Intelligent Drill Wear Condition Monitoring using Self-Organising Feature Maps

Jesal Ashar

A thesis submitted to Auckland University of Technology in fulfilment of the requirements for the degree of Master of Philosophy (MPhil)

2009

School of Engineering

Primary Supervisor: Dr. Guy Littlefair

Abstract

The rising demand for exacting performances from manufacturing systems has led to new challenges for the development of complex tool condition monitoring techniques. Although a wide range of monitoring methods have been investigated and developed, there has been very little migration of these innovations into industrial practice. The principal factor behind this phenomenon is the stochastic nature of the environment in which the system must function. A truly universal application has yet to be developed. The work presented here centres around the application of an unsupervised neural network model to the said problem. These networks learn without the aid of a human teacher or supervisor and learn to organise and re-organise themselves in accordance to the input data. This leads to the network structure reflecting the given input distribution more precisely than a predefined model, which generally follows a decay schedule. The dynamic nature of the process provides an evaluation of the underlying connectivity and topology in the original data space. This makes the network far more capable of capturing details in the target space. These networks have been successfully used in speech recognition applications and various pattern recognition tasks involving very noisy signals. Work is in progress on their application to robotics, process control and telecommunications.

The procedure followed here has been to conduct experimental drilling trials using solid carbide drills on a Duplex Stainless Steel workpiece. Duplex Stainless Steel was chosen as a preferred metal for drilling experiments because of this high strength, good resistance to corrosion, low thermal expansion and good fatigue resistance. During the drilling trials, forces on the

workpiece along the x, y and z axes were captured in real time and moments of the forces were calculated using these values. These three axial forces, along with their power spectral densities and moments were used as input parameters to the Artificial Neural Network model which followed the Self-Organising Map algorithm to classify this data. After the network was able to adapt itself to classify this real world data, the generated model was tested against a different set of data values captured during the drilling trials. The network was able to correctly identify a worn out drill from a new drill from this previously unseen set of data. This autonomous classification of the drill wear state by the neural network is a step towards creating a “universal” application that will eventually be able to predict tool wear in any machining operation without prior training.

Acknowledgements

I am indebted to the following:

Dr. Guy Littlefair for his continuous guidance and support throughout the development of this research.

Iscar Pacific for providing the tools which were used during experimental trials.

Contents

Figures

For my Shonu – my best friend and soulmate.

Chapter 1 Introduction

The condition monitoring based maintenance philosophy is emerging to be the key component in lowering operating costs and increasing machine availability. Society the world over is continually demanding less expensive products which must also provide greater technological advancement along with aesthetic refinement. These wants are driving the manufacturing industry to surpass previously impossible productivity and quality heights. The process of cutting and machining is no doubt the most widely-used mechanical processes in the industry and the cutting tool is still the central, crucial element in a huge array of manufacturing procedures thus placing an ever increasing burden on the performance of the cutting tool edge.

To obtain high levels of productivity, it is imperative for manufacturers to focus their attention on achieving automated processes with the least amount of human supervision. Successful capacious tool wear condition monitoring would ultimately lead to the optimised use of the tool cutting edge and thus present the manufacturer with productivity gains so placing their business in a strategic operating position. However, although a wide range of monitoring methods have been investigated and developed, there has been very little migration of these innovations into industrial practice (Rehorn, 2005). The principal factor behind this phenomenon is the stochastic nature of the environment in which the system must function. A truly all-encompassing application has yet to be developed.

The rationale behind the research presented here, focuses on the requirement for a sophisticated monitoring methodology capable of demonstrating its robustness, competency and accuracy when applied to demanding industrial environments. The work reported in this thesis is particularly centred on providing a greater insight into the development of a universal tool condition monitoring system. It also investigates the application of a self-organising feature map neural network to the said problem.

Chapter 2 Background

2.1 Tool Wear and Tool Life

Machining of metals is done mainly to achieve a higher level of surface finish, close tolerance, and to form complex geometric shapes which are otherwise difficult to obtain. This is done by the process of metal removal during the manufacture of components. A machine tool is one which removes the metal from a workpiece in the form of metal chips. Tools are integral parts of any machine, since without them no component can be finished. Under idealised conditions, the useful life of a tool is defined in terms of the amount of wear on one of the two primary working surfaces of the tool (KIM, 2002). The wear regions which develop on these two surfaces are termed as “crater wear” and “flank wear”, either of which will ultimately lead to the failure of the tool. Tool wear itself , refers to the degradation of the cutting tool concerning the general wear of the cutting or clearance surface, fracture and reduction of the tool mechanical properties due to high temperature (Dimla, 2000). The life of a cutting tool is thus determined by the amount of wear that has occurred on the tool profile which reduces the efficiency of cutting to an intolerable level, or eventually causes tool failure. When the tool wear reaches an initially accepted amount, there are two options:

i. To re-sharpen the tool on a tool grinder.
ii. To replace the tool with a new one.

The second possibility applies in two cases:

i. When the resources for tool re-sharpening are exhausted.

ii. The tool does not allow for re-sharpening, e.g. in case of indexable carbide inserts.

Gradual wear occurs at two principal locations on a cutting tool. Accordingly, two main types of tool wear can be distinguished- crater wear & flank wear.

These two wear types are illustrated in Fig. 1 and Fig. 2:

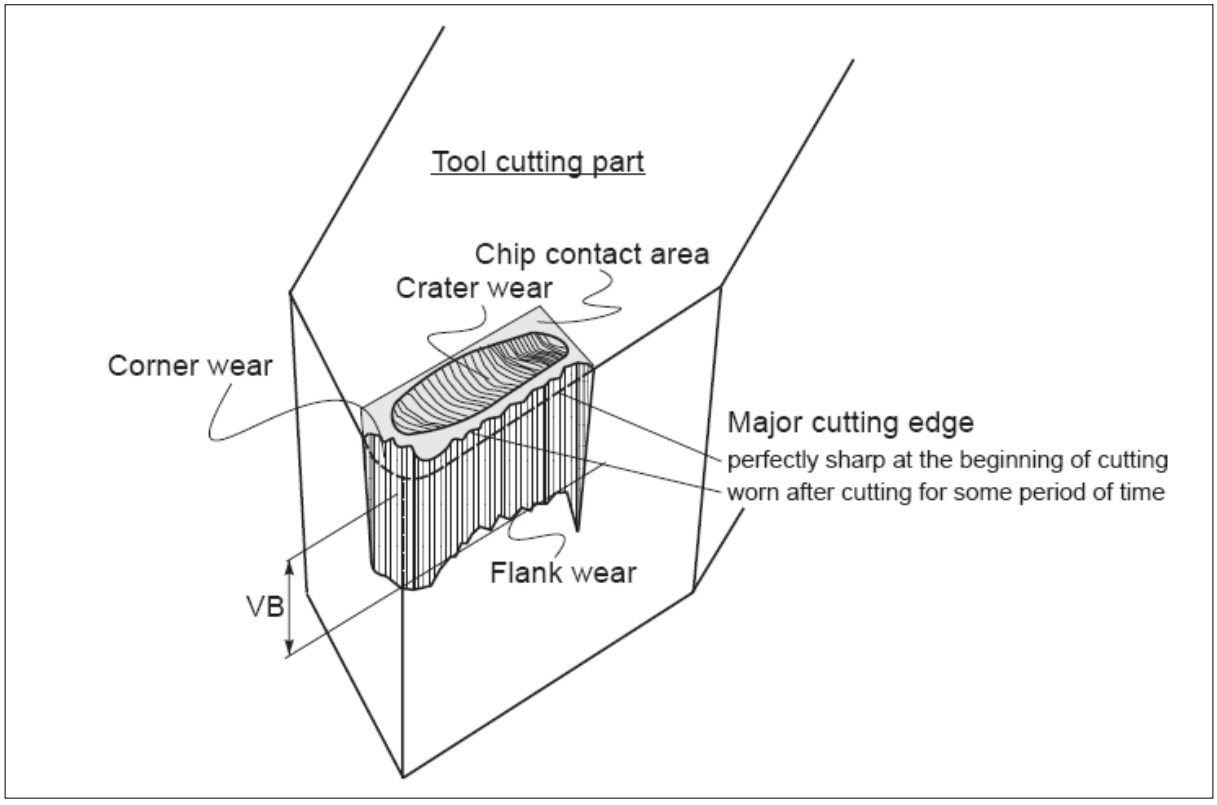


Figure 1: Types of wear observed in cutting tools

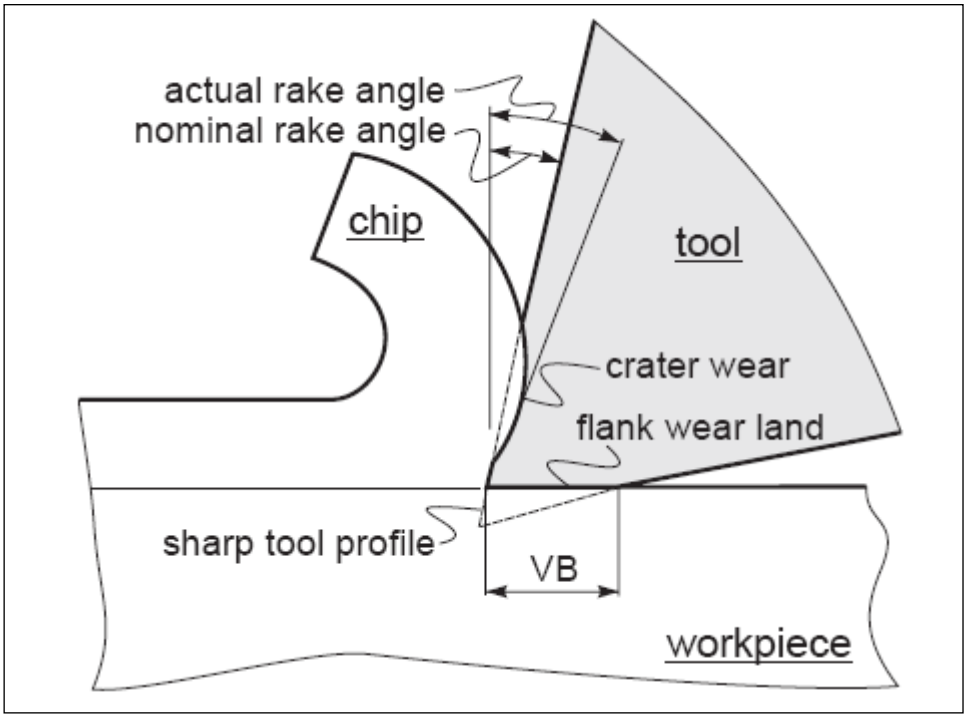


Figure 2: Cross-section perpendicular to the major cutting edge of a worn cutting tool showing the effect of crater wear on the tool rake angle and flank wear land.

- *Crater wear*: consists of a concave section on the tool face formed by the action of the chip sliding on the surface. Crater wear affects the mechanics of the process increasing the actual rake angle of the cutting tool and consequently, making cutting easier. At the same time, the crater wear weakens the tool wedge and increases the possibility for tool breakage. In general, crater wear is of a relatively small concern.
- *Flank wear*: occurs on the tool flank as a result of friction between the machined surface of the workpiece and the tool flank. Flank wear appears in the form of so called wear land and is measured by the width of this wear land, VB. Flank wear affects, to a great extent, the mechanics of cutting. Cutting forces increase significantly with flank wear. If the amount of flank wear exceeds some critical value (VB > 0.5~0.6 mm), the excessive cutting force may cause tool failure. It is

important to understand where these critical conditions exist and how they relate to cause tool wear.

Tool wear is a time dependent process. As cutting proceeds, the amount of tool wear increases gradually. But tool wear must not be allowed to go beyond a certain limit in order to avoid tool failure. The most important wear type from the process perspective is the flank wear; hence the parameter which has to be controlled is the width of flank wear land- V_B. This parameter must not exceed an initially set safe limit, which is about 0.4 mm for carbide cutting tools. The safe limit is referred to as allowable wear land (wear criterion), V_{Bk}. The cutting time required for the cutting tool to develop a flank wear land of width V_{Bk} is called tool life, T. This is a fundamental parameter in machining. The general relationship of VB versus cutting time is shown in the Fig. 3 (wear curve).

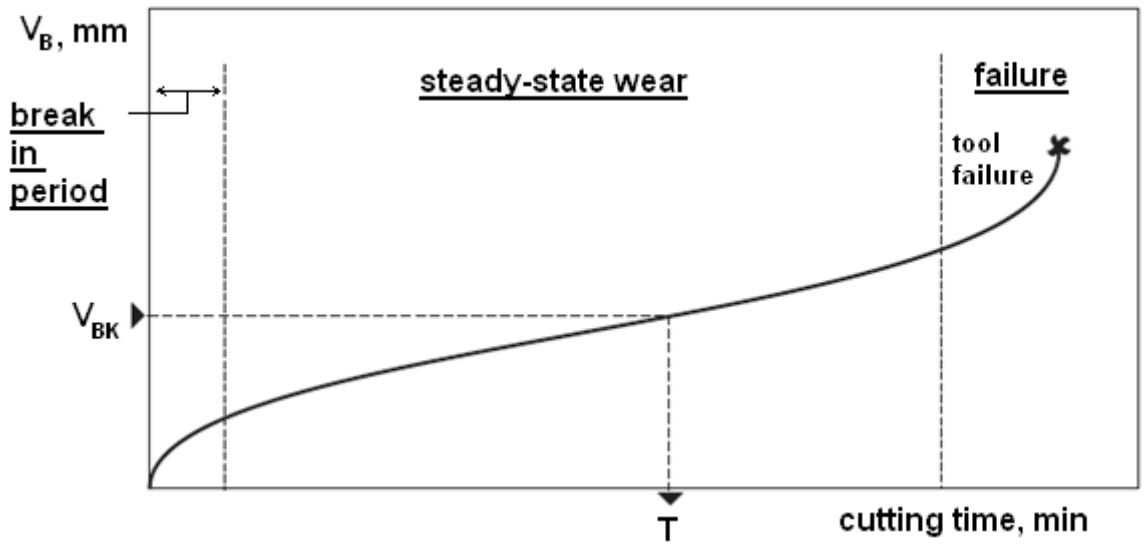


Figure 3: Flank Wear as a function of cutting time. Tool life T is defined as the cutting time required for flank wear to reach the value of V_{Bk}.

The slope of the wear curve (that is the intensity of tool wear) depends on the same parameters, which affect the cutting temperature as the wear of cutting

tool materials is a process extremely temperature dependent. Parameters which affect the rate of tool wear are:

- cutting conditions (cutting speed V, feed f, depth of cut d)
- cutting tool geometry
- properties of work material

Cutting speed is the most important amongst these parameters. As cutting speed is increased, wear rate increases, so the same wear criterion is reached in lesser time. Thus, tool life decreases with cutting speed. This phenomenon is illustrated in Fig. 4:

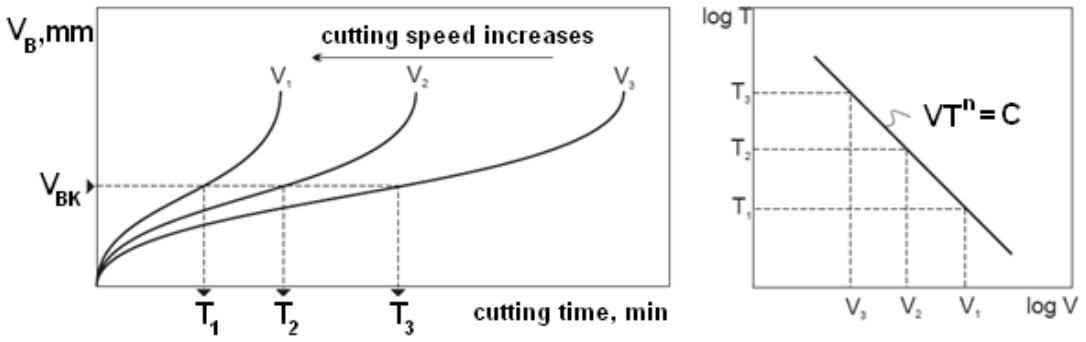


Figure 4: (Left) Effect of cutting speed on wear land width and tool life for three cutting speeds. (Right) Natural-log plot of cutting speed versus tool life.

2.2 Tool Condition Monitoring

In conventional machining environments, where a machine operator is present, a pre-determined tool life based on empirical data is generally sufficient. In contrast, underlining the concept of unmanned machining is the constant provision of tooling which can continually produce components to a specified tolerance level. Besides the disadvantage of the added cost element associated with this option, there is also the high wear rate phenomenon of the new cutting edge. The exceptionally complex problem of developing a mathematical model robust enough to provide accurate prediction of tool life, over the entire range of metal cutting conditions, is yet to be developed. Although some models do exist, these are very process specific and consequently cannot be universally applied (Dimla, 1997).

Tool life testing is the apparent solution, nonetheless, it is a very expensive and time consuming process (Noori-Khajavi, 1993). Even after exhaustive testing, one can only ascertain a range of cutting conditions applicable for a given workpiece material and a narrow band of part geometries, finishing requirements etc. The growing range of tool material substrates and coatings further compounds the modelling and testing problems. In the absence of an adequate solution for predicting tool life, some form of monitoring becomes essential. The monitoring system must be capable of dealing with the regular tool wear modes in addition to the irregular and catastrophic modes which are almost impossible to model.

Developments in manufacturing systems have necessitated more efficient metal cutting processes. Conventionally, cutting tools have been replaced at the end of a programmed, experimentally derived, in-cut time. However, in modern machining environments, a large number of variables co-relate and the application of traditional techniques leads to uneconomical and unproductive tool utilisation (Littlefair, 2007). However, the success of industrially deployed monitoring systems has been poor (Brophy, 2002). Probably the greatest single obstacle preventing the realisation of the "factory of the future" is the lack of a reliable and comprehensive tool condition monitoring system. The reasons that make tool condition monitoring systems important are:

- Unmanned production is possible only if there is a reliable and efficient method available for tool wear monitoring and tool breakage recognition.
- Tool wear controls the quality of the surface finish and the dimensions of the parts that are manufactured.
- In contemporary systems, tool changes are made based on conservative estimates of tool life. These approximations do not take into account sudden failures and at the same time lead to unreasonably high number of changes, since the comprehensive lifetime of the tools is not taken into account. Subsequently, valuable production time is lost.
- As a consequence, automated production control is not actually possible without a robust means for tool wear monitoring. (Jantunen, 2002)

The problem of machine tool downtime continues to plague the industry. Downtime can be considered as any duration of time during which no machining

operation is being performed on a given workpiece. One predicament is that there are several different sources that contribute to downtime, some of which are inevitable. It is often necessary to transfer workpieces from one machine tool to another, which requires dismantling and set-up time. Furthermore, machines need to undergo periodic maintenance to ensure their continued functioning under normal circumstances. However, there is another type of downtime that could be avoidable - downtime caused by the excessive wear and breakage of cutting tools during machining operations. Tool breakage is a major cause of unscheduled stoppage in a machining environment, and is costly not only in terms of lost time, but also in terms of capital destroyed.

Techniques for on-line wear monitoring can be grouped into two main categories: direct sensing and indirect sensing techniques. While direct methods of wear measurement have been attempted (Kurada, 1997), the popular methods have been indirect. Direct methods are less beneficial since the cutting area is largely unreachable making on-line monitoring impossible while the tool is engaged in cutting. These methods include touch trigger probes, optical, radioactive, proximity sensors and electrical resistance measurement techniques. Indirect methods take measurements while the tool is actively engaged, since it involves recording a variable that can be correlated to tool wear (i.e. indirect methods measure factors that result as a consequence of tool wear). Commonly used methods include cutting forces, acoustic emission, temperature, vibration, spindle motor current, torque and strain. These factors reflect far more than tool wear alone and parameters associated with tool wear must therefore be extracted from them and correlated to give a measure or extent of tool wear. The main practical drawback with this popular method is the need for calibration of the associated parameters in monitoring the cutting

process. The cutting conditions (speed, feed-rate and depth of cut) are known to affect the sensor signals and a range of methods have been suggested for separating the effects of these conditions from those of wear on the measured parameter (Sanjay, 2005). It is important to consider that it is necessary to monitor tool wear in order to establish the condition of the component being machined. Therefore, tool condition monitoring (TCM) is ultimately concerned with the end product.

2.2.1 TCM developments in drilling

Usually, reviews of TCM research present key findings with respect to the information processing methodologies that generated them. However, this often clouds the problem, and a cursory review of the available literature on TCM research can give the notion that everything has by now been done by somebody at some time. This is certainly not the case. To gain the most from previous research in the field of TCM, it is vital to consider the work performed in relation to the type of machining operation studied, note the important conclusions, and scrutinise the trends that have developed in this field.

In order to realise improved productivity and better quality, the monitoring of drill wear is an important issue. Since drill wear directly affects the machining quality and tool life, online monitoring and prediction of drill wear is a vital area of research. Drilling is a complex three-dimensional material removal operation. In drilling, the two points of the drill wear alternatively until they both have zero clearance at the margin, and become lodged within the workpiece. At this point, the drill will break if cutting is continued. In addition, chip flow creates significant friction between the cutter and the workpiece inside the drill hole. These

frictional forces can significantly change the dynamics of the system and can cause the cutter to break. Drills, like other cutters, can fail either from breakage or excessive wear. It has been determined that drills of a diameter less than 3 mm tend to fail by fracture, while larger tools will fail by excessive wear (Rehorn, 2005). Drill wear is a progressive development which takes place at the outer margin of the flutes of the drill due to the close contact and elevated temperatures between the tool and workpiece. However, under constant cutting conditions drill failure is a stochastic process. The reasons for altering drill life are the heterogeneities in the workpiece and drill materials, the irregularities in the cutting fluid motion and the inescapable unevenness introduced during the grinding of the cutting edges (Abu-Mahfouz, 2005).

Several works have already been reported in the broad area of tool condition monitoring. Abu-Mahfouz (Abu-Mahfouz, 2005) reported developing and implementing two supervised vector quantization neural networks for estimating the flank wear size of a twist drill. The two algorithms used were the learning vector quantization (LVQ) and the fuzzy learning vector quantization (FLVQ). They studied the effect of vibration signals on predicting drill wear. They also reported to have a success rate of 88% of drill wear prediction using Artificial Neural Networks (ANN). The ANNs were found to satisfactorily accommodate changes in the cutting conditions. Brophy, Kelly and Byrne (Brophy, 2002) designed a two-stage ANN to detect anomalies in the drill wear process. They trained their network to distinguish drill wear as “normal” or “abnormal”. They used spindle power as an input to the ANN rather than using statistical data extracted from a signal. They reported to have no false alarms when their network was tested using data acquired from 33 sets of tests. Panda et. al. (S. S. Panda, 2006) presented work dealing with the development a fuzzy back-

propagation neural network scheme for the prediction of drill wear. They conducted drilling operations over a range of cutting conditions. Spindle speed was varied along with the feed rate. High-speed steel (HSS) drills of different diameter sizes were used for drilling holes in a mild steel plate. Various combinations of spindle speed, feed rate, and drill diameter were used to perform 52 different drilling operations. They used spindle speed, feed rate, drill diameter, thrust force and torque as inputs to their ANN. They reported that the best neural network architecture (i.e. the number of neurons, learning rate and error co-efficient) was obtained by trial and error based on mean square error (MSE) in training, testing, and the number of iterations. Franco-Gasca et. al. (FRANCOGASCA, 2006) describe a driver current signal analysis to estimate the tool condition by using the discrete Wavelet Transform. This was used to extract the information from the original cutting force and through an autocorrelation algorithm tool wear was evaluated in the form of an asymmetric weighting function. The current was monitored from the motor driver to give a sensor-less approach. Experimental results presented claimed to show the algorithm performance was as expected. Jantunen (Jantunen, 2002) presents a summary of the methods applied to condition monitoring in drilling. The author concludes that in signal analysis, statistical parameters obtained from the time domain signal are extensively used. Fast Fourier and Wavelet Transform have also been used for tool wear and breakage detection by a number of research groups, but only a limited number of automatic diagnostic tools have been developed for diagnosis of the condition of the tool in drilling.

2.2.2 Comments on TCM in Drilling

The most commonly monitored variables are torque, feed force and cutting speed. A common method of signal processing in all TCM research is that of the Fast Fourier Transform (FFT), which is used to generate a power spectral density (PSD) function. However, the averaging natures of the FFT and PSD calculations tend to obscure the frequency content of transient and burst phenomena, such as breakage. The practice of measuring cutting forces is common to all machining research, but the use of torque is almost unique to drilling. The bulk of signal processing in drilling-based TCM research focuses on time domain methods and the use of statistical parameters. The use of the RMS of the signals collected is also widely accepted as a standard practice. When compared to other statistical parameters, including arithmetic mean, standard deviation, skewness, kurtosis, maximum and minimum, it was found that the RMS never performed best, although it was among the top four (El-Wardany, 1996).

A new type of statistical operation, known as the instantaneous ratio of absolute mean value ($RAMV_i$) is suggested in (El-Wardany, 1996). This is calculated by taking the ratio of the instantaneous absolute value of the measured variable's mean (AMV_i) to the absolute of the mean value at the start of drilling, (AMV_b). The AMV_b is a baseline value, hence the use of the subscript "b." Thus, it is: "a normalized mean value calculated with a time constant of one revolution."

It is apparent from the research done so far that further investigation is required in establishing systems for online applications in an industrial environment. Future research should be aimed at developing systems capable of deriving tool wear parameters from multiple data sensors and combining this data to provide a robust indicator of tool condition. Detailed investigations have shown that artificial neural networks trained under constant cutting conditions have limited validity over a broader range of process parameters. Growing complexity is one of the most significant characteristic of contemporary manufacturing. This complexity manifests itself in manufacturing systems, in the products being manufactured, in the processes, and the company structures. These systems operate in a stochastic environment amid ambiguity (Monostori, 2003). There are a range of signals (force, torque, temperature, mechanical vibration, acoustic emission, etc.) which co-relate to the state of the manufacturing process. These signals are the subjects of diverse control and monitoring algorithms. The intricacy of the problem and the associated uncertainties demand the application of novel techniques to realise fully automated sophisticated systems. This problem complexity further creates major issues in predicting tool wear accurately. Tool wear depends on a large number of factors including the properties of the materials involved; the physical and chemical properties of the surfaces; pressure; temperature; friction and relative velocities. In addition, the problem is complicated with the consideration of the complex three dimensional machining operations where process and operating variables, such as feed rate, cutting speed and engagement must be taken into consideration. The cost implications of introducing a suitable monitoring strategy are difficult to establish since applications are dissimilar. However, figures as high as 40% are not unimaginable (Littlefair, 2007).

2.3 Duplex Stainless Steels

Duplex stainless steel (DSS) is a dual-phase material with equal volumes of austenite and ferrite. Its structure results in some significant engineering properties. These properties have propelled the material into the mainstream in manufacturing processes with its usage continuing to grow. There has been only a modest amount of work conducted on the machinability of duplex steels, although there have been several studies talking about its general machining related topics. To aid in the greater adoption of Duplex Stainless Steels into the specialised engineering component sector, detailed analysis of its machinability and its post-machining microstructure is critical.

2.3.1 Metallurgy

The superior metallurgical properties of Duplex Stainless Steels stem from the mixture of Austenite (γ) and Ferrite (α) phases present in its structure. The Austenite phase is responsible for the relative ductility of the metal and its resistance to uniform corrosion; while the Ferrite phase is responsible for the superior strength as well as corrosion resistance of the metal (T. Saeid, 2008). The ideal duplex structure would consist of a 1:1 matrix of γ and α phases presenting themselves in a “banded structure” as illustrated in Figure 5. The lighter phase is Austenite and the darker phase is Ferrite (G. S. J. Reis, 2000).

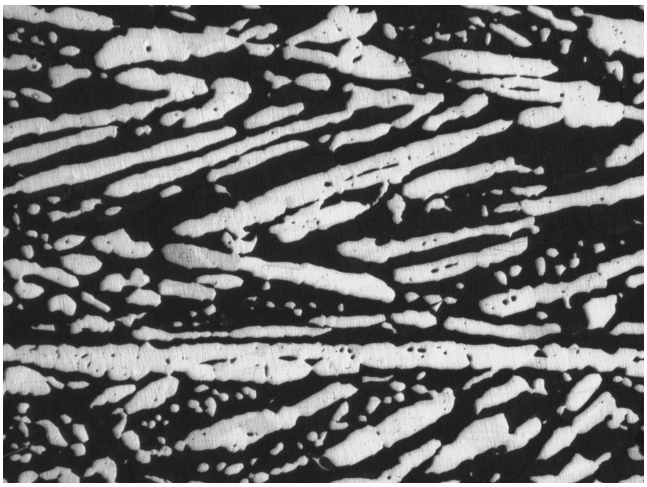


Figure 5: Typical "banded" microstructure of Duplex Stainless Steel (50x)

As seen in the figure above, both phases of Austenite and Ferrite exist in relatively large separate volumes. Also, these phases are approximately equal fractions rather than an inclusion phase embedded in the matrix formed by one of the other phases (T. Siegmund, 1995). When the material undergoes deformation during working, both phases in the metal are jointly modified - but due to an existing difference in the relative hardness of the phases, the strain distribution does not remain uniform (G. S. J. Reis, 2000; N. Jia, 2006). Strain concentrations appear in the softer ferrite phase, and this can lead to cracking, grain boundary and inter-phase sliding. Therefore, the processing of Duplex requires cautious control and monitoring of its heat treatment cycles. Poor control of temperature or soaking time prior to forging, for instance, can lead to inter-granular cracking as shown in Figure 6.

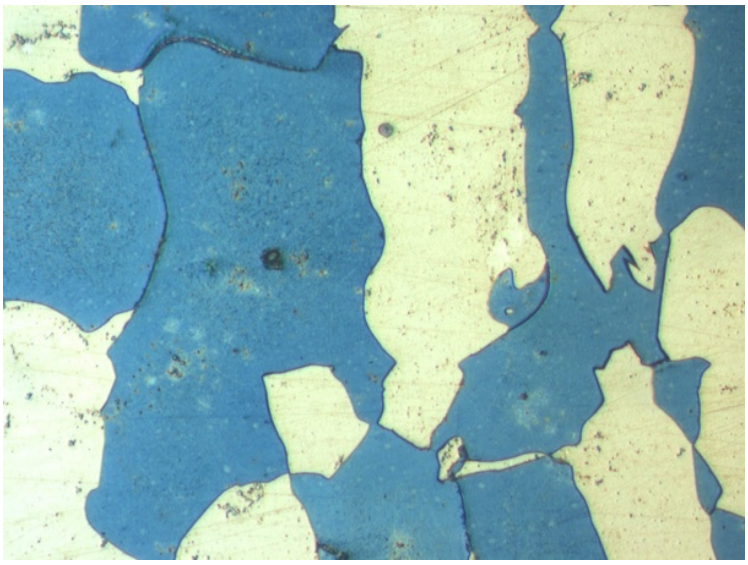


Figure 6: Inter-granular cracking of a 2205 grade Duplex stainless steel as a result of poorly controlled heat treatment prior to forging (200x)

2.3.2 Machinability

Duplex Stainless steel (DSS) belongs to a difficult to machine group of materials due to its high tendency to work harden; its high toughness and relatively low thermal conductivity (D. O'Sullivan, 2002; Dolinsek, 2003; J. Paro, 2001). Other problems with these steels stem from their high fracture toughness, increasing the tool/chip interface temperature which leads to poor surface finishes and poor chip breaking. Additionally, built-up-edge (BUE) formation, more common in the machining of ductile materials such as aluminium, is present even at elevated cutting speeds. This promotes the deterioration in the finish of the machined surface. Work hardening in austenitic stainless steels is caused due to martensite formation. Martensite is formed either due to plastic deformation or due to thermal effects, or a combination of both. In general,

the machinability of duplex is poor compared to other grades of stainless steel. This primarily stems from the high strength of the alloy but is exacerbated by a

lack of non-metallic inclusions and the low carbon content (Nilsson, 1992; Voronenko, 1997). To increase the machinability of the steel, its sulphur content can be increased but this reduces the corrosion resistance and also its ductility. A general indicator of relative machinability is presented in Figure 7 (Anon, 2001, 2005) where Austenitic 316 is used as a reference and compared to various common grades of DSS. As can be seen, the so-called lean Duplex (S32101) is more readily machined than 316 in contrast to the regular (S32205) and in particular the Super Duplex (S32507) which have poorer machinability characteristics. It should also be noted that when carbide cutting tools are used the machinability is quoted as being poorer than when High Speed Steel (HSS) tools are used. This is primarily due to the edge preparation where work hardening forms less readily with sharp cutting tools and hence the edge preparation on regular carbide tools lessens the machinability.

