

/*****************************************
**** FUNCTION: DBopenMaterial
*****************************************/
MakeFunction( DBopenMaterial, [],
        Let [file SelectFile( "Material Database Name", "*.wks" )] {
        Material:File = file;
        ResetImage( Edit2 );
} );

/*****************************************
**** FUNCTION: GotoDBSetUp
*****************************************/
MakeFunction( GotoDBSetUp, [],
        {
        ShowWindow( Session1 );
} );

/*****************************************
**** FUNCTION: ReturnDBsetup

R.G. Silva 1997
```

```
*****************************************/
MakeFunction( ReturnDBsetup, [],
        {
        HideWindow( Session1 );
        ShowWindow( SESSION );
} );

/*****************************************
**** FUNCTION: SelectTool
*****************************************/
MakeFunction( SelectTool, [],
        {
        DBOpenFile( Tool:File );
        ClearList( Tool:AvailSelection );
        Let [n DBGetNumberOfRows( )]
        {
        For PosRow From 1 To n
        Do {
        AppendToList( Tool:AvailSelection,
        DBReadCell( PosRow, 1 ) );
        };
        };
        Tool:Name = PostMenu( "Tool Selection", Tool:AvailSelection );
        DBSetRowPosition( GetElemPos( Tool:AvailSelection, Tool:Name ) );
        SetValue( Tool:CTaylor, DBReadField( 2 ) );
        Tool:NTaylor = DBReadField( 3 );
        DBCloseFile( Tool:File );
} );

/*****************************************
**** FUNCTION: SelectMaterial
*****************************************/
MakeFunction( SelectMaterial, [],
        {
        DBOpenFile( Material:File );
        ClearList( Material:AvailSelection );
        Let [n DBGetNumberOfRows( )]
        {
```

```
For PosRow From 1 To n
Do {
AppendToList( Material:AvailSelection,
DBReadCell( PosRow, 1 ) );
};
};
Material:Name = PostMenu( "Material Selection", Material:AvailSelection );
DBSetRowPosition( GetElemPos( Material:AvailSelection, Material:Name ) );
        DBCloseFile( Material:File );
} );

/*****************************************
**** FUNCTION: GotoMonitoring
*****************************************/
MakeFunction( GotoMonitoring, [],
        {
        ShowWindow( Session2 );
} );

/*****************************************
**** FUNCTION: ReturnMonitoring
*****************************************/
MakeFunction( ReturnMonitoring, [],
        {
        If ( TimeReference:Clock_Status #= On )
        Then PostMessage( "Monitoring is Running" )
        Else {
       HideWindow( Session2 );
       ShowWindow( SESSION );
        };
} );

/*****************************************
**** FUNCTION: MonitorOn
*****************************************/
MakeFunction( MonitorOn, [],
        {
        If (TimeReference:Clock_Status #= Off)
```

```
Then {
Taylor_Equation ();
ClearList(H_ART2:Wear_Class_History);
ClearList(H_SOM:Wear_Class_History);
ClearList(H_SOM:Time_History) ;
ClearList(H_ART2:Time_History) ;
TimeReference:Time_On = 0;
TimeReference:Time_Off=0;
TimeReference:Clock_Running = Off;
TimeReference:Clock_Status = On;
Perf_ART2:Pnn=0.0;
Perf_ART2:Rh=1;
Perf_SOM:Pnn=0.0;
Perf_SOM:Rh=1;
ResetImage(LinePlot1) ;
ResetClock( );
SetTimer( 1, 0, 5 );
}
Else PostMessage("Monitoring is Already Being Performed");
} );

/*****************************************
**** FUNCTION: MonitorOff
*****************************************/
MakeFunction( MonitorOff, [], {
        KillTimer( 1 );
        TimeReference:Clock_Status = Off;
} );

/*****************************************
**** FUNCTION: DBopenSOM
*****************************************/
MakeFunction( DBopenSOM, [],
        Let [file SelectFile( "SOM Database Name", "*.dat" )] {
        SOM:Weights_File_Source = file;
        ResetImage( Edit16 );
} );
```

```
/*****************************************
**** FUNCTION: DBopenART2
*****************************************/
MakeFunction( DBopenART2, [],
        Let [file SelectFile( "ART2 Database Name", "*.dat" )] {
        ART2:Weights_File_S = file;
        ResetImage( Edit17 );
} );

/*****************************************
**** FUNCTION: Help_DepthCut
*****************************************/
MakeFunction( Help_DepthCut, [],
PostMessage("Defines the value for depth of cut in mm") );

/*****************************************
**** FUNCTION: GotoModuleTest
*****************************************/
MakeFunction( GotoModuleTest, [], {
        ShowWindow( Session3 );
} );

/*****************************************
**** FUNCTION: ReturnModuleTest
*****************************************/
MakeFunction( ReturnModuleTest, [], {
        HideWindow( Session3 );
        ShowWindow( SESSION );
} );

/*****************************************
**** FUNCTION: Acquire_Data
*****************************************/
MakeFunction( Acquire_Data, [], SetValue (Inputs:lista,DAQ_PROC
        (Inputs:Sampling_Frq,Inputs:Band_1_L,outputs:Band_1_R,Inputs:Band_2_L,In
        puts:Band_2_R,Inputs:data2file,Inputs:da    ta2backupfile,Inputs:file_on)) );

/*****************************************
```

R.G. Silva 1997

```
**** FUNCTION: Run_SOM
*****************************************/
MakeFunction( Run_SOM, [],
        {
        SetValue(SOM:Class_Pos,SOM(SOM:Weights_File_Source,Inputs:data2file,
        SOM:aux_file,0,Sound:Abs_Deviation,Sound:Mean_Value,Sound:Kurtosis,Sou
        nd:Skewness,Sound:Frq_Band_1,Sound:Frq_Band_2,Vibration:Abs_Deviation,
        Vibration:Mean_Value,Vibration:Kurtosis,Vibration:Skewness,Vibration:Frq_Ba
        nd_1,Vibration:Frq_Band_2,Forces:Mean_Feed_Force,Forces:Mean_Tang_For
        ce,Current:Mean_Current));
} );

/*****************************************
**** FUNCTION: Run_ART2
*****************************************/
MakeFunction( Run_ART2, [],ART2:Class_Alloc = FormatValue ("%d",ART2
        (ART2:Weights_File_S,Inputs:data2file,ART2:Vigilance,0,Sound:Abs_Deviation
        ,Sound:Mean_Value,Sound:Kurtosis,Sound:Skewness,Sound:Frq_Band_1,Sou
        nd:Frq_Band_2,Vibration:Abs_Deviation,Vibration:Mean_Value,Vibration:Kurto
        sis,Vibration:Skewness,Vibration:Frq_Band_1,Vibration:Frq_Band_2,Forces:Me
        an_Feed_Force,Forces:Mean_Tang_Force,Current:Mean_Current)) );

/*****************************************
**** FUNCTION: Taylor_Equation
*****************************************/
MakeFunction( Taylor_Equation, [],
        {
        TaylorsEquation:Taylor_Tool_Life=TAYLOR(Tool:NTaylor,CutCond:CutSpeed,
        Tool:CTaylor);
} );

/*****************************************
**** FUNCTION: Help_Simulate
*****************************************/
MakeFunction( Help_Simulate, [],
PostMessage("Simulates the monitoring process fully integrated") );

/*****************************************

R.G. Silva 1997
```

```
**** FUNCTION: Veredict
*****************************************/
MakeFunction( Veredict, [],
        {
        Run_ART2 ();
        ART2:Prediction=GetNthElem(ART2:Wear_Class,GetElemPos(ART2:n_Class,
        ART2:Class_Alloc));
        Run_SOM ();
        SOM:Prediction=WEAR_Predictor(SOM:aux_file2,FormatValue(%d",GetNthEle
        m(SOM:Class_Pos,1)),FormatValue ("%d",GetNthElem(SOM:Class_Pos,2)));
        Taylor_Prediction();
        ToolState:Wear_Level = NULL;
        ForwardChain ([NOASSERT], State_of_Tool);
        ResetImage(LinePlot1);
} );

/*****************************************
**** FUNCTION: GotoTrain
*****************************************/
MakeFunction( GotoTrain, [],
ShowWindow(Session5) );

/*****************************************
**** FUNCTION: ReturnTrain
*****************************************/
MakeFunction( ReturnTrain, [],
HideWindow(Session5) );

/*****************************************
**** FUNCTION: TrainART2
*****************************************/
MakeFunction( TrainART2, [],
{
        ClearList(ART2:n_Class) ;
        ClearList(ART2:Wear_Class) ;
        SetValue(Global:aux3,ART2(ART2:Weights_File_Source,Inputs:data2file,ART2
        :Vigilance,ART2:Training_Limit,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0));
        For n From 1 To LengthList(Global:aux3) By 2 Do {
```

```
        AppendToList(ART2:n_Class,FormatValue("%d",GetNthElem(Global:aux3, n)));
        AppendToList(ART2:Wear_Class,GetNthElem(Global:aux3, n+1));
        };
        ART2:n_Clusters=FormatValue ("%d",LengthList(ART2:n_Class));
        PostMessage("Training DONE");
} );

/*****************************************
**** FUNCTION: TrainSOM
*****************************************/
MakeFunction( TrainSOM, [],
{
        SOM(SOM:Weights_File,Inputs:data2file,SOM:aux_file,SOM:Training_Limit,0,0
        ,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
        MAP_WEAR (SOM:aux_file,SOM:aux_file2);
        PostMessage("Training DONE");
} );

/*****************************************
**** FUNCTION: daq
*****************************************/
MakeFunction( daq, [],
{
        Inputs:file_on=1;
        Acquire_Data();
        PostInputForm("Flank Wear Reading", WearState, Wear_Measure, Reading) ;
        OpenWriteFile(Inputs:data2file, APPEND);
        WriteLine(WearState:Wear_Measure);
        WriteLine(TimeReference:TimerCounter);
        CloseWriteFile() ;
        Inputs:file_on=0;
} );

/*****************************************
**** FUNCTION: TimerFunc
**** Evaluation of Spindle Current
*****************************************/
MakeFunction( TimerFunc, [],


R.G. Silva 1997
```

```
{
        If (Inputs:file_on==1) Then {
        PostInputForm("Flank Wear Reading", WearState, Wear_Measure, Reading) ;
        OpenWriteFile(Inputs:data2file, APPEND);
        WriteLine(WearState:Wear_Measure);
        WriteLine(TimeReference:Time_On);
        CloseWriteFile() ;
        };
        If ( Current:Mean_Current > Current:Current_Trigger) Then {
        If (TimeReference:Clock_Running #= Off) Then {
        TimeReference:Time_Off=GetClock() - TimeReference:Time_On;
        TimeReference:Clock_Running = On;
        };
        TimeReference:Time_On = GetClock() - TimeReference:Time_Off;
        If(TimeReference:Time_On/TimeReference:SamplingFloor(TimeReference:Tim
        e_On/TimeReference:Sampling) < 0.5) Then {
        Veredict();
        };
        }
        Else {
        TimeReference:Clock_Running = Off;
        };
} );

SetFunctionComment( TimerFunc, "Evaluation of Spindle Current" );

/*****************************************
**** FUNCTION: Taylor_Prediction
*****************************************/
MakeFunction( Taylor_Prediction, [],
{
        TaylorsEquation:Taylor_Predict =
        TimeReference:Time_On*0.3/(TaylorsEquation:Taylor_Tool_Life*60);
} );

/*****************************************
**** FUNCTION: Correlation
*****************************************/


