Vibration Control of Rotor-Bearing Systems using Neural Networks

by

Mohsin Siddiqui

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

MECHANICAL ENGINEERING

June, 1996

VIBRATION CONTROL OF ROTOR-BEARING SYSTEMS USING NEURAL NETWORKS

BY

MOHSIN SIDDIQUI

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

MECHANICAL ENGINEERING

JUNE 1996

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis, written by

Mohsin Siddiqui

under the direction of his Thesis Advisor, and approved by his Thesis committee, has been presented to and accepted by the Dean, College of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

Thesis Committee :

Dr. Yaagoub N. Al Nassar (Chairman)

Dr. Ahmed Z. Al Garni (Co-Chairman)

Dr. Bekir S. Yilbas (Member)

Dr. Y. A. R. Khulief (Member)

Dr. Muhammed O. Budair
Department Chairman

Dr. Al – Shehri, Abdallah M.
Dean, College of Graduate Studies

Date: 30-12-96

Dedicated to

My

Mother,

and

Father

whose patience and perseverance

lead to this accomplishment.

Acknowledgments

In the name of Allah, Most Gracious, Most Merciful

"Read in the name of thy Lord and Cherisher, who created. Created man from a [leech-like] clot. Read and thy Lord is Most Bountiful. He Who taught [the use] of the pen. Taught man that which he knew not. Nay, but man doth transgress all bounds. In that he looketh upon himself as self-sufficient. Verily, to thy Lord is the return [of all]" (The Holy Quran, Surah 96)

All praise and glory be to Almighty Allah who gave me courage and patience to carry out this work, and peace and blessings of Allah be upon Prophet Muhammad. Acknowledge is due to King Fahd University of Petroleum and Minerals for providing support for this research.

My deep appreciation goes to my major thesis advisor Dr.Yaagoub N. Al Naṣsar, for his constant help, guidance and the countless hours of attention he devoted

throughout the course of this work. He was always kind, understanding and sympathetic to me.

Thanks are due to my Co-advisor Dr. Ahmed Z. Al Garni, for his constant encouragement, moral support and suggestions. Thanks are also due to my thesis committee members Dr.Bekir S. Yilbas and Dr.Yahya A. Khulief for their interest, cooperation, advice and constructive criticism.

Special thanks are due to my colleagues and friends for their help and encouragement. They have made my stay at KFUPM a very pleasant and unforgettable experience.

I am also indebted to the department Chairman Dr. M. O. Budair and other faculty members for their support.

Finally, my heartfelt thanks and gratefulness to my parents and other family members for their encouragement and moral support.

Contents

List of Figures

List of Tables

THESIS ABSTRACT

Name: MOHSIN SIDDIQUI

Title: VIBRATION CONTROL OF ROTOR BEARING SYSTEMS USING NEURAL NETWORKS

Degree: MASTER OF SCIENCE

Major Field: MECHANICAL ENGINEERING

Date of Degree: JUNE 1996

A neural network controller is described and implemented for controlling the vibrations of a Rotor Bearing System. A multi-layered neural network is used to model the inverse dynamics of the rotor-bearing system on-line; it is learned by backpropagation algorithm, and delta rule in which the difference between the actual control input to the plant, which is generated from the neural controller, and the input estimated from the inverse-dynamics model by using an actual plant output is minimized. The results show a satisfactorily diminished response of the rotor-bearing system when the controller is applied to the system.

Keywords: Vibrations, Rotating Machinery, Neural Network, Control, Inverse Dynamics.

Master of Science Degree
King Fahd University of Petroleum and Minerals, Dhahran.
June 1996

خلاصة الرسالة

الاسم : محسن صديقي

العنوان : التحكم في اهتزازات نظام كرسي تحميل دوراني باستخدام نظام الشبكات العصبية

الدرجة : ماجستير

التخصص : هندسة ميكانيكية

التاريخ : يونيه ١٩٩٦

يتم شرح واستخدام نظام الشبكات العصبية للتحكم في اهتزازات نظام كرسي تحميل دوراني • يتم استخدام نظام الشبكات العصبية متعددة الأغراض في عمل نموذج عكسي لديناميكية نظام تحميل دوراني متوازي ،حيث يتم تلقين النظام عن طريق مجموعة اهتزازات عكسية في التعليمات وعن طريق (دلتا) يتم تصغير الفرق بين معلومات التحكم الحقيقية المدخلة الى النظام التي يتم الحصول عليها من نظام الشبكات العصبية التحكمي والمعلومات المتوقعة من النموذج الديناميكي العكسي عن طريق استخدام بيانات حقيقية من النظام الدوراني • عندما يتم استخدام متحكم عصبي على نظام الدوراني فان النتيجة بالنسبة للاهتزازات الناتجة مقنعة •

درجة الماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

الظهران – المملكة العربية السعودية

يونيو ١٩٩٦م

Chapter 1

Introduction

1.1 Motivation

A rotating machinery is one of the most important classes of machineries. It is used extensively throughout the industrialized world. Its uses are extremely diverse; in power stations, marine propulsion systems, aircraft engines, machine tools, automobiles, medical equipment, household accessories and many other applications. A rotating components is often used in situations where its correct functioning is of crucial importance. Failure of machine components in applications such as aero-engines, turbogenerators, military equipment, space satellites and others, may put human life in jeopardy and cost huge sums to repair. Researchers began to take interest in the vibration study of rotors, as the demand for higher rotor-speeds increased. Vibration in rotating machinery causes high levels of noise and wear together with

increased rotor-bending stresses which lead to a low machine fatigue life. The most common source of vibration is imbalance; all rotating machines have some residual imbalance because balancing procedures may be imperfect, hence some vibration is always present during the operation. Other sources of vibration are instability of rotor which may be attributed to fluid film bearings, and the stiffness and damping coefficients of the rotating shaft.

There are many methods developed in order to control the vibrations. The introduction of passive vibration control devices such as Squeeze Film Damper (SFD) [1, 2] help to reduce rotor vibrations and instability, but these devices suffer normal limitations associated with passive devices. A Squeeze Film Damper (SFD) is used to provide hydrodynamic damping to the rolling element bearings, which otherwise would contribute no appreciable amount of damping to the rotor bearing system. Kinematically, the SFD is a journal bearing in which the journal can vibrate or whirl but not spin. The outer race of the rolling element bearing, or an attached sleeve, serves as the journal of the SFD. It is generally slotted or restrained loosely with a pin to prevent spin. Since the parameters of an SFD cannot be changed, one which is applicable to a given set of operating conditions may not be universally applicable, especially when several modes of the vibrations are excited.

Such limitations of SFD led to an evolving interest in active control of rotor vibrations. Several techniques have been adopted by researchers for active control of vibration [3, 4, 5]. Modern control theory may be employed to develop control

systems to reduce or eliminate the vibration of rotor bearing system [6]. Different controllers that can be developed using modern control theory including

- State Variable Feed Back (SVFB)
- Integral plus State Variable Feed Back (ISVFB)
- Linear Quadratic Servo (LQS)

With life getting more complex and more demands being placed on control engineers, the principles of conventional control prove to be insufficient to control many real life situations that require less stringent assumptions.

The difficulties that arise in complex systems can be broadly classified into three categories [7]. The first is computational complexity, the second is the presence of nonlinear relationship with many degrees of freedom and the third is the uncertainty which includes the model uncertainty, parametric uncertainty as well as the presence of disturbances and noise. These difficulties are also responsible for the continued evolution in the field of control. For example, some years back, implementation of self-tuning controller would have been considered computationally infeasible, but now the availability of microprocessor technology has provided an impetus for research into self-tuning control. Also, today the theory of nonlinear systems isn't developed as the linear theory is. Closed form solutions for many nonlinear problems are not present.

Building systems with some autonomy means that these systems should be capable of setting their own goals, depending on the constraints and objectives imposed by the higher levels of the hierarchy, present situation and also the background knowledge or experience gained. Such systems may often be required to reason under uncertainty, thus involving a great deal of judgmental imprecision. One approach here would be to mimic human decision making. However, this is easier said than done. The lack of understanding of this " human-like " intelligence means that current control techniques do not have the ability to learn or to take decisions under unstructured environments. Hence, the need arises for intelligent controllers. These controllers should be capable of decision making under uncertainty. They also possess a highly decentralized behavior to facilitate fast and simple processing.

These requirements posed by the increasingly challenging problems of modern society caused the control system community to look around for better " pastures ". A natural step in this evolution was the use of neural networks in control. Neural networks offer much hope as a tool for on-line learning, optimization and optimal policy making. Their remarkable learning capability is what sets them apart from other modelling techniques.

The desire to mimic the superior human recognition capabilities, which are still beyond human understanding, was the main driving force that led researchers to ultimately develop a new paradigm for learning called Neural networks. Hence neural networks can be described as a human attempt to create an artificial brain-

shades of science fiction. However, it is to be pointed out that although neural networks keep on being improved in their performance, they are still very far from being a match to the human nervous system.

Neural networks also have the capability of massive parallel processing in contrast to the sequential execution of the conventional Von Neumann digital computers. Neural Networks also provide, insignificant fault tolerance, since damage to a few links does not impair the overall system performance.

Currently the hardware implementation of neural networks is also a very active research area. Some manufactures, such as Intel, have produced a neural net chip which is supposed to have more effective output (for the neural net calculation) than even the Cray supercomputer.

An area of control systems which offers much scope for the use of neural networks is adaptive control. Here the controller has to adapt itself with change in the process parameters. A class of adaptive control algorithm is called self-tuning. A self-tuning algorithm is one in which the control signal generated from the estimated parameters turns out to be asymptotically the same control law generated by feedback if the process parameters were known.

In this work we have concentrated upon method of utilizing neural networks as self-tuning controllers. A strategy for the neural control of the linear as well as non-linear systems is presented. Simulations on rotor-bearing systems have been performed in order to investigate the feasibility and performance of the proposed

method.

1.2 Scope of the work

A basis of objective is established in the light of the relevant literature survey in the chapter 2. It gives a detailed survey of different fields related to this topic. A brief definition of different types of vibration controlling schemes is give. First an indepth survey on the work related to vibration problems and its control is given, different strategies and methods used are discussed in brief. Survey of neural network applied to control problem is also presented. several papers are cited , related to neural network as adaptive controllers. Finally a brief short comings of conventional controllers is given. A basis for the application of neural network is established.

Chapter 3 deals with the mathematical modelling of the proposed rotor bearing system. A two degree of freedom system is chosen to demonstrate the capabilities of neural network as adaptive self tuning controller. The equations of motions is derived using the laws of motion. State variable representation of these equation is presented. At the end of the chapter an eigen values of the system are calculated.

In chapter 4 the fundamental of neural network are described. Architecture and training of multilayered neural network utilizing the backpropagation algorithms are explained. Basic principles of the self-tuning controllers are also explained at the end of this chapter.

Chapter 5 explains the application of neural network as a self tuning controller. A brief introduction of Neuro-Identification and Neuro-Control is given. Also given at the end of the chapter are the explanation of neural network as implicit and explicit self tuning controllers.

Chapter 6 gives the working of the program, it explains the procedure of selecting the size of the weight matrix , the number of inputs and outputs to the neural network, and the selection of k and c.

Finally chapter 7 gives the simulation and results. With the conclusion and results presented in chapter 8.

Chapter 2

Literature Survey

2.1 Vibration Control

The literature relevant to the causes and the remedy of the vibration of rotor-bearing system is presented in this section. A rotor running on bearings has a threshold speed; above which it becomes unstable, this instability is characterized as subsynchronized whirling. Cross-coupling stiffness (K_{xy}, K_{yx}) destabilizes the system [8]. Newkirk and Taylor (1925)first reported that the oil whirl occurred only in lightly loaded oil-film bearings operated at a very small eccentricity ratio [9]. Reddi and Trumpler [10] investigated the phenomenon of oil whirl in full (306°) and partial (180°) journal bearings. Their study reveals that, to achieve successful design one must consider a system consisting of bearings and rotors. Many researchers worked on balancing techniques for rigid and flexible rotor-bearing systems [11].

Modern machines are highly complex and therefore, in addition to unbalance in rotor, there are may reasons for vibrations. The rotors used in modern machines are highly flexible. These are generally mounted on more than two bearings and rotate at very high speeds. The instability in these machines is generally produced by aerodynamic forces on impeller wheels, friction in the stressed rotor and hydrodynamic forces in their bearings. Studies that have been conducted to control the vibration of rotor-bearing system can be broadly classified into two groups: passive vibration control; and active vibration control. Squeeze-film dampers (SFD) have been used in the past and are in use today as successful devices for passive vibration control. These dampers were first applied in 1889 by C.A. Parson to its turbo bearings. Many investigators worked on the feasibility of controlling synchronous and nonsynchronous vibration of rotor-bearing systems using SFD [2],[3]. The SFD improved the stability of rotor-bearing system. However, an SFD has, limited use if several modes of vibration are excited at the same time.

Realizing the limitation of a passive SFD, studies on an active SFD were carried out [4]. An active SFD was found to be superior to a passive SFD. The performance of electromagnetic dampers as active vibration controllers has been investigated. It is found that such a damper offers improved control of the vibrational behavior of the system [5], [12]. In order to achieve improved control strategies an investigation has been conducted to determine the feasibility of applying control inputs through non-rotating components of machines [6].

Neural networks for control systems is a rapidly expanding field. The tremendous interest in the field of artificial neural networks can be inferred from the growth of papers, journals, conferences and conference sessions devoted to it. Neural networks can be used in many different areas with control systems being one of them. With the interest aroused in applications of neural networks in control systems, there have been quite a few papers [7], [13], [14], [15], [16], [17] on this topic in various journals. A survey of these indicate that some of the major topics for further research include

- Utilization of neural networks for identification and control of nonlinear systems.
- Improving upon the direct neural network based self-tuning methods for linear as well as non-linear plants
- Improving the learning and adaptation rate of neural networks.

The self tuning regulator was probably first introduced in [18]. An algorithm for single-input single-output linear systems (SISO) consisting of a least squares estimator and a linear controller has been proposed. However, not much work had been done in this area till the beginning of the seventies. The potential use of the recursive estimation algorithms in the field of adaptive control stimulated further interest in them. The advent of process computers and microprocessors provided added stimulus to the study of the synthesis of controllers and the implementation of recursive parameter estimation techniques. Astrom [19] brought self tuning con-

trol into the stochastic environment. The implementation aspects of the self tuning regulators have been discussed in [20]. Recently there have been a flurry of activities in applying neural networks to the control systems. An attempt has been made to utilize neural networks as implicit self tuning controllers in [14]. While the approach proposed by the authors could also be used to control non-linear systems, examples of only the linear systems have however been provided. Some methods of using the neural networks as explicit self tuning controllers have been given in [15] where nonlinear systems are treated and means of utilizing multilayer and recurrent networks are given. But the explicit methods of self-tuning are computationally at a disadvantage when compared to the implicit ones. This is because of the indirect way of the controller synthesis. Further, the structure of the plant is to be known. An excellent survey of the importance of neural networks from a control systems perspective and also the future areas for research have been given in [21]. The main focus is not the promise of artificial neural networks in the realm of modelling, identification and control on nonlinear systems. The design of practically viable controllers using neural networks , based on results in the nonlinear control theory has been dealt with in [17]. In [22], the objective as been to clarify the neural networks characteristics by comparison with the adaptive control theory. Stability analysis of the neural networks have also been touched upon. The convergence properties of the neural networks have been discussed in [23]. In [24] a detailed procedure for the convergence algorithm is given. The author has explained the backpropagation.

and its applications to the control problem. The issue of utilizing the neural networks control system continues to be a hot topic of research these days. This led for example, to complete issues of the IEEE control systems magazine being devoted to it in the last few years. Quite a few papers in these issues deal with the adaptive control of systems using the neural networks [16],[25]. With the increase in the demand for sophistication and accuracy in industry, the system requirements are getting complex and design processes cumbersome. The models which developed are nonlinear with many degrees of freedom. Conventional methods are applicable for optimal control of linear systems,but often requires a great deal of ad-hoc adjustment or compensation for nonlinear systems. Thus developing conventional controllers can be a time consuming task. Conventional control algorithms must be transferred to and executed by Von-Neuman CPU controllers, which usually requires tedious assembly language programming (at least for the most efficient code generation, in terms of time and money) to synthesize a real-time control system. Looking for an alternative which would cater to the above needs, artificial neural network techniques offers the opportunity to design adaptive controllers which

- Adapt to changing hardware
- Adapt to varying objective (error cost) functions
- Are suitable for implementation in a parallel processor acting as the controller thereby eliminating the programming and coding task.

With the increase in the computing capabilities it is becoming more conducive to handle complex and length computations of complex problems. In the present study, neural networks is applied to a simplified yet practical model of a rotor bearing system to exhibit the capabilities of neural networks as a controller. This work can be extended to incorporate more features in the model of rotor bearing system like, increasing the number of degrees of freedom. including the nonlinearity and so on. Earlier the neural networks was used as an adaptive controller for complex nonlinear problems like controlling inverted pendulum, active suspension systems etc. but to the best of our knowledge it has not been applied to rotor bearing system. Using backpropagation as a learning rule, a neural network controller is developed using the principle of inverse dynamics.

Chapter 3

Mathematical Modelling of Rotor-Bearing Systems

3.1 Rotor-Bearing System

In order to develop a dynamic model of a rotor-bearing system and to obtain a suitable set of oil-film bearing coefficients, the relevant literature has been reviewed. The bearings significantly influence the rotor's dynamic behavior. For many years rotating shafts and bearings were not considered to interact with each other [26]. A study of the relevant literature reveals that a considerable attention had been focussed on determination of stiffness and damping coefficients of fluid-film journal bearings for use in linearized lateral rotor vibration analysis [27] - [28]. Parallel advancements in computer based algorithms, measurement devices accelerated both

theoretical and experimental approaches to tackle the problems of rotor-bearing systems.

Studies have been conducted to provide a mean of selecting values of oil-film bearing parameters such that the fit between the measured and computed values can be optimized [29] - [30]. Investigation have been carried out to study the behavior of rotor-bearing systems with squeeze-film dampers [2]. It is found that experimental values of the stiffness and damping coefficients obtained by using the squeeze-film dampers are in good agreement with that of theoretical values. For the most general rotor-bearing system twelve coefficients are proposed, namely,four inertia, four damping and four stiffness terms. A survey of the relevant literature further reveals that a full twelve coefficients model is appropriate in some cases and in other cases a simplified eight or six coefficients model may be used [31], [8]. This chapter also deals with the development of the mathematical model of the rotor-bearing systems. A controller developed later will be applied to this model to control the vibrations, using the latest concept of neural network. It is also assumed that the control forces are applied on all masses in both vertical and horizontal direction. The rotor-bearing systems can be modelled by considering the rotor and the bearing as lumped masses and connected by a massless shaft. The method described here is commonly used to study the influence of bearings and supports on the rotor vibrations.

The configuration of Model is shown in figure 3.1. It comprises of a rotor modelled as a two lumped masses and connected by a massless shaft Figure 3.2. The rotor is symmetrically supported on two plain oil-film bearings. each of which is characterized by linearized coefficients. The bearing housings are mounted on rigid supports [4]

In the dynamic analysis of the model linearized stiffness and damping coefficients of hydrodynamic bearings are used. The analysis of a rotor-bearing systems is greatly facilitated by adoption of linearized models of bearing stiffness and damping coefficients [6], [32]., generally the cross-coupling terms, $K_{xy}, K_{yx}, C_{xy}, C_{yx}$ of oil film were assumed to be negligible [32], [10]. This assumption obscures the fundamental notion of controllability of the lateral vibration models from a single input [6]. Hence in deriving the mathematical model of the rotor-bearing systems we have considered the cross-coupling effects of the stiffness and damping of the oil-film.

