An advanced real-time predictive maintenance framework for large scale machine systems

Dheeraj Bansal
Doctor of Philosophy

ASTON UNIVERSITY

The University of Aston in Birmingham

November 2005

Aston University

An advanced real-time predictive maintenance framework for large scale machine systems

Dheeraj Bansal

Doctor of Philosophy

November, 2005

This thesis introduces and develops a novel real-time predictive maintenance system to estimate the machine system parameters using the motion current signature.

Recently, motion current signature analysis has been addressed as an alternative to the use of sensors for monitoring internal faults of a motor. A maintenance system based upon the analysis of motion current signature avoids the need for the implementation and maintenance of expensive motion sensing technology. By developing nonlinear dynamical analysis for motion current signature, the research described in this thesis implements a novel real-time predictive maintenance system for current and future manufacturing machine systems.

A crucial concept underpinning this project is that the motion current signature contains information relating to the machine system parameters and that this information can be extracted using nonlinear mapping techniques, such as neural networks. Towards this end, a proof of concept procedure is performed, which substantiates this concept. A simulation model, TuneLearn, is developed to simulate the large amount of training data required by the neural network approach. Statistical validation and verification of the model is performed to ascertain confidence in the simulated motion current signature. Validation experiment concludes that, although, the simulation model generates a good macro-dynamical mapping of the motion current signature, it fails to accurately map the micro-dynamical structure due to the lack of knowledge regarding performance of higher order and nonlinear factors, such as backlash and compliance.

Failure of the simulation model to determine the micro-dynamical structure suggests the presence of nonlinearity in the motion current signature. This motivated us to perform surrogate data testing for nonlinearity in the motion current signature. Results confirm the presence of nonlinearity in the motion current signature, thereby, motivating the use of nonlinear techniques for further analysis.

Outcomes of the experiment show that nonlinear noise reduction combined with the linear reverse algorithm offers precise machine system parameter estimation using the motion current signature for the implementation of the real-time predictive maintenance system. Finally, a linear reverse algorithm, BJEST, is developed and applied to the motion current signature to estimate the machine system parameters.

Keywords: Motion current signature, neural network, linear reverse algorithm, simulation model, nonlinear noise reduction

To my parents and my nephew, Ansh, who has kept my parents smiling while I wasn't there...

Acknowledgements

I have benefitted greatly from the assistance and support of many people in bringing this thesis and the work described herein to fruition, and I wish to take this opportunity to thank them with pleasure.

First and foremost, I would like to thank my supervisors, Professor Barrie Jones and Dr. David Evans, for their excellent supervision, stimulating ideas, and an uncanny ability to always attend to my needs. A special thanks to David Evans for his unmatched cooking analogies, which helped me improve my English with a sense of appetite.

A sincere thanks to Graham Elvis of Rockwell Automation, main author of the simulation model used in this thesis, for his generous help throughout the course of this project and for letting me build on his source code. A special thanks to John Durrant, Roger Brookes and Dr. Gurdial Singh for their many stimulating discussions and ideas. I am also grateful to K. S. Chadha for his motivation and moral support throughout my career and, especially, during this project. I would also like to thank the staff of the Molins ITCM, Coventry, in particular David Seaward and Mike Edkins, for help with recording the production machine data.

I would also like to register my appreciation to the members of the Biomedical Engineering Research Group, in particular my fellow PhD students Anthony Molloy, Betty Tam, Mark Prince, Robin Taylor, Mark Elliott, Iskandar Petra and Paul Slack for providing a wonderful and friendly environment in which I could nurture all my ideas at work. A special thanks goes to Diane Markley, Katy Barry and Sharen Lloyd, School of Engineering assistants, for all their help on countless occasions.

I am also pleased to thank Overseas Research Student Award, Prof. Barrie Jones and Compro for their financial support to this project.

A big "thank you" to Mark Elliott and Robin Ibbotson for all those weekly running sessions, which were invaluable to reduce the calories I gained by eating at "Chicken Hut", night after night. I am also indebted to the many friends which I have made during my three years at Aston, in particular Vish, Kul, Bhav, Helen, Ahmed, George, Fay, Melo, Pierre, Alex, Ducky, James, Lee, Joobee and Mish. Last but not the least, I would like to extend a very special thanks to Sandra, for her daily support, advice, friendship, room and nutella. Certainly, you are the best "Bill" ever!

Finally, I am eternally grateful to my family for everything. None of this would have been possible without you. I love you all!

Declaration

This thesis describes the work carried out between October 2002 and October 2005 in the Control and Dynamics Research Group, School of Engineering and Applied Science at Aston University under the supervision of Dr. David J. Evans and Professor Barrie Jones.

The work reported in this thesis has been entirely executed by myself. This thesis has been composed by myself and has not, either in its present form or in a derived form, been submitted in any previous application for a degree. The following publications have been published during the course of the work described herein:

1. "A real-time predictive maintenance system for machine systems", Dheeraj Bansal, David J. Evans and Barrie Jones, International Journal of Machine Tools and Manufacture, Volume 44, Issues 7-8, Pages 759-766, 2004.

2. "Application of a real-time predictive maintenance system to a production machine system", Dheeraj Bansal, David J. Evans and Barrie Jones, International Journal of Machine Tools and Manufacture, Volume 45, Issue 10, Pages 1210-1221, 2005.

3. "BJEST: A reverse algorithm for the real-time predictive maintenance system", Dheeraj Bansal, David J. Evans and Barrie Jones, International Journal of Machine Tools and Manufacture, Article in press to be published in 2006.

4. "A real-time predictive maintenance system for machine systems - An alternative to expensive motion sensing technology", Dheeraj Bansal, David J. Evans and Barrie Jones, IEEE Sensors for Industry Conference, Houston, February 2005.

5. "Design, build, implementation, validation and application of TuneLearn, a motion dynamics simulation model", Dheeraj Bansal, David J. Evans, Barrie Jones and Graham F. Elvis, IEEE Winter Simulation Conference, Washington DC, December 2004.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

PLC	- Programmable Logic Controller
NN	- Neural Networks
MLP	- Multi-layer Perceptron
RBF	- Radial Basis Functions
PCA	- Principal Component Analysis
SVD	- Singular Value Decomposition
PID	- Proportional Integral Derivative
BJEST	- Bansal-Jones Estimation
SACS	- Segment-wise Average Current Signature
ARMA	- Auto Regressive Moving Average

Chapter 1

Introduction

1.1 Background

Much emphasis has been given to the research and development of high speed machine systems possessing high levels of programmability and reconfigurability. Motion cycle demands in such machines are high; motors and drives are required to perform under demanding conditions, hence the need for a predictive maintenance system. Modern machines typically use some form of direct current (DC) motor and the framework described in this thesis is based upon such motors.

The DC motor is one of the first machines devised to convert electrical power into mechanical power. Recent improvements in DC machines, and specifically the emergence of the brushless motor have lead to its wide usage due to its high torque and small size when compared to the induction motors.

The need for new high-performance motors, with highly sophisticated capabilities, has produced an abundance of new types and sizes of DC motor. Nowadays, DC motors are widely used in many machine applications; with this there is a need for high reliability supported by an effective maintenance system. Recent studies have demonstrated that the predictive maintenance approach can ensure high reliability and performance (Cambrias and Rittenhouse, 1988; Discenzo, 1997; Haddad, 1991; Herbert, 1984; Penmann, 1986; Smeaton, 1987; Tavner and Penman, 1992).

This thesis introduces an effective, real-time, predictive maintenance system based on the current feedback of brushless servo type of DC motors. The aim of the proposed system is to localize and detect abnormal machine parametric conditions in order to predict mechanical abnormalities that indicate, or may lead to the failure of the motor (Isermann, 1984; Leith, 1988).

In this chapter, we introduce the main considerations of this thesis: the real-time predictive maintenance problem of large scale machine systems using the motion current signature; the application of the analytical and time-series techniques; and modelling of current characteristic of a brushless servo motor.

1.2 Machine maintenance

Every facility that produces a consumer product has some requirement for maintenance or upkeep of their machinery. Depending upon the product and, to some extent, the size of the facility, this maintenance activity may be continuous or periodic in nature. Some maintenance activities may consume a significant portion of the expenses and manpower of a facility.

Facility maintenance activities generally fall into three categories: breakdown, preventive, and predictive. Each category has particular costs and benefits associated with it.

Breakdown maintenance

This method has no continuous activity associated with it. Essentially, no maintenance activity is performed on machinery until it fails or produces unacceptable product.

At first impression, this method appears to be the most cost effective because the manpower and their associated costs are minimal.

But closer examination shows that when the machinery fails, considerable expense is required to allocate manpower on an emergency basis, repair/replacement parts. Additionally, lost revenues due to non-production can mount rapidly depending upon the manufacturing process or product linked with the failed machinery. Clearly, this maintenance methodology has the highest associated and unpredictable cost. In addition, an unexpected failure can be dangerous to personnel and the facility. This prompted researchers around the world to develop another maintenance system which will reduce unpredictability and cost.

Preventive maintenance

An advancement on a breakdown maintenance program is a preventive maintenance program. This periodic approach to maintenance has little continuous activity associated with it. It involves scheduling a regular outage, usually on an annual basis, where the entire machine, process or plant is shutdown, or removed from production, for careful inspection and routine replacement of specific parts.

This method has the highest cost for replacement parts because the facility may have a separate program or department with the sole purpose of maintaining an inventory of spare parts and scheduling outage activity. Maintenance costs are reduced because the "annual outage" or "turn around" is usually scheduled for a period when the product demand is low. Additional cost savings are realized because manpower and any heavy equipment are scheduled.

The method of preventive maintenance is specially beneficial in increasing the predictability of a breakdown. However, this methodology is not very successful in the following situations:

- if a machine part breaks down before outage;
- if the vendor or the material of the machine part changes;
- if there is some health left in the machine part.

Due to these factors we need a continuous and real-time process of maintenance.

Predictive maintenance

Throughout the decade of the 1980s many facilities began to seek solutions to reduce high maintenance cost and spare parts inventories. By adopting a continuous approach to facility maintenance these reductions can be realized. Supporting this approach was the profusion of portable data collectors and database software. As an extension to a portable data collector system is a permanently installed monitoring system. Many of these systems can be interfaced to advance software systems that can assist with signal analysis. Until present day, the key to this enhanced system is having the sensors installed which are available for signal acquisition continuously.

Using these systems, and the appropriate training necessary for signal interpretation, a facility can implement a predictive maintenance program. This method relies on the data collected, either on a continuous basis or on a routine, periodic basis, to dictate the required maintenance procedure and scheduling the maintenance activity.

1.3 Main components

The development of a real-time predictive maintenance system involves application of a number of different techniques from various avenues of research, ranging from neural networks, time-series analysis through to simulation modelling. The research performed as part of this thesis has explored a number of research areas to find a viable and effective solution to maintenance issues. Various components of this thesis are therefore mentioned in the following sub-sections.

1.3.1 Neural networks

Neural networks provide a valuable alternative to conventional mathematical techniques in order to map the complex nonlinear function between the motion current signature and the machine system parameters. The application of the neural network approach to the proposed system is tested in chapter 3 as part of the proof of concept procedure.

The tests indicate a high success rate of the neural network approach when applied to the classification of load inertia with motion current signature as as input. However, beyond the proof of concept, the system needs to be developed to perform exact value estimation of more than one system parameter, which prompts the requirement of large amount of training data due to the reasons which will be clear as we progress through this thesis. This multi-fold requirement of the training data motivates us to explore various avenues of training and test data generation including simulation modelling.

1.3.2 Simulation modelling

Mathematical simulation models are increasingly being used in problem solving and in decision making (Sargent, 2004). Thus, the use of a simulation model for generating the training data, covering harder to replicate machine conditions, like current limit over-run, is motivated in chapter 4. A simulation model, TuneLearn[1], of a closed loop form based on a PID controller is developed and is shown to be capable of mapping the motion current signature to the system parameters.

The developers and the users of the models, the decision makers using information derived from the results of the models, and the people affected by decisions based on the simulation models are all rightly concerned with whether a model and its results are "correct" and "useful" (Kleijnen, 1999; Sargent, 2004). This concern is addressed through model validation and verification, which is also presented in chapter 4.

The model is found to generate a good macro-dynamical replica of the motion current signature, however, a correct mapping of the micro-dynamical behaviour is still missing. A good understanding of the micro-dynamical structure of the motion current signature is important to apply a correct mathematical or modelling or statistical technique in a precise manner.

The validation procedure reveals that the micro-dynamical; structure could be because of the presence of nonlinearity in the motion current signature. Although, nonlinearity is expected in the signature, the nonlinear time-series analysis techniques can not be applied without mathe-

[1] The simulation model developed as part of joint University and Rockwell Automation research programme.

matically testing the signature for nonlinearity. This is explained in the next section.

1.3.3 Nonlinearity

The failure of the simulation model to map the micro-dynamics, and the view that the nonlinearity might be present in the motion current signature, motivates us to test the motion current signature for inherent nonlinear behaviour. Surrogate data testing based upon the null hypothesis that the signature is derived from a linear stochastic process is performed in chapter 5.

Motion current signature collected from a real production machine rejected the hypothesis, thereby, suggesting that the signature contains nonlinear components, including noise. Alongside, it is believed that the effect of the machine system parameters (inertia, friction and gravitation) does not lead to any nonlinear characteristic in the motion current signature. This argument inspires us to perform noise reduction on the motion current signature to have a clear understanding of the process.

1.3.4 Noise reduction

The noise reduction algorithm is considered in this thesis to gain a better understanding of the motion current signature. It is contended in chapter 6 that the application of the linear reverse algorithm, BJEST (developed in chapter 4), on the noise-less motion current signature may provide better results than the application of the neural network approach.

The tests with noise reduction technique confirm the belief and lead to a possible solution to the aims and objectives of this thesis.

1.4 Contributions of thesis

The overall aim of this project is to develop a real-time predictive maintenance system capable of detecting abnormal machine conditions with a minimum requirement of motion sensing technology. Currently, the machine maintenance technology largely depends upon the use of motion sensors, which are not only expensive but also increase maintenance overhead. The technique of the real-time predictive maintenance will represent a combination of existing methods drawn from the fields of neural networks, simulation modelling, dynamical systems and nonlinear techniques. In support of this aim, this thesis will focus on the development of a machine parameter

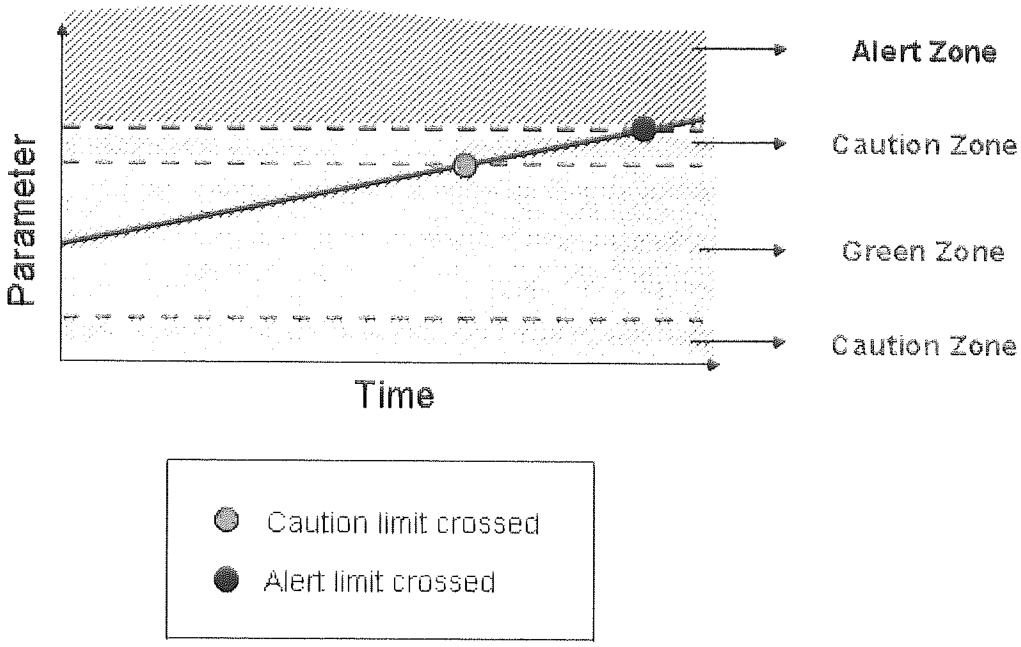

Figure 1.1: A method of displaying the real-time predictive maintenance system output. The parameter shown in the graph could be any of the machine parameters estimated using the technique.

estimation system, which significantly reduces or eliminates the use of expensive sensor technology.

This thesis motivates the use of the motion current signature as an input for the real-time predictive maintenance system. The motion current signature is readily available as a feedback to the motor drive from the motor, thus eliminating the need for any additional sensor in the real-time predictive maintenance system implementation. Measurements of the real-time motion current can be obtained using the motor-drive system's serial port interface. Interfacing simply requires the connection between the computer and the motor-drive system serial ports.

The analysis of the motion current signature is performed by the computer using basic mathematical tool-sets. The results obtained are displayed in the format chosen by the machine operator or the maintenance technician. One of the many different methods of displaying the results is shown in figure 1.1.

Figure 1.1 shows a sample plot of the variations in the value of a parameter in real-time. The settings related to various zonal levels are customisable and can be linked to an alarming system. The machine technician can monitor the graph and take action whenever there is an unexpected change in the value of the parameter.

1.5 Motivations and novelty

The fundamental tenet in our work is the assumption that the motion current signature contains relevant information pertaining to the machine system parameters and that this information can be extracted using nonlinear mapping techniques. Crucially, we claim that the effect of the most significant machine system parameters, in terms of their maintenance overheads, on the motion current signature is linear. We strongly believe that the use of the expensive motion sensing technology can be reduced or, after further research, eliminated by the implementation of the real-time predictive maintenance proposed in this thesis.

Over the past decade, a large amount of research interest has been centered around the development of the maintenance system framework for the motors and machines. The research areas have been:

- Development of a framework to monitor the internal condition of the motor on the basis of sensors and mathematical analysis techniques (Boothman et al., 1974; Chow, 1996; Keyhani and Miri, 1986; Leith, 1988; Penmann, 1986; Sood, Fahs and Henein, 1985; Tavner and Penman, 1992).

- Development of a motion current signature analysis framework for condition monitoring (Haddad, 1991; Kryter, 1989; Moseler and Isermann, 2000). However, all of the research until now in the field of the motion current signature analysis have been dedicated for the development of system which can monitor or detect machine incipient fault. This field of research has been termed as the *machine incipient fault detection and isolation* in the literature.

- Development of a framework to analyse the sensor based signals and motion current signatures using neural network and other artificial intelligence techniques (Betta, Ligouri and Pietrosanto, 1998; Cambrias and Rittenhouse, 1988; Chow, Sharpe and Hung, 1993; Herbert, 1984; Kim, Shin and Carlson, 1991; Lennox et al., 2001; Lin and Wang, 1996; Lipmann, 1989). Again, all the work performed in this part of research is also centered around systems for detecting motor internal and plant faults, rather than the faults developed outside a motor, which can affect the motor performance.

Since, there are a huge number of motor faults which are due to the factors originating outside the motor, the field of research related to the detection and identification of machine faults should be given equal importance. On the other hand, due to the expensive nature of the motion sensing technology, it is stressed in the literature that the newly developed predictive maintenance techniques should be least dependent up on the use of the sensors (Boothman et al., 1974; Discenzo, 1997; Lennox et al., 2001; Littlehales and Jones, 1997; Littlehales et al., 1998; Singh and Parikh, 1993; Smeaton, 1987).

The unavailability of a reliable machine fault isolation and detection system based upon identification of the machine system parameter changes motivates us to undertake this path of research. Additionally, as recommended in a lot of literature, this research focusses on the use of motion current signature as a possible source of information related to the machine parameters.

1.6 Plan of the thesis

This thesis will be organised in the following manner:

Chapter 1 is this introduction.

Chapter 2 gives the background to the field of machine maintenance and provides a survey of the various condition monitoring techniques that are relevant to our project. Where ever applicable, the aims and objectives laid out in this chapter will be linked with related aspects of the research as these are encountered. This chapter also presents a qualitative description of the real-time predictive maintenance system. Finally, this chapter motivates the development of a framework in light of the fact that there has been virtually very little research performed in this very important area of study.

Chapter 3 provides the proof of concept procedure performed to validate the concept behind the real-time predictive maintenance system using the neural network approach. A short introduction is given to the main concepts of the neural network theory and aspects that are relevant to this project are explained in greater detail. In particular the multi-layer perceptron type of neural network is introduced as a means of learning nonlinear mapping functions and hence forming the basis of the concept on which the real-time predictive maintenance resides. The final part of this chapter is devoted to the application of the neural network to classify load inertia using the motion current signature. The results of the application of the neural network are also presented.

Chapter 4 is the chapter devoted to the design, build, implementation, validation and verification of the simulation model. An overview of the aims and objectives, inputs and outputs and the block diagram of the simulation model is presented. The basic underlying algorithm of the simulation model is then provided, which is followed by a concise discussion of various validation and verification techniques proposed and applied in the literature. The results of the application of validation and verification techniques on the simulation model are also presented and discussed. The chapter also features the development and application of a linear reverse algorithm, called BJEST. The algorithm is developed in the chapter to "roughly" estimate the machine system parameters of the complex production machine used for validation and verification process.

Chapter 5 investigates nonlinear behaviour of the motion current signature. The chapter focusses on surrogate data testing technique to detect the presence of nonlinearity in the motion current signature. As there is a little prior knowledge about the micro-dynamics of the motion current signature, we first propose to detect the nature of the dynamics which would assist us to better visualize the signature. The result of the application of surrogate data test is then presented.

Chapter 6 details the nonlinear noise reduction procedure adopted in this thesis. After the confirmation of the presence of nonlinearity in the motion current signature in chapter 5, this chapter deals with the methods to reduce or remove the nonlinear component of the signature. This chapter begins with presenting noise determination and reduction techniques and then determines the presence of nonlinearity in the filtered data using the same surrogate data test as used in chapter 5. The estimation of the machine system parameters is then performed using the linear reverse algorithm on the filtered data, which shows encouraging results.

Chapter 7 concludes this thesis by presenting a unifying summary of the results obtained throughout the course of this project. The aims and objective laid out in the beginning of the thesis are also briefly reviewed and these are linked to the results which are presented. Finally, directions for further work are suggested and reviewed.

Chapter 2

Background

2.1 Introduction

Chapter 1 has already provided some insight to a predictive maintenance system and some of the issues and problems in its implementation. As the real-time predictive maintenance is a relative newcomer in the field of non-sensory based monitoring, it is characterized on one hand by high expectations and potential for further development, and on the other by many difficult technical problems that need to be resolved before it can be fully exploited. This chapter provides an introduction to this field of research as well as a review of traditional maintenance and monitoring techniques that have been used on machines. However, a comprehensive review of the theory and many issues surrounding machine maintenance systems is beyond the scope of this thesis and the interested reader is referred to many review articles in the literature (Boothman et al., 1974; Cambrias and Rittenhouse, 1988; Chow, 1996; Discenzo, 1997; Keyhani and Miri, 1986; Kryter, 1989; Penmann, 1986; Smeaton, 1987; Sood, Fahs and Henein, 1985; Tavner and Penman, 1992).

Section 2.2 introduces the origin and applicability of DC servo motors. Section 2.3 reviews the electrical equations that are used to model a DC servo motor, which forms the basis of the research carried out in this thesis. Because of their importance and greater relevance to this project, various intelligent system based machine maintenance techniques are described in detail in section 2.4. This section also reviews some of the traditional fault tree based machine maintenance techniques. Section 2.5 discusses the advantages and disadvantages of the machine maintenance techniques presented, and introduces the real-time predictive maintenance system, which is further explained in section 2.6.

2.2 DC servo motor

The direct current (DC) motor is one of the first machines devised to convert electrical power into mechanical power. Its origin can be traced to disc-type machines conceived and tested by Michael Faraday, the experimenter who formulated the fundamental concepts of electromagnetism.

Faraday's primitive design was quickly improved; many DC machines were built in 1800's when DC was the principal form of electric power generation (Fink and Beaty, 1978). With the advent of 60 Hz alternating current (AC) as the electric power standard in the United States and 50 Hz in the Europe, and invention of the induction motor with its lower manufacturing costs, the DC machine became less important for application into the manufacturing environment (Electro-craft, 1997). In recent years, the use of DC machines has become almost exclusively associated with applications where the unique characteristics of DC motor, the high starting torque for traction motor application, justify its cost, or where portable equipment must be run from a DC (or battery) power supply. Recent improvements in DC machines, and specifically the emergence of the brushless motor have lead to its wide usage due to its high torque and small size when compared to other electrical motors.

The need for new high-performance motors, with highly sophisticated capabilities, has produced an abundance of new types and sizes of DC motor. Nowadays, DC motors are widely used in many manufacturing machine applications; with this there is a need for high reliability supported by an effective maintenance system. Recent studies have demonstrated that the technique of predictive maintenance approach can ensure high reliability and performance (Cambrias and Rittenhouse, 1988; Discenzo, 1997; Haddad, 1991; Herbert, 1984; Penmann, 1986; Smeaton, 1987; Tavner and Penman, 1992).

Next, we present the electrical equation of a DC motor, which will serve as a basis for the predictive maintenance framework discussed in this thesis.

2.3 Electrical equations of a DC motor

This section presents the equations of a DC motor and derivation of the relation between torque requirement and velocity, which will be critical in designing the framework for the real-time predictive maintenance system.

According to Thevenin (Thevenin, 1883*a*,*b*), no matter how complex the electrical circuit, from the viewpoint of any pair of terminals, the circuit behaves as if it consisted only of a voltage source and an impedance. An impedance is a general expression which can be applied to any

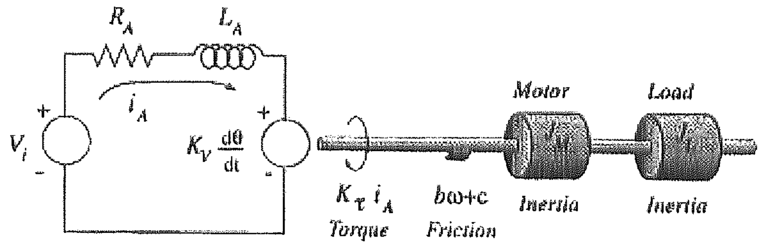

Figure 2.1: Schematic diagram of a DC servo motor showing various equivalent electrical components defining a DC servo motor operation. The diagram is inspired by the DC equivalent circuit of (Electro-craft, 1997)

electrical entity which *impedes*, resists, the flow of current, such as a resistance, a reactance or a combination of both resistance and reactance.

The first step in the computation of DC motor equations is to define the equivalent circuit, which is an electrical realisation of a physical instrument. The equivalent electrical circuit of a DC motor comprises of an impedance (a combination of R_A and L_A), connected in series to a voltage source, V_i. This equivalent circuit is shown in the figure 2.1. In figure 2.1, R_A is the armature resistance, L_A is the armature inductance, i_A is the armature current, V_i is the input voltage, K_V is the motor voltage constant, θ is the angular position of the motor shaft, K_τ is the motor torque constant, J_M is the motor inertia and J_L is the load inertia.

The internally generated voltage of a motor, $V_internal$, proportional to the motor velocity, ω, can be represented as:

$$V_internal = K_V\omega, \tag{2.1}$$

where K_V is a constant called the motor voltage constant. According to Kirchhoff's law, the relation between the variables of figure 2.1 can be given by:

$$V_i = L_A\frac{di_A}{dt} + R_Ai_A + K_V\frac{d\theta}{dt}. \tag{2.2}$$

Since the magnetic field in the motor is constant (Electro-craft, 1997), the current produces a proportional torque, T_g,

$$T_g = K_Ti_A, \tag{2.3}$$

where K_T is the motor torque constant. Let us denote the constant friction torque in the motor by T_f and the viscous friction torques (damping torque and velocity dependent friction torque), by $D\omega$, where D is the damping factor. Then, the opposing torque in the motor, T_m, is given by

$$T_m = T_f + D\omega. \tag{2.4}$$

Assuming that the motor is coupled to a load, with load moment of inertia, J_L, and the load opposing torque by T_L, the relation between the torques can be written in the form:

$$T_g = T_a + T_m + T_L, \tag{2.5}$$

in which T_a is the acceleration torque (Electro-craft, 1997) of the motor represented by:

$$T_a = (J_M + J_L)\frac{d\omega}{dt}, \tag{2.6}$$

Thus, combining the equations 2.3, 2.4, 2.5 and 2.6, the relationship between the motor torque and velocity can be written in the form:

$$i_A = \frac{(J_M + J_L)\frac{d\omega}{dt} + D\omega + T_f + T_L}{K_T} \tag{2.7}$$

Equation 2.7 is the dynamic equation of a motor relating the armature current of a motor to the system parameters, and along with equation 2.2, it describes the relations between the electrical and mechanical variables.

The parameters on the right of the equation 2.7 - motor inertia, load inertia, friction torque and load torque - are referred to as the machine system parameters. From here onwards, inertia and other parameters of equation 2.7 would be referred to as the machine system parameters.

Note that equation 2.7 is based on the assumption that motor velocity is the same as that of the load. While this assumption holds in most cases, for high-performance servo systems, the torsional resonance due to the deflections of the motor shaft and other elastic parts has to be taken into account (Electro-craft, 1997; Rockwell, 2000). Also, equation 2.7 does not represent the affect of the higher order terms, such as backlash and compliance on the armature current (Electro-craft, 1997).

Equations 2.3 and 2.7 show that the motor torque, T_g, is proportional to the armature current, i_A, and the armature current is dependent upon the inertia, friction and gravitation torque parameters of the motor. Therefore, it can be concluded that the machine system parameters affect the motor torque and current requirements. In literature, there are a number of terms used to refer to the armature current of the motor, such as the motor current feedback, motion current feedback or the motion current signature. We will be referring to the armature current

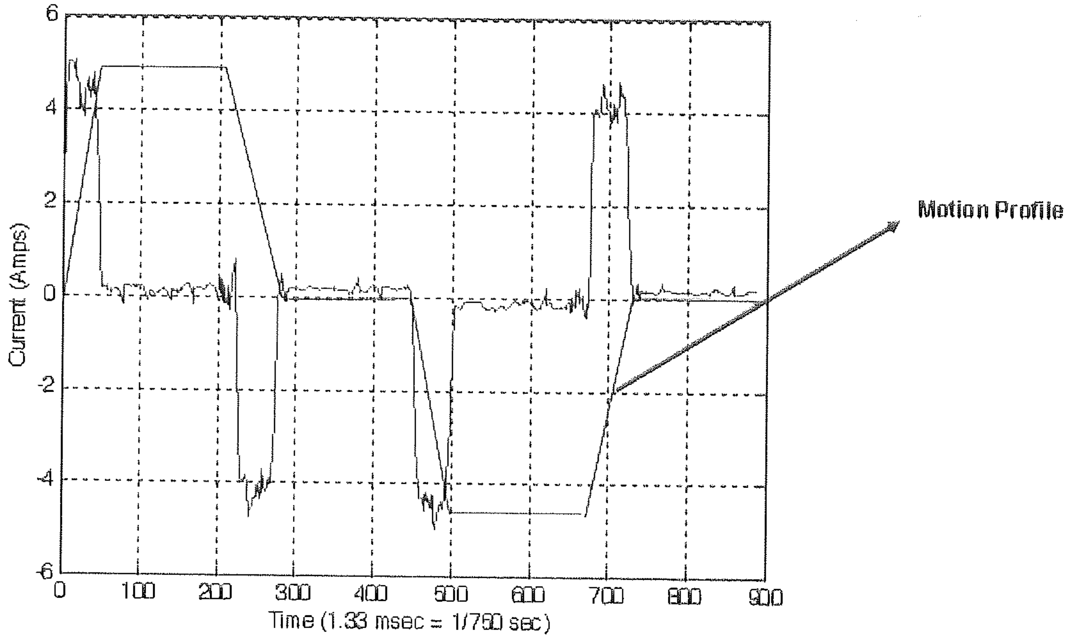

Figure 2.2: A diagram showing an example of the motion current signature superimposed on the motion profile used to generate such a signature.

as the motion current signature in this thesis. Figure 2.2 shows an example of the motion current signature superimposed on the velocity profile (motion profile) used to define the position command of the motor shaft.

The next section presents a background study of various techniques developed and employed for machine maintenance over the years along with a literature survey of the key papers.

2.4 Techniques used for machine maintenance

Monitoring of industrial processes and equipment is an essential part of a critical drive towards leaner, more competitive manufacturing. Effective monitoring can support cost reduction and efficiency improvement strategies. Functions such as maintenance, that are seen as being 'non-value adding', are being continuously required to reduce costs, whilst keeping equipment running in an optimum condition. Increasingly these systems are not only required to provide cost benefits, but also, by running equipment longer at optimum levels, be more energy efficient and environmentally friendly industrial processes. To this end, process and condition monitoring are being used to provide key information that is necessary to plan, implement and manage production in a strategic and more efficient way.

The technique of estimating machine system parameters using the motion current signature, proposed in this thesis, is a member of the family of *condition monitoring* techniques. Condition monitoring is the use of advanced technologies in order to determine equipment condition, and potentially predict failure (Frankowiak, Grosvenor and Prickett, 2004). It includes, but is not limited to, technologies such as:

- vibration measurement and analysis;
- infrared thermography;
- oil analysis and tribology ultrasonics;
- motor current signature analysis.