R.G. Silva 1997
```

```
MakeFunction( Correlation, [H Perf],
{
        If (LengthList(H:Wear_Class_History)>2) Then {
        Global:a=0;
        Global:b=0;
        Global:c=0;
        Global:d=0;
        Global:e=0;
        Let [n Min(LengthList(H:Wear_Class_History),Settings:Np)] {
        For i From 1 To n Do {
        Global:a=Global:a+GetNthElem(H:Wear_Class_History,LengthList(H:Wear_Cla
        ss_History)-i+1)*GetNthElem(H:Wear_Class_History,LengthList(H:Wear_Class
        _History)-i+1) ;
        Global:b=Global:b+GetNthElem(H:Wear_Class_History,LengthList(H:Wear_Cla
        ss_History)-i+1);
        Global:c=Global:c+GetNthElem(H:Time_History,LengthList(H:Wear_Class_History)-
        i+1)*GetNthElem(H:Time_History,LengthList(H:Wear_Class_History)-i+1) ;
        Global:d=Global:d+GetNthElem(H:Time_History,LengthList(H:Wear_Class_History)-  i+1) ;
        Global:e=Global:e+GetNthElem(H:Wear_Class_History,LengthList(H:Wear_Cla
        ss_History)-i+1)*GetNthElem(H:Time_History,LengthList(H:Wear_
        Class_History)-i+1) ;
        };
        };
        If (Abs(Min(LengthList(H:Wear_Class_History),Settings:Np)*Global:e-Global:b*
        Global:d) < 0.00000001) Then Perf:Rh = 1
        Else Perf:Rh=(Min(LengthList(H:Wear_Class_History),Settings:Np)*Global:e-
        Global:b*Global:d)/Sqrt((Min(LengthList(H:Wear_Class_History),Settings:Np)*G
        lobal:a-Global:b*Global:b)*(Min(LengthList(H:Wear_Class_History), Settings:Np)*Global:c-
        Global:d*Global:d));
        };
} );
□
```

References

Agogino, A.M., Srinivas, S. and Schneider, K.M. (1988) Multiple Sensor Expert System for Diagnostic Reasoning, Monitoring and Control of Mechanical Systems. *Mechanical Systems and Signal Processing* **2**(2), 165-185.

Anthony, D. (1993) The Use of Artificial Neural Networks in Medicine. *E-Mail - Warwick University*, 1-18.

Arnold, R.N. (1946) The Mechanism of Tool Vibration on the Cutting of Steel. *Proceedings Institution of Mechanical Engineers*, **159**, 261-285.

Au, Y.H.J., Hale, K.F., Jones, B.E. and Mardapittas, A. (1989) *Development of an Expert System For Tool Wear Monitoring*. Brunel Centre for Manufacturing Metrology, Brunel University, Uxbridge, Middlesex, UK.

Bailey, D. and Thompson, D. (1990) How to Develop Neural-Network Applications. *AI Expert*, June, 28-47.

Barrios, L.J., Ruiz, A., Bustos, P. and Ilbanez, A. (1994) Building the Knowledge Base of a Production System from the Raw Data of a Multi-Sensor System. *Sensors and Actuators* **42**, 599-603.

Betancourt, F., Cham, J.C., Barrios, L.J., Guinea, D. and Ruiz, A. (1990) A Comparative Analysis of Expert Systems for Tool Wear Monitoring in Cutting Processes. *International Conference on Artificial Intelligence Applications and Neural Networks*, Zurich, June.

Bezdek, J.C. (1993) A Review of Probabilistic, Fuzzy, and Neural Models for Pattern Recognition. *Journal Intelligent and Fuzzy Systems* **1**(1), 1-25.

Bhattacharyya, A. and Ghosh, A. (1964) Diffusion Wear of Cutting Tools. *Proceedings of the 5th International M.T.D.R. Conference, University Birmingham* September, 225-242.

Bonifacio, M.E.R. and Diniz, A.E. (1994) Correlating Tool Wear, Tool Life, Surface Roughness and Tool Vibration in Finish Turning with Coated Carbide Tools. *Wear* **173**, 137-144.

Braun, S., Rotberg, J. and Lenz, E. (1987) Signal Processing for Single Tooth Milling Monitoring. *Mechanical Systems and Signal Processing* **1**, 185-196.

Brunn, P. (1981) Machine Condition Monitoring. *Noise Vibration Control Worldwide* **12**, 249-251.

Burke, L. (1989) *Automated Identification of Tool Wear States in Machining Processes: An Application of Self-Organizing Neural Networks*, PhD Thesis, University of California, Berkley, USA.

Burke, L.I. and Rangwala, S. (1991) Tool Condition Monitoring in Metal Cutting: A Neural Network Approach. *Journal of Intelligent Manufacturing* **2**, 269-280.

Carpenter, G.A. and Grossberg, S. (1987) ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics* **26**, 4919-4930.

Chappell, G.J. and Taylor, J.G. (1993) The Temporal Kohonen Map. *Neural networks* **6**, 441-445.

Chitra, S.P. (1993) Use Neural Networks for Problem Solving. *Chemical Engineering Progress* April, 44-52.

Chiu, S.L. (1994) Fuzzy Model Identification Based on Cluster Estimation. *Journal Intelligence and Fuzzy Systems* **2**, 267-278.

Choi, G.S., Wang, Z.X. and Dornfeld, D.A. (1990) Development of an Intelligent On-Line Tool Wear Monitoring System for Turning Operations. *Japan-USA Symposium on Flexible Automation*, ISCIE Kyoto Japan, 683-690.

Chubb, J.P. and Billingham, J. (1980) Coated Cutting Tools - A Study of Wear Mechanisms in High Speed Machining. *Wear* **61**, 183-193.

Colding, B. and König, W. (1971) Validity of the Taylor Equation in Metal Cutting. *Annals CIRP* **XVIV**, 793-812.

Colwell, L.V. (1971) Methods for Sensing the Rate of Tool Wear. *Annals CIRP* **XVIV**, 647-651.

Crain, I.K. (1970) Computer Interpolation and Contouring of Two-Dimensional Data: A Review. *Geoexploration* **8**, 71-86.

Chryssolouris, G. and Domroese, M. (1989) An Experimental Study of Strategies for Integrating Sensor Information in Machining. *Annals CIRP* **38**(1), 425-428

Cser, L., Lange, K., Geiger, M. and Kals, J.A.G. (1993) Tool Life and Tool Quality in Bulk Metal Forming. *WIRE* **43**(1), 59-71.

Dan, L. and Mathew, J. (1990) Tool Wear and Failure Monitoring Techniques for Turning - A Review. *International Journal Machine Tools and Manufacturing* **30**(4), 579-598.

Danai, K. and Ulsoy, A.G. (1987) An Adaptive Observer For On-Line Tool Wear Estimation In Turning, Part 1: Theory. *Mechanical Systems and Signal Processing* **1**(2), 211-225.

Danai, K., Nair, R. and Malkin, S. (1992) An Improved Model for Force Transients in Turning. *Journal of Engineering for Industry* **114**, 400-403.

Das, A.B.S., Chattopadhyay and Murthy, A.S.R. (1996) Force Parameters for On-Line Tool Wear Estimation: A Neural Network Approach. *Neural Networks* **9**(9), 1639-1645.

Davies, A. (1994) The Intelligent Machine. *Manufacturing Engineer* **73**(4), 182-185.

Davis, J.C. (1986) *Statistics and Data Analysing in Geology*. John Wiley & Sons, 2nd Edition

Denoeux, T. and Lengelle, R. (1993) Initializing Backpropagation Networks With Prototypes. *Neural Networks* **6**, 351-363.

DeSieno, D. (1988) Adding a Conscience to Competitive Learning. *IEEE Proceedings on Numerical Networks* **1**, 117-124.

Dodhiawala, R., Sridharan, N.S., Raulefs, P. and Pickering, C. (1989) Real-Time AI Systems: A Definition and An Architecture. *Proceedings International Joint Conference On Artificial Intelligence*, 256-261.

Dornfeld, D.A. (1990) Neural Network Sensor Fusion for Tool Condition Monitoring. *Annals CIRP* **39**(1), 101-105.

Dornfeld, D.A., König, W. and Ketteler, G. (1993) Akteller Stand von Werkzeug- Und ProzeBuberwachung bei der Zerspanung. *International CIRP/VDI - Conference on New Developments in Cutting*, Dusseldorf/Germany 988, 363-376.

Du, R., Elbestawi, M.A. and Wu, S.M. (1995) Automated Monitoring of Manufacturing Processes, Part 1: Monitoring Methods. *Journal Engineering for Industry* **117**, 121-132.

Eysenck, M.W. and Keane, M.T. (1995) *Cognitive Psychology - A Student's Handbook*, Lawrence Erlbaum Associates Ltd, 3rd Edition.

Ezugwu, E.O., Arthur, S.J. and Hines, E.L. (1995) Tool-Wear Prediction Using Artificial Neural Networks. *Journal of Materials Processing Technology* **49**, 255-264.

Fenton, R.G. and Oxley, P.L.B. (1967) Predicting Cutting Forces At Super High Cutting Speeds From Work Material Properties And Cutting Conditions. *Proceedings of the 8th International M.T.D.R. Conference*, University Manchester, September, 247-258.

Fikes, R. and Kehler, T. (1985) The Role of Frame-Based Representation in Reasoning. *Communications ACM* **28**, 904-920.

Filippi, A. and Ippolito, R. (1969) Adaptive Control In Turning: Cutting Forces and Tool Wear Relationship for P10, P20, P30 Carbides. *Annals CIRP* **17**, 377-385.

Finnie, I. (1956) Review of the Metal-Cutting Analyses of the Past Hundred Years. *Spring Meeting ASME*, Portland, Oregon, March 18-21, 715-721.

Flachs, G.M., Jordan, J.N., Beer, C.L. and Scott, D.R. (1990) Feature Space Mapping for Sensor Fusion. *Journal of Robotic Systems* 7(3), 373-393.

Guinea, D., Barrios, L.J., Ruiz, A. and Betancourt, F. (1990) Multisensor Information Integration. *Sensor Review* July, 133-136.

Halgamuge, S.K., Grimm, C. and Glesner, M. (1994) Fuzzy - Neural Clustering Methods for Real - Time Classification. *ELITE - Foundation, EUFIT'94, Promenade* 9, D-52076 Aachen 528-533.

Harmon, P. and King, D. (1985) *Expert Systems: Artificial Intelligence in Business*. New York, John Wiley.

Hastings, W.F. and Oxley, P.L.B. (1976) Predicting Tool Life from Fundamental Work Material Properties and Cutting Conditions. *Annals CIRP* **25**(1), 33-38.

Hayes-Roth, F., Waterman, D.A. and Lenat, D.B. (1983) *Building Expert Systems*. Addison-Wesley Publishing Company, Inc.

Hebb, D.O. (1949) *The Organisation of Behaviour*. New York, John Wiley.

Heck, L.P. (1993) Signal Processing Research in Automatic Tool Wear Monitoring. *Proceedings ICASSP IEEE International Conference Acoustic, Speed and Signal Processing* **1**, 55-58.

Hopfield, J.J. (1982) Neural Networks and Physical Systems With Emergent Collective Computational Abilities. *Proceedings National Academy Science USA* **79**, 2554-2558.

Hopgood, A.A., Woodcock, N., Hallan, N.J. and Picton, P.D. (1993) Interpreting Ultrasonic Images Using Rules, Algorithms and Neural Networks. *European Journal of NDT* **2**(4), 135-149.

ISO 3685 (1977). *Tool-Life testing with single-point turning tools.*

Jantunen, E., Jokinen, H. and Holmberg, K. (1995) Monitoring of Tool Wear. *Proceedings of the 8th International Congress on Condition Monitoring and Diagnostic Engineering Management*, Queen's University, Kingston, Ontario, Canada 1, 143-150.

Jetly, S. (1984) Measuring Cutting Tool Wear On-Line: Some Practical Considerations. *Manufacturing Engineering* July, 55-60.

Jiang, C.Y., Zhang, Y.Z. and Xu, H.J. (1987) In-Process Monitoring of Tool Wear Stage by the Frequency Band-Energy Method. *Annals CIRP* **36**(1), 45-48.

Juneja, H. (1986) The Significance of Taylor's Equation. *Annals CIRP* **35**(1), 289-295.

