Intelligent Instrumentation, Control and Monitoring of Precision Motion Systems

TANG KOK ZUEA

NATIONAL UNIVERSITY OF SINGAPORE

2004

Intelligent Instrumentation, Control and Monitoring of Precision Motion Systems

TANG KOK ZUEA

(*M.Eng., B.Eng.(Hons), NUS*)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2004

Acknowledgments

I would like to express my appreciation to those who have guided me during my postgraduate course in National University of Singapore. Firstly, I wish to express my utmost gratitude to my supervisors, Professor Lee Tong Heng and Associate Professor Tan Kok Kiong for their unfailing guidance throughout the course of my candidature. I have indeed benefited tremendously from the many discussions I have with them.

I was also privileged by the close and warm association with my colleagues in the Mechatronics and Automation Laboratory. I would like to thank my colleagues, namely Dr Huang Su Nan, Chee Siong, Raihana, Ming Yang, Han Leong, Jim, Chek Sing and Guan Feng for their invaluable comments and advice. All this while, they have made my postgraduate course in NUS become an unforgettable and enjoyable experience.

I would also like to thank my family for their love and support. Specially, I wish to express my deep appreciation to Shona for her love, support and understanding. Finally, I would like to thank God for everything!

Contents

List of Figures

List of Tables

List of Abbreviations

A/D	*Analog − to − Digital*
ADC	*Analog − to − Digital Converter*
API	*Application Programming Interface*
CNC	*Computer Numerical Control*
DFT	*Discrete Fourier Transform*
DAQ	*Data Acquisition*
DARAM	*Dual Access Ramdom Access Memory*
DC	*Direct Current*
DSP	*Digital Signal Processing*
DSTP	*Datasocket Transfer Protocol*
et al.	*et alii*
etc.	*et cetera*
FFT	*Fast Fourier Transform*
FTP	*File Transfer Protocol*
HTTP	*HyperText Transfer Protocol*
I/O	*Input/Output*
IC	*Integrated Circuits*
KB	*Knowledge − Based*
LQR	*Linear Quadratic Regulator*
MEMS	*Micro − Electro − Mechanical Systems*
MIPS	*Mega Instructions Per Second*
MOSFET	*Metal Oxide Semiconductor Field − Effect Transistor*

NN	*Neural Network*
OPC	*OLE for Process Control*
PC	*Personal Computer*
PDA	*Personal Digital Assistant*
PMLM	*Permanent Magnet Linear Motor*
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
URL	*Universal Resource Locator*

Summary

High speed and high accuracy motion systems are essential elements in advanced manufacturing systems. Demands on higher productivity and product quality call for development of high performance positioning devices and accompanying robust control algorithms. The increasing complexity of precision motion systems coupled with the increasing demands in closed loop performance specifications necessitates the use of more complex and sophisticated controllers. It is desirable that these controllers are able to perform well under significant uncertainties in its operating environment, be able to compensate for system failures (within limits) without external interventions, and be sufficiently adaptable to deal with unexpected situations, new control tasks or changes in control objectives. Much benefits could be gained by combining intelligent control with the well-established tools in control theory. In this perspective, contributions in the areas of precision motion instrumentation, control and diagnostics are proposed in this thesis, with the aim of improving the performance of precision motion systems.

Firstly, an intelligent instrumentation methodology is developed for the purpose of adaptive online correction and interpolation of quadrature encoder signals, suitable for application to precision motion systems. Methods reported in the literature for the correction and interpolation of the encoder signals generally require explicit high precision analog-to-digital-converters (ADCs) in the control system, and a high speed digital signal processor (DSP) to compute the electrical angle to the required resolution. Therefore, they are not applicable to the typical controller with only a digital incremental encoder interface. Furthermore, it is cumbersome to integrate sinusoid correction with interpolation since the correction parameters must be calibrated offline. In this work, the radial basis functions neural network (RBFNN) is employed to carry out concurrently the correction and interpolation of encoder signals in real-time. Although the table look-up method may give similar results as the proposed approach, there is much savings in memory storage requirements using the proposed approach.

The following part of the thesis presents a intelligent control methodology for precision motion systems, based on a mixed PID/adaptive algorithm. A second-order linear dominant model is considered with an unmodeled part of dynamics that is possibly nonlinear and time-varying. The PID part of the controller is designed to stabilize the dominant model. The adaptive part of the controller is used to compensate for the deviation of the system characteristics from the dominant linear model to achieve performance enhancement. The advantage of the proposed controller is that it can cope with strong nonlinearities in the system while still using the PID control structure which is well-known to many control engineers. The proposed robust control scheme guarantees the boundedness of the system states and parameter estimation.

Two approaches to monitor and suppress mechanical vibrations in precision motion systems are presented next. The first approach utilizes an adaptive notch filter to identify the resonant frequencies and suppress any signal transmission into the system at these frequencies. The second approach uses a real-time analyzer to detect excessive vibration based on which appropriate actions can be taken, say to provide a warning or corrective action. This second approach can be implemented independently of the control system and as such can be applied to existing equipment without modification of the normal mode of operation. To expand the scope of precision motion control, the Internet is utilized for remote vibration monitoring of precision motion systems.

Simulation and experimental results are provided to highlight the effectiveness of

the proposed approaches.

Chapter 1

Introduction

Precision motion systems play an important role in many industries. Some of these industries include the microelectronics manufacturing, aerospace, biomedical and the storage media. The role of precision motion systems in the wide range of industries imposes challenging demands on precision motion systems as a result of the products' shrinking sizes, tighter specifications and very large production volumes of the final products. Furthermore, multi-functional products and product downsizing, which provides space-saving features, are expected in the modern world. The tough demands on the final products translates to different high precision and high speed requirements of precision motion systems in all the fabrication, inspection, assembly, and handling processes.