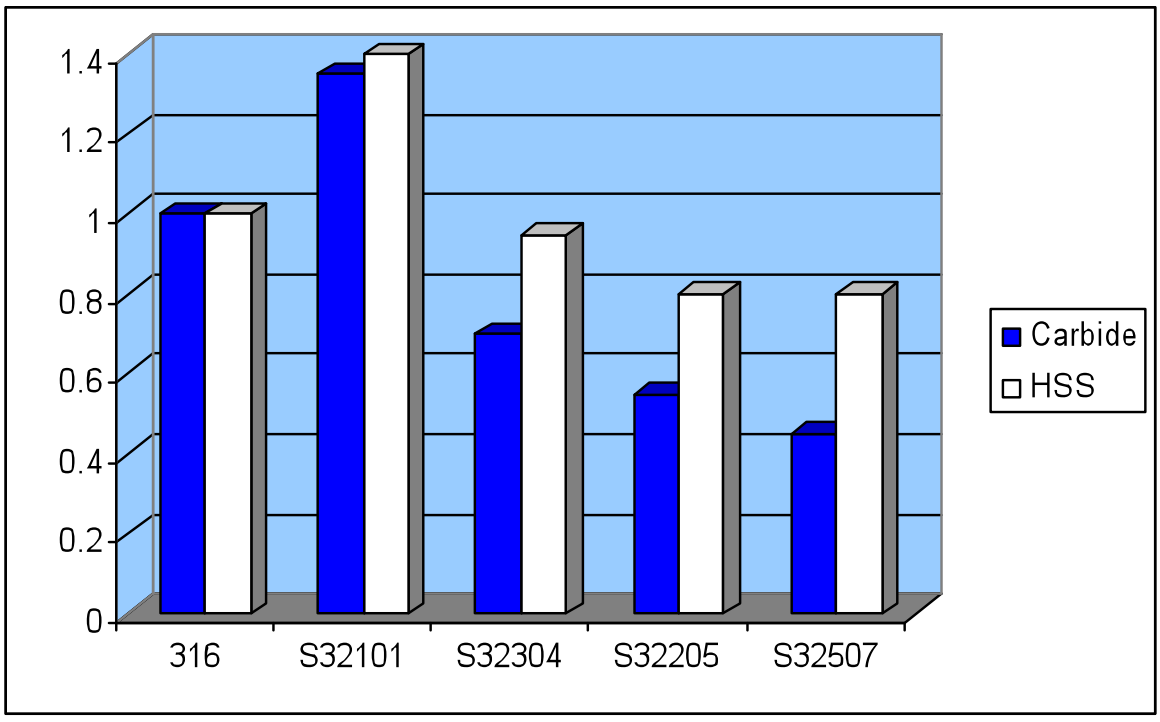


Figure 7: Relative machinability of various DSS grades referenced to

Austenitic 316 Stainless Steel (Smith, 2007).

2.3.3 Current Research

In order to further understand the deformation of the dual phase microstructure of DSS, a series of preliminary machinability trials were conducted in (Littlefair, 2008). These included both turning and drilling operations on a 2205 DSS sample. Figure 8 is a photomicrograph of a chip being formed during turning with a high-speed steel tool at 20m/min and 0.1mm/rev with a depth of cut of 1.5mm. The chip was "frozen" using a purpose built explosive quick-stop device which accelerates the tool away from the workpiece as cutting progresses (Littlefair, 2008). Both the phases can be visibly identified, particularly on the left. Of interest, is the austenite phase (lighter shade), which is the harder phase, being effectively squeezed through the softer ferrite phase as cutting progresses. In the right side of the figure, the ferrite is clearly accumulating in advance of the cutting tool with large amount of plastic deformation which due to the high forces and elevated temperature would lead to BUE on the insert given sufficient time (Littlefair, 2008). Austenite is not part of the BUE phase as can be seen by its absence in the region directly forming in advance of the tool (Littlefair, 2008).

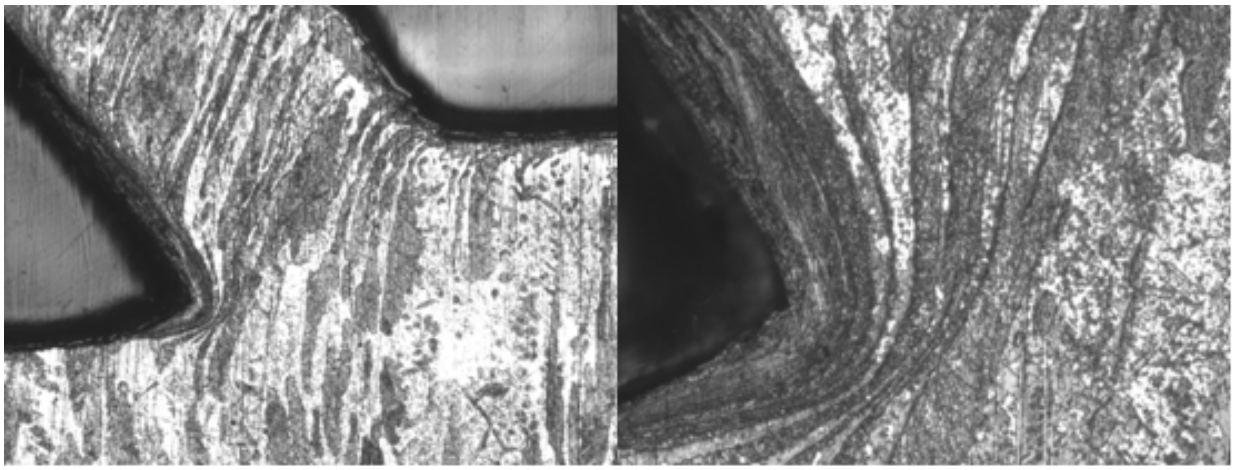


Figure 8: Photomicrographs of a quick-stop turning sample of 2205 Duplex

In a similar approach, drilling trials have been conducted on the same 2205 sample material. Figure 9 shows the bottom corner of a hole drilled with a split-point solid carbide 12.5mm drill at 45m/min and a penetration rate of 0.175mm/rev and a cutting fluid at three different magnifications. At these relatively modest parameters, there is no evidence of microstructural modification to the machined surface and only a minor amount of ferrite accumulation beneath the cutting edge of the tool.

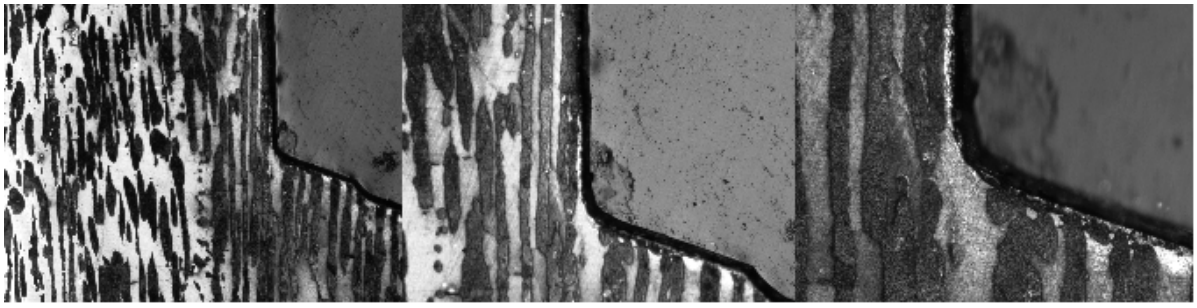


Figure 9: Drilled hole "corner" at 50x, 100x and 200x magnification

Figure 10 is a photomicrograph of a chip being formed during the same drilling process which generated the results in figure 9. Whilst it is more difficult to comprehend, as the chip is curving towards the sectioned surface due to the action of the rotating cutting edge, there are similarities to the turning chip sample with austenite being squeezed through the ferrite. In contrast to the turned sample however, there is greater segmentation of the ferrite phase indicating a more abusive cutting action.

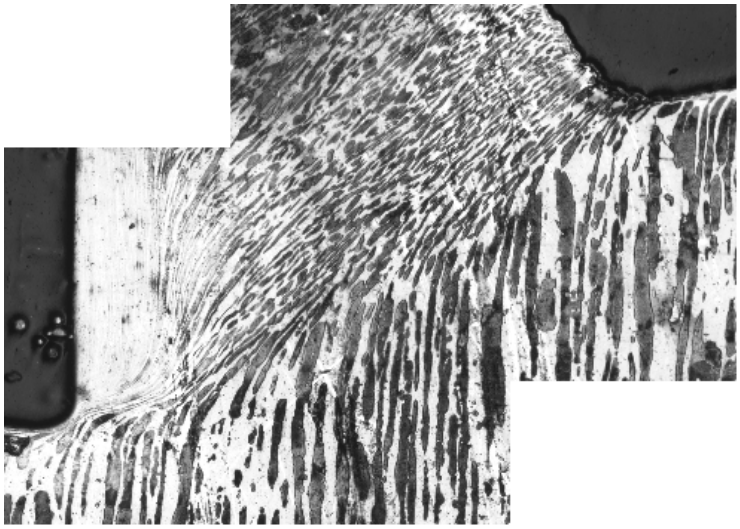


Figure 10: Photomicrograph of chip formation during drilling of 2205 DSS 50x magnification

Chapter 3 Intelligent Machines

The evidence is plain that a lack of awareness to structured tool management has resulted in the reduced performance of manufacturing systems. Plant tooling systems affect product design selections, machine loading, job consignments, capacity development, and real-time part routing assessments. With escalating automation in manufacturing systems, there is a budding need to integrate tool management scrupulously into system design, planning and control. In the course of the past decades, the field of Artificial Intelligence has advanced significantly in the direction of computerising human reasoning. Symbolic approaches are based on the hypothesis of symbolic representation—the idea that perception and cognitive processes can be modelled as acquiring, influencing, relating, and changing symbolic representations. Expert systems embody the earliest and mainly established type of intelligent systems attempting to personify the ‘‘knowledge’’ of a human expert in a computer program. Knowledge representation in these systems ensues symbolically having a structure consisting of production rules, outlines or semantic networks.

A different approach to intelligent systems involves creating computer algorithms with the structural designs and dispensation capabilities that imitate the processing characteristics of our biological nervous systems. The technology that endeavours to attain these results is called neural computing or artificial neural networks. These sub symbolic schemes work with numeric values and appear to be better suited for handling tasks involving perception and cognition, and possibly even tasks that call for combined perception and

cognition. Investigations have confirmed that—analogous to our present conception of biological compositions—adaptive ANN procedures appear to be a feasible solution for the lower level of intelligent, hierarchical control and monitoring systems where capabilities for real-time functioning, ambiguity handling, sensor integration, and learning are crucial features (Basim Al-Najjar, 2000).

3.1 Biological Intelligence

To segregate the human brain into partitions and perform a systematic analysis on how they are interconnected, we need to map and build a flow chart of signal activity, define structural roles for each section, and infer the associations and dependencies among the partitions. However, we only have a modest estimate at best about connectivity between neurons, how the neurons effect actions as an ensemble, and how information is converted to knowledge and intelligence (SG Wysoski, 2006). Attempts to replicate the connectivity of biological neuronal cells have been restricted to consider ensembles of neurons positioned equally in a two dimensional array where the connectivity may follow a given criteria (fully connected or partially connected to neighbouring cells) or be chosen arbitrarily (Kak, 2005). In an effort to discern what lies behind the way of thinking of human minds, one of the most exigent aspects for neuroscientists is that current technologies are unable to keep track of and quantify all the signals used for inter-neural communication, even in an infinitesimal segment of the brain. If this were indeed achievable, it would permit us to accurately comprehend the emergence of intelligence from an ensemble of neurons. A general practice is to complement experimental brain study with

the building of computational models supported by neuro-biological data and to study their properties by theoretical and simulation means. This is done to overcome the limitation of measuring and quantifying brain wave signals. Following this trend are the computational models of information processing with biological relevance, the so called biologically realistic neural networks.

3.1.1 Biological Neurons and Networks

As mentioned earlier, Artificial Neural Networks are modelled along the lines of biological neurons and the neuronal networks in the brain. It is therefore necessary, to consider the functioning of biological systems in order to emulate their behaviour using computers. The brain is a dense neural network consisting of an estimated 100 billion neurons that use biochemical processes to obtain, compute and convey information. A diagram of a nerve cell typical of those in the brain is shown in Figure 11. The output area of the neuron is a long branching fibre called the axon. The input area of the neuron is a set of branching fibres called dendrites.

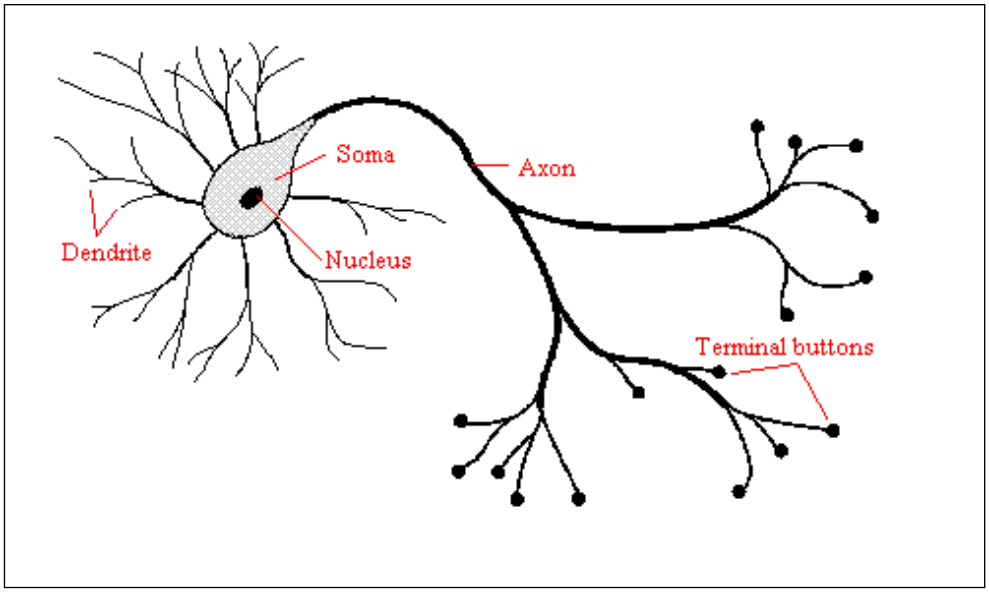


Figure 11: Schematic of a biological neuron

The dendrites of a neuron form a dendrite tree which is connected to thousands of other neurons. When any of the neurons fire, a positive or negative charge is received by a dendrite. The strengths of all the received charges are extrapolated through the processes of spatial and temporal précis. Spatial summation occurs when several weak signals are converted into a single big signal, while temporal summation converts a rapid series of weak pulses from one source into a large signal. The cumulative input is then passed to the cell body or soma. If the combined input to the neuron is greater than a threshold value, the neuron fires — i.e. an output signal is produced that is conveyed down the axon. The strength of the output is steady even if the input is barely above the threshold or multiplied several times. In addition, the output strength is not affected by the number of branches of the axon; the signal reaches each terminal with identical potency.

3.2 Artificial Intelligence

Artificial intelligence (AI) is the conception that, in principle, learning and other facets of human intelligence could be described accurately enough that a machine could be programmed to simulate it. Intelligent machines, and the branch of computer science which aims to create it, constitutes the field of Artificial Intelligence. Major AI textbooks define the field as "the study and design of intelligent agents" where an intelligent agent is a system that perceives its environment and takes actions which exploit its chances of success. John McCarthy, who coined the term in 1956, defines AI as "the science and engineering of making intelligent machines" (Skillings, 2006). This raises philosophical questions about the nature of the mind and limits of scientific hauteur. Artificial intelligence has been the subject of overwhelming optimism, has suffered dramatic setbacks and, today, has become an indispensable part of the technology industry, providing the “heavy lifting” for the most convoluted problems in computer science. The capability to build intelligent machines has intrigued humans since ancient times. With the advent of the computer and 50 years of research into AI programming techniques, this vision of creating intelligent machines is being swiftly realised. Researchers are creating systems which can mimic human thought, understand speech, beat the best human chess player, and perform a myriad of other feats never before conceived to be possible. In the pursuit to create intelligent machines, the field of Artificial Intelligence has been divided into several different approaches based on judgments about the methods and theories showing the most potential. These competing theories have led researchers into following one of two basic paths — bottom-up and top-down. Bottom-up theorists believe the best way to accomplish artificial intelligence is to build electronic replicas of the

brain's composite network of neurons, while the top-down approach attempts to impersonate the brain's behaviour with computer programs (Satoh, 2004). Expert Systems constitute a part of the top-down approach while Artificial Neural Networks emulate the bottom-up theory in their architectures.

3.2.1 Artificial Neural Networks

Neural networks are computational configurations inspired by the study of biological neural processing. The field is known by various names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning, and artificial neural networks. Artificial Neural Network architectures are motivated by models of the brain and nerve cells. Individual neurons are convoluted and have a multitude of parts, sub-systems, and control methods. Neurons exchange information by way of a range of electrochemical pathways. There are over 100 different classes of neurons, depending on the method of categorisation. Collectively neurons and their connections form a process that is not binary, established, nor synchronous (Seneker, 2002).

An artificial neural network is an attempt to simulate, within specialised hardware or by means of simulation software, the numerous layers of simple processing elements of neurons where each neuron is linked to a number of neighbouring neurons with changing coefficients of connectivity that represent the strengths of the connections. Learning is accomplished by adjusting the strength of these connections and in effect, the network outputs suitable results (J. Ashar & G. Littlefair, 2008). The basic components of a neural network are

modelled after the architecture of the brain. Some neural network structures are not strongly related to the brain and some do not have a biological counterpart in the brain. Yet, neural networks have a striking resemblance to the biological brain and, thus, share terminology from neuroscience. The fundamental unit of a neural network is the artificial neuron that simulates the basic functions of biological neurons. Artificial neurons are simpler than their biological counterparts; Figure 12 shows the elements of an artificial neuron.

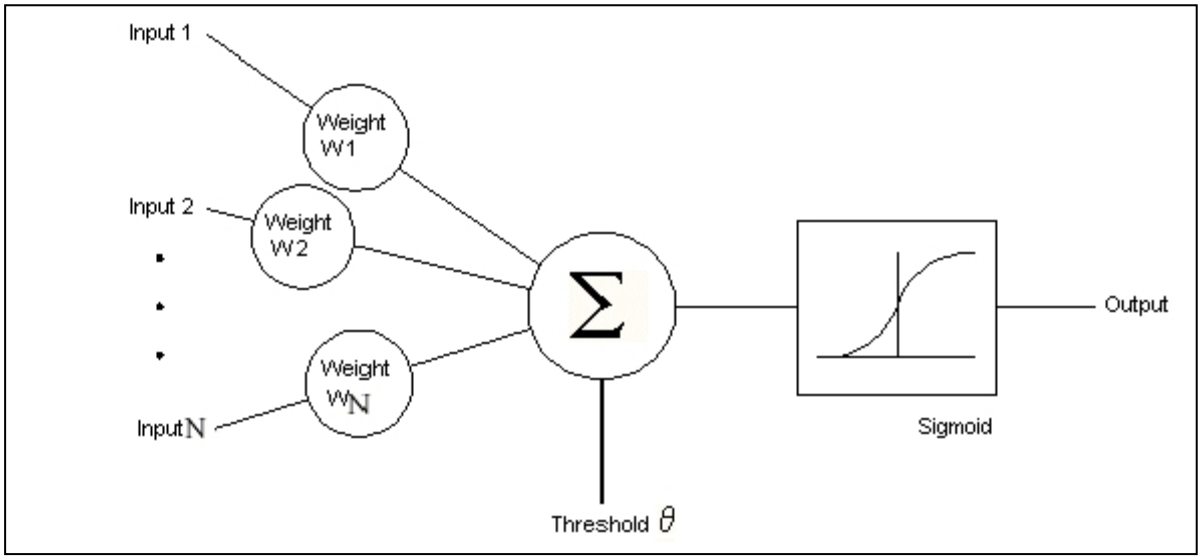


Figure 12: Elements of an Artificial Neuron

The inputs to the network are multiplied by a connection weight *Wn*. In the simplest case, these products are simply summed and processed by a transfer function to generate a result, and then an output. Although all artificial neural networks are constructed using this basic building block, the essentials may vary. Biological neural networks are constructed in three dimensions from infinitesimal components. While these neurons appear to be capable of unlimited interconnections, this is not true of artificial networks that are the bunching of simple artificial neurons. Clustering occurs by the creation of layers,

the size and number of which may vary, and then connecting these layers to one another. Essentially, all artificial neural networks have a similar topology. There is an input layer of neurons which forms the external link to receive inputs from the outside world and another layer of neurons provide the network's outputs to the outside world; all the remaining neuronal layers are the hidden or computational layers and these are hidden from view.

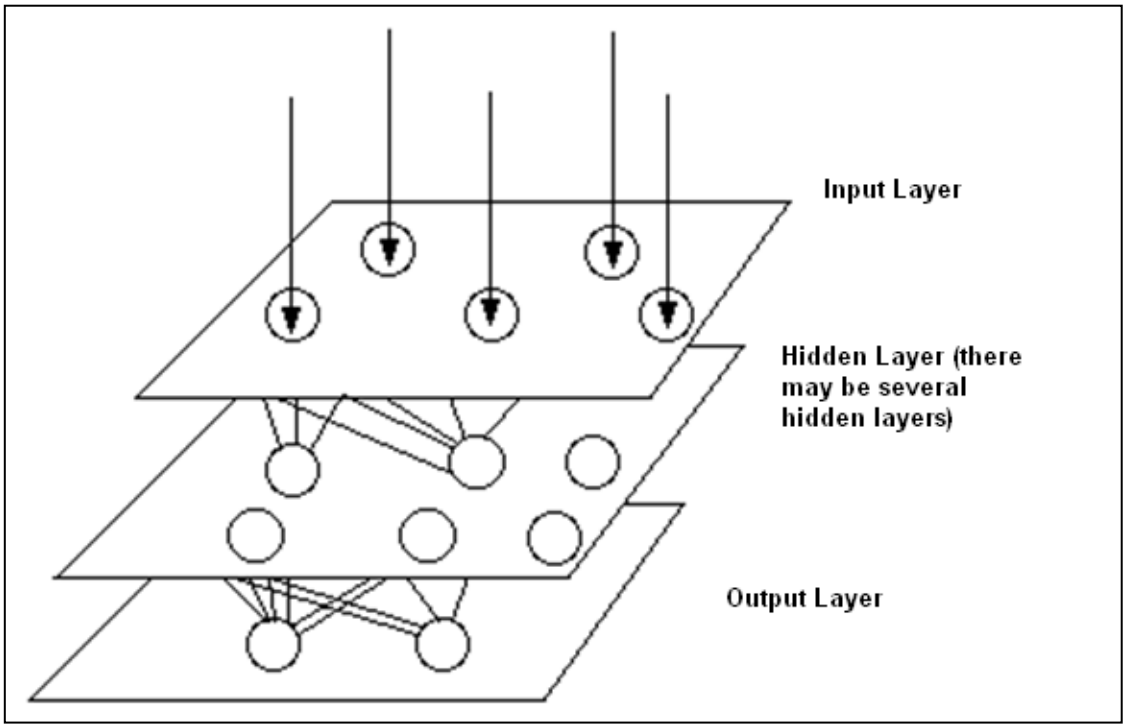


Figure 13: Layers in an Artificial Neural Network

Figure 13 illustrates how neurons in an ANN are organised into layers. The input layer consists of neurons receiving input from external sources. The output layer consists of neurons that communicate the results of the network to a user or entity. Additionally there are typically one or more hidden layers between the input and output layers, and layers are usually fully interconnected but are not required to be so (Paugam-Moisy, 2001).

ANNs can be used to model complex relationships between inputs and outputs or to find patterns in data. There is no precise agreed-upon definition among researchers as to what a neural network is, but most would agree that it involves a network of simple processing elements (neurons), which can exhibit complex global behaviour, determined by the connections between the processing elements and element parameters (J. Ashar & G. Littlefair, 2008). The original inspiration for the technique was from examination of the central nervous system. ANNs, like people, learn by example. An ANN may be configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons and this is true of ANNs as well. Neural networks with their remarkable ability to derive meaning from complicated or imprecise data can be used to mine patterns and detect trends that are too complex to be observed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

3.2.2 Learning in Artificial Neural Networks

ANNs are designed to learn from examples and from experience. The networks “learn” to perform better with more training and exhaustive testing. Just like a child must learn to walk before he can run, a neural network must be guided in its transition from a raw piece of computer code to a fully trained network capable of performing classification and prediction tasks in its area of

"expertise". There are three different learning paradigms that can be used to train a neural network. Supervised and unsupervised learning are the most common, with hybrid approaches between the two becoming increasingly common as well. Competitive learning may be considered to be the third paradigm. Artificial Neural networks are considered to be "machine learning algorithms". This is due to the fact that during training, the connection weights are transformed to influence better solutions to the given problem. Similar to the neuronal connections in our brains, neurons in an ANN are connected to each other through weighted connections, i.e. the strength of an inter-neuronal connection is defined by its weight. The ANN learns by adjusting these weight values in order to learn from a particular set of inputs. The higher the value of the weight, the stronger is the inter-neuronal connection and thus this connection has a higher probability of being excited.

3.2.2.1 Supervised Learning

Supervised learning is a technique by which a neural network learns a function from a set of training data. This set of training data consists of a set of inputs and a set of corresponding or desired outputs. The neural network is essentially being trained to learn a concept in the presence of a "supervisor". The task of the supervised neural network is to predict the value of the learned function for any valid input, subsequent to having seen a number of training examples (i.e. pairs of input and desired output). Supervised learning is achieved by incorporating a "teacher" into the learning algorithm or a "critic". The difference between the two is that a "critic" tells the neural network whether the output of its learning algorithm is right or wrong, while a "teacher" tells the neural network

what the correct answer or the target output should be. Learning with a critic takes longer because the network isn't told straight away what the correct output is (Lippmann, 1989). The network updates its weights so that it maximises the number of inputs on which it is correct, when learning with a critic, and thus learning with a critic takes longer than learning with a teacher. An illustration of how a neural network learns using the Supervised Learning paradigm is shown in Fig. 14.

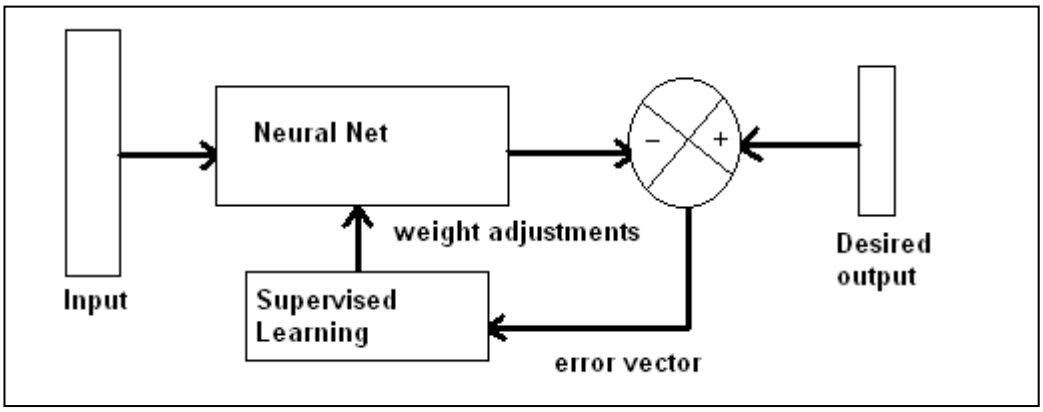


Figure 14: Supervised Learning Paradigm

The figure above also shows the learning rule for the "Error-Back propagation" type of neural network model where the error or the difference between the actual output and desired output is propagated back to the neural network and the networks adjusts its inter-connection weights so as to obtain the actual output as close to the desired output as possible.

3.2.2.2 Unsupervised Learning

An unsupervised neural network learns without a teacher or a critic. Here, the network does not receive any information or feedback from a supervisor; instead it relies on an internal criterion to guide its learning outcomes. This criterion that drives the learning procedure in essence states that the input topology and formatting must be mapped onto the output vectors or the learning outcomes. The objective is to create an output representation in which similar inputs stimulate output units that are close to one another; i.e. to form a topological map of the input data and representations. Thus, the relationships between the input parameters are preserved and mapped onto the output parameters. Figure 15 illustrates this principle. The objective is to create an output representation in which similar inputs trigger output units that are close to one another. The network is shown a series of shapes and colours. The network gradually changes the weights so that similar shapes are mapped to neighbouring units. From an initial random assignment, the red and black rectangles end up in the bottom left hand corner – because they are similar. The triangles and circles end up at the opposite extremes of the map. Unsupervised learning may also be referred to as self-organisation, in the sense that a neural network that learns without a supervisor does so by self-organising data presented to it and identifies their emergent collective properties (Fritzke, 1997).

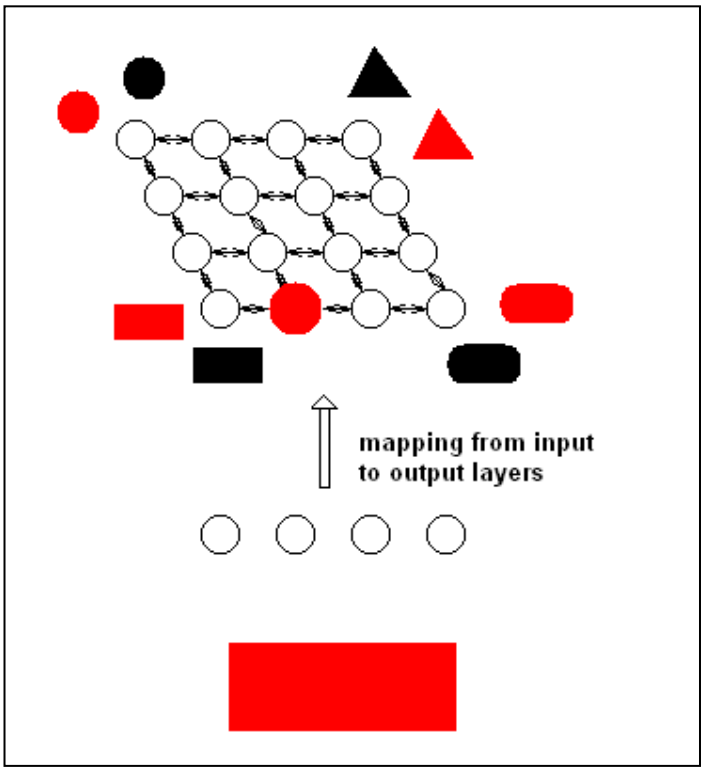


Figure 15: Unsupervised Learning. Input topology is preserved.

In the supervised learning paradigm, the network learns or trains offline while in the unsupervised learning paradigm the network learns or trains itself online. This is because in supervised learning the aspect of learning consists of a distinct or separate phase during which the network is trained followed by an operation phase. Unsupervised learning consists of learning and operating at the same time and thus may be considered to be online learning (Kangas, 1990).

3.2.3 The Self Organising Feature Map

The Self-Organising Map (SOM) or the Self-Organising Feature Map (SOFM) has the particular property of efficiently creating spatially organised "internal representations" of the various features in the input signals and their concepts.

The ability of the SOFM is unique amongst all the architectures and algorithms suggested for ANNs (Kohonen, 1990). Self-organising maps are a data visualisation technique invented by Professor Teuvo Kohonen which reduces the dimensionality of data through the use of self-organising neural networks. The problem that data visualisation attempts to solve is that humans simply cannot visualise high dimensional data as is and thus techniques are created to help us understand this high dimensional data (J. Ashar & G. Littlefair, 2008). Learning in a SOFM is achieved using the unsupervised learning paradigm. The learning outcomes emerge without the need of an external "teacher" providing the desired response to the network. In this type of network, neighbouring neuronal cells compete with each other for the input, execute their actions by means of joint lateral communications, and then develop adaptively into specific detectors of different signal patterns. This category of learning is called competitive learning. It is self-organising and unsupervised learning. The SOFM is a sheet-like artificial neural network, the cells of which become specifically tuned to various input signal patterns or classes of patterns through an unsupervised learning process (Kohonen, 1990). The basic concept underlying competitive learning is — for a given set of observations and assuming a set of variable reference vectors (initialised in a random sequence), for each time interval if the observables can somehow be simultaneously compared with each of the reference vectors from the initialised set of reference vectors then the best matching unit is to be updated so that it matches even more closely to the current input vector. If this comparison between the reference vector and input unit is done by calculating a distance measure then this distance must be decreased and all other reference vectors must be left intact. Thus, different reference vectors tend to be "tuned in" to the diverse domains of the given

input. Eventually, only one cell or local group of cells at a time gives the active response to the current input. An illustration of this principle is shown in Fig. 16.

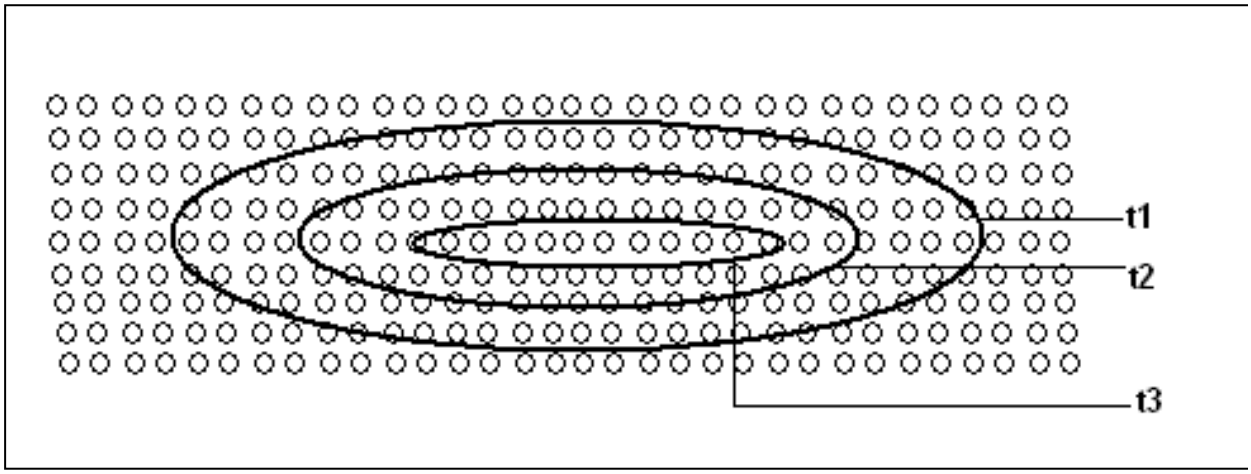


Figure 16: Monotonically decreasing neighbourhoods with time (t1<t2<t3).