Condition monitoring is most frequently used as a predictive or condition-based maintenance technique. However, there are other predictive maintenance techniques, other than condition monitoring, that can also be used, including the use of the human senses (look, listen, feel, smell etc.), machine performance monitoring, and statistical process control techniques. These are the techniques with a primary aim of gathering information about condition and performance of a machine or a motor without the need for *invasive* procedures.

We now review some of the more popular condition monitoring techniques frequently used for machine maintenance. The motive of this survey is to highlight the importance of research and development in the area of condition monitoring and maintenance.

2.4.1 Fault diagnostic techniques

Fault tree

A *fault tree* is a logical tree used to map a fault to the process knowledge (Kuzawinski and Smurthwaite, 1988). In most of the condition monitoring implementations, the effect of a fault diagnosis system is determined on the basis of the amount of process knowledge inherent in the representation method used to support fault identification. Fault trees have the ability to model a physical system failure as a combination of component failures with associated failure rates. A fault tree is a logical tree in which the intermediate nodes and the leaves represent all possible causes for the undesired event located at the top node of the tree. However, due to the existence of a node for each possible cause and effect combination, the fault tree analysis can become computationally intensive (Kuzawinski and Smurthwaite, 1988) for complex machines and processes. In a fault tree, the information about a process is represented in terms of a process flowsheet, which can be viewed as a directed graph with nodes representing the process components, and edges indicating the connections and directions of flows of products and processes.

To overcome the severe computational demands of the fault tree, Raaphorst et al. (Raaphorst, Netten and Vingerhoeds, 1995) recommended implementation of an automated fault tree generation algorithm. The idea behind it relies on three key items:

- process decomposition into basic units of equipment (components) and special structures such as control loops;
- definition of component models;
- plant-wide fault tree synthesis algorithm.

Fault tree analysis has been deployed for diagnostic purposes in flexible manufacturing systems (Hu, Starr and Leung, 1999).

The developed diagnostic model considered fault symptoms as representing the roots of the tree and expert knowledge, represented in terms of rules, was deployed to suggest the best matching fault cause and the required actions. Related work (Hu et al., 2000), proposed that correct process operation could be characterized by a sequence of states and events. The resulting approach to diagnostics was based on the acquisition of digital data and analogue parameters and a diagnostic expert system.

Event tree

Fault trees offer a perfect means for explaining deviations in a single process variable. However, a set of such deviations usually occurs during on-line diagnosis. In order to list the set of faults, which can explain the deviations, all the fault trees have to be processed. The processing can be time-consuming. Moreover, there is a lot of duplicated information, and the nodes which are not measurable are of no practical relevance. The information contained in the fault trees, therefore, has to be compiled into a more suitable form of a tree called an *event tree* (Kavcic and Juricic, 2000).

An event tree is a tree structure in which a fault is placed at the tree root while the deviations of measured variables appear in the branches of the tree. Thus, in contrast to fault trees, which provide a list of all possible causes for deviation in a process variable, the event trees provide all possible consequences emerging from a given fault.

An event tree reduces the size of the process representation model to a most effective and efficient tree by considering only the most frequently observed faults. Additionally, the fault-to-cause structure of an event tree makes the diagnosis faster and efficient (Frankowiak, Grosvenor and Prickett, 2004).

Binary decision diagram

Event tree reduces the computational overhead, however, in a complex and ageing machine process, the size of an event tree can result into a lot of processing effort. To further reduce the processing effort an approach based on a *Binary Decision Diagram (BDD)* was proposed (Andrews and Dunnett, 2000).

Event tree analysis is inaccurate and inefficient in the situations where there are dependencies amongst the branch point events. The dependencies may be due to component-failures in more than one of the fault trees. In these situations, the analysis methods based on the traditional fault-tree analysis are inaccurate and inefficient. Also, the inaccuracies are not consistent across the outcome events. If frequency predictions calculated in this way are then used in a risk assessment, the relative risks would be distorted and could lead to resources being used inappropriately to reduce the overall risk. A new approach using BDD addresses these deficiencies. In BDD, events are ordered to allow only two paths (true or false) to be followed after each event, so reducing the time to diagnose the problems. In this manner, the size of the tree remains the same as the event tree but the processing burden is reduced due to the binary approach of the tree.

Sequential model

Further work (Hu, Starr and Leung, 2003) in the field of event trees concentrated on operational fault diagnostics. A *sequential model*, representing the changes in the machine operating states, was employed to investigate faults based on the actual states. A logical diagnostic model was used to provide the fault source indication, by matching the controller's signals against the expected (modelled) states (Frankowiak, Grosvenor and Prickett, 2004).

Under normal operating conditions, a programmable logic controller (PLC) controls the manufacturing system according to the sequence of actions. At the same time, each step in the control sequence is monitored by the watch-dog-timer in the PLC. If the machine is in a normal condition, it will operate sequentially according to the preset control sequence. Therefore, if the machine control status is delayed too long at a certain action, it suggests the occurrence of a fault.

Upon the detection of a sequential control fault, diagnosis is carried out using the sequential diagnosis model. At first, the current values of all the signals in the PLC are read. Then, the start conditions of every step are analyzed according to the values read from the PLC. This process identifies the step in which the fault has occurred. In the end, each control command and condition of the faulty step are checked, till the exact fault is located.

Petri-nets

The use of systems that monitor processes via an indication of process states includes the deployment of *Petri-nets* (Yang and Liu, 1998).

In this approach, fault trees were converted into petri-nets, making use of their concept to describe the evolution and the state of system degradation. Petri-net are associated with process parameters and warning levels are used to provide the marking state. Following this approach, early detection was provided by the system, which had the capability to issue alarms. Shutdown capabilities were also supported, in order to prevent further equipment damage, if maintenance had failed to intervene (Frankowiak, Grosvenor and Prickett, 2004).

In another Petri-net based approach (Prickett, 1997) and (Davey et al., 1996), faults were detected when the operating time associated to an event was exceeded. Fault isolation could then be performed based on the indication of the process signal that had prevented the event from proceeding within the established time. Such an approach is restricted to discrete signals, although it could also be used to indicate the development of faulty conditions by recording changes in the process time constants.

Ajtonyi and Terstnszky (Ajtonyi and Terstnszky, 1994) described fault diagnosis methods that considered process signals as inputs into a stochastic model, which could evaluate faults based on the residual between the model output and the actual physical parameter.

Concerns were raised on the requirements of the computational system to provide a real-time response for such implementations. These concerns could apply to most of the methods outlined above, and have prompted research into more intelligent systems that are able to target diagnostic efforts more effectively.

2.4.2 Intelligent system based techniques

Intelligent system based techniques refer to those techniques which employ the use of neural networks and fuzzy logic approaches. Such systems offer improved accuracy and efficiency coupled with an ability to learn from examples (data). Next, some of the intelligent system based techniques are explained.

Neural network

The use of *neural-networks* for condition monitoring falls into the category of intelligent system based approaches (Frankowiak, Grosvenor and Prickett, 2004).

The non-linear behaviour associated with many real systems is a source of complexity that must be considered when implementing fault diagnostics. To do this, methods such as *Fault Detection and Isolation* (FDI) [12] have been used. In FDI, pattern recognition, neural networks and decision trees, are used to provide an effective approach to fault diagnosis.

Meziane et al. (Meziane et al., 2000) provided a review that considered the employment of intelligent systems in the manufacturing area. It indicated that, for the specific application of maintenance and fault diagnostics, researchers have mainly concentrated on knowledge based systems and neural networks. Knowledge based systems were largely used for fault classification and diagnostics due to their ability to incorporate human knowledge. Neural networks represented a better alternative for the cases where domain expertise was not available.

The formulation of expert system rules for a generic fault diagnostic application is considered to be a complex task, especially in cases where process parameters and their interaction vary according to different process settings. It has been suggested (Jantunen and Jokinen, 1996) that one approach to overcome these difficulties is to dynamically generate the set of rules for each process task. Off-line processing was employed to build the rules required for the fault diagnostic process.

Benefits from the use of neural networks for fault diagnostics come from their parallel processing capabilities, non-linear mapping capabilities, the ability of learning from examples and robustness. The use of neural networks in this context can be illustrated by Lennox et al. (Lennox et al., 2001), who employed neural networks to detect and predict the failure of a melt vessel. The implementation used existing data in the neural network training process. Failure prediction was based on the error between the measured and the predicted temperature, which was then compared against defined thresholds to issue alarms.

Two level neural network

Mageed et al. (Mageed, Sakr and Bahgat, 1993) contended that employing neural networks in complex industrial fault detection systems could result in oversized networks, making the learning process extremely difficult. They proposed a two level neural network. The first level provided fault detection and isolation. The second one, using the first level outputs, provided the indication of the different levels of fault in terms of the probability of occurrence.

A neural network based fault diagnosis system of a flash smelting process (Jms-Jounela et al., 2003) demonstrated the use of self-organising maps and heuristic rules to provide process diagnostics based on the state changes detected by a neural network. Messages were issued whenever rules were matched. It was recognised that to provide robust rules precise process measurement and a good knowledge base would be required. This is true in most implementations and can be seen as being something of a limitation to the wider application of the neural network based methods.

Fuzzy logic

The use of *fuzzy* based models may offer an alternative (Ball and Isermann, 1998). In this approach, a non-linear function was represented as the sum of discrete linear segments. Symptoms were generated on the basis of residuals and ratios between process measurements and the respective model outputs. A knowledge-based system was employed for fault decision purposes.

In a similar approach (Chafi, Akbarzadeh and Moavenian, 2001), a fuzzy decision method was used to extract the process symptoms which were then applied as inputs to a neural network to establish the most probable fault cause and a confidence index. The use of knowledge-based systems has, thus, been shown to be effective if enough rules exist to classify and diagnose faults, i.e. a good knowledge of the process and the application requirements are necessary. However, this can require extensive testing and training for each possible process variation, which is often time consuming and may be prohibitively expensive.

All the above mentioned intelligent system based approaches were initially used to improve prediction estimation accuracy of the condition monitoring process. However, the use of these techniques in real-time was proposed by Johnson (Johnson, 2003). The use of sensors can provide useful information related to machine condition for processing using neural network based techniques. This has led to extensive research into the sensor based machine monitoring techniques.

2.4.3 Sensor based machine monitoring

The choice of the adequate sensors to provide the best signal is important to improve confidence in sensor-based monitoring systems. This alone, however, may not be enough to support effective monitoring since it is important to not only acquire an accurate signal but to also be able to interpret what the signal means. A good overview of the basis of many of the approaches being developed in this area was provided by Johnson (Johnson, 2003). It is not possible to include a complete review of all the sensor-based research being undertaken; so developments are illustrated in the context of machine condition monitoring.

The methods applied to condition monitoring in drilling processes were considered by Jantunen (Jantunen, 2002). Justifying that direct measurements are not very efficient in economical and technical terms, he targeted those methods that indirectly measure sudden failure and tool wear which he classified into three groups. The first group considered cutting parameters such as the torque, feed force and drift force. This indicated that such signals tend to change with the increasing amounts of tool wear. The second group consisted of vibration and sound measurements. Such methods were considered adequate for rotating machines and were said to be easy to implement in terms of sensor deployment. The last group focused on spindle motor and axis feed drive currents. Current measurement was found to be easier to implement and gave better results.

Jantunen's review then considered the subjects of signal analysis and the diagnostic methods applied. He suggested that neural network methods provided the best results but, although capable of dealing with process non-linearity, these were seen to require huge amounts of data for training purposes, if all process dynamics were to be considered. Jantunen concluded that satisfactory results were normally achieved when the process parameters (cutting conditions, work-piece) were kept unchanged. Since this cannot normally be assured during real-life processes sensor-fusion was suggested to reduce such dependency.

Turning is another very important manufacturing process that has attracted researchers. Sick (Sick, 2002) carried out a review that surveyed 138 publications related to the subject. Only online and indirect methods were considered for both tool wear and fracture. Cutting force and vibration were indicated as the process parameters mostly employed. In this work, current measurement was suggested as the preferred method in terms of implementation cost. Sick observed that those researches employing multi-sensor approaches performed the information processing at the tool wear model level. Many different methods were employed for feature extraction. The extracted features were employed as inputs to different types of neural network configurations, therefore providing the model implementation to identify the tool condition. He also suggested that multi-sensor approaches could provide better results, by comparing process parameters obtained from different sensors. In his view such approach should be applied at early stages of the monitoring system in order to provide reliable input information.

A review of sensor signals for tool condition monitoring in cutting processes by Dimla (Dimla, 2000) concluded that the cutting force and vibration are the most widely used measurement parameters. However, the dynamics of the cutting process itself results in difficulties in detecting tool faults.

Dimla (Dimla, 2000) indicated that the sensor-fusion and distributed intelligence is the best way forward. An example of this type of approach used existing signals of the machine axis feed drive to monitor tool condition (Prickett and Grosvenor, 1999). It was argued that monitoring could be based upon observing the way in which the machine controller reacts in order to keep operations within set parameters. The use of existing signals, rather than the deployment of additional sensors, was presented as an alternative to traditional condition monitoring methods. The efficacy of this approach was demonstrated by monitoring the tachometer of the X-axis of a CNC milling machine. The signal signature provided evidence to distinguish a healthy tool from a broken one (Prickett and Johns, 2001). Arguments in favour were the fact that it does not require process disruption for the implementation, since all signals are already present. An extensive review of acoustic emission research by Rao (Rao, 2003) also indicated that advances in *intelligent signal processing* will produce more successful and wider applications of sensor based systems.

In summarizing, it must be recognized that many excellent and worthwhile systems have been developed and very successfully deployed. The requirement of a rapid reaction in order to immediately detect a faulty state means most successful systems are dedicated. Cost implications and the need for expert analysis have thus limited the take up of this research. However, the advent of low-cost and powerful computer networks has made it possible to consider the wider deployment of these methods.

Next, we briefly review and classify all the above mentioned machine maintenance techniques in light of the development of a real-time predictive maintenance system in this thesis.

2.5 Broad classification of machine maintenance techniques

As already stated in the previous section, depending upon the use of intelligent systems and sensors, the existing condition monitoring techniques can be divided into three main categories:

- fault diagnostic (non-intelligent) techniques;
- intelligent system based techniques;
- sensor-based intelligent techniques.

Fault diagnostic techniques are based upon the use of non-intelligent tree based architecture, which helps to infer the cause of the breakdown in a machine system on the basis of pre-programmed machine conditions. This maintenance methodology of reaction upon breakdown inhibits the use of the traditional fault diagnostic techniques in real-time. Additionally, the accuracy and efficiency of the systems based upon such techniques are poor due to the assumption regarding system linearity.

Intelligent system based techniques are based upon the use of the neural network and fuzzy logic based approaches due to their ability to learn complex non-linear mapping functions. Such systems offer improved accuracy and efficiency coupled with an ability to learn from examples. However, the use of neural network based intelligent system techniques require large amount of training data to map complex data generator, which can prove tedious, and sometimes impossible, in a non-accessible environment. The requirement of large amount of training data can be minimised by restricting the capabilities and functionalities of such systems, but that also reduces the effectiveness of the system.

The use of sensor based machine maintenance systems overcomes the limitation of the traditional fault diagnostic or the intelligent systems to not to be able to react in real-time with least preparation effort involved. Such systems use sensor based data for obtaining the information regarding the machine condition through various parameters, which is then mapped against the anticipated condition using intelligent techniques. In other words, sensor based techniques involve use of sensory data as input for the intelligent or tree based diagnostic systems. However, sensor based systems suffer from a few drawbacks:

- sensor based systems require the use of expensive sensing technology;
- sensing technology is hard to maintain and may over-burden the maintenance framework.

The failure of all the existing machine maintenance techniques to efficiently reduce the maintenance overheads motivates the development of a real-time predictive maintenance system. The system is, therefore, required to react in real-time with minimal or no use of the sensing technology.

This thesis introduces an effective, real-time, predictive maintenance system, which eliminates the use of expensive motion sensing technology by using the motion current signature of the motor to provide crucial machine condition and performance information.

A brief explanation of the real-time predictive maintenance system is provided in the next section.

2.6 Real-time predictive maintenance system

The aim of the real-time predictive maintenance system is to localize and detect abnormal machine conditions in order to predict mechanical abnormalities that indicate, or may lead to the failure of the motor (Isermann, 1984; Leith, 1988).

The use of the conventional predictive techniques requires an accurate mathematical model in order to predict the dynamic behaviour in response to a command input (Boothman et al., 1974; Chow, 1996; Keyhani and Miri, 1986; Kryter, 1989; Sood, Fahs and Henein, 1985; ?). This requires that the machine system parameters such as load, friction torques are accurately known. Tracking of the system parameter changes during the operation also necessitates costly instrumentation, which is difficult to justify in a production environment.

The approach presented in this thesis allows deviations from the normal dynamic behaviour to be predicted accurately based upon the interpretation of the motor current signature. The interpretation of the deviations can then be made using a simple machine or process specific event tree.

2.7 Summary

In this chapter our aim has been to introduce some of the most commonly used condition monitoring techniques. As we have pointed out, the most challenging aspects of a condition monitoring technique are the accuracy, efficiency and processing overhead. The consequence is that the cost of machine, along with the sensors, and the maintenance burden of the machine is increased considerably. Intelligent systems, such as neural-network and fuzzy logic, greatly help in reducing the number of sensors needed and provide improved accuracy and efficiency.

Traditionally, the use of the sensors has been greatly related to the measurement of vibration or temperature of a specific section or part of a machine (Frankowiak, Grosvenor and Prickett, 2004). With relation to the real-time predictive maintenance system, analysis of the vibration signal can provide significant inertia information, while the friction and external torque variations can be inferred using the analysis of the temperature signal. Some of the most common inertia defects are raw material defects and imbalance of the rotary components of the machine. Friction defects can be caused by wearing belt, rack or threaded components.

The traditional implementation of condition monitoring using the neural-network approach has largely been linked to the motor internal fault detection and isolation. The problem with such an inclination towards motor internal fault detection and isolation is that not all the machine faults are born within the motor and, hence, are easy to be missed. This prompts this project to be based upon the use of intelligent system based techniques to detect and isolate machine faults not particularly related to the motor internal conditions. Furthermore, in response to issues of cost, we will use the motion current signature, thereby, eliminating the use of the expensive motion sensing technology.

This approach of using an intelligent system based approach on the motion current signature to detect and isolate machine faults, now onward referred to as the real-time predictive maintenance system, has not been attempted in any other research effort to date.

Chapter 3

Proof Of Concept

3.1 Introduction

Chapter 2 introduced a real-time predictive maintenance system based up on the analysis of the motion current signature of DC servo motor and explained some of the challenges encountered in the field of machine maintenance. This project is focussed on the development of a novel framework for machine maintenance, based upon non-sensing technology, in which these problems may be more effectively addressed.

The real-time predictive maintenance system developed in this thesis is based upon the idea of extracting relevant machine system information from the motion current signature. However, the relationship between the motion current signature and machine system parameters, described by equation 2.7, is not fully defined to account for the parameters, such as backlash and compliance. Rewriting equation 2.7, to include the higher order terms, we obtain:

$$i_A = \frac{(J_M + J_L)\frac{d\omega}{dt} + D\omega + T_f + T_L + Z}{K_T}, \tag{3.1}$$

where Z is a collective term for the all the higher order factors affecting the motion current signature. Due to the lack of professional knowledge to clearly define the effect of Z on i_A (Electro-craft, 1997; Rockwell, 2000), alternative techniques to the conventional algorithmic approaches have to be considered to map the complex non-linear relationship between the machine systems parameters and the motion current signature.

Neural networks, with their ability to derive meaning from complicated and 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 (Bishop, 1995; Nabney, 2002). A trained neural network can be thought of as an "expert" in the category of the information it has been given to analyse. The neural network can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages of the neural network approach include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-organisation: A neural network can create its own organisation or representation of the information it receives during learning time.
- Real time operation: Neural network computations may be carried out in parallel, and special hardware devices are being designed and manufactured, which take advantage of this capability.
- Fault tolerance via redundant information coding: Partial destruction of a network leads to the corresponding degradation of performance.

Therefore, we plan to use neural networks to map the complex nonlinear function between the machine system parameters and the motion current signature to be used in the real-time predictive maintenance system.

Unlike many neural network based condition monitoring systems (Betta, Ligouri and Pietrosanto, 1998; Chow, Sharpe and Hung, 1993; Lennox et al., 2001; Lin and Wang, 1996), this approach is validated in an off-line proof of concept procedure, using data from an experimental test rig providing conditions typical of those used on the production machines. This chapter focusses on the prof of concept procedure performed to validate the concept that:

there exists a relationship between the motion current signature and the machine system parameters which can be identified using a nonlinear mapping technique such as the neural networks as shown in figure 3.1.

Section 3.2 outlines the basic algorithm of the various types of neural networks. Section 3.3 to 3.6 explain the feature extraction techniques, normalization, regularization, MATLAB and NETLAB. Section 3.7 explains the experimental setup; Section 3.8 presents the results of the application of the neural network technique on the data collected from the test rig. The results and conclusion of of the application are given in section 3.9 and 3.10 of this chapter.

3.2 Neural networks

A neural network is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. A key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly

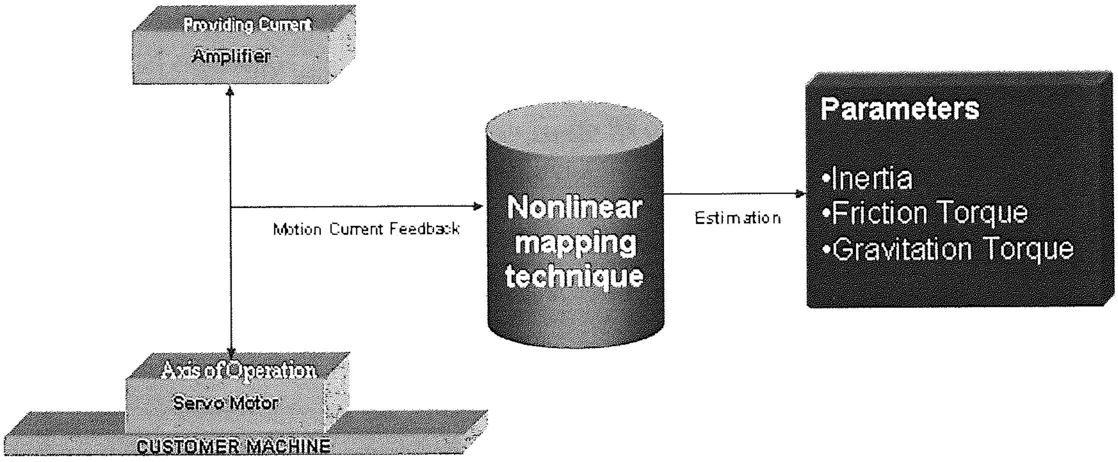

Figure 3.1: The conceptual diagram of the real-time predictive maintenance system showing the neural network based system estimating the machine system parameters using the motion current signature.

interconnected processing elements (neurones) working together to solve specific problems.

Neural networks versus conventional computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve.

Neural networks, on the other hand, process information in a similar way the human brain does and also learn by example, which increases the applicability of neural networks to the problems with unknown algorithmic solution.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks that are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

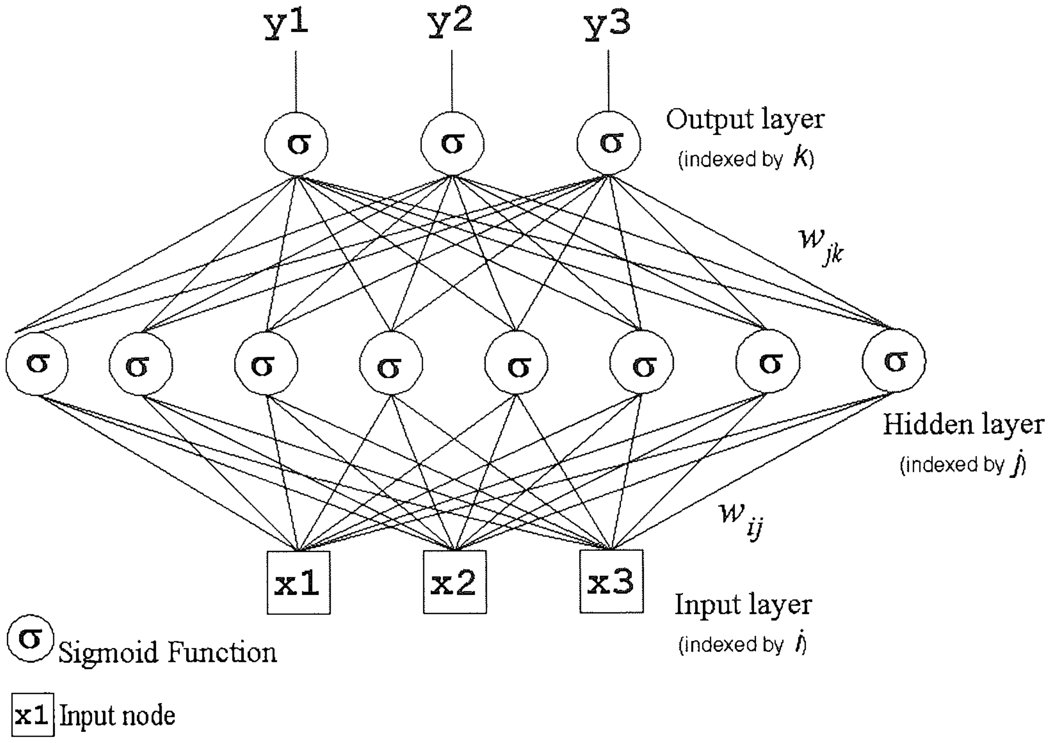

Figure 3.2: A typical multi-layer perceptron network with sigmoid activation function showing various layers and weights.

3.2.1 Various types of neural networks

There are many different types of neural network techniques developed and applied over the past decades. This section will provide a brief explanation of the most commonly used neural network techniques. However, a detailed explanation of various neural network approaches is beyond the scope of this thesis and a comprehensive insight into the most commonly used neural networks can be found in (Bishop, 1995).

3.2.1.1 Multi-layer perceptron

The Multi-Layer Perceptron (MLP) is probably the most widely used architecture for practical applications of neural networks (Nabney, 2002). In most cases the network consists of two layers of adaptive weights with full connectivity between inputs and hidden units, and between hidden units and outputs (Hornik, 1991; Hornik, Stinchecombe and White, 1989; Stinchecombe and White, 1989) (Figure 3.2).

The input values of the network are denoted by x_i where i = 1,...,d. The first layer of the network forms linear combinations of the input to form a layer of intermediate activation variables, a_j, represented by

$$a_j = \sum_{i=1}^{d} w_{ij}x_i + b_j, \tag{3.2}$$

where the variable a_j is associated with each hidden unit, w_{ij} represents the elements of the first layer of the network and b_j are the bias parameters associated with the hidden units. The variable a_j is then transformed by a non-linear activation function. The outputs of the hidden units after the transformation are then given by

$$z_j = tanh(a_j), \tag{3.3}$$

which has the property

$$\frac{dz_j}{da_j} = (1 - z_j^2). \tag{3.4}$$

The outputs, z_j, are then transformed by a second layer of weights and biases to give second-layer activation values, a_k,

$$a_k = \sum_{j=1}^{M} w_{jk}z_j + b_k, \tag{3.5}$$

where w_{jk} represents the elements of the second layer of the network. Finally, the second layer activation value, a_k, is passed through the output unit-activation function to give output values, y_k, where k = 1,...,c and c is the total number of outputs. There are three forms of activation functions used in the implementation of the MLP depending upon the type of problem dealt with by the network:

1. **Regression problem:** The appropriate choice for such type of problem is the linear function of the form

$$y_k = a_k. \tag{3.6}$$

2. **Classification problem involving multiple independent attributes:** The appropriate choice for such type of problem is the logistic sigmoid activation function due to its 2d characteristics, given by

$$y_k = \frac{1}{1 + exp(-a_k)}. \tag{3.7}$$

3. **Classification problem involving c mutually exclusive classes:** The most commonly used activation function for such type of classification problems is the softmax activation function due to its inherent 3d characteristics

$$y_k = \frac{exp(a_k)}{\sum_k exp(a_k)}. \tag{3.8}$$

The multi-layer networks can learn a suitable mapping from a given data set by using a suitable error function, which is then minimised with respect to the weights and biases in the network. The training algorithm involves an iterative procedure for minimization of an error function, with adjustments to the weights being made in a sequence of steps.

Error back-propagation One of the most commonly used training algorithm for a MLP type neural network is error back-propagation, which involves training of the MLP using gradient descent (Bishop, 1995) applied to a sum-of-squares function. This is a two stage process: in the first stage, the derivatives of the error function with respect to the weights is evaluated and in the second stage the derivatives are used to compute the adjustments to be made to the weights. Since it is in the first stage that the errors are propagated backwards through the network, the term back-propagation is used to describe the evaluation of the derivatives.

In a general, MLP network, each unit computes a weighted sum of its inputs in the form

$$a_j = \sum_i w_{ij} z_i, \tag{3.9}$$

where z_i is the activation of a unit, or input, which connects to the unit j_i and w_{ij} is the weight associated with that connection. The activation z_j of unit j can be computed by transforming the sum in equation 3.9 using an activation function $g(.)$ as

$$z_j = g(a_j). \tag{3.10}$$

As the suitable values of the weights are determined by minimizing an appropriate error functions, the error function, which can be written as a sum, over all patterns in the training set, of an error defined for each pattern, is to be defined. Let E be the error function and is written as

$$E = \sum_n E^n, \tag{3.11}$$

where n labels the patterns. The outputs of the various units depends up on the input pattern n. Also, E^n depends on the weight w_{ij} only via the summed input a_j to unit j. Application of the chain rule for partial derivative gives

$$\frac{\partial E^n}{\partial w_{ij}} = \frac{\partial E^n}{\partial a_j}\frac{\partial a_j}{\partial w_{ij}}. \tag{3.12}$$

Using equation 3.9 and by denoting $\frac{\partial E^n}{\partial a_j}$ by δ_j, we get

$$\frac{\partial E^n}{\partial w_{ij}} = \delta_j z_i. \tag{3.13}$$

Equation 3.13 tells us that the required derivative for adjusting the weights of an MLP network is obtained simply by multiplying the value of δ for the unit at the output end of the weight by the value of z for the unit at the input end of the weight (Bishop, 1995).

3.2.1.2 Radial basis functions

Radial basis function (RBF) network is the main practical alternative to the MLP for non-linear modelling. Instead of units that compute a non-linear function of the scalar product of the input vector and a weight vector, the activation of the hidden units in an RBF network is given by a non-linear function of the distance between the input vector and a weight vector (Nabney, 2002).

The output values, y_k, in the case of an RBF network is computed as

$$y_k(x) = \sum_{j=1}^{M} w_{jk}\phi_j(x) + b_k, \tag{3.14}$$

where ϕ_j are the basis functions and w_{jk} are the output layer weights. The most important benefit of an RBF network is that these type of networks are trained using a two stage process which is considerably faster than the methods used to train an MLP, such as the error back-propagation. In the first stage, the parameters of the basis functions are set so that they model the unconditional data density and in the second stage, the weights of the layers are determined using an efficient training algorithm. For stage one, a common viewpoint is that the sum of the basis functions, $\sum_{j=0}^{M}\phi_j$, should form a representation of the unconditional probability density of the input data. A more detailed explanation of RBF networks and their training algorithm is out of the scope of this thesis and the interested readers are advised to refer to (Bishop, 1995; Nabney, 2002) for a detailed description.

3.2.1.3 Bayes' theorem

The goal in the Bayes' theorem (after the Revd. Thomas Bayes, 1702-1761) is to perform classification in such a way as to minimize the probability of misclassification (Bishop, 1995). The Bayes's theorem can be written as

$$P(C_k|x^l) = \frac{P(x^l|C_k)P(C_k)}{P(x^l)}, \tag{3.15}$$

where $P(C_k|x^l)$ is called the posterior probability, since it gives the probability that the class is C_k. Bayes's theorem expresses the posterior probability in terms of the prior probability $P(C_k)$, together with the quantity $P(x^l|C_k)$ which is called the *class-conditional probability* of x^l for class C_k. The denominator in Bayes' theorem, $P(x^l)$, plays the role of a normalization factor, and ensures that the posterior probabilities sum to unity.

The posterior probability gives the probability of the pattern belonging to class C_k once the feature vector x has been observed. The probability of misclassification is minimized by selecting the class C_k having the largest posterior probability, so that a feature vector x is assigned to class C_k if

$$P(C_k|x) > P(C_j|x), \tag{3.16}$$

for all $j \neq k$.

The proof of concept procedure, explained in this chapter, used multi-layer perceptron type of neural network due to its wide usage in practical applications of neural networks (Bishop, 1995). The choice was also based upon the recommendation from my supervisor, Dr. David Evans, who is an information science specialist in School of Engineering and Applied Science, Aston University, Birmingham, UK.

Since neural networks can perform non-linear functional mappings between sets of variables, a single neural-network could, in principle, be used to map the raw input data directly onto the required final output values (Nabney, 2002). However, it is often recommended to incorporate additional steps, such as pre-processing and feature extraction, before the neural-network, to:

- filter the information adapted by the network,
- add the prior knowledge about the inputs and the outputs,
- to reduce the dimensionality of the input patterns.