1.1 Evolution of Precision Motion Systems

The historical roots of precision engineering are arguably in the field of horology, the development of chronometers, watches and optics, e.g., the manufacture of mirrors and lenses for telescopes and microscopes. Major contributions were made to the development of high precision machine tools and instruments in the late 1800s and early 1900s by ruling engines. Scales, reticules and spectrographic diffraction gratings were manufactured with increasing precision and resolution. Today, ultra-precision machine tools under computer control can position the tool relative to the workpiece to a resolution and positioning accuracy in the order better than sub-micrometers. It must be noted that achievable 'machining' accuracy includes the use of not only machine tools and abrasive techniques, but also energy beam processes such as ion beam and electron beam machining, as well as scanning probe systems for surface measurement and pick-and-place type of manipulation.

The microprocessor began to proliferate into many motion applications in the late 1970s. The main technology force for all precision motors is the continued evolution of both logic and power electronics. New power electronic devices joined microprocessors and other logic integrated chips (ICs) in providing more efficient and higher power devices as represented by the bipolar transistor in the early 1970s and the metal oxide semiconductor field-effect transistor (MOSFET) at the end of the 1970s. Packaging these devices into a step or servomotor drive moved in various directions. The personal computer (PC) board with integrated heat sinks for the power devices was used extensively. On-board logic circuitry became available for servodrives or amplifiers to control motor commutation, current, and velocity control. The servo boards were analog with output voltage signals from the generators as a function of speed providing the precision velocity signal measurements for use in the servosystem.

One main application area for precision motion systems is in the precision manufacturing industry. One such industry is the microelectronics manufacturing industry Manufacturing tolerances which are better than one part in 10^5 are now achievable.

Much credit must be given to the advancements in terms of research and development efforts dedicated to precision motion systems. Ultra-precision manufacture is poised to progress further and to enter the nanometer scale regime, i.e., nanotechnology. Increasing packing density on integrated circuits and sustained breakthrough in minimum feature dimensions on semiconductor set the pace in the electronics industry. Emerging technologies, such as micro-electro-mechanical Systems (MEMS) and computer numerical control (CNC) systems, expand further the scope of miniaturization and integration of electrical and mechanical components. However, design rules for precision motion systems with millimeter or sub-millimeter resolution do not apply for the micron and sub-micron range. Resolution in the sub-micron or lower realm cannot always be increased by simple means such as reducing the pitch of a lead-screw or increasing the gear ratio of a motor/gearhead unit. Stiction/friction, play, backlash, tilt, windup and temperature effects and many other disturbances will also limit accuracy and resolution. Thus, sub-micron positioning systems require a great deal of attention in design, manufacturing and selection of materials.

In view of the above motivation, the many control challenges ahead for precision motion systems are to achieve higher speed, higher precision, and yet maintain robust performance, in the face of several performance limitations such as system nonlinearities, system uncertainties and system dynamic constraints. With increased speed in manufacturing, a higher production rate can be achieved. On the other hand, products with better quality can be manufactured with increased precision. Maintaining robust performance assures consistent product quality. But, it is difficult to maintain, let alone increase precision when speed is increased.

In these recent years, several achievements in precision motion control are made possible by key technological advances taking place in the industry. Today's electronic control is becoming ever more proficient as new microprocessors, DSPs, and similar electronic devices supply the control platform with tremendous computing and process timing power. More powerful processors are allows more advanced control algorithm to be used. Advances in actuators, such as direct drive motors, linear motors, and brushless motors are reducing traditional difficulties such as backlash, friction, and parasitic system dynamics. The linear motor is hailed as the motion device of the next generation because of its superior performance compared to conventional linear positioning devices such as ball-screw drives. The increasing widespread industrial applications of linear motor in various semiconductor processes, precision metrology and miniature system assembly are self-evident testimonies of the effectiveness of linear motor in addressing the high requirements associated with these application areas. Advances is power semiconductors are allowing these new actuators to be driven in a more power-efficient and cost-effective fashion. Advances in bearing systems, particularly for low load situations such as fluid and magnetic bearings, are also reducing the effects of friction and stiction. Promising new materials such as composites and ceramics offer potential benefits in mechanical properties such as lowering mass, improving damping, and reduction in thermal effects. Finally, advances in sensors, due primarily to new techniques in optics, electronics, and signal processing, are allowing designers to get better feedback measurements.

Industry has favored classical controllers such as proportional-integrator-derivative (PID) controller due to their structural simplicity and well-known characteristics. As performance requirements become more stringent, conventional controllers often fail because of system uncertainties, the presence of high-order dynamics and nonlinearities such as friction (i.e., Coulomb, viscous and stiction) and actuator saturation.

1.2 Intelligent Precision Motion Systems

The increasing complexity of precision motion systems coupled with the increasing demands in closed loop performance specifications necessitates the use of more complex and sophisticated controllers. Yet as precision motion systems become more complex, uncertainty in modeling increases. The challenges that arise in the control of increasing complex precision motion systems can be broadly classified under three categories:

(1) **Computational Complexity.**([1]) With the increasing scope of precision motion control systems and the resulting rush toward more sophisticated computational architectures, more computing power at a higher speed is greatly desired in order to implement the complex control algorithms. The development of higher power DSPs and processors need to keep up with the pace of industry's demands.