3.2 Equations of Motion of Model

The four degree of freedom model is shown in 3.1. Each mass of the two lumped mass system has two degree of freedom along the vertical and horizontal axes. The rotor-bearing system has a journal mass m_b and midspan mass m_f per bearing. x_b, x_f and y_b, y_f are the horizontal and vertical displacements of the two masses m_b and m_f respectively. The stiffness of rotor per bearing is k_f .

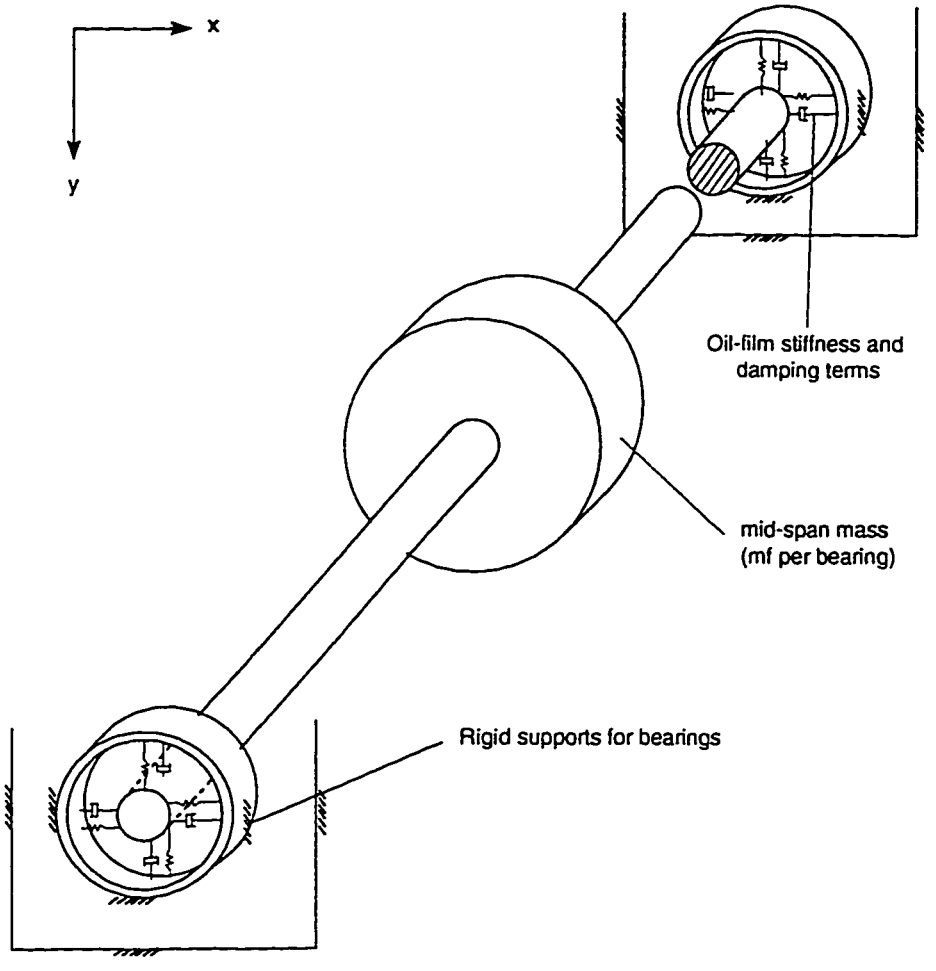

Figure 3.1: Physical Model of Rotor-Bearing System

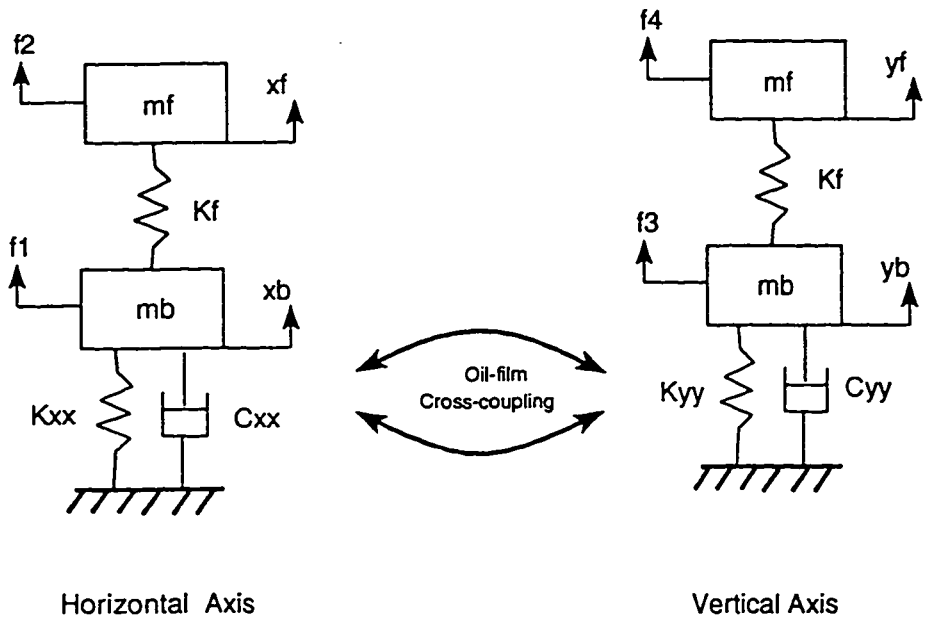

Figure 3.2: Schematic diagram of a vibrating rotor-bearing system

f_1 and f_2 are the inputs applied to m_b and m_f in horizontal direction and f_3 and f_4 are the inputs applied to m_b and m_f in vertical directions. The following equations of motions are derived using the laws of motion.

$$\ddot{x}_b = \frac{f_1}{m_b} - \frac{K_{xx}}{m_b}(x_b) + \frac{K_{xy}}{m_b}(y_b) - \frac{C_{xx}}{m_b}(\dot{x}_b) - \frac{C_{xy}}{m_b}(\dot{y}_b) + \frac{K_f(x_f - x_b)}{m_b} \tag{3.1}$$

$$\ddot{x}_f = \frac{f_2}{m_f} - \frac{K_f(x_f - x_b)}{m_f} \tag{3.2}$$

$$\ddot{y}_b = \frac{f_3}{m_b} - \frac{K_{yy}}{m_b}(y_b) + \frac{K_{yx}}{m_b}(x_b) - \frac{C_{yy}}{m_b}(\dot{y}_b) - \frac{C_{yx}}{m_b}(\dot{x}_b) + \frac{K_f(y_f - y_b)}{m_b} \tag{3.3}$$

$$\ddot{y}_f = \frac{f_4}{m_f} - \frac{K_f(y_f - y_b)}{m_f} \tag{3.4}$$

3.2.1 State Variable Representation of the Model

A very powerful approach to treat a dynamic system is the state-space formulation. The advantage is that the concept of state is best suited to the capability of high-speed solution of differential equations by the use of the digital Computer.

It is clear from figure 2 that eight state variables are required to define the motion of the system in both vertical and horizontal directions. A suitable set of variables are as follows

Horizontal Axis:

$x_1 = x_b$,journal displacement

$x_2 = \dot{x}_b$,journal velocity

$x_3 = x_f$, mid-span displacement

$x_4 = \dot{x}_f$, mid-span velocity

Vertical Axis:

$x_5 = y_b$,journal displacement

$x_6 = \dot{y}_b$,journal velocity

$x_7 = y_f$, mid-span displacement

$x_8 = \dot{y}_f$, mid-span velocity

Writing the equations 3.1 to 3.4 in the state space form

$$\dot{x}_1 = x_2 \tag{3.5}$$

$$\dot{x}_2 = -\frac{(K_{xx} + K_f)}{m_b}x_1 - \frac{C_{xx}}{m_b}x_2 + \frac{K_f}{m_b}x_3 - \frac{K_{xy}}{m_b}x_5 - \frac{C_{xy}}{m_b}x_6 + \frac{f_1}{m_b} \tag{3.6}$$

$$\dot{x}_3 = x_4 \tag{3.7}$$

$$\dot{x}_4 = \frac{K_f}{m_f}x_1 - \frac{K_f}{m_f}x_3 + \frac{f_2}{m_f} \tag{3.8}$$

$$\dot{x}_5 = x_6 \tag{3.9}$$

$$\dot{x}_6 = -\frac{(K_{yy} + K_f)}{m_b}x_5 - \frac{C_{yy}}{m_b}x_6 + \frac{K_f}{m_b}x_7 - \frac{K_{yx}}{m_b}x_2 - \frac{C_{yx}}{m_b}x_2 + \frac{f_1}{m_b} \tag{3.10}$$

$$\dot{x}_7 = x_8 \tag{3.11}$$

$$\dot{x}_8 = \frac{K_f}{m_f}x_5 - \frac{K_f}{m_f}x_7 + \frac{f_4}{m_f} \tag{3.12}$$

It is clear from that equations that the cross coupling between the vertical and horizontal axes is introduced by bearing cross-stiffness and cross-damping coefficients.

3.2.2 Eigenvalues

The eigenvalues of the coefficient matrix 'A' are obtained at different shaft speed, with and without cross coupling terms are given in tables 3.1, 3.2 and ??. The coefficient matrix A is given in Appendix A.

When we consider the system without the cross coupling terms, the eigenvalues all eigenvalues are negative. The eigenvalues are calculated at a shaft speed of 500 rpm, 1500 rpm, 3000 rpm, 15000 and even higher. At all rotor speeds the eigenvalues were found to be negative, thus indicating that the system is stable at all shaft speeds. This is given in table 3.1 and table 3.2. The influence of the cross-coupling terms ($K_{xy}, K_{yx}, C_{xy}, C_{yx}$), on the stability of the system can be observed from this

500 rpm	1500 rpm
1000000 *	100000 *
-7.239556301E-05 - 0.00002979439476i	-0.00232575226462 - 0.00095292179644i
-7.239556301E-05 + 0.00002979439476i	-0.00232575226462 + 0.00095292179644i
-8.815804228E-05 - 0.00106259498597i	-0.00252032746742 - 0.00971759906738i
-8.815804228E-05 + 0.00106259498597i	-0.00252032746742 + 0.00971759906738i
-1.218025255E-05 - 0.00107244993677i	-0.00035666028164 - 0.01066773942123i
-1.218025255E-05 + 0.00107244993677i	-0.00035666028164 + 0.01066773942123i
-0.40059612072813	-1.32826328186927
-2.90579168695725	-9.68377548944027

Table 3.1: Eigenvalues of the rotor bearing system without Cross Coupling terms.

3000 rpm	10000 rpm
100000 *	100000 *
-0.00610564820949	-0.00041442318745 - 0.00648403124238i
-0.00261971187466 - 0.00671249438131i	-0.00041442318745 + 0.00648403124238i
-0.00261971187466 + 0.00671249438131i	-0.00089297001879 - 0.00947837579919i
-0.00844557188294	-0.00089297001879 + 0.00947837579919i
-0.00065667538744 - 0.01049040958584i	-0.01684114153723
-0.00065667538744 + 0.01049040958584i	-0.09882031006273 - 0.01776069185852i
-0.65194126059813	-0.09882031006273 + 0.01776069185852i
-4.83817687045367	-1.43627008962538

Table 3.2: Eigenvalues of the rotor bearing system without Cross Coupling terms.

500 rpm	6500 rpm
1000000 *	100000 *
-7.797739471E-05 - 0.00006504907923i	-7.66157416E-06 - 0.00600206451403i
-7.797739471E-05 + 0.00006504907923i	-7.66157416E-06 + 0.00600206451403i
-0.00015548685142 - 0.00104818877876i	-0.0009133076751 - 0.00997328119123i
-0.00015548685142 + 0.00104818877876i	-0.0009133076751 + 0.00997328119123i
-1.149330821E-05 - 0.00107248553838i	-0.01249926021981
-1.149330821E-05 + 0.00107248553838i	-0.08583270096424 - 0.03734739974248i
-0.2275094643214	-0.08583270096424 + 0.03734739974248i
-3.07873389597098	-2.35763438043092

Table 3.3: Eigenvalues of the rotor bearing system with Cross Coupling terms.

6550 rpm	6600 rpm
100000 *	100000 *
-3.39797186E-06 - 0.00600673878000i	7.7090757E-07 - 0.00601135977162i
-3.39797186E-06 + 0.00600673878000i	7.7090757E-07 + 0.00601135977162i
-0.000914358886 - 0.00996617465736i	-0.00091534471863 - 0.00995910735047i
-0.000914358886 + 0.00996617465736i	-0.00091534471863 + 0.00995910735047i
-0.01258639241619	-0.01267373421261
-0.08516467809344 - 0.03891708464412i	-0.08450666921117 - 0.04039239406116i
-0.08516467809344 + 0.03891708464412i	-0.08450666921117 + 0.04039239406116i
-2.33947261203318	-2.32158474595586

Table 3.4: Eigenvalues of the rotor bearing system with Cross Coupling terms.

table. At lower shaft speeds all the eigenvalues are negative but at higher speeds (e.g 6600 rpm and higher) two of the eigenvalues are positive complex numbers. It implies that the cross coupling terms must be included in the model to identify bearing characteristics and to control vibrations. This is given in table 3.3 and table 3.4.

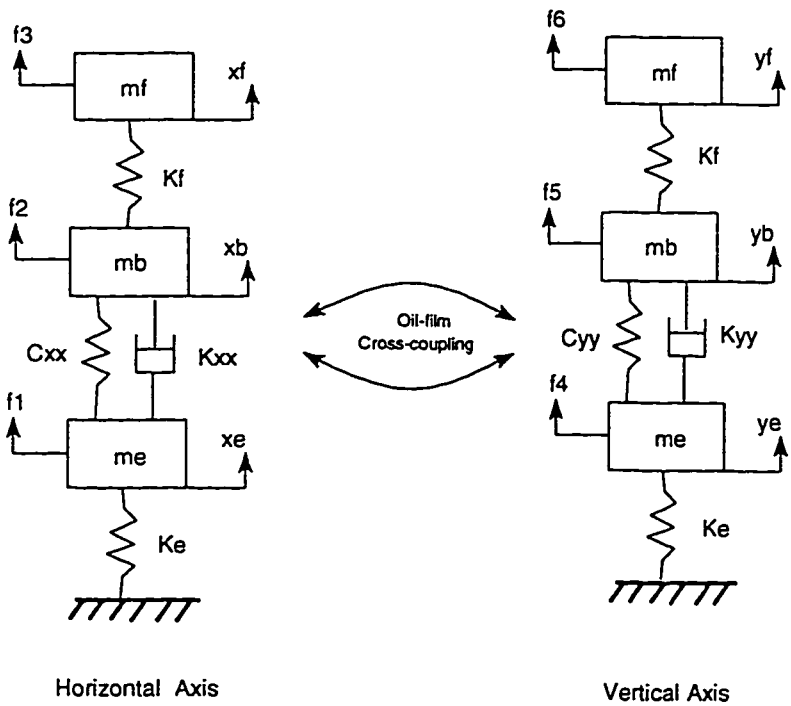

Figure 3.3: Schematic diagram of a vibrating rotor-bearing system for Model-II

3.3 Equations of Motion of Model-II

A schematic diagram to a vibrating rotor-bearing system for model-II is shown in figure 3.3. The system has three masses, two masses (m_b and m_f) are same as those in previous model and the third mass is the bearing housing mass (m_e) and consequently the system has six degrees of freedom. In this case f_1,f_2,f_3 are the inputs applied to the bearing housing, journal and mid-span masses respectively in the vertical direction and f_4,f_5,f_6 are the inputs applied to the bearing housing, journal and mid-span respectively in the horizontal direction. x_e,x_b and x_f are the displacements of bearing housing (m_e), journal mass(m_b) and mid-span mass(m_f) in vertical directions and $y_e, y_b and y_f$ are the displacements of bearing housing m_e, journal mass m_b and mid-span mass m_f respectively in the horiaontal directions. The following equations are derived by applying second law of motion:

$$\ddot{x}_e = \frac{f_1}{m_e} + \frac{K_{xx}}{m_e}(x_b - x_e) + \frac{K_{xy}}{m_e}(y_b - y_e) + \frac{C_{xx}}{m_e}(\dot{x}_b - \dot{x}_e) + \frac{C_{xy}}{m_e}(\dot{y}_b - \dot{y}_e) - \frac{K_e}{m_e}(x_e)$$

$$\ddot{x}_b = \frac{f_2}{m_b} - \frac{K_{xx}}{m_b}(x_b - x_e) - \frac{K_{xy}}{m_b}(y_b - y_e) - \frac{C_{xx}}{m_b}(\dot{x}_b - \dot{x}_e) - \frac{C_{xy}}{m_b}(\dot{y}_b - \dot{y}_e) - \frac{K_f}{m_b}(x_f - x_b)$$

$$\ddot{x}_f = \frac{f_3}{m_f} - \frac{K_f}{m_f}(x_f - x_b)$$

$$\ddot{y}_e = \frac{f_4}{m_e} + \frac{K_{yy}}{m_e}(y_b - y_e) + \frac{K_{yx}}{m_e}(x_b - x_e) + \frac{C_{yy}}{m_e}(\dot{y}_b - \dot{y}_e) + \frac{C_{yx}}{m_e}(\dot{x}_b - \dot{x}_e) - \frac{K_e}{m_e}(y_e)$$

$$\ddot{y}_b = \frac{f_5}{m_b} - \frac{K_{yy}}{m_b}(y_b - y_e) - \frac{K_{yx}}{m_b}(x_b - x_e) - \frac{C_{yy}}{m_b}(\dot{y}_b - \dot{y}_e) - \frac{C_{yx}}{m_b}(\dot{x}_b - \dot{x}_e) + \frac{K_f}{m_b}(y_f - y_b)$$

$$\ddot{y}_f = \frac{f_6}{m_f} - \frac{K_f}{m_f}(y_f - y_b)$$

State Variable Representation of Model II

A suitable set of state-variable for Model II are as follows:

The state variables in the vertical directions are :

$x_1 = x_e$= bearing housing displacement; $x_2 = \dot{x}_e$= bearing housing velocity;

$x_3 = x_b$= journal displacement; $x_4 = \dot{x}_b$= journal velocity;

$x_5 = x_f$= mid-span displacement; $x_6 = \dot{x}_f$= mid-span mass velocity;

The state variables in the horizontal direction are:

$x_7 = y_e$= bearing housing displacement; $x_8 = \dot{y}_e$= bearing housing velocity;

$x_9 = y_b$= journal displacement; $x_{10} = \dot{y}_b$= journal velocity;

$x_{11} = y_f$= mid-span displacement; $x_{12} = \dot{y}_f$= mid-span mass velocity;

Hence, the state-variable equations of the equations of the model are as follows:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{(K_{xx}+K_e)}{m_e}x_1 - \frac{C_{xx}}{m_e}x_2 + \frac{K_{xx}}{m_e}x_3 + \frac{C_{xx}}{m_e}x_4 - \frac{K_{xy}}{m_e}x_7 - \frac{C_{xy}}{m_e}x_8$$
$$+\frac{K_{xy}}{m_e}x_9 + \frac{C_{xy}}{m_e}x_{10} + \frac{f_1}{m_e}$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -\frac{(K_{xx}+K_f)}{m_b}x_3 - \frac{C_{xx}}{m_b}x_4 + \frac{K_{xx}}{m_b}x_1 + \frac{C_{xx}}{m_b}x_2 - \frac{K_{xy}}{m_b}x_9 - \frac{C_{xy}}{m_b}x_{10}$$
$$+\frac{K_{xy}}{m_b}x_7 + \frac{C_{xy}}{m_b}x_8 + \frac{K_f}{m_b}x_5 + \frac{f_2}{m_b}$$

$$\dot{x}_5 = x_6$$

$$\dot{x}_6 = \frac{K_f}{m_f}x_3 - \frac{K_f}{m_f}x_5 + \frac{f_3}{m_f}$$

$$\dot{x}_7 = x_8$$

$$\dot{x}_8 = -\frac{(K_{yy}+K_e)}{m_e}x_7 - \frac{C_{yy}}{m_e}x_8 + \frac{K_{yy}}{m_e}x_9 + \frac{C_{yy}}{m_e}x_{10} - \frac{K_{yx}}{m_e}x_1 - \frac{C_{yx}}{m_e}x_2$$
$$+\frac{K_{yx}}{m_e}x_3 + \frac{C_{yx}}{m_e}x_4 + \frac{f_4}{m_e}$$

$$\dot{x}_9 = x_{10}$$

$$\dot{x}_{10} = -\frac{(K_{yy}+K_f)}{m_b}x_9 - \frac{C_{yy}}{m_b}x_{10} + \frac{K_{yy}}{m_b}x_7 + \frac{C_{yy}}{m_b}x_8 - \frac{K_{yx}}{m_b}x_3 - \frac{C_{yx}}{m_b}x_4$$
$$+\frac{K_{yx}}{m_b}x_1 + \frac{C_{yx}}{m_b}x_2 + \frac{K_f}{m_b}x_{11} + \frac{f_5}{m_b}$$

$$\dot{x}_{11} = x_{12}$$

$$\dot{x}_{12} = \frac{K_f}{m_f} x_9 - \frac{K_f}{m_f} x_{11} + \frac{f_6}{m_f}$$

The above equations are written in state space.