This self-organisation ability can substantially reduce the programming burden which eventually brings down the overall cost of deploying a system. Building an automated system for a particular environment or a specific purpose involves high costs in terms of hours spent and resources needed for programming such a system. Traditionally, programming accounts for about one third of the total cost of a system but using the self-organisation ability, this cost is greatly reduced. Moreover, unsupervised models are often fast and their learning speeds, especially when using computational shortcuts, can be augmented to orders of magnitude greater than that of numerous other neural networks. Thus much larger maps than those used so far are quite realistic (Kohonen, 1990). Self-organising feature maps (SOFMs), or just Self Organising Maps (SOMs) are important unsupervised Artificial Neural Network models that have shown great potential in application fields such as speech recognition applications and various pattern recognition tasks involving very noisy signals (J. Ashar & G Littlefair, 2008). Commonly, the SOFM is used to learn the topology of sensory

inputs by clustering the data and is used in control basically as a classifier. The final sensory map can then be used to classify new incoming data. It is important to note that when using supervised models, the error signals are available directly at the output of the network and are explicitly used during network learning and training. In the unsupervised case, the error signals are not computed directly, rather through the use of the definitions in the network's learning rule. For this reason, when unsupervised neural models are used in modelling and control, they are usually referred to as self-supervised models (Kohonen, 1999). This type of learning is controlled by knowledge of the external world provided by sensors and the consequences of actions performed by the network. These networks have also provided acumen into how mammalian brains are organised (de Barreto, 2003). The self organising ability may be depicted as shown in Fig. 17.

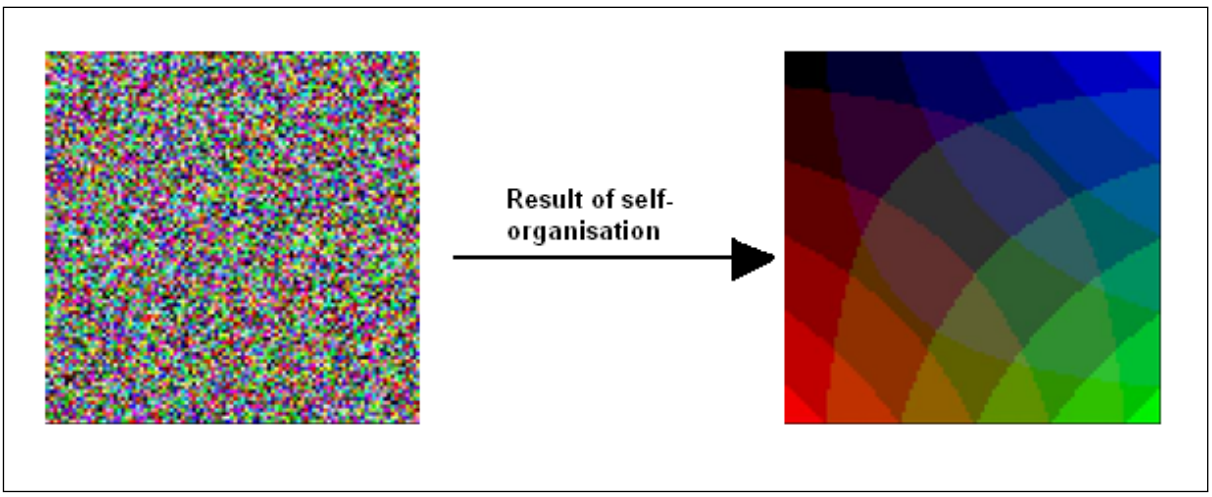


Figure 17: Self-Organisation

Over the past decades, the field of Artificial Intelligence has made great progress toward transcribing human reasoning into digital data. Figurative approaches are based on the hypothesis of symbolic depiction — the idea that

perception and cognitive processes can be modelled as acquiring, influencing, co-relating and adapting to the symbolic representations. Perhaps the best way to move forward is to shift the focus from modifying system behaviour to the processes of cognition that source the performance of the ANNs (Kohonen). Most works have concentrated on robotic systems that are solely sensory in nature. Recently, several studies have proposed the Self Organising Feature Map for the difficult tasks of non-linear modelling. The SOFM can extract features of input data based on incremental learning. The fundamental result in self-organisation is that if the input signals have a distinct probability density function, then the weight vectors of the cells try to match it; however complex its form.

The SOFM is a neural network that closely resembles how the brain organises memory into neuronal connections. Emulating the way in which human brains decode data from various sources (senses) holds tremendous value. Data fusion is the key process for accurate environmental perception. Data captured to characterise the condition of a complex piece of equipment should be as complete as the information we use for our cognitive purposes – i.e. contain primary and secondary data which is used to arrive at a consensus of opinion (Littlefair, 2007).

3.2.4 Linking the brain and the computer- chaos and synesthesia

It is a generalisation to say that it is impossible to artificially imitate the human brain due to the limitations of current computational resources. In actuality, the key concern for failing to properly emulate the human way of information processing is the existence of many un-interpreted details of the brain structure

and behaviour (SG Wysoski). Our brain is chaotic. Chaos has been found in how we process external senses, and may be key to memory. It has been implicated in at least one theory of the evolution of vocabulary as well as synesthesia (FIRE). Synesthesia is a neurological phenomenon in which the stimulation of one sensory organ leads to automatic or involuntary sensations in another sensory organ. In its most common form known as grapheme → colour synesthesia, letters and/or numbers are perceived as inherently coloured and having personalities. Synesthesia has been being diagnosed for almost three centuries, but the medical profession keeps forgetting about the condition. The word *Synesthesia* means "joined sensation" and shares a root with *anesthesia* which means "no sensation". Synesthesia is not an abnormality; in fact it is a normal brain development process that is intuitively presented to the consciousness in a minority of individuals. The condition symbolises a rare ability to hear colours, taste shapes, or experiences of other equally astounding sensory amalgamations whose nature seems too complex for most of us to envisage. Synesthetes are normal in the conventional sense of the term and they appear to be bright and intelligent. Standard neurological medical exams are also normal. Synesthetic associations are usually unidirectional, meaning that a particular synesthete sight may induce touch, but touch would not induce visual sensations (Ramachandran, 2001). Simulating synesthetic type of neurological behaviour in Artificial Neural Networks (the core of Artificially Intelligent Systems) will help shed light on their functioning and classification capabilities. This in turn may also deepen our understanding as to why these systems are unstable when applied in real world environments. The process of disassembly and reassembly takes on an entirely new meaning. The eventual goal is to create efficient, robust systems with extended autonomous control

over processes that are being employed – essentially creating the “factory of the future” (J. Ashar & G. Littlefair, 2008).

Most models of the brain do not include chaos. The models that do include the concept of organised chaos in their design, don't seem to be convincingly biological (Walter). In an attempt to discover what instigates the reasoning of human minds, one of the most testing aspects for scientific analysis is that the current technologies cannot keep track and measure all the signals used for inter-neural communication, even in an infinitesimal portion of the brain (Ramachandran, 2001). If this was possible, it would enable us to accurately appreciate the emergence of intelligence from a collection of neurons. In an attempt to overcome this limitation, a common practice is to complement the study with the development of intelligent computational models based on experimental data and to study their properties by theoretical and simulation means (Rehorn, 2005). The SOFM is proposed as a feasible elective to more traditional neural network architectures. Its analytical portrayal has already been developed further in the technical than in the biological direction. The learning results accomplished appear to be as expected; at least indicating that the adaptive processes at work in the map may be analogous to those encountered in the brain.

Chapter 4 Experimental Procedure

The work presented in the previous chapters has shed light on certain elements which were believed to offer the best capability for developing a robust, accurate, flexible and efficient tool wear monitoring system. A series of experimental trials and tests were designed and executed with the intention of determining the process by which an accurate description of the tool wear state could be provided. The approach was divided into the following stages:

- Collection of comprehensive information relating to multi-component tool force (in all the three directions) and torque.
- Pre-processing of raw data to reduce dimensionality and integrate vectors from various sensors while maintaining completeness in information.
- Application of state-of-the-art Self-Organising Feature Map type of neural network to coalesce integrated multisensory data, providing a detailed description of the tool wears state.

These stages are discussed in detail in the following sections.

4.1 Drilling Trials

To give maximum credibility to this research investigation, artificially created wear and wear produced by using unrealistic operating conditions was not perceived to be an appropriate course of action to follow. Accordingly, only wear produced by machining components with parameters prescribed by industrial

practices were employed. This methodology is the only way to ensure that the captured data is truly representative of anything which is likely to exist. Data from tools where flank wear has been produced by the process of drilling is only representative of drill flank wear and can be considered to be nothing else. The experimental drilling trials were designed with these factors to be of paramount importance.

Drilling tests were performed using a solid carbide drill on a duplex stainless steel workpiece. These steels offer good resistance to local and uniform corrosion. The duplex microstructure contributes to their high strength and high resistance to stress corrosion cracking. These properties make the steel best suited to be employed for experiments to measure drill wear ("Duplex Steel Information," November 2005.; Smith, 2007). The basic machining criteria for the experiments are detailed below and a schematic representation of the process is shown in Fig. 18. Figures 19 and 20 show the actual experimental setup.

Machining Variable	Machining Variable condition
Cutting Speed(s)	35 m/min & 45 m/min
Feed rate(s)	0.125 mm/rev & 0.175 mm/rev
Machine Tool	Drills from Iscar Pacific. Family:SCD-AP3 (Dia12.5); Solid carbide drills without coolant holes. Drilling depth 3xD
Workpiece material	Duplex Stainless Steel SAF 2205.

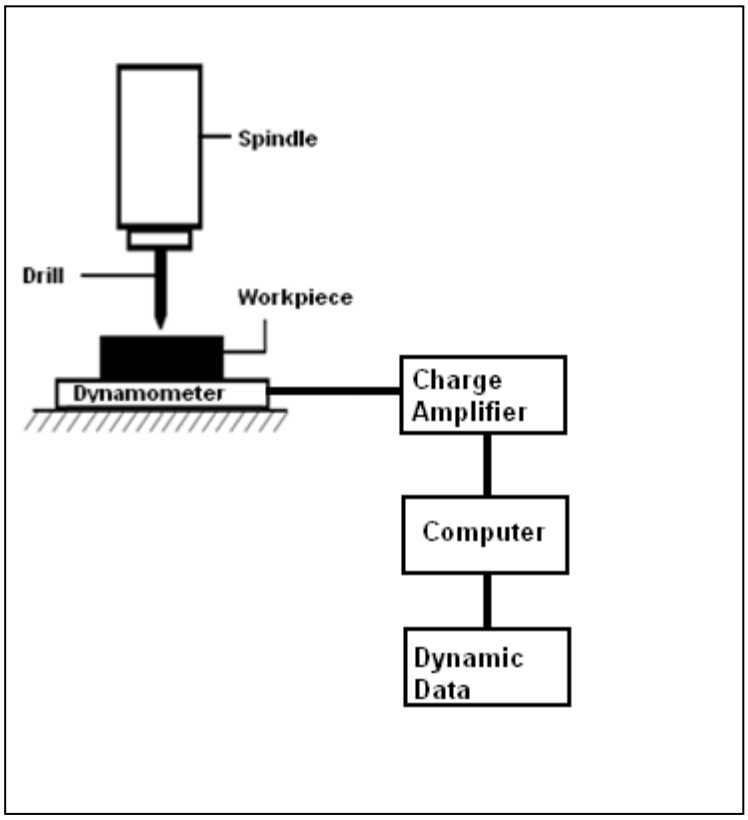


Figure 18: Schematic of experimental setup

Figure 19: Experimental setup showing (clockwise from left) Charge Amplifier, Computer, Drilling machine with Duplex Steel workpiece and drill attached.

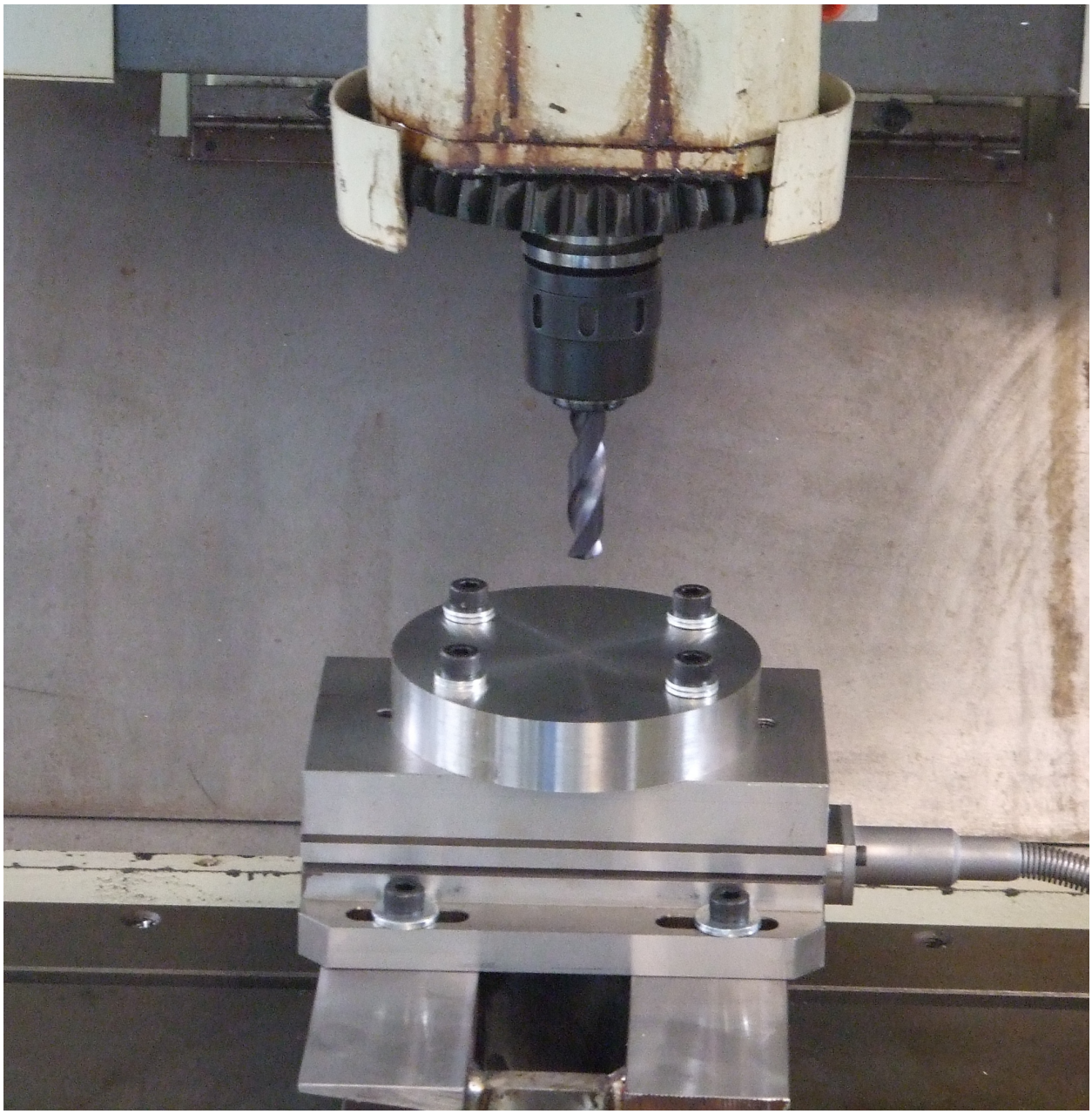


Figure 20: Close-up of the drill and workpiece

4.2 Force Measurement

At the outset of the experimental stage, the simultaneous measurement of the three perpendicular forces was identified as a priority. By analysing the forces along the three mutually perpendicular axes, latter analysis of the most sensitive components would be established. This was necessary since no clear consensus of opinion was provided in the literature and impulsive, unfounded selection was not considered to be a suitable avenue worth pursuing. The Kistler three component quartz dynamometer (type 9257B) was selected for the force measurements (Fig. 21).

Figure 21: Kistler three component dynamometer, 5 kN

The dynamometer consists of four 3-component force sensors fitted under high pre-load between a base plate and a top plate. Each sensor contains three

pairs of quartz plates: one sensitive to pressure in the Z direction and the other two responding to shear in the X and Y directions respectively. As a result, the dynamometer is able to detect the smallest changes in large forces. A major feature of this design is that the force components are measured practically without any displacement. Also, the dynamometer is rust-proof and protected against penetration of cutting fluids. A special thermal insulation coating is integrated in the top plate which renders the device insensitive to temperature influences. A full specification of the dynamometer and the calibration charts for the actual unit used are provided in Appendix A. In addition to the dynamometer, two other items of associated Kistler equipment were utilised in the measurement of forces. These were: armoured connecting cable type 1687B5; and a three channel charge amplifier type 5070A (Fig. 22). A full specification of the charge amplifier is provided in Appendix A.

Figure 22: Kistler four channel charge amplifier type 5070A

4.3 Data Capture and pre-processing

The data capture was conducted in real time using the DynoWare software along with the dynamometer and charge amplifier. Figures 23 and 24 show typical examples of the captured real-time data.

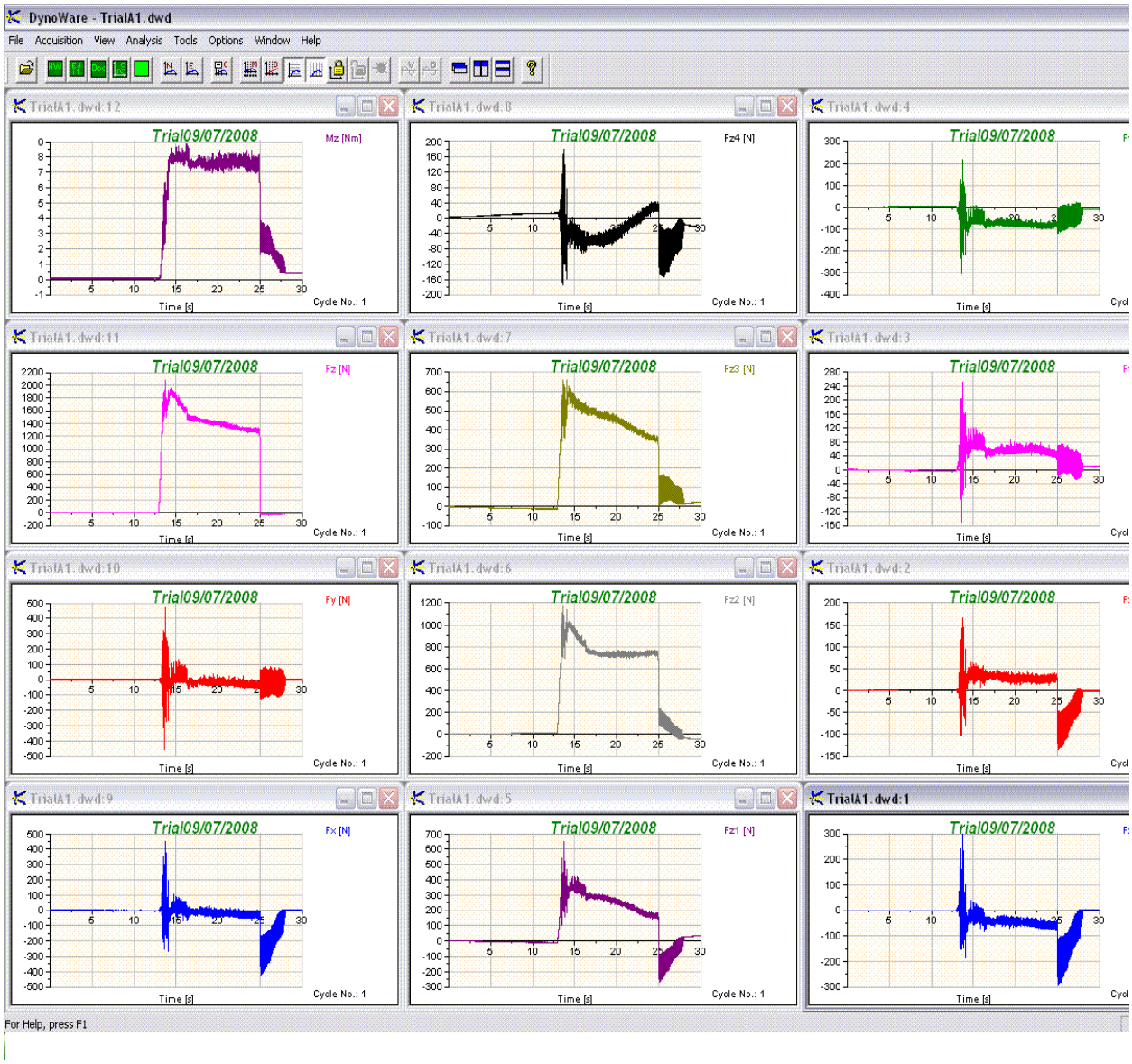


Figure 23: Example of captured real-time data

The figure above shows the forces captured along the three axes (x, y and z) along with the calculated forces from the 4 channels present in the dynamometer. The figure below shows the magnified version of the forces in the x direction, i.e. Fx

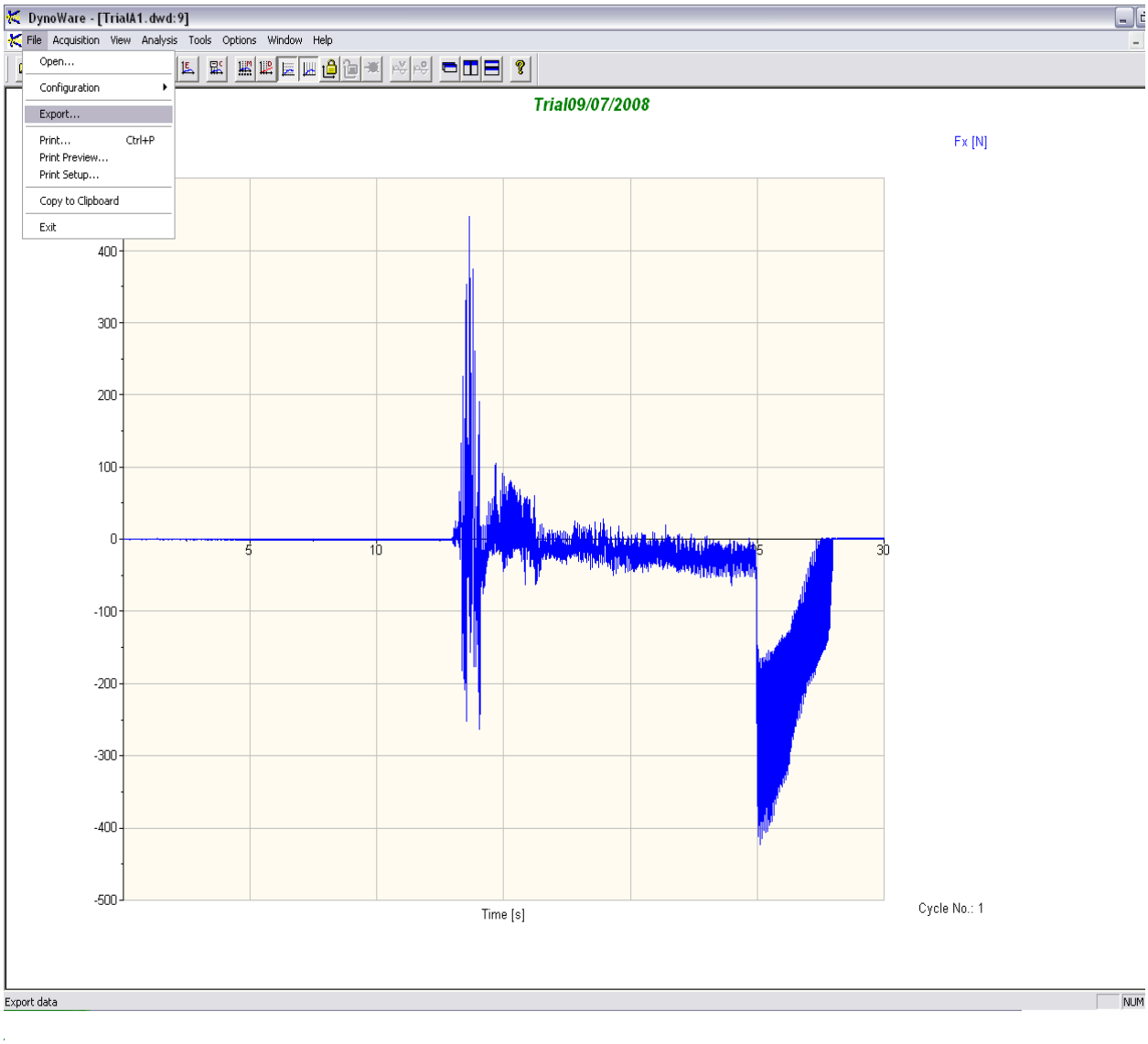


Figure 24: Magnified (Force along the x-axis)

This data is then exported as digital data into an excel worksheet. The specifications of all the forces are provided in Appendix B.

The digital data was analysed using the DaDiSP/32 software. DaDiSP is a powerful generic tool for data display and analysis. Once the data is captured, DaDiSP allows displaying the data for subsequent manipulation and analysis. Figure 25 shows how the data is manipulated using DaDiSP.

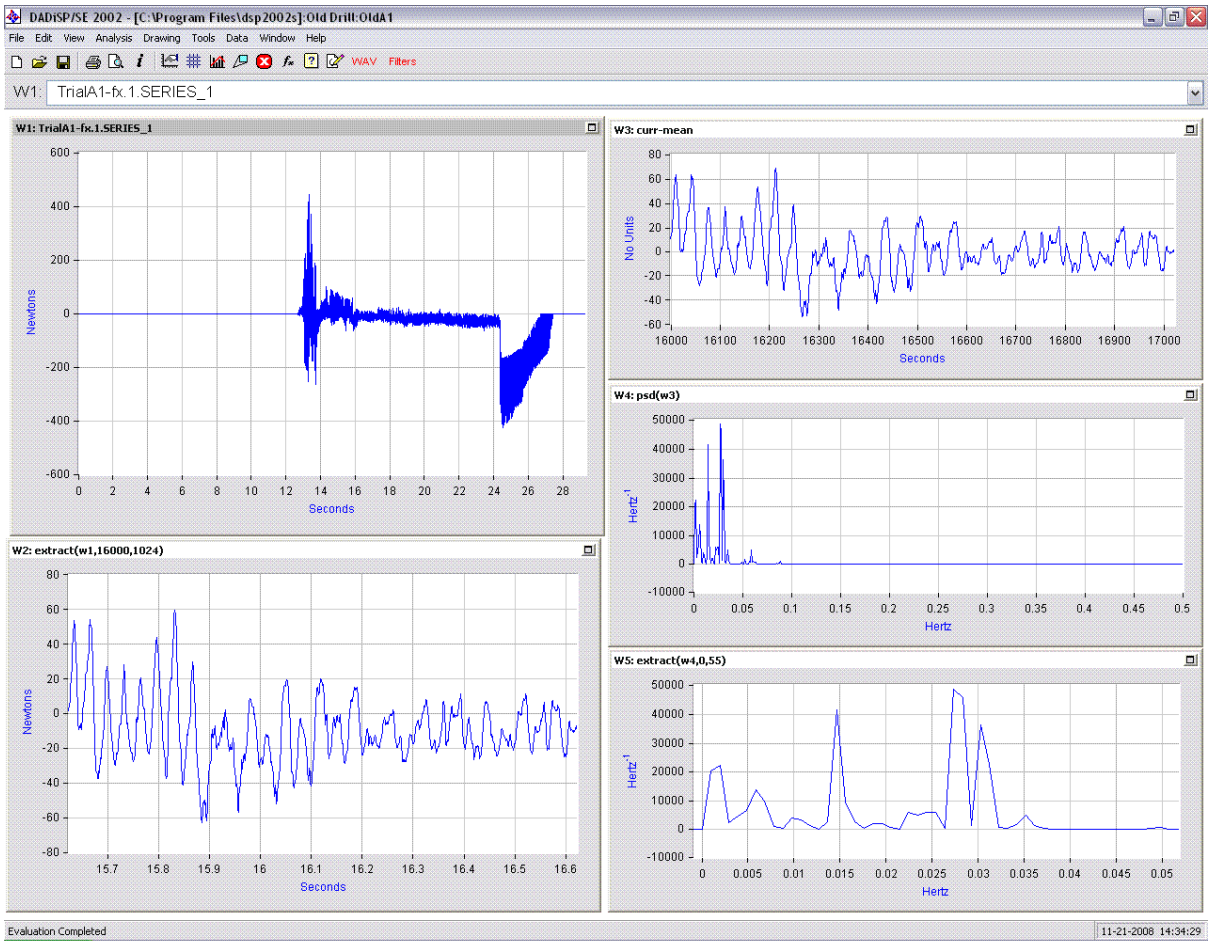


Figure 25: Using DaDiSP for data display, manipulation and analysis

Figure 25 shows the data being read and data evaluation or analysis being performed. The exported digital data is first read as a series in DaDiSP. This is shown in Fig. 25 in the first window from the top left. This data is then broken up into several parts of dynamic data which are extracted as shown in the bottom-left window in the figure. These extracted parts of the data have an overlap (of 48 sampling points) with each other to maintain the continuity and completeness in the output results. This extracted data is then analysed by calculating the Power Spectral Density (PSD) of the signal. This is done using the psd function. The dynamic part of the PSD is shown in the bottom-right window in Fig. 25. The power spectral density of the data shows the strength of the variations (energy) as a function of frequency. In other words, it shows at

which frequencies variations are strong and at which frequencies variations are weak. PSD is a very useful tool to identify oscillatory signals in time series data. It also gives the amplitudes of the data set. PSD analysis is especially useful to detect unwanted vibrations that stem from machining operations. The PSD gives an overall picture of the frequency of vibrations and thus aids in the identification of frequencies at which tool wear is noticeable.

To further characterise the data, it was considered to calculate the Kurtosis of the signals. Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. That is, data sets with high kurtosis tend to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. A uniform distribution would be the extreme case. Significant kurtosis values clearly indicate that data are not normal. The kurtosis distributions that were calculated are shown in Fig. 26. It is clear from the calculated kurtosis distributions that the distribution deviations are relatively high and some of the calculated values are significant. Thus, the kurtosis distributions were not considered to be a fitting indicator of the data distribution. The entire range of the calculated Kurtosis values is given in Appendix B.

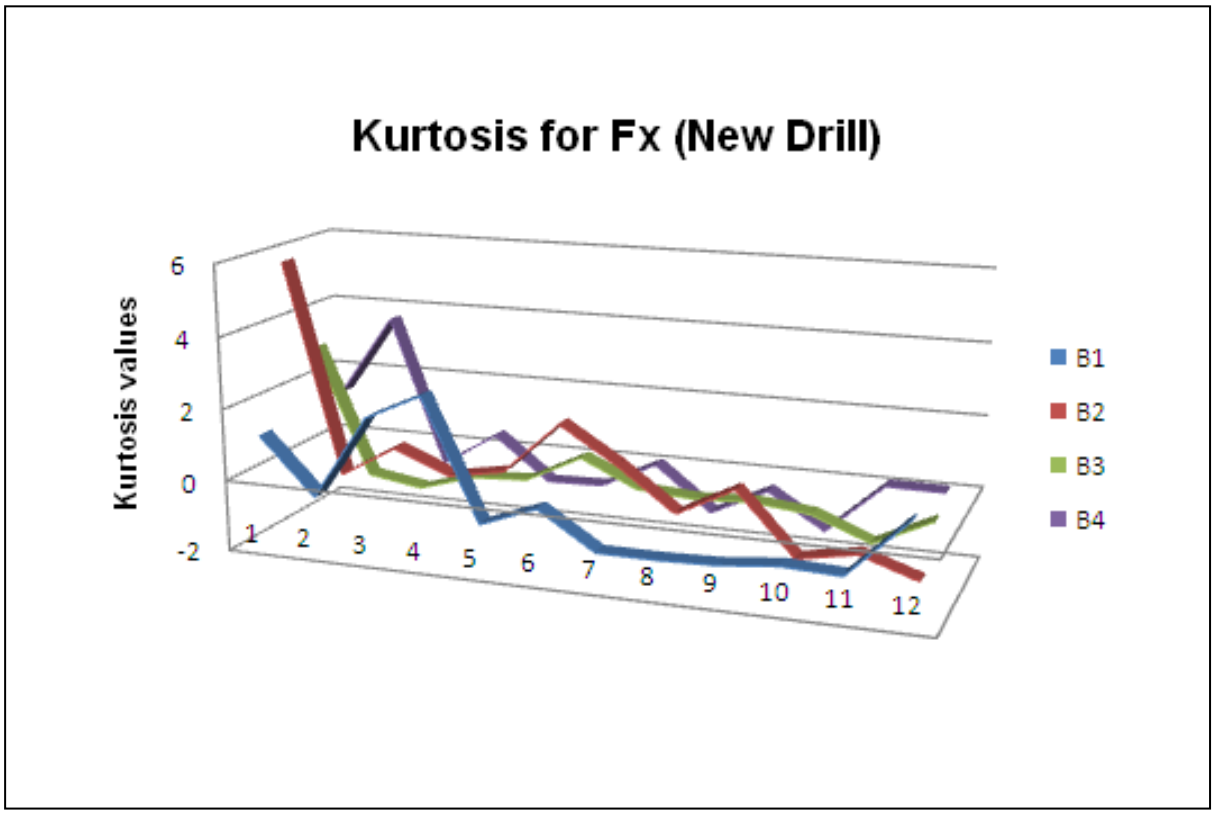


Figure 26: Kurtosis of forces in the x direction for trials with the new drill

4.4 Artificial Neural Network Processing

The program utilised for predicting tool wear during the process of drilling was a self-organising feature map. This program uses the generalised SOM algorithm described in the previous chapters. The operation of the program has two main modules: learning and classification. The learning in this program is done using the unsupervised paradigm of learning and the network learns to classify the inputs wear or no wear. During the learning phase, the neural network aims to reduce the error between the input data and the selected neuronal cells that represent the output. This reduction in error ultimately leads to only a few neurons being activated in the output layer for the corresponding input signals. Thus the network learns to classify or re-organises itself according to the

variations in the input domain. Figure 27 shows the graph of the error versus number of iterations during the learning phase.