A concise overview of the most commonly used pre-processing and feature extraction techniques is presented in the next section.

3.3 Pre-processing and feature extraction

Since the training of the neural network, of the form MLP or RBF, involves an iterative algorithm, it is generally convenient to process the whole training set using the pre-processing transformations, and then use the transformed data set to train the network (Nabney, 2002). One of the most important forms of pre-processing involves a reduction in the dimensionality of the input data as it exponentially reduces the number of iterations and the amount of training data needed to train a neural network (Bishop, 1995). At the simplest level, this could involve discarding a subset of the original inputs. Other approaches involve forming linear or non-linear combinations of the original variables to generate inputs for the network. Such combinations of inputs are called *features*, and the process of generating them is called *feature extraction.* Two of the most commonly used feature extraction techniques are explained below.

3.3.1 Principal component analysis

Principal Component Analysis (PCA) is essentially a technique for dimensionality reduction and is also known as *eigen-analysis.* A simple illustration of PCA is shown in figure 3.3, in which the first principal component of a two-dimensional data set is shown.

According to Bishop (Bishop, 1995), in PCA, the goal is to map vectors X_n in a d-dimensional space $(X_1, X_2, ., X_d)$ onto vectors z_n in an M-dimensional space $(z_1, z_2, ., z_m)$, where $M < d$. A vector X can be represented as a linear combination of a set of d orthonormal vectors u_i as

$$X = \sum_{i=1}^{d} z_i u_i, \tag{3.17}$$

where the vector u_i satisfies the orthonormality relation and hence the above equation can be transformed to

$$z_i = u_i^T X, \tag{3.18}$$

which can be regarded as a simple rotation of coordinate system. Now, suppose that for sake of reducing the dimensionality of the data, we want to retain only a subset $M < d$ of the basis vector u_i. The remaining coefficients will be replaced by a constant b_i so that each vector X is

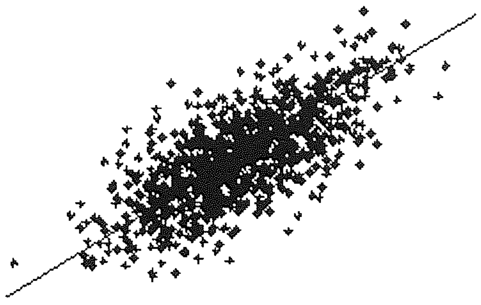

Figure 3.3: Principal component analysis of a 2-d data. The line shown is the direction of the first principal component, which gives an optimal (in the mean-square sense) linear reduction of dimension from 2 to 1 dimension.

approximated by the following expression

$$X' = \sum_{i=1}^{M} z_i u_i + \sum_{i=M+1}^{d} b_i u_i. \tag{3.19}$$

This represents a form of dimensionality reduction since the original vector X which contained d degrees of freedom is now approximated by a new vector z which has $M < d$ degrees of freedom. But, making the best approximation to the value of X would require a choice of the appropriate value of basis vector u_i and the coefficients b_i. The error in the value of the vector X can be given by the equation

$$X' - X = \sum_{i=M+1}^{d} (z_i - b_i) u_i. \tag{3.20}$$

The best approximation can, hence, be defined as the one which minimizes the sum of the squares of the errors over the whole data set. Thus, the minimization problem is

$$E = 0.5 \times \sum_{n=1}^{N} (X - X')^2 = 0.5 \times \sum_{n=1}^{N} \sum_{i=M+1}^{d} (z_i - b_i)^2. \tag{3.21}$$

If we set the derivative of E with respect to b_i to zero in equation 3.21, we find that

$$b_i = u_i^T X, \tag{3.22}$$

where the mean vector X can be defined as

$$X = \frac{1}{N}\sum_{n=1}^{N} X. \tag{3.23}$$

Combining equations 3.21, 3.22 and 3.23, we get

$$E = 0.5 \times \sum_{i=M+1}^{d} \lambda_i. \tag{3.24}$$

From equation 3.24, it is clear that the minimum error is obtained by discarding the $d - M$ smallest eigen-values, and their corresponding eigenvectors.

The linear dimensionality reduction method derived above is called *Karhunen-Loeve transformation* or PCA (Bishop, 1995). Each of the eigenvectors u_i are called the principal components.

In practice, the algorithm proceeds by first computing the mean of the vector X and then subtracting off this mean from the original value. Then the covariance matrix is calculated and its eigenvalues and eigenvectors are determined. The eigenvectors corresponding to the M largest eigenvalues are retained and the input vector X is projected onto the eigenvectors to give the components of the transformed vector z in an M-dimensional space.

There are a number of advantages of using PCA as a feature extraction technique, including:

- The computational overhead of the subsequent processing stages is reduced.
- Noise may be reduced, as the data not contained in the n first components may be mostly due to noise.
- A projection into a subspace of a very low dimension, for example two, is useful for visualizing the data.

However, the reduction in dimensionality is generally coupled with loss of information and it may happen that this information is vital for subsequent regression or classification phase. Thus, the dimensionality reduction using PCA has to be performed by carefully choosing the number of principal components to be considered for the analysis.

3.3.2 Singular value decomposition

Singular value decomposition (SVD) provides a mathematical way of extracting algebraic features from the data. SVD has been used in many fields such as data compression, signal processing and pattern analysis (Klema, 1980).

Let X denotes an $m \times n$ matrix of real-valued data and rank r, where without loss of generality $m \geq n$, and therefore $r \leq n$. The equation for SVD of X is then given by

$$X = USV^T, \tag{3.25}$$

where U is an $m \times n$ matrix, S is an $n \times n$ matrix, and V^T is also an $n \times n$ matrix. The columns of U are called the *left singular vectors*, the rows of V^T contain the elements of the *right singular vectors* and the non-zero diagonal elements of S are called the *singular values*. The diagonal elements of S are the square roots of eigenvalues of the correlation matrix represented as:

$$Z = X'X. \tag{3.26}$$

The rows of V are the corresponding eigenvectors of Z. Hence,

$$S = diag(S_1, S_2, ..., S_n). \tag{3.27}$$

Furthermore,

$$S_k > 0 for 1 \leq k \leq r, and \tag{3.28}$$

$$S_i = 0 for (r + 1) \leq k \leq n. \tag{3.29}$$

By convention, the ordering of the singular vectors is determined by high-to-low sorting of singular values, with the highest singular value in the upper left index of S matrix.

In principle, the number of non-zero eigenvalues generated by SVD is equal to the number of linearly independent vectors in the original data matrix (Bishop, 1995). This is true for well posed problems, but even the presence of errors due to numerical operations results in small eigenvalues that theoretically should be zero. Numerical errors are an insignificant problem compared to the inclusion of experimental error in the calculations.

Various statistics are available for identifying the mostly likely dimensionality of a data matrix. These statistics are designed to aid partitioning of the abstract factors into primary and secondary factors. The primary factors are those corresponding to the largest n eigenvalues and represent the set of abstract factors that span the true subspace for the data. The secondary factors are associated with the noise and, in principle, can be omitted from subsequent calculations. It is not possible to completely disassociate the true data from the error within the measured data; however the statistics guide the analyst in choosing the most appropriate number of abstract factors that describe the data and therefore the "best guess" dimensionality for the data matrix.

3.3.3 Relationship between PCA and SVD

There is a direct relationship between PCA and SVD in the case where principal components are calculated from the correlation matrix. One possible way of calculating SVD is to first calculate V^T and S by diagonalizing $X^T X$

$$X^T X = V S^2 V^T. \tag{3.30}$$

And then to calculate U as

$$U = X V S^{-1}. \tag{3.31}$$

Thus, using equation 3.30 and 3.31, we can conclude that the diagonalization of $X^T X$ yields V^T, which also yields the principal components. Hence, the right singular vectors obtained using the SVD are same as the principal components calculated using PCA. The eigenvalues of $X^T X$ are equivalent to S_k^2, which are proportional to the variances of the principal components.

Due to fundamental similarities between the PCA and SVD and due to the relatively simpler implementation of PCA (using NETLAB and MATLAB, which are explained in the next section), the proof of concept procedure uses PCA for dimensionality reduction and feature extraction of the input patterns.

3.4 Normalization

Normalization helps in scaling input variables so that they have similar magnitudes. This in turn makes sure that the network weights can all be expected to have similar values if the inputs are equally important, and so can be initialized randomly. Without normalization, network training often gets stuck in a local optimum because some of the weights are very long way from their best values (Nabney, 2002). To normalise the data, each input variable is treated independently and, for each variable x_i, the mean $\overline{x}_i$ and variance σ_i^2 is calculated on the training data. The rescaled variables are then defined by

$$\overline{x}_i^n = \frac{x_i^n - \overline{x}_i}{\sigma_i}, \tag{3.32}$$

and have zero mean and unity variance.

3.5 Regularization

A polynomial with an excess of free coefficients tends to generate mappings which have a lot of curvature and structure, as a result of over-fitting to the noise on the training data (Bishop, 1995). The technique of regularization encourages smoother network mappings by adding a penalty Ω to the error function (Bishop, 1995).

One of the simplest forms of regularizer is called *weight decay* and consists of the sum of the squares of the adaptive parameters in the network.

$$\Omega = \frac{1}{2}\sum_i w_i^2, \tag{3.33}$$

where the sum runs over all the weights and biases.

An alternative to regularization as a way of controlling the effective complexity of a network is the procedure of *early stopping*. The training of non-linear network models involves an iterative reduction of the error function defined with respect to the training data. During a typical training session, this error, referred to as the training error, generally decreases as a function of the number of iterations in the algorithm (Bishop, 1995). However, the error, termed as the validation error, measured with respect to a set of independent data, called the validation set, often shows a decrease at first, followed by an increase as the network starts to over-fit. Thus, early stopping involves the stopping of training at the point of smallest error with respect to new data, since this gives a network with the best generalization performance (Bishop, 1995).

The proof of concept procedure described in this chapter uses early stopping regularization to avoid over-fitting the neural network due to its simple implementation and applicability to the non-linear models.

The next section explains the implementation of the neural network and feature extraction algorithms for use in the proof of concept procedure.

3.6 MATLAB and NETLAB

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows many technical computing problems to be solved efficiently, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

NETLAB toolbox, developed by I. T. Nabney (Nabney, 2002), is a central tool necessary for the simulation of theoretically well-founded neural network algorithm for use in research and development. NETLAB consists of a library of more than 150 Matlab functions and scripts for the most common neural network algorithms, including MLP and RBF. Most of the functions in NETLAB operate on a data structure representing the current state of a model and matrices representing datasets.

The feature extraction, PCA, and neural network, MLP, algorithms used in this chapter are implemented using the NETLAB tool box on MATLAB.

Next, we explain the experimental setup of the proof of concept procedure.

3.7 The experimental setup

The aim of the proof of concept procedure was to validate the concept of the real-time predictive maintenance system by establishing the fact that there is a relationship between the machine system parameters and the motion current signature. This concept can be proven by conducting a simple test to classify a machine system parameter, either inertia, friction or gravitation, based upon the motion current signature using a neural network.

The training data for the neural network was collected using an experimental test rig providing conditions typical of those used on production machines. There are a number of variables in a typical operation of a test rig:

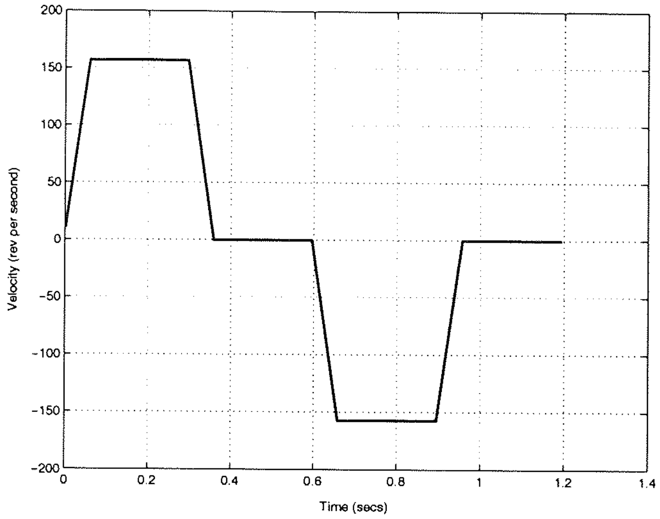

Figure 3.4: Motion profile of the motor shaft used for the proof of concept experiment. Motion profile is the velocity-time plot of machine end effector. An end effector is the terminal moving body of the machine assembly. The units of the motion profile can be rotary or linear depending upon the load type of the end effector.

1. **Motion profile:** The motion profile of the motor is a velocity-time plot which defines the position command of the motor shaft. The motion profile for the proof of concept was assumed fixed and is shown in figure 3.4.

   The motion profile was collected at the rate of 750 readings per second, which resulted in a total of 900 measurements over a period of 1.2 seconds (750 × 1.2).

2. **Motor and amplifier parameters:** The motor and amplifier parameters play a critical role in defining the current requirement of the motor during any operation cycle (Electrocraft, 1997; Rockwell, 2000). Even though the basic underlying principle of the effect of machine system parameters on the motion current signature remains same with the change in the motor and amplifier parameters, the change in the motor and amplifier parameters can either amplify or shrink the current requirement for a given position command. Therefore, the same motor and amplifier combination was used to collect the training data and to test the performance of the neural network. The motor parameters of the test-rig used for the proof of concept are shown in table 3.1.

3. **Tuning parameters:** Tuning parameters define the control configuration of a motor-amplifier combination to respond to the changes in the position command of the motor shaft. The most common of these parameters are the proportional (p) gain, the differen-

Table 3.1: Motor parameters of the experimental test rig used for the proof of concept.

Parameter Name	Value	Units
Motor Type	H-4075-R	-
Motor Inertia	0.00068	kgm^2
Motor torque constant	0.74	Nm-A
Motor peak torque	30.0	Nm
Motor poles	6	-
Motor rms torque	10	Nm
Motor damping factor	0.068	$\frac{Nm}{krpm}$
Motor friction torque	0.14	Nm
Motor maximum speed	3000	rpm

tial (*d*) gain and the integral (*i*) gain. There is a separate control configuration for the position, the velocity and the current. Although, the changing tuning parameters affect the current requirement, the real-time predictive maintenance system can not depend upon a certain tuning configuration due to the uncertain nature of the application dependent configuration. This motivates the use of changing tuning parameters to generate the training data for the proof of concept procedure. The use of changing tuning parameters will help assess the dependency of the classification process on the tuning configuration of the motor.

For the proof of concept procedure, the p-gain was varied from 0.1 to 0.3, in steps of 0.01, and v-gain was varied from 5.0 to 15.0, in increments of 1.00. The values 0.1 and 0.3, for the p-gain, and 5.0 and 15.0, for the v-gain, form the upper and lower boundary of the $1398 - H$ class of Rockwell Automation motor used on the test rig. The increments of 0.1 and 1.0 for the p-gain and v-gain, respectively, were selected to make sure that enough variation is present in the training data.

4. **Machine system parameters:** From equation 3.1, it can be seen that the acceleration torque, friction torque and load torque affect the motion current signature of a DC servo motor and the proof of concept procedure should use one of these machine system parameters for the the purpose of classification. However, the acceleration torque, due to load on the motor shaft, is easiest to vary on a test rig. This motivated the use of inertia as the machine system parameter to be used for estimation in the proof of concept. The experiment aimed to classify five distinct motor loads (inertia) given the motion current signature in spite of changing tuning parameters. Five readings of the motion current signature were taken per load per p-gain per v-gain. This resulted in a total of 5000 signatures with a sample size of 900 (Figure 3.5).

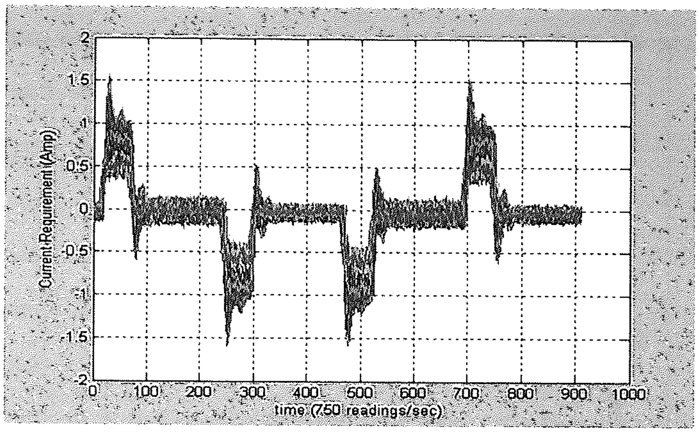

Figure 3.5: A plot showing all the 5000 collected patterns of the motion current signatures used for proof of concept procedure.

3.8 Application of the feature extraction and neural network techniques

All the collected patterns were normalized to zero mean and unity variance and then divided into three groups:

1. Training data: This group constitutes the data used for training the neural network. The number of patterns used to form this were half the total number, i.e. 2500 patterns.
2. Validation data: This group is used for early stopping regularization and consisted of a sixth of all the patterns, i.e. 838 patterns.
3. Target data: This group is utilized to assess the network performance after it is fully trained using the training data. This group had a third of all the patterns, i.e. 1662 patterns.

The distribution of the patterns into the training, validation and target data was based upon half-sixth-third (training-validation-testing) approach. Given the pattern size, the selection was based using the heuristic rule to maximize the generalization capability of the neural network model (Lipmann, 1989).

The features of the input data were extracted using PCA. The eigen-values were arranged in descending order, and the plot of the eigen-value against the number of principal components was generated (Figure 3.6). This plot can primarily be used to locate the number of principal components where "large" eigen-values cease and "small" eigen-values begin (Mardia, Kent and Bibby, 1979*b*). The aim was to identify M Principal Components (PCs) whose inclusion return sufficient information. Figure 3.6 highlights three such points (4, 10 and 14) where there is a

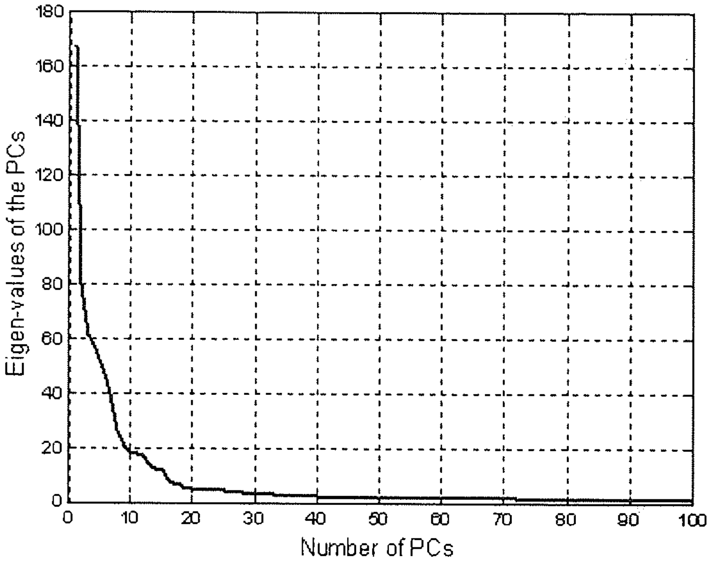

Figure 3.6: Plot of the eigen-values of the training data versus the number of PCs.

noticeable step in the value of the eigen-values. The cumulative contribution of information by the first 4 PCs to the data was found to be 60 % and hence it was not considered against 10 with a contribution of 85 % and 14 with a contribution of 96 %. The analysis with 14 principal components was adopted as that gave maximum cumulative contribution of information. This reduced the dimensionality of data from 900 patterns to 14 patterns.

An MLP network with a softmax activation function, was trained using scaled conjugate gradient (SCG) optimization algorithm (Bishop, 1995; Nabney, 2002). Early stopping regularization was used to avoid over-fitting and number of hidden units was varied from 5 to 50 in steps of 1 in order to access model order. The minimum validation error (obtained by using validation data set on trained neural network) was plotted against the number of hidden units in the MLP (Fig. 3.7). Figure 3.7 indicates that the validation error is minimum for 15 hidden units.

Neural network was trained with five output nodes, each corresponding to a load class for the purpose of classification. Table 3.2 shows the relationship between the shaft masses and the classes. The third column of the table, 1-of-N coding, indicates the value of the output nodes for a pattern to belong to each of the five load classes.

Hence, a neural network of 14 inputs, 15 hidden units and 5 outputs (1 of N coding) was used for the proof of concept procedure.

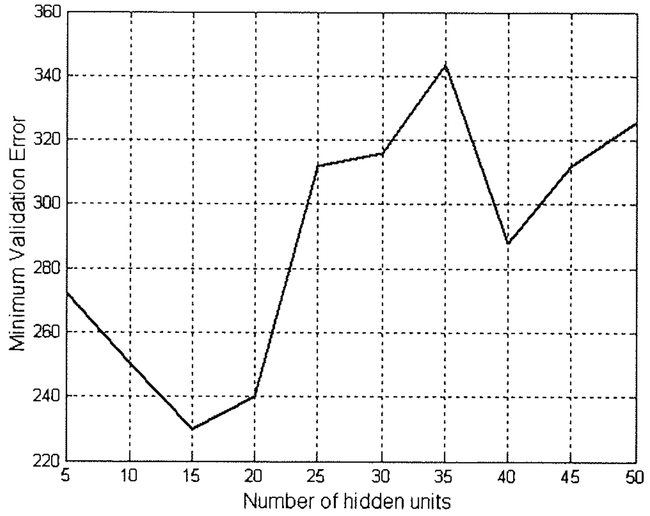

Figure 3.7: Plot of validation error versus number of hidden units. The number of hidden units is varied from 5 to 50 in steps of 1.

Table 3.2: Relationship between the shaft masses and the classes used for the proof of concept.

Description	Class	1-of-N coding
No load (shaft only)	1	10000
Small + shaft	2	01000
Smooth + shaft	3	00100
Big mass + shaft	4	00010
Cog + shaft	5	00001

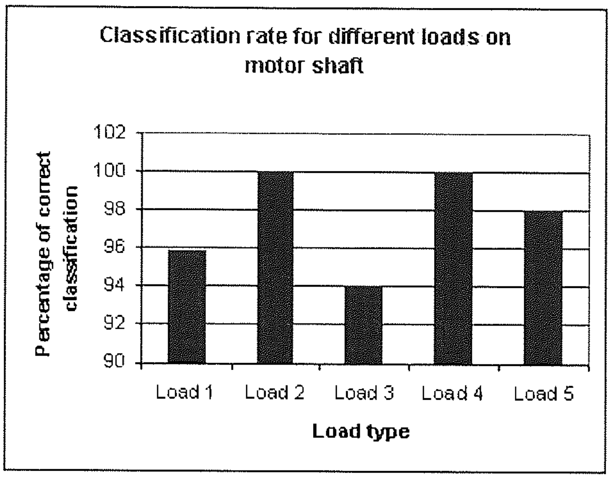

Figure 3.8: Percentage of correct load type classification obtained using the neural network approach.

3.9 Results

A confusion matrix was used to evaluate the performance of the classifier (Bishop, 1995; Nabney, 2002). Confusion matrix are a useful way of presenting the results of a classification model on a dataset. The matrix provides detailed information on the performance of the model on each class. In a confusion matrix C, the rows represent the true classes and the columns represent the predicted classes; the entry C_{ij} is the number of examples from class i that were classified as class j. Thus, an ideal matrix is one whose off-diagonal entries are all zero.

The response of the neural network to the test data was obtained and the confusion matrix was plotted (Table 3.3). The confusion matrix shows 97.59% correct classification for the test data. The off diagonal entries show that 40 out of 1662 samples are misclassified. A graph of correct classification percentage for a range of loads is shown, figure 3.8. Figure 3.8 shows that all the signatures of loads 2 and 4 were correctly classified. This is supported by all zero off diagonal entries in rows 2 and 4 of the confusion matrix (table 3.3).

3.10 Conclusion and further work

The high value (97.59%) of correct classification indicates that the neural network based approach can be used for predicting the value of the machine system parameter using the motion current signature. However, there are a few issues which need to be addressed before the proof of concept results can be used for the development of the real-time predictive maintenance system. They are:

- the experiment has only classified inertia as the machine system parameter;

Table 3.3: Confusion Matrix obtained using the neural network to classify the load information of the test data.

	Predicted Classes				
	1	2	3	4	5
Class 1	295	13	0	0	0
Class 2	0	346	0	0	0
Class 3	0	1	314	19	0
Class 4	0	0	0	322	0
Class 5	0	0	0	7	345

- the experiment has only performed classification, while the actual system is expected to compute the actual values of the machine system parameters.

As already mentioned, the real-time predictive maintenance system requires to estimate the value of the machine system parameters using the motion current signature. This means that the neural network used for such a system would need to be trained for the regression analysis and would also be required to estimate more than one parameter. The classification of load inertia into 5 classes requires far less amount of training data compared to, for example, the amount of data needed to train a neural network to estimate the value of the inertia, due to the increased network size and complexity (Bishop, 1995). This multi-fold increase in the requirement of the training data motivates an exploration into the possible sources and data collection methodologies of training data.

3.11 Summary

The objective of the work described in this chapter has been to validate the concept behind the real-time predictive maintenance system. In particular we have tried to focus on the idea of the machine system parameter classification/estimation using the motion current signature based upon the use of the neural network approach.

Our tests indicate that a high value of correct classification (97.59%) of the load inertia was obtained using the neural network approach.

This chapter concludes the proof of concept procedure and supports the idea of using the motion current signature for the real-time predictive maintenance system. In the next chapter, we review the methods of generating and collecting the large amount of data needed for training the neural network for the real-time predictive maintenance system.

Chapter 4

The Simulation Model

4.1 Introduction

The previous chapter has validated the concept behind the real-time predictive maintenance system and verified that the neural network approach can be used to extract information pertaining to the machine system parameters from the motion current signature. However, this project is an attempt to develop a framework capable of not just classifying but of estimating the value of the machine system parameters. Additionally, the real-time predictive maintenance system is to estimate more than one machine system parameter. This multi-fold requirement increases the relevance of the data used for training the neural network. This chapter focuses on the collection/generation of the data to be used to train the neural network for use in the real-time predictive maintenance system.

A fundamental requirement for the successful implementation of a neural network is the availability of relevant, information-rich training data. The real-time predictive maintenance system requires machine parameters to be varied to cover all the anticipated machine conditions; which ensures that any parameter variation can be interpolated (Kim, Shin and Carlson, 1991; Lipmann, 1989).

While an ideal solution would be to utilize training data from a real production system, this is impractical for a number of reasons:

- a large number of sensors would be required to collect data relevant to all machine parameters;
- machine faults are rare and unlikely to occur;
- it is impractical to scan the entire range of machine operations.

This motivates an alternative approach for developing a system which generates the training data using simulation models.

Simulation modelling is a valuable alternative provided that the model is fully validated against a real production machine (Isermann, 1984). The model can be used to generate the training data covering all anticipated machine conditions, including the rare and unlikely events of the machine operation cycle.

A simulation model, called TuneLearn, capable of generating the training data for the neural network was developed and is presented in this chapter. The model is a result of a joint Aston University and Rockwell Automation research programme. The preliminary algorithm of the simulation model was designed by Graham Elvis of Rockwell Automation. The software version of the algorithm, used in this project and termed as TuneLearn, was coded by the author. The simulation model is capable of:

- generating the motor current and velocity characteristics of a motor on the basis of system parameters, motion profile, motor-amplifier data and tuning configuration;
- modelling fault conditions, which are hard to replicate in an on-line environment.

As such, the model will be used to provide a mapping of system parameters and motion current signatures for all anticipated machine and tuning configurations. This mapping will then be used as the training data for the neural network (Figure 4.1).

This method of using a simulation model eliminates the need to collect the training data from a real production machine. Also, the time and effort needed to collect large quantities of data is reduced to a minimum.

This chapter presents the design, implementation, validation and verification of the simulation model.

Simulation model aims and objectives, inputs and outputs and the block diagram are explained in sections 4.2, 4.3 and 4.4, respectively. Section 4.5 details the validation and verification procedure theory and is followed by section 4.6 and 4.7 explaining the construction of the production machine and the test rig. Section 4.8 highlights the collection routine of the motion current signature. Section 4.9 and 4.10 present the linear reverse algorithm, BJEST, and it's application. Finally, section 4.11 and 4.12 give the results of the validation and verification process.

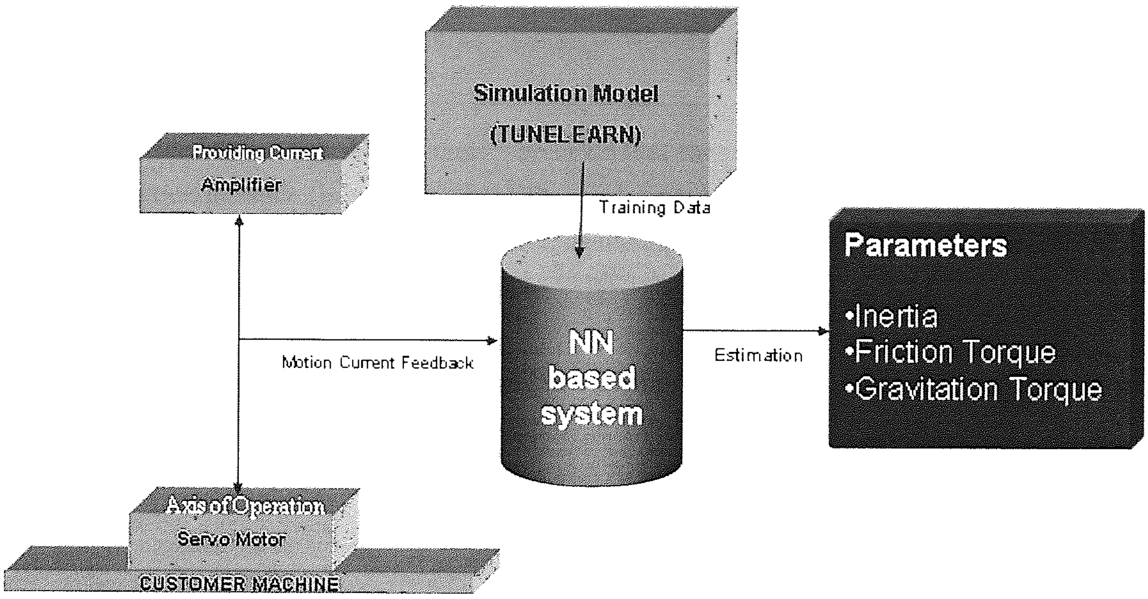

Figure 4.1: The conceptual diagram of the real-time predictive maintenance system. The diagram is inspired by figure 3.1 with a modification to show the simulation model (TuneLearn) as the source of the training data for the neural network approach.

4.2 Aims and objectives of the simulation model

The aim of the simulation model is to simulate the motion dynamics of a DC servo motor based upon:

- tuning configuration (PID) parameters;
- motor-amplifier parameters;
- motion profile;
- machine system parameters.

The motion current signature generated using the simulation model would be used to train the neural network for use in the real-time predictive maintenance system, as shown in figure 4.1. Apart from use in the real-time predictive maintenance system, the simulation model can also be used to model critical engineering or biomedical applications for determining the most efficient control/tuning parameters. In such applications, the tuning parameters can be varied in the simulation model until the most appropriate motion current signature is achieved. The graphical user interface of the simulation model makes it easier to analyse the motion current signature and effectively modify various input parameters to monitor the effect on the motion current signature.

The objective of the simulation model is to simulate the effect of the input parameters to compute the motion current and torque requirements. Next section presents the inputs and outputs of the simulation model.

4.3 Inputs and outputs of the simulation model

As already mentioned in the previous section, the inputs to the simulation model are:

- tuning configuration (PID) parameters;
- motor-amplifier parameters;
- motion profile;
- machine system parameters.

The detail of all the input parameters of the simulation model is presented in section 3.7. The output of the simulation model is the motion current signature. Thus, keeping the tuning configuration, motion profile and motor-amplifier combination constant, if the value of the machine system parameters is altered, the effect on the motion current signature can be examined using the simulation model. Similarly, the effect of the tuning configuration on the motion current signature can be monitored by keeping all the other input parameters same. The tuning configuration adjustment technique is useful in designing an application off-line in order to save critical industrial time when installing machines on-line.

Figure 4.2 shows the conceptual model of the simulation model showing the inputs and outputs.

4.4 Block diagram of the simulation model

The simulation model is of a closed loop form as shown in the figure 4.3. It contains three connected PID (proportional-integral-differential) loops in sequence: a position loop, a velocity loop and a current loop. The block diagram shown in figure 4.3 shows the calculation of the motion current signature from the motion profile input. The model takes the motor position requirement as the input and gives the motor current signature as the output. The model supports a wide range of control applications and, therefore, uses position, velocity and current loops. However, the choice of the loops is based up on the type of the control application, as indicated in table 4.1.