The neural network theory can be applied to these model for controlling the lateral vibrations of the rotor-bearing system. We have applied the neural controller to the previous model to control the its vibrations.

Chapter 4

Neural Networks and Tuning Controllers

Neural Networks can be described as an attempt by humans to create an artificial brain. In the current stage of development of Neural Networks, though, it would be more apt to describe them as a human attempt to mimic the way the brain is supposed to do things. This is in order to harness its versatility and its ability to infer and intuit from incomplete or distorted information. Learning about neural networks requires a new vocabulary. A neural network is not programmed, it's *taught*. A Neural Network's speed is measured not in terms of instructions per second but in terms of interconnection between neurons per second. The main motivation behind the development of the artificial neural network is the brain. The many different network paradigms and algorithms have been developed by researchers

in such a way as to be analogous to the supposed functioning of the brain. But then the current knowledge about the brain's overall operation is very limited. This constraints the network designers to go beyond current biological knowledge, seeking structures that perform useful functions. In many cases, this necessary shift discards biological plausibility, the brain becomes a metaphor. Networks are produced that are organically infeasible or require highly improbable set of assumptions about brain anatomy and functioning. Yet the artificial neural networks evoke comparisons with the brain since their functions are often reminiscent of human cognition. Hence it remains profitable to understand something of the mammalian nervous system - an entity which successfully performs the tasks to which our artificial systems only aspire.

4.1 Biological Neuron

The human nervous system, built of cells called neurons, is of staggering complexity. An estimated 10^{11} neurons participate in perhaps 10^{15} interconnection over transmission paths that may range for a meter of more. Each neuron is an electrically active cell. They interact with one another through the flow of local ionic currents between them. These ionic currents are driven by a voltage difference across the neuron's cell membrane.

A nerve impulse consists of a rapid voltage change which occurs in a localized

section of a neuron's membrane. Once initiated this information propagates to adjacent areas along the length of a nerve fibre which links cell bodies called somas into networks. At the end of each output fibre(axon) the electrical nerve impulses are converted into chemical signals across a synaptic gap and which may initiate electrical impulses in the input fibres(dendrites) of the sink cells.

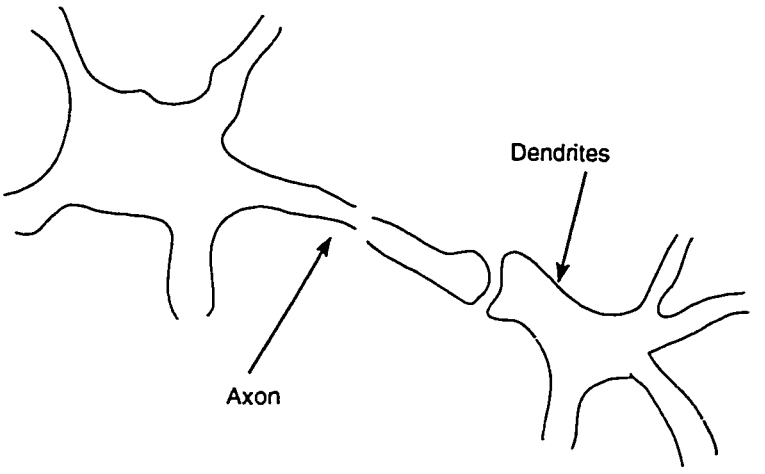

Figure 4.1: Biological Neuron

The resulting nerve impulses then travel along their dendrites to arrive at the target cells. Each cell body continually integrates the currents that arrive via its dendrites. This seemingly baroque mechanism is surprisingly powerful since not only is each neuron an independent simple processor but also at any instant of time a tremendous number of neurons can be active thus making the network mechanism highly parallel. The functioning of the artificial neural network closely parallels the,

mechanism outlines above.

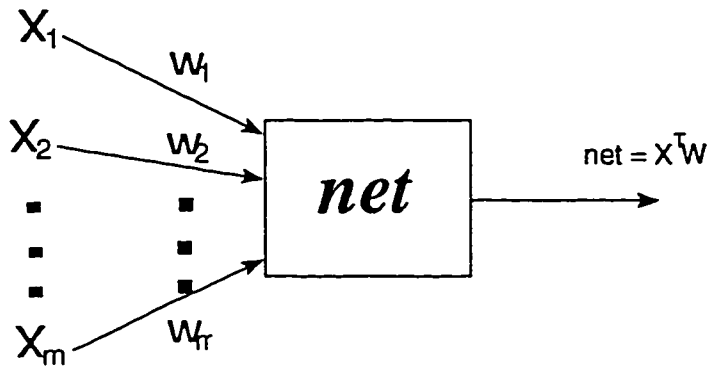

Figure 4.2: Artificial Neuron

4.2 Artificial Neuron

The artificial neuron was designed to mimic the characteristic of the biological neuron as understood by the human mind. In essence a set of input are applied to each neuron. Each input is applied over a separate link as shown in the figure 4.2. A weight is associated with each link. Each input is multiplied by the weight of the corresponding link through which it is applied, analogous to a synaptic strength. All the weighted inputs are then algebraically summed to determine the activation level of the neuron. A model that implements this idea is shown in figure 4.2. The different inputs $X_1, X_2, ...X_m$ are applied to one link each. These are multiplied by the link weight and finally algebraically summed up.

Despite the diversity of network paradigms, nearly all are based on this config-

uration. A set of inputs labeled $X_1, X_2, ...X_m$ is applied to the artificial neuron. Each signal is then multiplied by an associated weight $w_1, w_2, ...w_m$. Finally it is applied to the summation block. The summation block algebraically adds up all the inputs and sends the resultant output which is denoted as *net*. In vector notation, if X denotes a vector whose i^{th}component is X_i and W is a weight vector whose i^{th} component is w_i then, the resultant net can be expressed as

$$net = X^T W \tag{4.1}$$

4.2.1 Activation function

The signal *net* is usually further processed by an activation function f as shown in figure 4.3 to produce the neuron's output signal $\hat{Y}$.

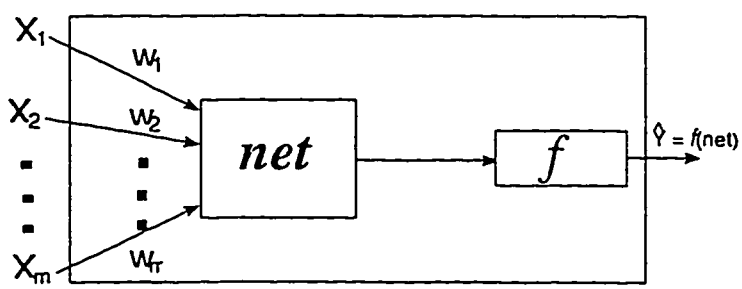

Figure 4.3: Artificial Neuron with Activation Function

Generally the activation function used is the so called *squashing function* expressed mathematically as

$$\hat{Y} = f(net) \tag{4.2}$$

where

$$f(net) = K\left(\frac{2}{1+e^{-c\,net}} - 1\right) \tag{4.3}$$

This function is taken from Simon Haykin [33].This is a generalized kind of sigmoidal function, whose slope can be varied by changing a parameter and also its saturation point can be varied. In the equation 4.3, K gives the saturation point. If it is set to 1, then the function is between ≤ 1; or if it is set to 2 then the output range will be ± 2 and so on. The parameter c has also a very important role to play. It gives the slope of the sigmoidal function. If c is large then the function will saturate very fast, if it is small then the function will be flat and hence output will reach its maximum value specified by K very slowly. It is shown later the behavior of the value of c has on the controller performance. If a very small value of c is choosen then it will have no effect on the system, where as a very large value of c will tend to destabilize the system and the Weight Matrix will be a ill-trained one.

Figure 4.4 depicts the sigmoidal logistic function. Another nonlinear activation function which commonly used is the hyperbolic tangent . It is shown in figure 4.5. Mathematically it is expressed as

$$f(net) = \tanh(net) \tag{4.4}$$

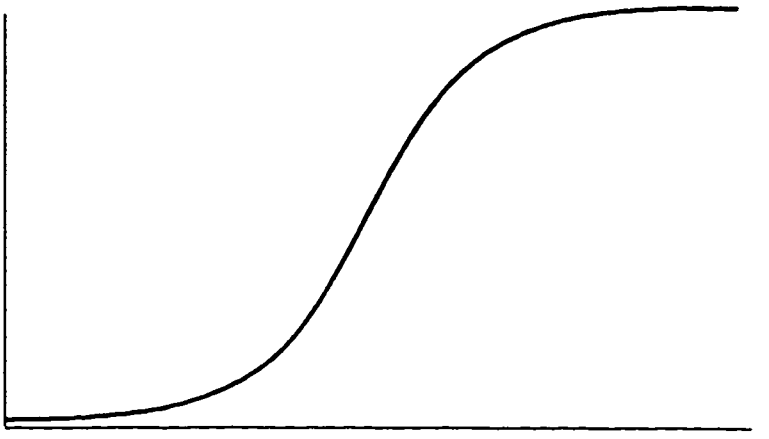

Figure 4.4: Sigmoidal Function

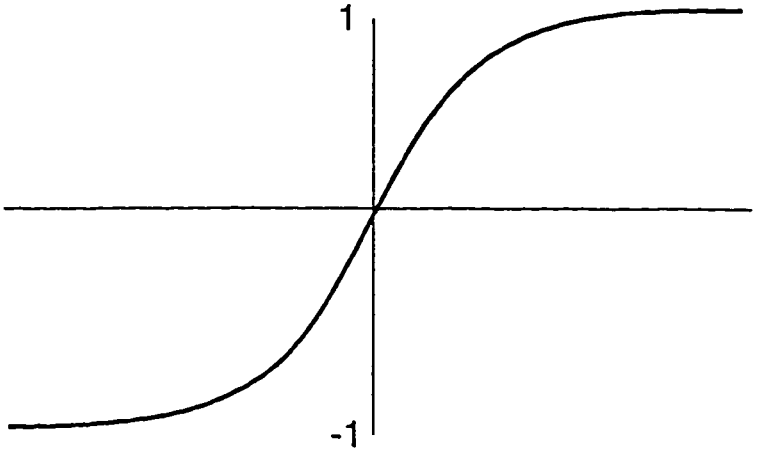

Figure 4.5: Sigmoidal Function

$$= \frac{1 - e^{-net}}{1 + e^{-net}}$$

4.3 Feedforward Network

Before we specify a learning rule, we have to define exactly how the outputs of a neural net depend on its inputs and weights. There are several configurations proposed by the researchers.

4.3.1 Single layered network

The power of neural computation comes from connecting the neurons into networks. The simplest network consists of a group of neuron arranged in a layer as shown in figure 4.6.

Neurons present in the same layer are not connected with each other in this type of structure. The circular nodes on the extreme left in figure 4.6 serve only to distribute the input. They perform no computation and hence they are also referred to as zeroth layer. The set of inputs X has each of it's elements connected to each neuron through a link each of which has a weight associated with it. Each neuron simply outputs the algebraic weighted sum of the inputs to the network, acted upon by the activation function. Actual networks may have many of the connections deleted.

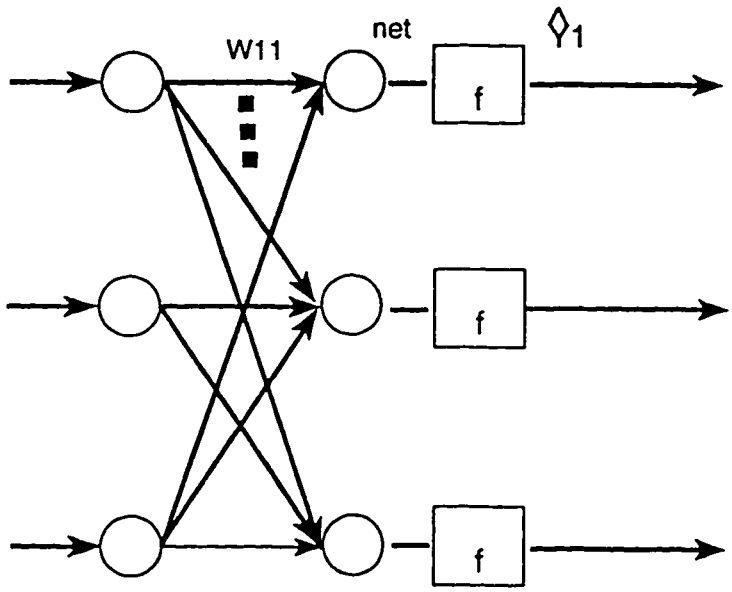

Figure 4.6: Single-Layer Neural Network

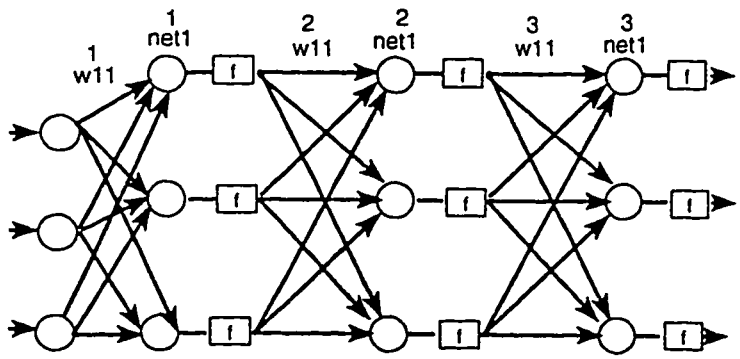

Figure 4.7: Multi-Layered Neural Network

4.3.2 Multilayer Artificial Neural Networks

Greater computational capabilities are offered by large, more complex multilayered networks as shown in figure 4.7. Each neuron in a layer is completely connected with all the neurons in the next layer. Neurons in the same layer are not connected. This is feedforward neural network architecture. Although networks have been constructed in every imaginable configuration, arranging the neurons in layers mimics the layered structure of certain portions of the brain. We have followed the structure presented by Werbos [24] as shown in the figure 4.8. Multilayered neural networks have been proven to have capabilities beyond those of single layer. But the non-linear activation functions are vital to the expansion of the network's capabilities beyond that of the single-layer network since without these functions multilayered networks provide no advantage in flexibility over a single layer network [34]. Much of the current fascination with neural networks has to do with their ability to learn. The most popular learning algorithm today is the backpropagation, which can be implemented rather easily in a microcomputer [35]. The network is trained is that application of a set of inputs produces a desired of at a least consistent set of outputs.

In this work, we have assumed the following logic :

$$x_i = X_i \qquad 1 \leq i \leq m \tag{4.5}$$

$$net_i = \sum_{j=1}^{i-1} W_{ij} x_j \qquad m \leq i \leq N + n \tag{4.6}$$

$$x_i = f(net_i) \qquad m < i \leq N + n \tag{4.7}$$

$$\hat{Y}_i = x_{i+N} \qquad 1 \leq i \leq n \tag{4.8}$$

where m is the number of the inputs to the network, X_i are the actual inputs to the network, the function f in the equation 4.7 is explained in the section above, and N is a constant which can be any integer you choose as long as it is no less than m. The value of $N + n$ decides how many neurons are there in the network (if we include the inputs as neurons). A brief explanation of the concept of layers is given in the next section. x_i is sometimes called the *activation level* of the neuron. The significance of these equations is illustrated in the figure 4.8.

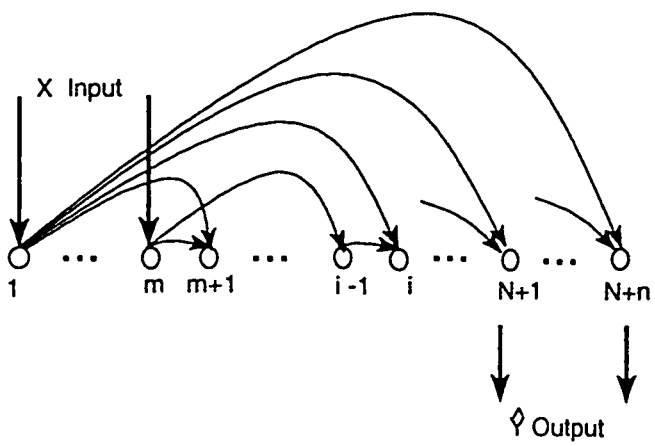

Figure 4.8: Network design for Backpropagation

There are $N + n$ circles, representing all of the neurons in the network, including the input neurons. The first m circles are really the copies of the inputs $X_1, X_2, \ldots, X_m$; they are included as a part of the vector x only as a way of simplifying the notation. Every other neuron in the network such as neuron number i, which calculates net_i and x_i , takes input from every cell which precedes it in

the network. Even the last output cell, which generates $\hat{Y}_n$, takes input from other output cells, such as the one which outputs $\hat{Y}_{n-1}$.

4.4 Backpropagation training algorithm

The proposition of the backpropagation algorithm has played a central role in the resurgence of interest in artificial neural network. This is mainly because of the limited capability of single layered neural network. Backpropagation is a systematic method for training multilayered artificial neural network.

The backpropagation algorithm is a gradient descent search technique. Let $Y(t)$ be the target output of the network and $\hat{Y}(t)$ be the actual output of the network. The difference between the target output and the actual output is the error denoted as E. In backpropagation we choose the weights W_{ij} so as to a minimize square error function over the training set:

$$E = \sum_{t=1}^{T} E(t) = \sum_{t=1}^{T} \sum_{i=1}^{n} (1/2)(\hat{Y}_i(t) - Y_i(t))^2 \tag{4.9}$$

Where T is the total number of training patterns and n is the number of outputs for each pattern. The aim of backpropagation algorithm is to minimize the error E, through an iterative search in weight space W. this is because the actual output $\hat{Y}$ is a function of its weights. This is simply a special case of the well-known method of least squares, used often in statistics, econometrics, and engineering; the uniqueness

of backpropagation lies in how this expression is minimized. The approach used here is illustrated in figure 4.9. In backpropagation we start with arbitrary values

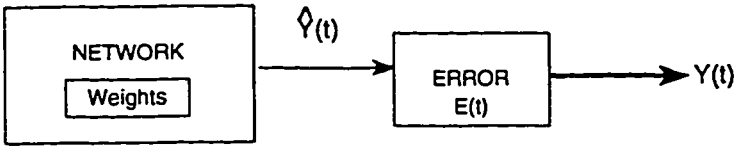

Figure 4.9: Backpropagation

for the weights **W**.(It is usual to choose random numbers in the range from -0.1 to 0.1, but it may be better to guess the weights based on prior information, in cases where prior information is available). Next we calculate the outputs $\hat{Y}(t)$ and the errors $E(t)$ for that set of weights. Then we calculated the derivatives of E with respect to all of the weights; this is indicated by the dotted line in the figure 4.9. If increasing a given weight would lead to more error, we adjust it upwards as shown in the equation 4.17. After updating all the weights according to the error, we start all over and keep on going through this process until the weights and the error settle down. (Some researchers iterate until the error is close to zero; however if the number of training patterns exceeds the number of weights in the network- as recommended by the studies on generalization-it may not be possible for the error to reach zero).