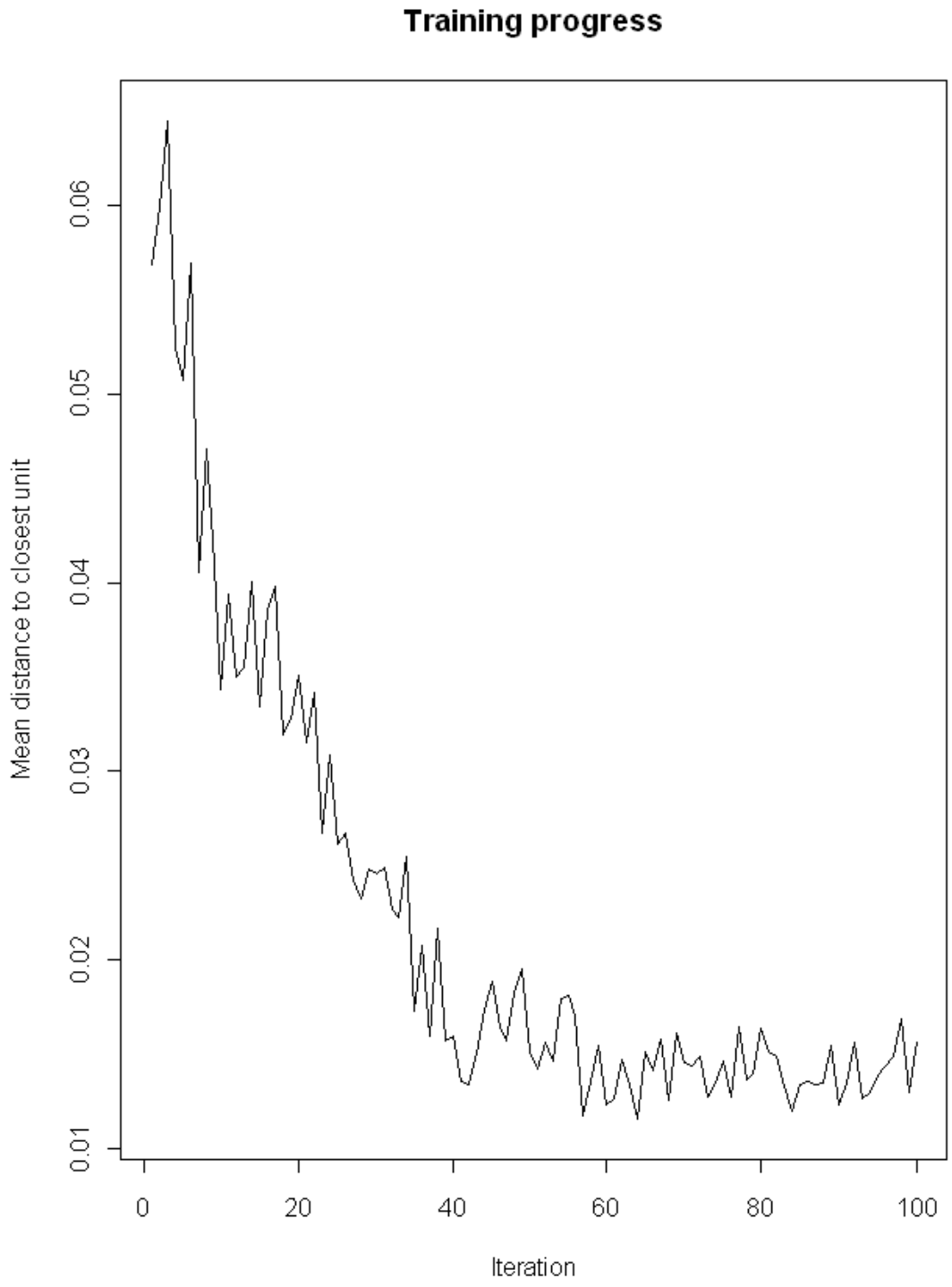


Figure 27: Reduction in classification errors during SOM training

Figure 27 shows the progress after 100 iterations of the program and as visible from the graph, the error values are still not at a minimum after 100 runs. Thus,

the learning phase of the network involved 100,000 iterations of the program. Once the error values were stabilised, the learning phase ended and the network was ready to classify. During classification, the network loads previously learnt patterns (it does so by adjusting the inter-neuronal connection weights) and calculates the output for a given set of input data. The output calculation or classification depends on the neurons which are excited in the output layer. The decision to excite which neurons in the output layer are decided by the SOM based on previous knowledge and learnt classification. This phenomenon is illustrated in Fig. 28.

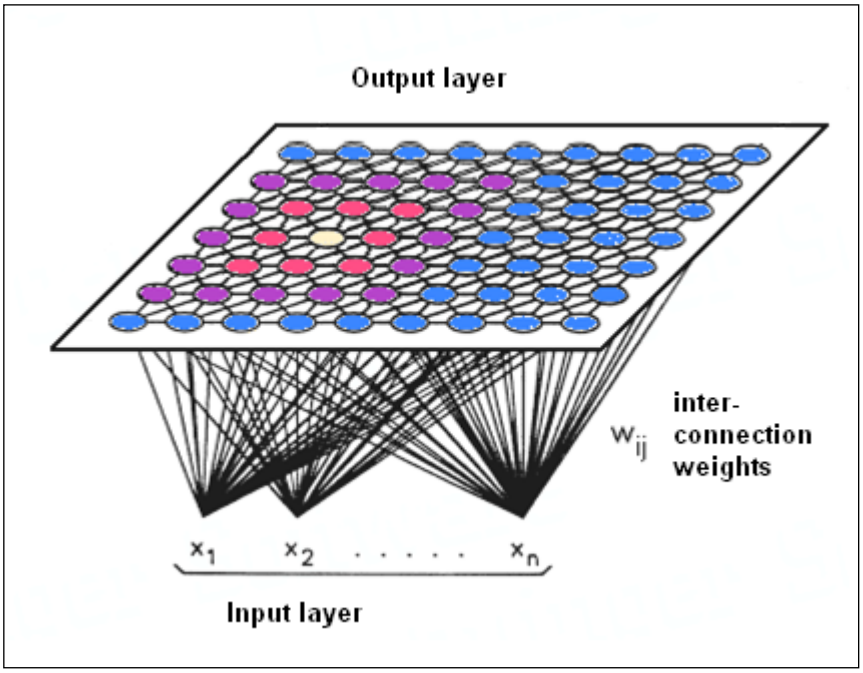


Figure 28: Generating an output by activating a neuron(s) in the output layer for corresponding inputs. Data dimensionality in a SOM - http://www.sis.pitt.edu/~ssyn/som/som.html

Chapter 5 Results and Observations

The results presented in this chapter consist of multisensory input values to the ANN for both learning and subsequent classification. The outputs obtained from the SOM have also been presented which indicate the classification performance of the ANN under varying conditions. Graphical plots have been used to visualise the data sets and the display of data is made as clear and practicable as possible. Neural network coding and simulation was done using R. R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical and graphical techniques, and is highly extensible (Team, 2008). The program written for the SOFM is based on the function provided in the “kohonen” package of R (R. Wehrens, 2007). Although the basic calculation subroutine is little changed, the data input, handling, execution, storage and output formatting is all original.

The learning and training progress with the SOM is shown in Figures 29 and 30 and the classification and prediction results appear in Figures 31 to 33.

5.1 Self-Organising Map training progress observations

Figure 29 shows a plot of the SOM after 100 iterations during the training phase. The figure shows the neurons within the SOM output layer and the distribution of data within these neurons. Depending on the input domain signals, the best matching unit from the output layer is chosen to represent the output. The output of the network is classified into acceptable or unacceptable tool wear. The neuron that is stimulated in the output layer will be the neuron

whose data distribution matches the closest to the input data distribution. It can be seen from the figure that the neighbourhoods or the sampling regions of the neurons do not overlap with each other and thus information sharing between the neurons is nil or very low.

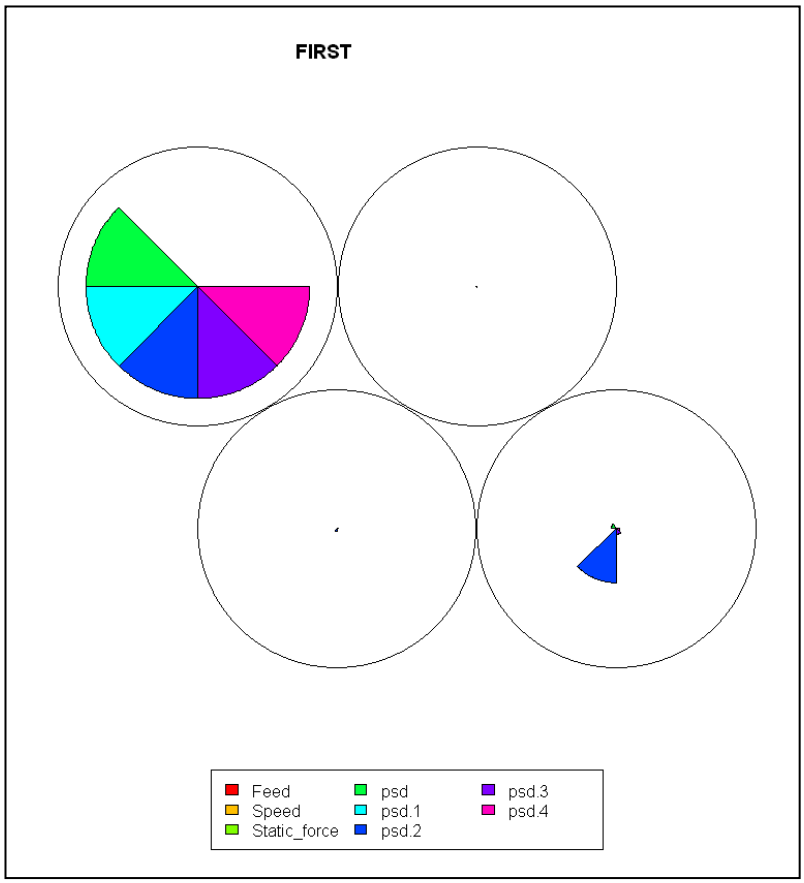


Figure 29: Plot of the first SOM after 100 iterations during the training phase

Figure 30 shows the neuronal plot of the SOM after 20,000 iterations during the training phase. The difference between Figures 29 & 30 is apparent. The neurons in Fig. 30 are more in number; the data distribution inside the neurons is gradually starting to form clusters or groups of neurons with analogous data distributions. This is the beginning of the classification stage. Also, the neuronal neighbourhoods are coming closer to each other ever so slightly indicating that

the data distribution within the neurons is inclining towards being fuzzy or representative of the “real world” data.

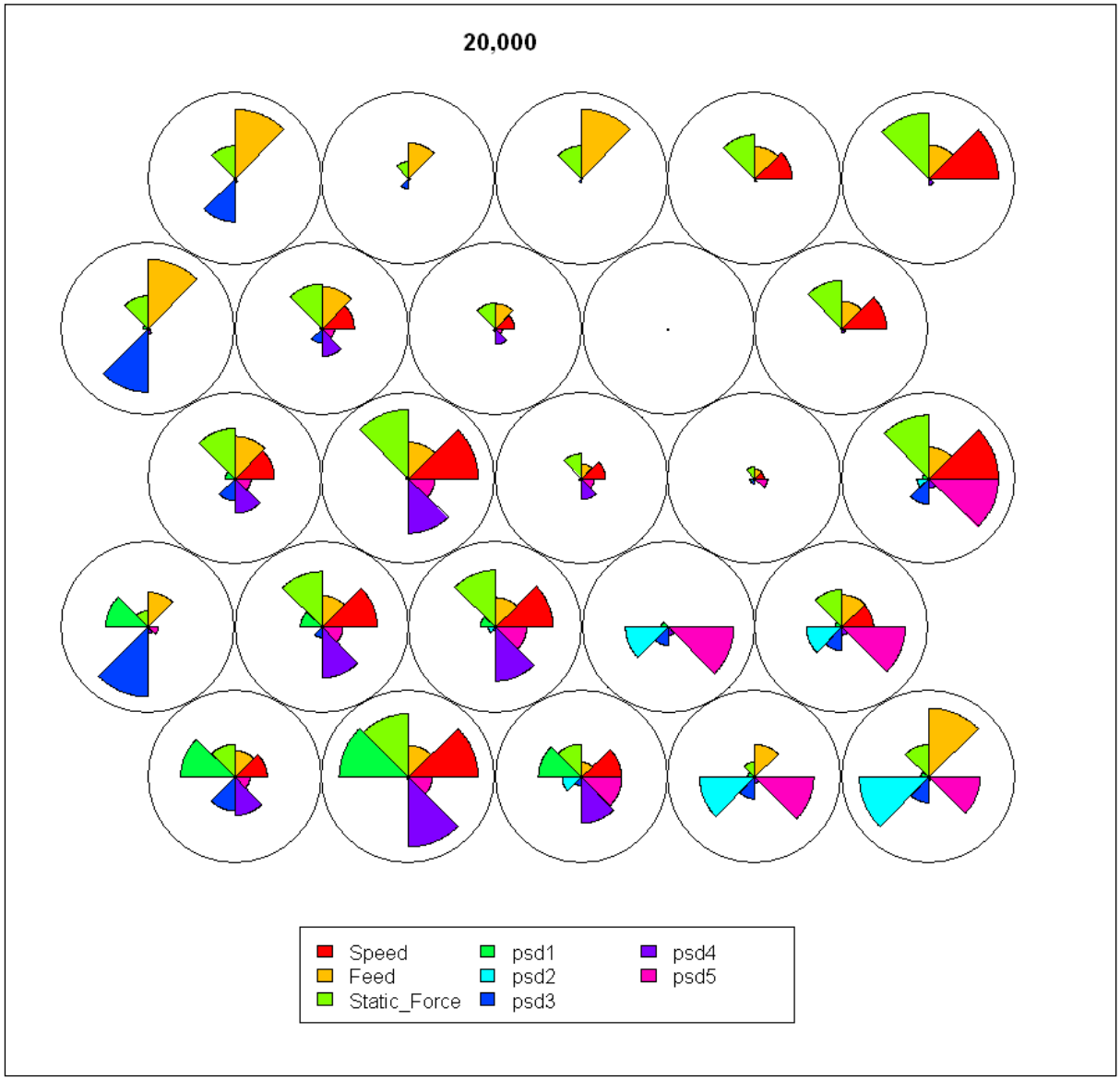


Figure 30: Neuronal plot of the SOM after 20,000 iterations during the training phase

5.2 Results of SOM training – Classification

Figure 31 shows the neuronal plot of the SOM that has undergone the learning process and has re-organised itself to match the patterns in the input domain.

This SOM is the result of 100,000 iterations of the program. A full listing of the program can be found in Appendix C.

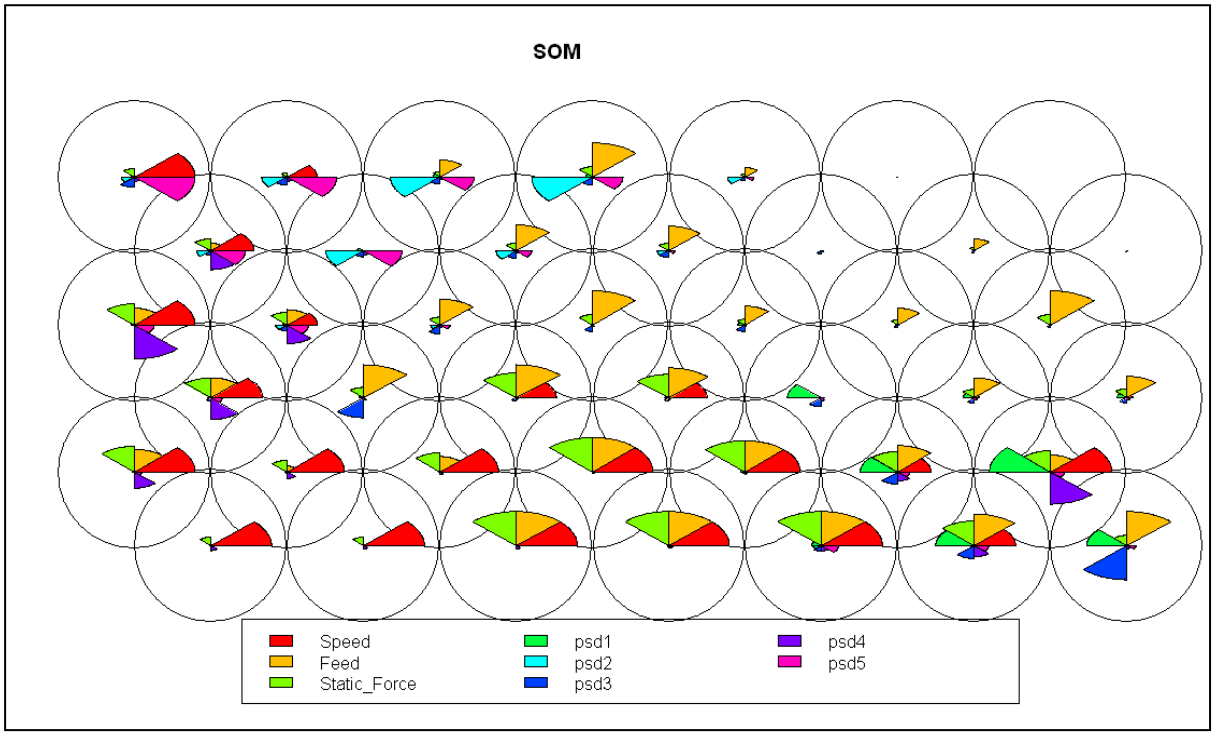


Figure 31: Neuronal plot of the SOM after 100,000 runs

In Fig. 31, we can see that the SOM has now re-organised itself to match the input patterns as closely as possible. The neurons with similar data distributions are bunched together and there is significant overlap between the neuronal neighbourhoods. Thus, the output layer distribution is fuzzy and corresponds to the real-world data distributions in the input domain. Due to this overlap between the neuronal neighbourhoods in the output domain, for a given set of input data distributions, a small set of neurons in the output layer will be excited or stimulated to give the output. This classification and re-organisation of the neurons within the neural network is done autonomously by the ANN during the learning phase and involves no human supervision or input. Thus, the

independent nature of the SOM serves us well for predicting the outcomes of non-linear processes in stochastic environments.

5.3 SOM prediction

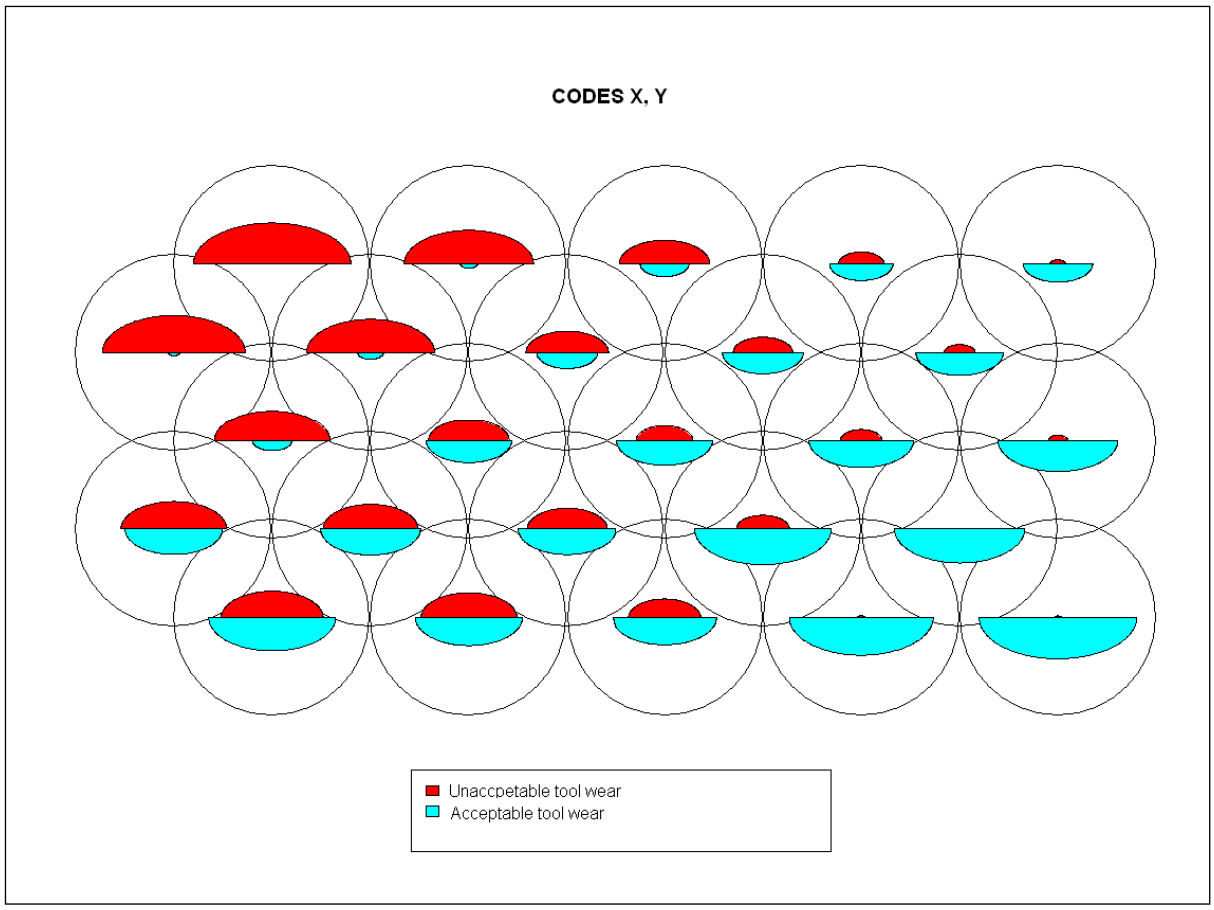


Figure 32: Classification results showing neuronal areas in the SOM corresponding to tool wear or no tool wear

Figure 32 shows the neuronal plot of the neural network during the prediction phase. The bottom right corner of the plot represents the neurons which correspond to the data distributions showing acceptable tool wear in the input domain. If the multi-sensory input to the SOM were indicative of acceptable levels of tool wear, then the output of the SOM would be presented by the excitatory response generated by these neurons. The neurons in the top-left corner of the map show the neurons that are representative of unacceptable

levels of tool wear. Thus, the neural network has "learned" to distinguish between two different levels of drill wear.

Figure 33 shows the prediction performed by the SOM in 100,000 iterations of the program.

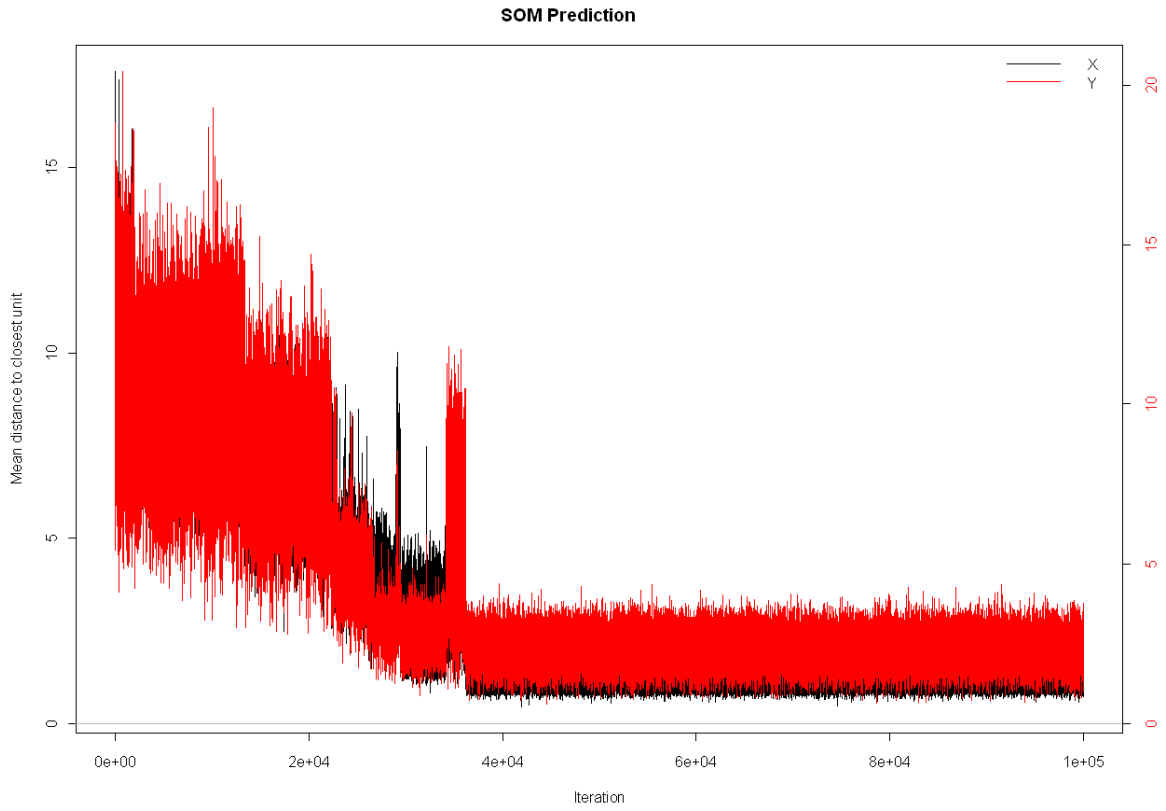


Figure 33: SOM prediction; X (Black) is the input while Y (Red) is the output.

The input domain is dynamic for the first 40,000 runs of the program and then stabilises. From the graph it is plain that the output patterns closely follow the input patterns and therefore the SOM prediction is accurate. The total number of miscalculations can be considered to be low in relation to the number of runs of the program that were made. This phenomenon would tend to indicate that the ANN has successfully learned the problem of tool wear detection and has reasonably good generalisation abilities. The odd misclassification probably suggests that there were local variations in the data caused due to the input

domain being dependent on the machine variables. The misclassifications were also limited to adjacent data sets in the input domain displaying a general trend of classifying to a lower level of tool wear.

Chapter 6 Conclusions and Discussion

The results presented in the previous chapter demonstrate the capabilities of the developed system by the evaluation of a particularly demanding set of data obtained from different machining configurations and conditions, albeit the same process. Accurate classification of tool wear provides evidence that the artificial neural network was able to identify similarities between the data sets that were given to it as input for learning and the test data sets. Moreover, it indicates that the data was highly process dependant with little (if any) machine dependencies. The Self-Organising Feature Map appears, therefore, to be an effective and efficient prediction model with adequate knowledge retained by reorganisation of the neurons constituting the map.

The promising performance presented here, is not merely a reflection of the ANN capabilities however. The pre-processing and integration techniques are the information suppliers on the problem, and as such, these have also demonstrated their adeptness allowing for judgemental decision making. Hence, successful classification is an indicator of the “system” performance.

Tool wear identification and monitoring is a complex phenomenon. Accurate modelling of the problem requires a highly evolved and comprehensive solution. The work done so far has mainly focussed on the use of neural networks which learn using the supervised learning paradigm. These networks perform well under known conditions, but even a minor deviation from their predefined parameters can cause such networks to fail. The principal aim of this research is to construct a robust and efficient system for tool wear monitoring in drilling operations.

The Self-Organising Feature Map is a neural network that closely resembles how the brain functions. Mirroring the way in which human brains decode data from various sources (senses) possesses tremendous merit. Data captured to characterise the condition of a complex piece of equipment should contain primary and secondary data which is used to arrive at a consensus of opinion.

Learning in this network is unsupervised, thus making it independent of human errors caused during the training phase. Furthermore, the network is able to adapt to changing environments and conditions. This flexibility in adaption integrates well with the stochastic nature of industrial environments.

The research presented demonstrates the type of system which can successfully be employed to monitor machining operations — an artificially intelligent program, providing information on a variety of distribution parameters, which can be successfully applied to a personal computer. Sufficiently long data samples, which ensure accuracy, need not result in prohibitively large computation times thus making the program's application to online tool condition monitoring a real possibility. The true robustness of the system is to be established by the application of the system in other industrial environments. The classification of tool wear using unsupervised neural networks is regarded as a strategic step forward in the progress towards the creation of a truly unmanned machining environment.

The findings of this research may be summarised below:

- There is a need for reliable and robust online tool condition monitoring systems capable of providing information on tool wear in process time.

- It appears unlikely that tool wear system variables can be accurately predicted using inputs from the human supervisor for the program.

- Artificial Neural Networks are the best adapted for modelling non-linear processes which make them inherently suitable for problems such as tool wear monitoring which itself is a highly non-linear and stochastic process.

- Complex time domain information can be satisfactorily expressed using the power spectral density of the data.

- The unsupervised learning paradigm is proven to be better suited and more robust for the prediction of tool wear as opposed to the supervised learning paradigm used by the majority of research in the field of TCM.

- The same basic system, once trained, is capable of accurately classifying tool wear during the process of drilling.

Chapter 7 Further Work

The research presented in this thesis has led to the identification of a number of areas which are considered worthy of further investigation and development. They may be identified as follows:

- For the developed system to be acceptable in true industrial environments, the system has to remain effective in changing environments with altering machine variables. Development of a universal TCM system is a particularly active research area with the continuous introduction of more advanced technologies.

- Having established that unsupervised learning is an effective way to go when it comes to neural network learning and development, the next evolutionary step would be create an algorithm for ANN learning which involves more human phenomenon in its design and behaviour. A combination of supervised and unsupervised learning would be a path worth pursuing.

- Having established that tool wear prediction is possible, the next step would be to automate the modification of tool offsets. This step is becoming considerably simpler to achieve with greater utilisation of micro-chip based machine controllers.

- The software and hardware elements for a comprehensive TCM system must be devised into a dedicated system for data-capture, pre-processing and prediction. These systems should include the memory capabilities for simultaneous data capture of various dynamic

characteristics and also allow suitably fast classification of tool wear for industry acceptance.

- Embedding the ANN architecture in a TCM system would enable greater levels of un-manned machining operations. The eventual aim would be to create machines that are capable of performing all the functions which are done by a machine operator in contemporary factory settings.

- Drilling is the most common machining operation and monitoring of small diameter drills is particularly crucial in automated factory settings. The work presented here is the adoption of the general techniques and methodologies used in other cutting processes such as milling and tapping. Monitoring individual inserts rather than the whole cutting tool, which at best would supply average tool wear value of the inserts, would be the next step towards the generalisation of TCM systems. The collection of in-cut data is undoubtedly the key element to effective tool wear monitoring and this can be simply achieved by utilising the relationship of the tool to the collected time domain information. The use of relatively short time domain signals in this work is therefore considered to be highly transportable to other machining operations.

References

1. Abu-Mahfouz, I. (2005). Drill flank wear estimation using supervised vector quantization neural networks. *Neural computing & applications, 14*(3), 167-175.

2. Anon. (2001). Practical guidelines for the fabrication of duplex stainless steel. *International Molybdenum Association*.

3. Anon. (2005). *Duplex stainless steel.* Paper presented at the Outokumpu Stainless AB, Avesta Research Centre, Avesta, Sweden.

4. Ashar, J., & Littlefair, G. (2008, October 1-3, 2008). *Drill Wear condition monitoring using Self Organizing Feature Maps.* Paper presented at the 7th International Conference on Industrial Tooling (IT'08) Mississippi State University, Mississippi, USA.

5. Ashar, J., & Littlefair, G. (2008, 29th - 31st October 2008). *Intelligent Drill Wear Condition Monitoring using Self Organising Feature Maps.* Paper presented at the 4th New Zealand Metals Industry Conference - Building Sustainability, SkyCity Convention Centre, Auckland, New Zealand.

6. Ashar, J., & Littlefair, G. (2008, July, 2008). Intelligent Machines. *Engineering News*. http://engineeringnews.co.nz/articles/july08/articles/manufacturing-technology.php

7. Basim Al-Najjar, I. A. (2000). Improving effectiveness of manufacturing systems using total quality maintenance *Integrated Manufacturing Systems, 11*(4), 267 - 276

8. Brophy, B. (2002). AI-based condition monitoring of the drilling process. *Journal of materials processing technology, 124*(3), 305.