(2) **Nonlinearity.** ([2]-[3]) Even in a purely deterministic context, the presence of nonlinearities in a dynamical system makes the control problem complex. Current research efforts in nonlinear control theory focus on geometric methods and attempt to extend well-known results in linear control theory to the nonlinear domain. Despite the great interest in this area, many fundamental theoretical issues related to nonlinear control are currently not yet well understood. What is more relevant for the purposes here is that many of the theoretical results available cannot be directly used for practical control in precision motion systems. Besides these, the model structure of complex precision motion systems, being nonlinear stochastic and time varying,

may not be amenable to simple linear time-invariant modeling.

(3) **Environmental Uncertainty.** ([4]-[5]) Practical systems encountered in the industries raise questions related to control when some part of the information essential for any mathematical analysis is unknown. In many situations, precision motion systems are subject to large unpredictable environmental disturbances.

The design of controllers, which perform satisfactorily in high-dimensional decision spaces in the presence of nonlinearity under various conditions of uncertainty, is a formidable problem. Pattern recognition, learning, adaptive control, robust control, and knowledge-based systems are applicable in relatively disjoint contexts. Although great advances have been made in each of these areas, the settings in which each can be applied are too limited to connote intelligence. Hence, in a recent proposal, Narendra and Koditschek [5] adopted the perspective that when such advanced capabilities (which are applicable to relatively narrow domains) are joined together in special ways, they can result in complex systems that respond appropriately to very challenging environments and even in situations for which they have not been explicitly designed. It is in this prespective that contributions are made in this thesis to combine intelligent control with the well-established tools in control theory (i.e., linear and nonlinear control theory, optimal control and game theory, and stochastic, adaptive, and learning control theories).

It is desirable to design new intelligent controllers that perform well under significant uncertainties in the system and in the environment in which it operates, be able to compensate for system failures (within limits) without external interventions, and be sufficiently adaptable to deal with unexpected situations, new control tasks or changes in control objectives. Intelligent control achieves automation via the emulation of biological intelligence. It either seeks to replace a human who performs a control task (e.g., a chemical process operator), or it borrows ideas from how biological systems solve problems and applies them to the solution of control problems (e.g., the use of neural networks for control).

There are many instances whereby the combination of intelligent control with the well-established tools in control theory will yield good results. For example, in a dynamical system whose characteristics are linear and are known exactly, the control input can be determined by the application of well-developed control techniques. Even at this level, when the characteristics are nonlinear, prescriptive methods for generating the control input are not readily available. When the dynamical system is linear but parametric uncertainty exists, adaptive control is a natural choice. Because the parameters vary with time, the controller parameters tune themselves but have no long-term memory. Pattern recognition together with adaptive control can be used for this purpose.

By common practice, many practical precision motion systems are first regulated or manually tuned by human operators before automatic controllers are installed. The plant operator has few apparent problems with plant nonlinearities or adjusting to slow parametric changes in the plant or with satisfying a set of complex static and dynamic process constraints. The human operator is able to respond to complex sets of observations and constraints, and to satisfy multiple subjective-based performance criteria. However, the control actions of the human are difficult to analyze as they are variable and subjective, prone to error, inconsistent and unreliable. In the case of safety critical and hazardous situations, such human actions may be potentially dangerous. It is desirable to incorporate the positive intelligent and creative attributes of human controllers, whilst avoiding the elements of inconsistency, unreliability, temporal instability, fatigue and other negative attributes associated with the human conditions.

In view of the above observations, control schemes that use different combinations of the well-developed control theory and artificial intelligence are developed in this thesis to develop precision motion control and diagnostic methodologies to achieve high performance (in terms of tracking accuracy, robustness, and disturbance and noise rejection). Particularly, the contributions are in the areas of intelligent instrumentation, control and monitoring. These areas will be highlighted below.

1.2.1 Instrumentation

To realize precision motion control, a precise measurement of the signals generated by the position encoders is essential, since it will determine the final achievable resolution, and hence accuracy of the motion control application. To increase the precision of the overall system, one approach is to increase the resolution of the encoders. However, this measurement precision is limited by the manufacturing technology of the encoders. To date, the scale grating on linear optical encoders can be manufactured to less than four micrometers in pitch, but clearly, further reduction in pitch will be greatly constrained by physical considerations. This implies an optical resolution of one micrometer can be currently achievable. Interpolation using soft techniques provides an interesting possibility to further improve on the encoder resolution, by processing the analog encoder signals online to derive the small intermediate positions.

The interpolation approaches in the literacture generally require explicit high precision ADCs in the control system, and a high speed DSP to compute the electrical angle to the required resolution. Therefore, they are inapplicable to the typical servo controller with only a digital incremental encoder interface. Furthermore, it is cumbersome to integrate sinusoid correction with interpolation since the correction parameters must be calibrated offline. As a result, most servo controllers which are able to offer interpolation have mainly inputs which are assumed to be perfect quadrature sinusoids. Hence, specifications relating to resolution may be achievable, but the accompanying accuracy cannot be guaranteed. Current efforts for sinusoid correction also does not consider error in the form of waveform distortion, i.e., the actual signal may be periodic, but not perfectly sinusoidal. These errors can become significant bottlenecks when sub-micron resolution and accuracy is required. In view of the shortcomings of the current approaches, an intelligent instrumentation methodology that will correct and interpolate the encoder signals concurrently is desired.

1.2.2 Control

As mentioned earlier, one of the many control challenges ahead for precision motion systems is to achieve high speed, high precision, and yet maintain robust performance, in the face of several performance limitations such as system nonlinearities, system uncertainties and system dynamic constraints. For a long time, classical controllers such as PID are favoured by the industry due to their structural simplicity and well-known characteristics. As performance requirements become more stringent, conventional controllers often fail because of system uncertainties, the presence of high-order dynamics and nonlinearities such as friction and actuator saturation. Furthermore, the limitations of PID control rapidly become evident when applied to more complicated systems such as those with a time-delay, poorly damped, nonlinear and time-varying dynamics.