4.4.1 Calculating the Derivatives: Theoretical Background

Many papers on backpropagation suggests that we need only use the conventional chain rule for the partial derivatives of E with respect to all of the weights. Under certain conditions this can be a rigorous approach, but its generally limited,and it requires great care with the side conditions; calculations of this sort can easily become confused and erroneous when networks and applications grow complex. Werbos [24], presented new chain rule for ordered derivatives:

$$\frac{\partial^{+}TARGET}{\partial z_i} = \frac{\partial TARGET}{\partial z_i} + \sum_{j \succ i} \frac{\partial^{+}TARGET}{\partial z_j} * \frac{\partial z_i}{\partial z_i} \tag{4.10}$$

where the derivative with the superscript ∂^{+} represent ordered derivatives, and the derivative without subscripts represent ordinary partial derivatives. This chain rule is valid only for ordered systems where the values to be calculated can be calculated one by one (if necessary)in the order of $z_1, z_2, \ldots z_n$, TARGET. The simple partial derivatives represent the direct impact of z_i on z_j through the system equations which determines z_j.The ordered derivatives represents the total impact of z_i on TARGET, accounting for both the direct and indirect effects. A detailed study of this chain rule can be found in the Phd Dessertation by Werbos in 1974. For example, suppose that we had a simple system governed by the following two equations, in order :

$$z_2 = 4 * z_1$$

$$z_3 = 3 * z_1 + 5 * z_2$$

The simple partial derivative of z_3 with respect to z_1 (the direct effect) is 3; to calculate the simple effect we only look at the equations which determines z_3. However, the ordered derivative of z_3 with respect to z_1 is 23 because of the indirect impact by way of z_2. The simple partial derivative measures what happens when we increase z_1(eg.by 1, in this example) and assume that everything else (like z_2)in the equation which determines z_3 remains constant. The ordered derivative measures what happens when we increase z_1, and also recalculate all other quantities-like z_3 - which are later than z_1 in the causal ordering we impose on the system.

This chain rule provides a straightforward, plodding recipe for how to calculate the derivatives of a given TARGET variable with respect to all of the inputs (and parameters) of an ordered differentiable system in only one pass through the system. We can write the ordered derivative of the TARGET with respect to z_i as " F_z_i," which may be described as " the feedback to z_i". In backpropagation, the TARGET variable of interest is the error E. This changes the appearance of our chain rule in the case to

$$F\_z_i = \frac{\partial E}{\partial z_i} + \sum_{j \succ i} F\_z_i \times \frac{\partial z_j}{\partial z_i} \tag{4.11}$$

4.5 Adapting the Network: Equations

For a given set of weights **W**, it is easy to use the forward pass calculations equations 4.5 to 4.8, 4.3 and 4.9 to calculate the $\hat{Y}(t)$ and E(t) for each pattern t. The trick is in how we then calculate the derivatives. Let us use the prefix "$F\_$" to indicate the ordered derivative of E with respect to whatever variable the "$F\_$" precedes. Thus, for example,

$$F\_\hat{Y}(t) = \frac{\partial E}{\partial \hat{Y}_i(t)} = \hat{Y}(t) - Y(t) \tag{4.12}$$

which follows simply by differentiating equation 4.3. By the chain rule for ordered derivatives as expressed in 4.11,

$$F\_x_i(t) = F\_\hat{Y}_{i-N}(t) + \sum_{j=i+1}^{N+n} W_{ij} \times F\_net_j(t) \tag{4.13}$$

where

$$i = N + n, \ldots, m + 1$$

$$F\_net_i(t) = f'(net) \times F\_x_i(t) \tag{4.14}$$

where

$$i = N + n, \ldots, m + 1$$

$$F\_W_{ij} = \sum_{t=1}^{T} F\_net_i(t) \times x_j(t) \tag{4.15}$$

where f' is the derivative of $f(net)$ as defined in equation 4.3. Note how 4.13 requires us to run backwards through the network in order to calculate the derivatives, as illustrated in figure 4.10: this backwards propagation of information is what gives

backpropagation its name. A little calculus and algebra, starting from 4.3, shows us that

$$f'(z) = \frac{2cK}{e^{cz}(1+e^{-cz})^2} \tag{4.16}$$

which we can use when we implement equation 4.15. Finally to adapt the weights, the usual method is to set

$$New\ W_{ij} = W_{ij} - \eta \times F\_W_{ij} \tag{4.17}$$

where the η is the learning rate and is taken as some small constant chosen on an adhoc basis.(The usual procedure is to make it as large as possible up to 1, until the error starts to diverge; however there are more analytic procedures [7] Werbos [36]).

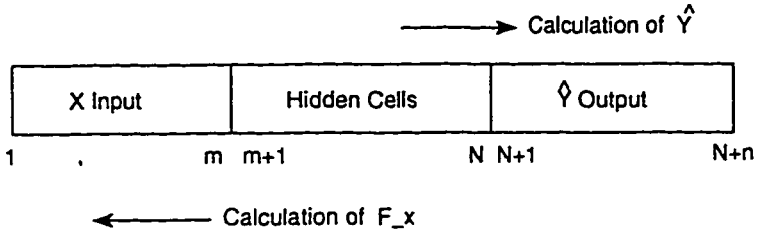

Figure 4.10: Backwards flow of derivative calculation.

4.5.1 Batch Learning and Pattern Learning

There are two ways through which one can make the neural network *learn*, depending upon the type of process. They are batch learning and pattern learning.

In batch learning, the weights are adjusted only after all T patterns are processed. It is more common to use pattern learning, in which the weights are continually updated after each observation.

In practice, maximum number of passes is usually set to an enormous number; the loop is exited only when a test of convergence is passed, a test of error size or weight change which can be injected easily into the loop.Most people using backpropagation use pattern learning with their data sets, as many passes through the data are needed to ensure convergence of the weights.

4.5.2 Training Overview

The objective of the training the network is to adjust the weights so that applications of a set of inputs produces the desires set of outputs. Training assumes that each input vector is paired with a target vector representing the desired output. Togethe r these are called a training pair. Usually a network is trained over a number of training pairs. Before starting the training process, all of the weights must be initialized to small random numbers. Training the backpropagation network requires the steps that follow.

1. Select the next training pair from the training set; apply the input vector to the network input terminals.

2. Calculate the output of the network (using the equations 4.5 to 4.8,forward pass).

3. Calculate the difference between the network output and desired output. This is error.

4. Adjust the weights of the network in a way that minimizes the error.

5. Repeat i through iv for each vector in the training set until the error for the entire set is acceptably low. Steps 1 and 2 constitute forward pass while steps 3 and 4 comprise the reverse pass.

4.6 Self Tuning Controllers

The fundamental components of a self-tuning system are discussed here. The concept of self-tuning control system is basically very simple. The two essential components of any self-tuning control system are the identification technique and the control law. By combining any identification scheme with particular control law, a wide spectrum of self tuning algorithms can be developed. In the case of conventional control schemes, the controllers developed are such that their parameters do not change with time. In other words the parameters

of these controllers are time-invariant. Thus an implicit assumption made in this case is that the parameters of the system to be controlled are constant. Hence it's assumed that a controller with constant parameters will accomplish the goals adequately. But this approach is not the best when dealing with actual systems. This is so because in everyday life, various factors such as changing environmental conditions, varying quality of the raw material as well as the effects of wear and tear on the plant performance cannot be neglected since these factors cause the plant parameters to drift and sometime s radically change. To take care of the resulting deterioration in the plant performance, one has to either change the controller which is a costly proposition or to resort to returning the controller which is the approach normally used in industry. In order to automatically overcome these practical problems associated with the control of an actual system, the self-tuning regulator has been proposed. With such a regulator, the unknown parameters of the system to be controller are estimated recursivel y. This estimation can be done using any of the established parameter estimation techniques. Once the plant parameters are estimated, the controller parameters are computed using any control law. The operation of a self tuning controller is schematically as in figure 4.11.

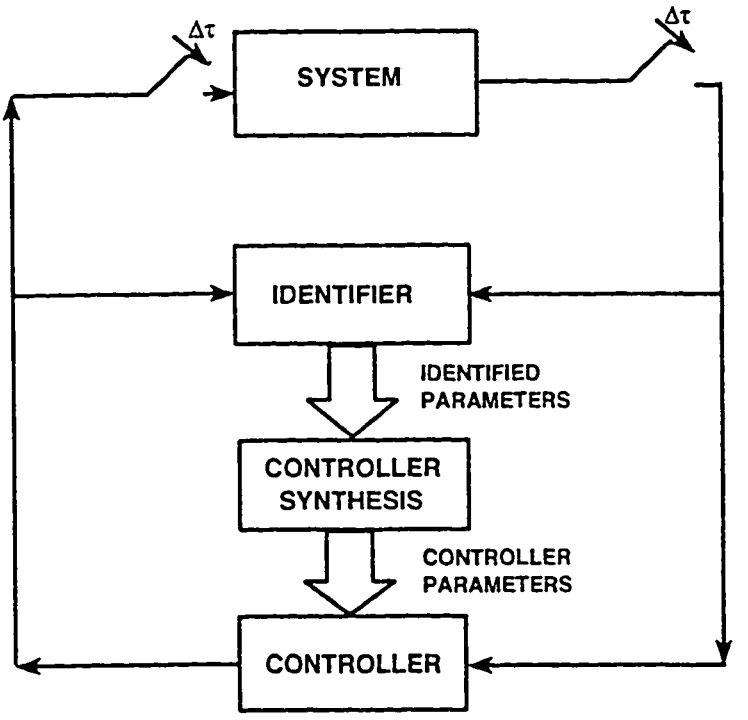

Figure 4.11: Self Tuning Controller

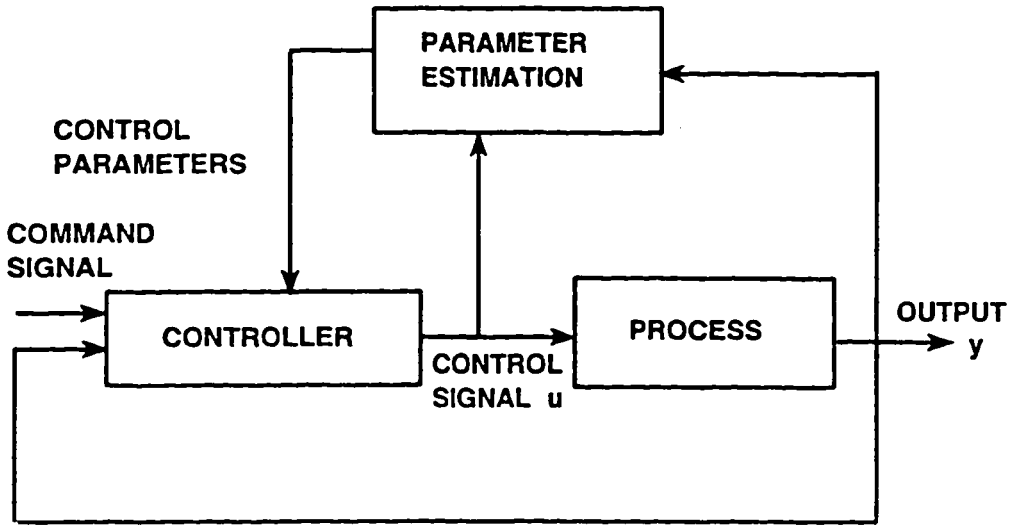

Figure 4.12: Implicit Self Tuning Controller

There are basically two broad class of algorithms depending on the intricacy of the design calculations involved. They are

- Explicit Self-Tuning Controllers
- Implicit Self-Tuning Controllers

Explicit Self-tuning Controllers

The simplest conceptual strategy is to parameterize the system in a natural way and calculate the controller parameters from the estimated plant parameters. This class of algorithms is commonly called indirect of explicit.This because the evaluation of the control law is indirectly achieved via the system model. Thus an explicit process model is used to determine the parameters of the controller using the estimated parameters in the first step. These two steps are repeated at each sampling interval. But the main disadvantage of this method is that the computation involved are many.

Implicit Self-tuning Controllers It is also possible to paramterize the system directly in terms of the controller parameters. Hence in this case , as soon as the parameters are identified the controller is directly obtained thereby obviating the need for the computing the control law again. Thus in this case the design calculations necessary to determine the controller parameters are simply the parameters in a one-step ahead predictor. This class of algorithms is commonly called direct of implicit because the control law is directly est

imated as shown in figure 4.12. One advantage of the implicit self-tuning controllers over the explicit one is that the design computations are eliminated from the used of the former ones since the controllers parameter are directly estimated, But the number of parameters involved in the design of implicit self tuning controllers many be large compared with the implicit ones. For the purpose of adaptive control of a system, recursive parameter identification has to be resorted to. Several techniques has been applied for the estimating the parameters. Some of the recursive parameter estimation methods are Least squares, Maximum likelihood, Stochastic approximation [7]. The different control strategies which can be used in the design of a self-tuning controllers are Minimum Variance technique, Generalized Minimum Variance technique, Predictive control [7]. The minimum variance and the generalized minimum variance techniques are quite sensitive to the correct determination of the model order and delay. Hence Long range predictive control methods were proposed. These methods deal with the behavior of the system over a horizon in the future as opposed to the minimum variance where the system behavior at the single point in the future is of interest.

Chapter 5

Self-Tuning Control Using Neural Networks

In recent years a lot of research activity has centered around neural networks. These network are being applied to many areas of engineering science. Two classes of neural networks which hold much interest are multilayer neural networks and the recurrent neural networks. Neural networks which have a remarkable learning ability can be viewed as class of functional representations. As a result they can be cinsidered to be dense in the space of contrinous functions and hence can be used for the modelling of even nonlinear systems. It has been shown that Cybenko [37] that any continous mapping over a compact domain can be approximated as accurately as necessary by a feedforward neu-

ral network, even with one hidden layer. This implies that fiven error $\epsilon > 0$, neural network with a sufficiently large number of nodes can be determined such that

$$|g(x) - NN(x)| < \epsilon \quad for\ allx\ in\ D$$

where g is the function to be approximated,in our case it is the rotor bearing system; D is a compact domain of a finite dimensional normed vector space. This provides the necessary theorictical basis for modelling linear as well as nonlinear systems through neural networks. Another factor responsible for the widespread use of feedforward neural networks is the simplicity of back-propagation algorithm. The above two factors are also responsible for making neural network attractive as system Identifiers and Controllers.

5.1 Neuro-Identification and Neuro-Control

5.1.1 Neuro-identification

The objective of system neuro-identification is to represent the system dynamics by neural networks. A neuro-system model can be trained to predict the actual system behavior so that several desired charactieristics of the system can be determined. The system to be identified is the rotor bearing system,

which can be represented as follows:

$$Y(t+1) = g[y(t), y(t-1), ...u(t)] \tag{5.1}$$

where $[y(t), u(t)]$ represent the output-input pair of rotor-bearing system, where the vector $y(t)$ represent the eight state variables of rotor-bearing system at time t, and $u(t)$ represent the four control forces at time t. The task is now to identify the system. One aspect of dealing with dynamic of time series data using neural networks is that both present and the past data have to be used to train the neural networks. As a result, two different learing schemes can be used to model dynamic systems. One possibility is to use the feedforward neural networks and use the actual past outputs of the system as the inputs to the neural networks. This is referred to as the series-parallel or equation error model in the literature. Another possibility is to use the estimated past outputs from the neural network as the current inputs to the network. This is referred to as the parallel of output error model. The two configurations are shown schematically in the figures 4.5 and 4.6. Mostly a series-parallel model is used. Another aspect is the selection of a scheme for different schemes have been discussed in the previous chapter. In our work we will be using pattern learning and the series-parallel identification model.

5.1.2 Neuro-control

The methods for designing neuro-controllers can be classified into three general groups [38] : neuro-controlllers trained to identify the dynamics of a reference model, neuro-controllers representing the inverse dynamics of the system to be controlled and neuro-controllers trained to optimize a performance measure of the system. We have concentrated on the second type of controller and this will be dealt in detail in the next section.

5.2 Neural Network as Self-tuning Controller

One way of using neural networks in adaptive control systems is through the self-tuning principle. As described earlier, self tuning controllers are basically of two types, namely implicit and explicit. A neural network can be used as an implicit self tuning controller provided that the controller dynamics can be directly learned by neural network. Using this approach it is possible to determine the control input directly as the output of the neural networks as the plant output follows the desired setpoint. Scheme for using the neural networks as self tuiniug controller both explicit and implicit scheme are shown in the figures 5.2, 5.1. Using a neural network as an explicit self tuning controller is not optimum from a computational point of view [7]. This is because of the extra computations involved. Hence we will be concerned with the utilization

of neural network as implicit self tuning controller. The neural network, when used as an identifier, tries to approximate the inverse dynamics of the model. If the function g in equation 5.1 is known then a neural network NN which approximates g^{-1} over the area of interest can be designed off-line. In adaptive control problems the function g is generally unknown and hence the neural network is to be trained on-line to estimate the function g using the input-output pairs. The NN has got the ability to learn the inverse dynamics of the plant. For example it will behave exactly inverse to the plant, taking in the states and giving out the forces responsible for the responses. We call this the *inverse model*.

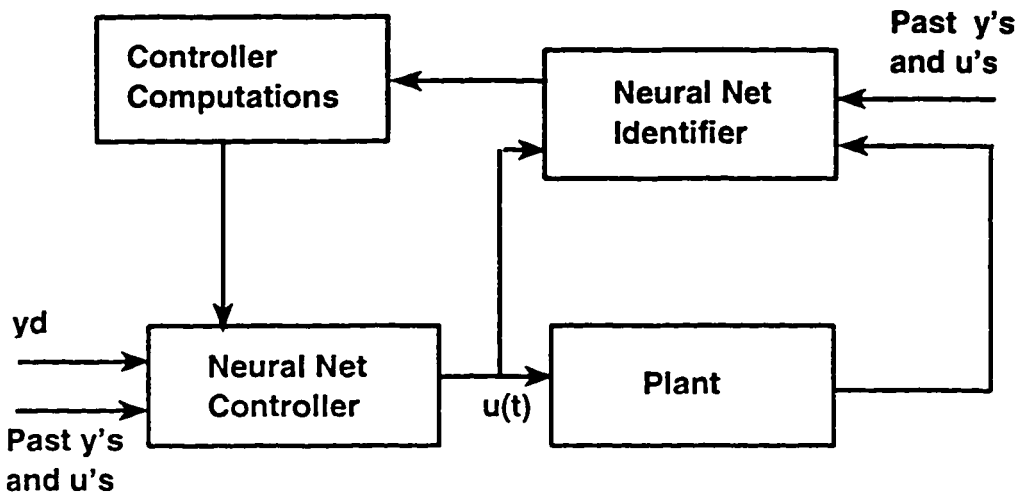

Figure 5.1: Neural Network as an Explicit Self-Tuning Controller.