9. D. O'Sullivan, a. M. C. (2002). Machinability of austenitic stainless steel SS303. *Materials Processing Technology, 124*, 153-159.

10. de Barreto, G. (2003). Self-Organizing Feature Maps for Modeling and Control of Robotic Manipulators. *Journal of intelligent & robotic systems, 36*(4), 407-450.

11. Dimla, D. (1997). Neural network solutions to the tool condition monitoring problem in metal cutting—A critical review of methods. *International journal of machine tools & manufacture, 37*(9), 1219-1241.

12. Dimla, D. (2000). On-line metal cutting tool condition monitoring. I: force and vibration analyses. *International journal of machine tools & manufacture, 40*(5), 739-768.

13. Dolinsek, S. (2003). Work-hardening in the drilling or austenitic stainless steels. *Materials Processing Technology, 133*, 63-70.

14. Duplex Steel Information. (November 2005.). Avesta, Sweden: 1008EN-GB:4. Centrum Tryck AB.

15. El-Wardany, T. (1996). Tool condition monitoring in drilling using vibration signature analysis. *International journal of machine tools & manufacture, 36*(6).

16. FIRE, K. (2003). LINKING CHAOS IN THE MODEL TO CHAOS IN THE BRAIN.

17. FRANCOGASCA, L. (2006). Sensorless tool failure monitoring system for drilling machines *International journal of machine tools & manufacture, 46*(3-4).

18. Fritzke, B. (1997). Unsupervised ontogenic networks. In R. B. Emile Fiesler (Ed.), *Handbook of Neural Computation* (Vol. 97, pp. 1-16): Institute of Physics Publishing and Oxford University Press. http://citeseer.ist.psu.edu/293655.html

19. G. S. J. Reis, A. M. B., O. (2000). Influence of the microstructure of duplex stainless steels on their failure characteristics during hot deformation. *Materials Research, 3*(2), 31-35.

20. J. Paro, H. H., and V. Kauppinen. (2001). Tool wear and machinability of HIPed P/M and conventional cast duplex stainless steels. *Wear, 249*, 279-284.

21. Jantunen, E. (2002). A summary of methods applied to tool condition monitoring in drilling *International Journal of Machine Tools and Manufacture, 42*(9), 997-1010

22. Kak, S. (2005). Artificial and biological intelligence. *Ubiquity - Association for Computing Machinery, 6*(42), 1.

23. Kangas, J. A. K., T.K. Laaksonen, J.T. . (1990). Variants of self-organizing maps. *Neural Networks, IEEE Transactions on, 1*(1), 93 - 99

24. KIM, J. (2002). Tool wear measuring technique on the machine using CCD and exclusive jig. *Journal of materials processing technology, 130*.

25. Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78*(9), 1464-1480.

26. Kohonen, T. (1999). Analysis of processes and large data sets by a self-organizing method. *Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on, 1*, 27 - 36

27. Kurada, S. (1997). A review of machine vision sensors for tool condition monitoring. *Computers in industry, 34*(1), 55.

28. Lippmann, R. (1989). Pattern classification using neural networks. *IEEE communications magazine, 27*(11), 47.

29. Littlefair, G. (2007, May 2007). *Multisensor Condition Monitoring using the Fusion Characteristics of Artificial Neural Networks.* Paper presented at the Vibration Association of New Zealand Annual conference, Hamilton, New Zealand.

30. Littlefair, G. (2008, October 27 – 30,). *Martensite Formation of Post Heat Treated ADI Due to Strain Induced Phase Transformation.* Paper presented at the 17th IFHTSE Congress, Kobe International Conference Center.

31. Monostori, L. (2003). AI and machine learning techniques for managing complexity, changes and uncertainties in manufacturing. *Engineering applications of artificial intelligence, 16*(4), 277.

32. N. Jia, R. L. P., Y. Wang et al. (2006). Interactions between the phase stress and the grain-orientation-dependent stress in duplex stainless steel during deformation. *Acta Materialia, 54*, 3907-3916.

33. Nilsson, J. O. (1992). Super duplex stainless steels. *Materials Science and Technology, 8*, 685-700.

34. Noori-Khajavi, A. (1993). On multisensor approach to drill wear monitoring. *CIRP annals, 42*(1).

35. Paugam-Moisy, H. (2001). Multi-network system for sensory integration. In *IJCNN 01 International Joint Conference on Neural Networks Proceedings (Cat No 01CH37222) IJCNN-01*.

36. R. Wehrens, L. M. C. B. (2007). Self- and Super-organising Maps in R: the kohonen package. *Journal of Statistical Software, 21*(5).

37. Ramachandran, V. S. H., E. M. . (2001). Synaesthesia: A Window Into Perception, Thought and Language. *JOURNAL OF CONSCIOUSNESS STUDIES, 8*(12), 3-34

38. Rehorn, A. (2005). State-of-the-art methods and results in tool condition monitoring: a review. *International journal of advanced manufacturing technology, 26*(7-8), 693-710.

39. S. S. Panda, D. C. a. S. K. P. (2006). Monitoring of drill flank wear using fuzzy back-propagation neural network *The International Journal of Advanced Manufacturing Technology, 34*(3-4), 227-235.

40. Sanjay, C. (2005). Modeling of tool wear in drilling by statistical analysis and artificial neural network *Journal of materials processing technology, 170*(3).

41. Satoh, I. (2004). Software Agents for Ambient Intelligence. In *2004 IEEE International Conference on Systems Man and Cybernetics (IEEE Cat No 04CH37583) ICSMC-04*.

42. Seneker, S. (2002). *Synesthetic Sensor Fusion Via A Cross-Wired Artificial Neural Network.* Unpublished Master of Arts in Liberal Studies, East Tennessee State University. http://etd-submit.etsu.edu/etd/theses/available/etd-0403102-164937/

43. SG Wysoski, L. B. (2006). Biologically Realistic Neural Networks and Adaptive Visual Information Processing. *Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand*.

44. Skillings, J. (2006). Getting machines to think like us. *Cnet news, Robotics*(July 3, 2006). Retrieved from http://news.cnet.com/Getting-machines-to-think-like-us/2008-11394_3-6090207.html?tag=mncol

45. Smith, C. (2007). Why use duplex stainless steel? *The Fabricator*.

46. T. Saeid, A. A.-z., H. Assadi et al. (2008). Effect of friction stir welding speed on the microstructure and mechanical properties of a duplex stainless steel. *Materials Science and Engineering: A, In Press*(Corrected Proof).

47. T. Siegmund, E. W., and F. Fischer. (1995). On the thermomechanical deformation behavior of duplex-type materials. *Mechanics, Pyhsics and solids, 43*(4), 495-532.

48. Team, R. D. C. (2008). R: A Language and Environment for Statistical Computing. Vienna, Austria.

49. Voronenko, B. (1997). Autanitic-ferritic stainless steels: A state-of-the-art review. *Metal Science and Heat Treatment, 39*(10), 20-29.

50. Walter, G. The Grey Walter Online Archive (Publication no. http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html). from Burden Neurological Institute, University of the West of England, Bristol: http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html

Publications

1. Ashar, J., & Littlefair, G. (2008, October 1-3, 2008). Drill Wear condition monitoring using Self Organizing Feature Maps. Paper presented at the 7th International Conference on Industrial Tooling (IT'08) Mississippi State University, Mississippi, USA.

2. Ashar, J., & Littlefair, G. (2008, 29th - 31st October 2008). *Intelligent Drill Wear Condition Monitoring using Self Organising Feature Maps.* Paper presented at the 4th New Zealand Metals Industry Conference - Building Sustainability, SkyCity Convention Centre, Auckland, New Zealand. (The paper was awarded the "best paper" award under the field of Light Alloy Manufacturing research)

3. Ashar, J. Littlefai, G. (2008, July, 2008). Intelligent Machines. *Engineering News*. http://engineeringnews.co.nz/articles/july08/articles/manufacturing-technology.php

Drill Wear condition monitoring using Self Organizing Feature Maps.

Paper presented at the 7th International Conference on Industrial Tooling (IT'08 - (2008, October 1-3, 2008)

Mississippi State University, Mississippi, USA

Drill Wear Monitoring using Self-organizing Feature Maps

J. Ashar, S. Singamneni, G. Littlefair

Abstract—

The rising demand for exacting performances from manufacturing systems has led to new challenges for the development of complex tool condition monitoring techniques. The work presented here centres around the application of a supervised, self-organizing feature map network model towards the development of a drill wear monitoring system. The neural network organizes itself depending on the input and thus makes for a better classification model than other network models that try and fit input data in a pre-defined structure. This leads to the network structure reflecting the given input distribution more precisely than a predefined model, which generally follows a decay schedule. The generation of tool wear during machining is a dynamic and fast paced developmental problem. The dynamic nature of the network model provides an evaluation of the underlying connectivity and topology in the original data space. This makes the network far more capable of capturing details in the target space.

I. Introduction

Developments in independent flexible manufacturing systems have required more efficient metal cutting procedures. Although a wide range of monitoring methods have been investigated and developed, there has been very little migration of these innovations into industrial practice. The principal factor behind this phenomenon is the stochastic nature of the environment in which the system must function. A truly universal application has yet to be developed [1].

Conventionally, cutting tools have been replaced at the end of encoded, experimentally derived, in-cut times based on chronicled tool life data. However in modern machining environments where large numbers of variables interact, the application of these conventional techniques leads to nonviable and inefficient tool utilisation. The major problem in predicting tool wear accurately stems from the complexity of the process which is dependent on a large number of interrelated variables including the properties of the materials involved, the physical and chemical properties of the surfaces, pressure, temperature, friction and relative velocities [2]. The pressures imposed on the processes and lack of system 'slack' have led to amplified use of tool condition monitoring (TCM) systems. In parallel, there has been wide-ranging research in academia. However, a closer examination shows that there has been very little migration of this research into industrial practice. Furthermore, the success of industrially deployed monitoring systems has been poor [3]. Probably the greatest single obstacle preventing the realisation of the "factory of the future" is the lack of a reliable and all-encompassing tool condition monitoring system.

II. Intelligent Systems

A. The Self-organizing Feature Map

Among the different neural-network learning paradigms, unsupervised learning has attractive characteristics. Learning in unsupervised neural networks emerges without the need of an external "teacher" who provides the desired response of the network. This self-organization ability can substantially reduce the programming burden that accounts for about one third of the total cost of a system. Moreover, unsupervised models are often fast and their learning speeds, especially when using computational shortcuts, can be augmented to orders of magnitude greater than that of numerous other neural networks. Thus much larger maps than those used so far are quite realistic

[4]. Self-organising feature maps (SOFMs), or just Self Organising Maps (SOMs) are important unsupervised Artificial Neural Network (ANN) models that have shown great potential in application fields such as speech recognition applications and various pattern recognition tasks involving very noisy signals. Commonly, the SOFM is used to learn the topology of sensory inputs by clustering the data and is used in control basically as a classifier. The final sensory map can then be used to classify new incoming data. It is important to note that when using supervised models, the error signals are available directly at the output of the network and are explicitly used during network learning and training. In the unsupervised case, the error signals are not computed directly, rather through the use of the definitions in the network's learning rule. For this reason, when unsupervised neural models are used in modelling and control, they are usually referred to as self-supervised models. This type of learning is controlled by knowledge of the external world provided by sensors and the consequences of actions performed by the network. These networks have also provided acumen into how mammalian brains are organised [5]. A visualisation of a SOFM is shown in fig. 1.

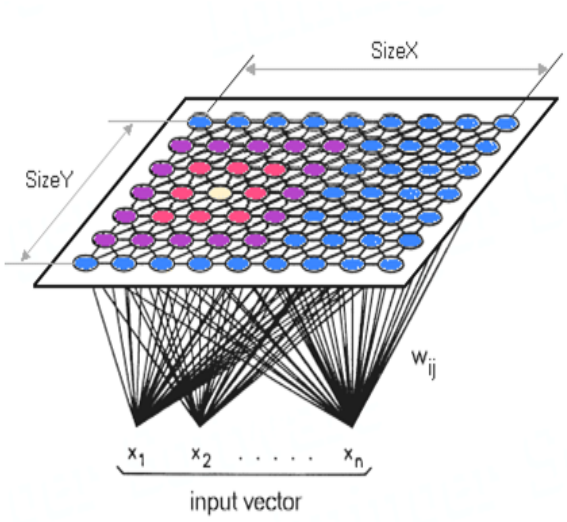


Figure 1: Kohonen's Self-organizing feature map

Over the past decades, the field of AI has made great progress toward digitising human reasoning. Figurative approaches are based on the hypothesis of symbolic depiction—the idea that perception and cognitive processes can be modelled as obtaining, influencing, correlating and adapting to the symbolic representations. Perhaps the optimal way to move forward is to shift the focus from modifying system behaviour to the processes of cognition that source the performance of the ANNs [6]. Most works have concentrated on robotic systems that are solely sensory in nature. Recently, several studies have proposed the Self Organising Feature Map for the difficult tasks of non-linear modelling. The SOFM can extract features of input data based on incremental learning [7]. central result in self-organization is that if the input signals have a distinct

probability density function, then the weight vectors of the cells try to match it, however complex its form.

The SOFM is a neural network that closely resembles how the brain functions. Emulating the way in which human brains decode data from various sources (senses) holds tremendous value. Perhaps the analogy to convey the approach most simply, is to consider just how many of us would cross the road without looking both ways but rather rely on our sense of hearing as the sole arbitrator? Data captured to characterise the condition of a complex piece of equipment should be as complete as the data we use to cross the road – i.e. contain primary and secondary data which is used to arrive at a consensus of opinion.

B. Linking the brain and the computer

It is an oversimplification to say that it is impossible to artificially imitate the human brain due to the limitations of current computational resources. In actuality, the key concern for failing to properly emulate the human way of information processing is the existence of many un-interpreted details of the brain structure and behaviour [8]. In an attempt to discover what instigates the reasoning of human minds, one of the most testing aspects for neuroscientists is that current technologies cannot keep track and measure all the signals used for inter-neural communication, even in a minute portion of the brain. If this was possible, it would enable us to accurately appreciate the emergence of intelligence from a collection of neurons. In an attempt to overcome this limitation, a common practice is to complement the study with the development of intelligent computational models based on experimental data and to study their properties by theoretical and simulation means [9]. The SOFM is proposed as a feasible elective to more traditional neural network architectures. Its analytical portrayal has already been developed further in the technical than in the biological direction. The learning results accomplished seem to be as expected; at least indicating that the adaptive processes at work in the map may be analogous to those encountered in the brain.

III. Experimental Procedure

Drilling tests were performed using a solid carbide drill on a duplex steel workpiece. The Kistler three component quartz dynamometer (type 9257B) was selected as the force measuring sensor. This type of dynamometer consists of four three-component force sensors fitted under high pre-load between a base-plate and a top-plate. Each sensor contains

three pairs of quartz plates: one sensitive to pressure in the Z direction and the other two in the X and Y directions. As a result, the dynamometer is able to detect the smallest dynamic changes in large forces. In addition, an armoured connecting cable type 1687B5 and an eight-channel charge amplifier type 5070 were utilised in the measurement of forces. A schematic of the experimental setup is shown in fig. 2.

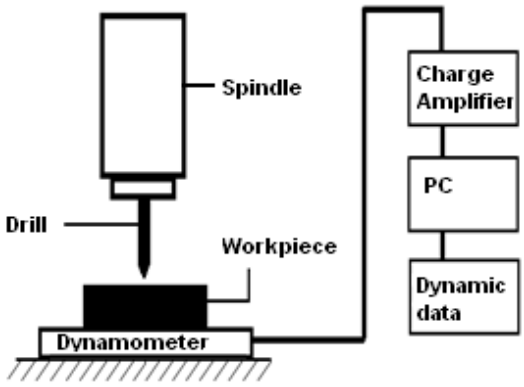


Figure 2: Schematic of the experimental setup

Trials were conducted using solid carbide drills used to make 12.5mm diameter holes on the duplex steel workpieces. The forces along three axes were measured, i.e. Fx, Fy and Fz. The moment around the Z-axis was calculated using these forces (see fig. 3).

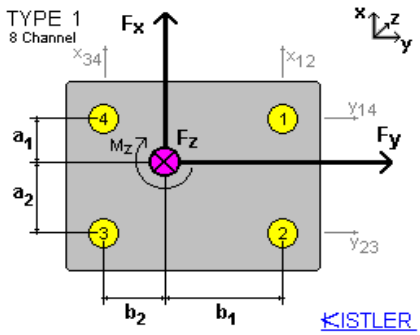


Figure 3: Snapshot of dynamometer measurement

A. Neural Net setup

The data obtained from the drilling trials was analysed and fed as a part of the input to a Self Organising Feature Map. The power spectral density and kurtosis calculations were done on the dynamic data acquired. This was done using DADisp software. The inputs to the Neural Network were the feed rate, drill speed, dynamic forces (psd) and the "static" force component. The network was trained to classify a used drill from a new one. A total of 58 drilling trials were conducted which yielded a collection of 576 data sets. 288 of these sets were used for training the network while the rest were used for testing. The neural network coding and simulation was done using R. R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical and graphical techniques, and is highly extensible [10]. The program written for the SOFM is based

on the function provided in the "kohonen" package of R [11]. Although the basic calculation subroutine is little changed, the data input, handling, execution, storage and output formatting is all original. Before processing is done by the neural network, a pre-processing procedure is carried out to reduce dimensionality of the data. This is achieved by computing the power spectral densities of the signals.

B. Classification

The SOFM was trained to classify a used drill from a new drill with a number of typical data sets corresponding to various wear categories. This classification was made on the basis of the input parameters viz, the forces in the x, y and z axes, the moment of forces about the z-axis, drill speed and feed rate. The training progress of the map is shown in fig. 4. The mean distance to the closest neuron on the map stabilises after approximately 40,000 runs. A snapshot of the classification pattern is shown in fig. 5.

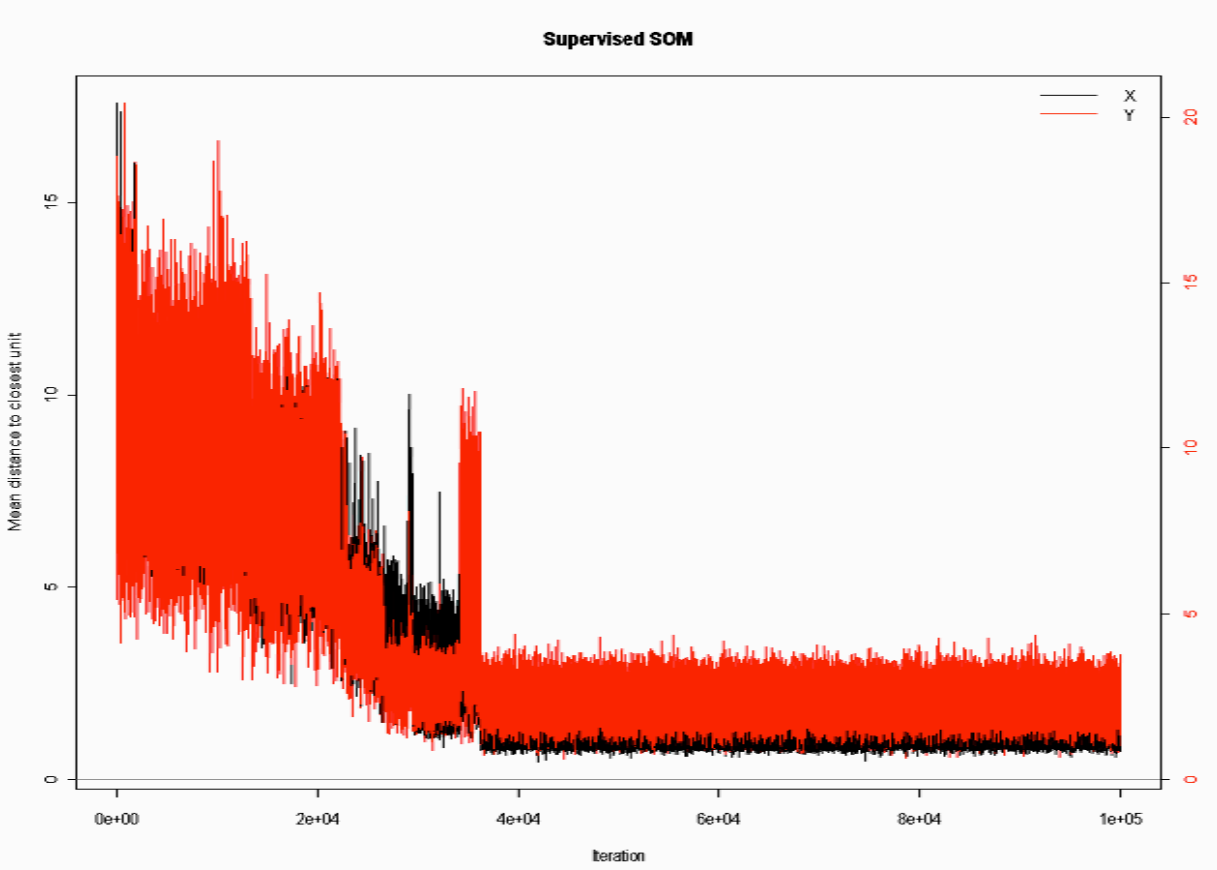


Figure 4: Progress after 100,000 runs

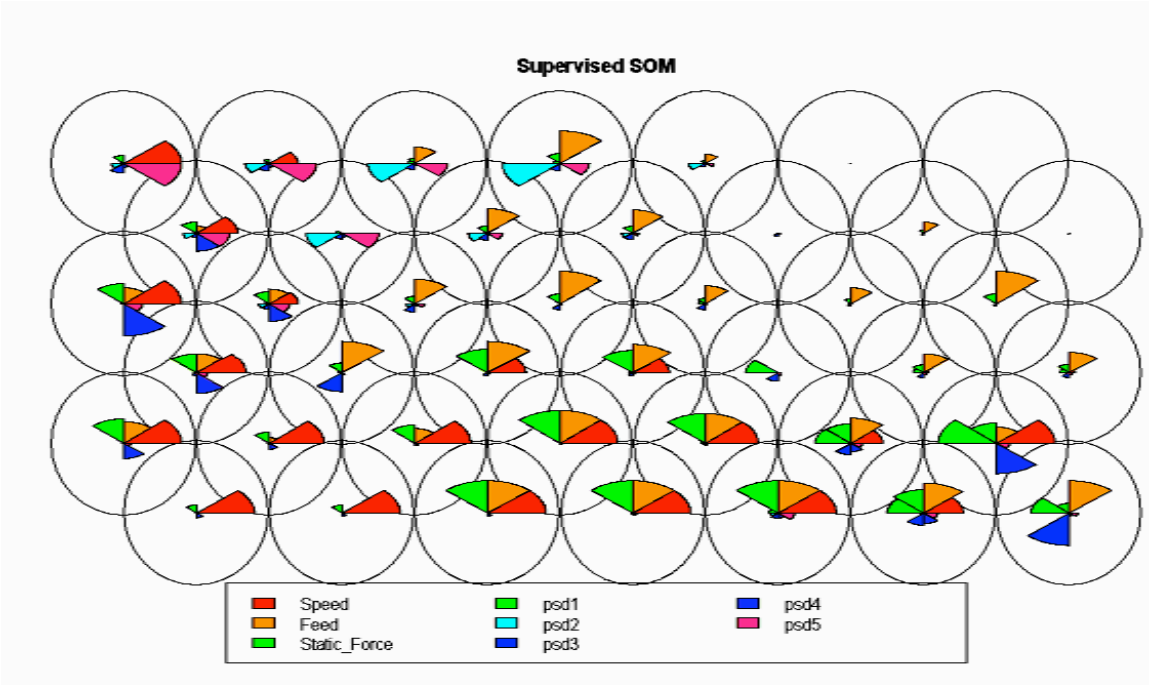


Figure 5: Snapshot of SOM classification after 100,000 runs

Subsequent classification of previously unseen data sets indicated that the network was able to correctly organise itself to new data sets.

IV. Conclusions

Tool wear identification and monitoring is a complex phenomenon. Accurate modelling of the problem requires a highly evolved and comprehensive solution. The work done so far has mainly focussed on the use of neural networks which learn using the supervised learning paradigm. These networks perform well under known conditions, but even a minor deviation from their predefined parameters can cause such networks to fail. The main aim of this research is to construct a robust and efficient system for tool wear monitoring in drilling operations.

The Self-Organising Feature Map is a neural network that closely resembles how the brain functions. Mirroring the way in which human brains decode data from various sources (senses) holds tremendous merit. Data captured to characterise the condition of a complex piece of equipment should contain primary and secondary data which is used to arrive at a consensus of opinion.

Learning in this network is unsupervised, thus making it independent of human errors caused during the training phase. Furthermore, the network is able to adapt to changing environments and conditions. This flexibility in adaption goes well with the stochastic nature of industrial environments.

The work presented here demonstrates the type of system which can successfully be employed to monitor machining operations. The true robustness of the system is to be established by the application of the system in other industrial environments. The classification of tool wear using unsupervised neural networks is regarded as a strategic step forward in the progress towards the creation of a truly unmanned machining environment.

V. References

1. Littlefair, G., *Multisensor Condition Monitoring using the Fusion Characteristics of Artificial Neural Networks*, in *Vibration Association of New Zealand Annual conference*. 2007: Hamilton, New Zealand.
2. FRANCOGASCA, L., *Sensorless tool failure monitoring system for drilling machines* International journal of machine tools & manufacture, 2006. **46**(3-4).
3. Brophy, B., *AI-based condition monitoring of the drilling process.* Journal of materials processing technology, 2002. **124**(3): p. 305.
4. Kohonen, T., *The self-organizing map.* Proceedings of the IEEE, 1990. **78**(9): p. 1464-1480.
5. de Barreto, G., Self-Organizing Feature Maps for Modeling and Control of Robotic Manipulators. Journal of intelligent & robotic systems, 2003. 36(4): p. 407-450.
6. Littlefair, G., Intelligent Machines, in Engineering News. 2008: Auckland.
7. Kohonen, T., Analysis of processes and large data sets by a self-organizing method. Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on, 1999 1: p. 27 - 36
8. SG Wysoski, L.B., Biologically Realistic Neural Networks and Adaptive Visual Information Processing. Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand, 2006.
9. Rehorn, A., State-of-the-art methods and results in tool condition monitoring: a review. International journal of advanced manufacturing technology, 2005. 26(7-8): p. 693-710.
10. Team, R.D.C., R: A Language and Environment for Statistical Computing. 2008: Vienna, Austria. p. R Foundation for Statistical Computing.
11. R. Wehrens, L.M.C.B., Self- and Super-organising Maps in R: the kohonen package. Journal of Statistical Software, 2007. 21(5).

Intelligent Drill Wear Condition Monitoring using Self Organising Feature Maps

Paper presented at the *4th New Zealand Metals Industry Conference - Building Sustainability*. 29th - 31st October 2008

HERA: SkyCity Convention Centre, Auckland, New Zealand.

AWARDED BEST PAPER AWARD FOR LIGHT ALLOW MANUFACTURING RESEARCH

INTELLIGENT DRILL WEAR CONDITION MONITORING USING SELF ORGANISING FEATURE MAPS

Jesal Ashar
Mach Research Group Group
AUT University
Auckland
New Zealand
+64 9 9219999x8109
jesal.ashar@aut.ac.nz

Dr. Sarat Singamneni
School of Engineering

AUT Univeristy
Auckland
New Zealand
+64 9 9219999
ssingam@aut.ac.nz

Dr. Guy Littlefair
Mach Research

AUT University
Auckland
New Zealand
+64 9 9219999
glittle@aut.ac.nz

ABSTRACT—
The rising demand for exacting performances from manufacturing systems has led to new challenges for the development of complex tool condition monitoring techniques. The work presented here centres around the application of a supervised, self-organising feature map network model towards the development of a drill wear monitoring system. The neural network organises itself depending on the input and thus makes for a better classification model than other network models that try and fit input data in a pre-defined structure. This leads to the network structure reflecting the given input distribution more precisely than a predefined model, which generally follows a decay schedule. The generation of tool wear during machining is a dynamic and fast paced developmental problem. The dynamic nature of the network model provides an evaluation of the underlying connectivity and topology in the original data space. This makes the network far more capable of capturing details in the target space.

I. INTRODUCTION

Developments in autonomous manufacturing systems have required more pliant and efficient metal cutting procedures. Although a wide range of monitoring methods have been investigated and developed, there has been very little migration of these innovations into industrial practice. The stochastic nature of the environment, in which manufacturing systems must function, is the primary reason behind this poor rate of conversion. A truly universal application has yet to be developed [1].

Conventionally, cutting tools have been replaced at the end of programmed, experimentally derived, in-cut times based on historical tool life data. However, in modern machining environments where large numbers of variables interact, the application of these conventional techniques leads to nonviable and inefficient tool utilisation. The principal difficulty in predicting tool wear accurately stems from the complexity of the process - which is dependent on a large number of interrelated variables including the properties of the materials involved, the physical

and chemical properties of the surfaces, pressure, temperature, friction and relative velocities [2]. The pressures imposed on the processes and lack of system 'slack' have led to amplified use of tool condition monitoring (TCM) systems. In parallel, there has been wide-ranging research in academia. However, a closer examination shows that there has been very little migration of this research into industrial practice. Furthermore, the success of industrially deployed monitoring systems has been poor [3]. Probably the greatest single obstacle preventing the realisation of the "factory of the future" is the lack of a reliable and all-encompassing tool condition monitoring system.

II. INTELLIGENT SYSTEMS

A. The Self-organising Feature Map

Self-organising maps are a data visualisation technique invented by Professor Teuvo Kohonen which reduces the dimensionality of data through the use of self-organising neural networks. The problem that data visualisation attempts to solve is that humans simply cannot visualise high dimensional data as is and thus techniques are created to help us understand this high dimensional data. Among the different neural-network learning paradigms, unsupervised learning has attractive characteristics. Learning in unsupervised neural networks emerges without the need of an external "teacher" who provides the desired response of the network. This self-organisation ability can substantially reduce the programming burden that accounts for about one third of the total cost of a system. Moreover, unsupervised models are often fast and their learning speeds, especially when using computational shortcuts, can be augmented to orders of magnitude greater than that of numerous other neural networks. Thus much larger maps than those used so far are quite realistic [4]. Self-organising feature maps (SOFMs), or just Self Organising Maps (SOMs) are important unsupervised Artificial Neural Network (ANN) models that have shown great potential in application fields such as speech recognition applications and various pattern recognition tasks involving very noisy signals. Commonly, the SOFM is used to learn the topology of sensory inputs by clustering the data and is used in control basically as a classifier. The final sensory map can then be used to classify new incoming data. It is important to note that when using supervised models, the error signals are available directly at the output of the network and are explicitly used during network learning and training. In the unsupervised case, the error signals are not computed directly, rather through the use of the definitions in the network's learning rule. For this reason, when unsupervised neural models are used in modelling and control, they are usually referred to as self-supervised models. This type of learning is controlled by knowledge of the external world provided by sensors and the consequences of actions performed by the network. These networks have also provided acumen into how mammalian brains are organised [5]. A visualisation of a SOFM is shown in fig. 1.

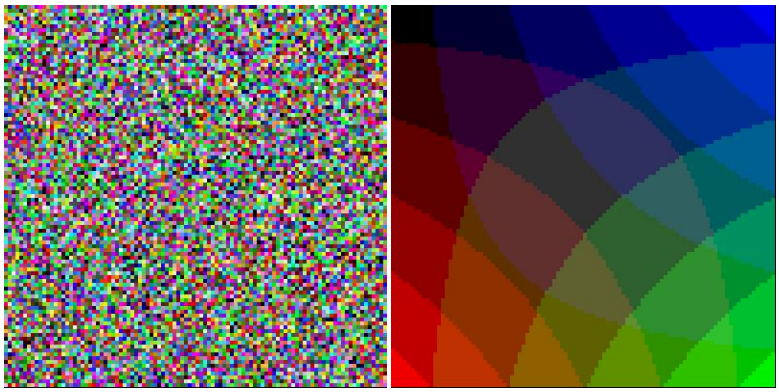


Figure 1: Self - organisation

Over the past decades, the field of AI has made great progress toward transcribing human reasoning into digital data. Figurative approaches are based on the hypothesis of symbolic depiction—the idea that perception and cognitive processes can be modelled as acquiring, influencing, co-relating and adapting to the symbolic representations. Perhaps the best way to move forward is to shift the focus from modifying system behaviour to the processes of cognition that source the performance of the ANNs [6]. Most works have concentrated on robotic systems that are solely sensory in nature. Recently, several studies have proposed the Self Organising Feature Map for the difficult tasks of non-linear modelling. The SOFM can extract features of input data based on incremental learning. The central result in self-organisation is that if the input signals have a distinct probability density function, then the weight vectors of the cells try to match it, however complex its form.

The SOFM is a neural network that closely resembles how the brain organises memory into neuronal connections. Emulating the way in which human brains decode data from various sources (senses) holds tremendous value. Perhaps the analogy to convey the approach most simply, is to consider just how many of us would cross the road without looking both ways but rather rely on our sense of hearing as the sole arbitrator? Data captured to characterise the condition of a complex piece of equipment should be as complete as the data we use to cross the road – i.e. contain primary and secondary data which is used to arrive at a consensus of opinion.