Kannatey-Asibu, E. (1985) A Transport-Diffusion Equation in Metal Cutting and its Application to Analysis of the Rate of Flank Wear. *Journal of Engineering for Industry* **107**, 81-89.

Kohonen, T. (1984) *Self-Organization and Associative Memory*, 3ed. Berlin/ Heidelberg, Springer.

Kohonen, T. (1988) An Introduction to Neural Computing. *Neural Networks* **1**, 3-16.

Kohonen, T. (1990) The Self Organizing Map. *Proceedings of IEEE* **78**, 1464-1480.

Kohonen, T. (1993) Physiological Interpretation of the Self-Organizing Map Algorithm. *Neural Network* **6**, 895-905.

Kohonen, T., Oja, E., Simula, O., Visa, A. and Kangas, J. (1996) Engineering Applications of the Self-Organizing Map. *Proceedings IEEE* **84**(10) 1358-1384.

König, W., Langhamer, K. and Schemmel, H. (1972) Correlation Between Cutting Force Components and Tool Wear. *Annals CIRP* **21**(1), 19-20.

Koren, Y., Ulsoy, A.G. and Danai, K. (1986) Tool Wear and Breakage detection Using a Process Model. *Annals CIRP* **35**(1), 284-288.

Kosko, B. and Isaka, I. (1993) Fuzzy Logic. *Scientific American* July, 62-67.

Laffey, T.J., Cox, P.A., Scmidt, J.L., Rao, S.M. and Read, Y. (1988) Real-Time Knowledge-Based Systems. *AI Magazine* **9**, 27-45.

Lau, W.S. and Rubenstein, C. (1978) The Influence of Tool Geometry on the Taylor Constant. *International Journal Machine Tool Design Research* **18**, 59-66.

Lee, J.H., Kim, D.E. and Lee, S.J. (1996) Application of Neural Networks to Flank Wear Prediction. *Mechanical Systems and Signal Processing* **10**(3), 265-276.

Lee, L.C. (1986) A Study of Noise Emission for Tool Failure Prediction. *International Journal Tool Design Research* **26**, 205-215.

Lee, L.C., Lee, K.S. and Gan, C.S. (1989) On the Correlation Between Dynamic Cutting Force and Tool Wear. International *Journal Machine Tools Manufacturing* **29**(3), 295-303.

Leem, C.S., Dornfeld, D.A. and Dreyfus, S.E. (1995) A Customized Neural Network for Sensor Fusion in On-Line Monitoring of Cutting Tool Wear. *Journal of Engineering for Industry, ASME* **117**, 152-159.

Lieslehto, J. and Koivo, H.N. (1991) An Expert System for Interaction Analysis of Multivariable Systems. *International journal of Adaptive Control and Signal Processing* **5**, 41-62.

Lieslehto, J., Tanttu, J.T. and Koivo, H.N. (1993) An Expert System for Multivariable Control Design. *Automatica* **29**, 953-968.

Lim, G.H. (1995) Tool-Wear Monitoring in Machine Turning. *Journal of Materials Technology* **51**, 25-36.

Lin, G.C.I., Mathew, P. and Watson, A.R. (1982) Predicting Cutting Forces For Oblique Machining Conditions. *Proceedings International Mechanical Engineers* **196**, 141-148.

Lin, R.M. and Ewins, D.J. (1993) Chaotic Vibration of Mechanical Systems With Backlash. *Mechanical Systems and Signal Processing* 7(3), 257-272.

Lippmann, R.P. (1987) An Introduction to Computing With Neural Nets. *IEEE ASSP Magazine* April, 4-22.

Machinery [Editor] (1989) Sound Out the Merits of Machine Monitoring. *Machinery and Production Engineering* **1**(15), 24-31.

Mackinnon, R., Wilson, G.E. and Wilkinson, A.J. (1986) Tool Condition Monitoring Using Multi-Component Force Measurements. *Proceedings 6th International Machine Tool Design & Research Conference*, 317-324.

Martin, P., Mutel, B. and Drapier, J.P. (1974) Influence of Lathe Tool Wear on the Vibrations Sustained in Cutting. *Proceedings 15th MTDR Conference*, 251-257.

Martin-del-Brio, B. and Serrano-Cinca, C. (1993) Self-Organizing Neural Networks for the Analysis and Representation of Data: Some Financial Cases. *Neural computing & Applications* **1**, 193-206.

Marui, E., Kato, S., Hashimoto M. and Yamada, T. (1988) The Mechanisms of Chatter Vibration in a Spindle-Workpiece System: Part 1 - Properties of Self-Excited Chatter Vibration in Spindle-Workpiece System. *Journal of Engineering for Industry, ASME* **110**, 236-241.

McCulloch, W.S. and Pitts, W. (1943) *A Logical Calculus of the Idea Imminent in Nervous Activity*. Bulletin of the Mathematical Biophysics, **5**, 115-133.

McGraw, K.L. and Cliffs, K.H. (1989) *Knowledge Acquisition: Principles and Guidelines*. Prentice-Hall, Inc.

McLain, D.H. (1976) Two Dimensional Interpolation from Random Data. *The Computer Journal* **19**(2), 178-181.

McNulty, G.J. and Popplewell, N. (1977) Health Monitoring of Cutting Tools Through Noise Spectra. *Ist Joint Polytechnic Symposium on Manufacturing Engineering* June 14&15, E41-6.

Medsker, L.R. (1994) *Hybrid Neural Network and Expert System*. Kluwer Academic Publishers.

Medsker, L.R. (1996) Microcomputer Applications of Hybrid Intelligent Systems, Journal of Network and Computer Applications, **19**, 213-234.

Micheletti, G.F., König, W. and Victor, H.R. (1976) In Process Tool Wear Sensors for Cutting Operations. *Annals CIRP* **25**, 483-496.

Milne, R.W. (1988) *Monitoring Process Control Systems*. Edinburgh University 55-63.

Nagy, Z. and Szalay, T. (1993) Control of Multisensor Systems by Neural Networks, *Department of Manufacture Engineering* - Technical University of Budapest.

Nair, R., Danai, K. and Malkin, S. (1992) Turning Process Identification Through Force Transients. *Journal of Engineering for Industry* **114**, 1-7.

National Critical Technologies Panel (1993) PB93-213742. The White House, Washington DC, USA.

Nisbett, R.E. and Wilson, T.D. (1977) Telling More Than We Can Know: Verbal Reports on Mental Processes. *Psychological Review* **84**(3), 231-259.

Novak, A. and Ossbahr, G. (1986) Reliability of the Cutting Force Monitoring in FMS-Installations. *Proceedings of the 26th International Machine Tool Design & Research*, 325-329.

Okafor, A.C., Marcus, M. and Tipirneni, R. (1991) Multiple Sensor IntegrationVia Neural Networks for Estimating Surface Roughness and Bore Tolerance in Circular End Milling - Part 1: Time Domain, Condition Monitoring and Diagnostic Technology, **2**, 49-57.

OMROM (1992) Fuzzy Guide Box & "Think it Over - Fuzzy Application ideas".

Opitz, D.W. and Shavlik, J.W. (1993) Heuristically Expanding Knowledge-Based Neural Networks. *Neural Networks* 1360-1365.

Opitz, H. and König, W. (1967) On The Wear of Cutting Tools. *Proceedings of the 8th International M.T.D.R. Conference*, University Manchester, September 173-190.

Oraby, S.E. and Hayhurst, D.R. (1991) Development of Models for Tool Wear Force Relationship in Metal Cutting. *International Journal of Mechanical Science* **33**(2), 125-138.

Oxley, P.L.B. (1989) *The Mechanics of Machining: An Analytical Approach to Assessing Machinability*. Ellis Horwood Limited.

Pandit, S.M. and Kashou, S. (1982) A Data Dependent Systems Strategy of On-Line Tool Wear Sensing. *Journal of Engineering for Industry* **104**, 217-223.

Pedersen, K.B. (1990) Wear Measurement of Cutting Tools By Computer Vision. *International Journal of Machine Tools Manufacturing* **30**, 131-139.

Petrie, A.M., Shihra, T.S. and Ismail, A. (1989a) The Development of an Automated System for On-line Tool Wear Monitoring. *1st International Machinery Monitoring Conference*, Las Vegas, September 1-7.

Petrie, A.M., Sihra, T. and Ismail, A. (1989b) An Experimental Investigation of the Use of Vibration Measurement for On-Line Tool Wear Monitoring. *International Conference Noise & Vibration*, August, H-8/H-15.

Pilafidis, E.J. (1971) Observations on Taylor "n" Values Used in Metal Cutting. *Annals CIRP* **XVIV**, 571-577.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992) *Numerical Recipes in C: The art of Scientific Computing*, 2nd Edition, Cambridge University Press.

Qin, S.J. and Rajagopal, B. (1993) Combining Statistics and Expert Systems with Neural Networks for Empirical Process Modelling. *ADV Instrumentation Control International Conference Exhibition* **48**, 1711-1720.

Rahman, M., Zhou, Q. and Hong, G.S. (1995) On-Line Cutting State Recognition in Turning Using A Neural Network. *International Journal Advanced Manufacturing Technology* **10**, 87-92.

PAGE NUMBERING AS ORIGINAL

Wilcox, S.J., Reuben, B., Rapti, A. and Psonis, S. (1993) Mould Breakage - Wear and Tear Recognition when Milling with the Aid of an Expert System. *International CIRP/VDI - Conference on New Developments in Cutting*, Dusseldorf/Germany 988, 377-399.

Wolf, W. and Magadomy, P. (1981) Feed Force Monitoring For Operation. *Paper IQ-161 SME Dearborn MI IQ81*(161), 1-12.

Wu, Y. and Du, R. (1996) Feature Extraction and Assessment Using Wavelet Packets for Monitoring of Machining Processes. *Mechanical Systems and Signal Processing* **10**(1), 29-53.

Ya, W., Shiqiu, K., Shuzi, Y., Qilin, Z., Shanxiang, X. and Yaozu, W. (1991) An Experimental Study of Cutting Noise Dynamics. *Machinery Dynamics and Element Vibrations* **36**, 313-318.

Yao, Y. and Fang, X.D. (1992) Modelling of Multivariate Time Series for Tool Wear Estimation in Finish-Turning. *International Journal Machine Tools Manufacturing* **32**(4), 495-508.

Zadeh. L.A. (1965) Fuzzy Sets. *Information and Control* **8**, 339-353.

Zhou, C. and Wysk, R.A. (1992) An Integrated System for Selecting Optimum Cutting Speeds and Tool Replacement Times. *International Journal Machine Tools Manufacture* **32**, 695-707.

Zimmermann, H. (1991) *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers,

Publications

1. R.G. Silva, R.L. Reuben, K.J. Baker and S.J. Wilcox, "A Neural Network Approach To Tool Wear Monitoring", 8th International Congress on Condition Monitoring and Diagnostic Engineering Management, Queen's University, Kingston, Ontario, Canada, June 26-28, 1995.

2. R.G. Silva, R.L. Reuben, K.J. Baker and S.J. Wilcox, "Tool Wear Monitoring of Turning Operations by Neural Network Classification of a Feature Set Generated From Multiple Sensors" - submitted for publication in the journal "*Mechanical Systems and Signal Processing*"

A NEURAL NETWORK APPROACH TO TOOL WEAR MONITORING

R.G. Silva[1]; R.L. Reuben[2]; K.J. Baker[1] and S.J. Wilcox[1]

[1]Department of Mechanical and Manufacturing Engineering,
University of Glamorgan, Pontypridd, Mid Glamorgan. UK
[2]Department of Mechanical Engineering,
Heriot-Watt University, Riccarton, Edinburgh. UK

Abstract