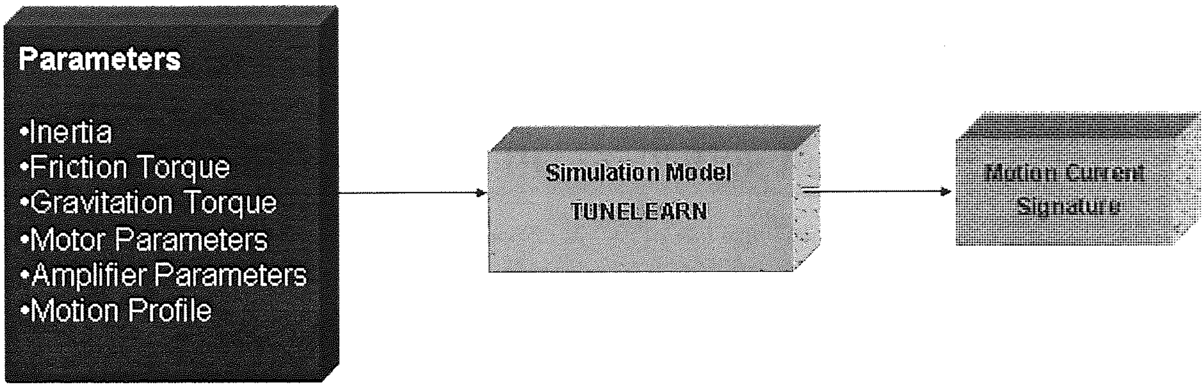

Figure 4.2: The conceptual model of the simulation model showing the inputs and outputs.

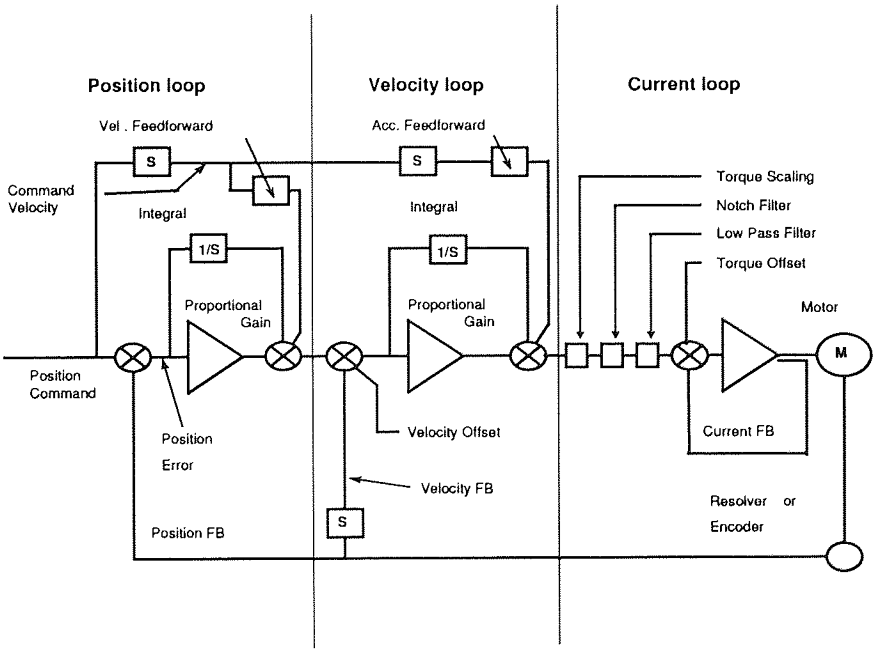

Figure 4.3: Block diagram of the simulation model highlighting the position, the velocity and the current loops.

Table 4.1: Loop configuration of the simulation model showing the status of various loops of the model depending upon the type of control application.

	Position Feedback Source	Velocity Feedback Source	Torque Offset	Velocity Loop	Position Loop
Position Servo	Motor	Motor	On	Yes	Yes
Velocity Servo	None	Motor	On	Yes	No
Torque Servo	None	None	On	No	No
Auxiliary Position Servo	Auxiliary	Auxiliary	On	Yes	Yes
Dual Position Servo	Auxiliary	Motor	On	Yes	Yes
Motor Dual Command Servo	Motor	Motor	Off	Yes	Yes
Auxiliary Dual Command Servo	Auxiliary	Auxiliary	Off	Yes	Yes

The position loop of the simulation model calculates the position error by subtracting motor position (obtained as feedback from the previous iteration of the closed loop calculation) from the position command. The output of the position loop is the velocity command, which is calculated using the position loop gains and the position error. The velocity command is carried forward as an input to the velocity loop.

The velocity loop calculates velocity error by subtracting the velocity feedback (obtained by differentiating the motor position) from the velocity command. The output of the velocity loop is current command, which is calculated using the velocity loop gains and the velocity error. The current command is carried forward as an input to the current loop.

The current command is filtered using a notch and a low pass filter within the current loop to remove any ripple effect induced into the signal due to the closed loop calculations with imprecise feedback. The filtered current command is then used by the amplifier to drive the motor. This filtered current command is the simulated motion current signature generated by the simulation model.

The following section explains the algorithm used in the simulation model.

4.4.1 Simulation model algorithm

Initially, the motion profile is broken into smaller motion profiles called the sub-profile, of a time slice duration. The *time slice* is a small time interval normally equal to the encoder count. Each sub-profile is then fed into the simulation model recursively; the algorithm of the simulation model is based upon the time slice approach.

Following; we describe the steps of the simulation model algorithm.

4.4.1.1 Initial calculations

The following set of calculation is performed to evaluate the various components of the torque command before the position command of the sub-profile can be used in the PID loops.

The total inertia (J) of the sub-profile is given by

$$J = J_M + J_L, \tag{4.1}$$

where J_M and J_L denote the motor inertia and load inertia respectively. The total inertia is then used to calculate the acceleration torque (A_T) of the sub-profile using

$$A_T = J \times \frac{\Delta\omega}{\Delta t}. \tag{4.2}$$

where ω is the angular velocity, Δt is the sub-profile duration equal to the time slice and $\frac{d\omega}{dt}$ is the angular acceleration of the sub-profile. The friction torque, F, of the system can be found by

$$F = (F_T + F_{Tm}) \times S, \tag{4.3}$$

where

S is a constant, and is:

1. equal to 1, when the motion is in the clockwise direction;
2. equal to -1, when the motion is in the anti-clockwise direction;
3. equal to 0, when the motor shaft is stationary.

F_T represents friction torque of the sub-profile and F_{Tm} is the motor friction torque. Adding acceleration torque , A_T, gravitation torque, G_T, and friction torque, F, we obtain

$$T = G_T + A_T + F, \tag{4.4}$$

where T is the total torque of the sub-profile.

4.4.1.2 Position loop

Please note that all the calculations hereafter are in the time domain and a superscript of t is used to denote it.

The position command of the sub-profile is written in the form

$$P^t_{cmd} = P^{t-1}_{cmd} + V^t_{ff} \times T_s, \tag{4.5}$$

in which the parameter P^{t-1}_{cmd} is the position command of the previous sub-profile, V^t_{ff} is the shaft velocity of the sub-profile and T_s is the time slice. The actual position, P^t_A, is then subtracted from the position command, P^t_{cmd}, to calculate the position error, P^t_{err}, at time t, so that

$$P^t_{err} = P^t_{cmd} - P^t_A. \tag{4.6}$$

The differential position error, P^t_{diff}, shows the rate of change of position error. Subtracting the position error of the previous sub-profile, P^{t-1}_{err}, from the position error, P^t_{err}, gives

$$P^t_{diff} = P^t_{err} - P^{t-1}_{err}. \tag{4.7}$$

Similarly, the integral position error, P^t_{int}, is given by

$$P^t_{int} = P^{t-1}_{int} + P^t_{err}. \tag{4.8}$$

The position loop is based upon a PID controller with velocity feed-forward gain, K_{ffv}. The velocity feed-forward gain, K_{ffv}, generates a velocity command signal proportional to the derivative of the position command, P^t_{cmd}. Using equation 4.6, 4.7 and 4.8 to form a PID equation, the velocity command, V^t_{cmd}, takes the form

$$\begin{aligned} V^t_{cmd} &= ((P^t_{err} \times K_p) + (P^t_{int} \times K_i) \\ &\quad +(P^t_{diff} \times K_d) \\ &\quad +(K_{ffv} \times V^t_{ff})). \end{aligned} \tag{4.9}$$

The velocity command from equation 4.9 is carried forward as an input to the velocity loop.

4.4.1.3 Velocity loop

Within the velocity loop, the actual velocity, V^t_A, is subtracted from the velocity command, V^t_{cmd}, so that

$$V_{err}^{t} = V_{cmd}^{t} - V_{A}^{t}, \tag{4.10}$$

where V_{err}^{t} is the velocity error of the sub-profile. Subtracting the velocity error of the previous sub-profile (V_{err}^{t-1}) from the velocity error (V_{err}^{t}) gives the differential velocity error,

$$V_{errdiff}^{t} = V_{err}^{t} - V_{err}^{t-1}. \tag{4.11}$$

Similarly, the integral velocity error, V_{errint}^{t}, is calculated using

$$V_{errint}^{t} = V_{errint}^{t-1} + V_{err}^{t}. \tag{4.12}$$

The acceleration feed-forward gain, K_{ffa}, generates a current command, I_{cmd}^{t}, proportional to the derivative of the velocity command. Taking the values of the velocity error, V_{err}^{t}, the differential velocity error, $V_{errdiff}^{t}$, and the integral velocity error, V_{errint}^{t}, from equations 4.10, 4.11 and 4.12 to form a PID equation, leads to

$$\begin{aligned} I_{cmd}^{t} &= ((V_{err}^{t} \times V_{gain}) + (V_{errint}^{t} \times I_{gain}) \\ &\quad +(V_{errdiff}^{t} \times D_{gain}) \\ &\quad +(\frac{V_{ff}^{t}}{T_s} \times K_{ffa})). \end{aligned} \tag{4.13}$$

The current command from equation 4.13 is passed as an input to the current loop.

4.4.1.4 Current loop

The current command, I_{cmd}^{t}, is scaled and filtered using a notch and a low pass filter. The filtered value is then termed as the motion current signature used by the amplifier to drive the motor.

Finally, the motor position acts as a feedback for the analysis of the next sub-profile, thereby, completing the closed loop control system.

4.4.2 Sample simulation to demonstrate the effectiveness of the model

The simulation model is designed to simulate the motion current signature and the current command on the same plot. This facilitates visual comparison and error measurement of current command and simulated current signature. The model also generates the velocity feedback which is mapped against the motion profile.

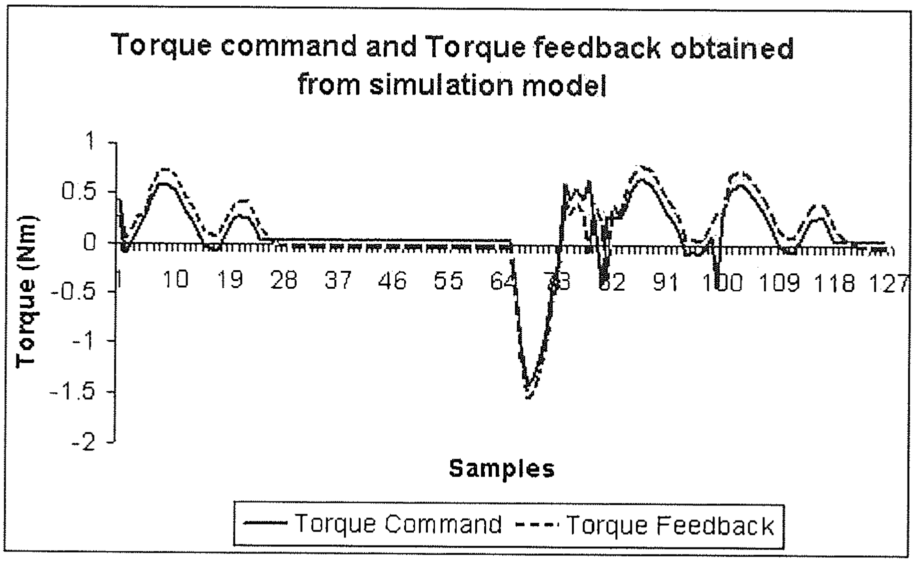

Figure 4.4: Sample simulation generated using the simulation model showing the torque command and the torque feedback. Torque command and torque feedback are analogous to current command and current signature, respectively, with motor torque constant as the conversion factor.

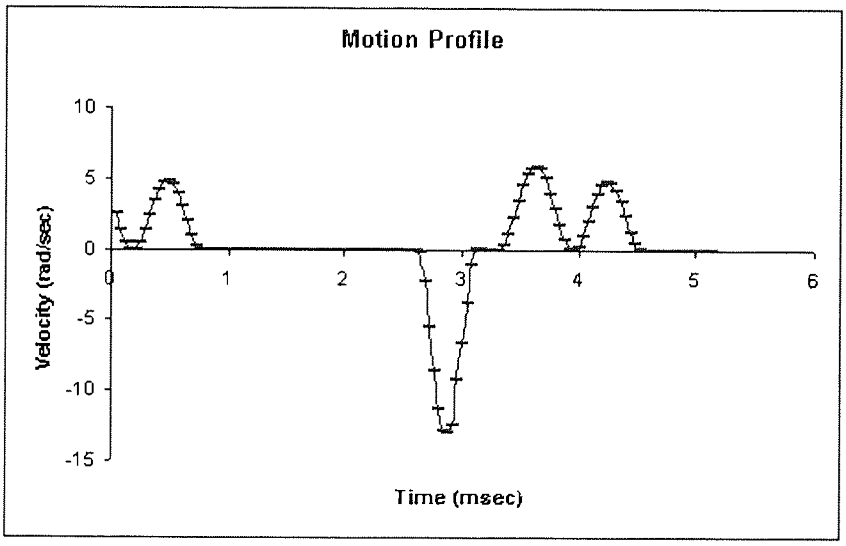

Figure 4.5: Motion profile used for generating the sample simulation shown in figure 4.4

Table 4.2: Machine parameters for sample test rig simulation

Parameter Name	Value	Units
Motor parameters		
Motor Type	H-4075-R	-
Motor Inertia	0.00068	kgm^2
Motor torque constant	0.74	Nm-A
Motor peak torque	30.0	Nm
Motor poles	6	-
Motor rms torque	10	Nm
Motor damping factor	0.068	$\frac{Nm}{krpm}$
Motor friction torque	0.14	Nm
Motor maximum speed	3000	rpm
Load parameters		
Load inertia	0.0119	kgm^2
External torque	-0.052	Nm
Friction torque	0.321	Nm
Tuning Parameters		
Position proportional gain	1000	$\frac{1}{s}$
Position integral gain	0	$\frac{1}{ms-s}$
Velocity proportional gain	2000	$\frac{1}{s}$
Velocity integral gain	0	$\frac{1}{ms-s}$

A sample simulated motion current signature generated using the model is shown in figure 4.4. The simulation is generated using the motion profile shown in figure 4.5 and machine parameters shown in Table 4.2.

The RMS error between the torque feedback, equivalent to the motion current signature and represented by dotted line in the figure 4.4, and the torque command, equivalent to the current command and shown by the solid black line in the figure 4.4, was calculated and was found to be 6%. Visual comparison of figure 4.4 also shows that the current command and motion current signature are very similar to each other. The low value of the RMS error and obvious similarities between the current command and signature reinforce the argument that the simulation model is capable of mapping the macro-dynamics of the motion current signature to the machine system parameters. However, further experiments to validate the simulation model against a real production machine need to be carried out to make sure that the simulation model is good enough to be used to generate training data for the neural network.

The next section explains the simulation model validation and verification process in detail.

4.5 Simulation model validation and verification

Simulation models are increasingly being used in problem solving and to aid in decision-making. The developers and users of these simulation models, the decision makers using information obtained from the results of these models, and the individuals affected by decisions based on such models have a justifiable concern of whether a model and its results are "correct". This concern is addressed through model validation and verification (Sargent, 2004).

Model validation is usually defined to mean "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" (Schlesinger, 1979).

Model verification is often defined as "ensuring that the computer program of the computerized model and its implementation are correct" (Kleijnen, 1999).

4.5.1 Simulation model validation

A simulation model is developed for a specific purpose (or application) and its validity has to be determined with respect to that purpose. It is often too costly and time consuming to determine that a model is absolutely valid over the complete domain of its intended applicability (Sargent, 2004). Instead, tests and evaluations are conducted until sufficient confidence is obtained that a model can be considered valid for its intended application (Kleijnen, 1999; Sargent, 2004).

According to Sargent (Sargent, 2004), there are three basic approaches for validating a simulation model:

1. **Team verification and validation:** In this approach, the model development team itself has to make the decision as to whether a simulation model is valid. A subjective decision is made based on the results of the various tests and evaluations, conducted as part of the model development process. As the TuneLearn was developed by the author alone with algorithmic inputs from the Rockwell Automation team, this approach is not considered as an effective way of validating TuneLearn.

2. **Independent verification and validation (IV&V):** This method uses a third (independent) party to decide whether the simulation model is valid. The IV&V approach should be used when developing large-scale simulation models, whose developments usually involve several teams. This approach is also used to help in model credibility, especially when the problem the simulation model is associated with has a high cost (Sargent,

2004). This approach is applicable for validating TuneLearn as it has been developed by multiple teams, based in the University and in Rockwell Automation.

The visual comparison is performed between the simulated motion current signature and the real motion current signature, collected from the real production machine and the test rig. The expert level comparison involves identification and reasoning of the differences between the simulated and the real output time series to gain confidence in the simulation model (Sargent, 2004). The simulation model is considered to be a good match of the real system if all the differences between the simulated and the real output are because of the factors beyond the scope of the simulation model. For example, the TuneLearn would be considered to be a good match of the real system if all the differences between the TuneLearn generated simulated output and the real output are due to the factors other than inertia, gravitation torque and friction torque. The effect of the inertia, friction torque and gravitational torque on the motion current signature is linear (equation 3.1) and can be easily identified in case of a noticeable difference between the real and the simulated output.

3. **Statistical techniques:** The most important statistical technique in the model validation is the sensitivity analysis (Sargent, 2004). This technique consists of changing the values of the input and internal parameters of a model to determine the effect upon the models behaviour or output (Kleijnen, 1995*a*,*b*; Sargent, 2004). The same relationships should occur in the model as in the real system. Those parameters that are sensitive, i.e., cause significant changes in the models behaviour or output, should be made sufficiently accurate prior to using the model. In principle, sensitivity analysis is based on a simple idea: change the model inputs and observe the behaviour. An important decision in sensitivity analysis is the selection of the model input parameters to be varied and the combination in which the parameters will be varied. The selection of input parameters is based upon the effect of the parameter on the model output, and the intended application of the model.

   An important term in the sensitivity analysis of TuneLearn is the segment-wise average current signature (SACS). The motion profile command of the system comprises of the straight lines called the velocity segments. The average current signature for a velocity segment is called the SACS.

   The experimental setup of the sensitivity analysis of TuneLearn is based upon the variation in the SACS due to the changes in the values of the inertia, friction torque and gravitational torque. The inertia, friction torque and gravitational torque are part of the system parameters. These system parameters have been chosen for the sensitivity analysis because these machine system parameters will be estimated using the real-time predictive

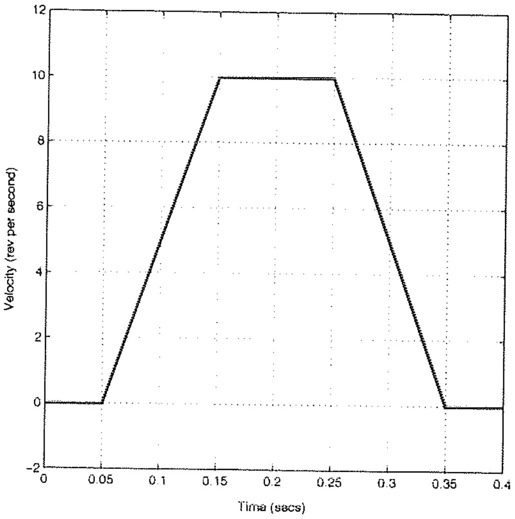

Figure 4.6: The positive trapezoidal motion profile used for the sensitivity analysis of the simulation model.

maintenance system. Additionally, the TuneLearn is intended to map the effect of system parameters (like inertia, friction torque and gravitational torque) on the motion current signature. Hence, the inclusion of the system parameters in the sensitivity analysis will ensure the validation of the model against its intended purpose. All the three system parameters: inertia, gravitational torque and friction torque, are varied in all the seven possible combinations, as would be explained later.

Apart from the three machine system parameters, the dynamics of the motion profile also has an effect on the motion current signature. The instantaneous acceleration of the motion profile affects the acceleration torque, which contributes to the total torque and current requirements (equation 3.1). This motivates the repetition of the sensitivity analysis for a positive and a negative motion profile to test for the correctness of the simulation model for the positive and negative accelerations. The two different motion profiles chosen for the sensitivity analysis are:

(a) Positive trapezoidal (Figure 4.6);

(b) Negative trapezoidal (Figure 4.7).

The motion profiles are divided into straight line segments called the velocity segments. Thus, the positive and the negative trapezoidal motion profiles have 5 velocity segments each (Figure 4.6 and 4.7).

As the sensitivity analysis is based upon the variation in the value of SACS, before

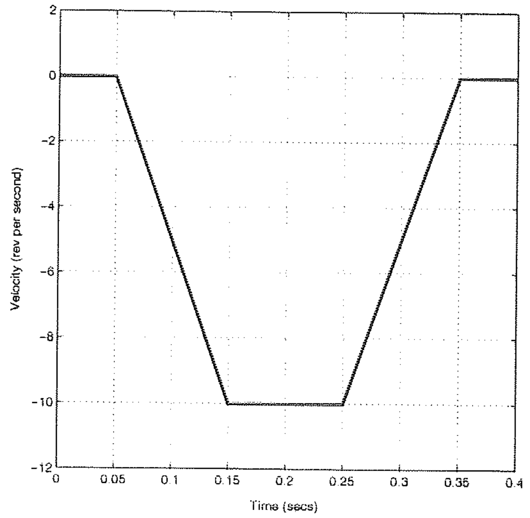

Figure 4.7: The negative trapezoidal motion profile used for the sensitivity analysis of the simulation model.

performing the sensitivity analysis, the initial level of the SACS, for each velocity segment, is determined using a zero value for all the system parameters (inertia, friction torque and gravitational torque). After this, the relative change in the value of SACS from the initial level is simulated for each scenario of changing system parameters. This process is repeated for both the motion profiles.

4.5.2 Simulation model verification

The use of statistics has been shown to be an effective technique for performing model verification (Beck et al., 1987; Kleijnen, 1995*a*,*b*, 1999; Sargent, 2004). It has been argued that, because simulation means experimentation, and any experimentation calls for statistical analysis, the use of statistical techniques for simulation verification should be preferred (Kleijnen, 1999). The statistical techniques have the advantage of yielding reproducible, objective, quantitative data about the quality of a given simulation model (Kleijnen, 1999).

According to Kleijnen (Kleijnen, 1999), the verification of the trace-driven simulation models is a unique process due to the large amount of time dependent data pertaining to each input configuration. The verification of such simulation models has to be done using the graphical comparison (Kleijnen, 1995*a*,*b*, 1999). Since TuneLearn is one such trace-driven simulation model, the verification of TuneLearn is also performed using the graphical comparison method.

In graphical comparison, the behaviour data of the model and the system are graphed for various sets of experimental conditions to determine if the model's output behaviour has sufficient accuracy for its intended purpose (Sargent, 1996). The verification of TuneLearn uses the behaviour graphs using scatter plots for the graphical comparison of simulated and actual motor current feedback.

This process involves the comparison of the time series of simulated output (y) with the time series of the real output (x) , both having the same input conditions, using the scatter plot. A line is then fit to the scatter plot, given by

$$y = \beta_0 + \beta_1 x, \tag{4.14}$$

where β_0 and β_1 denote the intercept and slope of the line respectively. According to Kleijnen (Kleijnen, 1999), the simulation model is a close match of the real system if:

1. $0 < \beta_1 < 1$, and
2. $0 < \beta_0 < \mu$,

in which it is assumed that the simulation output is a close match of the real output and hence $\mu = \mu_x = \mu_y$, where μ_x and μ_y are the mean of the real and simulated output time series respectively.

The primary requirement for performing the graphical comparison is the availability of the simulated output time series and the real output time series, collected under similar input conditions. The graphical comparison of TuneLearn is performed against a real production machine and an experimental test rig. The construction of the production machine and the test rig, the procedure for collecting the real output time series and the calculation of the input conditions are explained in the following sections.

Next, we describe the production machine and the experimental test rig used for performing the comparison.

4.6 The production machine

The production machine used for the validation and verification procedure was a tea-bag manufacturing machine, designed and customized by Molins ITCM, Coventry, UK. The machine had 8 axes of operation, all having the same motor and drive manufactured by Rockwell Automation. Table 4.3 presents the values of the motor and drive parameters used for modelling the machine in the simulation model.

The machine comprised of a computer based machine controller linked to all the axes and having a high performance drive combination. A controller and drive combination enables all

Table 4.3: Motor and drive parameters of the production machine used for performing the simulation model validation and verification.

Parameter Name	Value	Units
Motor parameters		
Motor Type	H-4075-R0H	-
Motor Inertia	0.00068	kgm^2
Motor torque constant	0.74	Nm-A
Motor peak torque	30.0	Nm
Motor poles	6	-
Motor rms torque	10	Nm
Motor damping factor	0.068	$\frac{Nm}{krpm}$
Motor friction torque	0.14	Nm
Motor maximum speed	3000	rpm
Drive parameters		
Drive Type	1398-DDM-30	-

the axes of the machine to operate in a controlled synchronous machine cycle. All the axes of the machine are explained below:

1. **Axis 1**: This axis was termed as a *Doser*. This axis was mainly rotary, high inertia, with some crank and gravitational effects.

2. **Axis 2**: This axis was termed as a *Maker*. This axis had transport belts for both sides and rotating jaws for movement, running at a constant speed. The axis was primarily a vertical Form, Fill and Seal machine with one axis.

3. **Axis 3**: This axis was termed as a *Skillet*. This axis had crank effects and motion was in two forward indexes and a reverse.

4. **Axis 4**: This axis was termed as a *Conveyor*. This axis was responsible for driving a simple geared conveyer.

5. **Axis 5**: This axis was termed as a *Gun*. This axis drove a number of glue guns.

6. **Axis 6**: This axis was termed as *Cartoner*. This axis was driving an assembly of belts and pulleys.

7. **Axis 7**: This axis was termed as a *Horizontal temper*. This axis had a reciprocating plunger action, pushing tea-bags into a box. The axis was fitted with a 3:1 planetary gearbox (Neugart PL115-03).

8. **Axis 8**: This axis was termed as a *Vertical Temper*. This axis also had a reciprocating plunger action, pushing tea-bags into a box. The axis was fitted with a 3:1 planetary

Table 4.4: Motor and drive parameters of the test rig used for simulation model validation and verification process.

Parameter Name	Value	Units
Motor parameters		
Motor Type	1326AB-B520E	-
Motor Inertia	0.004	kgm^2
Motor torque constant	2.33	Nm-A
Motor peak torque	39.1	Nm
Motor poles	8	-
Motor rms torque	13	Nm
Motor damping factor	0.13	$\frac{Nm}{krpm}$
Motor friction torque	0	Nm
Motor maximum speed	3000	rpm
Drive parameters		
Drive Type	1394C-AM50	-

gearbox (Neugart PL115-03).

Every axis of the production machine had a different motion profile and system parameters. For the simulation model validation and verification experiment, the motion current signature from each of the axis of the production machine was collected. However, the usability of an axis for the validation and verification process is dependent upon a number of factors, including the variations in the machine parameters. As would be discussed in section 4.10 in detail, only axis 3 was used for the validation and verification process due to absence of any shared inertia components. All the other axis of the production machine, other than axis 3, had variable and shared inertia.

4.7 The test rig

The test rig was a single axis machine with a motor, a drive and a system controller, all manufactured by Rockwell Automation. The motor and drive parameters of the test rig are given in table 4.4.

For the validation experiment, the current signature of the test rig was collected at zero inertia. The motion profile used for the validation and verification experiment is shown in figure 4.8.

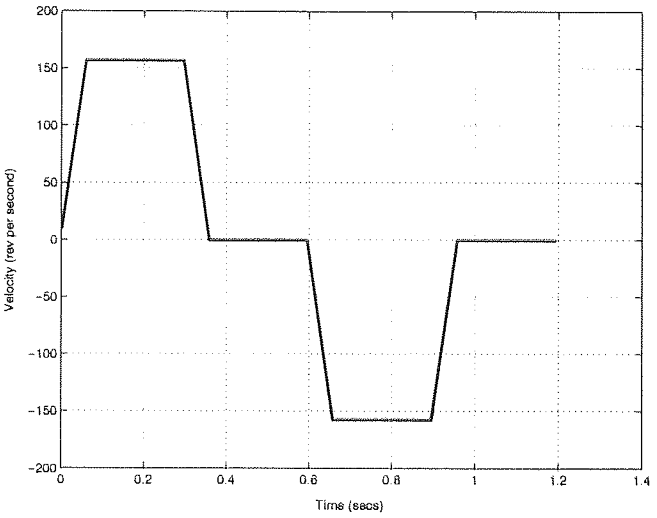

Figure 4.8: Test rig motion profile used for the validation and verification experiment.

4.8 Collection of the motion current signature from the production machine

The motion current signature was collected using the serial port command structure of the Rockwell Automation 1398 drives (Inc., 1996). The serial interfacing between the computer and the machine was established using Visual Basic (VB 6.0, Visual Studio 6.0, Microsoft Corporation).

Each axis of the production machine was capable of running at two different line speeds; normal and full (9 times the normal). The current and velocity feedbacks for both the line speeds were collected.

As already stated, the validation experiment requires both simulated output time series and the real output time series. The collected motion current signature is used as a real output time series. However, for simulating the motion current signature of the production machine using TuneLearn, the production machine system parameters and motion profile are required to be known.

The production machine had a large number of complex belt drives and shared inertias between different axis. Due to the complex architecture of the machine axes, the direct determination of machine parameters, such as inertia, friction torque and external torque was impractical. This motivated the development of an algorithm for calculating the system parameters (inertia, friction torque and gravitational torque) using the current and the velocity feedback. The algorithm has been named as BJEST (Bansal-Jones Estimation) technique after the name of the researchers. This technique is often termed as reverse engineering in sizing and selection science and significant work has been done in this field (Littlehales and Jones, 1997; Littlehales et al., 1998; Singh and Parikh, 1993).

Next section presents the BJEST algorithm, used for the calculation of machine parameters of the production machine using current and velocity feedback.

4.9 BJEST: Bansal-Jones Estimation

BJEST algorithm has been named after the researchers involved in its development; Dheeraj Bansal and Barrie Jones.

The algorithm is used for estimating the value of inertia, friction torque and gravitation torque of an axis using the motor current feedback and the motor velocity feedback. In practice, the motion current signature is dependent upon a number of factors (Penmann, 1986) including:

- acceleration torque;
- friction torque;
- gravitation torque;
- backlash;
- resonance.

However, it is impractical to calculate higher order terms like backlash and resonance of a complex machine (Littlehales et al., 1998). Therefore, the BJEST assumes:

- The motor torque feedback comprises of only acceleration, friction torque and gravitation torque constituents in absence of any higher order terms;
- The value of inertia, friction torque and gravitation torque of an axis does not change within an operation cycle;
- An axis does not have any shared inertia and friction components.

The assumptions are made so as to make sure that there is a single value of machine system parameters relating to a motion current signature of an operation cycle. The shared and variable machine system parameters introduce uncertainty in the estimation process. The three torque components can be detailed as follows:

1. **Inertia based acceleration torque:** Proportional to the rate of change of angular velocity i.e. angular acceleration;

2. **Friction torque:** A fixed bidirectional torque present whenever there is a motion. It is assumed that the viscous friction is very small compared to the constant friction torque and hence it is neglected in the BJEST;

3. **Gravitation torque (external torque):** A fixed offset which is always present in a system.

BJEST works on the basis of the instantaneous readings of the current signature and is dependent upon the encoder feedback resolution. The calculation of all the torque components is performed simultaneously. The method is based upon the regression analysis and uses the concept of pseudo-inverse (Mardia, Kent and Bibby, 1979*a*) to solve simultaneous linear equations. The algorithm is detailed in the following section.

4.9.1 Pseudo-inverse

Consider the following system of equations:

$$\begin{aligned} a_1x + b_1y + c_1z &= d_1 \\ a_2x + b_2y + c_2z &= d_2 \\ a_3x + b_3y + c_3z &= d_3 \end{aligned}$$

Where $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3, d_1, d_2$ and d_3 are constant coefficients, and x,y and z are unknown variables.

Assuming that the determinant of coefficients be:

$$A = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix},$$

the determinant of unknown variables be:

$$X = \begin{vmatrix} x \\ y \\ z \end{vmatrix},$$

and the determinant of right hand equivalent of the system of equations be:

$$D = \begin{vmatrix} d_1 \\ d_2 \\ d_3 \end{vmatrix}.$$

Then, the system of equations can be written as

$$AX = D. \tag{4.15}$$

Re-arranging equation 4.15, we obtain

$$X = A^{-1}D. \tag{4.16}$$

Since A, in general, is a non-square matrix, it does not itself has a true inverse. Hence, the BJEST uses the property of the pseudo-inverse (Mardia, Kent and Bibby, 1979a), so that

$$PI(A) \times A = I, \tag{4.17}$$

in which the parameter $PI(A)$ is the pseudo-inverse of the matrix A and I is the identity matrix. The pseudo-inverse $PI(A)$ of the matrix A is given by

$$PI(A) = (A'A)^{-1}. \tag{4.18}$$

The inverse $(A'A)^{-1}$ exists only if the matrix A has a full rank (Mardia, Kent and Bibby, 1979a). The situation, where A is not well conditioned, is not discussed in this thesis because the matrix A was found to be of the full rank in all the discussed cases (Mardia, Kent and Bibby, 1979a).