In this perspective, the structural simplicity and well-known characteristics of classical controllers such as PID can be combined with artifcial intelligence to achieve robust control of precision motion systems. This intelligent controller should be able to stabilize the nominal system while taking into consideration the system nonlinearities.

1.2.3 Monitoring

Mechanical vibration in machines and equipment can occur due to many factors, such as unbalanced inertia, bearing failures in rotating systems such as turbines, motors, generators, pumps, drives and turbofans, poor kinematic design resulting in a non-rigid and non-isolating support structure, component failure and/or operation outside prescribed load ratings. The machine vibration signal can be typically characterized as a narrow-band interference signal anywhere in the range from 1 Hz to 500 kHz. When the machine is used to perform highly precise positioning functions, undue vibrations can lead to poor repeatibility properties, impeding any effort for systematic error compensation. This results directly in a loss of achievable precision and accuracy. It would be desirable if the vibration suppression capability can be incorporated into the control structure. Undesirable vibrations can then be filtered out of the system before they can cause any other complications.

As it is essential to monitor and suppress vibration in precision motion systems, it would be desirable to have an external diagnostic tool that performs vibration monitoring. This vibration monitoring and control device will be very useful to prevent equipment damage from the severe shaking that occurs when a machine malfunctions or vibrates at a resonant frequency.

1.3 Remote Monitoring and Control

To further expand the scope of precision motion control, the power of the Internet could be harnessed to perform remote monitoring and control of precision motion systems. In the current economy, many manufacturing processes are widely distributed geographically, due to economy-related factors in manufacturing and distribution. The layout of an entire plant can now be rather extensive, spreading across continents in certain cases. Therefore, it has become an important challenge to be able to optimize any synergy opportunities in the operations of these distributed systems. In many cases, the same set of processes to manufacture the same product (or to monitor the same process) can be cloned over different plants. This requires close coordination and synchronization of the distributed operations, as well as an efficient remote monitoring and control facility in place. Thus, an extensive and 'borderless' approach towards the effective monitoring of the distributed points is crucial to enhance overall efficiency and operational costs.

Harnessing the power of the Internet for the networking of plants will make it possible to collect more information from the shopfloor and to disseminate it far and wide through every level of the company structure. The fast expanding infrastructure of the Internet, in terms of its high volume of traffic and the large number of network nodes around the globe, makes it highly suited for the networking of plants at different locations. Indeed, the ultimate aim for remote monitoring and control capability for systems is to ensure static and mobile workers maximize their productivity for the business. This added capability may not be applicable for all types of systems and situations. It is to be stressed that the main capabilibity of the proposed approach is monitoring. Due to the added feature of remote monitoring, security and reliability are two main considerations here. Emerson Process Managment [6] and GE Fanuc Automation [7] are two active players in the remote monitoring and control business.

1.4 Contributions

The aim of this thesis is to design robust precision motion systems that perform satisfactorily in high-dimensional decision spaces in the presence of nonlinearities under various conditions of uncertainty. Experimental results are provided in this thesis to support the various proposed approaches. The contributions made in this thesis can be summarized as follows:

Intelligent Instrumentation: Adaptive Online Correction and Interpolation of Quadrature Encoder Signals Using Radial Basis Functions

Precision motion control and positioning is a core requirement behind many robotics and drive control applications. To enable it, a precise measurement of the signals generated by the position encoders is essential, since it will determine the final achievable resolution, and hence accuracy of the motion control application. However, this measurement precision is limited by the manufacturing technology of the encoders. Interpolation using soft techniques provides an interesting possibility to further improve on the encoder resolution, by processing the analog encoder signals online to derive the small intermediate positions. The methods reported in the literature generally require explicit high precision ADCs in the control system, and a high speed DSP to compute the electrical angle to the required resolution. Therefore, they are inapplicable to the typical servo controller with only a digital incremental encoder interface. Furthermore, it is cumbersome to integrate sinusoid correction with interpolation since the correction parameters must be calibrated offline. In this work, the radial-basis function neural network (RBFNN) is employed to carry out concurrently the correction and interpolation of encoder signals in real-time. A two-stage RBFNN is used in the implementation of the proposed approach. This approach can be readily applied to most standard servo controllers.

Intelligent Control: Combined PID and Adaptive Nonlinear Control for Precision Motion Systems

The PID controller has remained, by far, as the most commonly used controller in practically all industrial control applications. The reason is that it has a simple structure which is easy to be understood by the engineers. Over the years, many techniques have been suggested for tuning of the PID parameters. Among them, the model-based tuning methods appear to be very encouraging. However, the limitations of PID control rapidly become evident when applied to more complicated systems such as those with a time-delay, poorly damped, nonlinear and time-varying dynamics. In this work, an intelligent controller comprising of a PID and an adaptive controller is presented for a class of nonlinear servo mechanical system. In the proposed approach, a second-order model with an unknown nonlinear term that is nonlinear and time-varying is used as the dominant model of a class of nonlinear systems. PID control is applied to stabilize the nominal system based on this dominant model. The system nonlinearity is compensated using an adaptive scheme employing the RBFNN. The stability and tracking performance associated with the scheme is regional in system states.