A method is presented for monitoring the progressive increase of tool wear during a turning operation by means of two Neural Network algorithms, the Self Organising Map (SOM) and Adaptive Resonance Theory (ART2). Audible sound emission, machine vibration, cutting forces and spindle current have been measured as a function of tool wear. Features obtained from these sensors which are suitable for real time application were used as inputs to the Neural Networks. Features selected were the average, absolute deviation, skewness, kurtosis and some bands of the frequency spectrum. Some of these parameters are shown to be more sensitive to the seriously worn stage. The cutting force is shown to be more sensitive than the feed force, although both show a progressive increase with tool wear. The spindle current is shown to vary with flank wear. Because of the sharp changes in the motor load during the tool entry and exit, the current sensor is an excellent detector of cut state. The two Neural Network structures are shown to be able to distinguish different wear levels.

1. Background

Successful automation of machining operations relies, to a great extent, on the ability of artificial systems to recognise process abnormalities and initiate corrective action. In the absence of human operators, this function has to be performed with sensors and associated decision-making systems which are able to interpret incoming sensor information and decide on the appropriate control action. For full automation, intelligent systems are expected to replace the knowledge, experience, and the combined sensory and pattern recognition abilities of human operators. Successful implementation of these different tasks depends on two factors: first, the quality of information received by the monitoring sensor, and second, the technique used to process this information in order to make decisions.

Wear monitoring has been performed using many different sensing techniques. These techniques include; temperature, motor current, acoustic emission (AE), audible emissions, vibration and force. Some of these have been successfully applied under laboratory conditions although industrial applications have been rather unsuccessful. Clearly, the quality of the sensor information is adequate to make judgements of the state of wear in idealised conditions but much work has to be performed in information processing and decision making in order to correctly classify the tool wear state from the available sensors. It is therefore the aim of this work to integrate some of the above mentioned sensors to extract the largest possible amount of information from the cutting process and provide an indication of the wear level.

Previous work on the relationship between audible emissions and tool wear has established that audible emissions are capable of indicating the extent of the cutting edge wear, Weller *et al.*[1]. McNulty *et al.*[2] have also highlighted the use of noise spectra for tool life evaluation applied to several cutting processes and have found significant changes in certain frequency bands that appear to be characteristic of wear in certain cutting processes. Lee [3] found that, during turning the machine noise exhibited a wear related change of sound pressure level (SPL) at certain frequencies (4 - 6 kHz) for several materials. A drop in the SPL before the tertiary zone (third and last stage of wear) was suggested as an end of tool-life predictor. Experiments carried out by Wu Ya *et al.* [4] using two different types of turning tool showed that both the tool angle and the cutting speed exerted no great influence on the average cutting noise.

Vibration has also been used to recognise the wear state of a tool whilst turning [5,6] and the main advantage of this method is its ease of application.

Taking into consideration previous research (e.g. Jiang *et al.* [6]) vibration has been chosen in this work as a secondary source of information because of the correlation between machine tool vibration and tool wear that have been demonstrated successfully in the laboratory. The vibrations arising from the shearing action of the tool is transmitted to the base of the machine where they are transduced by the accelerometer.

Another sensor is the motor current. This can be directly related to the power required by the turning machine and is in turn related to the total torque on the motor shaft for a given set of cutting conditions. Aside from the specific cutting forces the loads that influence the current include those associated with worn tools, misaligned components, and faulty lubrication systems [7]. Measurement of motor current has the advantage that the sensors are non intrusive, inexpensive and easy to incorporate and maintain.

A parameter that can be relatively easily measured is cutting force. Cutting forces change as the tool wears and have often been used to detect tool wear in the laboratory [8]. Some results have shown regimes where a linear relationship between these forces and tool wear exists [9]. A method of wear estimation for carbide tools, using a function of the cutting forces, has been presented in [10].

The use of more than one sensor can enhance performance of a monitoring system, because each should provide independent information on the level of tool wear, and this is particularly useful in cases where the sensor information is of limited applicability. Through appropriate analysis, the dependence of the features on changes in process conditions can be analysed [11,12]. This allows for improved reliability in making decisions on the state of tool wear under a range of machining conditions. There are several approaches to this problem, but most are statistical which unfortunately makes them slow and tedious. Another approach is the use of neural networks. These are capable of learning input patterns and the corresponding outputs according to a given learning rule. After training they can be used to classify, and thus recognise, new instances of similar patterns. One way in which such a decision-making stage could be utilised is by a combined system. These would consist of a neural network and a simple statistical analysis of the acquired data, the latter as a feature extractor for the Neural Network. This approach is used in this work, with the overall scheduling being undertaken by a Neural Network.

2. Experimental Apparatus and Procedure

Experimental work aimed at evaluating the proposed system was carried out on a turning centre although the method should be applicable to other machines and processes.

The turning operation was carried out on an MT 50 CNC Slant Bed Turning Centre, with a cutting speed of 350 m/min. and a feed rate of 0.25 mm/rev. The depth of cut was 1.0 mm. The insert was a TP25C grade coated cemented carbide (CNMG 120408) and the workpiece a free cutting mild steel (ENA), 135 mm in length.

The vibration instrumentation system consisted of an accelerometer type DJB A01/T, having a mounted axial resonant frequency of 15 kHz, magnetically fixed to the base of the Turning Centre in the direction of maximum vibration (assumed to be perpendicular to the feed). A matching charge amplifier type B&K 2626 was used to provide the required interfacing ranges for the data acquisition inputs.

Audible emissions were picked up via a microphone type ECM-1028. This was mounted on the tool post to minimise changes in signal amplitude due to cutting path changes. An amplifier was used to provide the required interface voltage level for the data acquisition board.

Two half Wheatstone bridge strain gauges were mounted on the tool in order to measure 2 components of the cutting force. Matching amplification circuits were used to achieve the required voltage levels. The current was taken directly from the CNC machine using the built in sensor.

A data acquisition board type PC-30PGL handled the acquisition of all the signals, saving the acquired data to permanent storage. Sampling was carried at 33.3 kHz with tool wear and sensor data being acquired at intervals of 2.5 min., taking into account an expected life, for each insert, of about 15 minutes.

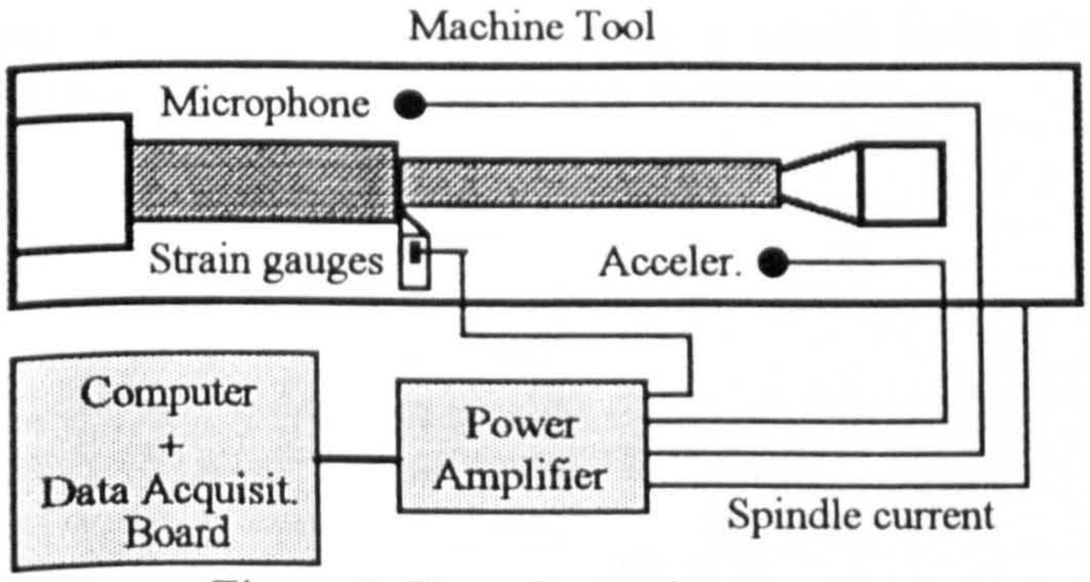


Figure 1 Experimental apparatus

3. Signal Processing and Neural Nets Used

In the present study 15 inputs (spindle current; average, absolute deviation, skewness and kurtosis for sound and vibration; tangential and feed force; 2 spectrum bands from both the sound and vibration) have been used although speed would not be compromised by a higher number of inputs. The inputs comprised data from all the sensors using statistical parameters such as average, absolute deviation, skewness and kurtosis. The number of frequency bands was selected in order not to compromise data reliability due to misuse of certain frequencies subject to changes which are not due to tool wear (e.g. machine environment). Therefore, two bands were selected from the power spectrum for the audible emissions (Section 4) and two for the machine vibration (Section 4).

Kohonen's algorithm [13] (Self Organising Map - SOM) creates a vector quantiser by adjusting weights from common input nodes (X_i) to output nodes arranged in a two dimensional grid as shown in Figure 2.

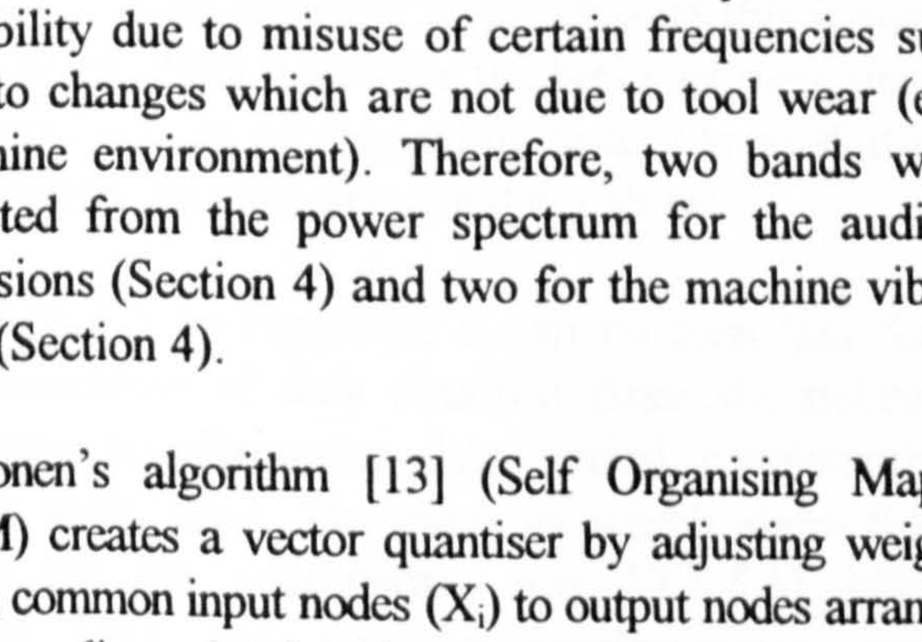


Figure 2 SOM basic structure

The aim of Kohonen's algorithm is to generate a mapping of a higher dimensional space of input signals onto a, usually two-dimensional, discrete lattice of formal neurones. The map is generated by establishing a correspondence between inputs and neurones such that the topological (neighbourhood) relationship among the inputs is reflected as faithfully as possible in the arrangement of the corresponding neurones in the lattice (Ritter *et al.* [14]).

Another type of neural network, called Adaptive Resonance Theory (ART), creates and organises categories for features and has the ability to respond immediately to experiences. The basic principles of adaptive resonance theory were first introduced by Grossberg [15], Figure 3.

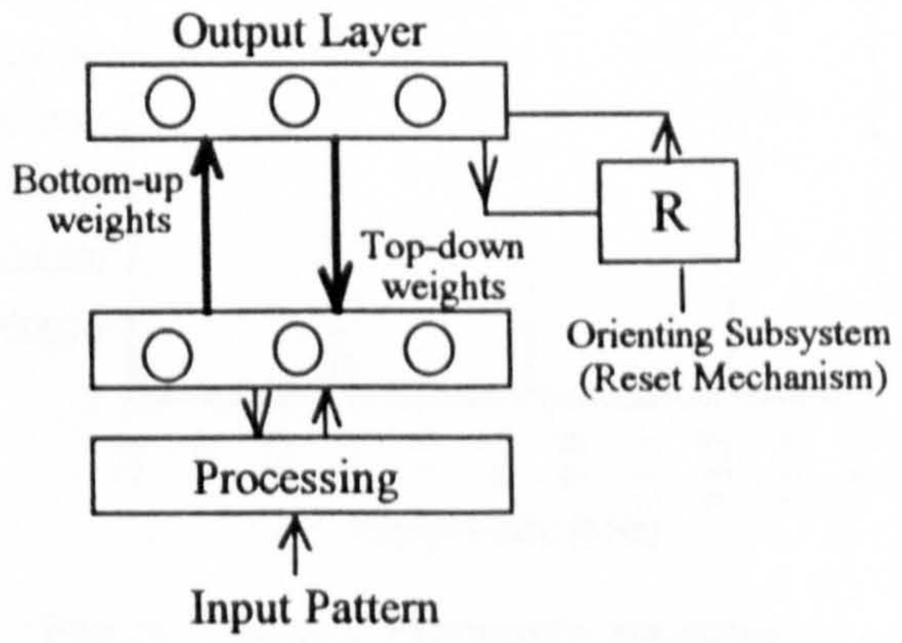


Figure 3 A representation of the ART network, Grossberg [15]

Each set of 15 inputs was used for training the neural networks. Both NNs (SOM and ART2) were provided with a module for reading and writing data from/into a file.

The above configuration was chosen for the low cost and simplicity required for a viable on-line tool wear monitoring system.

4. Analysis of Parameters Related to Flank Wear

Normally, building a neural network model involves data collection, data pre-processing, variable selection, network training, and network validation. The former is necessary because real process data often contain missing values, outliers, and noise. These data have to be conditioned or pre-processed before they can be used for network training. Feature selection is a procedure to identify important features