4.9.2 BJEST algorithm

According to the BJEST, the torque feedback equation of a motor can be written as:

Total Torque = Acceleration torque + Friction torque + Gravitation torque.

All the components of the total torque can be expanded individually. The acceleration torque takes the form

$Acceleration\ torque = Acc \times J$,

in which Acc is the instantaneous acceleration and J is the inertia. The friction torque is given by

$Friction\ Torque = Sign \times F,$

where $Sign$ signifies positive or negative velocity of the motor based up on the current and F means the magnitude of the friction torque. The $Sign$ is a constant value and is

1. equal to 1, when the velocity is positive;
2. equal to -1, when the velocity is negative;
3. equal to 0, when the the velocity is zero.

The gravitation torque is a constant torque effect, which is always added to the total torque and it takes the form

$Gravitation\ Torque = G_t.$

Combining the expressions for the acceleration torque, friction torque and gravitation torque, the equation for the total torque can be given by

$$T = (Acc \times J) + (Sign \times F) + (G_t). \tag{4.19}$$

For n different readings of the torque feedback, the system of equation takes the form

$$\begin{aligned}
(Acc_1 \times J) + (Sign_1 \times F) + (G_t) &= T_1 \\
(Acc_2 \times J) + (Sign_2 \times F) + (G_t) &= T_2 \\
\ldots\ldots\ldots\ldots\ldots + \ldots\ldots\ldots\ldots\ldots\ldots + \ldots\ldots &= \ldots\ldots \\
(Acc_n \times J) + (Sign_n \times F) + (G_t) &= T_n
\end{aligned}$$

in which the parameters Acc_1, Acc_2, ..., Acc_n, $Sign_1, Sign_2$, ..., $Sign_n$ and T_1, T_2, ..., T_n are the known variables, while J, F and G_t are the unknown variables.

The determinants of the system of equations would be:

$$A = \begin{vmatrix} Acc_1 & Sign_1 & 1 \\ Acc_2 & Sign_2 & 1 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots \\ Acc_n & Sign_n & 1 \end{vmatrix},$$

$$X = \begin{vmatrix} J \\ F \\ G_t \end{vmatrix},$$

and

$$D = \begin{vmatrix} T_1 \\ T_2 \\ \dots \\ T_n \end{vmatrix}.$$

Using the equations 4.16, 4.17 and 4.18, we obtain

$$X = (A'A)^{-1} \times D, \tag{4.20}$$

where X is the matrix of unknown system parameters.

Thus, the BJEST can be used to estimate the values of the inertia, friction torque and gravitation torque, if the torque feedback and the motion profile are known. However, it is stressed again that BJEST is only a highly assumptive estimation algorithm as it neglects all the nonlinear and higher order terms. The only use of BJEST in this form is to "roughly" estimate the values of the machine system parameters of the complex production machine to be able to model it in the TuneLearn.

Next, we present the results of the application of the BJEST on the current signature collected from the production machine.

4.10 Results of the application of the BJEST

One of the important requirements of the application of the BJEST is the absence of variable and shared inertias in the axis of operation. However, a close examination of the production machine resulted in the conclusion that the axis 1, 2, 4, 5, 7 and 8 had shared and variable inertia. Hence, the BJEST is applied only to axes 3 and 6 of the machine.

From the axes 3 and 6; axis 3 was chosen for the application of the BJEST due to easier access to the amplifier and the time constraint of use of the production machine at the production facility. The values of inertia, friction torque and gravitation torque obtained after the application of the BJEST to the axis 3 are presented in table 4.5.

The imprecise in the values of the inertia, friction torque and gravitation torque (Table 4.5) are due to the assumptions of the BJEST, in which the effect of nonlinear and higher order terms in neglected. However, the BJEST is just used as an approximation algorithm in this chapter. For the validation and verification process, the mean values of inertia, friction torque and gravitation torque are used as the machine system parameters for the simulation purposes.

Table 4.5: Results obtained by application of the BJEST to axis 3.

Data Set	Total Inertia (J) kgm^2	Load Inertia $J - 0.00068$ kgm^2	Friction Torque (F) Nm	Gravitational Torque (G) N-m
1	0.0124	0.01172	0.3178	0.0533
2	0.0128	0.01212	0.2874	0.0557
3	0.0131	0.01242	0.2989	0.0432
4	0.0126	0.01192	0.3210	0.0520
5	0.0119	0.01122	0.2961	0.0586
6	0.0125	0.01182	0.3102	0.0525

As stated above, the BJEST is an assumptive algorithm and, in it's current form, it only determines a rough estimate of the values of the system parameters. This implies that the simulated output, generated using the system parameters predicted using the BJEST, may indicate some offset in the value of the current signature on comparison to the real output. But the offset can be eliminated by adjusting the values of the model input system parameters. Apart from the offset differences (linear), there can also be non-linear (ripple, backlash and compliance related) differences between the simulated and the real output. The linear components, due to their larger contribution, affect the macro-dynamics of the current signature. On the other hand, the nonlinear components have micro-dynamical effect on the current signature due to their smaller affect. For the work to follow, the current signature would be considered with respect to it's macro and micro dynamical constitution.

4.11 Application of the validation process

As discussed in section 4.5, the validation experiment performed in this thesis comprises of two techniques; sensitivity analysis and visual comparison. Following, we describe the results obtained by using each of these analysis and comparison techniques.

4.11.1 Sensitivity Analysis

The sensitivity analysis is performed using the motion current signature collected from the test rig. The experimental setup of the sensitivity analysis is shown in table 4.6 in which the load inertia, friction torque and gravitational torque are varied in all the seven possible combinations. The step size of the all the input parameters was kept equal to the resolution of the simulation model; for inertia the step size was 0.0001 kgm^2, while for the friction and the gravitational torque, the step size was 0.01 Nm. All the scenarios, presented in table 4.6, were considered

Table 4.6: The experimental setup of the sensitivity analysis showing all the possible combinations of variations in the machine system parameters.

Scenario	Inertia	Friction	Gravitational
		Torque	Torque
	(kgm^2)	(Nm)	(Nm)
Initial	0	0	0
1	0.0001	0.01	0.01
2	0.0002	0.01	0.01
3	0.0001	0.02	0.01
4	0.0001	0.01	0.02
5	0.0002	0.02	0.01
6	0.0002	0.01	0.02
7	0.0001	0.02	0.02
8	0.0002	0.02	0.02

Table 4.7: Acceleration and sign values of all the segments of the positive and negative trapezoidal motion profiles used for the sensitivity analysis.

	Positive trapezoidal		Negative trapezoidal	
	A ($\frac{rad}{s^2}$)	S	A ($\frac{rad}{s^2}$)	S
Segment 1	0.00	0.00	0.00	0.00
Segment 2	628.32	1.00	-628.32	-1.00
Segment 3	0.00	1.00	0.00	-1.00
Segment 4	628.32	1.00	-628.32	-1.00
Segment 5	0.00	0.00	0.00	0.00

for both positive and negative motion profiles.

For calculating the expected behaviour of the simulation model, the values of load inertia, friction torque and gravitational torque are replaced in equation 4.20 and the total torque feedback is calculated. As is clear from equation 4.20, the total torque feedback is dependent upon the *acceleration* and the *sign* of the segment; the total torque feedback calculated for each segment separately will serve as the SACS for the sensitivity analysis. The acceleration and sign of all the segments of the motion profiles (considered for the validation experiment and shown in figures 4.6 and 4.7) are shown in table 4.7.

The expected behaviour of the simulation model was calculated using tables 4.6 and 4.7, and equation 4.20.

Table 4.8: The expected (E) and observed (O) SACS deviation of the simulation model for all the segments of the positive trapezoidal motion profile. All the values are in Nm.

Scenario	Segment 1		Segment 2		Segment 3		Segment 4		Segment 5	
	E	O	E	O	E	O	E	O	E	O
1	0.0100	0.0099	0.0830	0.0827	0.0200	0.0199	-0.0430	-0.0430	0.0100	0.0099
2	0.0100	0.0101	0.1460	0.1462	0.0200	0.0199	-0.1060	-0.1062	0.0100	0.0101
3	0.0100	0.0100	0.0930	0.0932	0.0300	0.0310	-0.0330	-0.0331	0.0100	0.0100
4	0.0200	0.0200	0.0930	0.0932	0.0300	0.0300	-0.0330	-0.0330	0.0200	0.0199
5	0.0100	0.0099	0.1560	0.1560	0.0300	0.0300	-0.0960	-0.0960	0.0100	0.0099
6	0.0200	0.0201	0.1560	0.1560	0.0300	0.0300	-0.0960	-0.0964	0.0200	0.0200
7	0.0200	0.0200	0.1030	0.1031	0.0400	0.0404	-0.0230	-0.0229	0.0200	0.0205
8	0.0200	0.0200	0.1660	0.1660	0.0400	0.0402	-0.0860	-0.0856	0.0200	0.0202

All the scenarios, shown in table 4.6, were simulated using the TuneLearn and tables 4.8 and 4.9 show the expected and observed behaviour of the simulation model for the positive and the negative trapezoidal motion profile respectively. Please note that the values shown in the tables 4.8 and 4.9 are the deviations in the value of SACS from the initial scenario (Table 4.6). Tables 4.8 and 4.9 can be used to detect the sensitivity of the simulation model to the changes in the value of the machine system parameters.

Tables 4.8 and 4.9 show that the simulation model is very sensitive to the model input changes. The model is found to accurately respond to the changes of the order of 0.0001 kgm^2 in the value of inertia and to the changes of the order of 0.01 Nm in the value of friction or gravitation torque. The relative accuracy of the TuneLearn to the input parameter changes was found to be of the order of 99.5%.

However, as already mentioned, the sensitivity analysis of the simulation model is based upon the variations in the value of SACS. The SACS does not account for the current signature behaviour of the model within the segment. Additionally, the sensitivity analysis of the model is only a theoretical way of validation. This motivates the visual comparison of the simulation model against a real production machine and an experimental test rig.

4.11.2 Visual comparison

The visual comparison of the simulation model involves identification and reasoning of the differences between simulated and real output time series. The main requirement of the validation using the visual comparison is the availability of the simulated and the real output time series.

Table 4.9: The expected (E) and observed (O) SACS deviation of the simulation model for all the segments of the negative trapezoidal motion profile. All the values are in Nm.

Scenario	Segment 1		Segment 2		Segment 3		Segment 4		Segment 5	
	E	O	E	O	E	O	E	O	E	O
1	0.0100	0.0099	-0.0630	-0.0630	0.0000	0.0000	0.0630	0.0627	0.0100	0.0099
2	0.0100	0.0101	-0.1260	-0.1257	0.0000	0.0001	0.1260	0.1261	0.0100	0.0101
3	0.0100	0.0100	-0.0730	-0.0733	-0.0100	-0.0101	0.0530	0.0529	0.0100	0.0100
4	0.0200	0.0200	-0.0530	-0.0530	0.0100	0.0100	0.0730	0.0731	0.0200	0.0199
5	0.0100	0.0099	-0.1360	-0.1359	-0.0100	-0.0100	0.1160	0.1161	0.0100	0.0099
6	0.0200	0.0201	-0.1160	-0.1161	0.0100	0.0099	0.1360	0.1360	0.0200	0.0200
7	0.0200	0.0200	-0.0630	-0.0627	0.0000	0.0000	0.0630	0.0630	0.0200	0.0205
8	0.0200	0.0200	-0.1260	-0.1258	0.0000	0.0000	0.1260	0.1264	0.0200	0.0202

As already mentioned, the graphical comparison of the simulation model is performed against a real production machine and an experimental test rig. Figures 4.9 and 4.10 show the time series of simulated output along with the time series of the real output for real production machine and the experimental test rig respectively.

The differences between the real and the simulated output are marked as shown in figures 4.11 and 4.12.

The simulation model would be considered to be a good match of the real system if all the differences shown in figures 4.11 and 4.12 are due to the factors other than inertia, gravitation torque and friction torque.

The experts involved in the visual comparison of the simulation model were Graham Elvis (Commercial Engineer, Rockwell Automation, Crewe, UK), John Durrant (Former Manager, Rockwell Automation, Crewe, UK), Roger Brookes (Former Manager, Rockwell Automation, Crewe, UK) and Prof. Barrie Jones (Aston University, Birmingham, UK).

In the following sub-section, we discuss the differences shown in the Figures 4.11 and 4.12 for the real production machine and the test rig respectively.

4.11.2.1 Real production machine

The differences, figure 4.11, between the simulation model generated simulation and the real system output generated using the real production machine are discussed below:

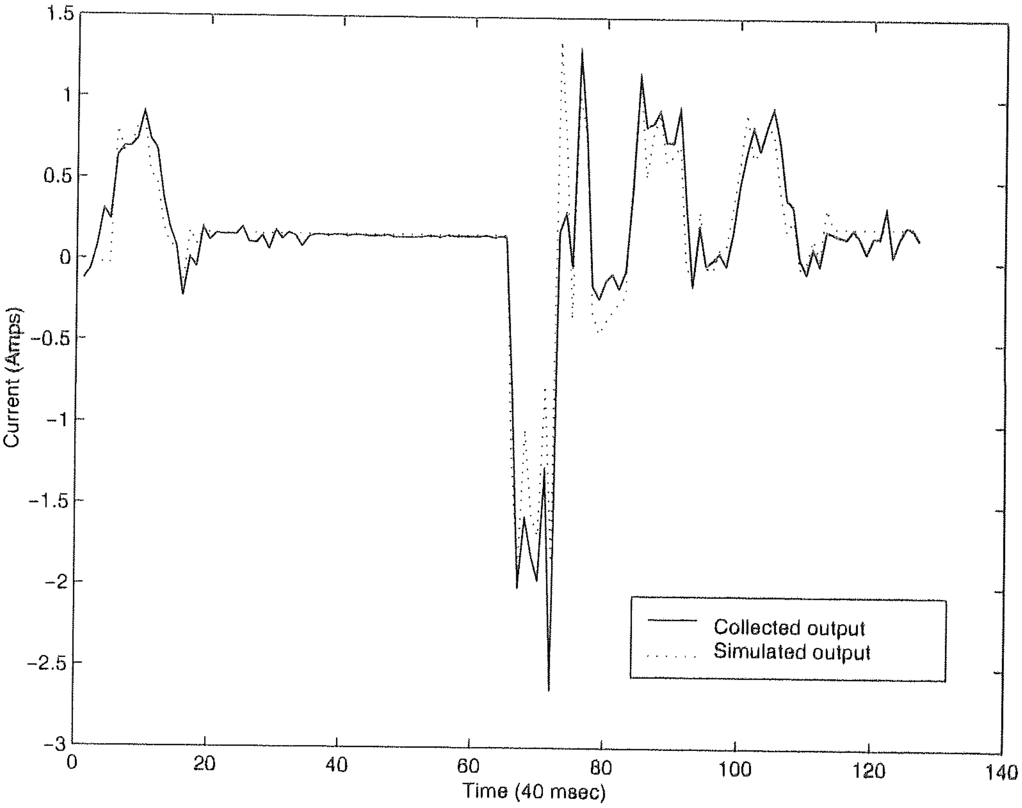

Figure 4.9: Simulated and collected (real) output time series, motion current signature, for the real production machine.

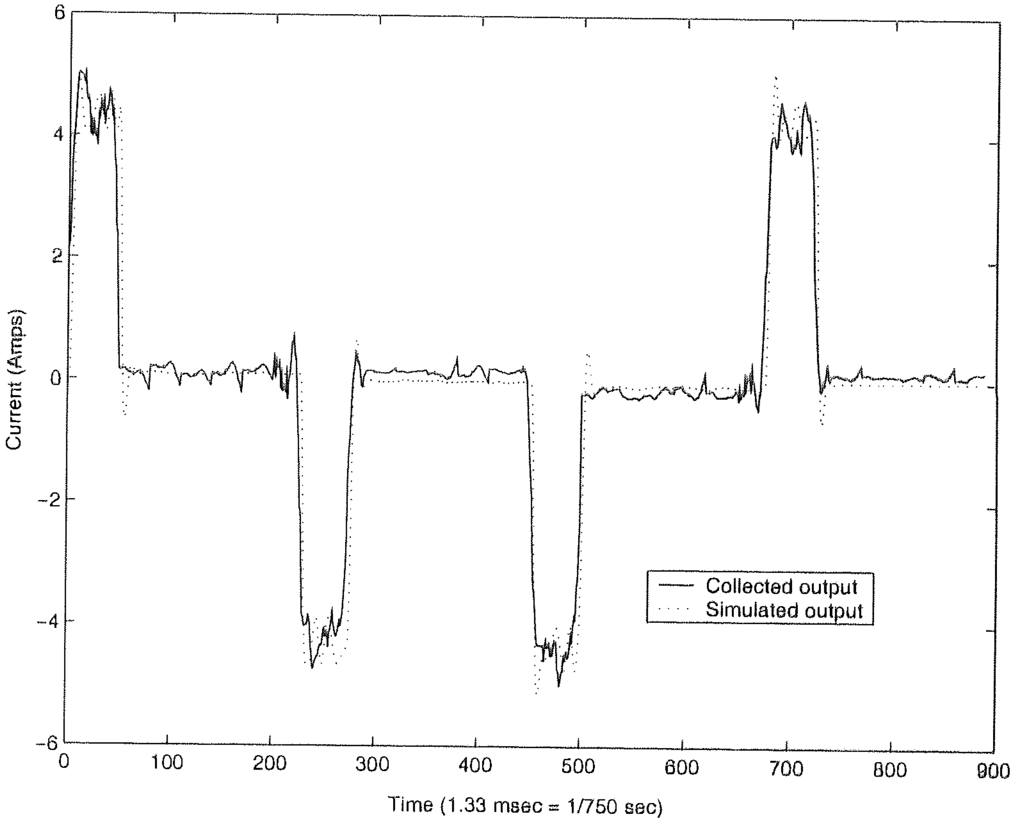

Figure 4.10: Simulated and collected (real) output time series, motion current signature, for the experimental test rig.

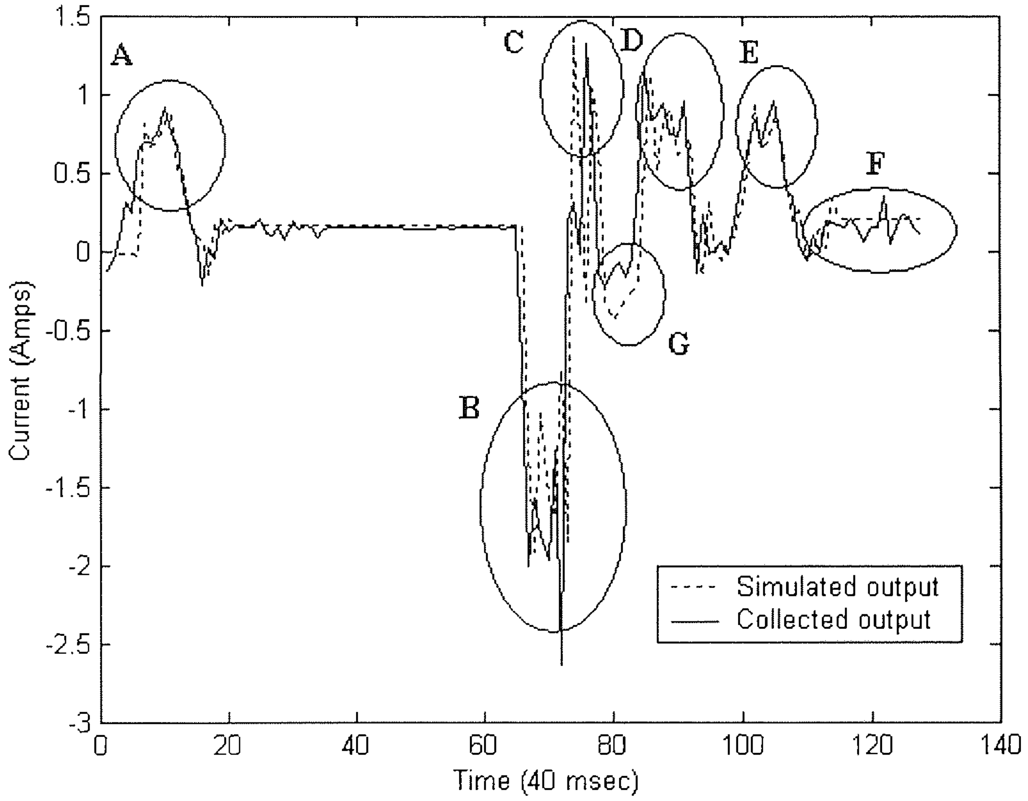

Figure 4.11: Simulated and real current signature of the real production machine showing the differences marked with circles and alphabets.

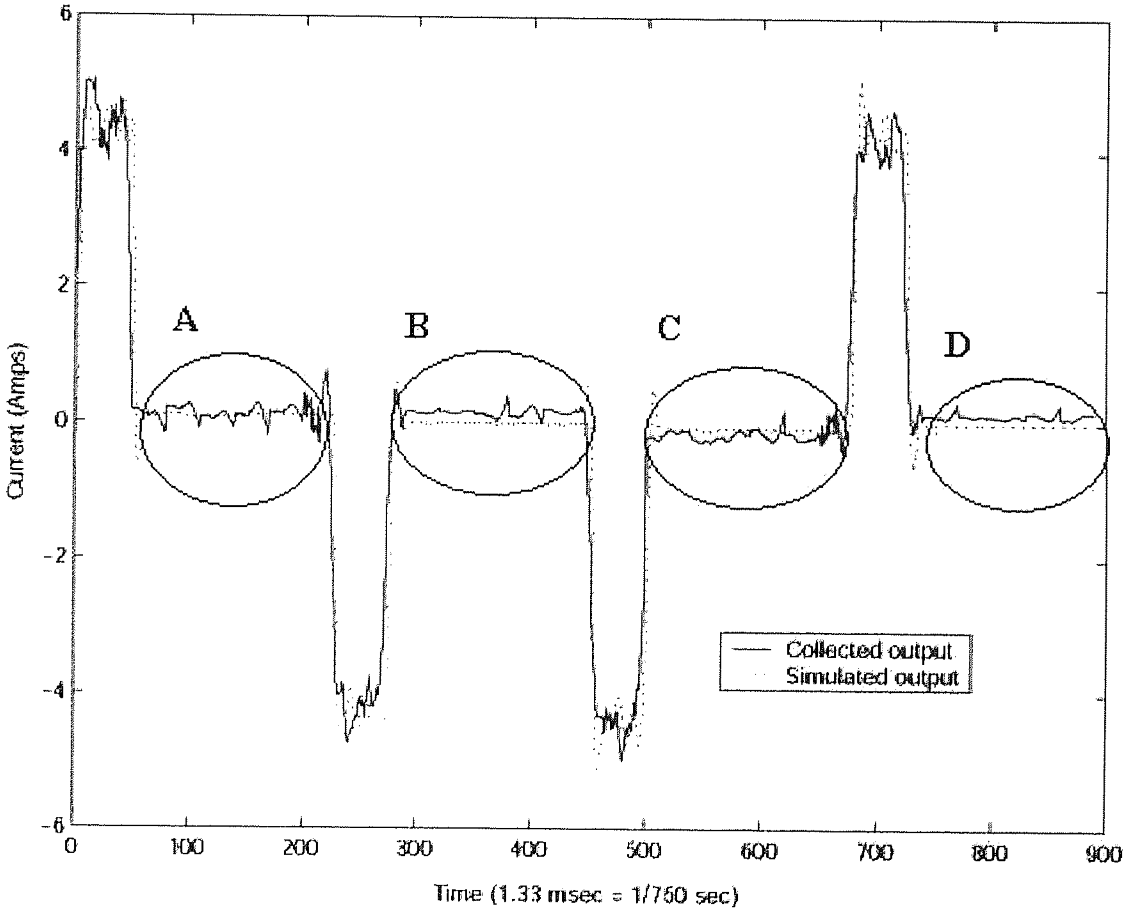

Figure 4.12: Simulated and real current signature of the experimental test rig showing the differences marked with circles and alphabets.

- A, B, D, E and F: These differences are due to the poles ripple curve. This kind of random behaviour totally depends upon the start position of the ripple curve. The simulation model is programmed to behave in a similar manner, but the real machines start at a different position leading to a random ripple curve. These differences can be removed by averaging the curve over a number of runs and are not due to the incorrect mapping of the machine system parameters.

- C and G: These differences are due to inaccurate friction torque and gravitation torque values. The friction torque and the gravitational torque, for modelling the real production machine in the simulation model, were calculated using the BJEST. As already mentioned in section 4.9, the BJEST is an approximation algorithm and only roughly estimates the values of the inertia, the friction torque and the gravitation torque. These differences can be eliminated by adjusting the values of the friction torque and the gravitational torque. A part of these differences is also caused due to the ripple curve.

4.11.2.2 Experimental test rig

The differences, Figure 4.12, between the simulation model generated simulation and the real system output generated using an experimental test rig are presented below:

- A: This difference is due to poles ripple curve and can be removed by averaging the curve over a number of runs.
- B and D: These differences comprise of two separate parts; a positive shift and a ripple effect. The positive shift is due to static friction and was removed by modelling the experimental test rig with a 0.2 Nm static friction torque. The ripple effect is due to the position correction.
- C: This difference contain two separate parts; a negative shift and a ripple effect. The negative shift is due to static friction, subtracted from the total torque because the motor is regenerating, and is removed by modelling the experimental test rig with a 0.2 Nm static friction torque. The ripple effect is due to the position correction.

The discussion establishes that all the differences between the simulated time series, generated using the simulation model, and the real system output time series, generated using the real production machine and the test rig, were due to the factors other than the inertia, the friction torque and the gravitational torque. This reinforces the argument that the simulation model is capable of mapping the motion current signature to the system parameters. However, the accuracy of the real-time predictive maintenance system has to be calculated by using it to detect the system parameter deviations in a real world scenario.

4.12 Application of the verification process

As discussed in section 4.5, the verification process of the simulation model is performed by the graphical comparison. Following, we describe the details of the graphical comparison technique to verify the operations of the simulation model.

4.12.1 Graphical comparison

Figure 4.9 and 4.10 are used for the graphical comparison of the simulation model in this thesis.

The following sections explain the graphical comparison of the simulated output against the real output of the real production machine and the test rig.

4.12.1.1 Real production machine

Figure 4.13 shows the scatter plot of the time series of the simulated and the real output obtained from the real production machine. A line was then fit to the scatter plot (figure 4.13). The equation of the line is given by

$$y = 0.0291 + 0.8439x. \tag{4.21}$$

Applying the validatory conditions given in section 4.5, we find

1. $0 < \beta_1 = 0.8439 < 1$, and
2. $0 < \beta_0 = 0.0291 < \mu = 0.1504$,

where $\mu_x = 0.1474$ and $\mu_y = 0.1535$. Therefore, μ is taken as 0.1504, the mean of μ_x and μ_y.

The equation given above satisfies the condition given in section 4.5 and justifies that the simulation model is a close match of the real system defined by a real production machine.

4.12.1.2 Experimental test rig

Figure 4.14 shows the scatter plot of the time series of simulated and the real output obtained from an experimental test rig. A line was then fit to the scatter plot (figure 4.14). The equation of the line is given by

$$y = 0.0607 + 0.9969x. \tag{4.22}$$

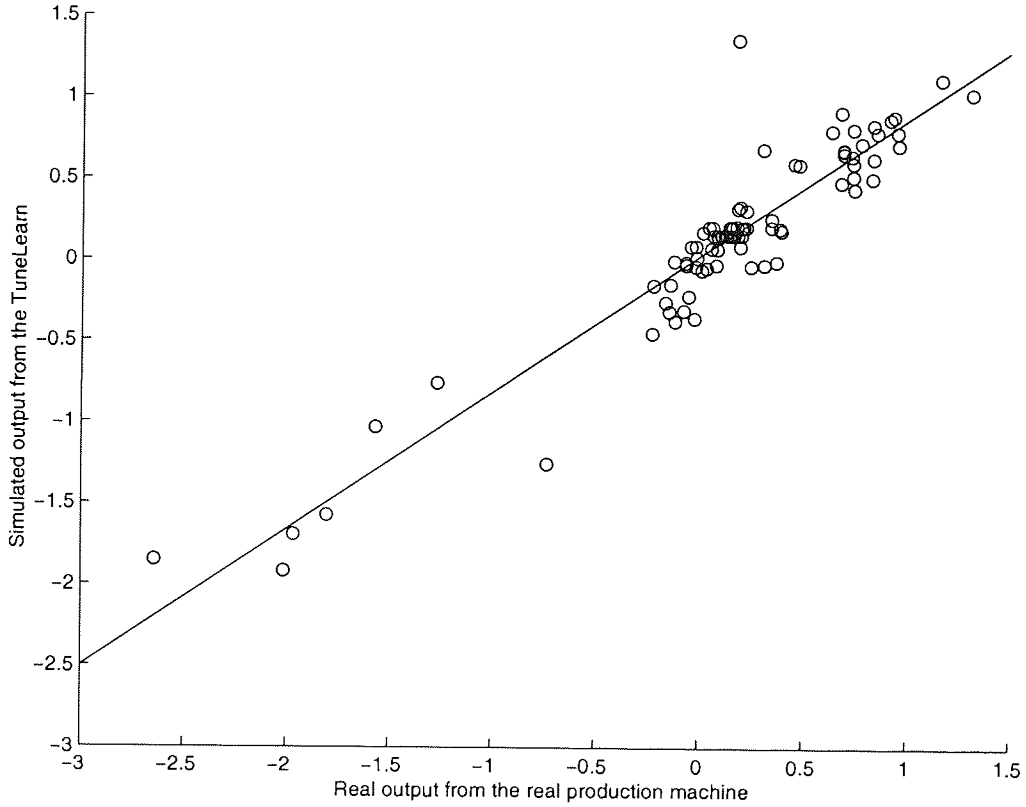

Figure 4.13: Scatter plot of the time series of simulated output and the real output obtained from the real production machine.

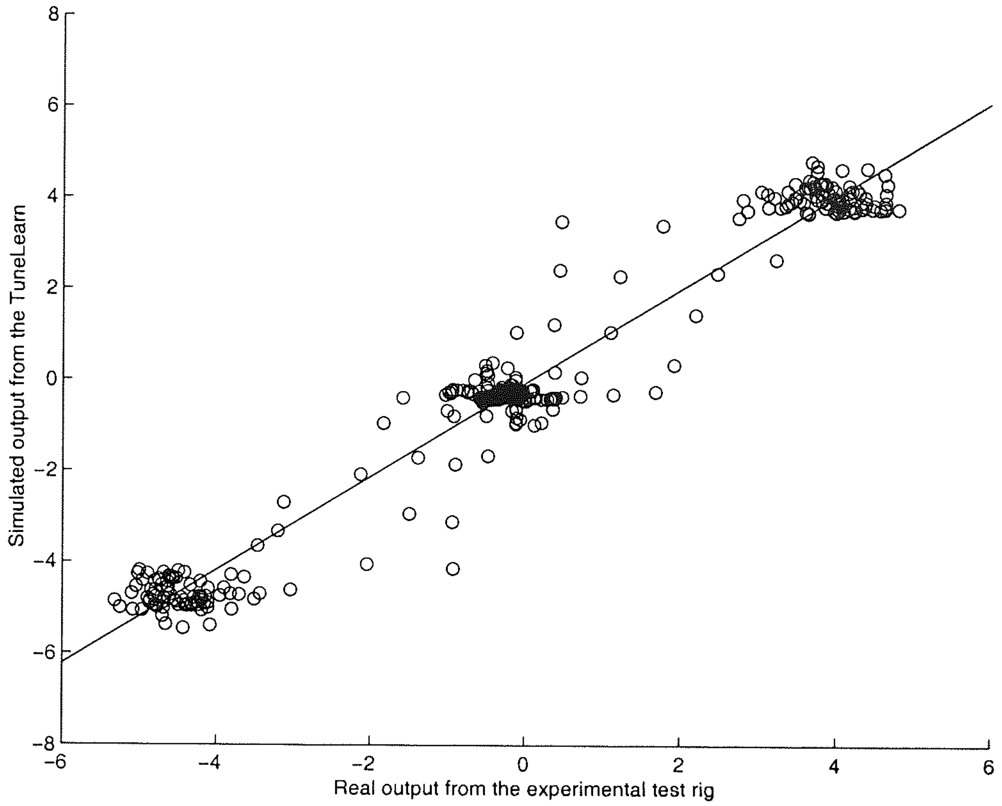

Figure 4.14: Scatter plot of the time series of simulated output and the real output obtained from an experimental test rig.

Applying the validatory conditions given in section 4.5, we find

1. $0 < \beta_1 = 0.9969 < 1$, and

2. $0 < \beta_0 = 0.0607 < \mu = 0.2752$.

where $\mu_x = 0.2416$ and $\mu_y = 0.3088$. Therefore, μ is taken as 0.2752, the average of μ_x and μ_y.