A. Linking the brain and the computer- chaos and synaesthesia

It is a generalisation to say that it is impossible to artificially imitate the human brain due to the limitations of current computational resources. In actuality, the key concern for failing to properly emulate the human way of information processing is the existence of many un-interpreted details of the brain structure and behaviour [7]. Our brain is chaotic. Chaos has been found in how we process external senses, and may be key to memory. It has been implicated in at least

one theory of the evolution of vocabulary as well as synaesthesia [8]. Synesthesia is a neurological phenomenon in which the stimulation of one sensory organ leads to automatic or involuntary sensations in another sensory organ. In its most common form known as grapheme → colour synesthesia, letters and/or numbers are perceived as inherently coloured and having personalities. Synesthesia has been being diagnosed for almost three centuries, but the medical profession keeps forgetting about the condition. The word *Synesthesia* means "joined sensation" and shares a root with *anesthesia* which means "no sensation". Synesthesia is not an abnormality; in fact it is a normal brain development process that is intuitively presented to the consciousness in a minority of individuals. The condition symbolises a rare ability to hear colours, taste shapes, or experiences of other equally astounding sensory amalgamations whose nature seems too complex for most of us to envisage. Synesthetes are normal in the conventional sense of the term and they appear to be bright and intelligent. Standard neurological medical exams are also normal. Synesthetic associations are usually unidirectional, meaning that a particular synesthete sight may induce touch, but touch would not induce visual sensations. Simulating synesthetic type of neurological behaviour in Artificial Neural Networks (the core of Artificially Intelligent Systems) will help shed light on their functioning and classification capabilities. This in turn may also deepen our understanding as to why these systems are unstable when applied in real world environments. The process of disassembly and reassembly takes on an entirely new meaning. The eventual goal is to create efficient, robust systems with extended autonomous control over processes that are being employed – essentially creating the "factory of the future" [9]. Most models of the brain do not include chaos. Those that do, don't seem convincingly biological. In an attempt to discover what instigates the reasoning of human minds, one of the most testing aspects for scientific analysis is that the current technologies cannot keep track and measure all the signals used for inter-neural communication, even in an infinitesimal portion of the brain. If this was possible, it would enable us to accurately appreciate the emergence of intelligence from a collection of neurons. In an attempt to overcome this limitation, a common practice is to complement the study with the development of intelligent computational models based on experimental data and to study their properties by theoretical and simulation means [10]. The SOFM is proposed as a feasible elective to more traditional neural network architectures. Its analytical portrayal has already been developed further in the technical than in the biological direction. The learning results accomplished seem to be as expected; at least indicating that the adaptive processes at work in the map may be analogous to those encountered in the brain.

I. EXPERIMENTAL PROCEDURE

Drilling tests were performed using a solid carbide drill on a duplex steel workpiece. The Kistler three component quartz dynamometer (type 9257B) was selected as the force measuring

sensor. This type of dynamometer consists of four three-component force sensors fitted under high pre-load between a base-plate and a top-plate. Each sensor contains three pairs of quartz plates: one sensitive to pressure in the Z direction and the other two in the X and Y directions. As a result, the dynamometer is able to detect the smallest dynamic changes in large forces. In addition, an armoured connecting cable type 1687B5 and an eight-channel charge amplifier type 5070 were utilised in the measurement of forces. A schematic of the experimental setup is shown in fig. 2.

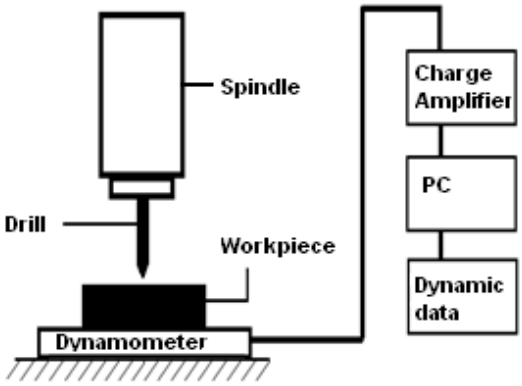


Figure 2: Schematic of the experimental setup

Trials were conducted using solid carbide drills used to make 12.5mm diameter holes on the duplex steel workpieces. The forces along three axes were measured, i.e. Fx, Fy and Fz. The moment around the Z-axis was calculated using these forces (see fig. 3).

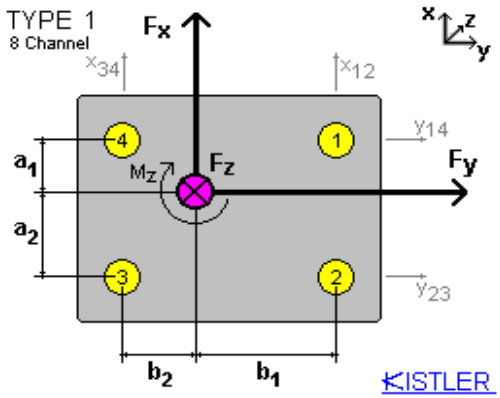


Figure 3: Snapshot of dynamometer measurements

A. Neural Network setup

The data obtained from the drilling trials was analysed and fed as a part of the input to a Self Organising Feature Map. The power spectral density and kurtosis calculations were done on the dynamic data acquired. This was done using DADisp software. The inputs to the Neural Network were the feed rate, drill speed, dynamic forces (psd) and the "static" force component. The network was trained to classify a used drill from a new one. A total of 58 drilling trials were conducted which yielded a collection of 576 data sets. 288 of these sets were used for training the network while the rest were used for testing. The neural network coding and simulation was done using R. R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical and graphical techniques, and is highly extensible [11]. The program written for the SOFM is based on the function provided in the "kohonen" package of R [12]. Although the basic calculation subroutine is little changed, the data input, handling, execution, storage and output formatting is all original. Before processing is done by the neural network, a pre-processing procedure is carried out to reduce dimensionality of the data. This is achieved by computing the power spectral densities of the signals.

A. Classification

The SOFM was trained to classify a used drill from a new drill with a number of typical data sets corresponding to various wear categories. This classification was made on the basis of the input parameters viz, the forces in the x, y and z axes, the moment of forces about the z-axis, drill speed and feed rate. The training progress of the map is shown in fig. 4. The mean distance to the closest neuron on the map stabilises after approximately 40,000 runs. A snapshot of the classification pattern is shown in fig. 5. Subsequent classification of previously unseen data sets indicated that the network was able to correctly organise itself to new data sets.

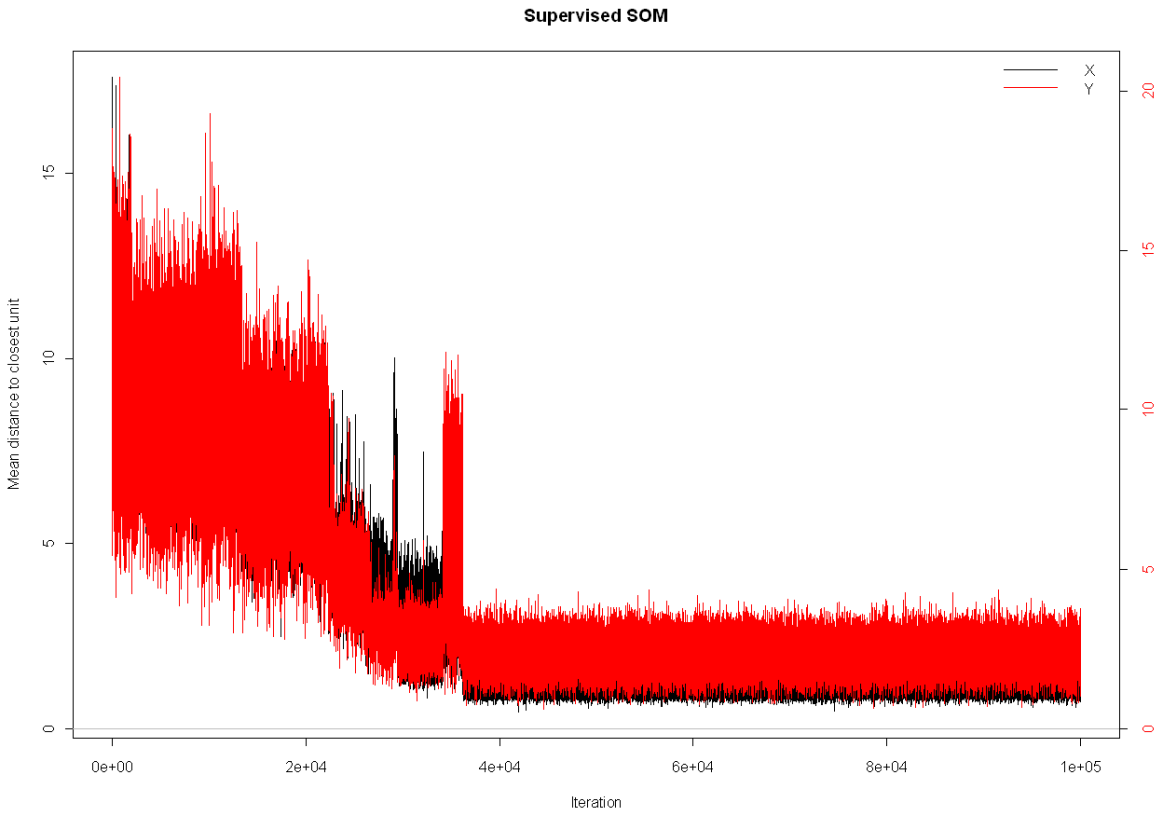


Figure 4: SOM progress after 100,000 runs

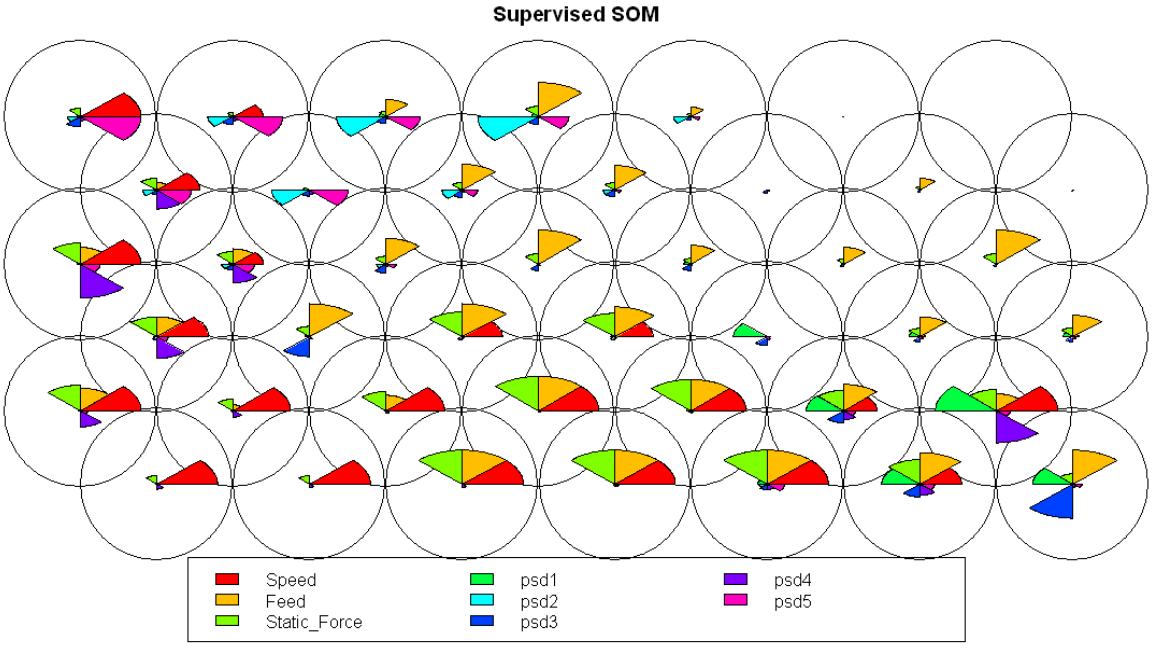


Figure 5: Snapshot of SOM classification after 100,000 runs

I. CONCLUSIONS

Tool wear identification and monitoring is a complex phenomenon. Accurate modelling of the problem requires a highly evolved and comprehensive solution. The work done so far has mainly focussed on the use of neural networks which learn using the supervised learning paradigm. These networks perform well under known conditions, but even a minor deviation from their predefined parameters can cause such networks to fail. The main aim of this research is to construct a robust and efficient system for tool wear monitoring in drilling operations.

The Self-Organising Feature Map is a neural network that closely resembles how the brain functions. Mirroring the way in which human brains decode data from various sources (senses) holds tremendous merit. Data captured to characterise the condition of a complex piece of equipment should contain primary and secondary data which is used to arrive at a consensus of opinion.

Learning in this network is unsupervised, thus making it independent of human errors caused during the training phase. Furthermore, the network is able to adapt to changing environments and conditions. This flexibility in adaption goes well with the stochastic nature of industrial environments.

The work presented here demonstrates the type of system which can successfully be employed to monitor machining operations. The true robustness of the system is to be established by the application of the system in other industrial environments. The classification of tool wear using unsupervised neural networks is regarded as a strategic step forward in the progress towards the creation of a truly unmanned machining environment.

REFERENCES

1. Littlefair, G., *Multisensor Condition Monitoring using the Fusion Characteristics of Artificial Neural Networks*, in *Vibration Association of New Zealand Annual conference*. 2007: Hamilton, New Zealand.

2. FRANCOGASCA, L., *Sensorless tool failure monitoring system for drilling machines* International journal of machine tools & manufacture, 2006. **46**(3-4).

3. Brophy, B., *AI-based condition monitoring of the drilling process.* Journal of materials processing technology, 2002. **124**(3): p. 305.

4. Kohonen, T., *The self-organizing map.* Proceedings of the IEEE, 1990. **78**(9): p. 1464-1480.

5. de Barreto, G., *Self-Organizing Feature Maps for Modeling and Control of Robotic Manipulators.* Journal of intelligent & robotic systems, 2003. **36**(4): p. 407-450.

6. Kohonen, T., *Analysis of processes and large data sets by a self-organizing method.* Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on, 1999 **1**: p. 27 - 36

7. SG Wysoski, L.B., *Biologically Realistic Neural Networks and Adaptive Visual Information Processing.* Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand, 2006.

8. FIRE, K., *LINKING CHAOS IN THE MODEL TO CHAOS IN THE BRAIN.* 2003.

9. Ashar, J.L., G., *Intelligent Machines*, in *Engineering News*. 2008: Auckland.

10. Rehorn, A., *State-of-the-art methods and results in tool condition monitoring: a review.* International journal of advanced manufacturing technology, 2005. **26**(7-8): p. 693-710.

11. Team, R.D.C., *R: A Language and Environment for Statistical Computing*. 2008: Vienna, Austria. p. R Foundation for Statistical Computing.

12. R. Wehrens, L.M.C.B., *Self- and Super-organising Maps in R: the kohonen package.* Journal of Statistical Software, 2007. **21**(5).

Ashar, J. & Littlefair, G.

Intelligent Machines

***Engineering News* (July, 2008).**

http://engineeringnews.co.nz/articles/july08/articles/

manufacturingtechnology.php

Intelligent machines By Guy Littlefair

The Machining and Machinability Research group at AUT University may have found a way to significantly improve the current performance of Tool Wear Monitoring and Tool Breakage Detection Systems by incorporating human phenomena into the working of these systems. Jesal Ashar, who is undertaking the research, says: "Looking at a human medical phenomenon known as Synesthesia, otherwise known as the sixth sense syndrome, we are able to better understand the working of Intelligent Systems and more importantly how they react to varying stimuli."

Synesthesia is a neurological phenomenon in which the stimulation of one sensory organ leads to automatic or involuntary sensations in another sensory organ. In its most common form known as grapheme – colour synesthesia, letters and/or numbers are perceived as inherently coloured and having personalities. Synesthesia has been diagnosed for almost three centuries, but medicine keeps forgetting about the condition. The word synesthesia means "joined sensation" and shares a root with anesthesia which means "no sensation". Synesthesia is not an abnormality; in fact it is a normal brain development process that is intuitively presented to the consciousness in a minority of individuals. The condition symbolises a rare ability to hear colours, taste shapes, or experiences of other equally astounding sensory amalgamations whose nature seems complex for most of us to envisage. Synesthetes are normal in the conventional sense of the term. They appear to be bright and intelligent. Standard neurological medical exams are also normal. Synesthetic associations are usually unidirectional, meaning that for a particular synesthete, sight may induce touch, but touch would not induce visual sensations.

An artificial neural network (ANN), often just called a "neural network" (NN), is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In more practical terms neural networks are non-linear statistical data modelling tools. Jesal says: "They can be used to model complex relationships between inputs and outputs or to find patterns in data. There is no precise agreed-upon definition among researchers as to what a neural network is, but most would agree that it involves a network of simple processing elements (neurons), which can exhibit complex global behaviour, determined by the connections between the processing elements and element parameters."

The original inspiration for the technique was from examination of the central nervous system. ANNs, like people, learn by example. An ANN may be configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

An impression of the functioning of Artificial Neural Networks Simulating synesthetic type of neurological behaviour in Artificial Neural Networks (the core of Artificially Intelligent Systems) will help shed light on their functioning and classification capabilities. This in turn, may also deepen our understanding as to why these systems are unstable when

applied in real world environments. The process of disassembly and reassembly takes on an entirely new meaning. The eventual goal is to create efficient, robust systems with extended autonomous control over processes that are being employed – essentially creating the "factory of the future".

Perhaps the best way to move forward is to shift the focus from modifying system behaviour to the processes of cognition that source the performance. As the saying goes, prevention is better than cure. There is no complete theory that has been presented to date which explains how sensors influence and lead each other to produce more accurate or confident perception. It has been hypothesised in literature that the strength of inter-neuronal connections in our brains is the foundation for memories. The stronger the connections are, the better is the memory recall. However, we only have a modest estimate of the connectivity in our brains, how a collection of grey cells performs as an ensemble and how information is programmed.

Jesal sees great potential for her research and comments: "So, where does tool wear figure in all this? The answer is simple – to improve upon the systems that are currently used to monitor tool wear by creating "biologically realistic" ANNs. The future may well be the factory that runs on auto-pilot!"

Appendix A

Specifications for:

Kistler 3 Component Force Dynamometer

Kistler 4 channel Charge Amplifier

Force – FMD

KISTLER

P.
1 ... 4

3-Komponenten-Dynamometer F_x, F_y, F_z
Dynamomètre à 3 composantes F_x, F_y, F_z
3-Component Dynamometer F_x, F_y, F_z

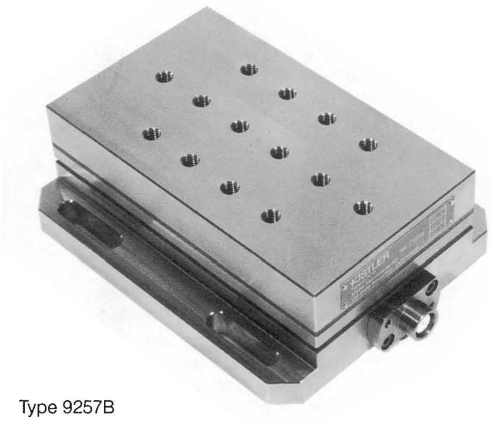
Type

9257B, 9403

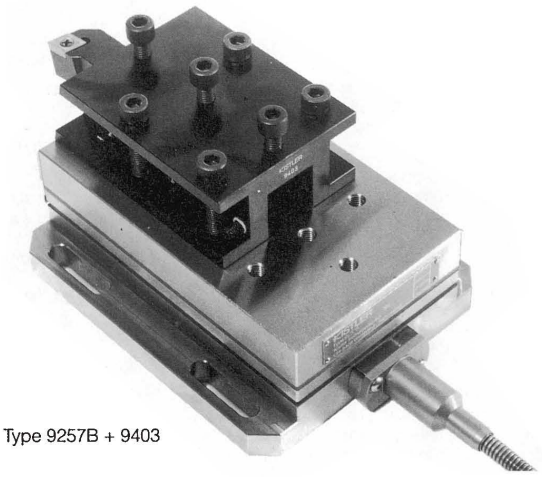
Quarzkristall-Dreikomponenten-Dynamometer zum Messen der drei orthogonalen Komponenten einer Kraft. Das Dynamometer besitzt eine grosse Steifheit und demzufolge eine hohe Eigenfrequenz. Das grosse Auflösungsvermögen ermöglicht das Messen von kleinsten dynamischen Änderungen grosser Kräfte.

Dynamomètre à cristal de quartz à trois composantes pour mesurer des trois composantes orthogonales d'une force. Le dynamomètre possède une grande rigidité et par conséquent une fréquence propre élevée. Sa très haute résolution permet de mesurer les moindres variations de larges forces.

Quartz three-component dynamometer for measuring the three orthogonal components of a force. The dynamometer has a great rigidity and consequently a high natural frequency. Its high resolution enables the smallest dynamic changes in large forces to be measured.



Type 9257B



Type 9257B + 9403

Technische Daten	Données techniques	Technical Data			
Bereich F_z bei F_x und F_y ≤0,5 F_z	**Gamme** F_z pour F_x et F_y ≤0,5 F_z	**Range** F_z for F_x and F_y ≤0,5 F_z	F_x, F_y, F_z F_z	kN kN	–5 ... 5 *) –5 ... 10 **)
Kalibrierter Teilbereich 1	**Gamme partielle étalonnée 1**	**Calibrated partial range 1**	F_x, F_y F_z	N N	0 ... 500 0 ... 1000
Kalibrierter Teilbereich 2	**Gamme partielle étalonnée 2**	**Calibrated partial range 2**	F_x, F_y F_z	N N	0 ... 50 0 ... 100
Überlast F_z bei F_x und F_y ≤0,5 F_z	**Surcharge** F_z pour F_x et F_y ≤0,5 F_z	**Overload** F_z for F_x and F_y ≤0,5 F_z	F_x, F_y, F_z F_z	kN kN	–7,5/7,5 –7,5/15
Ansprechschwelle	**Seuil de réponse**	**Threshold**		N	<0,01
Empfindlichkeit	**Sensibilité**	**Sensitivity**	F_x, F_y F_z	pC/N pC/N	≈–7,5 ≈–3,7
Linearität, alle Bereiche **Hysterese,** alle Bereiche **Übersprechen**	**Linéarité,** toutes les gammes **Hystérésis,** toutes les gammes **Cross talk**	**Linearity,** all ranges **Hysteresis,** all ranges **Cross talk**		% FSO % FSO %	≤±1 ≤0,5 ≤±2
Steifheit	**Rigidité**	**Rigidity**	c_x, c_y c_z	kN/µm kN/µm	>1 >2
Eigenfrequenz **Eigenfrequenz** (montiert an Flanschen)	**Fréquence propre** **Fréquence propre** (installé sur brides)	**Natural frequency** **Natural frequency** (mounted on flanges)	f_o (x, y, z) f_o (x, y) f_o (z)	kHz kHz kHz	≈3,5 ≈2,3 ≈3,5
Betriebstemperaturbereich **Temperaturkoeffizient** der Empfindlichkeit	**Gamme de température d'utilisation** **Coefficient de température** de la sensibilité	**Operating temperature range** **Temperature coefficient** of sensitivity		°C %/°C	0 ... 70 –0,02
Kapazität (pro Kanal) **Isolationswiderstand** (20 °C) **Masseisolation**	**Capacité** (de canal) **Résistance d'isolement** (20 °C) **Isolé à la masse**	**Capacitance** (of channel) **Insulation resistance** (20 °C) **Ground insulation**		pF Ω Ω	≈220 $>10^{13}$ $>10^{8}$
Schutzart **Gewicht**	**Classe de protection** **Poids**	**Protection class** **Weight**		– kg	IP 67 ***) 7,3

*) Kraftangriff innerhalb und max. 25 mm oberhalb der Deckfläche.
**) Bereich beim Drehen, Kraftangriff bei Punkt A.
***) Mit Anschlusskabel Typen 1687B5, 1689B5

*) Point d'application de la force au-dedans et max. 25 mm au-dessus de la plaque supérieure.
**) Gamme lors du tournage, point d'application au point A.
***) Avec câble de connexion types 1687B5, 1689B5

*) Application of force inside and max. 25 mm above top plate area.
**) Range for turning, application of force at point A.
***) With connecting cable Types 1687B5, 1689B5

1 N (Newton) = 1 kg · m · s^{-2} = 0,1019... kp = 0,2248... lbf; 1 inch = 25,4 mm; 1 kg = 2,2046... lb; 1 Nm = 0,73756... lbft

000-151m-02.91 (DB06.9257Bm)

Beschreibung

Das Dynamometer besteht aus vier Dreikomponenten-Kraftsensoren, die unter hoher Vorspannung zwischen einer Grundplatte und einer Deckplatte eingebaut sind. Die Kraftsensoren enthalten je drei Quarzkristall-Plattenpaare, wovon das eine auf Druck in der z-Richtung und die beiden andern auf Schub in der x- bzw. y-Richtung empfindlich sind. Die Kraftkomponenten werden praktisch weglos gemessen.

Die Ausgänge der vier eingebauten Kraftsensoren sind im Dynamometer so zusammengeschaltet, dass auch Mehrkomponenten-Kraft- und Momentmessungen möglich sind. Die acht Ausgangssignale sind an die 9-polige Flanschdose geführt.

Die vier Sensoren sind masseisoliert eingebaut. Damit werden Erdschleifenprobleme weitgehend ausgeschaltet.

Das Dynamometer ist rostbeständig und gegen das Eindringen von Spritzwasser bzw. Kühlmittel geschützt. Zusammen mit dem Anschlusskabel Typ 1687B5/1689B5 genügt das Dynamometer der Schutzklasse IP 67.

In die Deckplatte ist eine spezielle thermische Isolationsschicht eingebaut, die das Dynamometer gegen Temperatureinflüsse weitgehend unempfindlich macht.

Description

Le dynamomètre se compose de quatre capteurs de force à trois composantes montés sous précontrainte élevée entre une plaque de base et une plaque supérieure. Les capteurs de force comprennent chacun trois paires de plaquettes en cristal de quartz; l'une est sensible à la pression selon l'axe z alors que les deux autres sont sensibles au cisaillement selon l'axe x resp. y. Les composantes de la force sont mesurées pratiquement sans déformation.

Les sorties des quatre capteurs de force incorporés sont branchées à l'intérieur du dynamomètre de façon à rendre possible des mesures de forces et moments à plusieures composantes. Les huit signaux de sortie sont disponibles sur la prise femelle à bride et à 9 pôles.

Les quatre capteurs sont montés avec isolement par rapport à la masse. Ainsi les problèmes de circuits de retour par la terre sont largement éliminés.

Le dynamomètre est résistant à la rouille et protégé contre la pénétration de projections d'eau et d'agents refrigérants. Ensemble avec le câble type 1687B5/1689B5 il correspond à la classe de protection IP 67.

Un recouvrement thermique spécial est installé dans la plaque supérieure qui rend le dynamomètre largement insensible contre les influences de température.

Description

The dynamometer consists of four three-component force sensors fitted under high preload between a baseplate and a top plate. Each sensor contains three pairs of quartz plates, one sensitive to pressure in the z direction and the other two responding to shear in the x and y directions respectively. The force components are measured practically without displacement.

The outputs of the four built-in force sensors are connected inside the dynamometer in a way to allow multicomponent measurements of forces and moments to be performed. The eight output signals are available at the 9-conductor flange socket.

The four sensors are mounted ground-insulated. Therefore ground loop problems are largely eliminated.

The dynamometer is rustproof and protected against penetration of splashwater and cooling agents. Together with the connecting cable Type 1687B5/1689B5 it corresponds to the protection class IP 67.

A special thermal isolation coating is integrated in the top plate which renders the dynamometer largely insensitive to temperature influences.

Anwendungsbeispiele

- Dynamisches und quasistatisches Messen der drei orthogonalen Komponenten einer Kraft.
- Schnittkraftmessungen beim Drehen, Fräsen, Schleifen usw. Die hohe Empfindlichkeit und die niedere Ansprechschwelle lassen in Verbindung mit den kalibrierten Teilbereichen auch exakte Messungen an kleinen Werkzeugen und beim Schleifen zu.
- Messungen an Modellen im Windkanal usw.
- Ergonomische Messungen.

Exemples d'application

- Mesures dynamiques et quasistatiques des trois composantes orthogonales d'une force.
- Mesures des efforts de coupe lors du tournage, du fraisage, du rectifiage, etc. La grande sensibilité et le seuil de réponse bas conjointement avec les games de mesure partielles etalonnées permettent des mesures exactes sur de petits outils ou lors du rectifiage.
- Mesures sur des modèles dans canaux aérodynamiques, etc.
- Mesures ergonométriques.

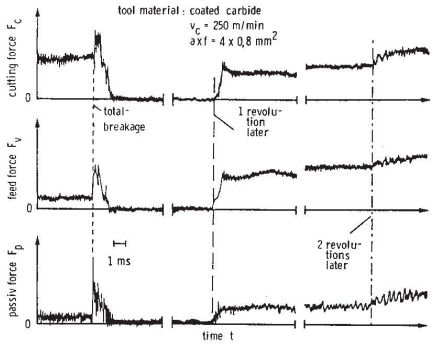
Application Examples

- Dynamic and quasistatic measurement of the three orthogonal components of a force.
- Measuring cutting force when turning, milling, grinding etc. In conjunction with the calibrated partial ranges the high sensitivity and low threshold allow exact measurements on small tools and when grinding.
- Measurements on wind tunnel models, etc.
- Ergonomic measurements.

Die Diagramme unten wurden beim Drehen von CK53N aufgezeichnet. Der Verlauf der Zerspankraftkomponenten bei einem Totalbruch des Hartmetallwerkzeuges ist sehr gut ersichtlich.

La figure ci-dessous montre l'enregistrement lors du tournage de CK53N. Les variations des composantes des efforts de cisaillement lors d'une brisure totale de l'outil en alliage dur sont clairement visibles.

The recorder chart shown below plots the turning of CK53N. The variations of the cutting force components during a carbide tool total break are clearly visible.



Verlauf der Zerspankraftkomponenten bei einem Totalbruch; aus KISTLER-Sonderdruck Nr. 20.112d:

Prozessbegleitendes Erkennen von Werkzeugbruch und Verschleisswertgrenzen, von o. Prof. Dr.-Ing. h.c. W. König; Dipl.-Ing. W. Kluft (aus Industrie-Anzeiger Nr. 96 vom 1.12.82, Verlag W. Girardet, Essen).

Variations des composantes des efforts de cisaillement lors d'une brisure totale; Extrait du tirage à part KISTLER No 20.112e:

Cutting Force Measurements as a Source Data: Sensing of Tool Breakage and Wear Limit Values During Processing, from o. Prof. Dr.-Ing. h.c. W. König , Dipl.-Ing. W. Kluft (de "Industrie-Anzeiger" Nr. 96 du 1.12.82, Verlag W. Girardet, Essen).

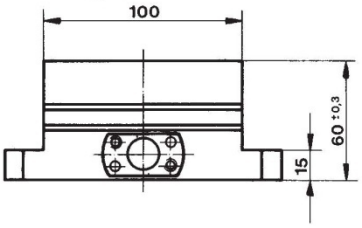
Cutting force components during total break; from KISTLER reprint No 20.112e:

Cutting Force Measurements as a Source Data: Sensing of Tool Breakage and Wear Limit Values During Processing, from o. Prof. Dr.-Ing. h.c. W. König; Dipl.-Ing. W. Kluft (of "Industrie-Anzeiger" No 96, 1.12.1982, Verlag W. Girardet, Essen).