from all available process variables which have a significant effect on the condition to be predicted.

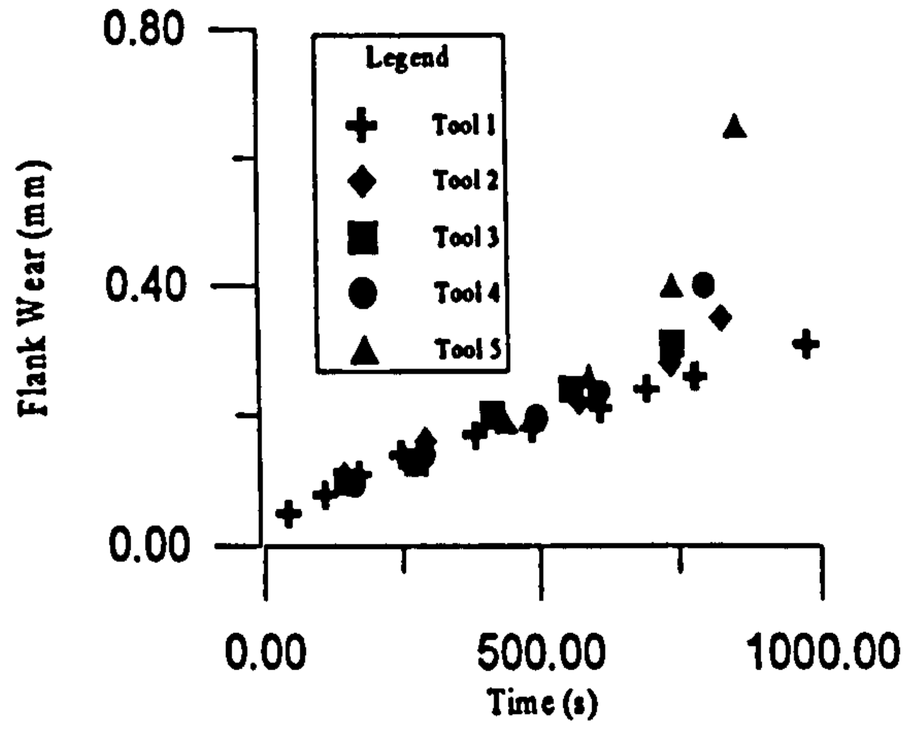


Figure 4 Flank wear versus time

Figure 4 shows the evolution of flank wear with time. This was measured using an engineering microscope and it can be seen that there is an approximately linear relationship between flank wear and time.

Analysis of the mean, standard deviation, skewness and kurtosis of the sound and vibration data did not yield a correlation with tool wear. Despite this it was decided to use them is training the NN's as there may have been features corresponding to tool wear that a simple regression analysis would not show.

From Figure 5 to Figure 10 it can be seen that the power spectrum of data obtained from the microphone and accelerometer data varied consistently with the wear level. The intervals which show this relation for the microphone, are: [3.5;5.5] kHz, [6.2;7.5] kHz and for the accelerometer, are: [3.6;5.2] kHz, [6.2;7.2] kHz.

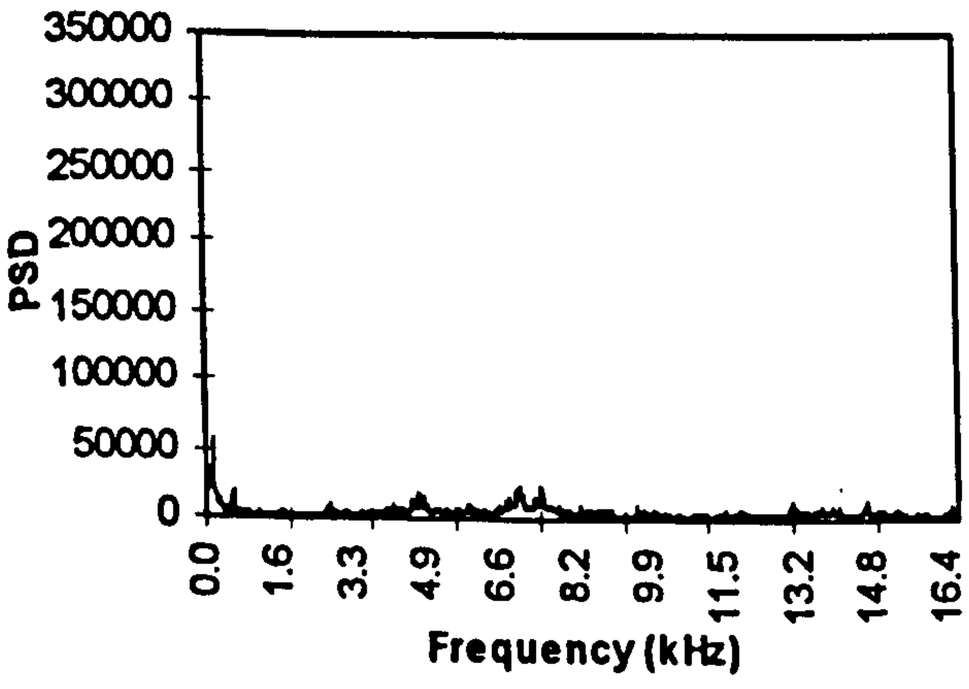


Figure 5 Sound: Frequency spectrum of a new tool

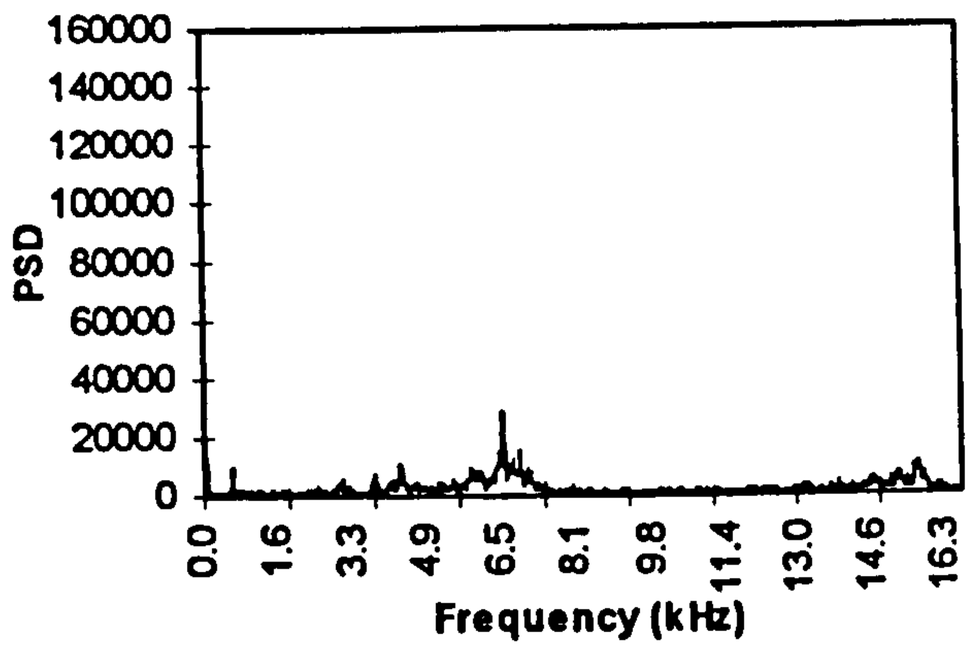


Figure 6 Vibration: Frequency power spectrum of a new tool

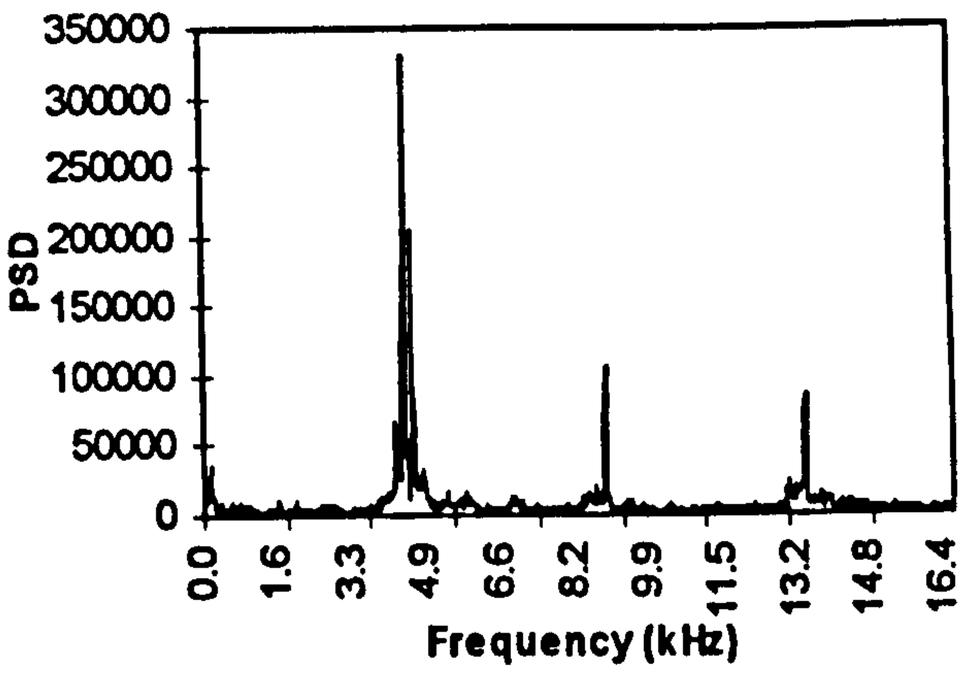


Figure 7 Sound: Frequency spectrum of a worn tool

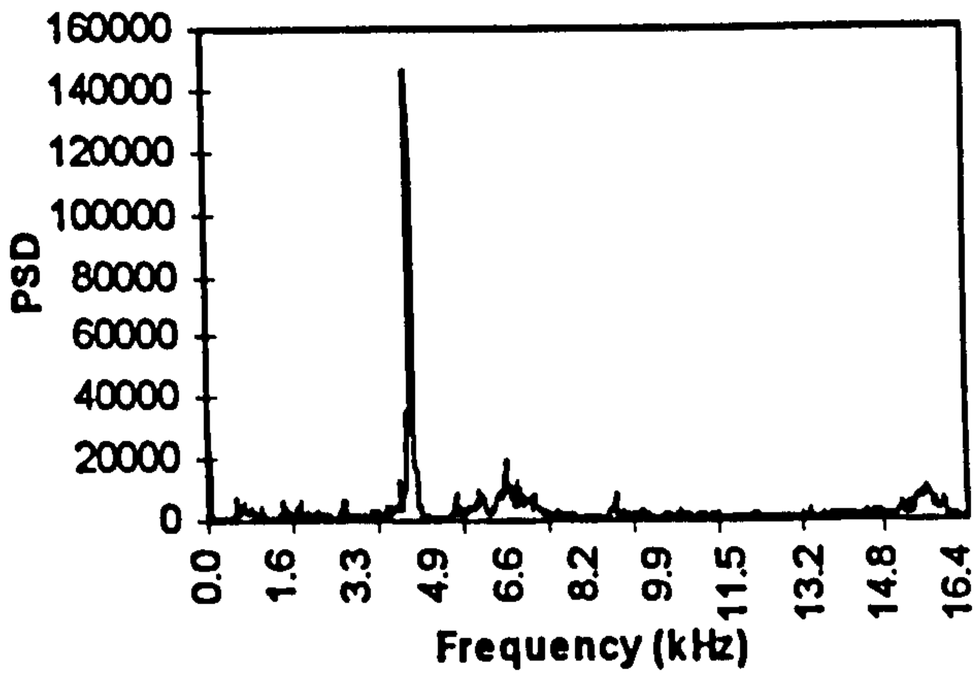


Figure 8 Vibration: Frequency spectrum of a worn tool

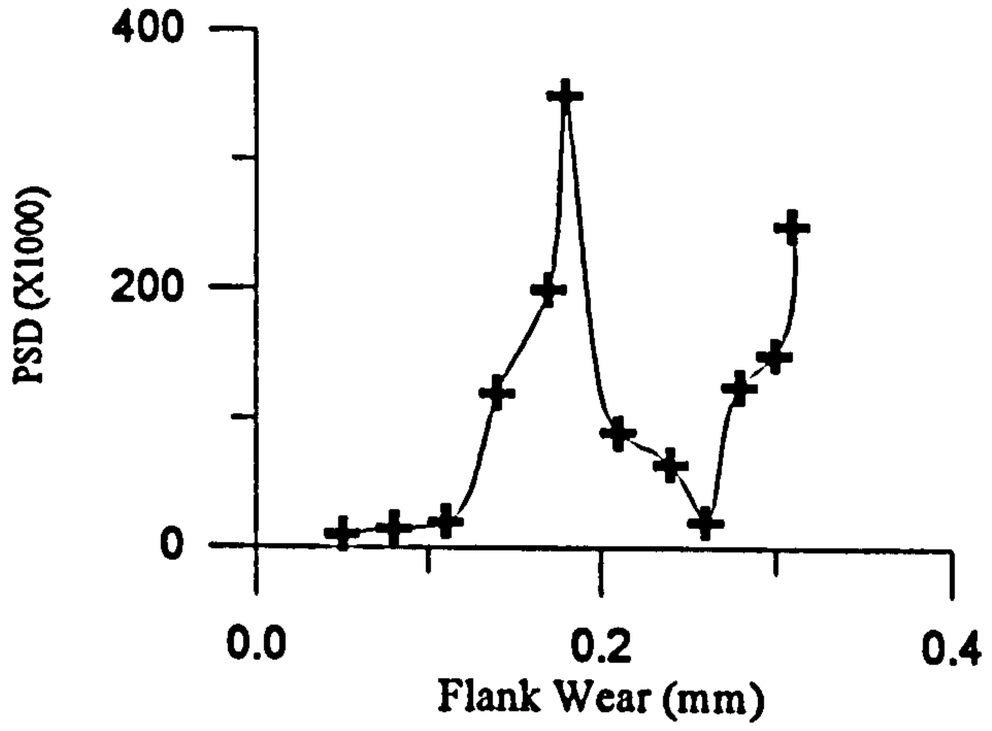


Figure 9 Frequency magnitude variation with flank wear, sound [3.5;5.5] kHz

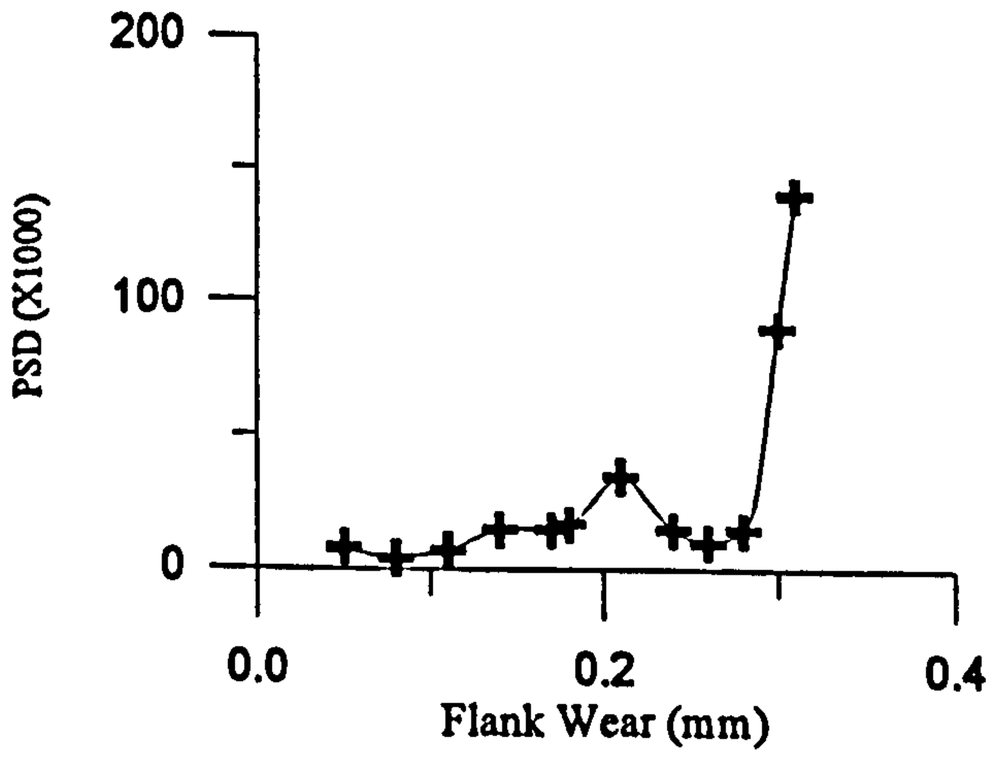


Figure 10 Frequency magnitude variation with flank wear, vibration [3.5;5.5] kHz

Feed force (Figure 11) appears to exhibit a small but consistent changes with flank wear.

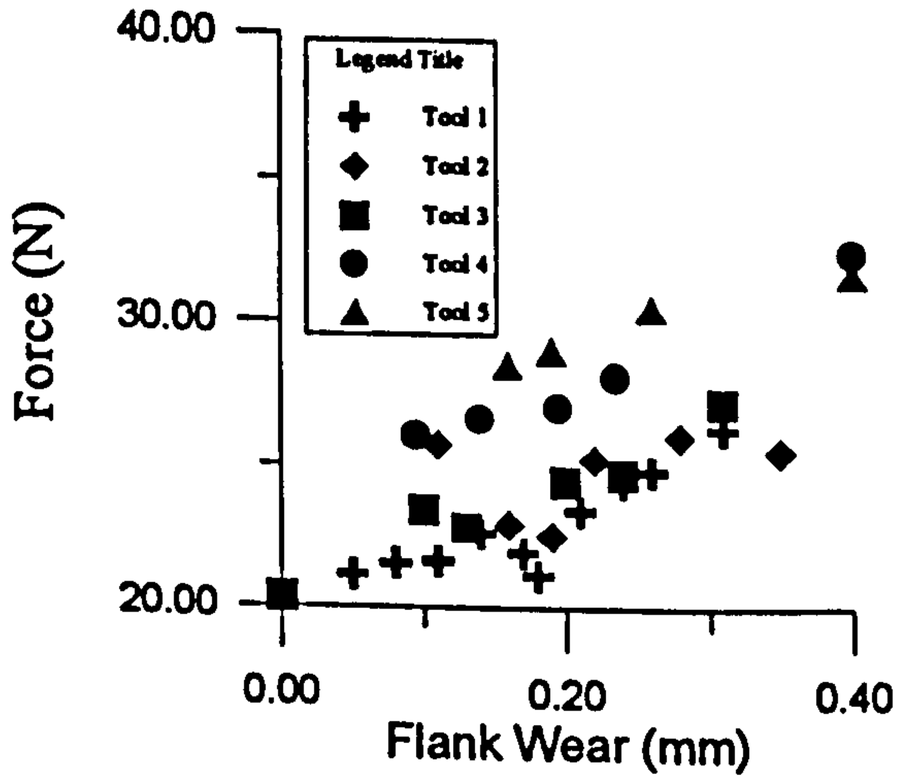


Figure 11 Feed force versus wear

The cutting force also shows a relation to wear much more clearly. As the wear increases the cutting force also increases, and as can be seen this increase is repeatable and consistent between different tool inserts.

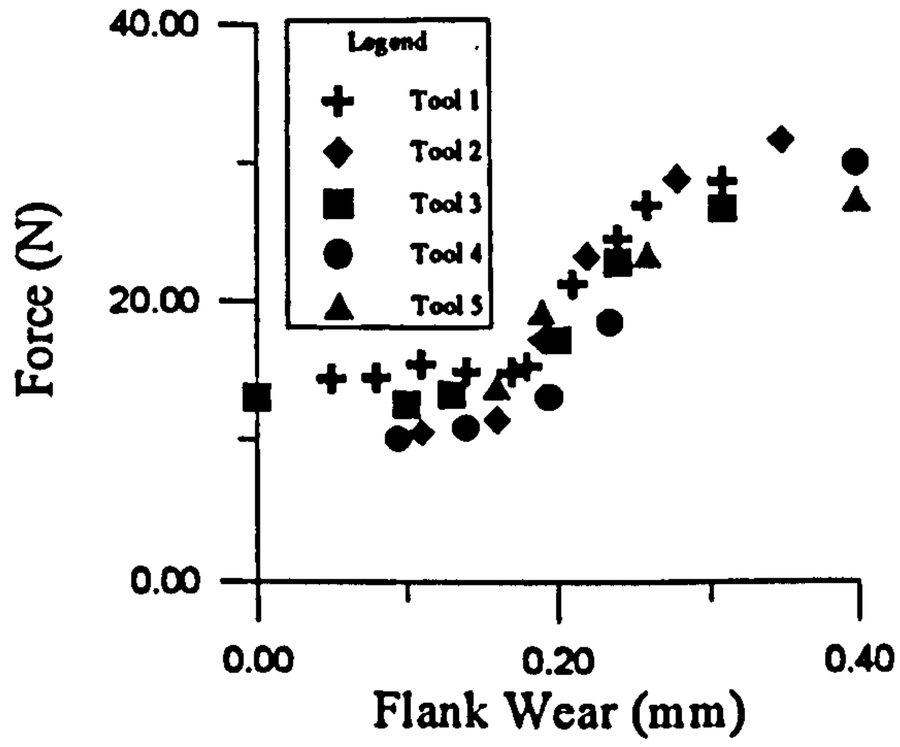


Figure 12 Cutting force versus wear

From these results the following parameters were chosen as features for Neural Network training and evaluation: spectrum bands from sound and vibration; statistical features from sound and vibration; feed and cutting force; spindle current.

5. Test with the SOM

The success of applying neural networks to a problem depends upon the type of problem domain and the representativeness of the data sets that are used to train the neural network.

The SOM was coded based on the theory developed by Kohonen [13]. The present network consists of two layers of neurones. The first is the sensory or input layer, consisting in this case of 15 neurones, one for each feature obtained from the sensors. The computation is carried out in the second layer, called the map, that also acts as the output layer and this was 10×10 neurones.

The learning procedure consists of two stages. In the first, the map unfolds until a global ordering of the neurones is reached. Every neurone tunes to a pattern or class of patterns, and neighbour neurones tune to similar inputs. In the second stage, the statistical distribution of the synaptic weights approaches that of the input variables.

A set of experiments was carried out using the SOM. The results achieved demonstrate the ability of this Neural Network to classify sets of data into consis-

tent classes. Figure 13 to Figure 18 show some of the results.

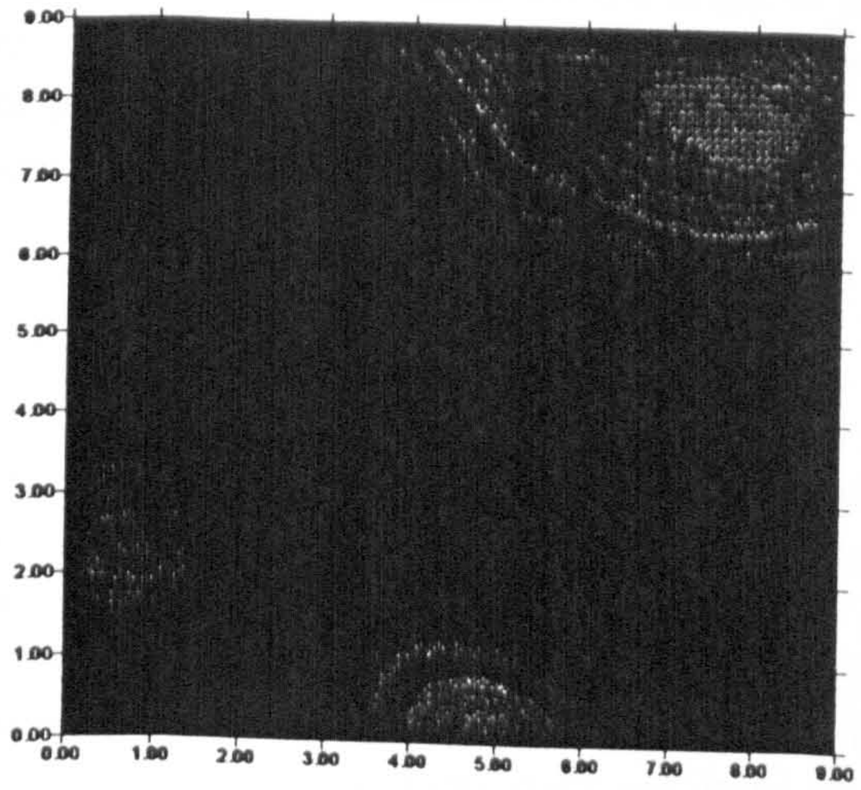


Figure 13 Contour plot for a tool with VB_B=0.05 mm

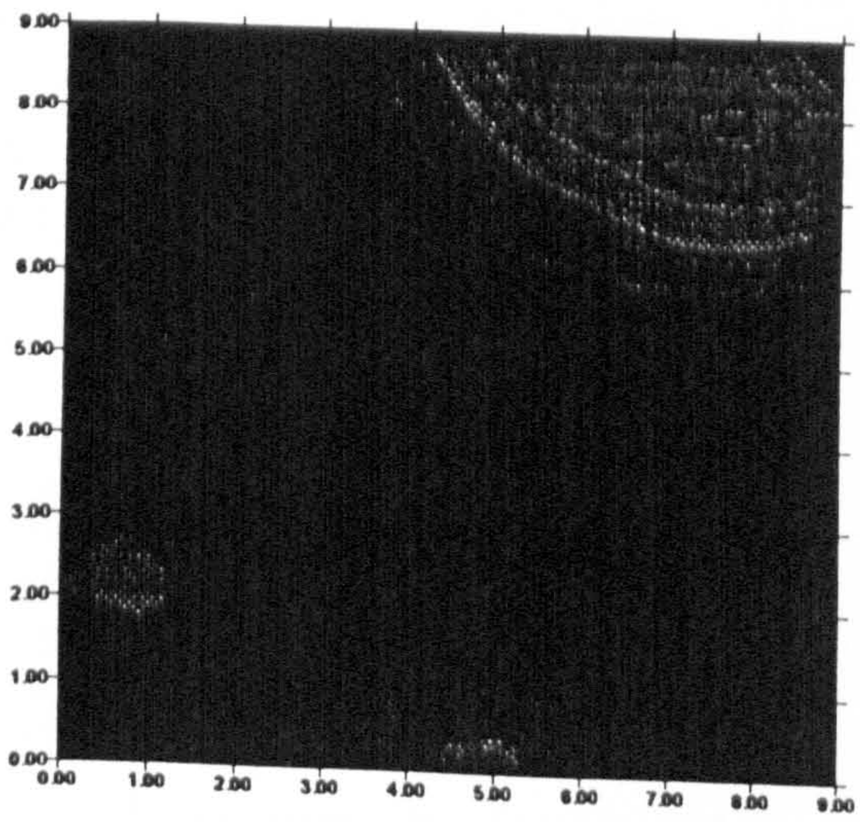


Figure 14 Contour plot for a tool with VB_B=0.11 mm

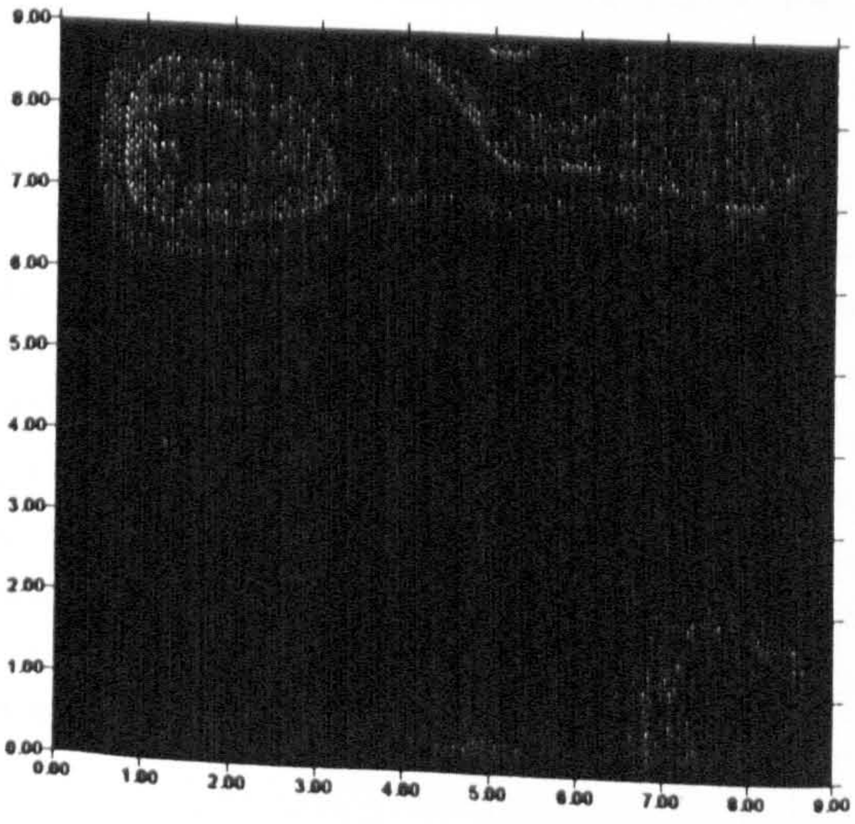


Figure 15 Contour plot for a tool with VB_B=0.17 mm

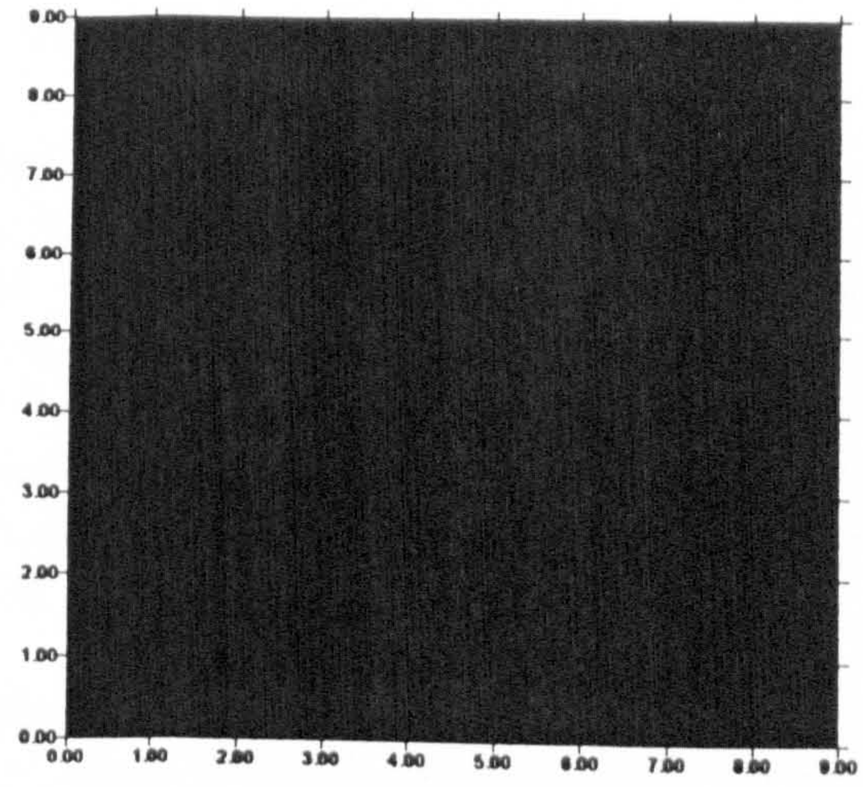


Figure 16 Contour plot of a tool with VB_B=0.21mm

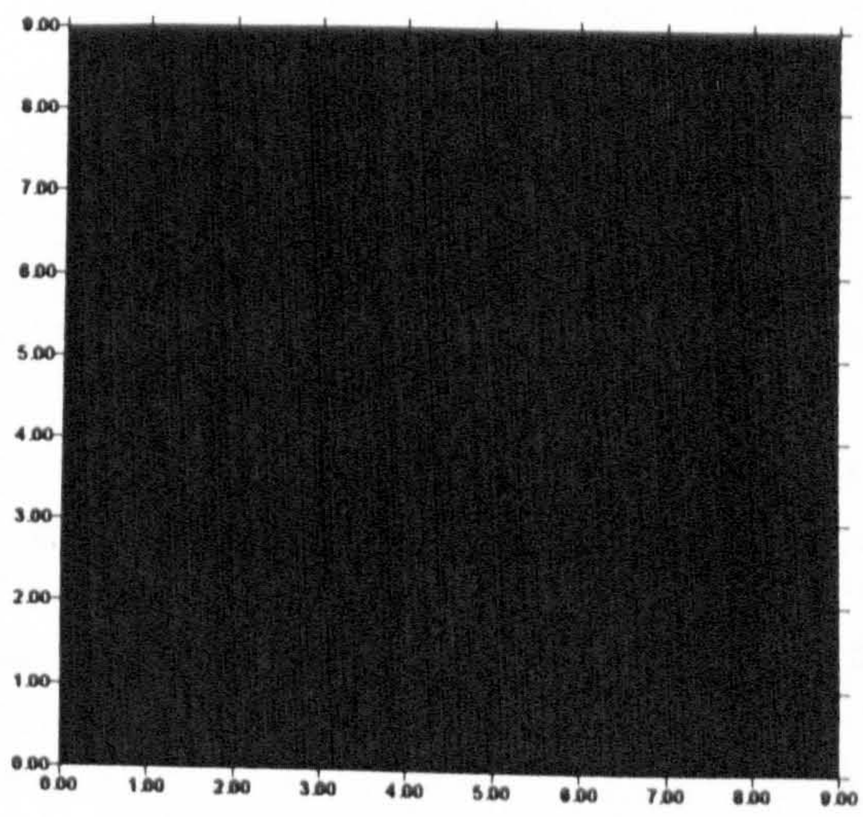


Figure 17 Contour plot for a tool with VB_B=0.26

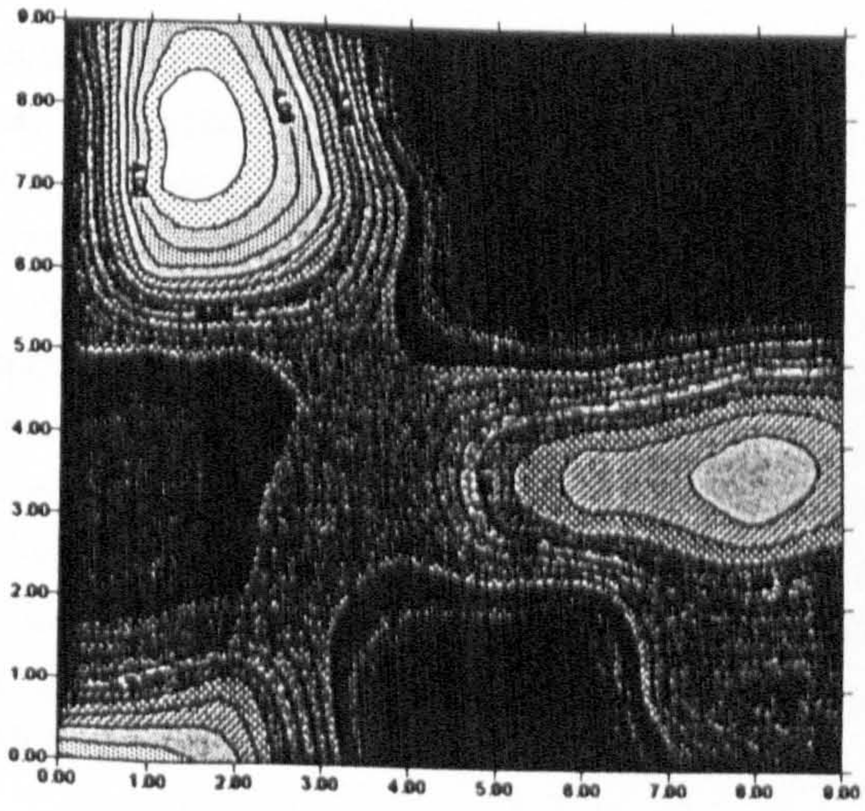


Figure 18 Contour plot for a tool with VB_B=0.31 mm

From the contour maps it can be seen that several distinct areas were created, the shaded areas correspond to the area allocated for the stage of tool wear. This network shows good performance although the interpretation of results is rather difficult, that is it does not provide a direct measure of wear.

6. Test with the ART2

The ART2 network was coded using theory presented by Carpenter & Grossberg [16]. In a similar way to the SOM, data was presented as an input vector which was composed of 15 inputs (described above). The output is presented in the form of classification, that is, for each data set it is allocated a class number.

One such simulation result is summarised in Figure 19, which shows how the ART2 architecture has quickly learned to group 34 sets of inputs into six recognition categories after a single presentation of each input.

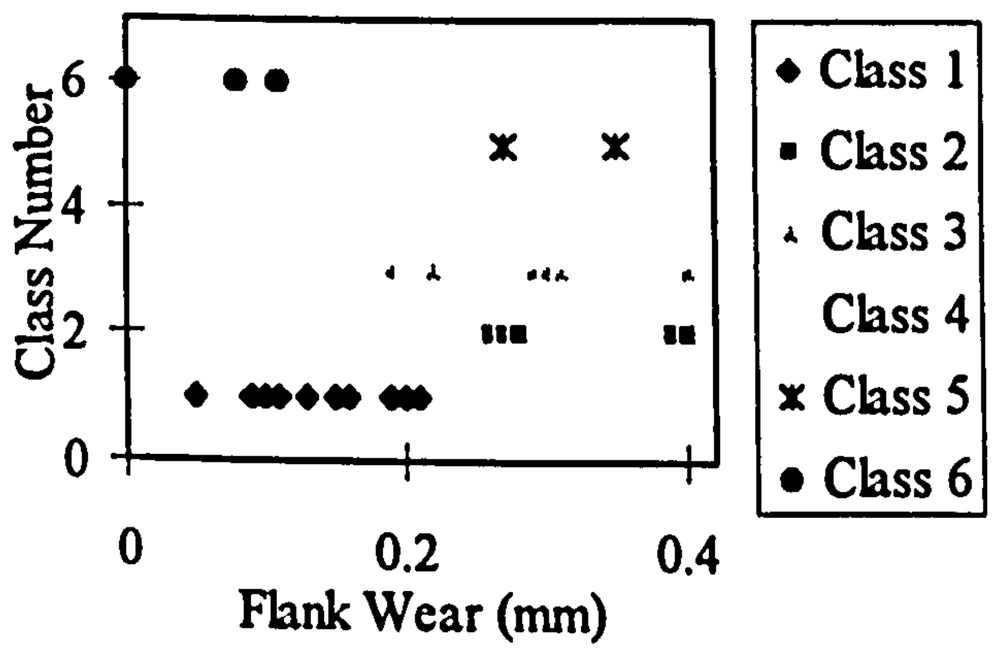


Figure 19 ART2 class attribution

The reset function in ART2 essentially checks if the "winning" node at the output layer actually matches the input pattern as closely as desired. The user inputs the vigilance to quantify "closely enough". In our case the value was set to p = 0.996, if set higher it will create finer classes and eventually many more than previously. If set lower it will be more perdular and classify previously "different" patterns into the same class. The vigilance parameter varies from application to application and depends also on the definition required by the application.

The lower layer of ART normalises the pattern, then suppresses noise, then normalises. A squashing function accomplishes the noise suppression, Carpenter & Grossberg [16].

7. Discussion of Results

The results obtained from the statistical and frequency parameters, as well as forces and spindle current, are somewhat difficult to interpret considering them one at a time as some appear to correlate, whilst others appear to hold no correlation with tool wear. This can be overcome by taking into account the neural networks' ability to extract information from apparently scattered information.

The use of a Self Organising Map (SOM) structure has shown that classification was performed quite efficiently (Figure 13 to Figure 18), although the interpretation of results was not that easy, due to the complexity of the output structure. Distinct areas were observed for each different tool wear stage, this being consistent for all sets of experiments. The interpretation would be possible with the use of some sort of "interpreter", e.g. possibly another neural network. The ART2 would possibly be the best feature extractor for such an ordinated output pattern.

The ART2 appears promising although the results are greatly dependent on the vigilance parameter. It is not yet known if this will be affected by a change in data source (different machine). As speed is one of the critical points on the implementation of an on-line tool wear monitoring system it is advisable to use this neural network since, it responds quickly. However, more work is required to evaluate the influence of each data set on the networks classification performance.

Successful automation of machining operations relies, to a great extent, on the ability of artificial systems to recognise process abnormalities and initiate corrective action. The combination of sensorial information and inference rules provide a structure to cope with the poorly understood cutting process. The sensory information is associated with experience-based memory triggers and a decision is then made as to whether the tool wear level warrants interruption of the process, checking of the tool condition and then, if necessary, initiation of tool changing procedures.

The Expert System approach to signal understanding attempts to address the apparent limitation of statistical

pattern recognition systems described above by utilising a higher level of knowledge to classify the signal. The approach emulates a human expert's decision-making process by applying expert knowledge in the form of queries and rules (Knowledge Base). The aim of this work is to use both a neural network and Expert System in an integrated fashion to provide maximum cover for the machining operation. Their integration by the use of If-Then-Else rules and goals, through the Expert System, is regarded as innovative.

This work has illustrated the potential of Neural Networks when applied to tool wear monitoring. Further work will be needed to develop an integrated condition monitoring. This hybrid system, comprising a Neural Network(s) and an Expert System, will be able to infer results gained from sensors, based on built-in knowledge and also from past experience.

8. Conclusion

The results presented in this paper show that pattern classification using different sources of information is possible. The present experiments show the results applied to tool wear monitoring. Both networks (SOM and ART2) show an ability to generalise and efficiently store and retrieve information. The inherent parallelism of such networks make them attractive for real time tool wear monitoring.

Future work is aimed at developing a hybrid system comprising a neural network(s) and Expert System (ES).

References

[1] E.J. Weller, H.M. Scrier and B. Weichbrodt, "What Sound can be Expected from a Worn Tool?", *ASME Paper* 68-wa/pro-4, 1-10 (1969)

[2] G.J. McNulty and N. Popplewell, "Health Monitoring of Cutting Tools through Noise Spectra", *1st joint Polytechnic Symp. on Manuf. Eng.*, Jun 14&15 E4 1-6 (1977)

[3] L. C. Lee, "A Study of Noise Emissions for Tool Failure Prediction", *Int. J. Tool Des.*, 26 205-15 (1988)

[4] Wu Ya, Ke Shiqiu, Yang Shuzi, Zhang Qilin, Xu Shanxiang and Wen Yaozu, "An Experimental Study of Cutting Noise Dynamics", *Machinery Dyn. and Element Vibr.*, 36 313-8 (1991)

[5] M. Shiraishi, "Scope of In-Process Measurement, Monitoring and Control Techniques in Machining Processes - Part 1: In-process Techniques for Tools", *Precision Eng.* 10(4) 179-89 (1988)

[6] C.Y. Jiang, Y.Z. Zhang and H.J. Xu, "In-Process Monitoring of Tool Wear Stage by the Frequency Band-Energy Method", *Annals CIRP*, 36(1) 45-48 (1987)