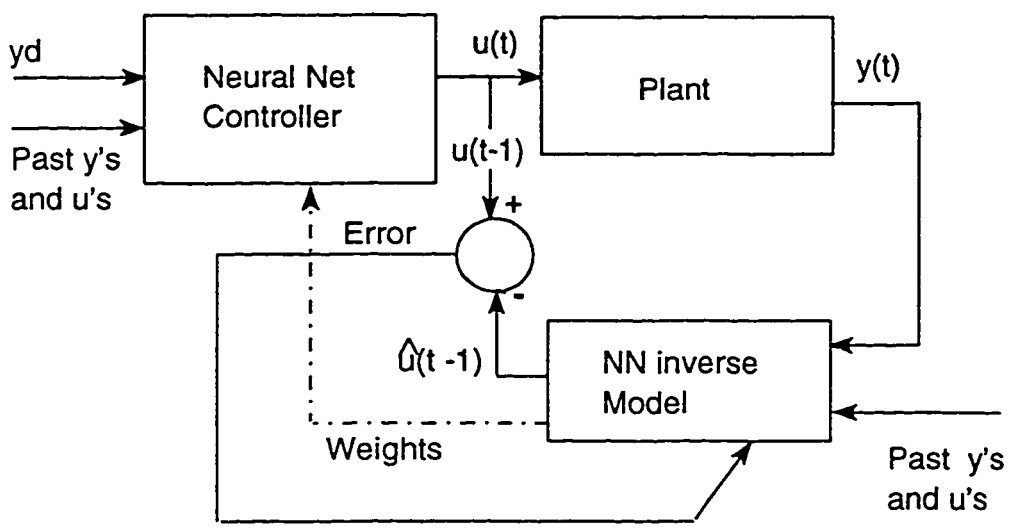

Figure 5.2: Neural Network as an Implicit Self-Tuning Controller.

The other neural network in the forward path function as the controller with its output being the actual plant input. The connection weights are adjusted such that the difference between the actual plant input and the estimated plant input (as determined by the neural network used as identifier) is minimized. This learning has been called the direct learning architecture [39]. Denoting the network by NN[y(t),u(t),W] where y and u denote the output and the input at the tth instant respectively, the estimation of g^{-1} reduces to a parameter estimation problem where the parameter to be estimated are the weights of the neural network. In the figure [39] the training error is given equation 4.9, writing this in a new form we have

$$e(t+1) = u(t+1) - \hat{u}(t+1) \tag{5.2}$$

Generally the function to be minimized is the instantaenous squared error given by

$$E = \frac{1}{2}\|e(t+1)\|^2$$

Using the well known delta rule the weight update algorithm is given in equation 4.17. The learning rate η is an important factor in the training of feedforward networks. While large values of η can lead to oscillation and instability, small values of η make the convergence very slow. In the network shown in figure 5.2, there are basically three components namely the controller, the rotor-bearing system and the inverse model.

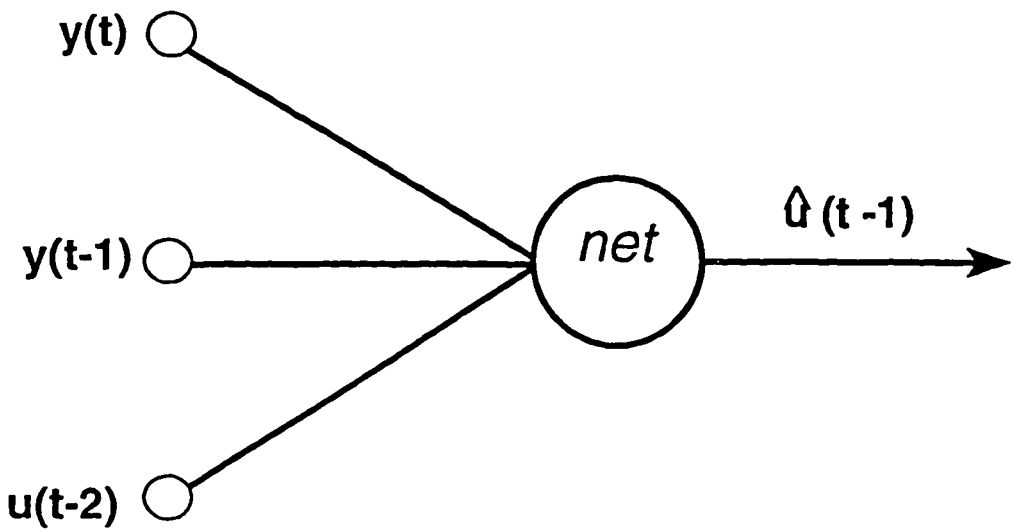

Figure 5.3: Inverse Neural model for the rotor-bearing system.

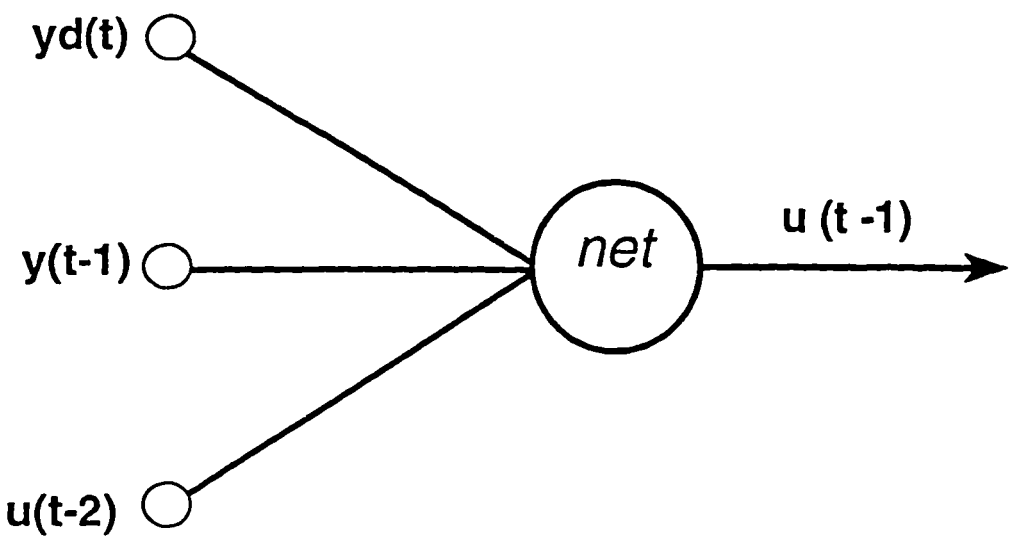

Figure 5.4: Neural Controller for the rotor-bearing system.

Inverse Model

The inputs to the inverse model are the current states $y(t)$ of the plant at time t, the previous states $y(t-1)$ at time t-1 and the previous control force $u(t-2)$. This is shown in the figure 5.3 The output $\hat{u}(t-1)$ is just an estimate for the actual control input signal $u(t-1)$ at time $t-1$ applied to the plant at time t-1. This difference between the inverse model ouput and the controller ouput is the error upon which the weight are updated upon which the new control force depends.

Controller

The inputs to the Controller are the desired states $y_d(t)$ of the plant at time t, the previous states $y(t-1)$ at time $t-1$ and the previous control force $u(t-2)$. The output of the neural controller is the one time step ahead control force $u(t-1)$ which is directly applied to the rotor bearing system. Both the neural controller and the inverse neural model form multi-input and multi-output system. There is only one weight matrix which is update at evey time step.

Rotor-Bearing System

The inputs to the rotor bearing system is the control forces generated by the neural controller $u(t-1)$, the outputs are the states $y(t)$.

5.3 Application of ANN to Inverse Dynamics Problem

5.3.1 Selection of Neural network

Number of processing elements in Input layer

The number of processing e,ements in the input layer are 8 states from Rotor Bearing system plus 4 control forces generated by the neural controller. Therefore the total number of inputs are $m = 12$. These are the input neurons which doesnt take part iin the computations. As suggested by Watanabe [14], we can increase the number of inputs to 16 or even more, i.e we can input the previous history of the states from two time steps earlier.

Number of processing elements in the Intermediate layer or Hidden layer

The number of processing elements in the hidden layer have a restriction that these should be more than the number of input neurons m. Here we have taken $N = 15$; A larger value of N will result in a larger size of weight matrix whose size is $(N + ns) \times (N + ns)$. It would take more number of iterations to train and hence more CPU time. Not always there is improvement in the solution for larger N. Also it was observed that with larger value of N, there was no considerable change in the results.

Number of processing elements in the Output layer

The number of processing elements in the output layer are the four forces generated by the neural controller; which are $ns = 4$. These four outputs ranges between -1 to $+1$. This value is fixed by taking $K = 1$. This output from the neural controller is given to the rotor bearing system. Which in turn generates states.

This network is a fully connected one. Every neuron takes a input from the previous ones except the input neurons. Figure 5.5

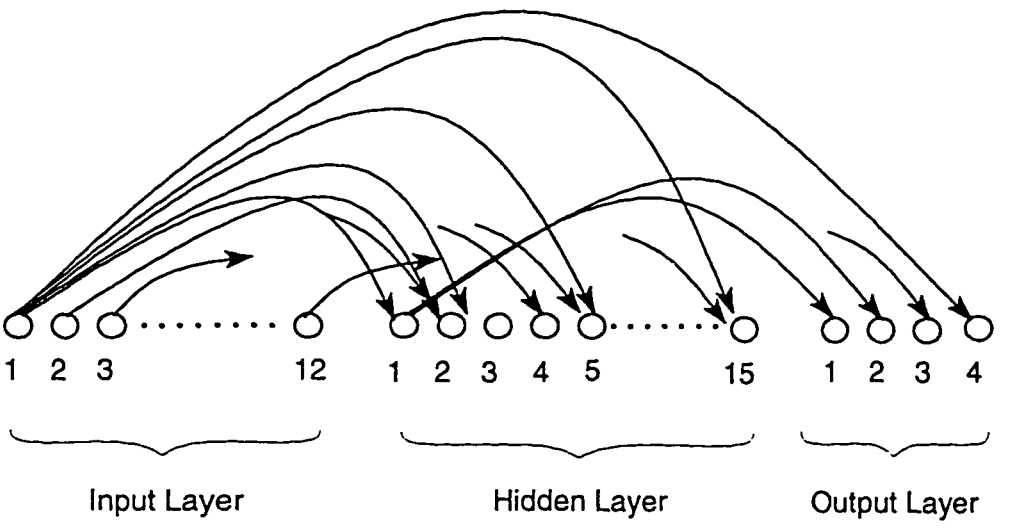

Figure 5.5: Schematic diagram of the neural network showing the number of inputs, intermediate/hidden and outputs neurons.

5.3.2 Learning Mechanism

One of the most popular method for learning is the the supervised learning. In supervised learning we try to adapt an ANN so that its actual outputs $\hat{Y}$ come close to some target output Y for a training set which contains say T patterns. The goal is to adapt the parameters of the network so that it performs well for patterns from outside the training set. To solve the supervised learning probem, there are two steps

- We must specify the "Topology" (connections and equations) for a network which inputs say X(t) and outputs Y(t). The relation between the inputs and the outputs must depend onn a set of weights "W" which can be updated.
- A learning rule should be specified, which updates the weights after each iterations so as to make the actual outputs $\hat{Y}$ approximately equal to the desired outputs Y.

Now applying this to the problem of rotor bearing system, when at time t a disturbance of unit step is applied to the rotor bearing system the states/responses of the system are generated. There are eight states for the system we are considering (four displacements and four velocities). These states are first normalized before giving as an input to the neural network along with the four control forces from the neural controller. First the twelve inputs are given to

the Inverse neural model, whose input are 12 and outputs are the estimate of the control force, which is $\hat{u}$

We start with arbitrary values for Weight matrix. It is usual to choose uniform random numbers in range from -0.1 to 0.1. After we calculate the outputs $\hat{u}(t)$ the error is calculated $E(t) = u(t) - \hat{u}(t)$ where u(t) is the control force generated by a controller. Then we calculate the derivatives of E with respect to the weights. (Explanaiton of the program is given in Appendix) Using the ?? the weights are updated and with this updated weight matrix we find the control forces. The input to the controller are the previous states. The number of the inputs to the controller are same as those of the inverse neural model. The outputs of the controller are the four controller forces $u(t)$, which are applied at different locations in the rotor bearing system under consideration.

Chapter 6

Simulation and Results

The response of the rotor bearing system without a controller owing to an external disturbance is presented along with the response of the system when the neural controller is applied.

6.1 Parameter values and Simulation

Critical dimensions of Rotor and the Journal Bearing.

The effective masses per bearing and the necessary bearing dimensions are:

Burrows and Stanway [31] considered the bearing as a multi-input multi-output plant and applying state-space techniques, they developed a coherent strategy for identification of oil-film dynamics. But their work is restricted to

Table 6.1: Rotor-bearing parameters

m_b	0.2 Kg
m_f	12.33 Kg
Load per bearing, W	189 N
Radial Clearance, cl	$0.051mm$
Diameter of journal, D	$51mm$
Land length of the bearing, L	$12mm$

rigid bodies rotors. Kamal and Asfar [40] considered a more general model of rotor-bearing system and applied system identification techniques to estimate the oil-film coefficients can be determined. However, our objective is to study the performance of the controllers so we have considered the oil-film parameters from the available data [41]. The following non-dimensional bearing parameters are used at the eccentricity ratio, $\epsilon_o = 0.8$

$k_{xx} = 9.042$, $k_{xy} = 5.326$,$k_{yy} = 1.848$,$k_{yx} = -0.674$

$c_{xx} = 9.042$, $c_{xy} = 5.326$,$c_{yy} = 1.848$,$c_{yx} = -0.674$

The computed values of oil-film stiffness coefficients are evaluated from as follows [42],[11], [40]

$$K_{xx} = k_{xx}\frac{W}{c_l}$$

Similarly we calculate the rest of the oil-film stiffness coefficients. Damping coefficients depend on the shaft speed which can be evaluated as follows

$$C_{xx} = c_{xx}\frac{W}{\omega c_l}$$

Where ω is the shaft speed in rad/s.

Simulation

The dynamic response of the mathematical model of the rotor bearing is simulated on Sun10 computer. In order to solve the differential equations of the rotor-bearing system, a program $DE.FOR$ written in fortran is used. For plotting purposes, XMGR is used to plot the results, and for various figures and flowcharts, graphical packages **Adobe Illustrator** and **Diagram** were used.

6.2 Computer Program

To solve the control problem, the implicit algorithm 5.2 was followed which resulted in a computer code written in FORTRAN. A Flowchart of this program is shown in the figure 6.1. A detailed explanation of the structure of the code is explained in Appendix B

6.3 Behavior of the System Without a Controller

The response of the vertical deflection of the mid-span of the shaft owing to a unit step external disturbance at the mid-span is shown in the figure 6.6 (solid

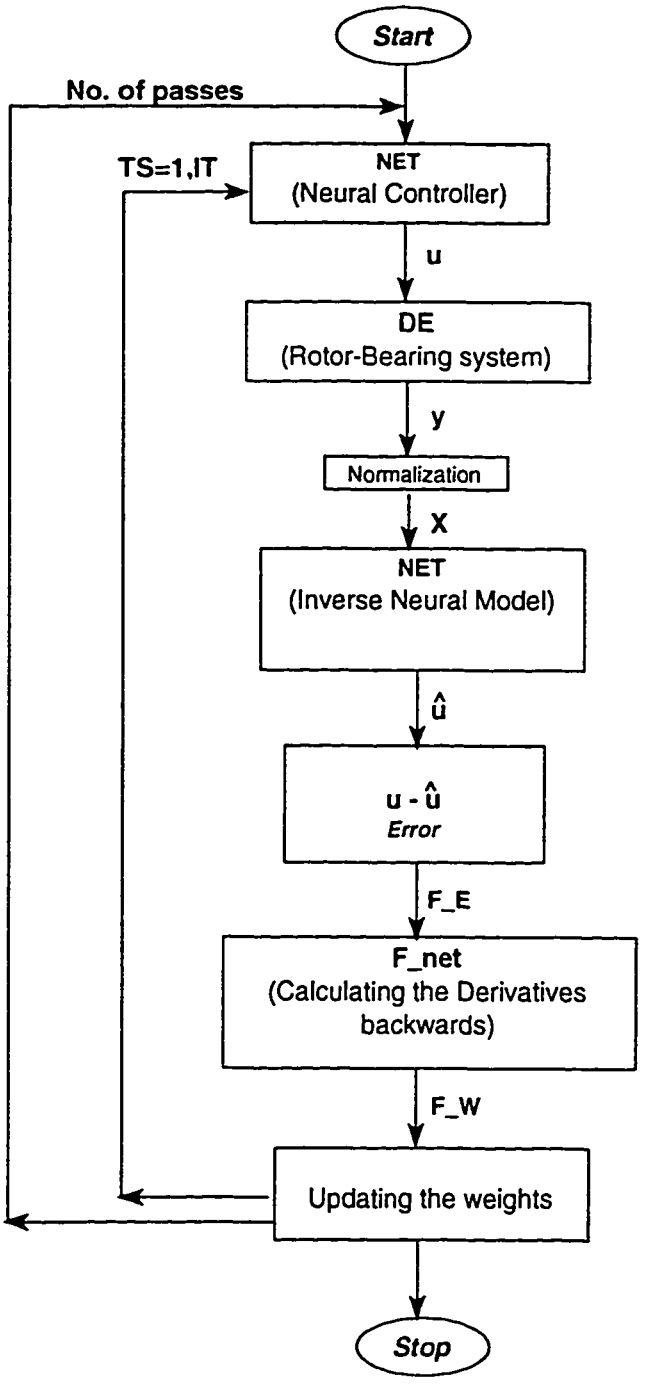

Figure 6.1: Flow chart for the Adaptive Self Tuning Control.

line). The figure displays a plot of the deflection of the shaft against time. The response is obtained for a shaft speed of 4500 rpm. Oscillations about the equilibrium position occurred and each successive amplitude diminishes from the preceding amplitude which indicates that the response is underdamped. The horizontal deflection of the mid-span is also shown in the figure 6.4 and the nature of oscillations displayed are similar. For the same disturbing input we see a similar trend in the other responses of the system. See figures 6.8, 6.2.

6.4 Behavior of the system with the Neural Controller

The neural controller was applied to the rotor bearing system running at a shaft speed of 4500rpm. The value of η was taken as 0.01, and K as 1. The sensitivity factor c in equation 4.3 play an important role. This gives more flexibility . Just by changing these 2 parameters (K and c) one can not only change the range of operation but also the controller sensitivity to the inputs. If a sensitive controller is needed, then one should go for a high value of c and vice-versa. By sensitivity it is meant that the amount of change in controller output for a given amount of change in input. The value of K was taken as

1, which gives the range of ± 1. The value of c was taken as 0.05. The higher values of c results in a very fast convergence to K, which results in a ill trained weight matrix. The response of journal in horizontal direction is shown in the figure 6.2. When the system is vibrating freely(dotted line) the second peak value is 2.480e-08 meters at time 2.000e-02 seconds. Also we observe that the steady state is achieved at the time 0.15 seconds and at the deflection of 9.183e-09. The same figure also shows the response (solid line) under the influence of the controller. It can be observed that the response is further diminished. The peak value is reduced to 8.9999e-09 occurring at time 2.0e-02 . Also we can see that the steady state is achieved at the time 8.50e-02 and at the deflection of -2.115e-10 meters. A similar trend is observed in figure 6.4. But in the responses of mid-span and journal in the vertical direction, we see a lower steady state value but the time of reaching the steady state is same for with and without controller. But the peaks are reduced and the steady state is shifted to wards zero. Table 6.2 and 6.3gives the times of steady states and the peak values at different time for all the responses.