Intelligent Monitoring: Monitoring and Suppression of Vibration in Precision Motion Systems

Mechanical vibration in machines and equipment can occur due to many factors. Equipment may be damaged as a result of the severe shaking that occurs when a machine malfunctions or vibrates at a resonant frequency. Moreover, when the machine is used to perform highly precise positioning functions, undue vibrations can lead to poor repeatibility properties. This piece of work addresses two approaches to deal with mechanical vibrations. The first approach utilizes an adaptive notch filter (narrow-bandstop filter) to identify the resonant frequencies and suppress any signal transmission into the system at these frequencies. The second approach uses a real-time analyzer to detect excessive vibration based on which appropriate actions can be taken, say to provide a warning or corrective action. This second approach can be implemented independently of the control system and as such can be applied to existing equipment without modification of the normal mode of operation. To expand the scope of vibration monitoring, an exemplary application concerning the remote vibration monitoring and control of machines distributed over different locations, via the Internet, is presented to illustrate the principles of the proposed configuration.

1.5 Outline of Thesis

The contributions of this thesis is organized in the following manner.

Chapter 2 considers the development of an adaptive online approach for the correction and interpolation of quadrature encoder signals, suitable for application to precision motion systems. It is based on the use of a two-stage double-layered RBFNN. The principles of the proposed interpolation approaches are then explained. The learning and updating procedures of the two stages of the RBFNN are also described. Several considerations that are pertinent to the interpolation problem are subsequently discussed in the chapter.

In Chapter 3, a robust control method for precision motion systems, based on a mixed PID/adaptive algorithm is covered. A second-order linear dominant model is considered with an unmodeled part of dynamics that is possibly nonlinear and time-varying. The different components of the composite controller are described in detail here. The derivations for the stability of the proposed controller and the boundedness of the system states and parameter values are then presented.

Two approaches to reduce the damage caused by the mechanical vibrations in precision motion systems are presented in Chapter 4. The design of an adaptive notch filter is first discussed. Following this, the hardware and software aspects of a real-time analyzer are described in detail. The working principle of the real-time analyzer, based on a fuzzy fusion technique, is then explained to illustrate how the analyzer could be used to continuously monitor the machine vibrations and suppress undesirable vibrations. To extend the capability of this vibration monitoring application, the hardware and software aspects of a remote vibration monitoring and control application are discussed.

Throughout the thesis, simulations are provided for all algorithms proposed to demonstrate their usefulness. Real-time experimental results are then presented to assert their practical applicability. Finally in Chapter 5, directions of future work, and general conclusions are documented.

Chapter 2

Intelligent Instrumentation: Adaptive Online Correction and Interpolation of Quadrature Encoder Signals Using Radial Basis Functions

2.1 Introduction

Precision motion control and positioning is a core requirement behind many robotics and drive control applications. To enable it, a precise measurement of the signals generated by the position encoders is essential, since it will determine the final achievable resolution, and hence accuracy of the motion control application. To increase the precision of the overall system, one approach is to increase the resolution of the encoders. However, this measurement precision is limited by the manufacturing technology of the encoders. To date, the scale grating on linear optical encoders can be manufactured to less than four micrometers in pitch, but clearly, further reduction in pitch will be greatly constrained by physical considerations. This implies an optical resolution of one micrometer can be currently achievable. Interpolation using soft techniques provides an interesting possibility to further improve on the encoder resolution, by processing the analog encoder signals online to derive the small intermediate positions.

The error sources associated with positional information obtained this way can be classified under pitch and interpolation errors. Pitch errors arise mainly due to scale manufacturing tolerances and mounting distortion. They can be compensated via the same procedures which are carried out for general geometrical error compensation. Interpolation errors, on the other hand, are associated with the accuracy of subdivision within a pitch. Ideal signals from encoders are a pair of sinusoids with a quadrature phase difference between them. Interpolation operates on the relative difference in the amplitudes and phases of these paired sinusoids. Therefore, interpolation errors will occur if the pair-periodic signals deviate from the ideal waveforms on which the interpolation computations are based. These deviations must be corrected before interpolation.

One possible approach to compensate the mean value offset, phase and amplitude errors for two quadrature sinusoidal signals was introduced by Heydemann [8]. He used least squares fitting to compute these error components efficiently and made correction for two non-ideal sinusoidal signals. Using this method, Birch [9] was able to calculate optical fringe fractions to nanometric accuracy. By making use of the amplitude variation with angle, Birch divided one period of sinusoidal signal into N equiangular segments to increase the effective electrical angle resolution. A micro step controller [10] and encoder code compensation technology [11] have been developed based on this method. Relevant applications can also be found in [12] and [13]. To increase the resolution of optical encoders, Cheung [14] used logic gates, comparators and digital filters to perform the sine/cosine interpolation. This method employed hardware, complemented with some software programming to achieve its results. An absolute high performance, self calibrating optical rotary positioning system was designed by Madni *et al.* [15]. In this approach, a series of sine/cosine signals from the encoders are digitized by high precision ADCs and interpolation and calibration is performed by the DSP programs. ServoStar's [16] motor drives offers the ability to accept signals from various feedback devices and encoders. These encoders provide analog-encoded motor position data to the drive amplifier. The advantage of these analog signals is that they can be resolved to extremely small intervals, providing a lot of data about the motor shaft position while maintaining reasonable data transmission rates. The disadvantage is that analog signals are notably susceptible to noise pickup and require good wiring installation practices.

These interpolation approaches generally require explicit high precision ADCs in the control system, and a high speed DSP to compute the electrical angle to the required resolution. Therefore, they are inapplicable to the typical servo controller with only a digital incremental encoder interface. Furthermore, it is cumbersome to integrate sinusoid correction with interpolation since the correction parameters must be calibrated offline. As a result, most servo controllers which are able to offer interpolation have mainly inputs which are assumed to be perfect quadrature sinusoids. Hence, specifications relating to resolution may be achievable, but the accompanying accuracy cannot be guaranteed. Current efforts for sinusoid correction also does not consider error in the form of waveform distortion, i.e., the actual signal may be periodic, but not perfectly sinusoidal. The work in the literature considers 'ideal' sinusoids when performing interpolation. The errors in the waveforms have to be compensated carefully when sub-micron resolution and accuracy is required.