[7] J.L. Stein and K.-C. Shin, "Current Monitoring of Field Controlled DC Spindle Drives", *Sensors and Controls for Automated Manuf. and Robotics, ASME-PED*, 18 7-66 (1985)

[8] Li Dan and J. Mathew, "Tool Wear and Failure Monitoring Techniques for Turning - A Review", *Int. J. Machine Tools Manuf.*, 30(4) 579-98 (1990)

[9] K. Langhanner, "Cutting Forces as Parameters for Determining Wear on Carbide Lathe Tools and as Machinability Criterion for Steel.", *Carbide J. Soc. Carbide Tool Eng.* (1976)

[10] R. Mackinnon, G.E. Wilson and A.J. Wilkinson, "Tool Condition Monitoring Using Multi-Component Force Measurement", *Proc. 6th Int. Machine Tool Design and Research Conf.* 317-24 (1986)

[11] A. Ruiz, D. Guinea, L. J. Barrios and F. Betancourt, "An Empirical Multi-Sensor Estimation of Tool Wear", *Mech. Sys. and Signal Processing*, 7 105-19 (1993)

[12] S. Rangwala and D. Dornfeld, "Integration of Sensors via Neural Networks for Detection of Tool Wear States", *Proc. Winter Annual Meeting ASME PED*, 25 109-20 (1987)

[13] T. Kohonen, "The Self Organising Map", *Proc. IEEE*, 78 1464-80 (1990)

[14] H. Ritter and K. Schulten, "Kohonen's Self Organising Map: Exploring their Computational Capabilities", *IEEE Int. Conf. Neural Networks*, 1 109-16 (1988)

[15] S. Grossberg, "Competitive Learning: From Interactive Activation to Adaptive Resonance", *Cognitive Science*, 11 23-63 (1987)

[16] G.A. Carpenter and S. Grossberg, "ART2: Self-Organisation of Stable Category Recognition Codes for Analogue Input Patterns", *Applied Optics*, 26 4919-30 (1987)

Tool Wear Monitoring of Turning Operations by Neural Network

Classification of a Feature Set Generated from Multiple Sensors

R.G. Silva[1]; R.L. Reuben[2]; K.J. Baker[1] and S.J. Wilcox[1]

[1]Department of Mechanical and Manufacturing Engineering,

University of Glamorgan, Pontypridd, Mid Glamorgan. UK

[2]Department of Mechanical and Chemical Engineering,

Heriot-Watt University, Riccarton, Edinburgh. UK

Abstract

Feature extraction and decision-making is a matter of considerable interest for condition monitoring of complex phenomena with multiple sensors. In tool wear monitoring the extraction of subtle aspects from signals in the face of a range of transient and static events not related to tool wear offers a special challenge for diagnostic and control systems because of the broad range of information in the signal. Features based on frequency spectra and statistical transformations of a number of sensor signals have been studied in an attempt to obtain a reliable indication of the evolution of tool wear. Two neural networks, plus expert system utilising Taylor's tool life equation, were compared for the task of the tool wear state classification. Despite the complexity of the data and subsequent testing by the removal of two of the most clearly systematic features, a reproducible diagnosis of tool wear was obtained.

Introduction

In order to justify the capital investment associated with the installation of flexible manufacturing equipment it is necessary to achieve the maximum possible utilisation. One of the challenges this poses lies in devising methods for the classification of cutting tool wear. This seemingly simple task has posed considerable difficulty, probably due to the fact that tool wear introduces small changes in a process with very wide dynamic range. The task can be subdivided into a number of stages; sensor selection and deployment, generation of a feature or set of features indicative of tool condition and finally classification of the collected and processed information so as to determine the level of wear on the tool.

The process by which a metal cutting tool wears are well documented and a number of different mechanisms and types of wear geometry have been identified, depending on cutting tool material, workpiece material, tool geometry and cutting conditions. Several approaches can be used to measure tool wear; e.g. flank wear, length of wear notch and crater wear. Since flank wear is the simplest to measure, it is usually chosen as the basis for the tool failure criterion [1].

Analytical models have been used to study the effects of tool geometry on, for example, cutting forces [2]. Such models are of value in suggesting which sensors to use and their likely sensitivity to wear, although this analytical models may be too complex or difficult to implement in a real-time tool wear monitoring system. Empirically derived relationships are only weakly reliable for a limited set of conditions and can normally only be used for approximate calculations [3]. On the other hand, a rule based empirical inference method requires an expert familiar with relational mechanisms and an ability to translate those into

simple inference rules. As such "experts" do not exist (except in a narrow sense), another empirical, non-expert-based method is needed.

The overall objective of this work is to develop a tool wear monitoring system, in which, signal processing, neural networks and decision-making techniques are used. This paper deals with the developing plus testing of this system for one set of cutting conditions where tool sensor/feature approach is developed based on a set of cutting experiments using five sensors. The features extracted from the sensor signals have been assessed in terms of their ability to diagnose tool wear when presented to two different types of neural networks and the ability of an Expert System based on Taylor model of tool life to remove misclassification is demonstrated.

Methodology

In general, monitoring methods can be divided into two classes: model-based methods and feature-based methods. Model-based methods have two significant limitations; firstly, many machining processes are non-linear time-variant systems, which makes them difficult to model and, secondly, the signals obtained from sensors are dependent on a number of other factors, such as, machining conditions which, in the context of tool wear monitoring, constitute noise. Thus, it is often difficult to identify whether a change in the signal obtained from a sensor is due to a change, for example, in the cutting conditions or to wear of the cutting tool. Feature-based monitoring methods use suitable features of the sensor signal to identify the process conditions. These features could be derived from the time and/or frequency domain, for example mean, variance, skewness, kurtosis, crest factor, or power in a specific frequency band. According to Du *et al.* [4], a large number of such methods have been developed for the monitoring of manufacturing processes. It is unclear, however, which

feature recognition method performs best and, in fact, most of the literature only shows that a specific method works for a specific application [5].

In this work, pattern recognition is used to identify the effects on a number of features brought about by tool wear in the face of the "noise" caused by non-steady events and variations in the process conditions. A basic pattern recognition classifying scheme is shown in Fig. 1. General requirements can be listed *a priori*; some of these are general to most classifying schemes, and others are specific to this application [6].

There is obviously some interdependence between the choice of the signal processing schemes and the features on one hand, and the performance of the classifier on the other. Therefore the most critical choice is that of the feature vector, which should respond to the effects being monitored but show minimal sensitivity to any other disturbance. If possible, the choice of features should reflect some of the "expert" knowledge concerning the specific behaviour of the monitored system. Such specifically chosen features will often perform better than those based on very general considerations [7].

The sensors used to monitor the process form the final link in the chain and should, as far as possible, be simple to operate, robust and easily read and interpreted, since condition measurements are often made in rather awkward environments. Tool wear monitoring has been performed using many different types of sensor and the most common measures are temperature, feed and spindle motor currents, acoustic emission (AE), audible emissions, vibration and components of the cutting force. Based on a review by Micheletti et al [8] there is sufficient evidence to justify further study of at least four easily measured quantities. These are:

1. Cutting forces (feed and tangential)

2. Spindle current

3. Audible emissions

4. Machine vibration

Pre-Processing Methods

The condition or operational monitoring of machinery requires the detection of a signal related to the event or condition of interest and the identification of features that reveal information about it. For tool wear classification, most monitoring systems either use the raw signals directly without pre-processing, or simply low pass filter them with the aim of reducing the contribution from corrupting sources. While relatively easy to implement, these techniques have not proven to be particularly effective at reducing the variations and tend to remove vital information. To achieve this end it is necessary to examine the tool wear signal and noise generating processes [10].

The use of multiple sensors has been shown to enhance the performance of tool wear monitoring systems, because each sensor type provides notionally independent information that is related to the level of tool wear. It also reduces the sensitivity of the system to any particular sensor's drawbacks, requiring less precision from a single sensor and thus potentially requires less sophisticated signal processing [11]. Through appropriate analysis, the dependence of each feature to changes in process conditions can be analysed [12]. This provides improved reliability in making decisions on the state of tool wear in the face of (perhaps minor) changes in machining conditions.

Reddy [13] has reviewed the use of all the sensors of interest here for the application of tool wear monitoring. Table 1 summarises the most promising features identified from Reddy's review.

Experimental Apparatus and Procedure