Table 6.2: Peak values.

State	Without Controller		With Controller	
	Peak Values		Peak Values	
	time (sec)	displacement	time (sec)	displacement
Journal				
Vertical	1.5e-02	3.3112e-08	1.5e-02	2.705e-08
Horizontal	2.0e-02	2.4800e-08	2.0e-02	8.999e-09
Mid-span				
Vertical	1.5e-02	1.2663e-07	1.5e-02	1.117e-07
Horizontal	2.0e-02	3.3433e-08	2.0e-02	1.216e-08

Table 6.3: Steady State values.

State	Without Controller		With Controller	
	Steady States		Steady States	
	time (sec)	displacement	time (sec)	displacement
Journal				
Vertical	0.15	2.4391e-08	1.5e-02	2.374e-08
Horizontal	0.15	9.1839e-09	8.5e-02	-2.11e-10
Mid-span				
Vertical	0.15	9.4778e-08	0.15	7.963e-08
Horizontal	0.145	8.4103e-09	6.0e-02	-4.86e-09

6.4.1 Parametric Study

A study of the responses of the rotor bearing system is done, by varying the values of k,c and the speed of the rotor. There are many parameters which affect the performance of the controller. Some of them are the learning rate, number of maximum iterations for training the controller, speed of the shaft, sensitivity factor c and k.

Varying c

Three different values of c, namely c = 0.05, 0.15 and 0.25 are considered. Figure 6.10 and Figure 6.11 show the horizontal and vertical deflections of the mid-span of the rotor-bearing system. In Figure 6.10, at c=0.05, and k = 1, we find that the system is least affected by the controller. At c=0.15, and k=1, we observe that the vibrations are dying out early but the steady state is changed. The learning rate for all these cases was fixed as 0.01, and for 1000 iterations to train the network, and the shaft speed being 4500 rpm. At c = 0.25 and k = 1, we find that the response of the system is different and it takes more time to die the vibrations. The value of c = 0.25 can be made more effective that ie it can successfully control the vibrations by adjusting the learning rate and the number of iterations. Similar behavior is observed in the response of the mid-span in the horizontal direction as well. This is shown in figure 6.11 .

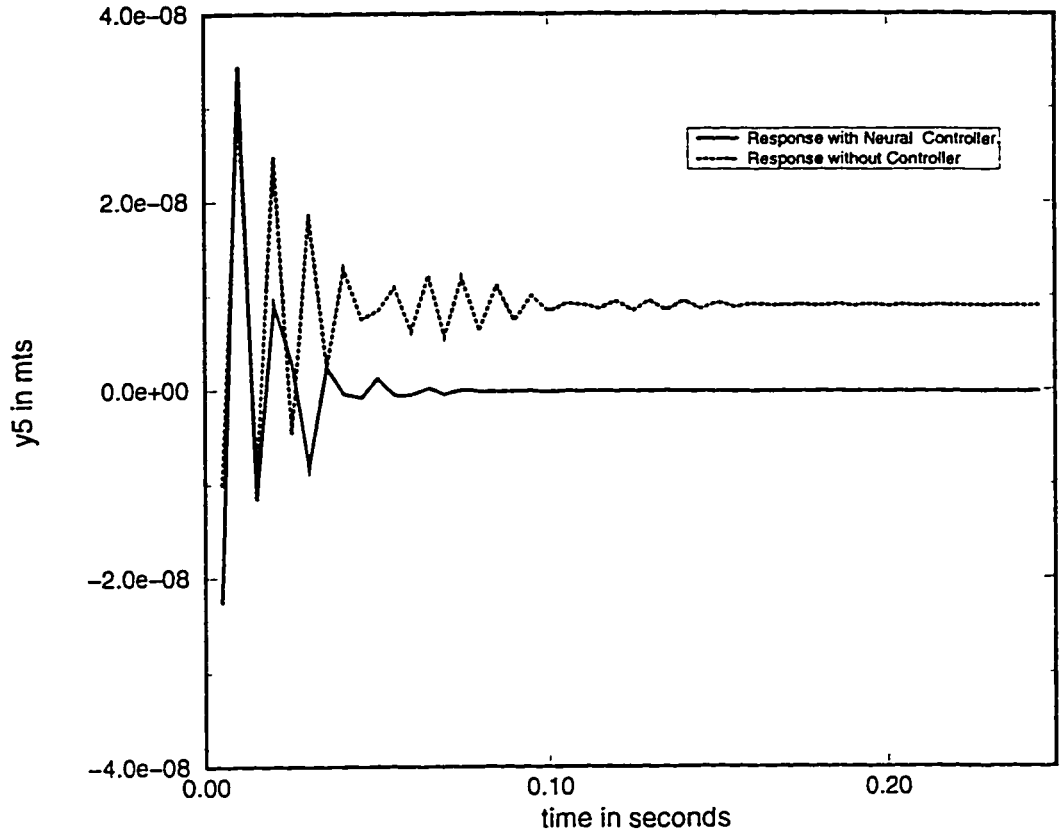

Figure 6.2: Response of the journal in horizontal direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller.

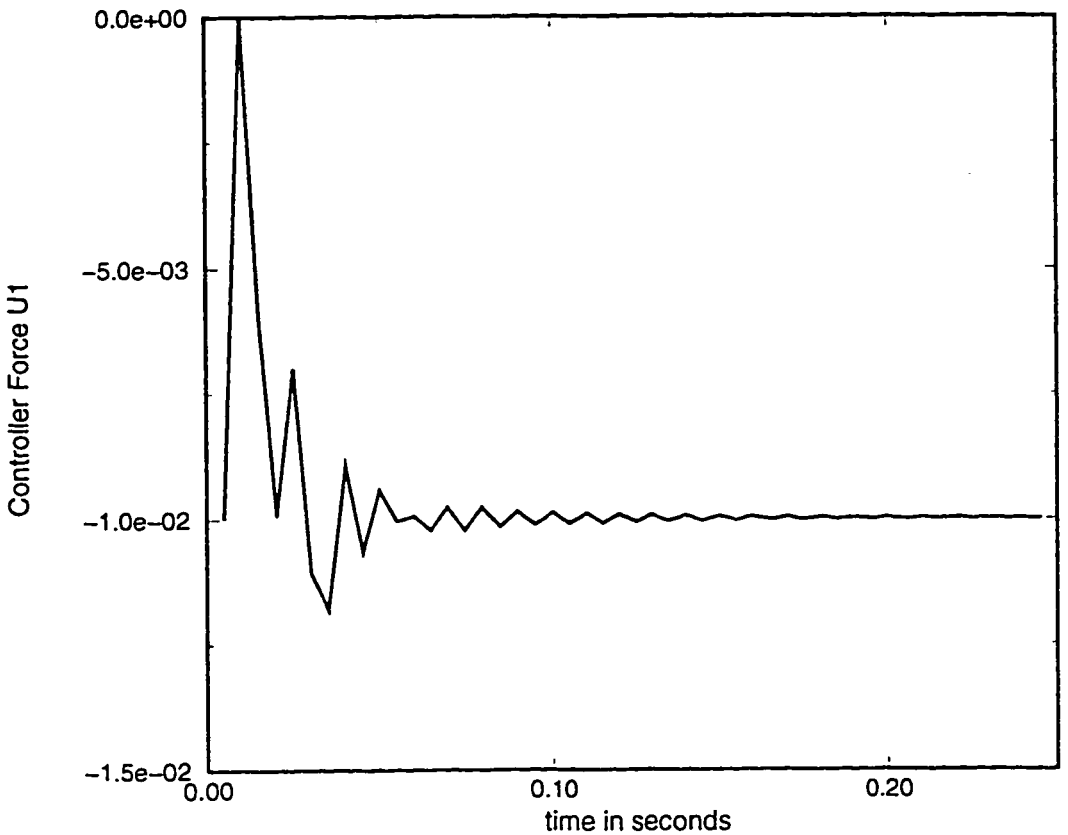

Figure 6.3: Control force generated by the neural controller which is applied to the journal in horizontal direction.

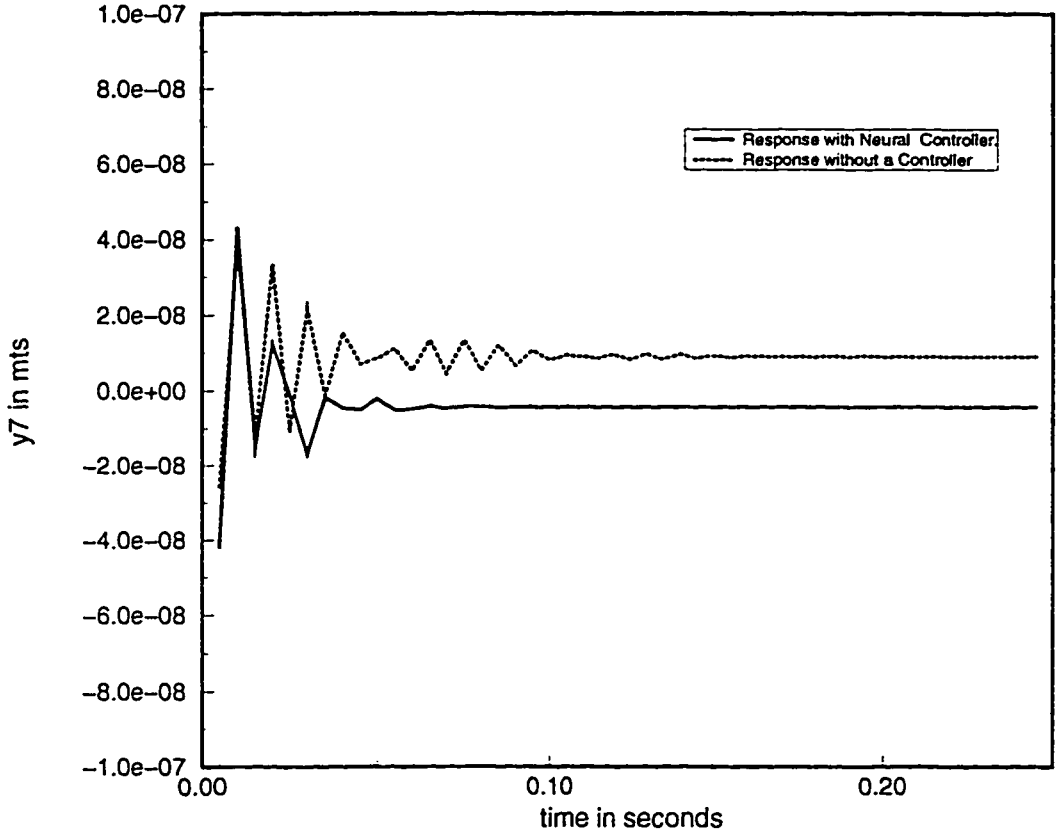

Figure 6.4: Response of the mid-span in horizontal direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller.

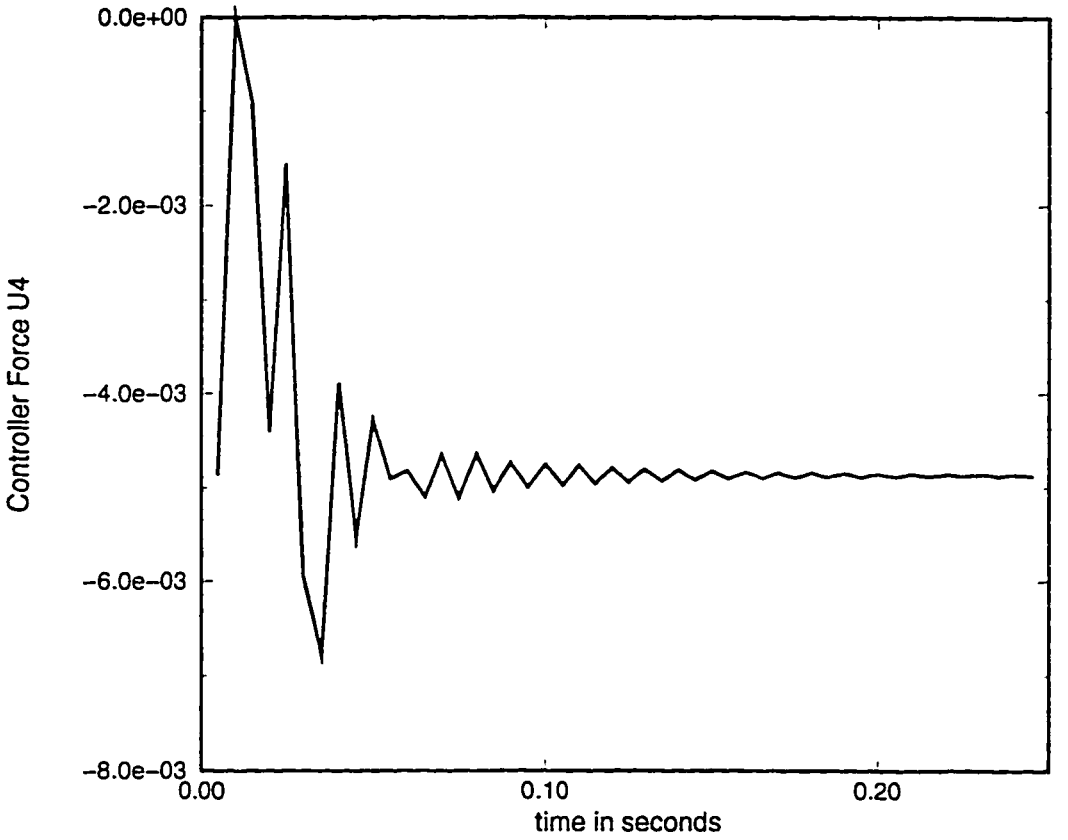

Figure 6.5: Control force generated by the neural controller to be applied to the mid-span in vertical direction.

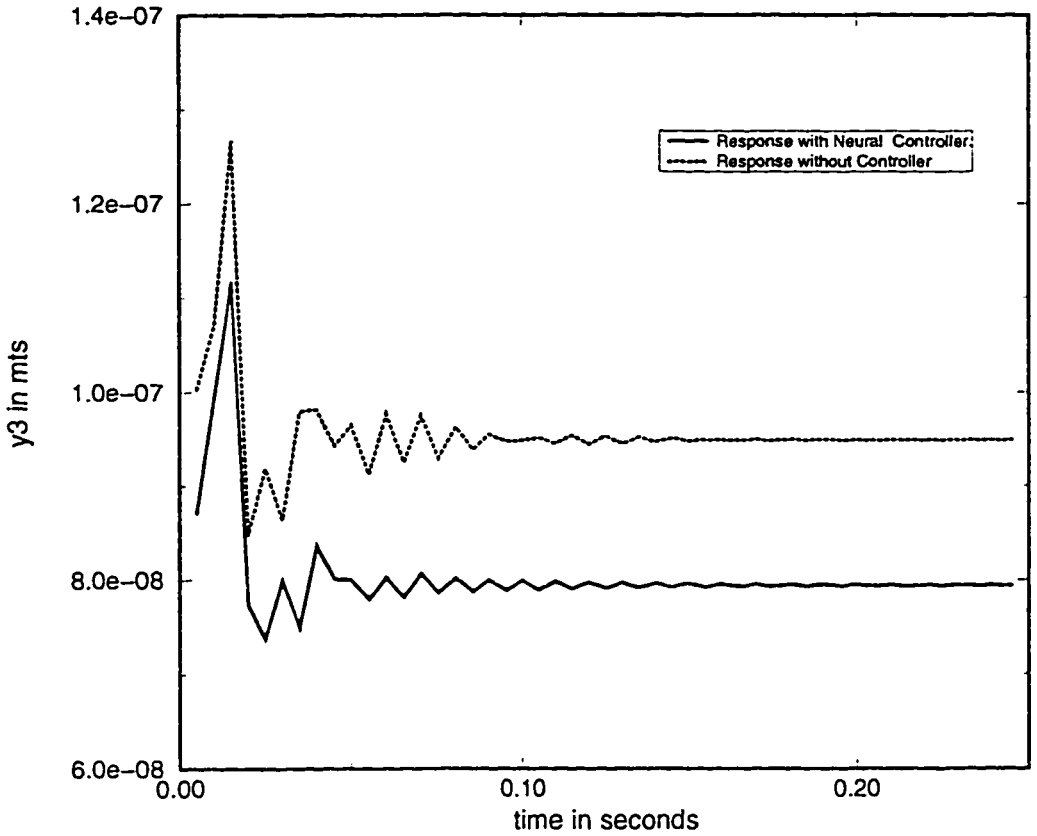

Figure 6.6: Response of the mid-span in vertical direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller.

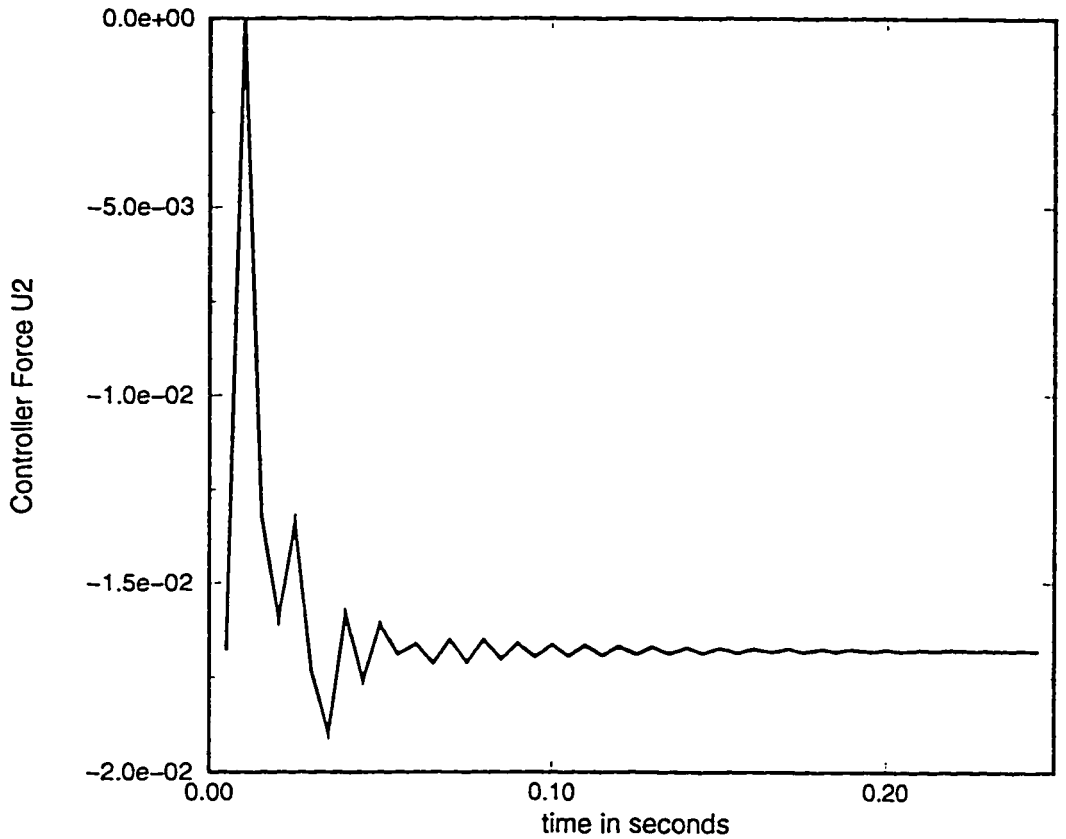

Figure 6.7: Control force generated by the neural controller which is applied to the mid-span in vertical direction.