This proves that that simulation model is a close match of the real system defined by the experimental test rig.

The graphical comparison of the TuneLearn against the real production machine and the test rig has re-confirmed that the model is a close macro-dynamical match to a real system.

4.13 Summary

The accuracy of the real-time predictive maintenance system is a direct function of the validity of the simulated data, used for training the neural network. In this chapter, a simulation model was designed, implemented and validated to meet the training data requirements of the real-time predictive maintenance system. The objective validation of the simulation model, TuneLearn, was performed against an on-line production machine and an experimental test rig. It has been shown that the TuneLearn is capable of mapping the macro-dynamics of the motion current signature to the machine system parameters.

The sensitivity analysis resulted in a 99.5% accuracy to the deviations in the values of the inertia, the friction torque and the gravitation torque. The visual comparison was used to incorporate the expert knowledge in the validation process. Afterwards, the graphical comparison verified that the scatter plot of the simulated and the real output are within the acceptable limits. The simulation model cleared all the validation and verification tests used in this chapter.

However, the simulation model was not able to precisely map the micro-dynamics of the motion current signature. The failure to map the micro-dynamics could be due to a number of reasons, including:

- the underlying data generator of the motion current signature could be nonlinear and, hence, it is difficult to map the micro-dynamics of the motion current signature using linear algorithm;
- the simulation model algorithm is not capable of mapping the higher order terms, like backlash and resonance, which have a significant effect on the micro-dynamics of the motion current signature.

This motivates us to test the motion current signature for inherent nonlinear behaviour. The next chapter tests the nonlinearity of the motion current signature using nonlinear statistical techniques.

Chapter 5

Nonlinearity in motion current signature

5.1 Introduction

The previous chapter explained the design, implementation, validation and verification of the simulation model. The simulation model has been shown to be capable of mapping the machine system parameters against the motion current signature. The model was validated using the sensitivity analysis and the visual comparison using independent experts. The verification of the model was carried out using the graphical comparison. Although the statistical techniques showed that the model is accurate for its purpose, the visual comparison by experts pointed out that the model is not performing accurately to map the micro-dynamics of the motion current signature. As the simulation model does not have the implementation of the corresponding non-linear algorithm, the simulation model is unable to replicate the micro-dynamics of the motion current signature.

Whilst validating the simulation model, a reverse algorithm called BJEST (Bansal-Jones Estimation), for estimating the machine input parameters using the motion current signature, was designed and proven to be successful in estimating the macro-dynamics of the motion current signature (section 4.9). Additionally, according to Bishop (Bishop, 1995), the performance of the neural network largely depends upon a number of factors including:

- the quality of the training data;
- quality and type of the preprocessing of the input data;
- the type of the neural network technique adopted;
- the training methodology used.

The success in estimating the macro-dynamics of the motion current signature using BJEST, referred to as the reverse algorithm here onwards, and the dependence of the performance of a neural-network approach upon training data, motivates us to enhance the reverse algorithm to incorporate nonlinear characteristic. This chapter is an attempt to verify the non-linear characteristic of the motion current signature and , upon verification, to enhance the reverse algorithm to incorporate the nonlinear characteristics of the motion current signature. However, the first step is to statistically test the motion current signature for nonlinearity.

In this chapter, we perform a series of experiments to explicitly test for the presence of nonlinear dynamics in the motion current signature.

Sections 5.2 and 5.3 detail the basic terminology necessary to understand the nonlinear time-series techniques. Section 5.4 explain the testing algorithm for nonlinearity which is followed by section 5.5, which presents the results of the surrogate data testing.

5.2 Linear vs. Nonlinear systems

An important problem in engineering and physical sciences is that of identifying a model for a physical system (or process) given observations on the input and output of the system. The study of the time series can be divided into two main categories depending on the nature of the transition function f. The first class, where the transition function is linear, is accordingly termed linear systems. Similarly, the term nonlinear systems is used to refer to cases where the transition function is nonlinear (Kantz and Schreiber, 2004).

Why the division into these two classes? At least part of the reason is historical: it is only recently that the ability to effectively analyze nonlinear systems has become widely attainable.

The linearity can be defined as:

Suppose $y_1(t), y_2(t)$ are the outputs corresponding to $x_1(t), x_2(t)$ respectively. Then the system is said to be linear if,and only if, a linear combination of the inputs, say $\lambda_1 x_1(t), \lambda_2 x_2(t)$, produces the same linear combination of the outputs, namely $\lambda_1 y_1(t), \lambda_2 y_2(t)$, where λ_1, λ_2 are any constants.

The nature of linear dynamics is such that it permits only one of three classes of behaviour:

- Convergence to a stationary point;
- Sustained oscillation at one or a number of fixed frequencies;

- Exponential growth (instability).

As systems belonging to the first or the last classes are of little interest, the focus of linear dynamical systems analysis has been on the study of frequency components. Under this paradigm, the signal of interest is described as being the combination of a large number of separate frequency generators. For long periods of time, the analysis of dynamical systems using linear tools was probably the only really effective method of characterizing noisy data. Consequently, a large body of research has grown up around a number of methods which either implicitly or explicitly assume linear dynamics. These include all methods related to Fourier transforms, frequency and power spectral methods, as well as linear modelling techniques such as auto-regressive and ARMA (Auto Regressive Moving Average) models and Kalman filters. Certainly, the characterization of time series via power spectra has had a long and distinguished history of successful applications in an exceedingly wide variety of research areas and remains one of the most reliable methods for time series analysis. At the same time, it is not true to say that nonlinear systems have received no attention at all. The last few years have seen a massive expansion in the scope of research into systems that are not adequately handled by these traditional methods. The existing body of linear tools has been augmented by a host of new approaches that are capable of modelling systems that show nonlinear behaviour but not chaos (Kantz and Schreiber, 2004).

The nonlinear time series analysis techniques strictly assume that the system has been tested for nonlinearity, which stresses the importance of nonlinearity detection.

5.3 Basic concepts and terminologies

Before we proceed with any more explanation, it is in order to explain a few basic terms associated with system analysis 5.1. A *system* will be taken to be a set of variables which change over time, and which have a some natural connectedness (such as all being measured from the same individual). A *dynamical system* will be defined to be a system in which the present state (the values of all of the variables and all of their derivatives) is somehow dependent on previous states of the system. A *deterministic system* will be taken to be a system in which the present state is entirely dependent on previous states of the system. A *linear system* is a system in which all of the dependence of the current state on previous states can be expressed in terms of a linear combination. A *linear stochastic system* is a system in which all of the dependence of the current state on previous states can be expressed in terms of a linear combination and the residual unpredictable portions can be expressed as additive, independent, identically distributed, random variables.

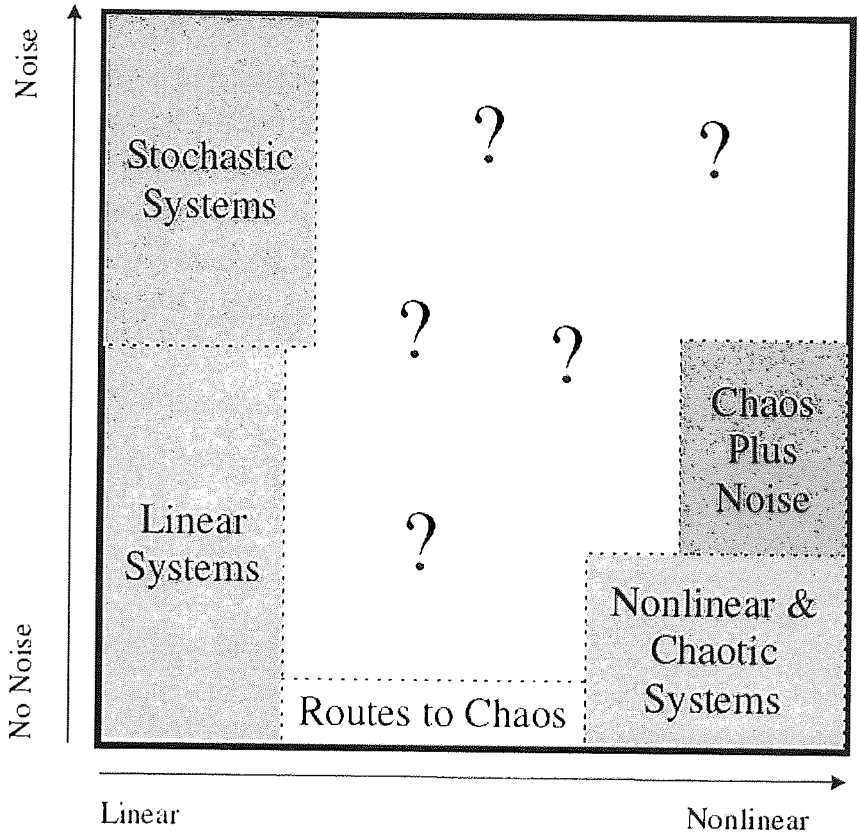

Figure 5.1: A diagram of the types of systems which may potentially underlie physiological and psychological phenomena. Note that there are many possibilities of which little is known. (After a diagram presented by Thomas Schreiber at the Nonlinear Techniques in Physiological Time Series conference in Dresden, October 1995.)

A *nonlinear system* is a system in which the dependence of the current state on previous states cannot be expressed entirely as a linear combination; even if some of the dependence can be captured in a linear combination of the previous states, something extra is required to capture all of the dependence. That extra something is frequently an interaction between variables, i.e. a multiplicative term involving the previous states. A *nonlinear stochastic system* exists somewhere among the question marks in Figure 5.1; very little is known about these systems.

If an infinitesimal difference in the initial conditions (initial state) of two realizations of the same dynamical system causes an exponential divergence between the trajectories of the two systems, then that system is said to have *sensitive dependence on initial conditions.* A *chaotic system* is a nonlinear dynamical system which exhibits sensitive dependence on initial conditions. Linear dynamical systems never show sensitive dependence on initial conditions. Some nonlinear dynamical systems show sensitive dependence with some initial conditions and no sensitive dependence with other initial conditions. These systems are sometimes chaotic and sometimes not chaotic depending on their state, and can be changed from a nonchaotic regime to a chaotic regime and back again by external manipulations of their variables.

The next section presents the test for nonlinearity performed on the motion current signature to test the presence of nonlinear behaviour.

5.4 Testing for Nonlinearity

Given an arbitrary time series generated by an unknown process, it is important to test whether the time series needs to be modelled as a deterministic system and given determinism, whether a linear deterministic system is sufficient to model the measured data or if a nonlinear analysis is required. If there is no evidence of deterministic behaviour, there is little point in pursuing a dynamical systems model; a correlational model will fit the measured data just as well. If there is no evidence of nonlinearity, then there is little point in pursuing a nonlinear dynamical systems model when a linear model will fit the measured data just as well. In almost every case, the simpler model is to be preferred over the more complex one (Golyandina, Nekrutkin and Zhigljavsky, 2001).

Making these determinations requires performing tests to verify the nature of the data. Normal statistical theory will not help in this case, since the calculation of standard errors requires some model of the process which generated the data. The test for nonlinearity of the motion current signature performed in this chapter is an attempt to determine such a model. A variant on the bootstrap method (Efron, 1979*a*,*b*) for empirically determining the distribution of a statistic has been used in order to overcome this problem (Horn, 1965). Theiler, et. al. (1992) (Theiler et al., 1992) have called this the method of surrogate data.

Surrogate data testing is a method used to test for nonlinearity in the motion current signature (Theiler et al., 1992). The motivation for pursuing this avenue of research are:

- The simulation model, which was based upon the linear modelling techniques, has failed to replicate the micro-dynamics of the motion current signature (section 4.11.2 and 4.13);
- Prior knowledge regarding the motion current signature and the effect of higher order factors, such as friction and backlash, indicate the presence of nonlinearity.

Nevertheless, the reasons given above can prove insufficient to justify the use of nonlinear techniques for analyzing motion current signature. The fact that a system contains nonlinear components does not prove that this nonlinearity is also reflected in a specific signal measured from that system. Due to these reasons, formal validation of the presence of nonlinearity is desirable.

The simplest way to test for nonlinearity is to calculate some general measure of nonlinearity or chaos, examples being various high order statistics and Lyapunov exponents (Kantz and Schreiber, 2004). One popular method has been to calculate both the correlation dimension and first Lyapunov exponent - a combination of convergence of the correlation dimension and a positive exponent was then taken as an indication of nonlinearity (Golyandina, Nekrutkin and Zhigljavsky, 2001). However, it is now generally accepted that measures such as these are not sufficient, by themselves, to establish chaotic or nonlinear behaviour in the data (Govindan, Narayanan and Gopinathan, 1998; Theiler et al., 1992). In particular, the measurement of such statistics is often prone to noise contamination and requires large input data sets, which increases the computational overhead. In addition, using longer data sets increases the likelihood of encountering nonstationarities. Errors associated with the acquisition of data such as inappropriate sampling frequency, noise filtering and digitization error can all lead to erroneous values of these statistics being returned. Finally, even if we were able to determine these values with sufficient accuracy, the actual distribution for the nonlinear statistic in question is generally not known except for the simplest of models.

The method of surrogate data analysis (Theiler et al., 1992) solves some of these problems by providing a suitable statistical framework in which nonlinearity tests may be performed more reliably. It is based on the principle of Monte Carlo methods: the idea is to sample in the space of possible time series matching some carefully chosen null hypothesis, then perform a standard statistical t-test to reject this hypothesis. The basic idea is to generate a population of null hypothesis data sets (surrogates) appropriate to the test of interest and then use the distribution of some nonlinear invariant (such as the fractal dimension of a time delay embedding) of these surrogates to estimate a confidence interval around the mean of the invariant.

Then if the nonlinear invariant of the measured data lies outside the confidence interval of the surrogates, the null hypothesis is rejected.

Thus, the first step is to specify a null hypothesis against which the nonlinearity of the data is tested. Hypothesis testing (Papoulis, 1991) is a method used in the statistical analysis to state the alternative (for or against the hypothesis) which minimizes certain risks. Essentially, an assumption about the data, known as the null hypothesis (H_0), which is expected to contradict, is made.

The observed dynamical system could fall into one of the four categories:

1. linear deterministic (e.g. Newtonian, undamped pendulum, sinewave);
2. nonlinear deterministic (e.g. Lorenz, Henon map);
3. linear stochastic (e.g. a linear Markov model);
4. nonlinear stochastic (e.g. a nonlinear Markov model).

In this chapter, the null hypothesis, H_0, for motion current signature is that the observed data is an element of a linear stochastic Gaussian process. After the null hypothesis is identified, a nonlinear parameter is extracted from the data. In theory, this can be anything, provided it is able to reject data not belonging to the class of models defined by the null hypothesis. Lyapunov exponents and correlation dimensions have been popular choices (Unsworth et al., 2001). The most important requirement when choosing this parameter, however, is that it should have a relatively localized distribution when applied to data, that conforms to the null hypothesis. In other words, the value of this quantity should be sensitive to changes near regions of interest since this will increase the ability to reject data with different characteristics from the null hypothesis.

A cautionary note about null hypothesis testing is always in order. It is wise to remember that the null hypothesis can only be rejected, it cannot be accepted. Thus the surrogate data methods cannot rule out the presence of a nonlinear dynamical system as the generating process for a measured time series. If the null hypothesis cannot be rejected, then the conclusion is only that the measured time series does not support the conclusion of a (nonlinear) dynamical system. This may be because the underlying process is truly not a (nonlinear) dynamical system, or it could be because the surrogate data test lacks sufficient power in the measured time series. Since there is no model for the standard errors of the test, one cannot properly calculate the power of these surrogate data tests.

5.4.1 Generating surrogate data

Constructing the surrogate data sets can take many forms and will naturally vary depending on the particular null hypothesis that one desires to test. The first class of null hypotheses that need to be tested is that there is no dynamical process operating in the data. One possible surrogate data set considers the null hypothesis that the data are independent, identically Gaussian distributed random variables. A pseudo-random number generator can be used to generate a sample of surrogate time series that each have the same mean and variance as the measured time series.

Another variant of this null hypothesis of no dynamical system preserves the amplitude distribution of the original signal, but destroys the time relations between the samples. In this case, the easiest method to generate a sample of surrogate time series is to shuffle the ordering of the samples in the measured time series. Each surrogate so generated will have all of its time dependencies removed, but the original amplitude distribution of the measured time series will be preserved.

The second class of null hypotheses are that the data were generated by a linear stochastic system. One method of generating a surrogate data set for this null hypothesis is to fit an autoregressive model to the data and then generate a sample of surrogate time series by iterating that model. There is a problem with this method: there are many possible autoregressive models which would need to be fit in order to find the best fitting alternative. A clever way around this problem was suggested by Osborne et. al. (Osborne et al., 1986). The Fourier transform of the time series is applied, a uniform random number between 0 and 2π is added to the phase spectrum of the Fourier series, and then the inverse Fourier transform is applied. The effect of this method is to generate a surrogate which shuffles the time ordering of the data while preserving the linear autocorrelations in the time series. The resulting surrogate fits the null hypothesis that the time series is a linear stochastic process (colored noise).

A refinement of this method involves iteratively phaserandomizing the Fourier series and then correcting the resulting amplitude distribution for minor anomalies introduced by the phaserandomization process (Schreiber and Schmitz, 1996). This process is called polished surrogates and is the method which will be used to generate the linear stochastic null hypotheses for the present work.

In brief, the idea is to directly optimise the spectrum of a randomly generated time series such that it matches that of the original data. First, an initial data set is chosen either by generating some random numbers, or by using the phase randomisation method mentioned above. Next, the power spectra of the new set is optimised via a two step iterative procedure:

1. The Fourier transform is performed on the surrogate data, and the magnitude of the Fourier coefficients are set to that of the original data.

2. The inverse Fourier transform is calculated. However, changing the Fourier amplitudes will have affected the distribution of the data. To remedy this, a two step procedure is applied: firstly, both the candidate surrogate set and the real dataset are arranged so that the largest data points come first, second largest second, and so forth. However the original time indices of the surrogate set are retained, and are used to rearrange the ordered version of the original data set. This ensures that the static distribution of the new time series now matches that of the original data set, while the dynamical ordering of the data points are changed to match the surrogate dataset. However, this reordering will again change the power spectra. This so step one is carried out again and so on. With each iteration, the changes required become increasingly smaller until convergence is reached.

Further, as recommended in (Kugiumtzis, 1999), we terminated the training after the adjustment of the spectra, rather than the amplitude distribution, as this method results in surrogate data with the same autocorrelation structure as the data. The slight deviation in the amplitude spectra that this results in was deemed to be acceptable. For a more detailed description of this algorithm, or of surrogate data testing in general, please refer to (Schreiber and Schmitz, 1996; Theiler et al., 1992).

The next sub-section describes the discriminating statistics of the test.

5.4.2 Discriminating statistics of the test

A discriminating statistic is the aspect of the data that has to be tested with the null hypothesis. In principle, any nonlinear statistic which assigns a real number to a time series can be used. Higher order autocorrelations such as $\langle S_n S_{n-1} S_{n-2} \rangle$ are cheap in computation but quantities inspired by nonlinear science seem to be more popular because they are particularly powerful if the data set has a nonlinear deterministic structure.

A particularly higher-order autocorrelation, $\langle S_n S_{n+1}^2 - S_n^2 S_{n+1} \rangle$, which measures time asymmetry, is the most powerful measure of nonlinearity (Kantz and Schreiber, 2004; Schreiber and Schmitz, 1996). Due to this reason, time asymmetry will be used as the discriminating statistic with the intention of testing different aspects of the data. It uses the fact that the statistics of linear stochastic processes are always symmetric under time reversal, since the power spectrum itself does not contain any information about the direction of time. There are a number of ways for measuring time asymmetry but one of the more useful methods, and also the one which is used in this paper, is given by the following equation:

$$\phi_{rev} = \frac{[(x_n - x_{n-\tau})]^3}{[(x_n - x_{n-\tau})^2]^{3/2}}, \tag{5.1}$$

where ϕ_{rev} is the time asymmetry measure of n generated surrogate data sets and τ is the delay used for reconstruction (Kantz and Schreiber, 2004).

5.4.3 Evaluating significance

Evaluating the significance of the result is one of the most important step in the test for non-linearity. The most common method is to calculate the deviation of the data from the mean in terms of numbers of standard deviation. However, this method depends on the assumption that the distribution of the statistic is according to the t distribution, which is either difficult to prove or untrue (Kantz and Schreiber, 2004). This paper used the method proposed in (Theiler et al., 1992), which is based upon the idea of rank statistics. The basic idea being: a residual probability α of a false rejection is chosen and then, for a double-sided test such as the time asymmetry measure, $\frac{2}{\alpha}$ surrogate data sets are needed to obtain $(100 - \alpha)\%$ confidence in the results. Hence, in order to attain the commonly used 95% significance mark, 40 surrogate sets for the time asymmetry test have been used.

The next section details the experimental results of the surrogate data testing performed on motion current signature.

5.5 Results of surrogate data testing

The experiment was carried out in a number of stages, they are:

- **Step 1:** The motion current signature, referred to as the original signature from here onwards, from a real production machine was collected. Table 5.1 presents the values of the motor and drive parameters of the production machine used for the experiment.

- **Step 2:** The discriminating statistic of the original signature was computed.

- **Step 3:** The surrogate data sets (40 in total) were generated.

- **Step 4:** Subsequently, the discriminating statistic of all the surrogate data sets was computed.

- **Step 5:** The histogram of the discriminating statistic of all the surrogate data sets was plotted against the discriminating statistic of the original signature. The histogram

Table 5.1: Motor and drive parameters of the production machine.

Parameter Name	Value	Units
Motor parameters		
Motor Type	H-4075-R0H	-
Motor Inertia	0.00068	kgm^2
Motor torque constant	0.74	Nm-A
Motor peak torque	30.0	Nm
Motor poles	6	-
Motor rms torque	10	Nm
Motor damping factor	0.068	$\frac{Nm}{krpm}$
Motor friction torque	0.14	Nm
Motor maximum speed	3000	rpm
Drive parameters		
Drive Type	1398-DDM-30	-

presents the distribution of the test statistic under the assumption of the null hypothesis (T_s), and the vertical line denoting the value of the test statistic for the original signature (T_0).

- **Step 6:** If T_0 is not drawn from the distribution of T_s, the original signature can be concluded to be a nonlinear signal with a confidence level of 95%.

The experiment was repeated for 2 different axis of the production machine.

Figure 5.2 shows the original signature collected from axis 3 (chosen due to the reason described in section 4.10) of the production machine used for the experiment. Figure 5.3 details 5 of the 40 generated surrogate data sets. Figure 5.4 presents the distribution of the test statistic for surrogate data sets. The vertical line denotes the test statistic for the original signature.

As mentioned above, the surrogate data set is generated as an element of linear stochastic Gaussian process and if the test statistic value of the original signature does not match the test statistic distribution of the surrogate data sets, it can be safely concluded that the original signature is drawn from a non-linear stochastic Gaussian process with a 95% confidence level.

According to figure 5.4, the test statistic T_0 of the original signature is not drawn from the same distribution of the test statistic T_s of the surrogate data sets. This establishes the nonlinearity of the motion current signature with 95% confidence level.

The same experiment was repeated for axis 6 of the production machine and the results are presented in figure 5.5,5.6 and 5.7.

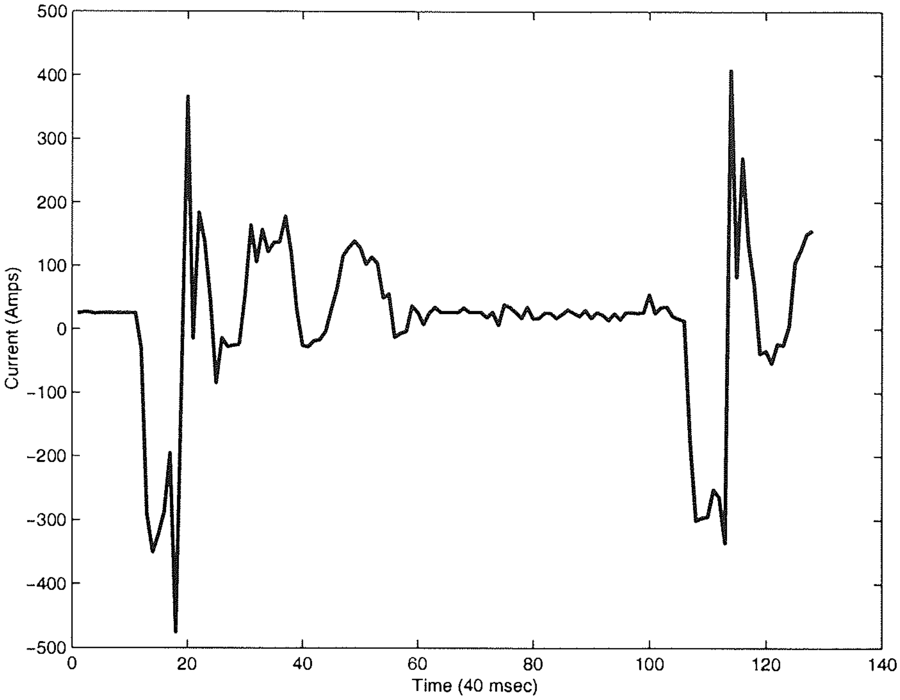

Figure 5.2: The original signature collected from axis 3 of the production machine.

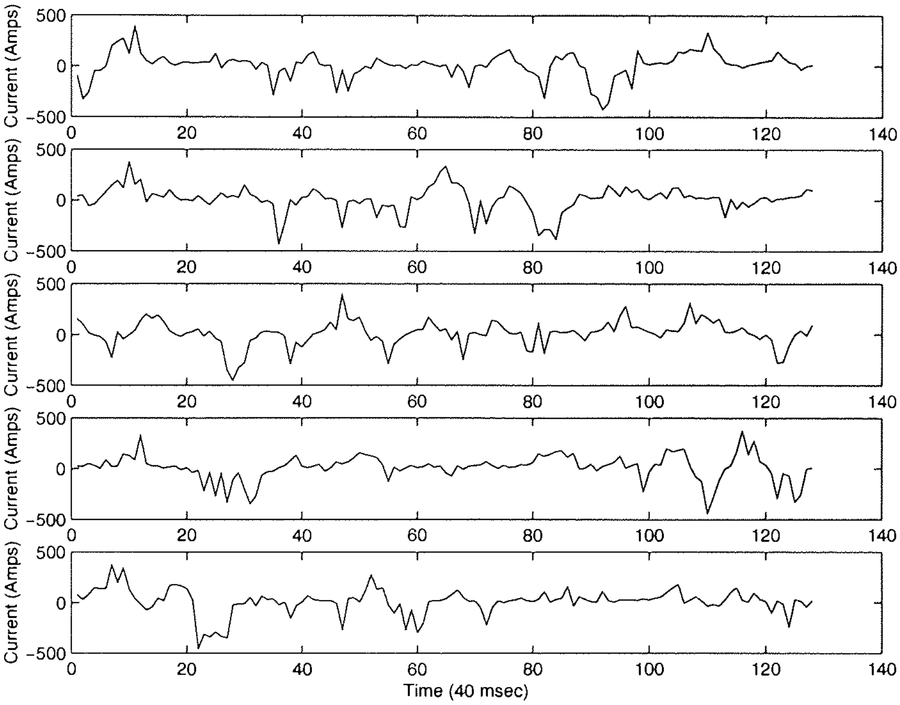

Figure 5.3: 5 of the 40 surrogate data sets generated for the experiment by directly optimizing the spectrum of a randomly generated time series such that it matches that of the original data in figure 5.2.

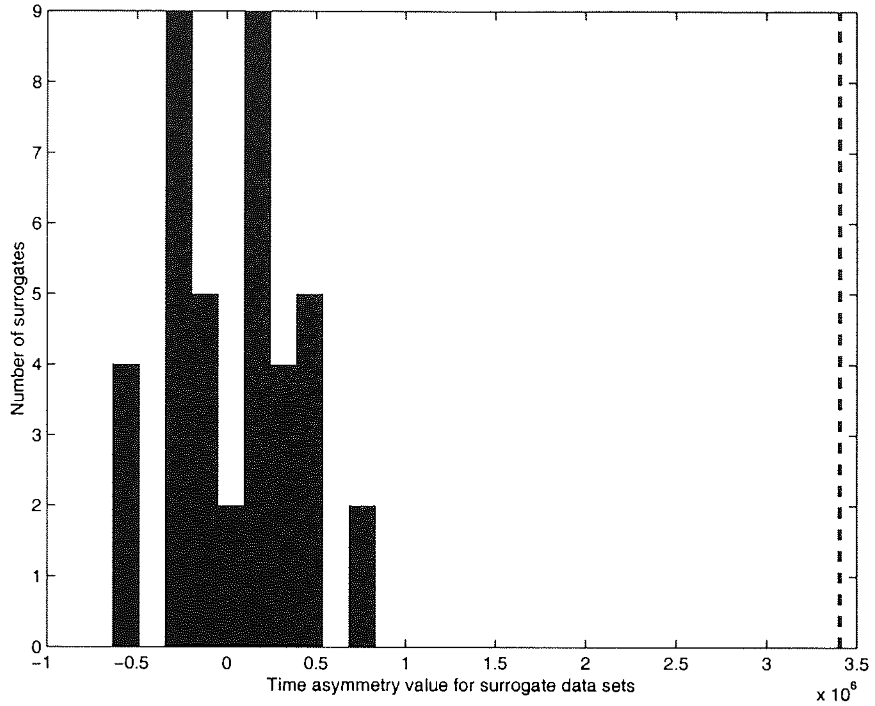

Figure 5.4: Distribution of the test statistic (ϕ) of the surrogate data sets against the test statistic (ϕ) of the original signature from axis 3.

Experiment on axis 6 (Figure 5.7) confirms the outcome of the results from axis 3 (Figure 5.4), and establishes the nonlinearity of the motion current signature with 95% confidence level.

5.6 Summary

The objective of this chapter has been to present a brief introduction to relevant elements of the time series analysis. In particular we have tried to focus on the idea of the motion current signature as an unobservable time series. An important issue of the presence of nonlinearity in the motion current signature has been addressed. Surrogate testing methodology has been applied to ascertain the presence of nonlinear components in the motion current signature. The results of the tests on the sample motion current signature are presented and appear to indicate that the motion current signature are a mixture of linear and stochastic component but with definite nonlinear components mixed in it. It has been shown that the motion current signature obtained from a production machine does not have the same distribution as a linear stochastic process, thus confirming the presence of nonlinear components in the signature. This chapter concludes that further analysis of the motion current signature must involve nonlinear analysis to obtain better estimation of the system parameters.

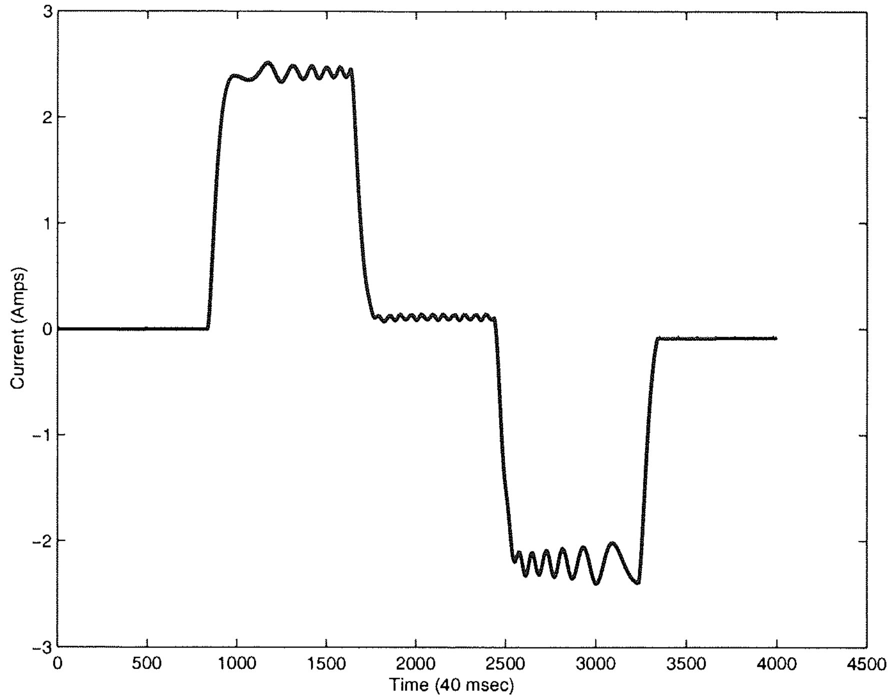

Figure 5.5: A motion current signature collected from axis 6 of the production machine.