000-151m-02.91 (DB06.9257Bm)

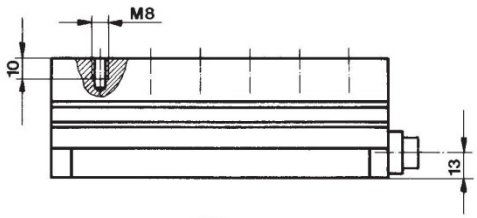
Dynamometer Typ 9257B

Abmessungen



Dynamomètre type 9257B

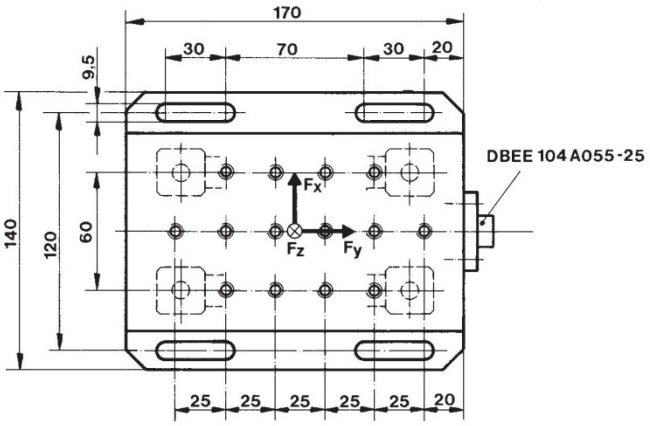
Dimensions



Dynamometer Type 9257B

Dimensions

Fräsen, Schleifen
Fraisage, rectifiage
Milling, grinding



Dynamometer Typ 9257B

Abmessungen mit montiertem Stahlhalter

Stahlhalter	Typ 9403
Anschlusskabel	Typ 1687B5 / 1689B5

Dynamomètre type 9257B

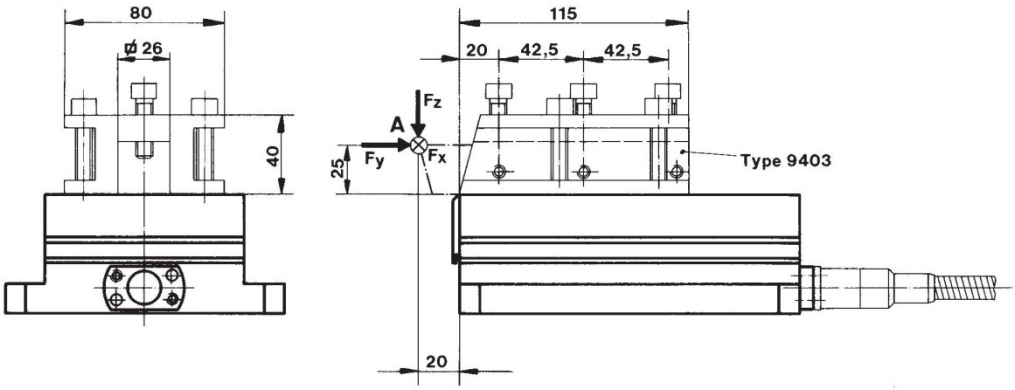
Dimensions avec porte-outil monté

Porte-outil	type 9403
Câble de connexion	type 1687B5 / 1689B5

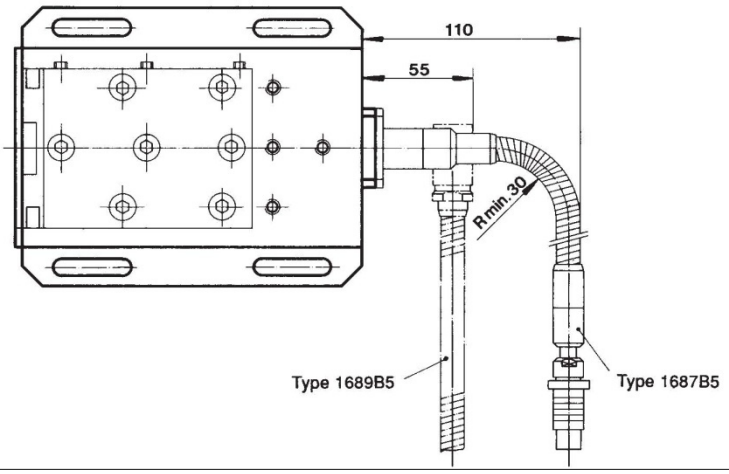
Dynamometer Type 9257B

Dimensions with mounted tool holder

Tool holder	Type 9403
Connecting cable	Type 1687B5 / 1689B5



Drehen
Tournage
Turning



000-151m-02.91 (DB06.9257Bm)

Montage

Das Dynamometer kann mit Schrauben oder Pratzen auf jede plangeschliffene, saubere Montagefläche, wie z.B. auf einen Werkzeugmaschinentisch montiert werden. Es ist zu beachten, dass durch unebene Auflageflächen innere Verspannungen auftreten können, welche die einzelnen Messelemente zusätzlich stark belasten sowie das Übersprechen vergrössern können.

Zum Aufspannen der krafteinleitenden Teile, wie Drehstähle und Werkstücke, stehen in der Deckplatte vierzehn M8x1,25 Sackgewinde zur Verfügung. Die Auflageflächen der krafteinleitenden Teile müssen plangeschliffen sein, damit eine gute mechanische Ankopplung an die Deckplatte erreicht wird.

Für eine einwandfreie Montage von Drehstählen bis zu einem Schaftquerschnitt von 26x26 mm kann der Stahlhalter Typ 9403 verwendet werden.

Der Stahlhalter ist im Lieferumfang nicht enthalten, er muss separat bestellt werden.

Lieferumfang: siehe Preisliste.

Montage

Le dynamomètre peut être fixé au moyen de vis ou de brides sur toute surface plane rectifiée comme p.ex. sur un plateau de machine-outil. Toutes les inégalités ou irrégularités de la surface de montage peuvent avoir pour conséquence des tensions internes engendrant ainsi des sollicitations supplémentaires sur les divers éléments de mesure ainsi qu'un accroissement du cross talk.

La plaque supérieure possède quatorze taraudages borgnes M8x1,25 pour la fixation de la pièce introduisant la force telle que l'outil de coupe ou la pièce à usiner. Les faces d'appui des pièces introduisant l'effort doivent également être planes afin de garantir une liaison mécanique parfaite avec la plaque supérieure.

Pour assurer un montage parfait des outils de coupe jusqu'à une section de 26x26 mm il est recommandé d'utiliser le porte-outil type 9403.

Ce porte-outil ne fait pas partie de la fourniture et doit donc être commandé séparément.

Etendu de la fourniture: voir Prix-Courant.

Mounting

The dynamometer may be mounted with screws or claws on any clean, face-ground supporting surface, such as the table of a machine tool for example. Uneven supporting surface may set up internal stresses, which will impose severe additional loads on the individual measuring elements and may also increase cross talk.

For mounting the force-introducing components, such as lathe tools and workpieces, fourteen M8x1,25 mm blind tap holes in the cover plate are available. The supporting surfaces for the force-introducing parts must be face-ground to obtain good mechanical coupling to the cover plate.

For satisfactory mounting of lathe tools up to 26x26 mm shank cross section, the tool holder Type 9403 may be used.

This holder is not included in the standard accessories and must therefore be ordered separately.

Scope of delivery: see Price List.

Zubehör

Für 3-Komponenten-Kraftmessung
F_x, F_y, F_z

• Stahlhalter	Typ 9403
• Anschlusskabel (3adrig)	Typ 1687B5 Typ 1689B5
• Verlängerungskabel (3adrig)	Typ 1688B5 Typ 1688B10
• Verteilkästchen	Typ 5407A

Für 6-Komponenten-Kraft- und Momentmessung
F_x, F_y, F_z / M_x, M_y, M_z

• Anschlusskabel (8adrig)	Typ 1677A5 Typ 1679A5
• Verlängerungskabel (8adrig)	Typ 1678A5 Typ 1678A10
• Verteilkästchen	Typ 5405A

Accessoires

Pour mesurer de forces à 3 composantes
F_x, F_y, F_z

• Porte-outil	type 9403
• Câble de connexion (3 fils)	type 1687B5 type 1689B5
• Câble de rallonge (3 fils)	type 1688B5 type 1688B10
• Boîtier de distribution	type 5407A

Pour mesurer de forces et moments à 6 composantes
F_x, F_y, F_z / M_x, M_y, M_z

• Câble de connexion (8 fils)	type 1677A5 type 1679A5
• Câble de rallonge (8 fils)	type 1678A5 type 1678A10
• Boîtier de distribution	type 5405A

Accessories

For 3-Component Force Measurements
F_x, F_y, F_z

• Tool holder	Type 9403
• Connecting cable (3 leads)	Type 1687B5 Type 1689B5
• Extension cable (3 leads)	Type 1688B5 Type 1688B10
• Distribution box	Type 5407A

For 6-Component Force and Moment Measurements
F_x, F_y, F_z / M_x, M_y, M_z

• Connecting cable (8 leads)	Type 1677A5 Type 1679A5
• Extension cable (8 leads)	Type 1678A5 Type 1678A10
• Distribution box	Type 5405A

Elektronik

Eine Dreikomponenten-Kraftmessanlage benötigt neben dem Dynamometer noch drei Ladungsverstärker, welche die Ladungssignale des Dynamometers in Ausgangsspannungen umwandeln, die proportional zu den auftretenden Kräften sind.

Electronique

Outre le dynamomètre, une installation de mesure de force à trois composantes comprend encore trois amplificateurs de charge qui transforment les signaux de charge du dynamomètre en tensions de sortie proportionnelles aux forces appliquées.

Electronics

Besides the dynamometer, a three-component force measuring system also needs three charge amplifiers, which convert the dynamometer charge signals into output voltages proportional to the forces sustained.

Mehrkomponenten-Messanlagen

Weitere Einzelheiten betreffend Schnittkraft-Messanlagen
siehe Datenblatt IN6.9255/57/65.

Systèmes pour mesurer à plusieurs composantes

D'autres informations concernant des systèmes pour mesurer les efforts de coupe
voir notice technique IN6.9255/57/65.

Systems for Multicomponent Measurements

Further information concerning systems for cutting force measurements
see Data sheet IN6.9255/57/65.

000-151m-02.91 (DB06.9257Bm)

Measure & Analyze – MCA

KISTLER
measure. analyze. innovate

Multi-Channel Charge Amplifier

for Multi-Component Force Measurement

This instrument is ideal for multi-component force-torque measurement with piezoelectric dynamometers or force plates. Piezoelectric force sensors produce an electric charge which varies in direct proportion with the load acting on the sensor. The charge amplifier then converts the electric charge into a proportional voltage.

4-channel version for cutting force measurements
8-channel version for multi-component force-torque measurement
8-channel version optionally with 6-component analog summing calculator
Menu-controlled operation as with Type 5015A
Direct signal evaluation
Suitable for data acquisition software DynoWare Type 2825A-02

Description
Type 5070A... is available as a 4-channel or 8-channel version. As an option, the 8-channel version can also be provided with a 6-component analog summing calculator. In the case of Kistler multi-component dynamometers, this summing calcula-tor calculates in real time mode the resulting force as well as the three components of the resulting torque vector. Dynamometer-specific values required for torque calculation can be set directly on the instrument.

The graphics-capable liquid crystal display shows all settings including the instantaneous, minimum and maximum values of a charge amplifier channel. The various channels can be switched onto the display as required. The instrument is set up by means of various menus with the universal press-and-turn knob. All functions can, however, also be controlled externally via RS-232C (optionally IEEE-488).

Application
The 4-channel instrument is particularly suitable for cutting force measurement with Kistler dynamometers and the data acquisition software DynoWare Type 2825A-02. The 8-channel instrument is suitable for 6-component force-torque measure-ment in the laboratory as well as in research and development. For example, wheel force measurement on a tire test stand, reaction force measurements on engine-transmission units, monitoring of forces and torques in vibration tests etc.


KISTLER

Technical Data

Charge Amplifier

Number of channels		4
Option		8
Connector type		BNC neg.
Option		Fischer 9-pole neg.
Measuring range FS	pC	±200 … 200 000
Option	pC	±600 … 600 000
Error (0 … 50 °C) typ./max.	%	<±0,3/<±1
Drift, measurement type DC/long		
at 25 °C	pC/s	<±0,05
at 50 °C	pC/s	<±0,2
Frequency range (20 Vpp)	kHz	≈0 … >45

Voltage Output

Connector type		D-Sub 15f
Output voltage	V	±10
Output current	mA	<±2
Output resistance	Ω	10
Reset-measure transition	pC	<±2
Zero point error (Reset)	mV	<±10
Output interference signal (0,1 Hz … 1 MHz)	mVpp	<10

Low-Pass Filter

Order		2
Cutoff frequency (–3 dB)	Hz	100, 300, 600, 1 000, 2 000
Error	%	<±5

KISTLER
measure. analyze. innovate.

High-Pass Filter

Zero point error	mV	<±15
Time constant		
Range 200 ... 200 000 pC		
200 ... 6 269 pC	s	10
6 270 ... 200 000 pC	s	340
Time constant		
Range 600 ... 600 000 pC		
600 ... 18 809 pC	s	33
18 810 ... 600 000 pC	s	1 023
Error (time constant)	%	<±20

Signal Evaluation

Measurand renewal		
Instantaneous value	ms	300
Minimum value	ms	300
Maximum value	ms	300
Bar display	ms	50

Summing Calculator (Option)

Specifications are valid incl. charge amplifier

Number of summation outputs		6
Error (0 ... 50 °C) typ./max.	%	<±0,5/<±1
Output voltage	V	±10
Output current (short-circuit proof)	mA	±2
Output resistance	Ω	10
Zero point error (Reset)	mV	<±10
Output interference signal (0,1 Hz ... 1 MHz)	mVpp	<10
Frequency range (20 Vpp)	kHz	≈0 ... >45

RS-232C Interface

Standard		RS-232C (V.24)
Connector type		D-Sub 9f
Pin allocation		
Pin 2		R X D
Pin 3		T X D
Pin 5		GND RS
Max. input voltage, continuous	V	±20
Max. voltage between signal ground and protective ground	V_{RMS}	<20
Baud rates		1 200/9 600/ 19 200/ 38 400/ 57 600/115 200
Data bit		8
Stop bit		1
Parity		none

IEEE-488 Interface (Option)

Standard		IEEE-488.1-1987
Connector type		Microribbon Series 57, (24-pole)
Interface functions		SH1, AH1, L4, LE0, T6, TE0, SR1, RL2, PP0, DC1, DT1, C0, E1
Uniline commands		IFC, REN, EOI, SRQ, ATN
Multiline commands		DCL, SDC, GET, UNL, UNT, SPE, SPD
Address range		0 ... 30

Remote Control

(Digital input and 24 V supply)

Remote measure and trigger with 10 kΩ pullup to +5 V

Connector type		D-Sub 9f
Input level		
High (Reset, Stop trigger)	V	>3,5
or		Input open
Low (Measure, Start trigger)	V/mA	<1/<4
Max. input voltage	V	±30
Supply (output)	V DC	+24/±20 %
Output current (short-circuit proof)	mA	<200

Mains Connection

Mains connector type (2P + E, Protective Class I)	Type	IEC 320C14
Voltage	VAC	100 ... 240
Voltage tolerance	%	±10
Mains frequency	Hz	50 ... 60
Power consumption	VA	20
Voltage between signal ground and protective ground	V_{RMS}	<50

Further Technical Data

Degree of protection IEC60529 (DIN40050)	IP	40
Operating temperature	°C	0 ... 50
Storage temperature	°C	–10 ... 60
Relative air humidity non-condensing	%	<80
Vibration resistance (20 Hz ... 2 kHz, duration 16 min., cycle 2 min.)	g	<10
Shock resistance (1 ms)	g	<200
Case dimensions		
without frame (WxHxD)	mm	213,4x128,7x230
with frame (WxHxD) (Option)	mm	247,5x142x253,15
Front panel according to DIN 41494, Part 5	HE/TE	3/42
Weight	kg	3,8

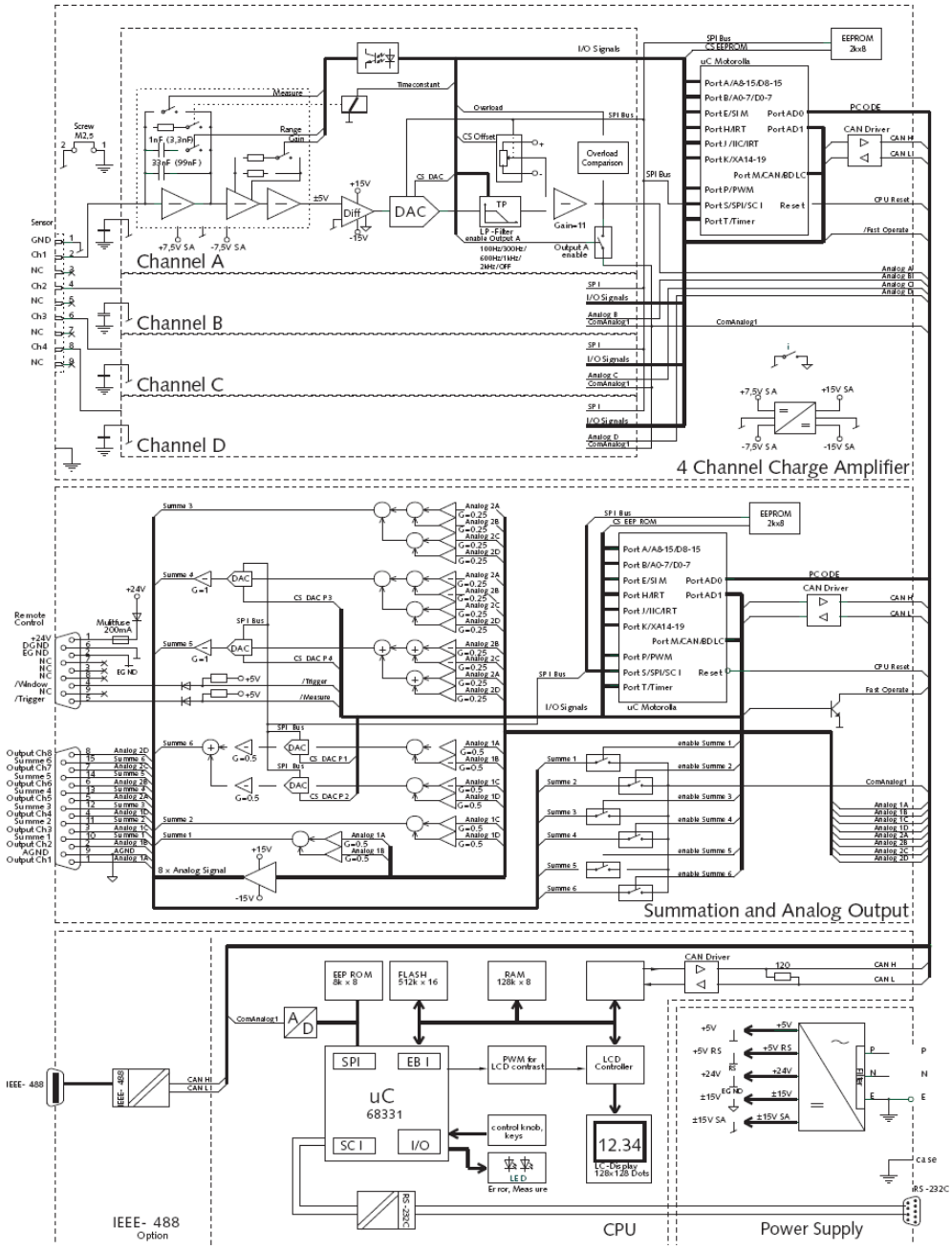


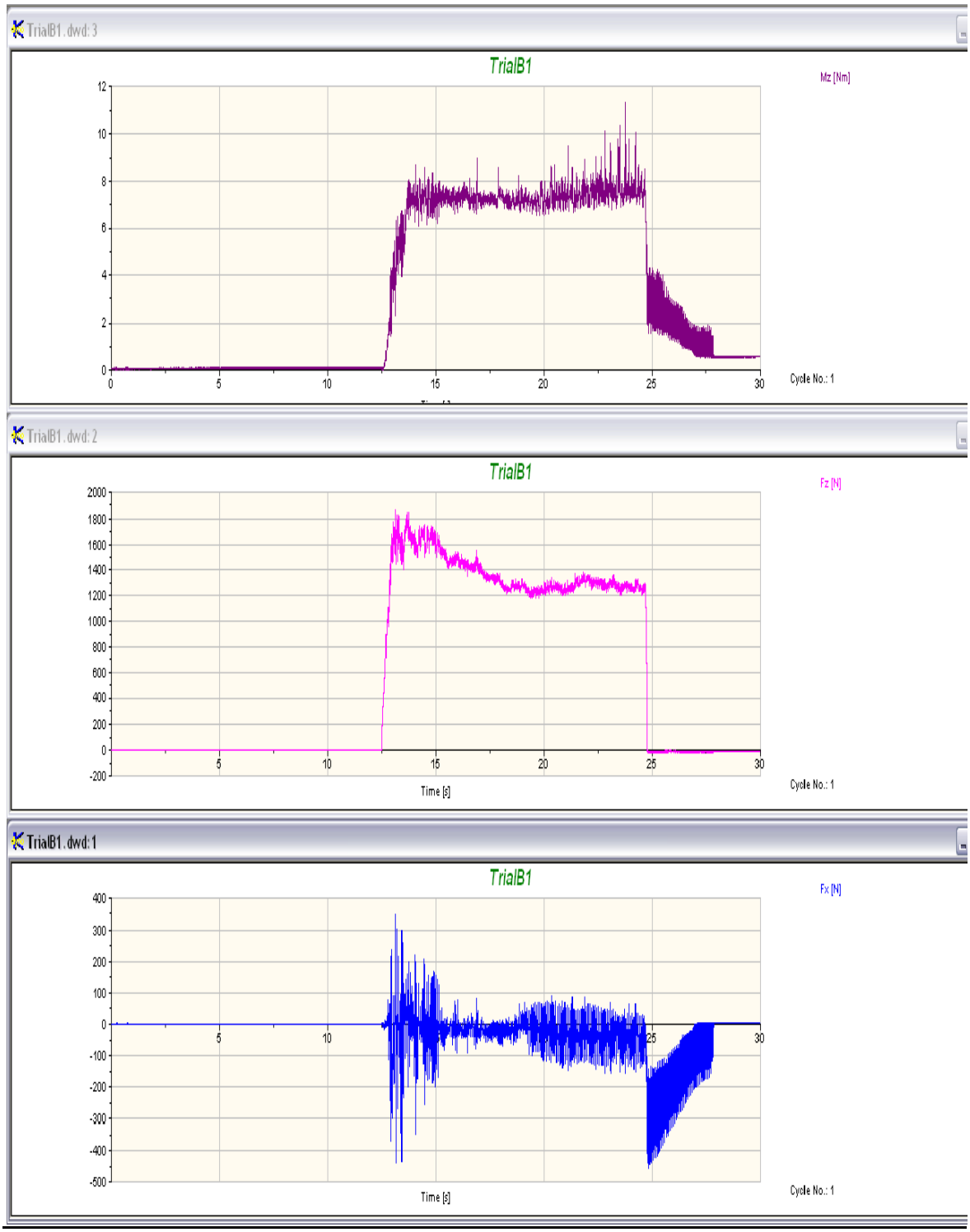
Fig. 1: Block schematic diagram Type 5070A…

Appendix B

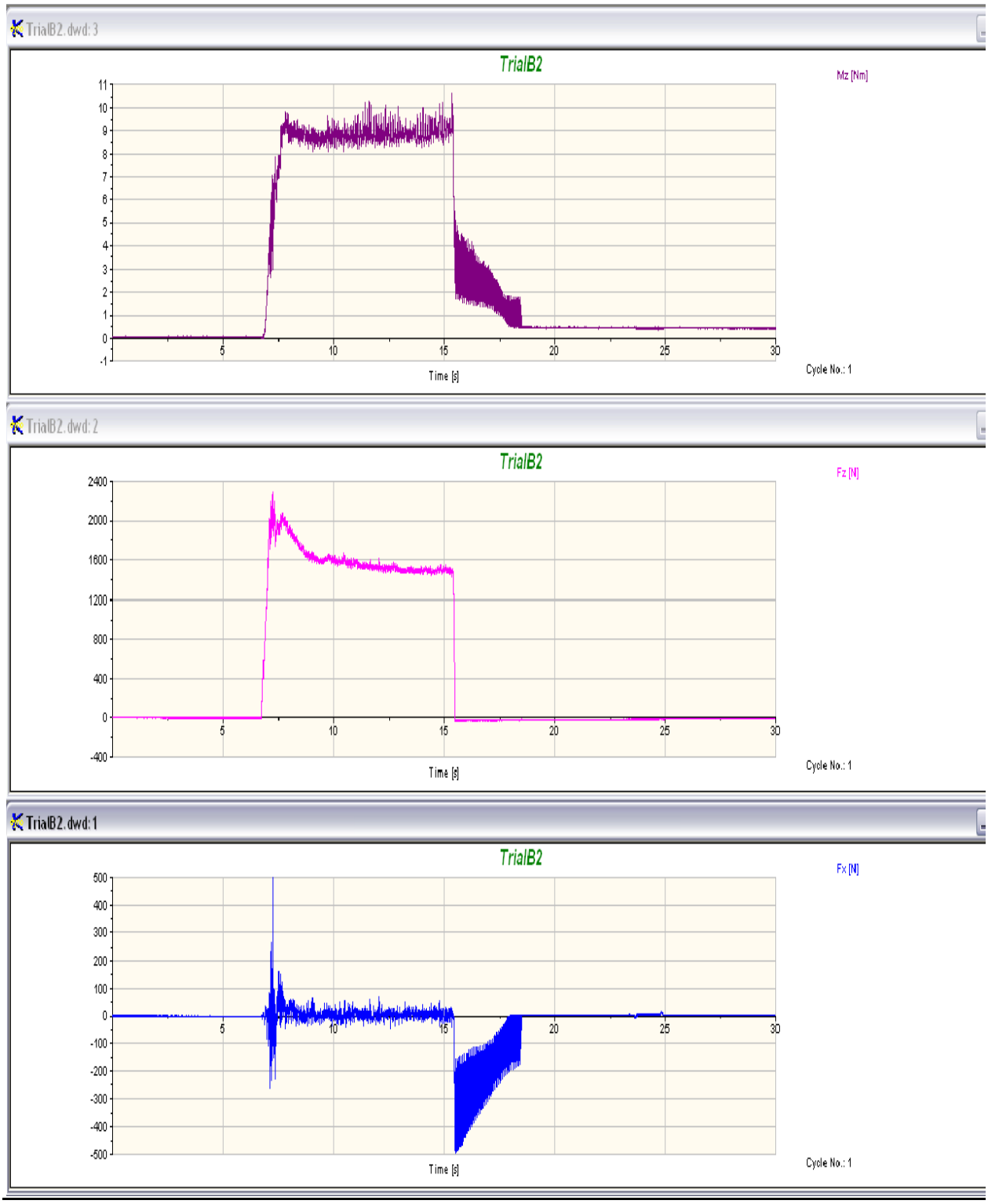
Real time captured forces in the X, Y & Z directions

Kurtosis Values

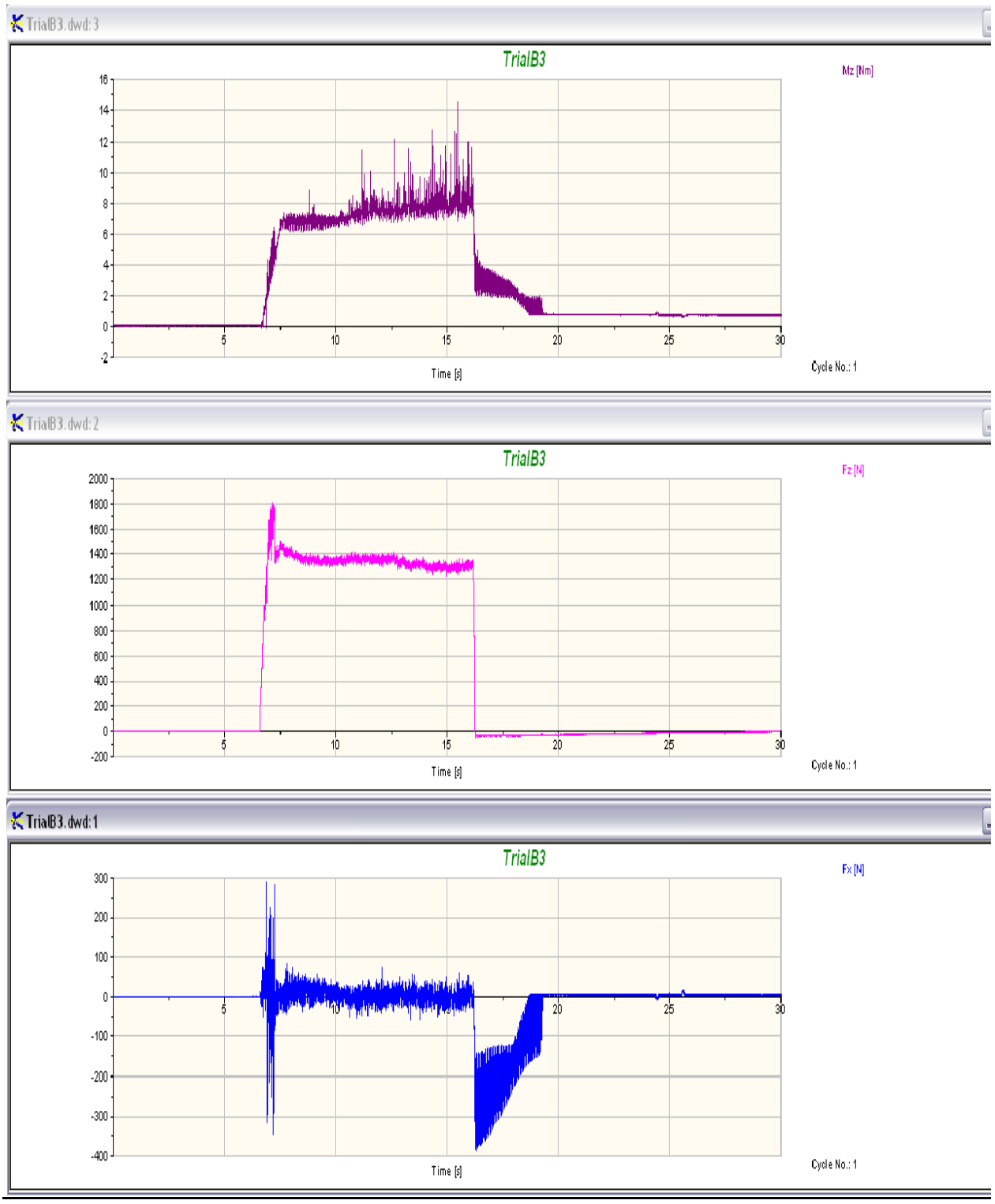
Trial 1 with New Drill



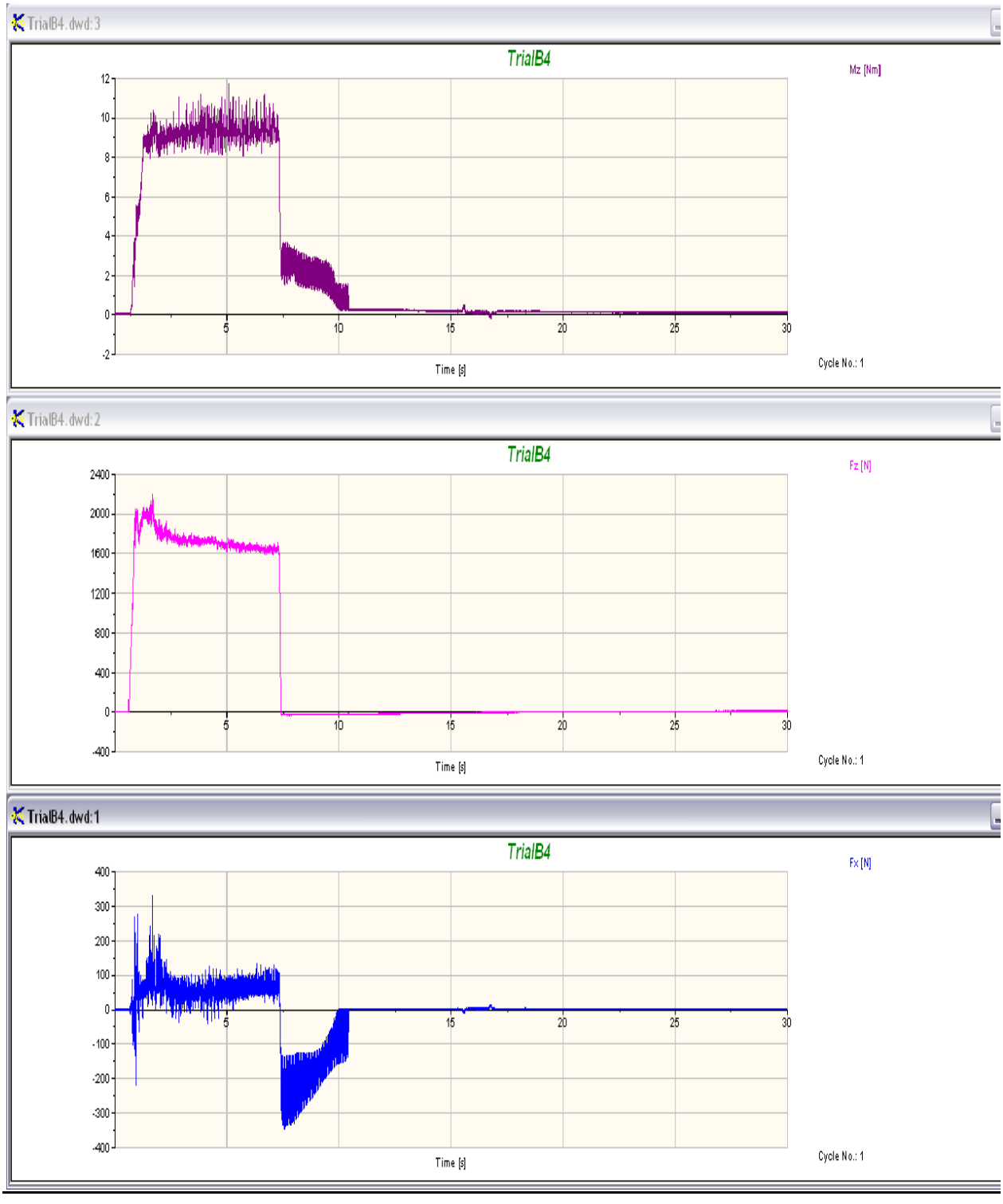
Trial 2 with New Drill



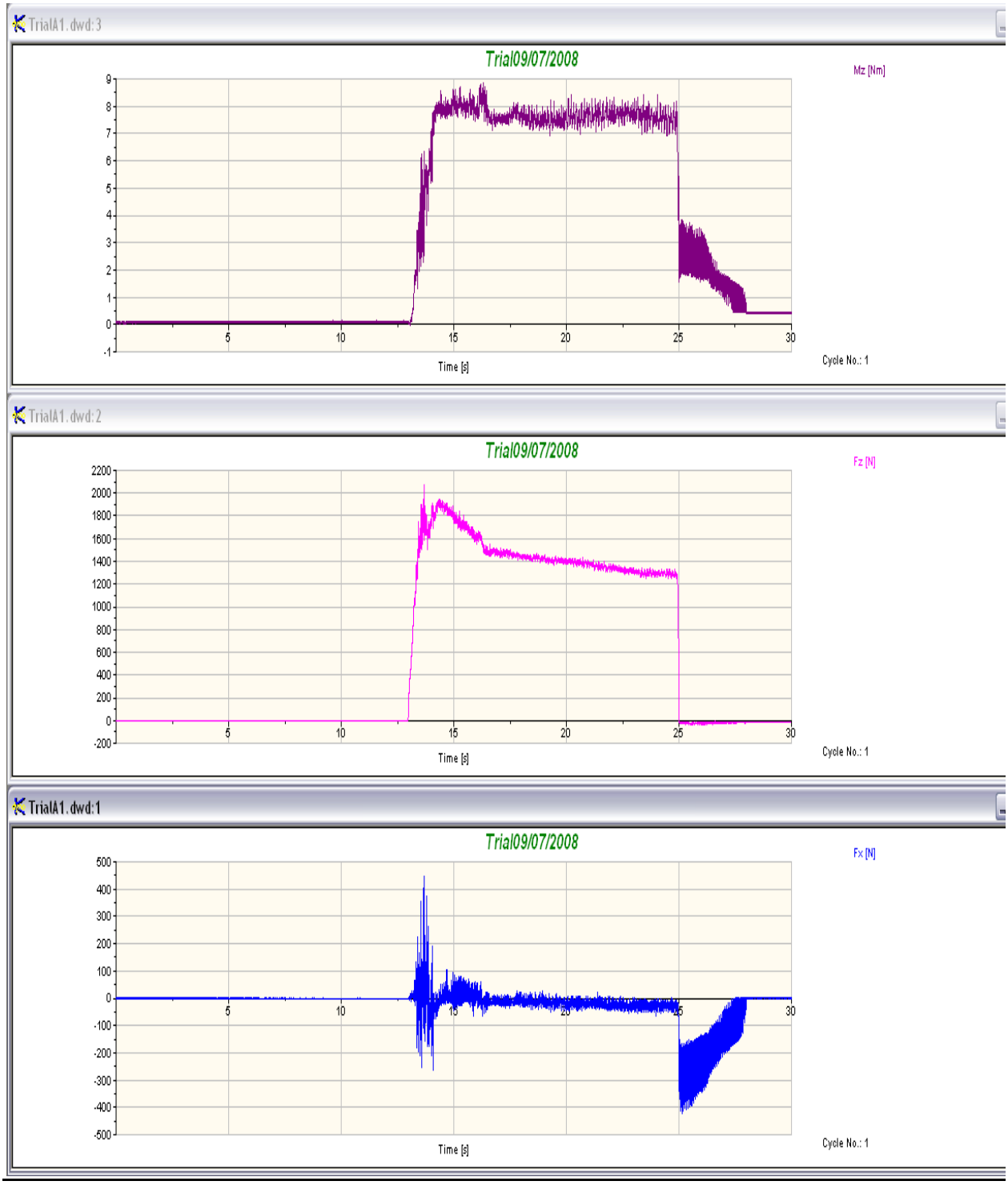
Trial 3 with New Drill


TrialB3.dwd: 3
TrialB3
Mz [Nm]
Time [s]
Cycle No.: 1
TrialB3.dwd: 2
TrialB3
Fz [N]
Time [s]
Cycle No.: 1
TrialB3.dwd: 1
TrialB3
Fx [N]
Time [s]
Cycle No.: 1