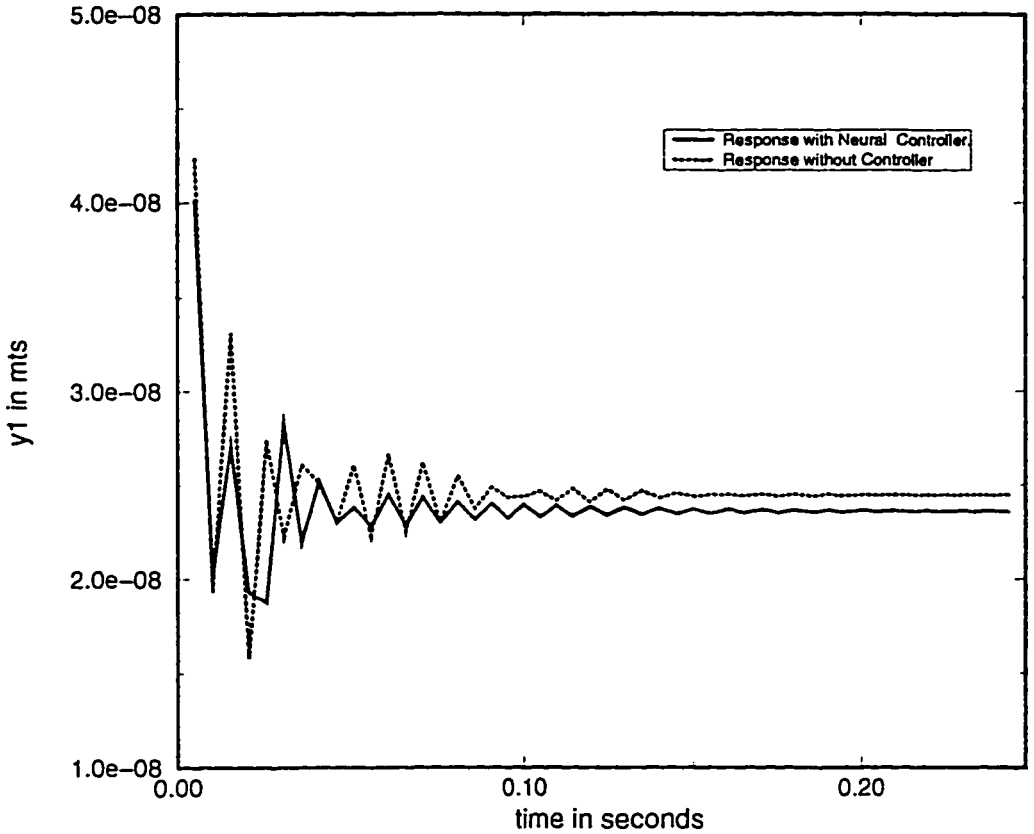

Figure 6.8: Response of the journal in vertical direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented, with and without the controller.

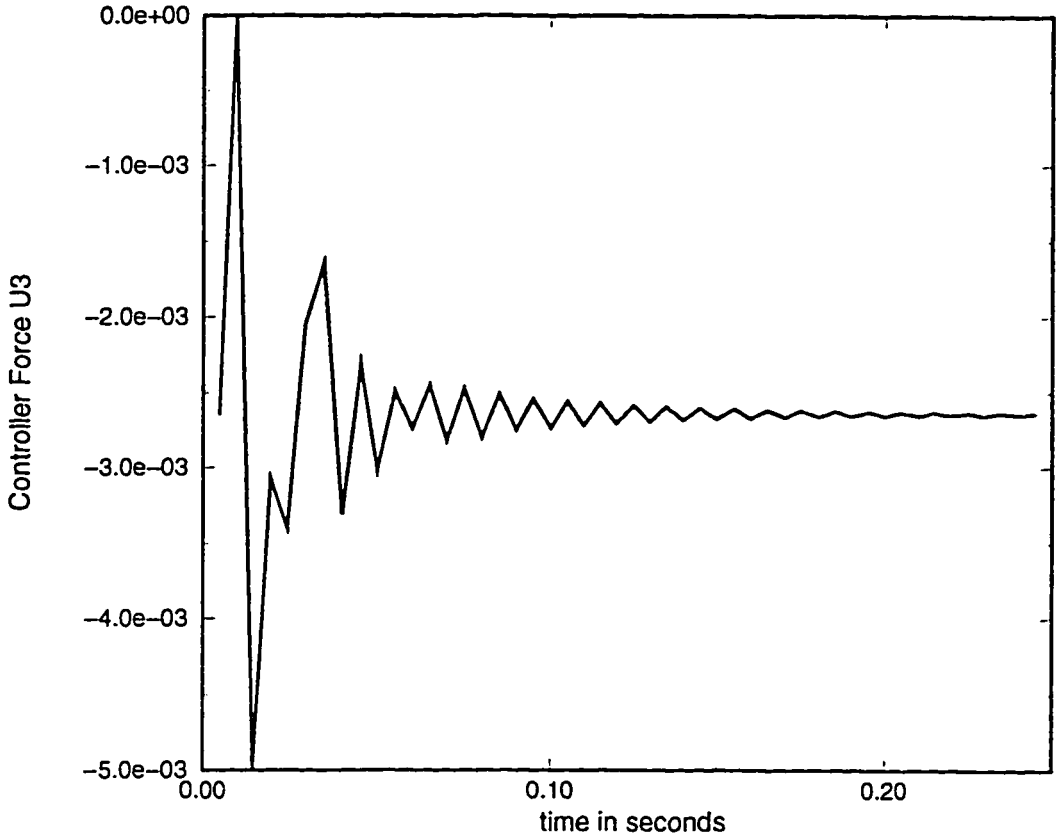

Figure 6.9: Control force generated by the neural controller which is applied to the journal in vertical direction.

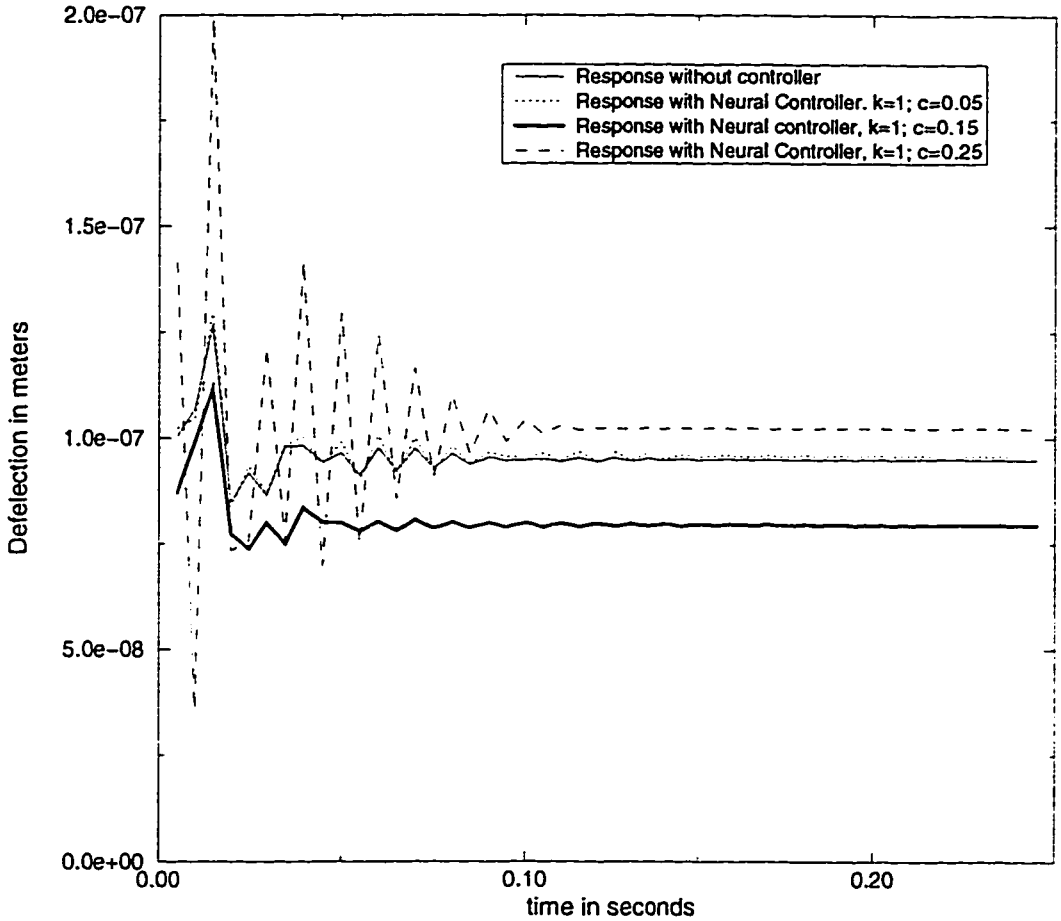

Figure 6.10: Response of the mid-span in vertical direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller. The responses are plotted by fixing k=1 and varying c

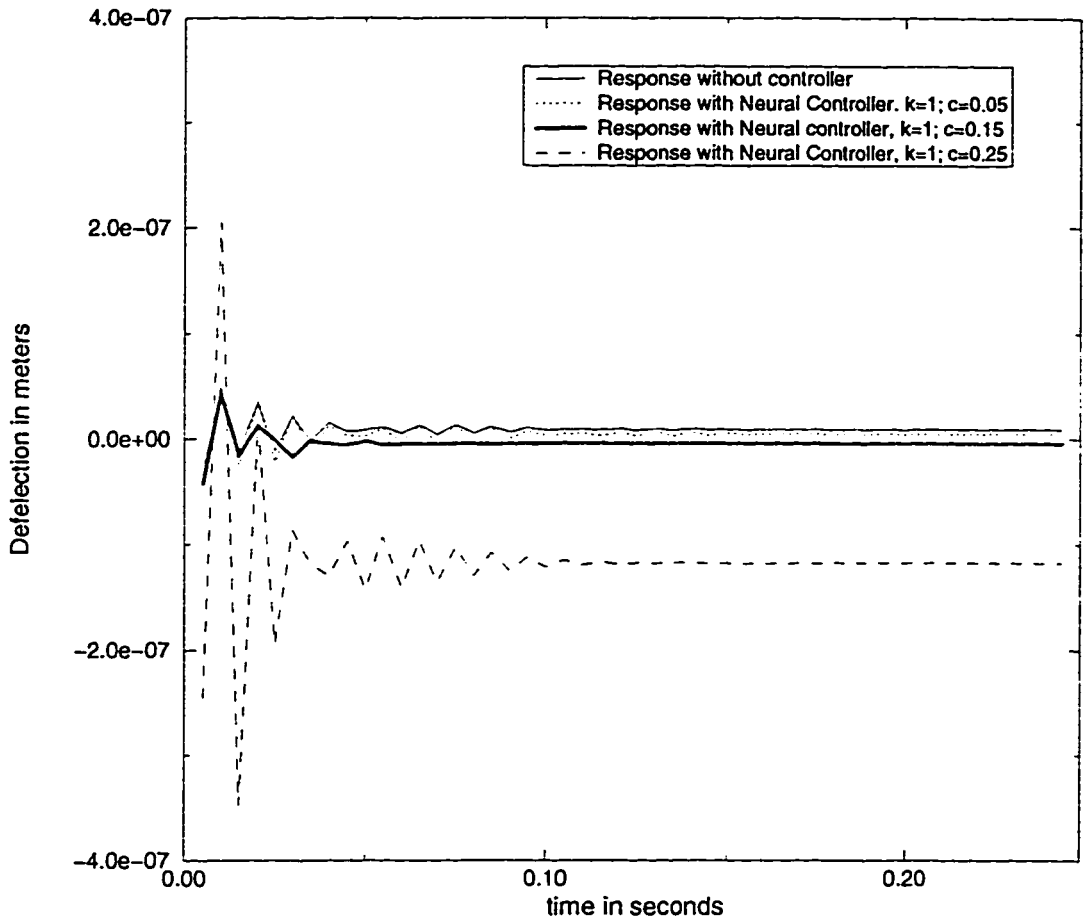

Figure 6.11: Response of the mid-span in horizontal direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller. The responses are plotted by fixing k=1 and varying c

Varying k

Three diffrent values of k namely, k=1,2,3 are considered. Figures 6.12 6.13 show the vertical and horizontal response of the mid-span of the rotor bearing system. Here k acts as the magnifying factor for the signal generated from the controller. To study the effect of k, the value of c was taken as 0.15. The study is done for k=1, 2 and 3. The behavior at c=0.15, and k=1 is explained earlier. At c=0.15 and k=2, we observe a marked change in the behavior of the system, again this can be corrected by varying the values of η and maximum iterations for these values of k and c. A similar trend is observed for the value of k=3. We can say that at some η and maximum iterations, the optimal values of c and k can be selected by observing the behavior of the system.

Varying RPM

By fixing $\eta = 0.01$, maximum iterations as 1000, k=1, c=0.15 and by varying the rpm 1500,2500, 3500 and 4500, the behavior of the system is studied. At shaft speed of 1500rpm, the vibrations takes more time to die out. Its steady state occurs at 1.0e-07. For shaft speeds of 2500 and 3500 the amplitude is almost same as before but dampensa little more quickly. In the case of shaft speed of 4500 the amplitude drops to attain lower steady states values. Higher shaft values have been tried but they gave unstable behavior, which shows that we were operating on the natural frequency of the system. We can say that the control is achieved with the particular band of rotor speed under the

particular choice of k and c.

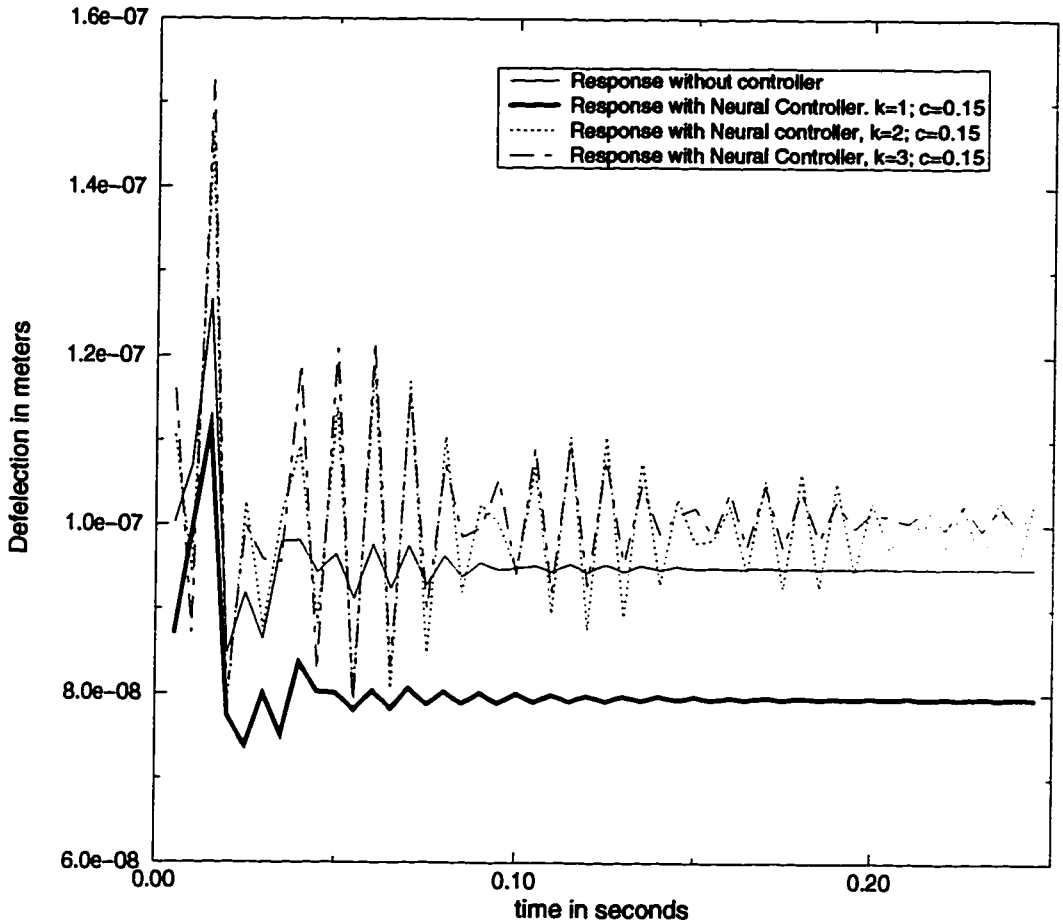

Figure 6.12: Response of the mid-span in vertical direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller. The responses are plotted by fixing k=1 and varying c

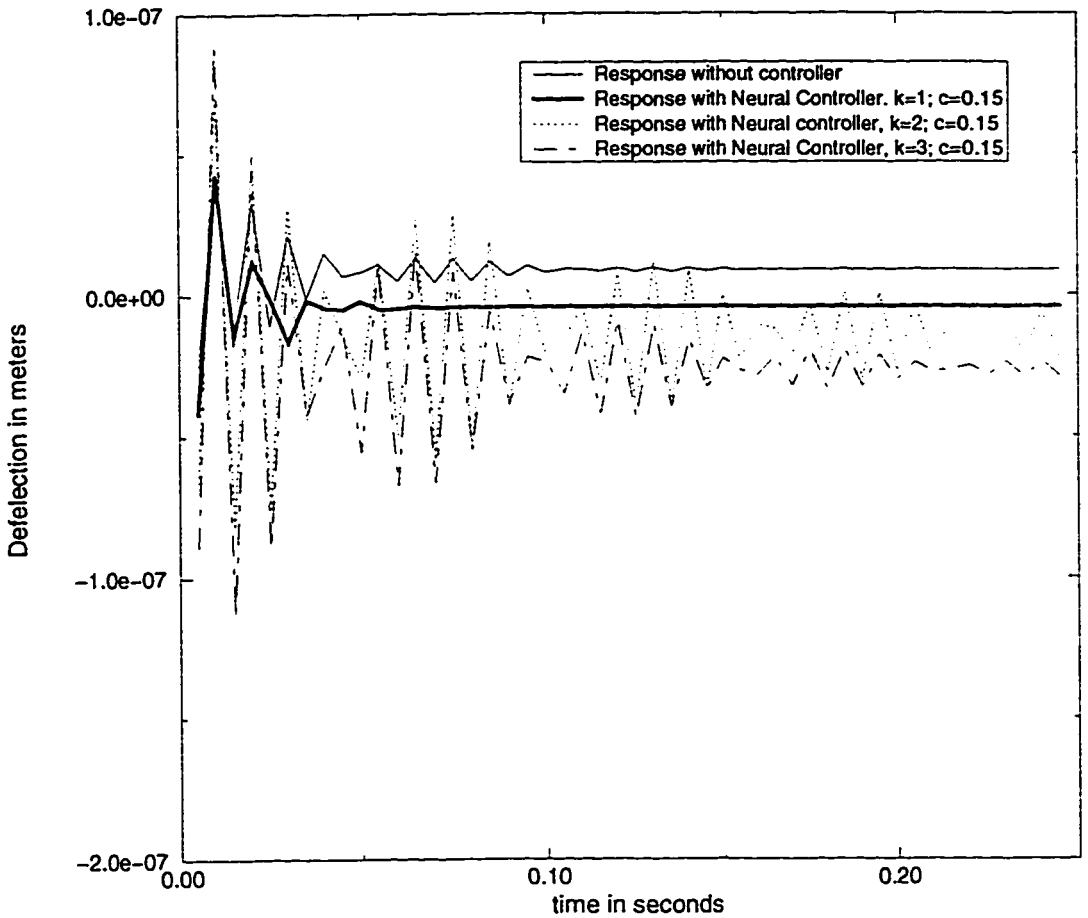

Figure 6.13: Response of the mid-span in horizontal direction at a shaft speed of 4500rpm when an external unit disturbance is applied at the mid-span in the vertical direction. Responses for both cases is presented with and without the controller. The responses are plotted by fixing c=1 and varying k