A set of tool wear cutting data was acquired by machining a block of mild steel under a given set of cutting conditions, with a coated cemented carbide tip (Table 2). The set of sensors used were (Table 3); an accelerometer for measuring vertical vibration, a microphone for recording the sound emission, a strain gauged tool holder for force measurement and a meter for the spindle current of the CNC machine. The turning operation was carried out on an MT 50 CNC Slant Bed Turning Centre. The experimental set-up and instrumentation can be seen in Fig. 2. A data acquisition board type PC-30PGL(Amplicon) handled the acquisition of all the signals, saving the data to permanent storage. The analogue signals were sampled at 20 kHz with tool wear and sensor data being acquired at intervals of 2 min, taking into account an expected life of about 15 minutes for each insert. Sample data were recorded, at a position that was approximately in the middle of the bar, for 6 inserts each with a data record length of 512 points.

Results of Feature Extraction

Each 512 point record was processed to generate the features used in the classification stage. Some of these features are presented here in graphical form simply to give an idea of the complexity of the problem of recognising a given tool wear state from the features. The following 12 features were extracted from the sound and vibration data: the absolute deviation, average, kurtosis, skewness and two bands of the frequency spectra. Three

additional features were presented from the means of the spindle current and feed and tangential forces.

As can be seen Fig. 3 and Fig. 4. both feed and tangential forces show an increase with tool wear which is consistent between tools.

The analysis of the data in the frequency domain is often useful in providing additional features, so Fast Fourier Analysis (FFT) was performed using a Bartlett window and subtracting the mean before performing the FFT.

The waterfall plot in Fig. 5 shows the evolution of the frequency spectrum of sound emission against tool wear. Both the sound and vibration spectra exhibited changes with wear at a frequency of 2.5 kHz, although this was not consistent in absolute terms for all inserts. Because of this variation, such features cannot be used alone for tool wear monitoring. Further analysis was performed using Daubechies wavelet filter analysis but this did not produce any stronger correlation.

The remaining features (absolute deviation, mean, kurtosis and skewness of both sound and vibration) exhibited little correlation with flank wear, e.g. Fig. 6, data points appearing to be randomly distributed through the entire space.

The influence of sample size on the statistical parameters can be seen for the example of kurtosis in Fig. 7 (obtained from processing a data set of 8192 points). This shows that the kurtosis is rather sensitive to sample size and may provide some explanation for the poor correlations observed with the statistical parameters. However it is difficult to establish a "correct" sample size and 512 points was chosen here in the interest of enabling real time processing of the 15 features.

Although the statistical parameters did not present any obvious relation to the evolution of tool wear, it was felt that it was not possible at this stage to judge their importance for tool wear monitoring due to the complexity of the process. The second part of this paper, however, shows that such data can still be used to monitor the cutting process.

Neural Network Classification

Bailey and Thompson [14] have surveyed the application of neural networks and have developed several criteria for selecting neural network algorithms. Based on these criteria, the ART2 and SOM approaches were chosen principally for their ability to extract patterns from noisy data and their unsupervised learning capability

Abstraction of hardly accessible knowledge and generalisation for distorted sensor signals are two of the most attractive features of neural networks when applied to sensor fusion in tool wear monitoring. Despite the current popularity of backpropagation, as a supervised learning algorithm, its need for a correct tool classification in every training sample limits its successful application to on-line tool wear monitoring. This is because the machining operation must be interrupted in order to acquire correct information about tool condition, and because the system must handle numerous combinations of tool type, material, and cutting conditions, a supervised learning procedure like backpropagation is undesirable. Thus, it would be helpful to have a neural network utilise "unsupervised" training samples without the need for correct tool wear information. This would allow the system to be based on an interpretation of the resulting self-organisation with the fewest number of "supervised" samples. The two algorithms selected for this application are described briefly below.

ART2 Algorithm

The adaptive resonance theory (ART) architecture, creates and organises categories for objects and has the ability to respond immediately to its experiences. The ART architecture first created by Carpenter and Grossberg [15] are designed to teach themselves new categories and continue storing information without disregarding any kind of information.

The ART network, Fig. 8, is most easily understood as a device for classifying input patterns. The goal is to present a sequence of patterns to such a network and have each "appropriately" classified by this network. An ART algorithm can classify and recognise input patterns without ever have an omniscient teacher present. The heart of an ART network consists of two interconnected layers of neurones, F_1 and F_2, which comprise the attentional system. The input leads to activity in the feature detector neurones in F_1. This activity passes through connections (synapses) to the neurones in F_2. Each F_2 neurone adds together its input from all the F_1 neurones and responds. A parameter called the attentional vigilance parameter (R) determines how fine the categories will be, if vigilance increases (decreases) due to environmental feedback, then the system automatically searches for and learns finer (coarser) recognition categories. Gain control parameter enable the architecture to suppress noise up to a prescribed level. The architectures global design enables it to learn effectively despite the high degree of non linearity of such mechanisms [31].

SOM Algorithm

The Self-Organising Map (SOM) [16], is an unsupervised neural model that projects a high dimensional input space onto a (usually) one or two dimensional output space by using unsupervised training (Fig. 9). This output space is represented by a discrete lattice of neurones, usually arranged in a rectangular manner. The idea of such a model is to generate

topology mappings, where a low-dimensional image of the high dimensional input space is built into the rectangular array. Neighbourhood neurones on the map tune to similar features of the sensory or input space, in a self-organising competitive learning manner.

Thus, the SOM associates to each input pattern a representative output pattern. This method of model building can be seen as performing vector quantisation in that it seeks models which minimise the quantisation error [17]. Models are adjusted incrementally as new data are presented. An interesting aspect of the SOM is that some ordering takes place resulting in adjacent models in pattern space being near each other model space. The main disadvantage of the SOM for this application is that output has to be interpreted.

In tool wear monitoring, thanks to topological ordering, Kohonen's Feature Map requires fewer samples with the correct level of wear because the interpretation of an output node can give information useful to the interpretation of its neighbouring nodes. Similarly, the ART2 requires less sample data with the correct classification since similar patterns are self-organised into similar categories.

Classification of Experimental Results

Tests were carried out on all the acquired experimental data in order to determine the ability of the neural networks to classify tool wear. These tests were carried out off-line but with a view to on-line implementation of a tool wear classification system. The first test, carried out with both NNs, was aimed at assessing their ability to classify when trained with all features. The NNs were trained with 36 data sets (four tools) and tested with 18 sets (two tools) with each data set being constructed from the 15 features identified earlier. The training data was presented randomly to the NNs with the SOM being trained for 30,000 epochs (one epoch

represents one presentation of the training data to the NN), which had been found to give the best performance. The ART2 only required 4,000 presentations, although good results could be achieved with only one epoch. The results obtained from the presentation of the selected features show that both NNs are capable of recognising, and classifying the data into their associated wear levels, (Fig. 10 and 11).

Fig. 12 and 13 show the result of using the procedure described above but this time eliminating both forces and the spindle current. It can be seen that, although not very accurate, the NNs still classify the unseen samples with the SOM performing better than the ART2. However, there is a significant number of miss-classifications and a means to eliminate these will be discussed shortly.

Table 4 shows a statistical analysis of the results shown in Fig. 10 and 13. The standard deviation gives an illustration of the certainty of the flank wear prediction a given value of VB_B and the 95% confidence limits on this are also shown.

One way in which the miss-classifications could be eliminated would be by the introduction of an external "supervisor", that uses appropriate knowledge about the cutting process to remove illogical classifications by the neural networks. An appropriate knowledge in the case of tool wear could be generated by Taylor's tool life equation ($VT^n=C$, which provides a relationship between cutting speed, V, tool life, T, and two parameters, n and C, dependent on tool and workpiece materials). This model is a useful way of establishing a tentative value for the expected tool life and, from this, in defining intervals at which it is useful to classify a tool as worn or not. From experience gained in the experiences described here and from the analysis of other wear curves [18] Taylor's tool life equation can give estimates within ±35%