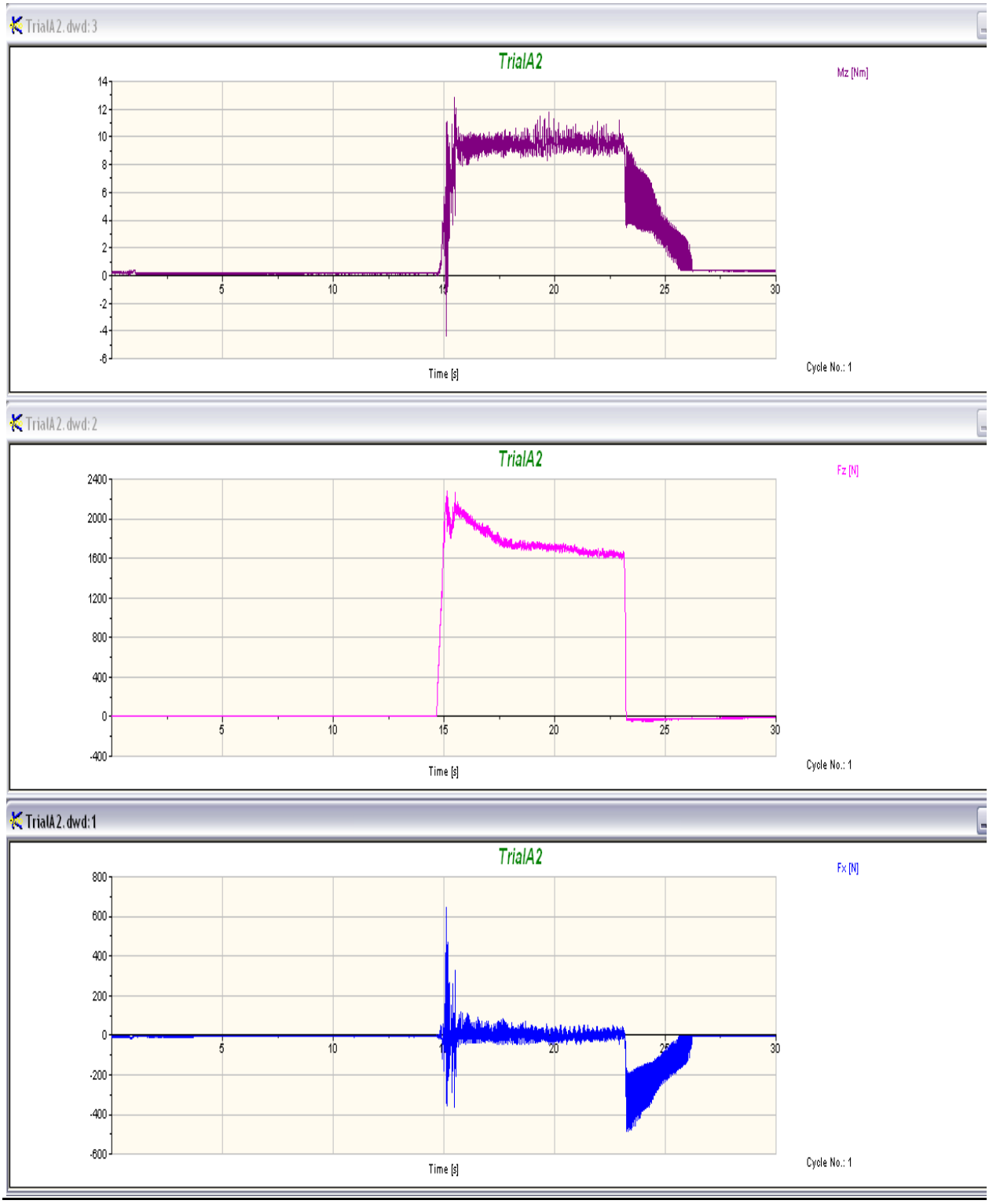
Trial 4 with New Drill



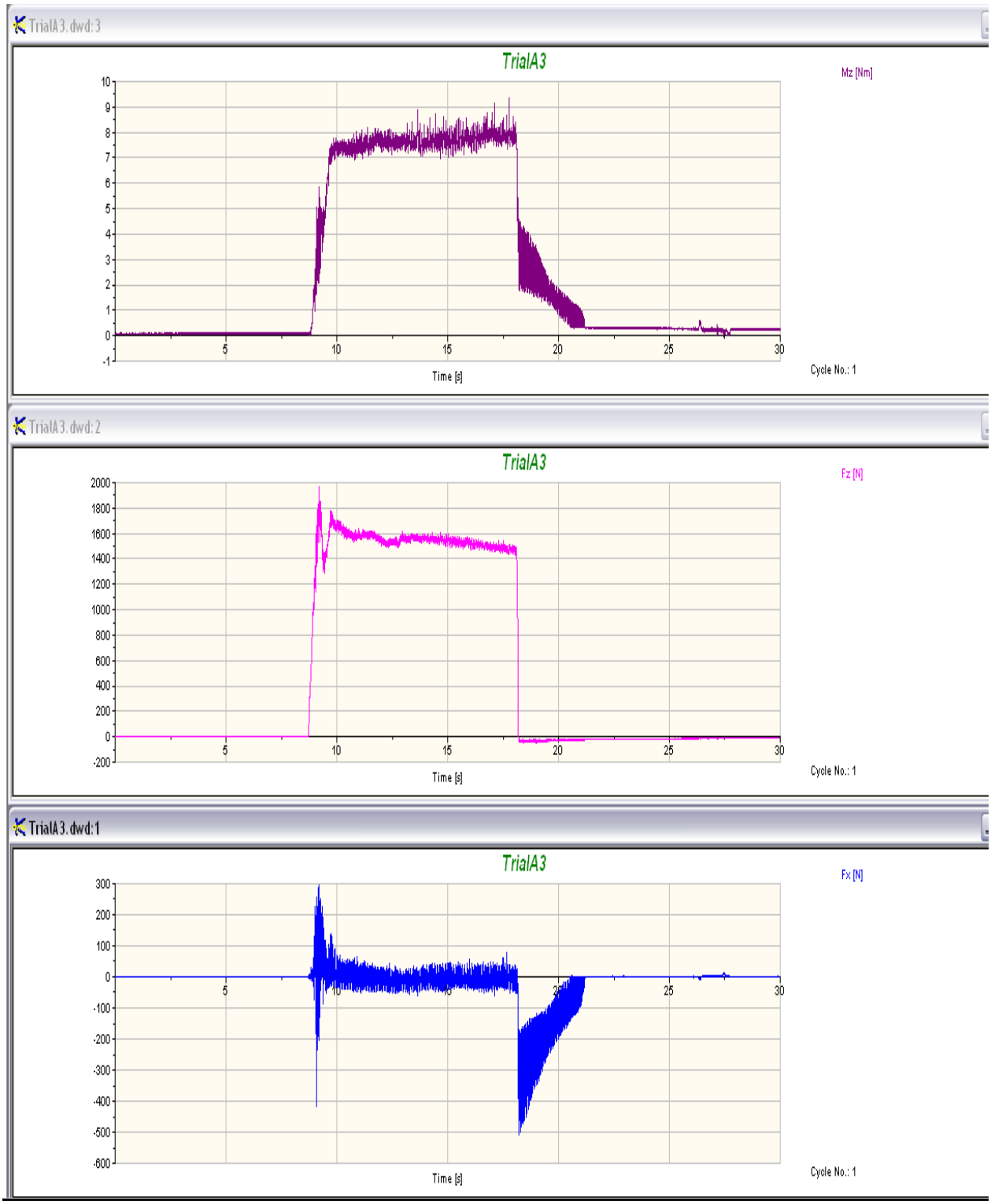
Trial 1 with Old Drill



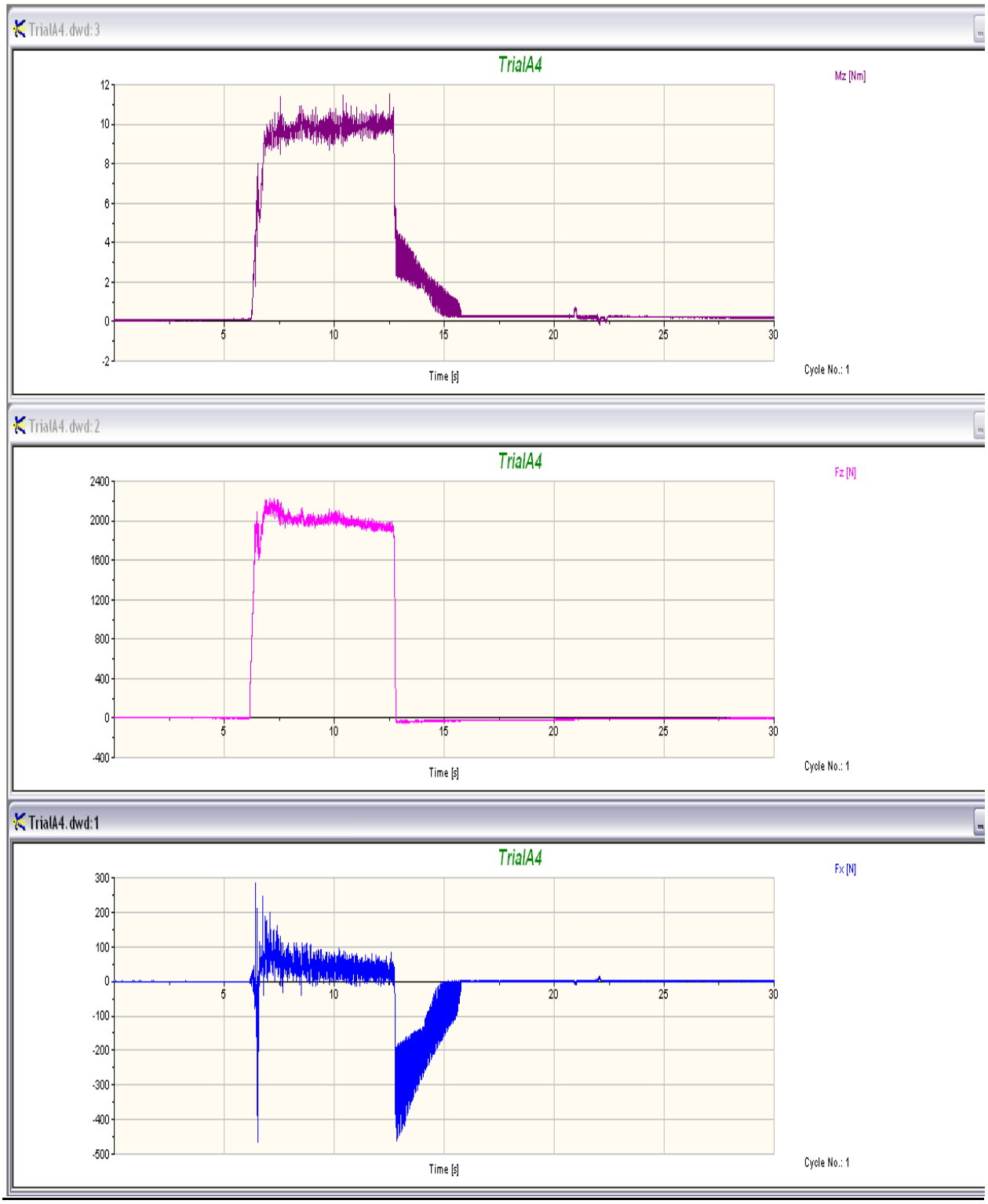
Trial 2 with Old Drill



Trial 3 with Old Drill


TrialA3.dwd:3
TrialA3
Mz [Nm]
Time [s]
Cycle No.: 1
TrialA3.dwd:2
TrialA3
Fz [N]
Time [s]
Cycle No.: 1
TrialA3.dwd:1
TrialA3
Fx [N]
Time [s]
Cycle No.: 1

Trial 4 with Old Drill



Kurtosis Calcualtions

New Drill

Fx				Fz			
B1	B2	B3	B4	B1	B2	B3	B4
1.234161	5.774268	3.0212	1.386911	-0.19991	0.084146	-0.39431	1.569302
-0.35883	-0.16224	-0.59805	3.644684	-1.02704	-1.08661	-0.62956	2.042835
2.009547	0.751534	-0.83997	-0.54127	0.327316	-0.55991	-0.05622	0.88213
2.746289	0.117009	-0.37014	0.44593	-0.03307	-0.2214	-0.13927	0.878356
-0.60905	0.398038	-0.27756	-0.76693	-1.35327	-0.29	-0.24862	-0.30025
-0.00703	1.871876	0.513207	-0.72718	-0.57236	1.851851	-0.49212	0.28965
-1.02292	0.89099	-0.22082	0.038277	-0.15919	1.389957	-0.09674	-0.65727
-1.03085	-0.27271	-0.32932	-1.1915	0.734978	-0.0286	-0.04756	-0.03699
-0.982	0.556317	-0.25524	-0.40055	-0.26771	-0.91558	0.064098	-0.50348
-0.81289	-1.14224	-0.41631	-1.3815	-0.34726	-0.26519	-1.12224	3.859846
-0.8854	-0.83523	-1.05064	0.067225	-0.22099	0.013078	0.141593	-0.32809
0.728332	-1.37425	-0.27478	0.054967	-0.66521	2.335223	-0.1063	-0.56187

Old Drill

Fx				Fz			
A1	A2	A3	A4	A1	A2	A3	A4
1.135855	2.992714	3.617506	7.056242	0.95521	0.895487	0.363737	0.79278
-0.08088	-0.80606	0.695667	0.347392	-0.8616	-0.70452	-0.32244	-0.81905
-0.46805	-0.82886	-1.02302	-0.41282	-0.85125	-0.3514	-0.28051	0.483402
0.63835	-0.81705	-0.93672	-0.63098	1.63802	0.421371	-0.41189	0.002822
-0.73314	-1.01344	-0.54452	-0.76082	-0.20957	-0.10657	0.272743	-0.04375
-0.51683	-0.77046	-0.86683	-0.89071	-0.34247	-0.67823	0.02816	-0.4884
-0.51933	-0.64479	-0.92363	-0.79516	-0.25302	0.18785	-0.38202	-1.58427
-0.78087	-0.72107	-0.79784	-0.64703	-0.26497	0.300043	-0.38025	-0.02196
-0.89263	-0.87309	-0.60922	-0.81085	-0.02879	0.40476	-0.4298	-0.22537
-0.75525	-0.86335	-1.29362	-0.43876	-0.15174	-0.3469	-1.92008	-0.07072
-0.825	-1.07997	-0.97196	0.223145	-0.41403	-0.63003	-0.12475	-0.17425
0.316232	8.686706	-0.88032	0.454731	-1.05269	8.6785	-0.3286	-0.43116

Appendix C

Self-Organising Map Algorithm Program Listing

Function SOM:

```
function (data, grid = somgrid(), rlen = 100, alpha = c(0.05,
    0.01), radius = quantile(nhbrdist, 0.67) * c(1, -1), init,
    toroidal = FALSE, n.hood, keep.data = TRUE)
{
    if (!is.numeric(data))
        stop("Argument data should be numeric")
    data <- as.matrix(data)
    nd <- nrow(data)
    ng <- nrow(grid$pts)
    if (missing(init)) {
        init <- data[sample(1:nd, ng, replace = FALSE), , drop = FALSE]
    }
    else {
        init <- as.matrix(init)
        if (nrow(init) != ng | ncol(init) != ncol(data) | !is.numeric(init))
            stop("incorrect init matrix supplied")
    }
    codes <- init
    if (missing(n.hood)) {
        n.hood <- switch(grid$topo, hexagonal = "circular", rectangular = "square")
    }
    else {
        n.hood <- match.arg(n.hood, c("circular", "square"))
    }
    grid$n.hood <- n.hood
    nhbrdist <- unit.distances(grid, toroidal)
    if (length(radius) == 1)
```

```
        radius <- sort(radius * c(1, -1), decreasing = TRUE)
    changes <- rep(0, rlen)
    res <- .C("SOM_online", data = as.double(data), codes = as.double(codes),
        nhbrdist = as.double(nhbrdist), alpha = as.double(alpha),
        radii = as.double(radius), changes = as.double(changes),
        n = as.integer(nrow(data)), p = as.integer(ncol(data)),
        ncodes = as.integer(nrow(init)), rlen = as.integer(rlen),
        PACKAGE = "kohonen")
    changes <- matrix(res$changes, ncol = 1)
    codes <- res$codes
    dim(codes) <- dim(init)
    colnames(codes) <- colnames(init)
    if (keep.data) {
        mapping <- map.kohonen(list(codes = codes), newdata = data)
        structure(list(data = data, grid = grid, codes = codes,
            changes = changes, alpha = alpha, radius = radius,
            toroidal = toroidal, unit.classif = mapping$unit.classif,
            distances = mapping$distances, method = "som"), class = "kohonen")
    }
    else {
        structure(list(grid = grid, codes = codes, changes = changes,
            alpha = alpha, radius = radius, toroidal = toroidal,
            method = "som"), class = "kohonen")
    }
}
```

Function to initialise:

```
function (data, Y, grid = somgrid(), rlen = 100, alpha = c(0.05,
    0.01), radius = quantile(nhbrdist, 0.67) * c(1, -1), xweight = 0.75,
    contin = !(all(rowSums(Y) == 1)), toroidal = FALSE, n.hood,
```

```
    keep.data = TRUE)
{
    if (!is.numeric(data))
        stop("Argument data should be numeric")
    data <- as.matrix(data)
    nd <- nrow(data)
    nx <- ncol(data)
    if (is.vector(Y))
        Y <- matrix(Y, ncol = 1)
    ny <- ncol(Y)
    ng <- nrow(grid$pts)
    xdists <- ydists <- rep(0, ng)
    starters <- sample(1:nd, ng, replace = FALSE)
    init <- data[starters, , drop = FALSE]
    codes <- init
    if (!contin) {
        codeYs <- 0.5 + 0.5 * (Y[starters, ] - 0.5)
    }
    else {
        codeYs <- Y[starters, ]
    }
    if (missing(n.hood)) {
        n.hood <- switch(grid$topo, hexagonal = "circular", rectangular = "square")
    }
    else {
        n.hood <- match.arg(n.hood, c("circular", "square"))
    }
    grid$n.hood <- n.hood
    nhbrdist <- unit.distances(grid, toroidal)
    if (length(radius) == 1)
```

```
    radius <- sort(radius * c(1, -1), decreasing = TRUE)
changes <- rep(0, rlen * 2)
if (contin) {
    res <- .C("BDK_Eucl", data = as.double(data), Ys = as.double(Y),
        codes = as.double(codes), codeYs = as.double(codeYs),
        nhbrdist = as.double(nhbrdist), alpha = as.double(alpha),
        radii = as.double(radius), xweight = as.double(xweight),
        changes = as.double(changes), xdists = as.double(xdists),
        ydists = as.double(ydists), n = as.integer(nd), px = as.integer(nx),
        py = as.integer(ny), ncodes = as.integer(ng), rlen = as.integer(rlen),
        PACKAGE = "kohonen")
}
else {
    res <- .C("BDK_Tani", data = as.double(data), Ys = as.double(Y),
        codes = as.double(codes), codeYs = as.double(codeYs),
        nhbrdist = as.double(nhbrdist), alpha = as.double(alpha),
        radius = as.double(radius), xweight = as.double(xweight),
        changes = as.double(changes), xdists = as.double(xdists),
        ydists = as.double(ydists), n = as.integer(nd), px = as.integer(nx),
        py = as.integer(ny), ncodes = as.integer(ng), rlen = as.integer(rlen),
        PACKAGE = "kohonen")
}
changes <- matrix(res$changes, ncol = 2)
codes <- list(X = matrix(res$codes, nrow(init), ncol(init)),
    Y = matrix(res$codeYs, ng, ny))
colnames(codes$Y) <- colnames(Y)
if (keep.data) {
    mapping <- map.kohonen(list(codes = codes), newdata = data,
        whatmap = 1)
    structure(list(data = data, Y = Y, contin = contin, grid = grid,
```

```
            codes = codes, changes = changes, alpha = alpha,
            radius = radius, toroidal = toroidal, unit.classif = mapping$unit.classif,
            distances = mapping$distances, method = "bdk"), class = "kohonen")
    }
    else {
        structure(list(contin = contin, grid = grid, codes = codes,
            changes = changes, alpha = alpha, radius = radius,
            toroidal = toroidal, method = "bdk"), class = "kohonen")
    }
}
```

Function for mapping:

```
function (xdim = 8, ydim = 6, topo = c("rectangular", "hexagonal"))
{
    topo <- match.arg(topo)
    x <- 1:xdim
    y <- 1:ydim
    pts <- as.matrix(expand.grid(x = x, y = y))
    if (topo == "hexagonal") {
        pts[, 1] <- pts[, 1] + 0.5 * (pts[, 2]%%2)
        pts[, 2] <- sqrt(3)/2 * pts[, 2]
    }
    res <- list(pts = pts, xdim = xdim, ydim = ydim, topo = topo)
    class(res) <- "somgrid"
    res
}
<environment: namespace:class>
```

Function for training:

```
function (data, Y, grid = somgrid(), rlen = 100, alpha = c(0.05,
```

```
    0.01), radius = quantile(nhbrdist, 0.67) * c(1, -1), xweight = 0.5,
    contin = !(all(rowSums(Y) == 1)), toroidal = FALSE, n.hood,
    keep.data = TRUE)
{
    if (!is.numeric(data))
        stop("Argument data should be numeric")
    data <- as.matrix(data)
    nd <- nrow(data)
    nx <- ncol(data)
    if (is.vector(Y))
        Y <- matrix(Y, ncol = 1)
    ny <- ncol(Y)
    ng <- nrow(grid$pts)
    xdists <- ydists <- rep(0, ng)
    starters <- sample(1:nd, ng, replace = FALSE)
    init <- data[starters, , drop = FALSE]
    codes <- init
    if (!contin) {
        codeYs <- 0.5 + 0.5 * (Y[starters, ] - 0.5)
    }
    else {
        codeYs <- Y[starters, ]
    }
    if (missing(n.hood)) {
        n.hood <- switch(grid$topo, hexagonal = "circular", rectangular = "square")
    }
    else {
        n.hood <- match.arg(n.hood, c("circular", "square"))
    }
    grid$n.hood <- n.hood
```

```
nhbrdist <- unit.distances(grid, toroidal)
if (length(radius) == 1)
    radius <- sort(radius * c(1, -1), decreasing = TRUE)
changes <- rep(0, rlen * 2)
if (contin) {
    res <- .C("XYF_Eucl", data = as.double(data), Ys = as.double(Y),
        codes = as.double(codes), codeYs = as.double(codeYs),
        nhbrdist = as.double(nhbrdist), alpha = as.double(alpha),
        radii = as.double(radius), xweight = as.double(xweight),
        changes = as.double(changes), xdists = as.double(xdists),
        ydists = as.double(ydists), n = as.integer(nd), px = as.integer(nx),
        py = as.integer(ny), ncodes = as.integer(ng), rlen = as.integer(rlen),
        PACKAGE = "kohonen")
}
else {
    res <- .C("XYF_Tani", data = as.double(data), Ys = as.double(Y),
        codes = as.double(codes), codeYs = as.double(codeYs),
        nhbrdist = as.double(nhbrdist), alpha = as.double(alpha),
        radius = as.double(radius), xweight = as.double(xweight),
        changes = as.double(changes), xdists = as.double(xdists),
        ydists = as.double(ydists), n = as.integer(nd), px = as.integer(nx),
        py = as.integer(ny), ncodes = as.integer(ng), rlen = as.integer(rlen),
        PACKAGE = "kohonen")
}
changes <- matrix(res$changes, ncol = 2)
colnames(changes) <- c("X", "Y")
codes <- list(X = matrix(res$codes, nrow(init), ncol(init)),
    Y = matrix(res$codeYs, ng, ny))
colnames(codes$Y) <- colnames(Y)
if (keep.data) {
```

```
        mapping <- map.kohonen(list(codes = codes), newdata = data,
            whatmap = 1)
        structure(list(data = data, Y = Y, contin = contin, grid = grid,
            codes = codes, changes = changes, alpha = alpha,
            radius = radius, toroidal = toroidal, unit.classif = mapping$unit.classif,
            distances = mapping$distances, method = "xyf"), class = "kohonen")
    }
    else {
        structure(list(contin = contin, grid = grid, codes = codes,
            changes = changes, alpha = alpha, radius = radius,
            toroidal = toroidal, method = "xyf"), class = "kohonen")
    }
}
```

Function for predicting:

```
function (object, newdata, trainX, trainY, unit.predictions = NULL,
    threshold = 0, whatmap = NULL, weights = 1, ...)
{
    mapping <- NULL
    if (missing(newdata)) {
        if (!is.null(object$data)) {
            newdata <- object$data
            mapping <- object$unit.classif
        }
        else {
            stop("No data given with which to predict")
        }
    }
    if (is.null(mapping))
        mapping <- map(object, newdata, whatmap, weights)$unit.classif
```

```
if (is.null(unit.predictions)) {
    if (object$method %in% c("xyf", "bdk")) {
        unit.predictions <- object$codes$Y
    }
    else {
        if (object$method == "supersom" & !is.null(whatmap)) {
            whatmap <- check.whatmap(object, whatmap)
            if (length(whatmap) < length(object$data))
              unit.predictions <- object$codes[-whatmap]
        }
        else {
            if (missing(trainY))
              stop("For unsupervised forms of mapping, trainY is required")
            if (is.list(trainY))
              stop("Prediction for trainY lists not implemented")
            if (is.vector(trainY))
              trainY <- matrix(trainY, ncol = 1)
            nY <- ncol(trainY)
            trainingMapping <- NULL
            if (missing(trainX) & !is.null(object$data)) {
              trainX <- object$data
              trainingMapping <- object$unit.classif
            }
            nX <- ifelse(is.list(trainX), nrow(trainX[[1]]),
              nrow(trainX))
            if (nX != nrow(trainY))
              stop("Unequal number of rows in trainX and trainY")
            if (is.null(trainingMapping))
              trainingMapping <- map(object, trainX)$unit.classif
            unit.predictions <- matrix(NA, nrow(object$grid$pts),
```

```
            nY)
            huhn <- aggregate(trainY, by = list(cl = trainingMapping),
              mean)
            if (R.version$major <= "2" & R.version$minor <
              "6.0") {
              unit.predictions[sort(as.numeric(levels(huhn[,
                1]))), ] <- as.matrix(huhn[, -1])
            }
            else {
              unit.predictions[huhn[, 1], ] <- as.matrix(huhn[,
                -1])
            }
            nas <- which(apply(unit.predictions, 1, function(x) all(is.na(x))))
            nhbrdist <- unit.distances(object$grid, object$toroidal)
            for (i in seq(along = nas)) {
              unit.predictions[nas[i], ] <- colMeans(unit.predictions[nhbrdist[nas[i],
                ] == 1, , drop = FALSE], na.rm = TRUE)
            }
            colnames(unit.predictions) <- colnames(trainY)
          }
        }
    }
    if (!is.null(object$contin) && !object$contin) {
        prediction <- classmat2classvec(unit.predictions, threshold =
threshold)[mapping]
    }
    else {
        if (is.list(unit.predictions)) {
            prediction <- sapply(unit.predictions, function(x) x[mapping])
        }
        else {
```

```
            prediction <- unit.predictions[mapping, ]
        }
    }
    list(prediction = prediction, unit.classif = mapping, unit.predictions =
unit.predictions)
}
```

Function for plotting:

```
function (x, type = c("codes", "changes", "counts", "mapping",
    "property", "quality"), classif = NULL, labels = NULL, pchs = NULL,
    main = NULL, palette.name = heat.colors, ncolors, bgcol = NULL,
    zlim = NULL, heatkey = TRUE, property, contin, whatmap = NULL,
    codeRendering = NULL, keepMargins = FALSE, ...)
{
    type <- match.arg(type)
    switch(type, mapping = plot.kohmapping(x, classif, main,
        labels, pchs, bgcol, keepMargins, ...), property = plot.kohprop(x,
        property, main, palette.name, ncolors, zlim, heatkey,
        contin, keepMargins, ...), codes = plot.kohcodes(x, main,
        bgcol, whatmap, codeRendering, keepMargins, ...), quality =
plot.kohquality(x,
        classif, main, palette.name, ncolors, zlim, heatkey,
        keepMargins, ...), counts = plot.kohcounts(x, classif,
        main, palette.name, ncolors, zlim, heatkey, keepMargins,
        ...), changes = plot.kohchanges(x, main, keepMargins,
        ...))
    invisible()
}
function (x, main, keepMargins, ...)
{
```

```
    if (is.null(main))
        main <- "Training progress"
    nmaps <- ncol(x$changes)
    if (nmaps > 1) {
        if (!is.null(colnames(x$changes))) {
            varnames <- colnames(x$changes)
        }
        else {
            varnames <- paste("Matrix", 1:ncol(x$changes))
        }
    }
    if (nmaps == 2) {
        if (!keepMargins) {
            opar <- par("mar")
            on.exit(par(mar = opar))
        }
        par(mar = c(5.1, 4.1, 4.1, 4.1))
        huhn <- x$changes
        huhn[, 2] <- max(x$changes[, 1]) * huhn[, 2]/max(x$changes[,
            2])
        ticks <- pretty(x$changes[, 2], length(axTicks(2)))
    }
    else {
        huhn <- x$changes
    }
    matplot(huhn, type = "l", lty = 1, main = main, ylab = "Mean distance to
closest unit",
        xlab = "Iteration", ...)
    abline(h = 0, col = "gray")
    if (nmaps == 2)
        axis(4, col.axis = 2, at = ticks * max(x$changes[, 1])/max(x$changes[,
```

```
            2]), labels = ticks)
    if (nmaps > 1)
        legend("topright", legend = varnames, lty = 1, col = 1:nmaps,
            bty = "n")
}
```