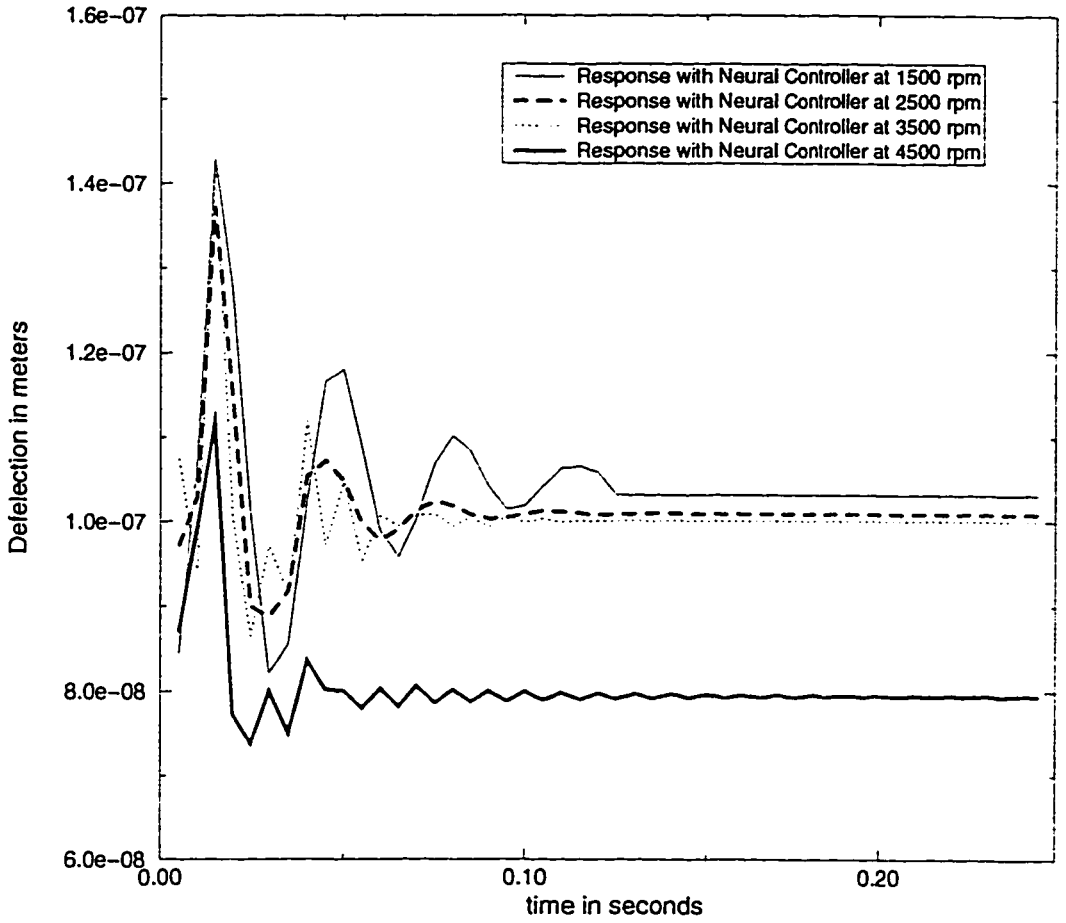

Figure 6.14: Response of the mid-span in vertical direction at different shaft speeds when an external unit disturbance is applied at the mid-span in the vertical direction. The responses are plotted by fixing c=0.15, and k=1

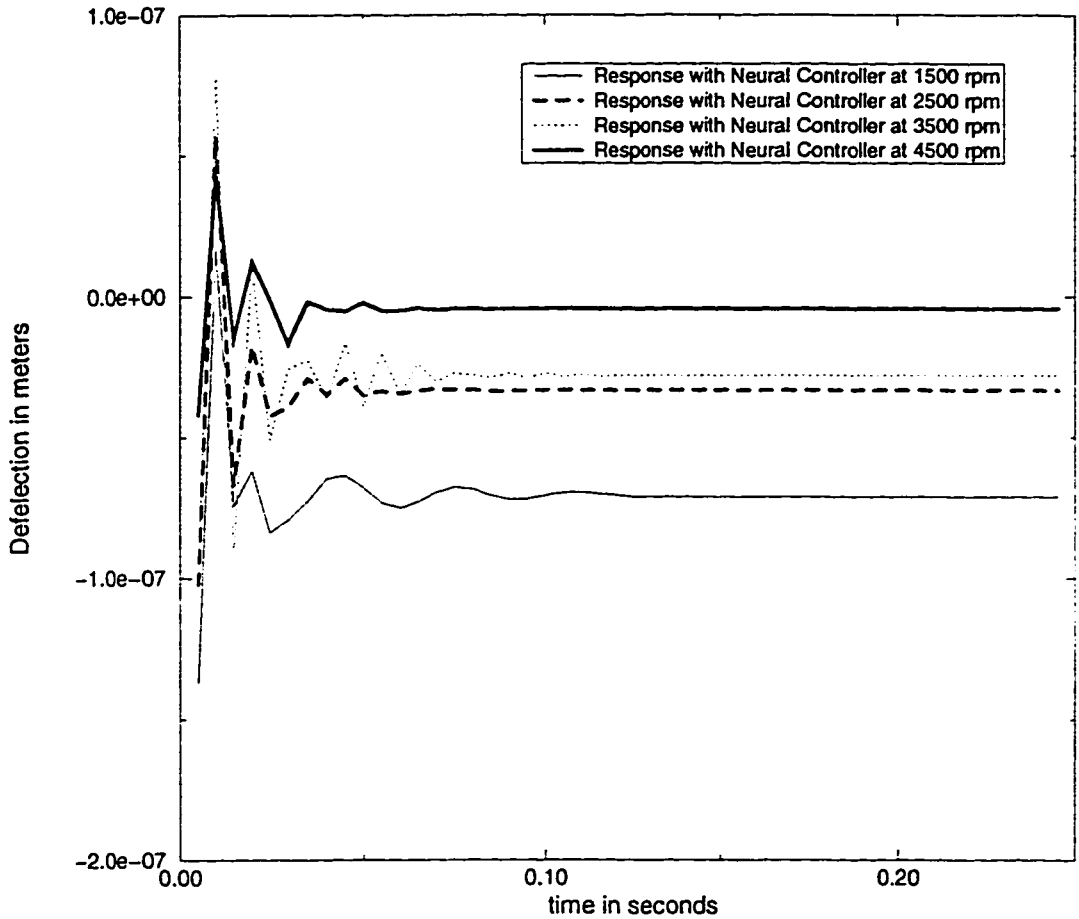

Figure 6.15: Response of the mid-span in horizontal direction at different shaft speeds when an external unit disturbance is applied at the mid-span in the vertical direction. The responses are plotted by fixing c=0.15, and k=1

Chapter 7

Conclusions and Recommendations

A model of rotor bearing systems is presented. Its mathematical modelling is done using the lumped approach. The effect of fluid film parameters for Stiffness and Damping on the rotor stability at different speeds is studied. An area of control system which offers much scope for the use of neural networks is adaptive control. Here the controller has to adapt itself with a change in the process paramters. A class of adaptive control algorithms is called self tuning. In this work we have concentrated upon method of utilizing neural network as implicit self tuning controller. The identification of the system is done using the principle of inverse dynamics. Backpropagation is used as a

learning algorithm. The code for the proposed algorithm is written in fortran and implemented on SUN 10 computers.

It can be deduced from the behavior of system that, it exhibits a stable response when subjected to a unit step disturbance when the cross coupling terms for the fluid film are considered to be zero, even higher speeds. When these cross coupling terms are included the system, it is observed that the system is stable at lower speeds and unstable at higher speeds. It can be concluded that it is not suitable from a practical point of view to neglect the cross coupling terms. When a neural network controller is applied to the system, the disturbance is damped out smoothly but slowly. The steady state deflection of the rotor can be varied by varying the parameter c, k and learning rate η. When a neural network controller is applied to the system (at 4500), it was seen that the steady state is achieved earlier in the horizontal deflections and the peaks are considerably reduced. The similar trend is observed for the responses in the vertical direction. However, the system shows oscillation before reaching the steady state. A versitle control stategy can be achieved by varying the learning rate η, the sensitivity factor c and the magnification factor k. A parametric study shows that for a specific value of k a value of c should be selected to make the controller effective with a suitable learning rate. A combination of k=1, c=0.15 and $\eta = 0.01$ was found to be effective. With higher values of k and c using the same value of $\eta = 0.01$ the controller

was ineffective and the disturbance of the system was not damped and there was a change in the steady state. Therefore with proper selection of learning rate, c and k, the disturbance can be effectively be controlled. A study was also carried out by varying the speed of the Rotor Bearing system. It was found that at lower rotor speeds the values of c and k should be higher and at higher rotor speeds the values of η, c and k should be relatively lower.

7.1 Recommendations

(a) Schemes to use neural networks as controllers based on off-line design strategies can also be investigated.

(b) With this same algorithm, the controller for nonlinear plants can also be developed.

(c) The study of the system subjected to different disturbance inputs can be carried out.

(d) Nonlinearity can be introduced in the stiffness of the shaft.

(e) The same system with a flexible foundation can be studied.

.1 Appendix

.1.1 Coefficient Matrices

The differential equations of the rotor bearing system can be represented in a vector-matrix notation, as follows:

$$\dot{X} = AX + Bu$$

or

$$\begin{bmatrix} \dot{x}_v \\ \dot{x}_h \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{Bmatrix} x_v \\ x_h \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{Bmatrix} u_v \\ u_h \end{Bmatrix}$$

Where the state vector motion x_v and horizontal motion x_h, the vertical input u_v and horizontal input u_h, the matrices are defined as follows for the model.

$$A_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{xx}+K_f}{m_b} & -\frac{C_{xx}}{m_b} & -\frac{K_f}{m_b} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{K_f}{m_f} & 0 & -\frac{K_f}{m_f} & 0 \end{bmatrix} ; A_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{-K_{xy}}{m_b} & \frac{-C_{xy}}{m_b} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{K_{yx}}{m_b} & -\frac{C_{yx}}{m_b} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; A_{22} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{yy}+K_f}{m_b} & -\frac{C_{yy}}{m_b} & -\frac{K_f}{m_b} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{K_f}{m_f} & 0 & -\frac{K_f}{m_f} & 0 \end{bmatrix}$$

$$B_{11} = B_{22} \begin{bmatrix} 0 & 0 \\ \frac{1}{m_b} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_f} \end{bmatrix}; B_{12} = B_{21} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

.2 Appendix

.2.1 Subroutines and Structure of the program

The different subroutines used in the program are **NET**, **DE**, **WMAT**, **F_NET**. **SUBROUTINE NET(TS,IT,M,N,NS,X,W,XS,U)**. X and W are technically the inputs to the subroutine, while u is the output vector. The inputs are the past y's and u's. The output from the plant is first normalized and then given as input to the **NET**. This happens at each time step. Therefore the input vector for any time t, will be of size 20 elements. For example we have $X = \{y(t), y(t-1), u(t-1)\}$, and for each time step we have the eight elements in the y vector and four elements in the u vector. The purpose of

this subroutine is to do the forward calculations of equations 4.5 to 4.8 and 4.3, the same subroutine is called for both the controller and the inverse neural model. TS is the current time step, IT is the maximum number of time steps.

SUBROUTINE WMAT(ISEED,N,NS,W). The inputs to this subroutine are N and ns. The output to this subroutine is the W matrix. In this case the N was taken as 22 and the ns is 4, therefore the size of W matrix is $(26, 26)$. This subroutine is called just once at the beginning, at this time the weights are random.

DE. The main arguments to this subroutine are the y's,u's, and time. The input to this subroutine is the control force u and the outputs are the states of the system i.e., y's. This subroutine solves the differential equations for the rotor-bearing system given in earlier chapters. **DE.FOR** is taken from " THE INITIAL VALUE PROBLEM BY L. F. SHAMPINE AND M. K. GORDON.

SUBROUTINE FNET(TS, IT, M, N, NS, F_NETI, F_XS, F_E, W, XS, F_W). The main arguments of this subroutine are the F_E, W, F_W. The inputs to this subroutine are the F_E which is calculated from the equation 4.9, and the Weight matrix W. The array F_W is the only output of this subroutine, upon which the weights are updated as in equation 4.17. The

F_NETI, F_XS are the backpropagation calculations.

Usually the number of passes is set to a high number, in our case it was set to 1000 iterations or an error criteria is set to some small value. First all the patterns are processed and the weights are updated after each time step, then this procedure is repeated for 1000 iterations till the error is settled.

Bibliography

[1] Gunter, E. J., Barrett, L. E., and Allaire, P. E. Stabilization of Turbomachinery with Squeeze film dampers-Theory and applications. *Institution of Mechanical Engineers Conference on Vibration in Rotating Machinery, Cambridge*, 1976.

[2] Cunningham, R. E. Steady-state unbalance response of a three disk flexible rotor on flexible, damped support. *Journal of Mechanical Design, Transaction of ASME*, 100, 1978.

[3] Nikolajsen, J. L. and Holmes, R. Investigation of Squeeze-film Isolators for the Vibration Control of a Flexible rotor. *Journal of Mechanical Engineering Science*, 21(4):247–252, 1979.

[4] Mu, C., Darling, J., and Burrows, C. R. An appraisal of a proposed active squeeze-film damper. *Journal of Tribology. ASME Trans.* 90(68):1–5, 1990.

[5] Gondhalekar, V. M., Nikolajsen, J., and Jayawant, B. V.,. Electro-

magnetic control of flexible trnsmission shaft vibration. *Proc. IEEE*, 126(10):1008–1010, 1979.

[6] Stanway, R., burrows, C. R. Active vibration control of a flexible Rotor on flexibly-monted Journal Bearings. *Journal of Dynamics systems, Measurements and Control*, 103, 1981.

[7] Farooq Anjum, M.M. *Self Tuning Control of Nonlinear Systems: A Neural net Based Approach.* Van Nostrand Reinhold,, 1988.

[8] Burrows, C. R., Kucuk, N. C., Sahinkaya, M. N and Stanway, R. Linearized squeeze film dynamics: Model structure and interpretation of experimentally derived parameters. *Journal of Mechanical Engineering Science., Part C:*, 204, 1990.

[9] Goodwin, M.J. *Dynamics of Rotor-Bearing Systems.* Unwin Hyman Ltd., London., 1989.

[10] Reddi, M. M. and Trumpler, P. R. Stability of the High-speed Journal bearing under steady load, Part 1: The incompressible film. *Journal of Engineering for Industry. Trans of ASME.*, 1962.

[11] Rao, J. S. *Rotor-Dynamics.* John Willey and Sons., 1983.

[12] Schweitzer, G., and Lange, R. Characteristics for a Magnetic rotor-bearing for Active vibration control. *Institution of Mechanical Engineers*

Conference on Vibration in Rotating Machinery, Cambridge, Paper No. 239/76, 1976.

[13] Narender K. S. Intelligent control. *IEEE Control Systems magazine*, 1991.

[14] Fukuda T. Watanabe K. and Tzafestas S.G. An adaptive control for carma systems using linear neural networks. *International Journal of Control*, 56(2):483–497, 1992.

[15] Narendra K.S. and Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Transaction on Neural Networks.*, 1(1):4–27, 1990.

[16] Chen F.C. Backpropagation neural networks for nonlinear self tuning adaptive control. *IEEE Control systems magazine*, 10(2):44–48, 1990.

[17] Narender K. S. and Levin A. U. Control of nonlinear dynamical systems using neural networks; Controllability and stabilization. *IEEE Transactions on Neural Networks*, 4(2):192–206, 1993.

[18] Kalman R. E. Design of self optimization control system. *Transaction of ASME*, 1959.

[19] Astrom K. J. and Wittenmark B. On Self tuning regulators. *Academic Press, Lund Institute*, 1970.

[20] Astrom and Wittenmark B. Practical issues in the implementation of self tuning control. *Automatica*, 20(5):595–605, 1984.

[21] Hunt K. J. et al. Neural networks for control systems - a survey. *Automatica*, 28(6):1083–1112, 1992.

[22] Yabuta T. and Yamada T. Neural network controller characteristics with regard to adaptive control. *IEEE Transactions on systems, man and cybernetics.*, 22(1):170–176, 1992.

[23] Chen S. et al. Parallel recurive prediction error algorithm for training layered neural networks. *International Journal of Control*, 51(6):1215–1228, 1990.

[24] Paul Werbos. Backpropagation Through Time: What it does and how to do it. *Proceding IEEE*, 78(10):1550–1560, Oct 1990.

[25] Narender K. S. and Mukhopadhyay S. Intelligent control using neural networks. *IEEE Control Systems magazine*, 12(2):11–19, 1992.

[26] Shapiro, W., and Rumbarger, J. H. Bearing influence and representation in Rotor dynamic analysis. *Flexible Rotor-Bearing System Dynamics, ASME*, 1972.

[27] Adams, M. L., and Rashidi, M. On the use of rotor-bearing instability thresholds to accurately measure bearing rotor-dynamic properties. *Journal of Vibration, Acoustics, Stress and Reliability in Design*, 107, 1985.

[28] Morton, P. G. Measurement of Dynamic characteristics of a large sleeve bearing. *ASME Journal of Lubrication Technology*, 93, 1971.

[29] Emerson, W., and Walter, D. P. The estimation of parameters in a Linear Rotor-Bearing system model. *ASME Design conference, Chicago*, 1978.

[30] Burrows, C. R., and Stanway, R. Identification of Journal bearing characteristics. *ASME Journal of Dynamic systems, Measurements, and Control*. 99, 1977.

[31] Burrows, C. R., and Stanway, R. A coherent strategy for estimating linearized oil-film coefficients. *Proc. R. Soc. Lond.*, A-370, 1980.

[32] Strenlicht, B. Elastic and Damping properties of Cylindrical Journal bearings. *ASME Journal of Basic Engineering*, 81, 1959.

[33] Simon Haykin. *Neural Networks - A Comprehensive Foundation*. Macmillan Publishing Company, 1994.

[34] Wasserman P. *Neural Computing*. Van Nostrand Reinhold,, 1988.

[35] Touretzky D. S. and Pomerleau D. A. What's hidden in the hidden layers. *BYTE*. 1989.

[36] Werbos Paul. Maximizing long-term gas industry profits in two minutes in Lotus using neural network methods. *IEEE trans. sys., Man.,Cyber.*, March-April 1989.

[37] Cybenko G. Approximation by superposition of sigmoidall function. *Mathematics of Control signals and systems*, 2(4):303–314, 1989.

[38] Antonio Moran and Masao NAgai. Optimal Active Control of Nonlinear Vehicle Suspensions Using Neural Networks. *JSME International Journal*, 37(4):707–718, 1994.

[39] Psaltis et al. A multilayered neural network controller. *IEEE Control Systems magazine*, 1988.

[40] Kamal A.F. Moustafa and Asfar. Identification of Journal Bearing Modal Parameters. *International Journal of Analytical and Experimental Analysis*, 5, 1990.

[41] Burrows, C.R. and Sahinkaya, M.N. Frequency domain estimation of Linearized Oil-Codfficients. *Journal of Lubrication Technology*, 104, 1982.

[42] Vance, J. M. *Rotor-dynamics of Turbomachimery*. John Willey and Sons., 1988.

Vita

Mohsin Siddiqui

Born in Hyderabad, India in September 1969.

Finished high school from All Saints High School, Hyderabad, India in 1985.

Graduated with a degree in Bachelor of Engineering. (Mechanical Engineering) from Osmania University, Hyderabad, India in 1991.