In this chapter, the radial basis function neural network (RBFNN) [17] is employed to carry out concurrently the correction and interpolation of encoder signals. This is the first application of neural network for this purpose. Neural networks (NNs) ([17] and [18]) are inherently useful for approximating nonlinear and complex functions. This is especially true for functions where only the input/output pairs are available and the explicit relationships are unknown. The RBFNN is one popular and commonly used configuration of neural network which uses a set of basis functions in the hidden units. The effective interpolation of the available sinusoidal signals can be seen as the generalization process for the available data. One main challenge, to be addressed in this chapter, is to realize an adequate fit with the simplest RBFNN structure possible by minimizing the redundancy present in the data mapping process. The square quadrature signals are derived from the sinusoidal signals, after they are corrected and interpolated. These square signals are then decoded by the control system's counter to obtain position measurements. The correction and interpolation process can only be performed using the sinusoidal quadrature signals, not the square ones. Thus the focus of the chapter is on the correction and interpolation of the sinusoidal encoder signals.

A two-stage RBFNN is used in the implementation of the proposed approach. The first RBFNN stage is concerned mainly with the correction of incoming non-ideal encoder signals, including the compensation of mean, phase offsets, amplitude deviations and waveform distortion. This RBFNN can be updated adaptively online to reflect any subsequent changes or drift in the characteristics of the encoder signals. The second RBFNN stage serves to derive high order sinusoids from the corrected signals from the first stage, based on which a series of high frequency binary pulses can be converted which, in turn, can be readily decoded by standard servo controllers. Factors affecting the limit and accuracy of interpolation will be discussed in the chapter. Simulation and experimental results are provided to highlight the principles and practical applicability of the proposed method. The main strength of the proposed approach, as compared to the other current approaches, is its adaptive nature to correct and interpolate the encoder signals. It is also simple to implement the interpolation and correction features of the proposed approach to existing control structures. There is no need for additional hardware. Although the look-up table method [26] may give similar results as the proposed approach, there is much saving in memory storage requirement using the proposed approach. Obtaining the sinusoidal encoder signals is integral to the successful implementation of the proposed approach. In some encoders, these sinusoidal encoder signals are not available due to constraints in their mechanical design.

2.2 The RBF Neural Network

The RBFNN is commonly used for the purpose of modeling uncertain and nonlinear functions. Utilizing the RBFNN for modeling purposes can be seen as an approximation problem [19] in a high-dimensional space. Consider the RBFNN, which is depicted as a two-layered processing structure in Figure 2.1. The hidden layer consists of an array of computing units, i.e., ϕ_1, ϕ_2, ..., ϕ_N. These hidden units provide a set of functions of the input vectors (i.e., x_1, x_2, ..., x_J) as they are expanded into the higher dimension hidden-unit space. The mapping from the input vectors to the outputs of the hidden units is nonlinear, whereas the mapping from the outputs of the hidden units to the final output of the RBFNN is linear.

The general mapping function [18] of the RBFNN can be represented by:

$$f(\bar{x}) = \sum_{i=1}^{N} w_i \phi_i(\bar{x}), \tag{2.1}$$

$$\phi_i(\bar{x}) = exp(-\frac{||\bar{x} - c_i||^2}{2\sigma_i^2}), \tag{2.2}$$

where $\phi_i(\bar{x})$ denotes the basis function and $\bar{x} = [x_1, x_2, .., x_J]^T$. Each hidden unit contains a parameter vector called a center (c_i), and it calculates a squared distance between the center and the input vector ($\bar{x}$). The result is then divided by the width (σ_i) and passed through an exponential function. The second layer of the RBFNN acts as a summer with a set of weights, i.e., w_1, w_2, ..., w_N. The free variables that needs to be tuned are the weights w_i's, the centers c_i's and the widths σ_i's. The reader may refer to ([17]-[20]) for more examples and applications of the RBFNN.

2.3 Principles of Proposed Interpolation Approach

The overall configuration of the two-stage RBFNN is shown in Figure 2.2. It consists of two stages; the precompensation stage and the interpolation stage. The inputs to the precompensation stage are the quadrature signals direct from the encoders $\bar{u}_1$

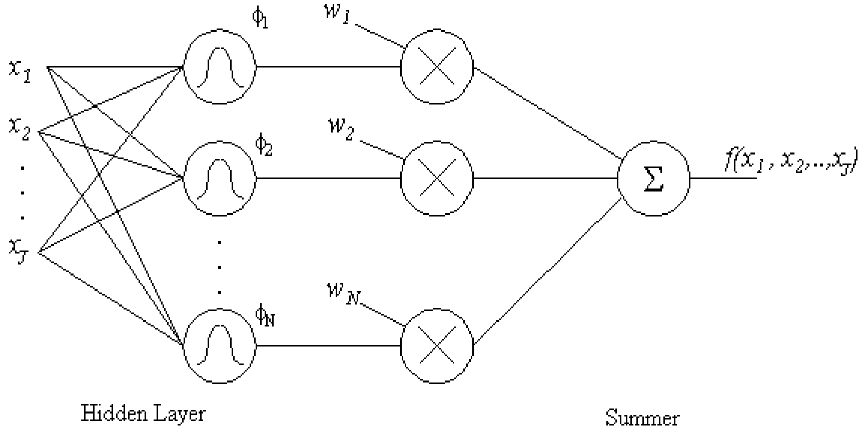


Figure 2.1: Structure of a two-layered RBFNN.