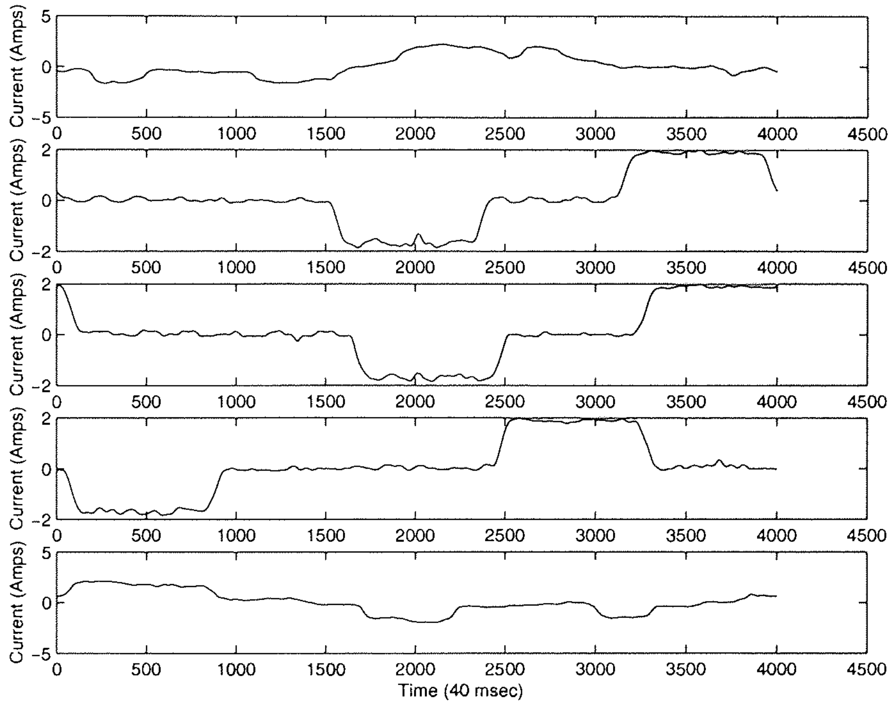

Figure 5.6: 5 of the 40 surrogate data sets generated for the experiment by directly optimizing the spectrum of a randomly generated time series such that it matches that of the original data in figure 5.5.

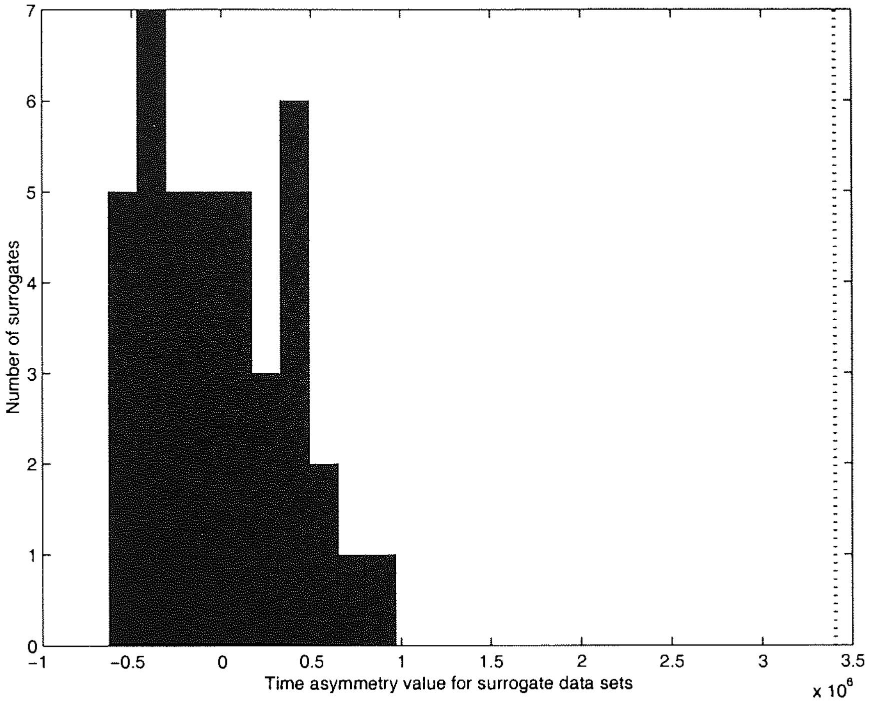

Figure 5.7: Distribution of the test statistic (ϕ) of the surrogate data sets against the test statistic (ϕ) of the original signature from axis 6.

The next chapter deals with nonlinearity reduction, so that the linear reverse algorithm (presented in section 4.9) can be applied on the filtered motion current signature to estimate the system parameters.

Chapter 6

Nonlinear noise reduction and BJEST

6.1 Introduction

After testifying the presence of nonlinearity in the motion current signature in the previous chapter, the predictive analysis for the real-time predictive maintenance system has to be enhanced to incorporate nonlinear behaviour. The broad-band spectrum of signals from nonlinear systems usually makes traditional linear algorithms incapable of mapping nonlinear behaviour. Due to this reason many researchers have studied noise reduction methods applicable to nonlinear systems (Davies, 1994; Kostelich and Yorke, 1988, 1990; Kostelich and Schreiber, 1993). This motivates the enhancement of the reverse algorithm to include nonlinear noise reduction as a forestep before the linear analysis. In simpler terms, the idea is to reduce the nonlinear noise of the motion current signature and then apply the linear reverse algorithm, BJEST, to estimate the values of the machine system parameters (figure 6.1.

In this chapter, section 6.2 details the basics of the nonlinear noise reduction technique. Section 6.3 details the Schreiber noise reduction technique used for reducing the noise from the motion current signature. Section 6.4 and 6.5 present the results of application of the noise reduction and linear reverse algorithm technique, respectively.

6.2 Nonlinear noise reduction

The task of noise reduction is a central theme in a wide variety of fields. Methods for optimal signal/noise separation include linear signal processing techniques such as Wiener filtering and Kalman filtering as well as nonlinear methods such as manifold decomposition and phase space projection (Kantz and Schreiber, 2004). There is often a great deal of overlap in the underlying

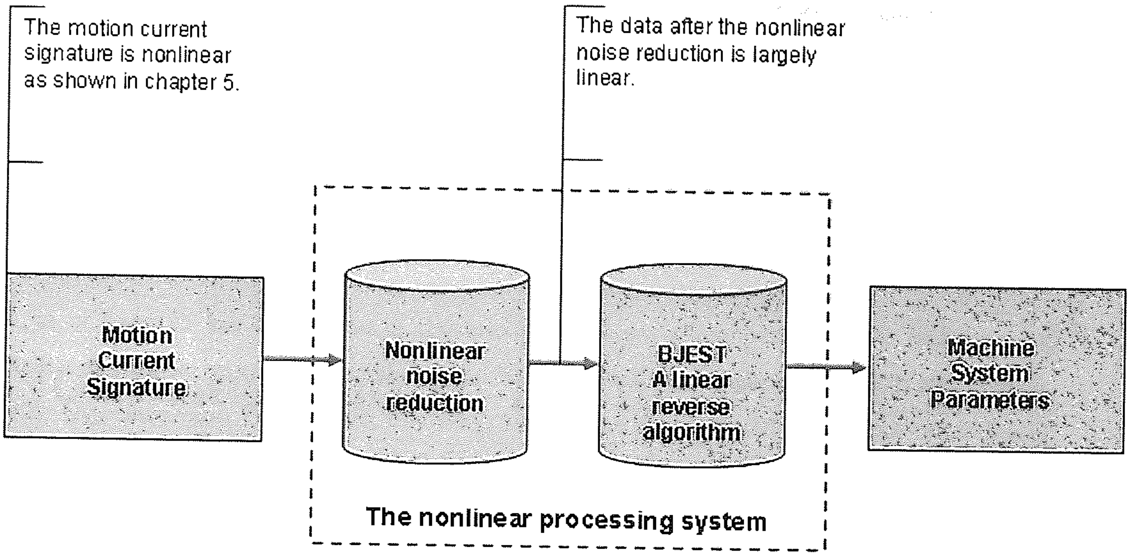
The motion current signature is nonlinear as shown in chapter 5.
The data after the nonlinear noise reduction is largely linear.
Motion Current Signature
Nonlinear noise reduction
BJEST A linear reverse algorithm
Machine System Parameters
The nonlinear processing system

Figure 6.1: The idea behind the application of the nonlinear noise reduction and BJEST techniques in the real-time predictive maintenance system.

concepts on which methods from separate areas are based, and significant improvements in both understanding and technique may be gained through cross-disciplinary study (Johnson and Povinelli, 2005).

There are two types of noises:

- measurement noise, which means corruption of data in observation process without interfering with the dynamics itself;
- dynamical noise, which denotes the perturbation of the system coupled to dynamics, occurring at each time step.

The noise reduction technique used for filtering the noisy signal is different for different type of noise. Owing to the presence of nonlinearity in the motion current signature due to higher order factors, such as backlash, resonance and friction, this chapter will concentrate on noise reduction techniques for nonlinear dynamical noise.

Noise reduction means decomposition of a time series value into two components, the signal and random fluctuations. In nonlinear noise reduction, it is assumed that the data is an additive superposition of the two different components and these components are distinguishable by some objective criterion.

A classical statistical tool for obtaining this distinction is the power spectrum. Random noise has a flat, or at least a broad spectrum, whereas periodic or quasi-periodic signals have sharp spectral lines (Kantz and Schreiber, 2004). After both components have been identified in the spectrum, a Wiener filter (Kantz and Schreiber, 2004) can be used to separate the time series accordingly. This approach fails for deterministic chaotic dynamics because the output of such systems usually leads to broad band spectra itself and thus possesses spectral properties generally attributed to random noise.

A lot of research has been done in the area of noise reduction pertaining to real-time and chaotic dynamical systems (Broomhead and King, 1986; Grassberger et al., 1993; Kantz et al., 1993; Ott, Sauer and Yorke, 1994; Pikovsky, 1986). The simplest nonlinear noise reduction algorithm, which is capable of deterministic and stochastic chaotic dynamical analysis, are based upon the idea of replacing the central coordinate of each embedding vector by the local average of the coordinate (Schreiber, 1993*b*). This amounts to a locally constant approximation of the dynamics and is based on the assumption that the dynamics is continuous. A noise reduction algorithm, termed as Schreiber noise reduction (Schreiber, 1993*b*), has proven to be an effective reduction technique for the nonlinear data series. The next section explains the technique in brief.

6.3 Schreiber noise reduction

Schreiber noise reduction couples a noise level determination method and a noise reduction method. The level of noise in the data is first estimated using the method of Schreiber (1993) (Schreiber, 1993*a*), and method of Schreiber (1993b) (Schreiber, 1993*b*) is then employed to reduce noise. A brief account of the two methods (noise determination and noise reduction) used in the present study is presented below.

6.3.1 Noise level determination

The noise level determination method of (Schreiber, 1993*a*) is derived under the assumption that the data are samples from a low-dimensional attractor contaminated by noise, which is Gaussian. A simple analytical expression is derived for the rescaled correlation integral, that is a function of the noise level as well as the dimension of the underlying chaotic attractor.

The method is based on the correlation integral defined as

$$C(r) = \frac{2}{T(T-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} H(r - |x_i - x_j|), \tag{6.1}$$

where H is the Heaviside step function, and |.| denotes the maximum norm, $T = N - m + 1$, m is the embedding dimension, and $x_i = (x_i, x_{i+1}, ..., x_{i+m-1})$.

The correlation integral counts the number of pairs of points (Y_i, Y_j) on the attractor whose distance apart is smaller than r. If the time series is characterized by an attractor, the correlation integral is found to obey a power law with $C(r) \approx r^d$, as $r \rightarrow 0$ and $N \rightarrow \infty$. The exponent d, defined as the correlation dimension of the attractor, can be obtained from the slope of log $C(r)$ versus log r. The correlation integral measured by the maximum norm for a Gaussian white noise process with standard deviation σ is

$$C_{Gauss}(r) = erf(\frac{r}{2\sigma})^m. \tag{6.2}$$

In fact, the correlation integral can be measured in 3 different ways. Apart from the maximum norm, the Euclidean norm or a Gaussian Kernel can also be used. Different considerations of the correlation integral result into different sensitivities on the slope increase.

Which one of the method is preferable for $N \rightarrow \infty$? None is preferable, according to Schreiber (Schreiber, 1993*a*), because all of them, independently from their sensitivity to noise, are designed to give the very same result, which is the noise standard deviation (σ).

6.3.2 Noise reduction

Schreiber (Schreiber, 1993*b*) proposed an extremely simple but robust noise reduction method, which has been found to be suitable for trajectories contaminated with high noise levels. The procedure of the method is as follows.

Suppose we have a scalar time series $X_i, i = 1, 2, , N$, where the X_i are composed of a clean signal Y_i with some noise η_i added so that $X_i = Y_i + \eta_i$. The main idea of the noise reduction method is to replace each measurement X_i by the average value of this coordinate over points in a suitably chosen neighborhood. The neighborhoods are defined in a $(k + 1 + l)$-dimensional phase space reconstructed by delay coordinates using information on k past coordinates and l future coordinates given by

$$Y_i = (X_{i-k}, X_{i-k+1}, ..., X_i, ..., X_{i+l-1}, X_{i+1}). \tag{6.3}$$

The present coordinate of X_i is then replaced by X_i^{corr} given by

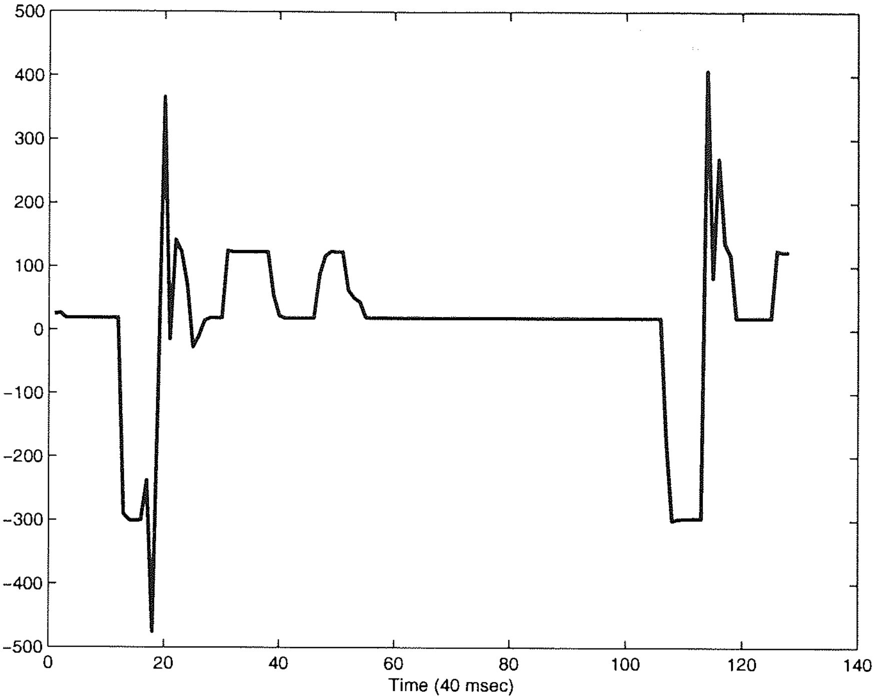

Figure 6.2: The motion current signature of figure 5.2 after noise filtering using Schreiber noise reduction technique.

$$X_i^{corr} = \frac{1}{|C_i^{\tau}|} \sum_{C_i^{\tau}} X_{\not{i}}. \tag{6.4}$$

According to this procedure, only the central coordinate in the delay window is corrected since only this coordinate is optimally controlled from past and future. The X_i^{corr} values can then be used to reconstruct the phase space and the procedure can be repeated. Further details on the selection of the parameters, e.g. the neighborhood size, embedding dimension, are out of scope of this thesis and can be obtained from Schreiber (Schreiber, 1993*b*).

6.4 Application of the Schreiber noise reduction on the motion current signature

Schreiber noise reduction technique was applied to the motion current signature collected from axis 3 of the production machine (figure 5.2) and the filtered signal is shown in the figure 6.2.

The values of the delay (τ), embedding dimension and radius were set to the minimum possible of 1, 2 and 0.1 respectively (Kantz and Schreiber, 2004).

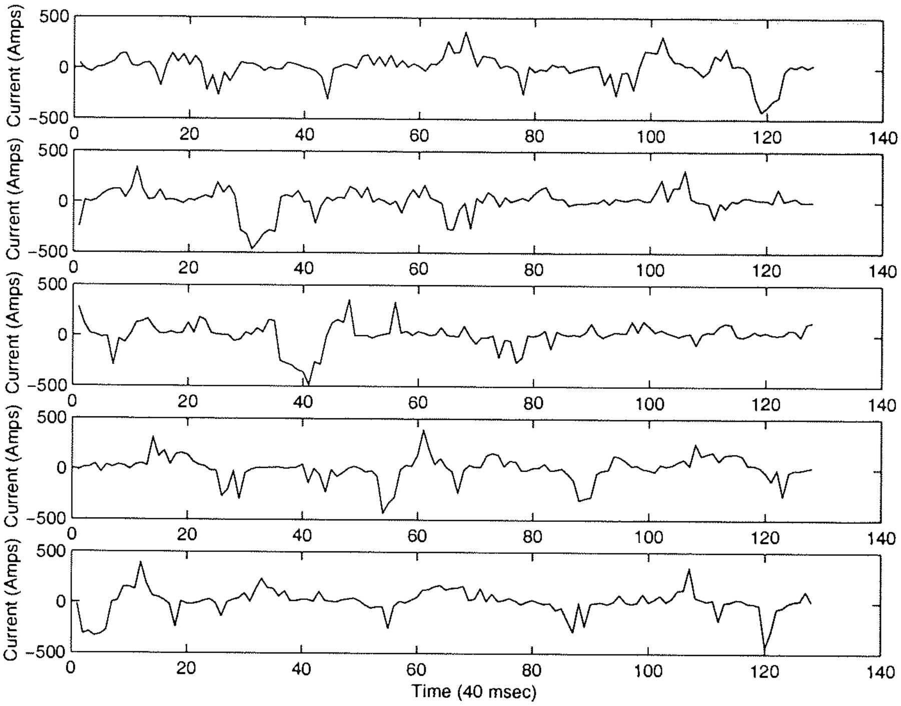

Figure 6.3: 5 of the 40 surrogate data sets generated for the experiment by directly optimizing the spectrum of a randomly generated time series such that it matches that of the original data in figure 6.2.

As already mentioned in section 6.2, the nonlinearity in the motion current signature is due to dynamical noise. Clearly, if the filtered signature (figure 6.2) is found to be linear, it can be concluded that the noise reduction algorithm has removed all or most of the nonlinear dynamical noise.

The capability of the Schreiber noise reduction algorithm to reduce nonlinear noise from the motion current signature was tested using the same procedure as section 5.5. The original signature used in this case was the filtered signature shown in figure 6.2. Figure 6.3 shows 5 of the 40 generated surrogate data sets and figure 6.4 presents the distribution of the test statistic for surrogate data sets. The vertical line denotes the test statistic for the original signature.

According to figure 6.4, the test statistic T_0 of the original filtered signature is drawn from the same distribution of the test statistic T_s of the surrogate data sets. Figure 6.4 highlights that the filtered signature failed to contradict the null hypothesis, establishing that it has reduced or no nonlinear noise component. Thus, the linear reverse algorithm can be used for estimating machine system parameters using filtered motion current signature.

The experiment was then repeated using the motion current signature from axis 6 of the production machine. The experiment is detailed as follows:

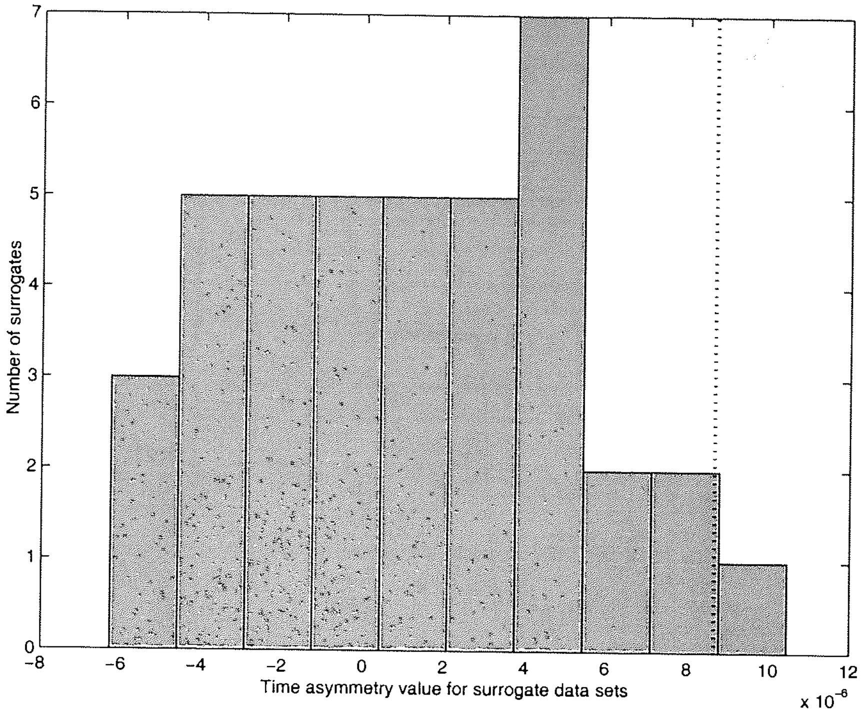

Figure 6.4: Distribution of the test statistic (ϕ) of the surrogate data sets against the test statistic (ϕ) of the original filtered signature of axis 3.

- **Figure 6.5:** The filtered signature generated using the Schreiber noise reduction technique on the original signature of figure 5.5.
- **Figure 6.6:** The surrogate data sets generated using the filtered signature of figure 6.5.
- **Figure 6.7:** The distribution of the test statistic of the surrogate data sets (figure 6.6) plotted along with the value of the test statistic of the original signature of figure 6.5.

The results of figures 6.4 and 6.7 re-confirm the belief that the motion current signature collected from the production machine has nonlinear dynamical noise which can be filtered to result into a mostly linear signal using Schreiber noise reduction technique.

The reverse algorithm, developed in section 4.9, was applied to the filtered signature to obtain the values of the system parameters. The next section shows the results of the application of the reverse algorithm (BJEST) after the reduction of noise using the Schreiber noise reduction technique on the motion current signature.

6.5 Results of the application of linear reverse algorithm

Data used for enhanced reverse algorithm testing was same as the one used in section 4.10, in which the reverse algorithm was applied on the data collected from the production machine for

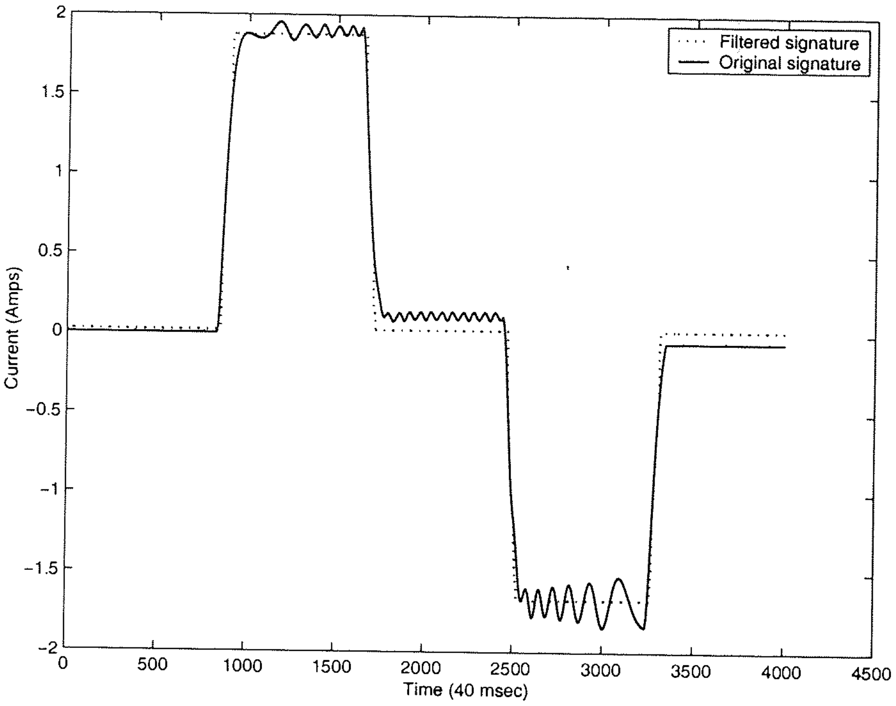

Figure 6.5: The motion current signature of figure 5.5 superimposed on the filtered signature. The noise filtering was performed using the Schreiber noise reduction technique.

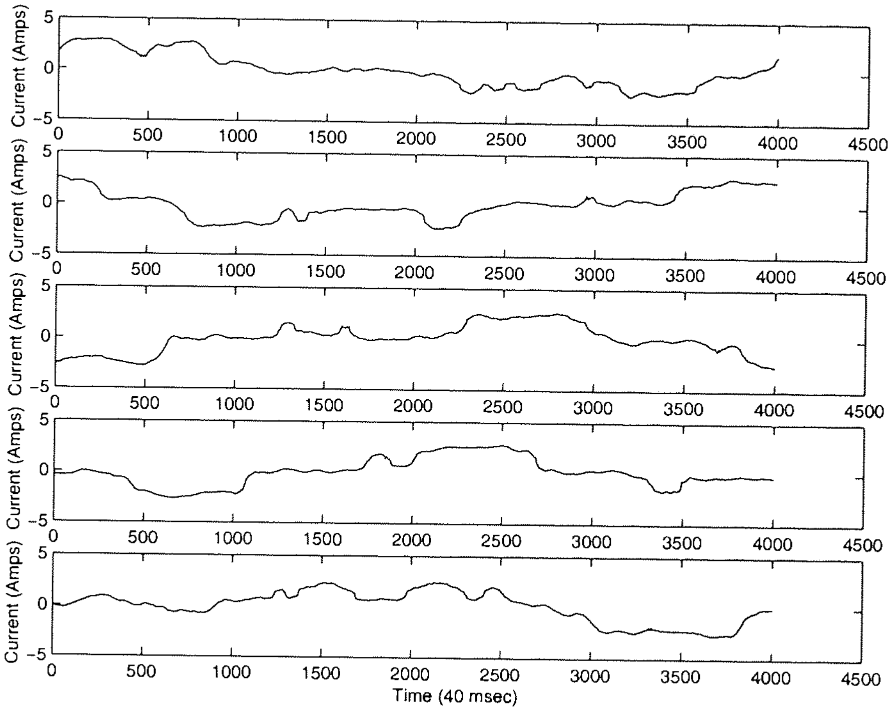

Figure 6.6: 5 of the 40 surrogate data sets generated for the experiment by directly optimizing the spectrum of a randomly generated time series such that it matches that of the original data in figure 6.5.

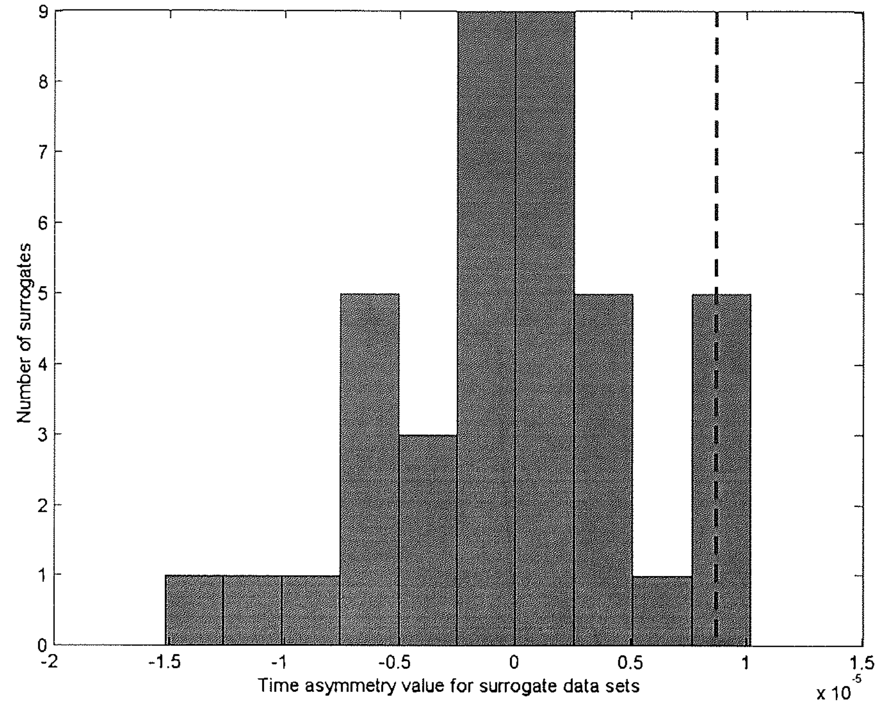

Figure 6.7: Distribution of the test statistic (ϕ) of the surrogate data sets against the test statistic (ϕ) of the original filtered signature of axis 6.

validation and verification of the simulation model. Table 4.5 shows the results obtained by using the reverse algorithm on non-filtered motion signature. It is apparent from table 4.5 there were significant fluctuations in the predicted values of the system parameters when the input to the reverse algorithm was a noisy nonlinear motion current signature. The inconsistency in the predicted values was due to the fact that a linear reverse algorithm was applied to the noisy nonlinear data. The presence of fluctuations in the predicted values can, to a large extent, be attributed to the nonlinearity of the input signature.

Table 6.1 highlights the results obtained using the reverse algorithm after the motion current signature is filtered using the Schreiber noise reduction technique. The results shown in table 6.1 were computed by filtering the noisy nonlinear motion current signature using Schreiber noise reduction technique and then reverse algorithm was applied to compute the values of the system parameters.

The values of the system parameters, indicated in tables 4.5 and 6.1, were then plotted to demonstrate the prediction fluctuations (figures 6.8, 6.9 and 6.10) for inertia, friction torque and gravitation torque.

Figures 6.8, 6.9 and 6.10 show that the fluctuations in the predicted values are very low when the reverse algorithm is applied after filtering the noise out of the data using the Schreiber

Table 6.1: Results obtained by applying reverse algorithm to the filtered signatures of axis 3 of the production machine. One of the signatures used for the experiment is shown in figure 4.

Data Set	Total Inertia (J) kgm^2	Load Inertia $J - 0.00068$ kgm^2	Friction Torque (F) Nm	Gravitational Torque (G) N-m
1	0.0123	0.01162	0.2986	0.0503
2	0.0124	0.01172	0.2948	0.0507
3	0.0123	0.01162	0.2999	0.0501
4	0.0123	0.01162	0.2908	0.0520
5	0.0124	0.01172	0.2967	0.0516
6	0.0123	0.01162	0.2943	0.0525

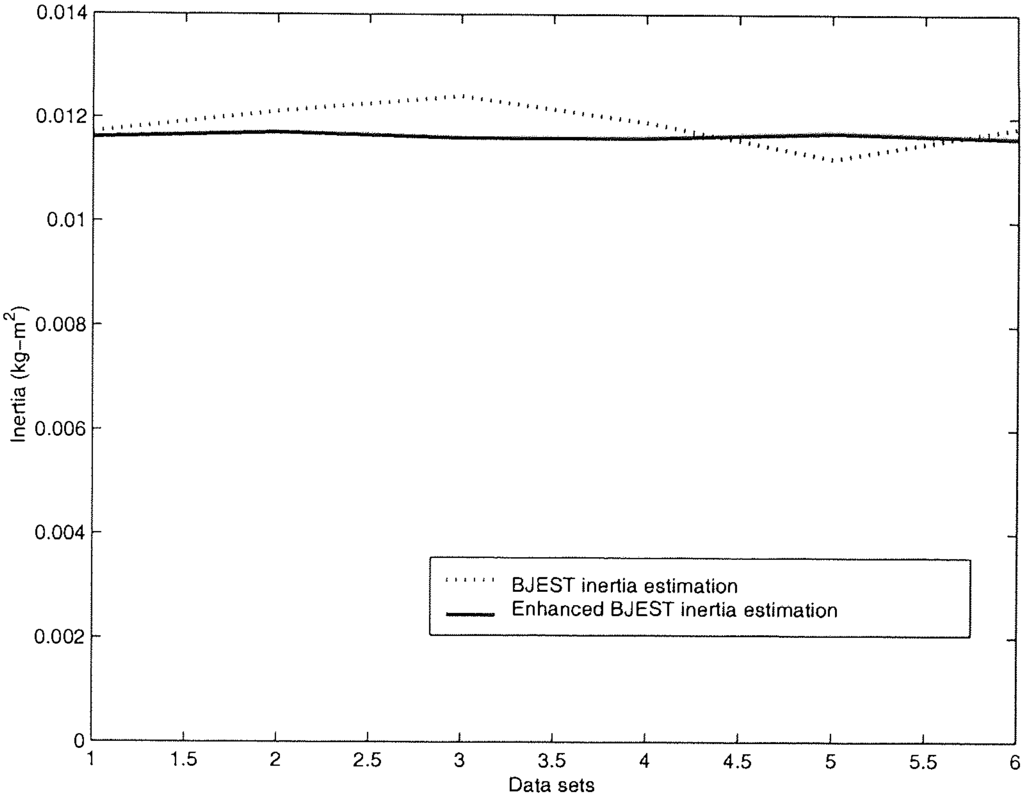

Figure 6.8: Inertia estimates using linear reverse algorithm compared to the estimates using enhanced reverse algorithm. The graph shows the variation in the inertia for twelve different data measurements, referred to as data sets.