of the actual tool life. Based on this estimates a set of rules can be written to establish classification confidence limits.

$$\begin{aligned} &) \; IF\ (VB_{NN} > VB_{Taylor} + 0.15)\ THEN\ exclude \\ &2) \; IF\ (VB_{NN} < VB_{Taylor} - 0.15)\ THEN\ exclude \end{aligned} \tag{2}$$

In order for such rules to be used Taylor parameters, n and C, for the workpiece and tool material have to be known. The material used here was a free-machining steel with specified sulphur content of between 0.25 and 0.35% for which Taylor parameters n, and C can be taken as 0.33 and 823 respectively [18].

The proposed method eliminates most of the outliers and enhances classification. Fig. 14 and 15 show the previous method applied to the case where three of the strongest features were excluded. Both methods were combined in order to complement each other, the neural networks estimate the tool wear values and the Taylor's tool life equation establish the confidence limits based on empirical knowledge.

Discussion

In the experiments carried out here, the features most highly correlated with tool wear were the forces (both feed and tangential), the spindle current and the frequency bands associated with sound and vibration. The remaining features exhibited no apparent correlation with tool wear. The NNs achieved a high classification accuracy when using all the features, as can be observed from Fig. 10 and 11. The removal of the three strongest features (forces and spindle current) lead to a substantial number of miss-classifications (Fig. 12 and 13), which can be alleviated by using empirical knowledge to eliminate outliers. An increased accuracy would probably be achieved with more training samples, as this should enable the NNs to generalise

more and also help them reduce the number of misplaced data points. A larger sample size for the statistical parameters may also be useful, but this has implications for the processing time.

In order to discuss the results more fully a measure of the performance of the monitoring system is needed. Clearly, the better the results in Figs. 16 to 18 the closer they will be to a straight line with a 45^{o} slope (optimum classification), and hence a simple linear regression analysis should give a quantitative measure of the accuracy of classification. Table 5 shows the value b, the linear regression coefficient in $y=b.x$, and also the correlation coefficient r with the values obtained being compared to ideal values of unity for b and r. As an example, the value of b for the Taylor tool life equation is 1.1 where r is, of course, unity (if the evolution of wear with time is linear).

It has been demonstrated that some outliers can be successfully eliminated by the application of rules based on Taylor's model equation, resulting in an improved monitoring system performance. Table 6 shows the improvement in the correlation coefficients achieved after the application of Taylor's prediction on the test set. As can be seen, the NNs predictions tend to slightly underestimate the wear, whereas Taylor equation used here is slightly conservative. The use of the Taylor equation to eliminate outliers improves the NNs predictions in that the correlation of coefficients increase towards the ideal value unity as does the slope. From a study of alternative Taylor equations it appears that the slope of the Taylor line will always exceed unity. Overall, this means that the combined approach succeeds in its aim of giving an on-line estimate of tool condition without the conservatism associated with the use of empirical rules.

Conclusions

A neural network based approach has been presented for the classification of tool wear in terms of 15 features calculated from the outputs of 5 sensors. Two types of neural network were used, the Self Organising Map and Adaptive Resonance Theory, in order to classify the statistical and frequency domain features of the sensor signals.

Sample classification is achieved with high accuracy using the full set of features by both NNs (SOM *r=0.946*; ART2 *r=0.914*). Eliminating three of the strongest features, classification is still achieved but with reduced accuracy (SOM *r=0.782*; ART2 *r=0.691*), although this represents an extreme case of sensor failure. Applying a Taylor model to identify and eliminate outlier data improves the network predictions in all cases and it appears that repeated application of this approach might lead to a close prediction of tool wear that would be available for the Taylor model or neural network alone.

References

1. I. Finnie 1956 *Spring Meeting of the ASME, Portland, Ore. March,* 18-21. Review of the Metal-Cutting Analyses of the Past Hundred Years.

2. E.L. Stern and R.P. Pellini 1993 *Manufacturing Science and Engineering* **64**, 445-451. A Study on the Effect of Tool Wear on Machining Forces.

3. C.S. Leem, D.A. Dornfeld and S.E. Dreyfus 1995 *Journal of Engineering for Industry* **117**, 152-159. A Customized Neural Network for Sensor Fusion in On-Line Monitoring of Cutting Tool Wear.

4. R. Du, M.A. Elbestawi and S.M. Wu 1995 *Journal of Engineering for Industry* **117**, 133-141. Automated Monitoring of Manufacturing Processes. Part 2: Applications.

5. L. Dan and J. Mathew 1990 *International Journal of Machine Tools and Manufacturing* **30**, 579-598. Tool Wear and Failure Monitoring Techniques For Turning - A Review.

6. R. Du, M.A. Elbestawi and S.M. Wu, 1995 *Journal of Engineering for Industry* **117**, 121-132. Automated Monitoring of Manufacturing Processes, Part 1: Monitoring Methods.

7. S. Braun, J. Rotberg and E. Lenz 1987 *Mechanical Systems and Signal Processing* **1**, 185-196. Signal Processing For Single Tooth Milling Monitoring.

8. G.F. Micheletti, W. Koening and H.R. Victor 1976 *Annals of the CIRP* **25**, 483-496 In Process Tool Wear Sensors For Cutting Operations.

9. L.V. Colwell 1971 *Annals of the CIRP XVIV* 647-651. Methods for Sensing the Rate of Tool Wear.

10. L.P. Heck 1993 *Proceedings ICASSP IEEE, International Conference on acoustic, speed and signal processing* **1**, 55-8. Signal Processing Research in Automatic Tool Wear Monitoring.

11. A.M. Agogino, S. Srinivas and K.M. Schneider 1988 *Mechanical Systems and Signal Processing*, **2**, 165-185. Multiple Sensor Expert System for Diagnostic Reasoning, Monitoring and Control of Mechanical Systems.

12. A. Ruiz, D. Guinea, L.J. Barrios and F. Betancourt 1993 *Mechanical Systems and Signal Processing* 7, 105-119. An Empirical Multi-Sensor Estimation of Tool Wear.

13. Y.B. Reddy 1992 *Journal of Information Science and Technology* 1, 91-103. Multisensor Data Fusion: State-of-the-Art.

14. D. Bailey and D. Thompson 1990 *AI Expert* June, 38-47. How to Develop Neural-Network Applications.

15. G.A. Carpenter and S. Grossberg 1987 *Applied optics* 26, 4919-4930. ART2: Sef-Organization of Stable Category Recognition Codes For Analog Input Patterns.

16. T. Kohonen 1990 *Proceedings of IEEE* 78, 1464-1480. The Self Organizing Map.

17. M. Sabourine and A. Mitiche 1993 *Neural Networks* 6, 275-283. Modelling and Classification of Shape Using a Kohonen Associative Memory With Selective Multiresolution.

18. M.C Shaw 1989 *Metal Cutting Principles*, Oxford Series on Advanced Manufacturing 3

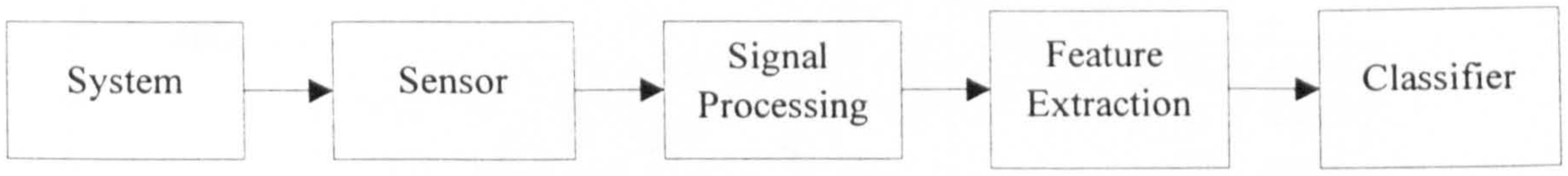


Fig. 1: Pattern Recognition Classifying Scheme

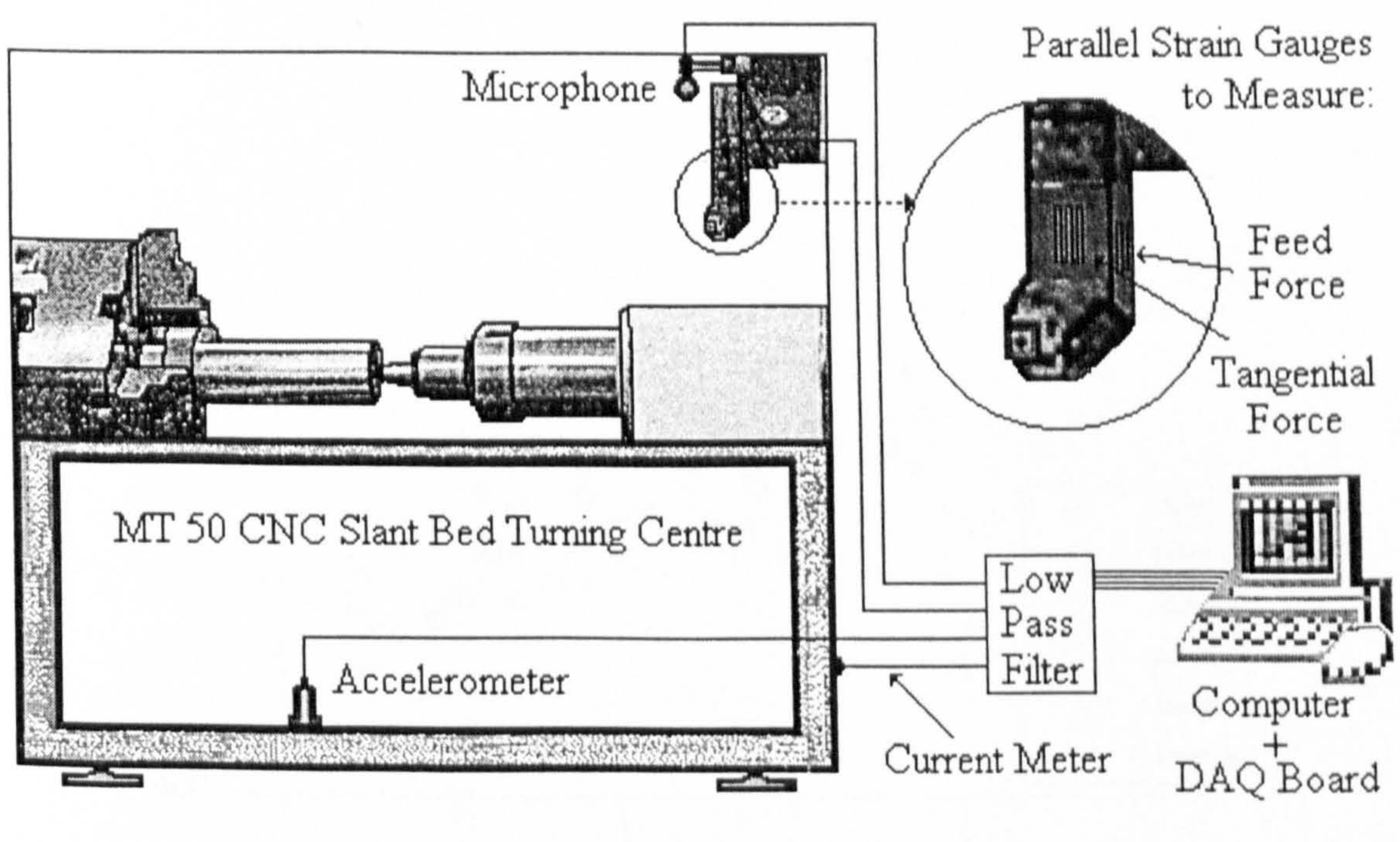


Fig. 2: Schematic diagram of lathe with sensor set

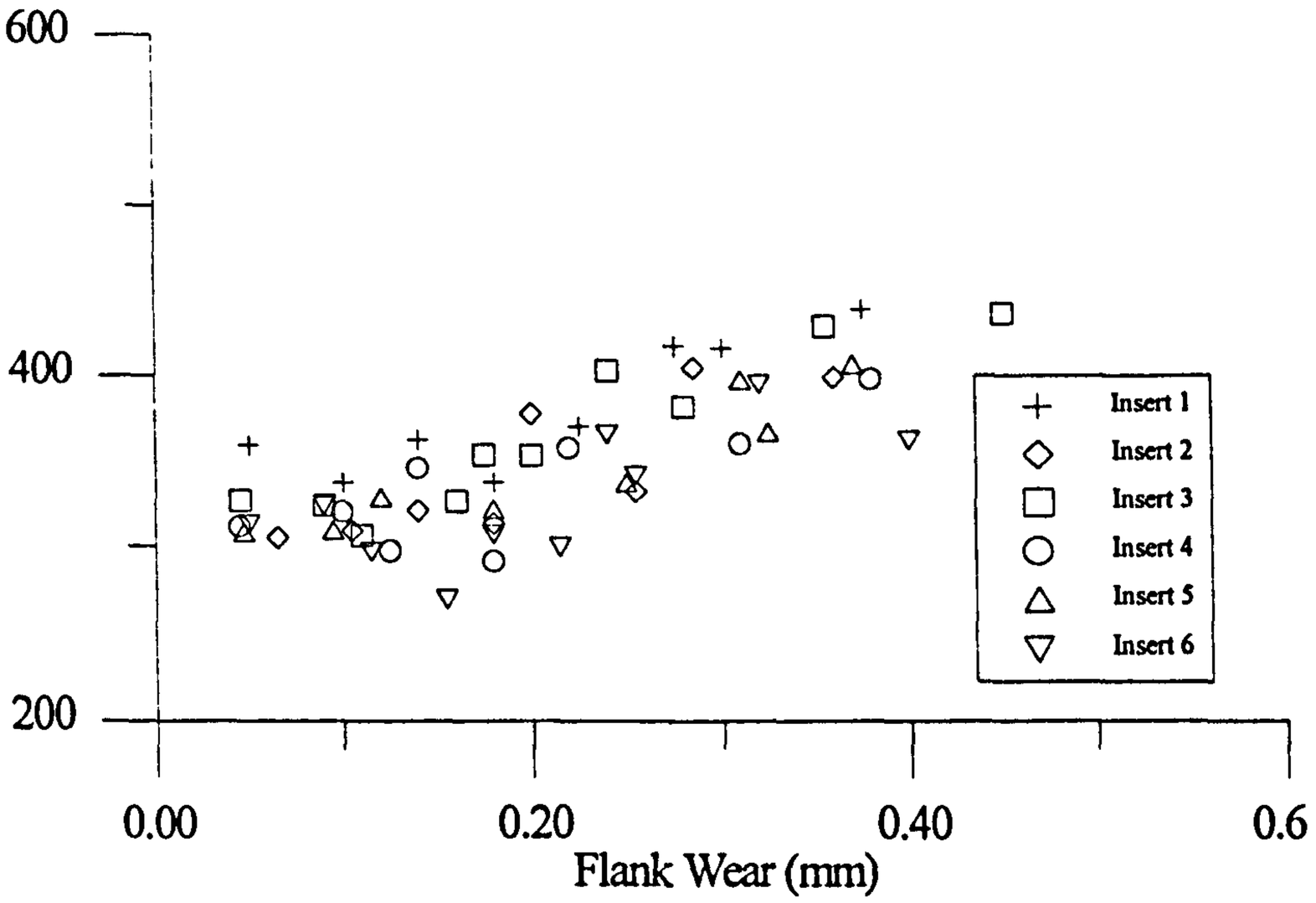


Fig. 3: Feed force evolution with flank wear (512 point average)

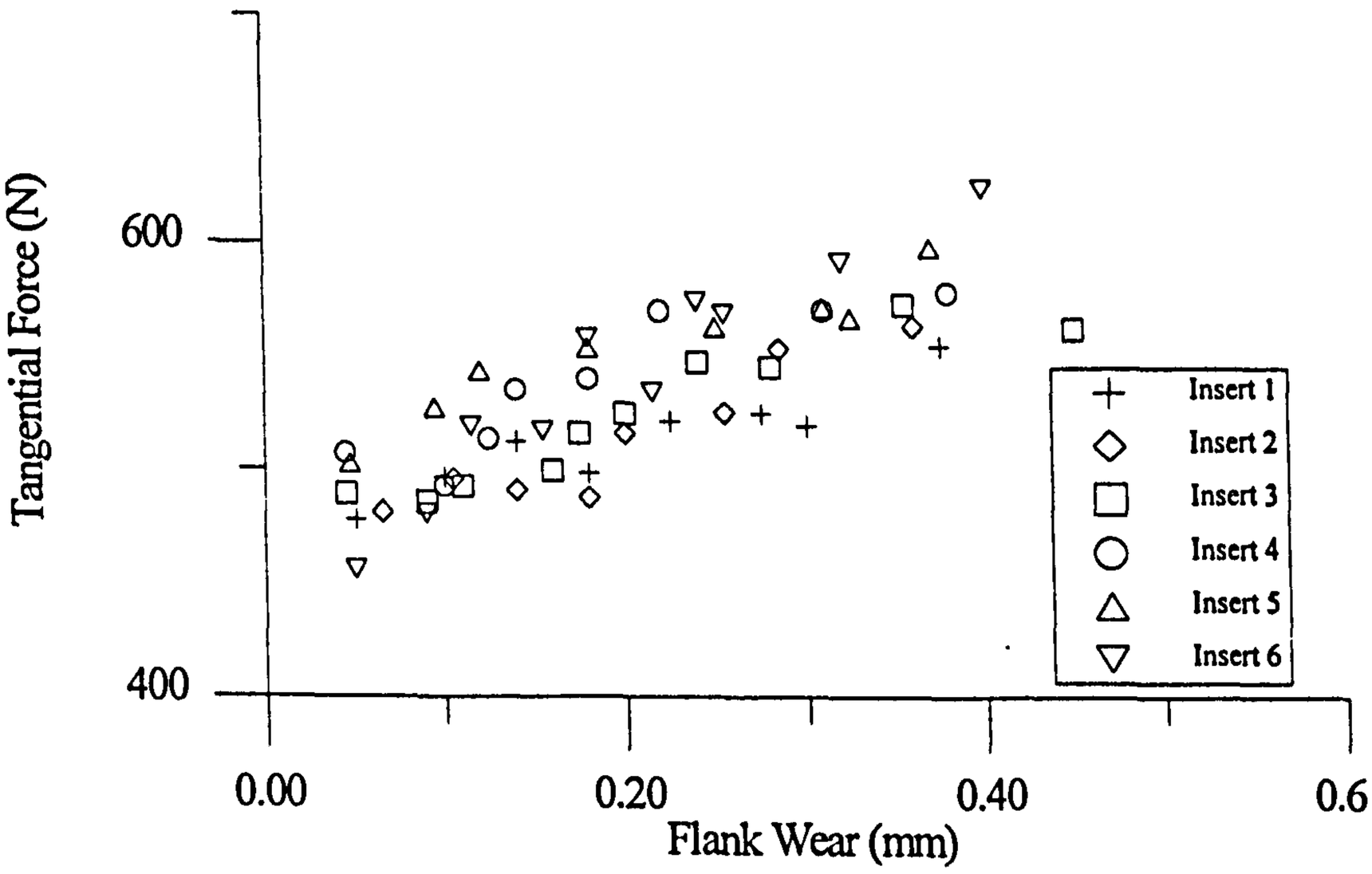


Fig. 4: Tangential force evolution with flank wear (512 point average)

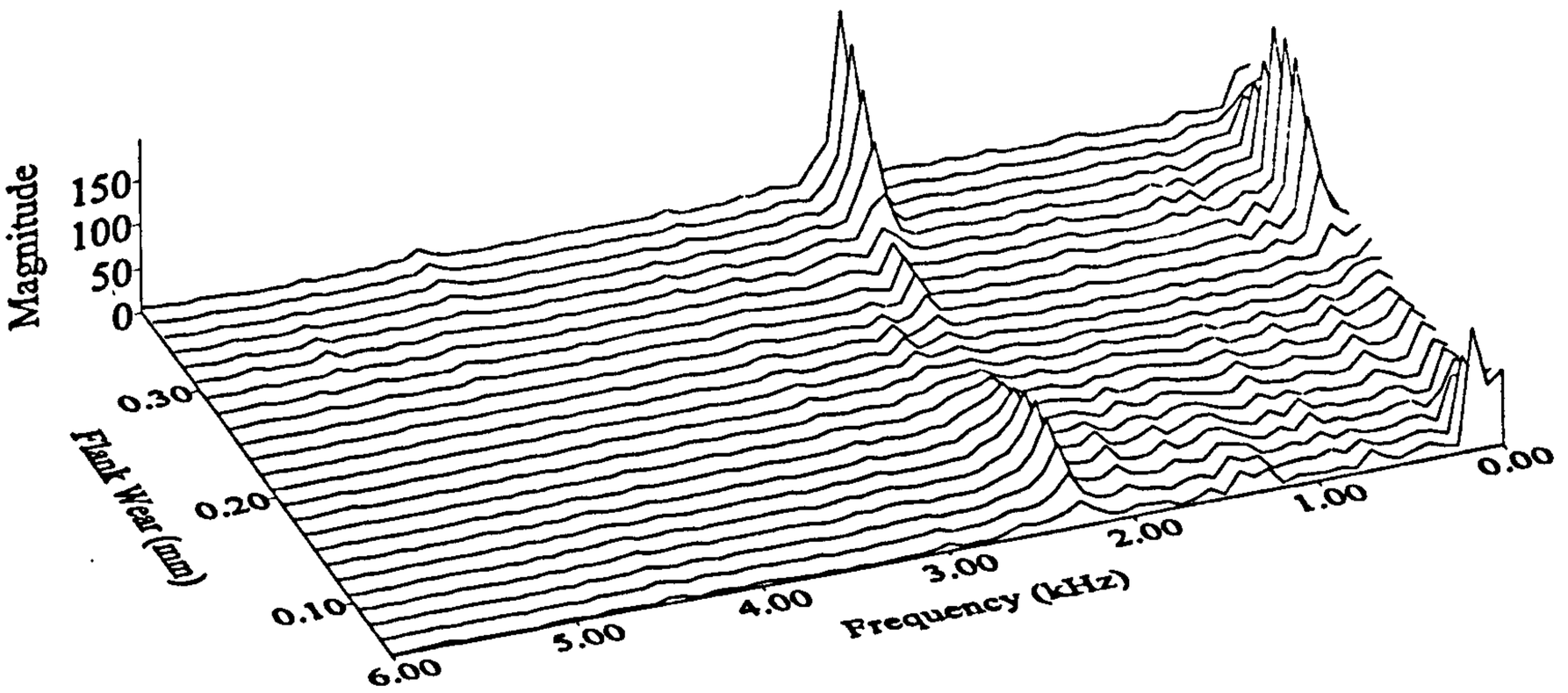


Fig. 5: Waterfall plot of frequency spectrum of sound emissions (Insert 1)

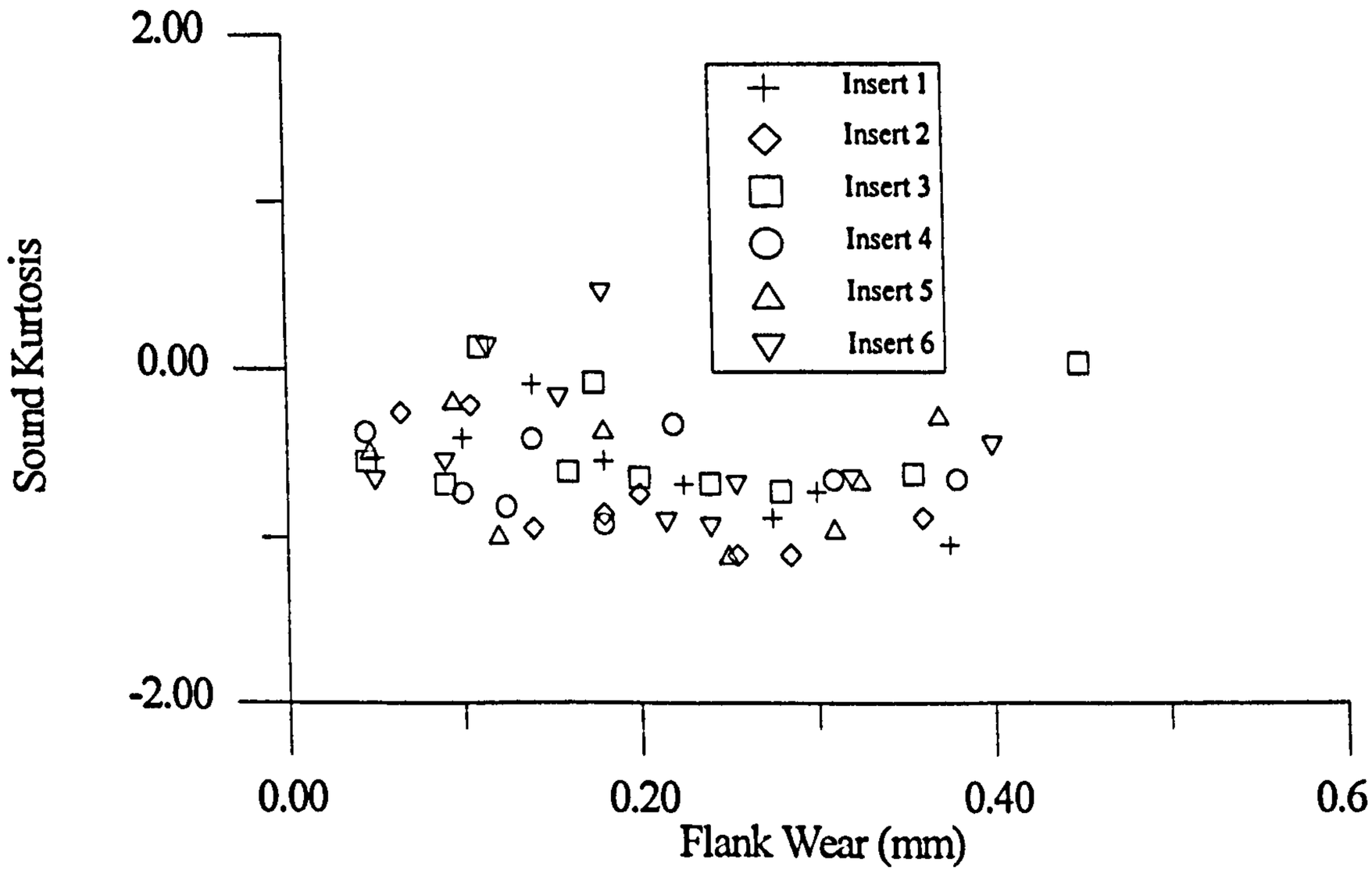


Fig. 6: Sound Kurtosis evolution with flank wear (512 point series)

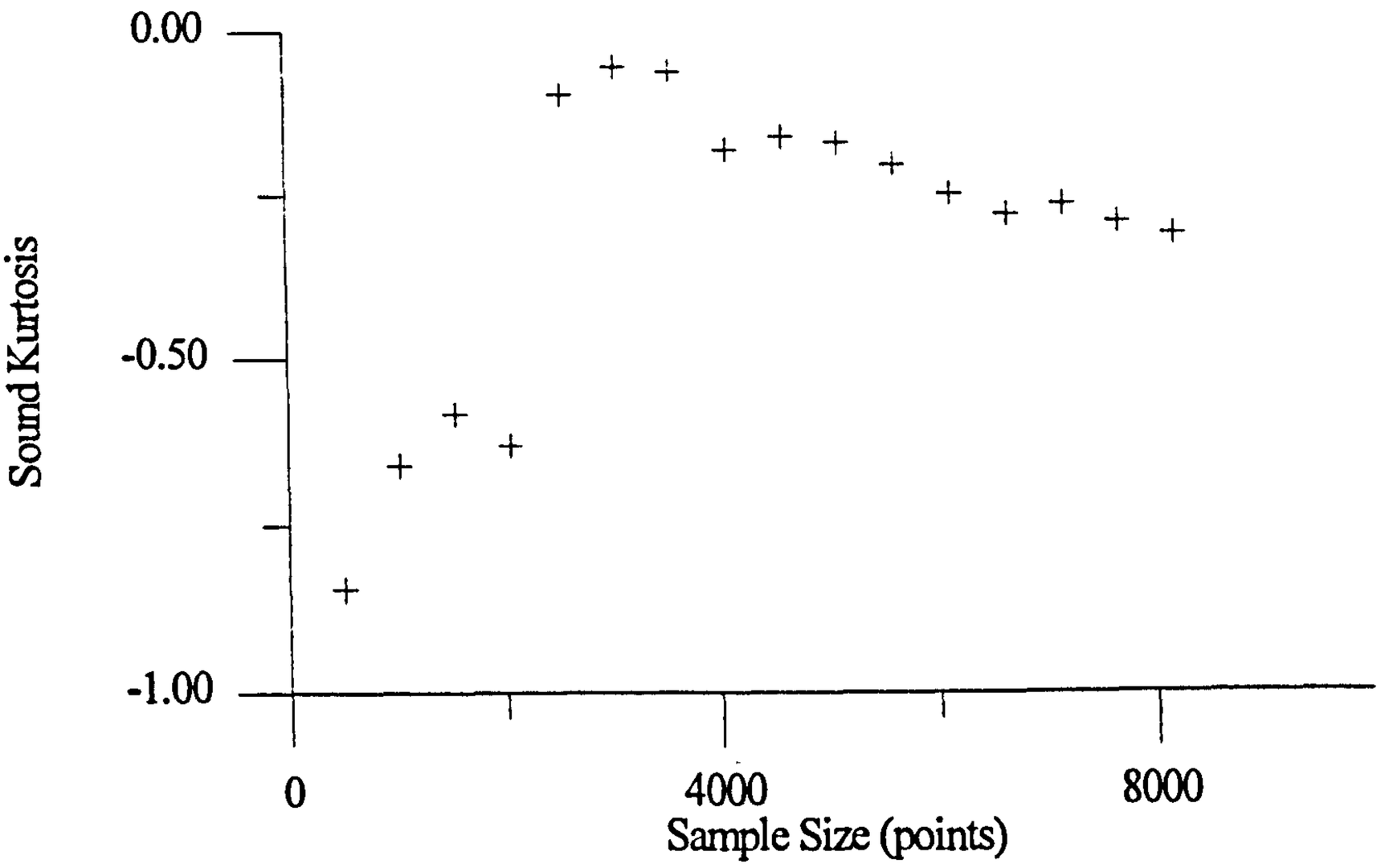


Fig. 7: Effect of sample size on the claculated Kurtosis of sound (VB_B = 0.22 mm)

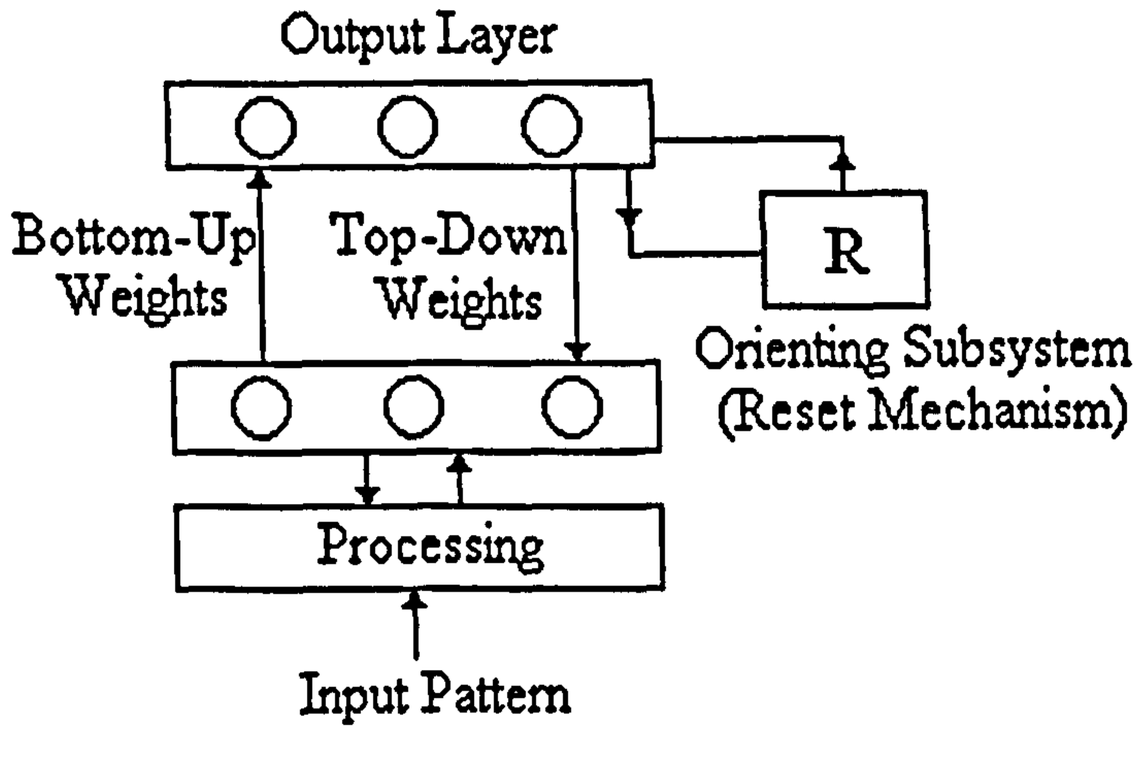


Fig. 8: ART2 processing levels

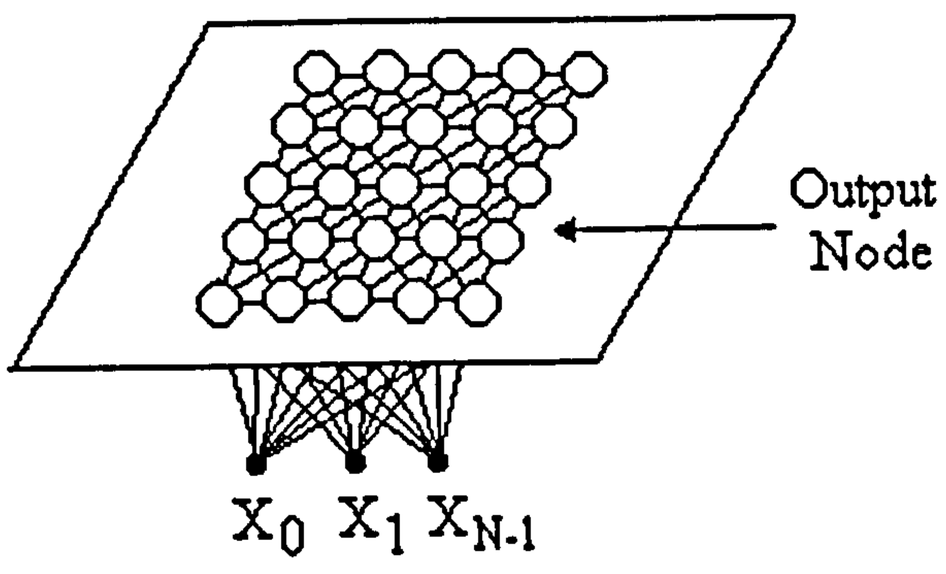


Fig. 9: SOM 2D projection map

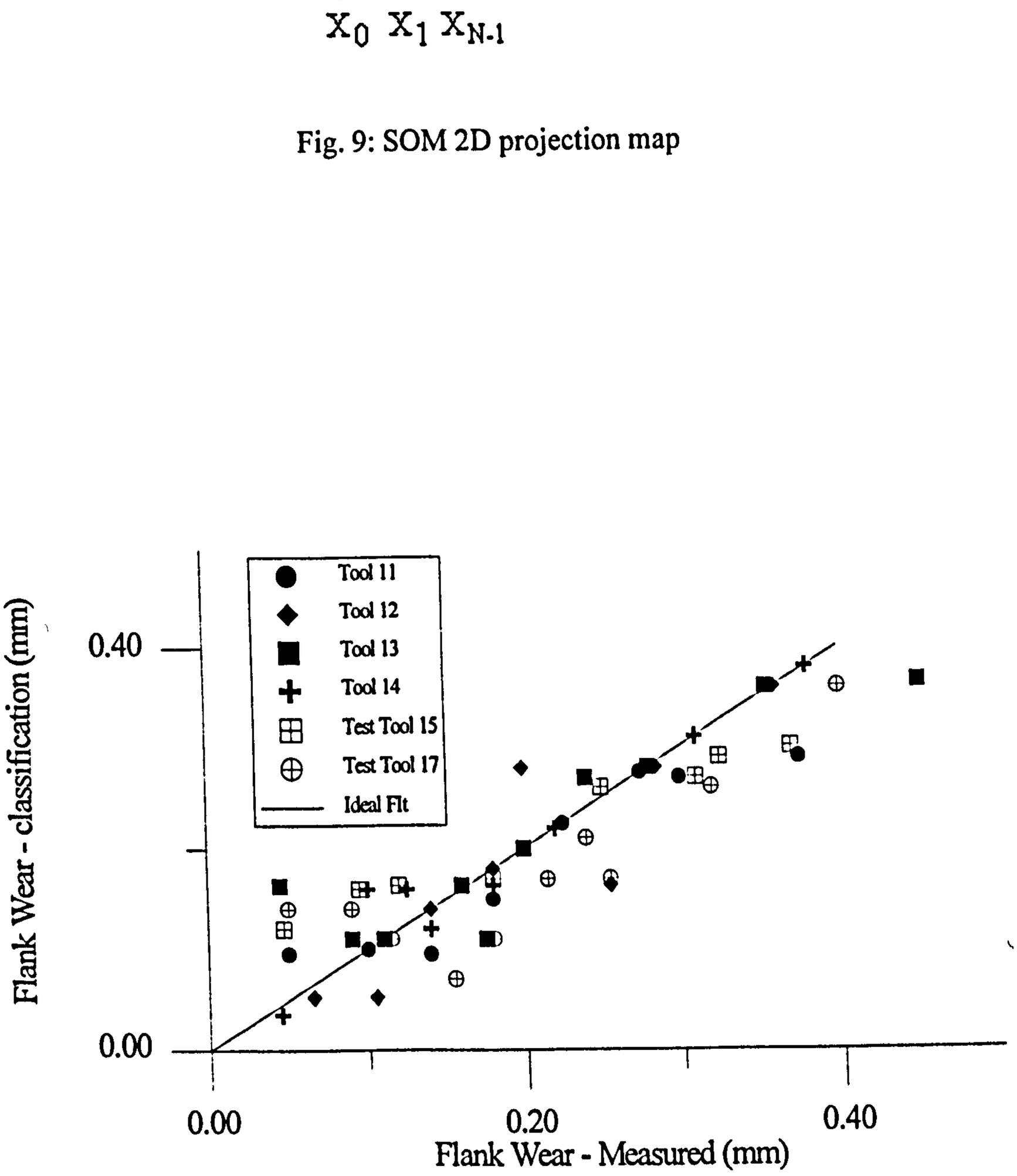


Fig. 10: SOM classification with full features set

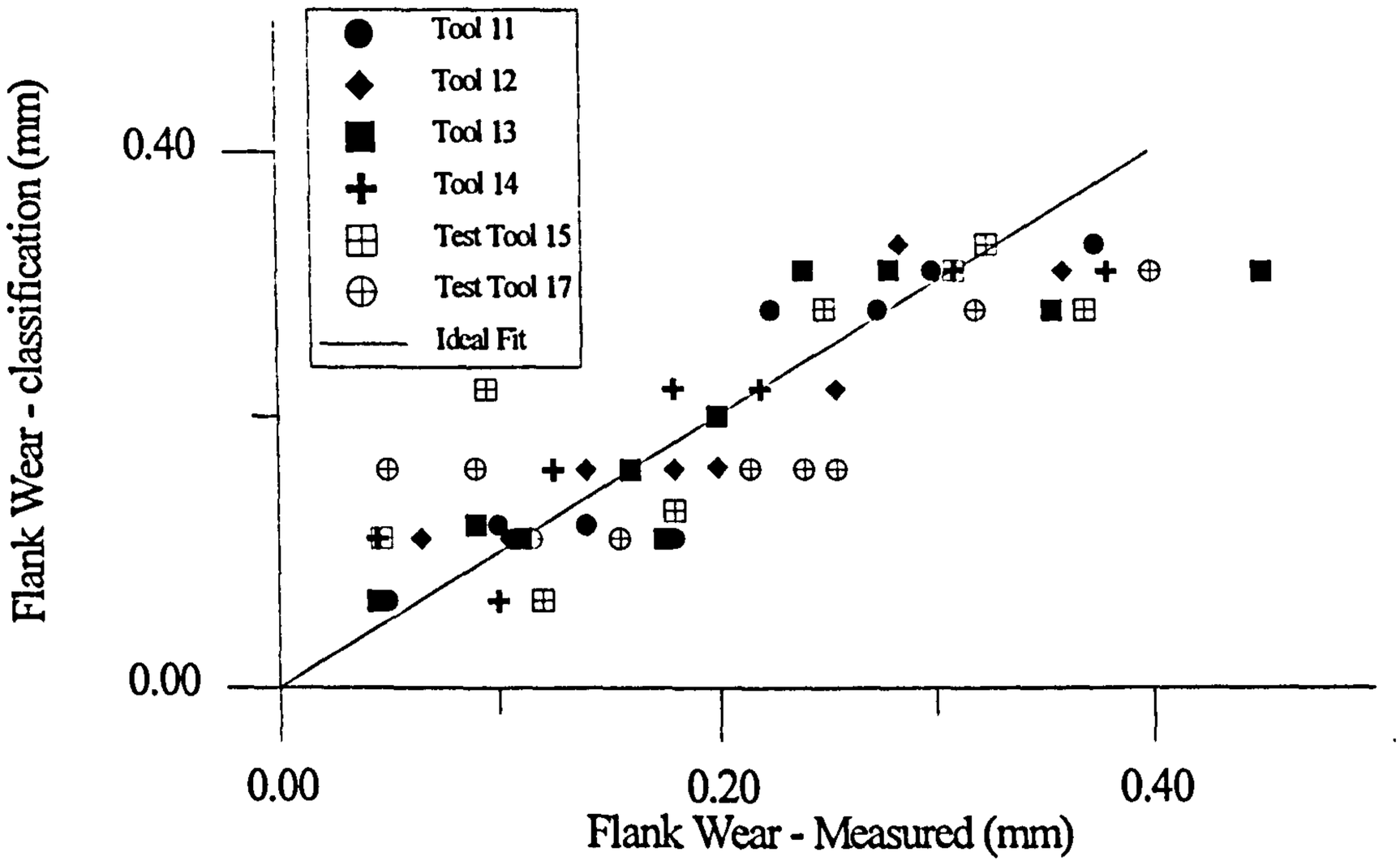


Fig. 11: ART2 classification with full features set

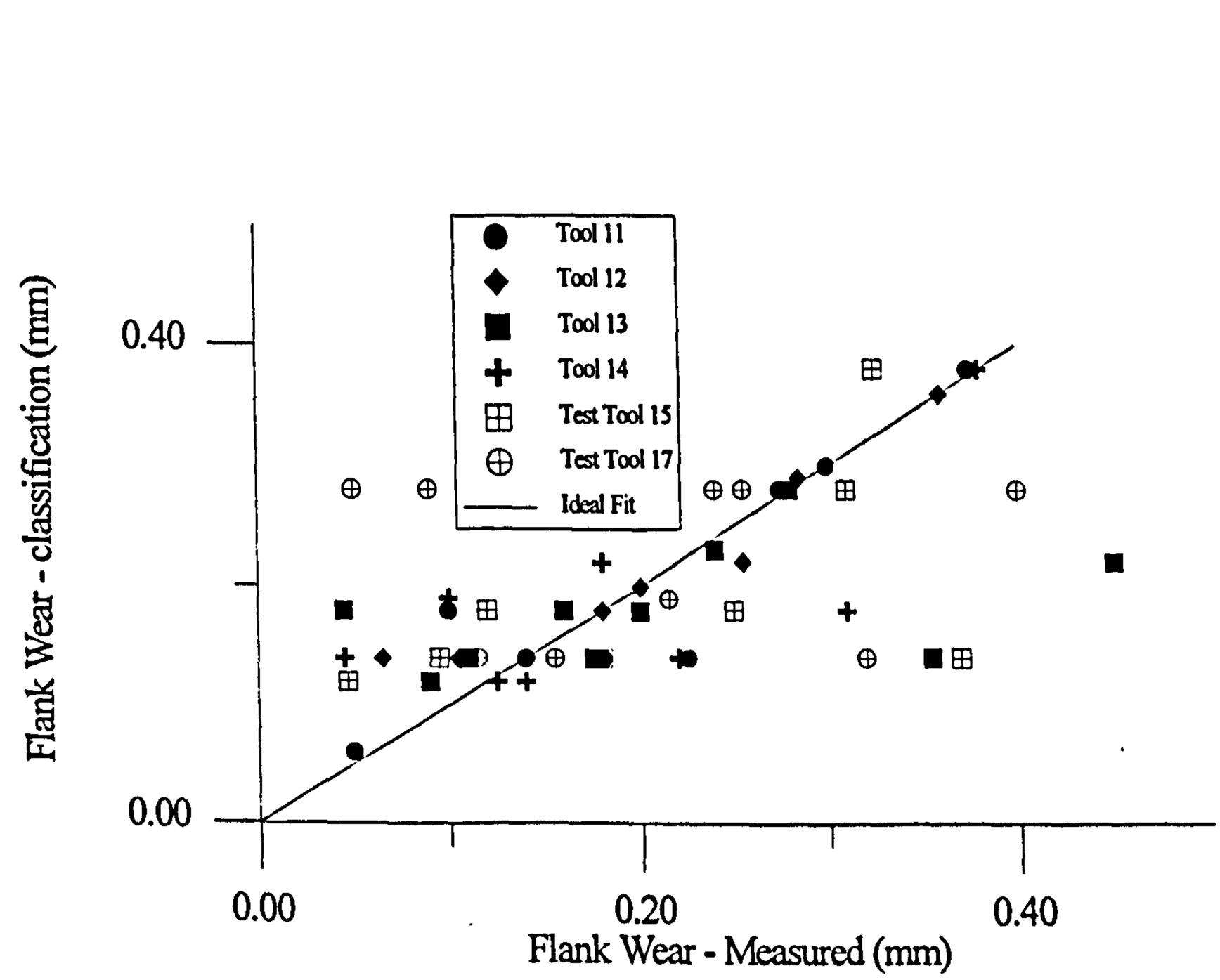


Fig. 12: SOM Classification with reduced features set

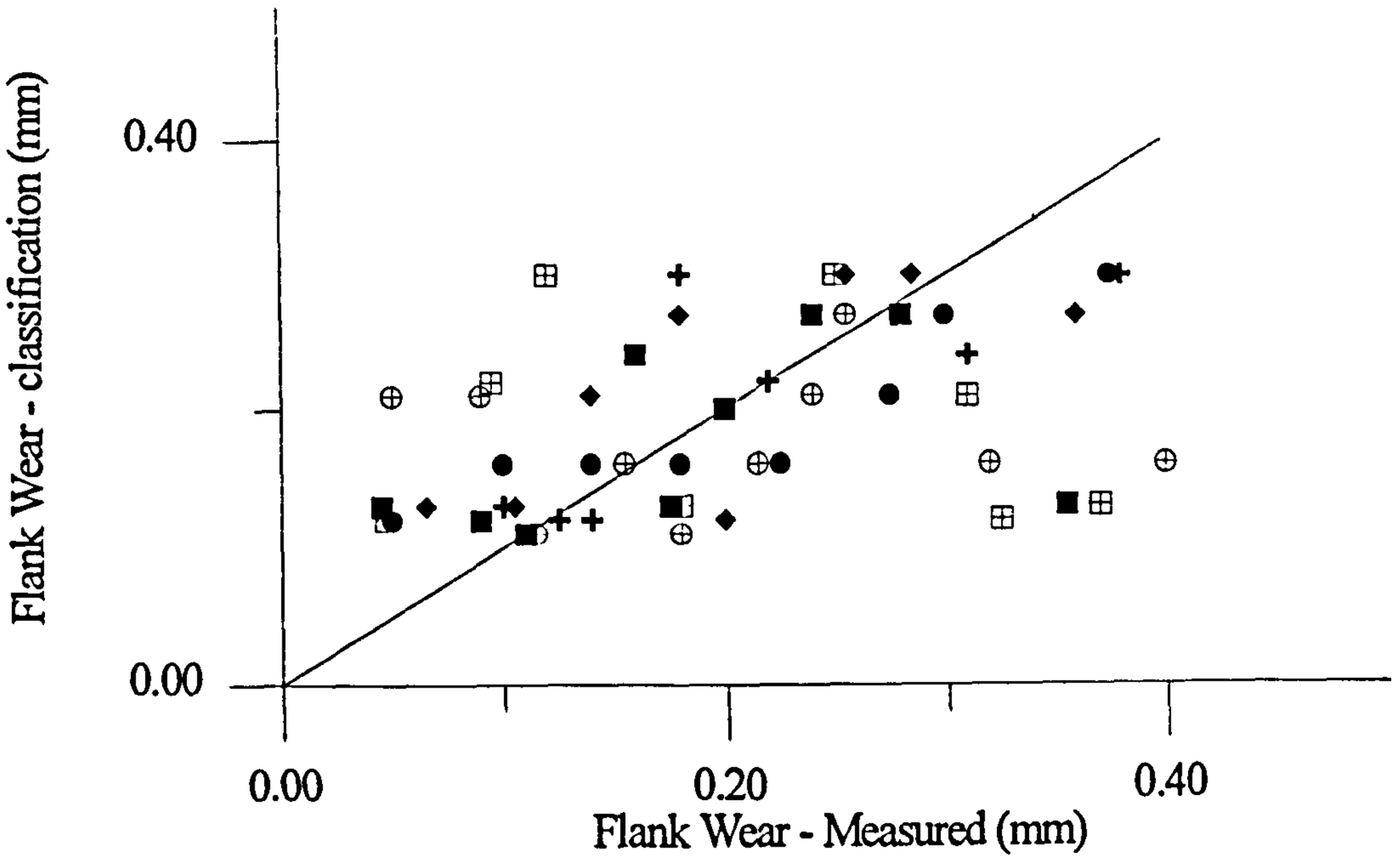


Fig. 13: ART2 Classification with reduced features set

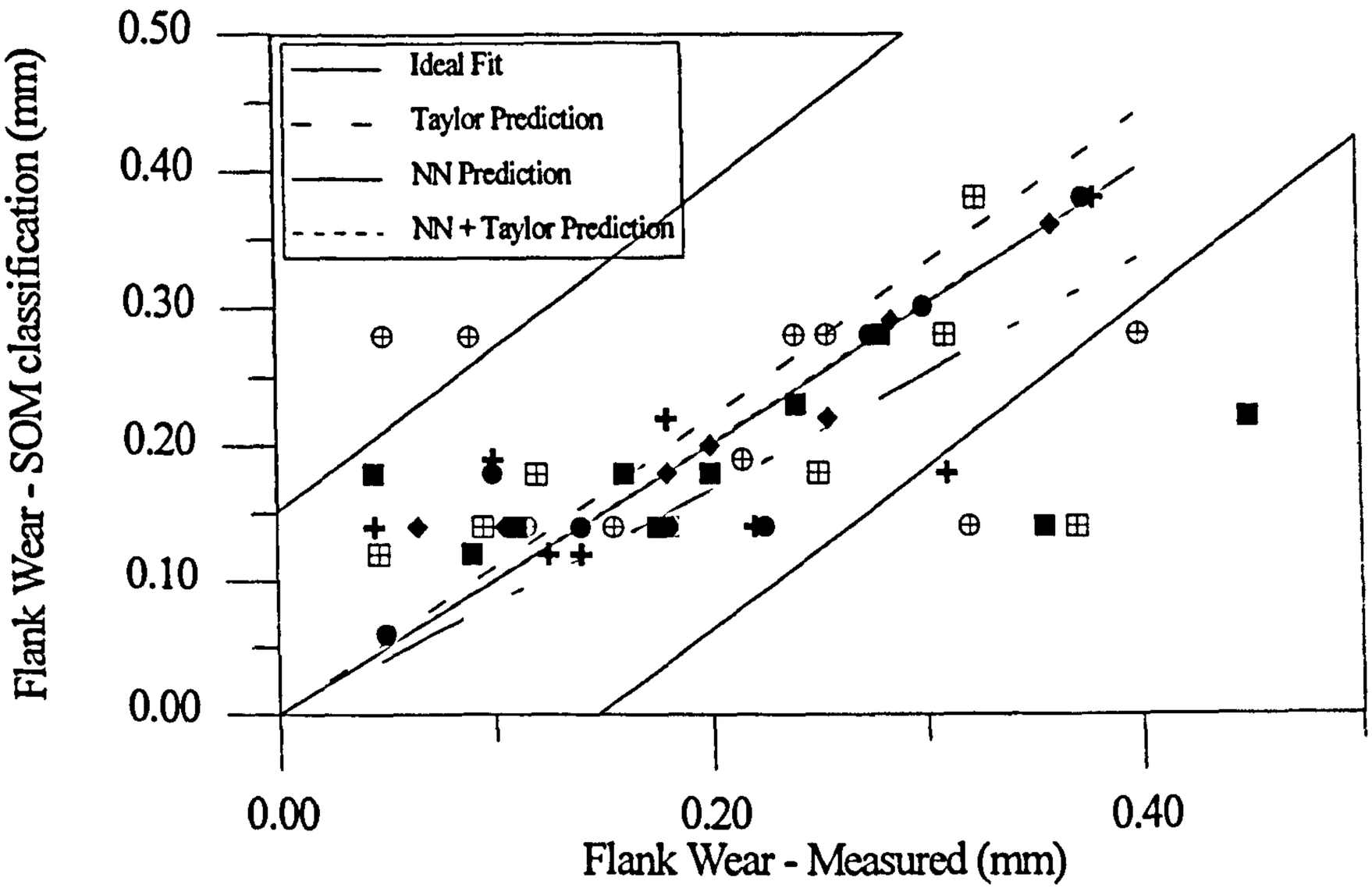


Fig. 14: Supervised classification of data from Fig. 12

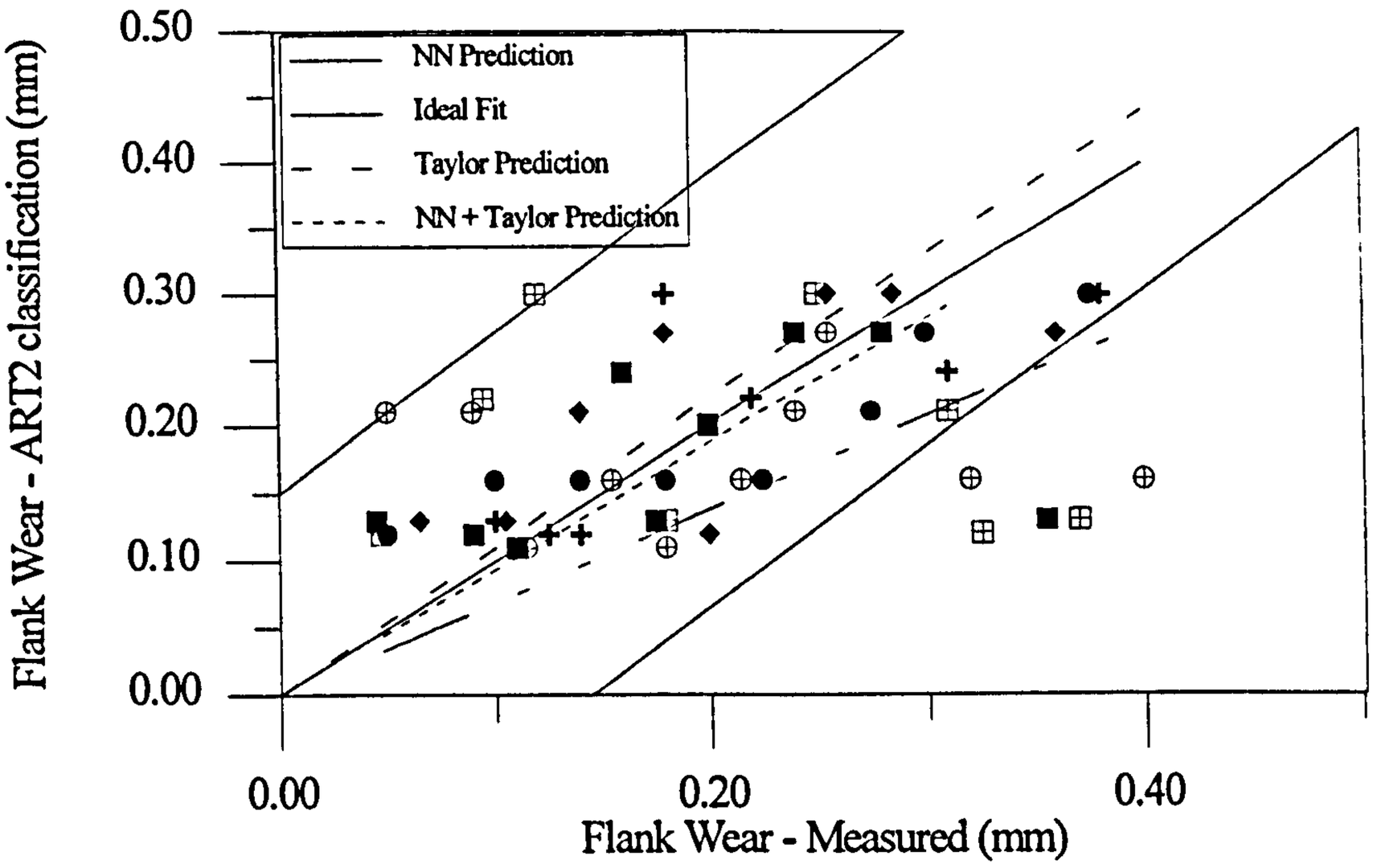


Fig. 15: Supervised classification of data from Fig. 13

Table 1 Data analysis methods

VARIABLE	STATISTICAL ANALYSIS	TIME ANALYSIS
Sound	Mean, Absolute Deviation, Kurtosis, Skewness	FFT, Wavelet Transf.
Vibration	Mean, Absolute Deviation, Kurtosis, Skewness	FFT, Wavelet Transf.
Feed Force	Mean	
Tangential Force	Mean	
Spindle Current	Mean	

Table 2 Experimental Conditions

Cutting Speed	350 m/min.
Depth of cut	1.0 mm
Feed rate	0.25 mm/rev
Insert type	TP25C grade coated cemented carbide (CNMG 120408)
Workpiece	Free cutting mild steel (EN1A), 135 mm in length, Ø 75 mm

Table 3 Instrumentation

SENSOR	DESCRIPTION	MOUNTING
Accelerometer	Kistler 8752A50 & Piezotron Coupler - Kistler 5108	Base of the turning centre
Microphone	ECM-1028, matching amplifier	Tool Post
Strain gauges	Two half Wheatstone bridge	Feed and tangential direction
Current Meter	CNC built in sensor	

Table 4 Statistical analysis of graphical results

NN / Feature Set	SD (mm)	95% Confidence Limits	
SOM - Full feature set	0.0480	0.0058	-0.0209
SOM - Reduced feature set	0.0880	0.0198	-0.0296
ART2 - Full feature set	0.0549	0.0067	-0.0239
ART2 - Reduced feature set	0.0958	0.0194	-0.0345

Table 5 Linear regression parameters for Fig. 10 to Fig. 13

TESTS	b	r
ART2 - Full feature set	0.930	0.960
Test set	0.858	0.924
ART2 - Reduced feature set	0.902	0.896
Test set	0.689	0.691
SOM - Full feature set	0.946	0.964
Test set	0.871	0.946
SOM - Reduced feature set	0.868	0.894
Test set	0.836	0.782

Table 6 Performance evaluation of improved classification

TEST SET	b		r	
	NNs	NNs + Taylor	NNs	NNs + Taylor
ART2 - Reduced feature set	0.689	0.935	0.691	0.872
SOM - Reduced feature set	0.836	1.005	0.782	0.955