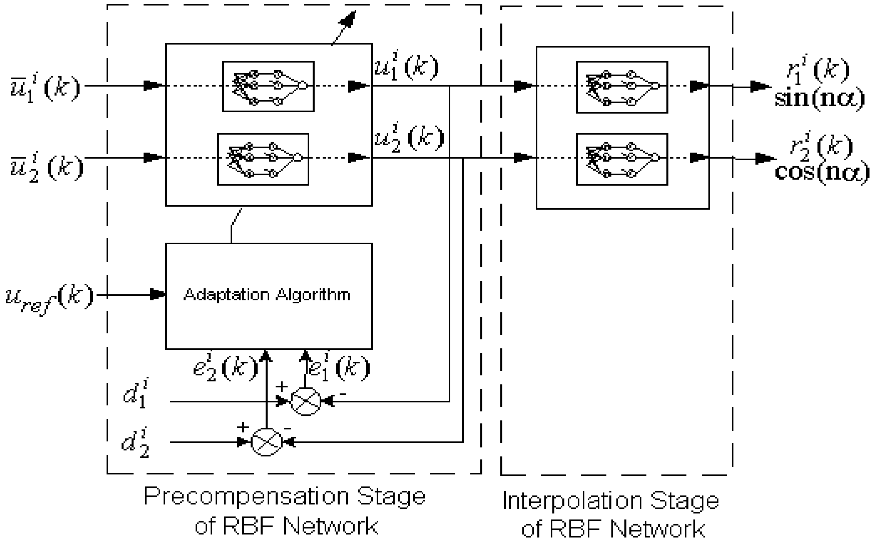


Figure 2.2: Overall configuration of the two-stage RBFNN.

and $\bar{u}_2$. The outputs of the precompensation stage are fed as inputs to the interpolation stage. The outputs from the interpolation stage are the higher order sinusoids, $sin(n\alpha)$ and $cos(n\alpha)$, where n refers to the order of interpolation.

Each of the two stages has the configuration of a two-layered RBFNN as shown in Figure 2.1. The precompensation stage corrects the errors in the raw encoder signals. An adaptation algorithm is used to refine the correction process, since the error characteristics in the raw signals exhibit a tendency to drift from time to time. An online batch updating process is used to update the RBFNN for the precompensation stage whenever a new batch of M time samples of the signals becomes available. The updating process is based on the modified Recursive Least Squares (RLS) algorithm [21]. The adaptation algorithm proposed here is different from RLS in that time averages of the input signals is not used in the proposed algorithm. Unlike the RLS, the proposed algorithm is independent of the stochastic properties of the input signsls. The interpolation stage is used to derive the high order sinusoids based on the corrected signals forthcoming from the precompensation stage.

For both stages, the objective may be described as follows:

Given a set of W different points in a p dimensional input space, $\{i.e., \mathbf{x}^t \in \Re^p, t = 1, 2, .., W\}$ and a corresponding set of W points in a q dimensional output space, $\{i.e., \mathbf{d}^t \in \Re^q, t = 1, 2, .., W\}$, the goal is to find a mapping function $\Im : \Re^p \rightarrow \Re^q$ that fulfills the relationship, such that

$$\Im(\mathbf{x}^t) = \mathbf{d}^t, \qquad t = 1, 2, ..., W. \tag{2.3}$$

For the precompensation stage (Figure 2.2), the mapping function will map the raw encoder signals ($\bar{u}_1$, $\bar{u}_2$) to the corrected ones (u_1, u_2) which in turn become the inputs to the interpolation stage. For the interpolation stage, the mapping function will fulfill the map from u_1 and u_2 to the higher-order sinusoids r_1 and r_2 respectively. A reference mark is available on most encoders. This reference mark modulates a reference output signal u_{ref} from the encoder once per mechanical revolution.

2.3.1 Precompensation Stage

Commonly encountered errors in the encoder signals include mean, phase offsets, amplitude deviation and waveform distortion. To reduce interpolation errors, it is necessary to correct these errors prior to interpolation. This section will describe how these error components can be corrected in an adaptive manner in the precompensation stage of the RBFNN of Figure 2.2. As mentioned earlier, an adaptive approach is useful for this purpose, since the error characteristics in the raw encoder signals can drift with time.

Ideally, the quadrature encoder signals (denoted by u_1 and u_2 respectively) are identical sinusoidal signals displaced by a phase of $\pi/2$ with respect to each other, described by:

$$\begin{cases} u_1 = A\cos\alpha \\ u_2 = A\sin\alpha \end{cases} \tag{2.4}$$

where α denotes the instantaneous phase and A denotes the amplitude of the signals. If there is no waveform distortion, the general equations relating the ideal and practical encoder signals can be obtained according to Heydemann's method [8],

$$\begin{cases} \bar{u}_1 = u_1 + m_1 \\ \bar{u}_2 = \frac{A_1 \cos(\alpha - \varepsilon)}{G} + m_2 \end{cases} \tag{2.5}$$

where m_l and m_2 are the mean values of the signals and ε is the phase shift. $\bar{u}_1$ and $\bar{u}_2$ are values obtained from the encoder. $G = \frac{A_1}{A_2}$ and A_1, A_2 are the actual amplitudes of the encoder signals. The offset parameters m_1, m_2, ε and G can be estimated using a least squares estimation method operating on the raw signals.