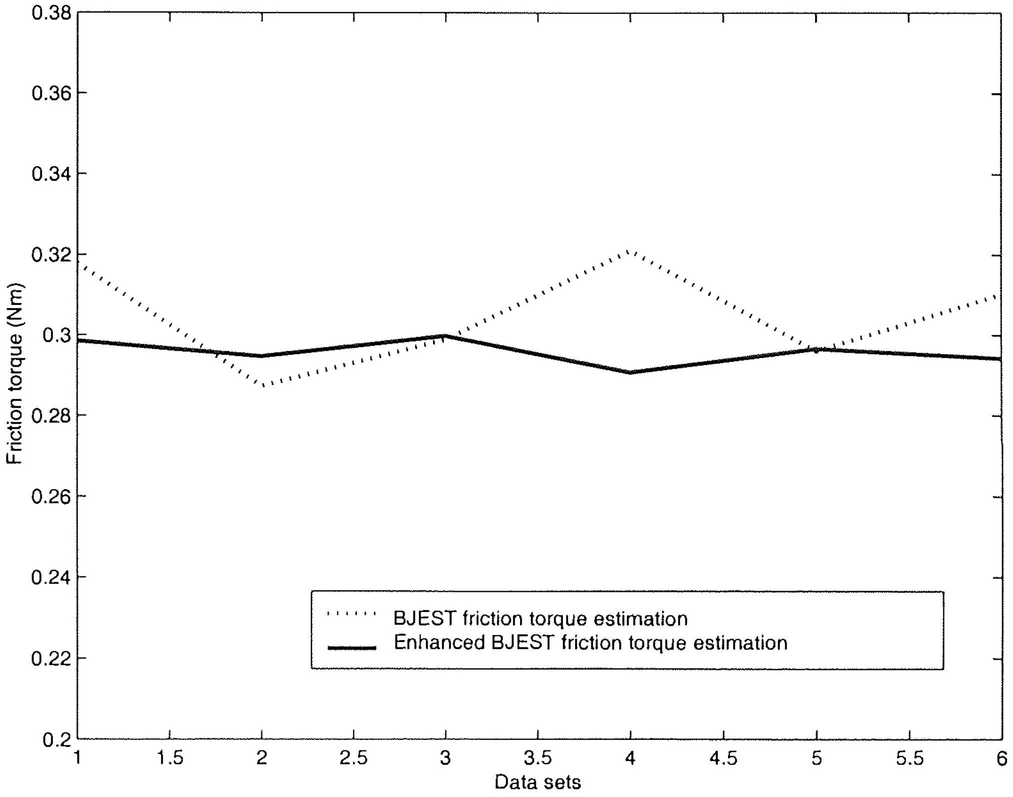

Figure 6.9: Friction estimates using linear reverse algorithm compared to the estimates using enhanced reverse algorithm. The graph shows the variation in the friction for twelve different data measurements, referred to as data sets.

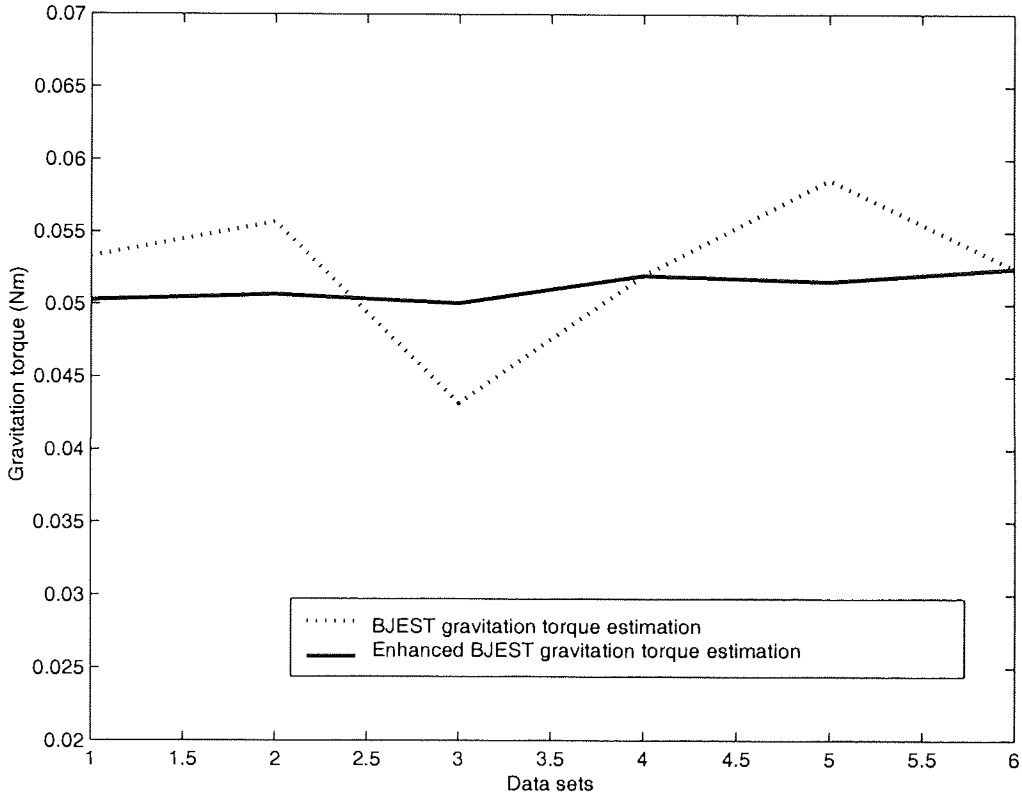

Figure 6.10: Gravity estimates using linear reverse algorithm compared to the estimates using enhanced reverse algorithm. The graph shows the variation in the gravity for twelve different data measurements, referred to as data sets.

noise reduction technique. Due to the reason that the machine system parameters have not changed during the operation cycle of the machine, a constant single value prediction over various machine operation cycle would prove that the estimation of the parameters is accurate. Lower variation in the predicted values of the parameters after the reverse algorithm is applied following noise reduction proves that the predictive capability of the real-time predictive maintenance system has improved by using this technique.

6.6 Summary

The focus of this chapter has been the nonlinear noise reduction of the motion current signature after the previous chapter confirmed the presence of the nonlinear components in the motion current signature. The noise reduction technique was explained in brief followed by application to the motion current signature collected from a real production machine.

It has been argued that if the filtered data, obtained after noise reduction of the motion current signature, is tested against the null hypothesis adopted in chapter 5, we should be able to determine the success of the noise reduction technique. The motion current signature were then filtered and tested against the hypothesis that the data is derived from a linear stochastic process. Surrogate data testing did not reject the hypothesis confirming that the data belongs to a linear process and there was little or no nonlinear component in the filtered data.

The machine system parameters were then estimated using the filtered data and it was found that the predictions obtained by such a method were more accurate and consistent than the ones obtained in chapter 5, where noisy motion current signature was used. This has proven that the new improved reverse algorithmic technique, Schreiber noise reduction followed by BJEST, is capable of defining the framework of the real-time predictive maintenance system. These techniques put together will also eliminate the requirement of any training data, thereby overcoming the problem of lack of micro-dynamical capabilities of the simulation model.

This chapter concludes the applications and results section of this thesis. The following and final chapter brings the thesis to a close by providing a summary of the key findings uncovered during the course of this project.

Chapter 7

Conclusions

7.1 Introduction

In the first and second chapter of this thesis, the concept of the real-time predictive maintenance system was promoted. The model was based upon the idea that the armature current of the DC servo motor depends upon the machine system parameters according to the relation:

$$i_A = \frac{(J_M + J_L)\frac{d\omega}{dt} + D\omega + T_f + T_L}{K_T}. \tag{7.1}$$

In the third chapter of this thesis the proof of concept of the real-time predictive maintenance system was performed. Chapter 4 was dedicated to the design, implementation, validation and verification of the simulation model to be used for generating the training data. Chapter 5 and 6 of this thesis were focussed on the use of the nonlinear time-series analysis techniques, including testing for nonlinearity and nonlinear noise determination and reduction.

7.2 General overview

The results of the machine parameter estimation were very successful. This thesis has described the use of a variety of novel, real-time, processing techniques to estimate machine system parameters using motion current signature. The key results of this thesis are summarised below:

- It has been demonstrated through the use of neural networks and nonlinear time series approaches that there is a predictive relationship between the motion current signature and the machine system parameters. The high accuracy (97.59%) of the proof of concept experiment showed the possibility of extracting machine system parameter information from the motion current signature.

- A simulation model, TuneLearn, was developed to cater to the training data requirement of neural networks. The simulation model was capable of mapping the motion current signature against the machine system parameters. The inputs of the simulation model were the motion profile, the motor-amplifier parameters, the tuning configuration, the load information and the machine system parameters. The output were the motion current signature and the motion velocity feedback. The validation and verification experiment was conducted to ascertain confidence in the simulated training data, which demonstrated the macro-dynamical simulating capabilities of the simulation model.

- A linear reverse model, BJEST, was designed to estimate the machine system parameters using the motion current signature, as an alternative to the use of neural networks approach.

- Surrogate data testing of the motion current signature, collected from a production machine, highlighted the nonlinear nature of the signatures. This confirmed the belief that the if the nonlinear noise of the motion current signature could be reduced or removed, the "*clean*" signal would be an ideal input for the linear reverse algorithm due to its linear nature.

- The nonlinear noise reduction technique developed by Schreiber (Schreiber, 1993*a*,*b*) was adopted to filter the motion current signature. It is shown that the filtered signature had little or no nonlinear component, supporting the belief that the filtered signature is a perfect input for the linear reverse algorithm.

- The reverse algorithm was applied to the filtered data to extract machine system parameter information and the results were very successful. We were able to find that the estimation obtained using the reverse algorithm following noise reduction were consistent and accurate.

7.3 System setup

The development of the real-time predictive maintenance has focussed on minimum setup requirements. The system setup only requires serial interfacing between the computer and the motor-drive system. All the maintenance calculations are mathematical and are conducted by the computer using basic mathematical toolkits. The output of the estimation calculations are displayed in a customised graph, depicting a real-time variation of the parameter. A basic outline of the setup procedure is shown in figure 7.1.

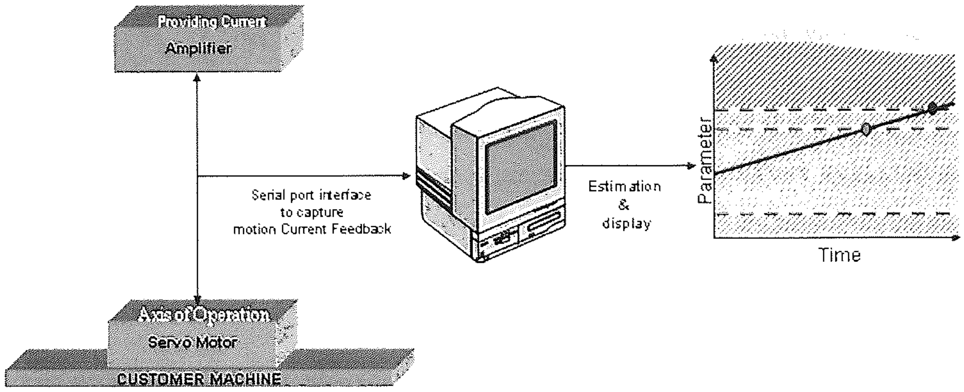

Figure 7.1: A basic outline of the system setup from a technician's point of view. All the estimation calculations are performed by the computer.

Figure 7.1 shows the serial port interface between the computer and the motor-drive system. The estimation and display results are shown to be the output of the calculations performed by the computer program.

7.4 Future work

As the machine predictive maintenance in real-time is such an important requirement, the potential for further research in this area is enormous. Similarly, there has been a corresponding explosion of interests in the general class of motion current signature analysis. In applying the motion current signature analysis to the predictive maintenance requirement of the machine systems using neural network and nonlinear time-series techniques, this work has been an attempt to bridge the gap between the two areas of research. While we feel that we have made some headway in this direction, there undoubtedly exists numerous avenues in which the research described in this thesis may be improved and extended. Possible extensions may be divided into two categories: additions and modifications to the technique itself, and the development of the technique for other systems.

7.4.1 Methodological extensions

There are a number of areas where improvements could be made to obtain improved understanding and results of the real-time predictive maintenance system. They are listed in order of their appearance in this thesis as follows:

- Even though the neural network approach has not been adopted as the final solution recommended by this thesis, the approach holds keys to enormous amount of possible extensions to the proposed framework. The disadvantage associated with the use of the neural network has been the vast requirement of the training data, which led this research to explore avenues of simulation modelling. However, it is believed that the use of a single neural network model for estimating each machine system parameters would reduce the training data requirement. Additionally, other promising neural network techniques, such as the bayesian approach, should be considered to benchmark the most useful neural network technique for the system.

- The definition of the micro-dynamical structure of the motion current signature could answer a lot of key questions associated with the motion current signature analysis. Such a knowledge would not only help in the correction of the simulation model, but also lead to the possibility of development of a nonlinear reverse algorithm. This nonlinear reverse algorithm could be used to map the motion current signature to the corresponding machine system parameters value, without the use of any neural network approach.

- The proposed technique has been shown to be effective in a controlled production environment. However, a rigorous laboratory and industrial testing for the technique for correctness is desirable prior to its deployment for practical purposes.

7.4.2 Practical applications

The thesis has primarily focussed on the real-time predictive analysis of the DC motor. However, it is easy to foresee that the proposed system of nonlinear analysis based estimation could be applied for predictive advantages of other equipment, such as transformers and generators.

The main assumption in the implementation of the real-time predictive maintenance system has been the fact that the torque producing current is directly proportional to the torque. The proportionality assumption has created a direct relationship between the machine system parameters and the current signature as shown in equation 7.1, which fits into the operational domain of brushless servo type of DC motors. However, the system can also be extended to the maintenance of other types of motors, in addition to the brushless servo type of DC motors, which do not strictly obey equation 7.1. For example, the current and torque relation for an induction motor is non-linear. However, using the nonlinear time series techniques, explained in chapter 6, some or all of these nonlinearities can be eliminated. The filtered motion current signature from the induction motor then acts as an input for the real-time predictive maintenance system.

Bibliography

Ajtonyi, I. and G. Terstnszky. 1994. "Fault diagnosis in industrial processes via parallel processing." *Proceedings of CONTROL, IEE* pp. 1117–1121. 2.4.1

Andrews, J.D. and S.J. Dunnett. 2000. "Event-tree analysis using binary decision diagrams." *IEEE Transactions on Reliability* 49:230–238. 2.4.1

Ball, P. and R. Isermann. 1998. "Fault detection and isolation for nonlinear processes based on local linear fuzzy models and parameter estimation." *Proceedings of American Control Conference* pp. 1605–1609. 2.4.2

Beck, M. B., J. R. Ravetz, L. A. Mulkey and T. O. Barnwell. 1987. "On the problem of model validation for predictive exposure assessments." *Stochastic Hydrology and Hydraulics* 11:229–254. 4.5.2

Betta, G., C. Ligouri and A. Pietrosanto. 1998. "An advanced neural-network based instrument fault detection and isolation scheme." *IEEE Transaction on Instrumentation and Measurement* 47(2):507–512. 1.5, 3.1

Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition.* Oxford Press. 3.1, 3.2.1, 3.2.1.1, 3.2.1.1, 3.2.1.2, 3.2.1.3, 3.2.1.3, 3.3, 3.3.1, 3.3.1, 3.3.2, 3.5, 3.5, 3.8, 3.9, 3.10, 5.1

Boothman, D.R., E.C. Elgar, R.H. Rehder and R.J. Woodall. 1974. "Thermal tracking - A rational approach to motor protection." *IEEE Transaction Power Apparatus and System* pp. 1335 – 1344. 1.5, 2.1, 2.6

Broomhead, D. and G. P. King. 1986. "Extracting qualitative dynamics from experimental data." *Physica D* 20:217–236. 6.2

Cambrias, S. and S.A. Rittenhouse. 1988. "Genetic guidelines for life extension of plant electrical equipment." *Electric Power Research Institute Rep. EL-5885* . 1.1, 1.5, 2.1, 2.2

Chafi, M.S., M. Akbarzadeh and M. Moavenian. 2001. "Fault detection and isolation in nonlinear dynamic systems: a fuzzy-neural approach." *Proceedings of IEEE International Fuzzy Systems Conference* pp. 1072–1075. 2.4.2

Chow, M.Y., R.N. Sharpe and J.C. Hung. 1993. "On the application and design of artificial neural networks for motor fault detection - Part I." *IEEE Transaction on Industrial Electronics* 40:181–188. 1.5, 3.1

Chow, T.W.S. 1996. "Condition Monitoring of electrical machines using third order spectrum analysis." *Conference Rec. 1996 IEEE-IAS Annual Meeting* 1:679–686. 1.5, 2.1, 2.6

Davey, A., R. Grosvenor, P. Morgan and P. Prickett. 1996. "Petri-net based machine tool failure and diagnosis." *Proceedings of COMADEM* pp. 723–731. 2.4.1

Davies, M. 1994. "Noise reduction schemes for chaotic time series." *Physica D* 79:174–192. 6.1

Dimla, D.E. 2000. "Sensor signals for tool-wear monitoring in metal cutting operations-a review of methods." *International Journal of Machine Tools and Manufacture* 40:1073–1098. 2.4.3

Discenzo, F.M. 1997. "Motor diagnostics: Technological drivers leading to 21st century predictive diagnostics." *Proc. 1997 of International conference of Maintenance and Reliability* 1:30.01–30.12. 1.1, 1.5, 2.1, 2.2

Efron, B. 1979*a*. "Bootstrap methods: Another look at the jackknife." *The Annals of Statistics* 7:1–26. 5.4

Efron, B. 1979*b*. "Computers and the theory of statitics: Thinking the unthinkable." *SIAM Review* 21:460–480. 5.4

Electro-craft. 1997. *Standard Product Catalog.* 2.2, 2.1, 2.3, 2.3, 2.3, 3.1, 2

Fink, Donald G. and H. Wayne Beaty. 1978. *Standard Handbook for Electrical Engineers, Eleventh Edition.* New York: McGraw-Hill. 2.2

Frankowiak, Marcos, Roger Grosvenor and Paul Prickett. 2004. "A review of the evolution of microcontroller-based machine and process monitoring." *International Journal of Machine Tools and Manufacture* pp. 1–18. 2.4, 2.4.1, 2.4.2, 2.7

Golyandina, N., V. Nekrutkin and A. Zhigljavsky. 2001. *Analysis of Time Series Structure: SSA and Related Techniques.* Chapman and Hall. 5.4

Govindan, R., K. Narayanan and M. Gopinathan. 1998. "The evidence of deterministic chaos in ECG: Surrogate and predictability analysis." *Chaos* 8:495–502. 5.4

Grassberger, P., R. Hegger, H. Kantz, C. Schaffrath and T. Schreiber. 1993. "On noise reduction methods for chaotic data." *Chaos* 3:127–141. 6.2

Haddad, S.D. 1991. "Condition monitoring and fault diagnosis for predictive maintenance and quality control." *Proceedings of 3rd International Machinery Monitoring and Diagnostic Conference* pp. 1–8. 1.1, 1.5, 2.2

Herbert, R.G. 1984. "Computer techniques applied to the routine analysis of rundown vibration data for condition monitoring of turbine-alternators." *Proceedings of the International conference on Condition Monitoring* 47:229–242. 1.1, 1.5, 2.2

Horn, J. L. 1965. "Fluid and crystallized intelligence." Unpublished doctoral dissertation, University of Illinois, Urbana. 5.4

Hornik, K. 1991. "Approximation capabilities of multilayer feedforward networks." *Neural networks* 4:251–257. 3.2.1.1

Hornik, K., M. Stinchecombe and H. White. 1989. "Multilayer feedforward networks are universal approximators." *Neural networks* 2:359–366. 3.2.1.1

Hu, W., A.G. Starr and A.Y.T. Leung. 1999. "Integrated hierarchical diagnostic reasoning for FMS's." *Proceedings of First International Conference on the Integration of Dynamics, Monitoring and Control* . 2.4.1

Hu, W., A.G. Starr and A.Y.T. Leung. 2003. "Operational fault diagnosis of manufacturing systems." *Journal of Material Processing Technology* 133:108–117. 2.4.1

Hu, W., A.G. Starr, Z. Zhou and A.Y.T. Leung. 2000. "A systematic approach to integrated diagnosis of flexible manufacturing systems." *International Journal of Machine Tools and Manufacture* 40:1587–1602. 2.4.1

Inc., Reliance Motion Control. 1996. *Host Command Reference of UltraMaster version 1.60.* 4.8

Isermann, R. 1984. "Process fault detection based on modelling and estimation methods- a survey." *Automatica* 20:387–404. 1.1, 2.6, 4.1

Jantunen, E. 2002. "A summary of methods applied to condition monitoring in drilling." *International Journal of Machine Tools and Manufacture* 42:997–1010. 2.4.3

Jantunen, E. and H. Jokinen. 1996. "Reduction of data needed in an expert system for condition monitoring of FMS using regression analysis techniques." *Proceedings of COMADEM* pp. 675–684. 2.4.2

Jms-Jounela, S.L., M. Vermasvuori, P. Endn and S. Haavisto. 2003. "A process monitoring system based on the Kohonen self-organizing maps." *Control Engineering Practices* 11:83–92. 2.4.2

Johnson, M. T. and R. J. Povinelli. 2005. "Generalized phase space projection for nonlinear noise reduction." *Physica D* 201:306–317. 6.2

Johnson, P. 2003. "Basic dynamic signal acquisition and analysis for machine condition monitoring." *Proceedings of ICOM* pp. 421–426. 2.4.2, 2.4.3

Kantz, H. and T. Schreiber. 2004. *Nonlinear time series analysis.* Cambridge University Press. 5.2, 5.4, 5.4.2, 5.4.2, 5.4.3, 6.2, 6.4

Kantz, H., T. Schreiber, I. Hoffmann, T. Buzug, G. Pfister, L. G. Flepp, J. Simonet, R. Badii and E. Brun. 1993. "Nonlinear noise reduction: A case study on experimental data." *Physical Review E* 48:1529–1538. 6.2

Kavcic, M. and D. Juricic. 2000. "CAD for FaultTree-based diagnosis of industrial processes." *Engineering Applications of Artificial Intelligence* 14:203–216. 2.4.1

Keyhani, A. and S.M. Miri. 1986. "Observers for tracking of synchronous machine parameters and detection of incipient faults." *IEEE Transaction Energy Conservation* 1. 1.5, 2.1, 2.6

Kim, D.S., Y.S. Shin and D.K. Carlson. 1991. "Machinery diagnostics for rotating machinery using back-propagation neural network." *Proceeding of 3rd International Machinery Monitoring and Diagnostic Conference* pp. 309–320. 1.5, 4.1

Kleijnen, J. P. C. 1995*a*. "Case Study: statistical validation of simulation models." *European Journal of Operational Research* 87:21–34. 3, 4.5.2

Kleijnen, J. P. C. 1995*b*. "Verification and validation of simulation models." *European Journal of Operational Research* 82:145–156. 3, 4.5.2

Kleijnen, J. P. C. 1999. "Validation Models: Statistical techniques and data availability." *Proceedings of the Winter Simulation Conference* pp. 647–654. 1.3.2, 4.5, 4.5.1, 4.5.2, 4.5.2

Klema, V. C. 1980. "The singular value decompositon: Its computation and some applications." *IEEE Transaction on Automatic Control* 25:164–176. 3.3.2

Kostelich, E. J. and J. A. Yorke. 1988. "Noise reduction in dynamical systems." *Physical Review A* 38:1649–1652. 6.1

Kostelich, E. J. and J. A. Yorke. 1990. "Noise reduction: finding the simplest dynamical system consistent with the data." *Physica D* 41:183–196. 6.1

Kostelich, E. J. and T. Schreiber. 1993. "Noise reduction in chaotic time-series data: A survey of common methods." *Physical Review E* 48:1752–1763. 6.1

Kryter, R.C. 1989. "Condition monitoring of machinery using motor current signature analysis." *Sound & Vibration* pp. 14–21. 1.5, 2.1, 2.6

Kugiumtzis, D. 1999. "Test your surrogate data before you test for nonlinearity." *Physical Review E* 60:2808–2816. 5.4.1

Kuzawinski, K.M. and R. Smurthwaite. 1988. "Automated fault tree analysis via AI/ES." *Proceedings of Annual Reliability and Maintainability Symposium, IEEE* pp. 331–335. 2.4.1

Leith, D. 1988. "Condition Monitoring of electrical machines by real-time expert system." *Proceedings of International Conference of Electrical* 3:297–302. 1.1, 1.5, 2.6

Lennox, B., G.A. Montague, A.M. Frith, C. Gent and V. Bevan. 2001. "Industrial application of neural network - an investigation." *Journal of Process Control* 11(5):497–507. 1.5, 2.4.2, 3.1

Lin, Chang-Ching and Hsu-Pin Wang. 1996. "Performance analysis of rotating machinery using enhanced cerebellar model articulation controller (E-CMAC) neural networks." *Computers and Industrial Engineering* 30(2):227–242. 1.5, 3.1

Lipmann, R.P. 1989. "Pattern classification using neural networks." *IEEE Communication magazine* pp. 47–64. 1.5, 3.8, 4.1

Littlehales, P.A. and B. Jones. 1997. "Software Scheme for Machine Design." *Proceedings of Drives and Controls Conference* . 1.5, 4.8

Littlehales, P.A., B. Jones, L.A. Smelov and N.J. Leighton. 1998. "An Interactive Approach to the Selection of Servo Drive components." *Proceedings of Drives and Controls Conference* . 1.5, 4.8, 4.9

Mageed, M.F.A., A.F. Sakr and Bahgat. 1993. "Fault detection and identification using hierarchical neural network-based system." *IEEE* pp. 338–343. 2.4.2

Mardia, K. V., J. T. Kent and J. M. Bibby. 1979*a*. *Multivariate Analysis.* Acamdemic Press. 4.9, 4.9.1, 4.9.1

Mardia, K.V., J.T. Kent and J.M. Bibby. 1979*b*. *Multivariate Analysis.* Acamdemic Press. 3.8

Meziane, F., S. Vadera, K. Kobbacy and N. Proudlove. 2000. "Intelligent systems in manufacturing: current developments and future prospects." *Integrated Manufacturing Systems* 11:218–238. 2.4.2

Moseler, O. and R. Isermann. 2000. "Application of model based fault detection to a brushless DC motor." *IEEE Transaction on Industrial Electronics* 47:1015–1020. 1.5

Nabney, Ian T. 2002. *NETLAB : Algorithm for Pattern Recognition.* Springer. 3.1, 3.2.1.1, 3.2.1.2, 3.2.1.2, 3.2.1.3, 3.3, 3.4, 3.6, 3.8, 3.9

Osborne, A. R., A. D. Kirwin, A. Provenzale and L. Bergamansco. 1986. "A search for chaotic behavior in large and mesoscale motions in the pacific ocean." *Physica D.* 23:75–90. 5.4.1

Ott, E., T. Sauer and J. A. Yorke. 1994. *Coping with Chaos.* New York: Wiley. 6.2

Papoulis, A. 1991. *Probability, random variables and stochastic processes.* Electrical and Electronic Engineering Series New York: Chaos. 5.4

Penmann, J. 1986. "Condition monitoring of electrical drives." *Proceedings of Institute of Electrical Engineers* 133:142–148. 1.1, 1.5, 2.1, 2.2, 4.9

Pikovsky, A. 1986. "Noise filtering in the discrete time dynamical systems." *Soviet Journal of Communications and Technological Electronics* 31:911–914. 6.2

Prickett, P. 1997. "A Petri-net based machine tool maintenance management system." *Industrial Management and Data Systems* 97:143–149. 2.4.1

Prickett, P. and C. Johns. 2001. "The development of an on-line milling cutter tool breakage detection system." *Proceedings of I.Mech.E., Journal of Engineering Manfufacture* 215:147–160. 2.4.3

Prickett, P.W. and R.I. Grosvenor. 1999. "Non-sensor based machine tool and cutting process condition monitoring." *International Journal of COMADEM* 2:31–37. 2.4.3

Raaphorst, A.G.T., B.D. Netten and R.A. Vingerhoeds. 1995. "Automated fault-tree generation for operational fault diagnosis." *Proceeding of Electric Railways in a United Europe, IEE* pp. 173–177. 2.4.1

Rao, R.B.K.N. 2003. "Advances in acoustic emission technology in COMADEM: a state-of-the-art review." *Proceedings of COMADEM* pp. 1–18. 2.4.3

Rockwell. 2000. *Motion control glossary.* 2.3, 3.1, 2

Sargent, R. G. 1996. "Some subjective validation methods using graphical displays of data." *Proceedings of the Winter Simulation Conference* pp. 345–351. 4.5.2

Sargent, R. G. 2004. "Validation and verification of simulation models." *Proceedings of the Winter Simulation Conference* pp. 17–28. 1.3.2, 4.5, 4.5.1, 2, 3, 4.5.2

Schlesinger. 1979. "Terminology for model credibility." *Simulation* pp. 103–104. 4.5

Schreiber, T. 1993*a*. "Determination of the noise level of chaotic time series." *Physical Review E* 48:R13R16. 6.3, 6.3.1, 6.3.1, 7.2

Schreiber, T. 1993*b*. "Extremely simple nonlinear noise reduction method." *Physical Review E* 47:2401–2404. 6.2, 6.3, 6.3.2, 6.3.2, 7.2

Schreiber, T. and A. Schmitz. 1996. "Improved surrogate data for nonlinearity tests." *Physical Review Letters* 77:635–638. 5.4.1, 5.4.1, 5.4.2

Sick, B. 2002. "On-line and indirect tool wear monitoring in turning with artificial neural networks: a review of more than a decade of research." *Mechanical Systems and Signal Processing* 16:487–546. 2.4.3

Singh, G. and H. Parikh. 1993. "Stepper motor data management: A structured approach." *Proceedings of IMCSD symposium* . 1.5, 4.8

Smeaton, R.W. 1987. *Motor Application and Maintenance Handbook.* New York: McGraw Hill. 1.1, 1.5, 2.1, 2.2

Sood, A.K., A.A. Fahs and N.A. Henein. 1985. "Engine fault analysis, Part I: Statistical methods." *IEEE Transaction of Industrial Electronics* 32:294–300. 1.5, 2.1, 2.6

Stinchecombe, M. and H. White. 1989. "Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions." *Proceedings of the International Joint Conference on Neural Networks* 1:613–618. 3.2.1.1

Tavner, P.J. and J. Penman. 1992. *Condition Monitoring and Electrical Machines.* New York: Research Studies Press Ltd. Wiley. 1.1, 1.5, 2.1, 2.2

Theiler, J., S. Eubank, A. Longtin, B. Galdrikian and J. D. Farmer. 1992. "Testing for nonlinearity in time series: the method of surrogate data." *Physica D* 58:77–94. 5.4, 5.4.1, 5.4.3

Thevenin, L. 1883*a*. "Extension de la loi d'Ohm aux circuits electromoteurs complexes (Extension of Ohm's law to complex electromotive circuits)." *Annales Telegraphiques* 10:222–224. 2.3

Thevenin, L. 1883*b*. "Sur un nouveau theoreme d'electricitep dynamique (On a new theorem of dynamic electricity)." *C. R. des Seances de l'Academie des Sciences* 97:159–161. 2.3

Unsworth, C.P., M.R. Cowper, S. McLaughlin and B. Mulgrew. 2001. "A new method to detect nonlinearity in a time-series: synthesizing surrogate data using a Kolmogorov-Smirnoff tested, hidden Markov model." *Physica D* 155:51–68. 5.4

Yang, S.K. and T.S. Liu. 1998. "A Petri-net approach to early failure detection and isolation for preventive maintenance." *Quality and Reliability Engineering International* 14:319–330. 2.4.1

Appendix A

Journals

This appendix presents all the journals published as part of this research for reference purposes. The journals have not been referenced anywhere in the main text to avoid any linking confusions.

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

International Journal of Machine Tools & Manufacture 44 (2004) 759–766

INTERNATIONAL JOURNAL OF
MACHINE TOOLS
& MANUFACTURE
DESIGN, RESEARCH AND APPLICATION

www.elsevier.com/locate/ijmactool

A real-time predictive maintenance system for machine systems

Dheeraj Bansal[a,*], David J. Evans[b], Barrie Jones[a]

[a] *Mechanical Engineering Department, School of Engineering and Applied Science, Control and Dynamics Research Group, Aston University, Birmingham B4 7ET, UK*

[b] *Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK*

Received 27 October 2003; received in revised form 21 January 2004; accepted 2 February 2004

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

International Journal of Machine Tools & Manufacture 45 (2005) 1210–1221

INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE DESIGN, RESEARCH AND APPLICATION

www.elsevier.com/locate/ijmactool

Application of a real-time predictive maintenance system to a production machine system

Dheeraj Bansal[a,*], David J. Evans[b], Barrie Jones[a]

[a]Control and Dynamics Research Group, Mechanical Engineering Department, School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, UK
[b]Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK

Received 10 November 2004; accepted 30 November 2004
Available online 5 February 2005

ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT

International Journal of Machine Tools & Manufacture xx (2005) 1–11

INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE DESIGN, RESEARCH AND APPLICATION

www.elsevier.com/locate/ijmactool

BJEST: A reverse algorithm for the real-time predictive maintenance system

Dheeraj Bansal[a,*], David J. Evans[b], Barrie Jones[a]

[a]*Mechanical Engineering Department, Control and Dynamics Research Group, Aston University, Birmingham B4-7ET, UK*
[b]*Neural Computing Research Group, Aston University, Birmingham B4-7ET, UK*

Received 9 August 2005; accepted 17 August 2005