Using a two-layered RBFNN, this correction can be easily accomplished as a mapping from raw signals to ideal signals. In addition, unlike the Heydemann's method, waveform distortion can be addressed directly in the mapping function. To enable the precompensation stage of the RBFNN to adaptively fine-tune its parameters in concert with possible variation in the error characteristics, an adaptation algorithm is necessary. The adaptation algorithm used here is a modified version of the Recursive Least Squares algorithm [21]. The parameters of the RBFNN are updated in the Lyapunov sense so that the error in (2.6) can converge to zero asymptotically. The following algorithm is used to update the parameters of the RBFNN. The tuning of the other parameters in the RBFNN, e.g., the centers c_i's and widths σ_i's, follows a similar procedure as shown below for the weight vector W.

For the kth frame (batch) of data

$$\begin{aligned}
{e_j}^i(k) &= {d_j}^i - {u_j}^i(k), \\
E_j(k) &= \frac{\sum_{i=1}^{M} ({e_j}^i)^2(k)}{M}, \\
{u_j}^i(k) &= \sum_{r=1}^{N} w_{rj}(k) \phi_{rj}^i({\bar{u}_j}^i(k)), \\
\phi_{rj}^i({\bar{u}_j}^i(k)) &= exp[-\frac{\parallel {\bar{u}_j}^i(k) - c_j(k) \parallel^2}{2\sigma_j(k)^2}],
\end{aligned}$$

$$\begin{aligned} W_j(k) &= [w_{1j}(k) \quad w_{2j}(k) \quad .... \quad w_{Nj}(k)]^T, \\ \Phi_j^i(k) &= [\phi_{1j}^i(\bar{u}_j{}^i(k)) \quad \phi_{2j}^i(\bar{u}_j{}^i(k)) \quad .... \quad \phi_{Nj}^i(\bar{u}_j{}^i(k))]^T, \qquad (2.6) \end{aligned}$$

where

$$\begin{aligned} W_j(k) &= W_j(k-1) + \delta_j^i(k)\lambda_j^i(k), && (2.7) \\ \delta_j^i(k) &= \frac{\Phi_j^i(k)}{\parallel \Phi_j(k) \parallel^2}\left[1 - \rho\frac{e_j^i(k-1)}{\mid \lambda_j^i(k) \mid}\right], && (2.8) \\ \lambda_j^i(k) &= d_j^i - W_j^T(k-1)\Phi_j^i(k), && (2.9) \end{aligned}$$

for $i = 1, 2, ..., M; \;\; 0 \leq \rho < 1 \; and \;\; j = 1, 2.$

As mentioned earlier, the free parameters of the RBFNN are trained using a batch mode of updating. One kth frame represents M data points. That is to say, when every M data points are collected by the system, the free parameters of the precompensation stage are updated using (2.7). The convergent properties of the algorithm will be presented in the following. Suppose the Lyapunov energy function is chosen as:

$$V(k) = E_j(k). \qquad (2.10)$$

Therefore,

$$\begin{aligned} \Delta V(k) &= V(k) - V(k\text{-}1) \\ &= E_j(k) - E_j(k-1) \\ &= \frac{\sum_{i=1}^{M}[e_j{}^i]^2(k)}{M} - E_j(k-1) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{M}\sum_{i=1}^{M}({d_j}^i - {u_j}^i(k))^2 - E_j(k-1) \\
&= \frac{1}{M}\sum_{i=1}^{M}({d_j}^i - {W_j}^T(k){\Phi_j}^i(k))^2 - E_j(k-1) \\
&= \frac{1}{M}\sum_{i=1}^{M}({d_j}^i - W_j^T(k-1){\Phi_j}^i(k) - \lambda_j^{iT}(k)\delta_j^{iT}(k){\Phi_j}^i(k))^2 - E_j(k-1) \\
&= \frac{1}{M}\sum_{i=1}^{M}(\lambda_j^{iT}(k)(1 - \delta_j^{iT}(k){\Phi_j}^i(k)))^2 - E_j(k-1). \qquad (2.11)
\end{aligned}
$$

Substituting (2.9) to (2.11),

$$
\begin{aligned}
\Delta V(k) &= \frac{1}{M}\sum_{i=1}^{M}(\rho e_j^i(k-1))^2 - E_j(k-1) \qquad (2.12) \\
&= -\frac{1}{M}[1-\rho^2]E_j(k-1) \\
&< 0.
\end{aligned}
$$

Therefore, following the Lyapunov theory on stability ([22] and [23]), the approximation error (2.6) is stable and will converge to within a hypersphere centered at origin with radius τ, where τ is a small value. Figure 2.3 shows the fundamental encoder signals before and after the precompensation stage of the RBFNN. The various error components underlying in the raw fundamental signals would have been corrected for after this stage. The subsequent interpolation stage will only deal with ideal sinusoidal signals.

During the adaptive training phase of the RBFNN, the correct sine and cosine waves d_1 and d_2 have to be fed to the parameter adaptation algorithm as shown in Figure 2.2. These 2 signals have to be synchronized with the raw encoder signals $\bar{u}_1$ and $\bar{u}_2$ in order to feed the correct input-output pair to the adaptation algorithm.

Consider one raw encoder signal $\bar{u}_1^i$, the input-output pair for one frame is $\{(\bar{u}_1^1, d_1^1), (\bar{u}_1^2, d_1^2), ..., (\bar{u}_1^M, d_1^M)\ \}$.

To sum up the training procedure for the precompensation stage, the free parameters, i.e., weight vector W, the centers c_i's and the width of centers σ_i's, are updated according to (2.7). The updating of the above mentioned variables is continued until the error function E is below a preset threshold or negligibly small value. The variables are guaranteed to converge to the 'ideal' values as derived in (2.11). The 'ideal' values are defined as those which provide the error E as below a preset threshold or negligibly small value. The updating of the free parameters of the RBFNN for the precompensation stage is done adaptively in a batch pattern manner.

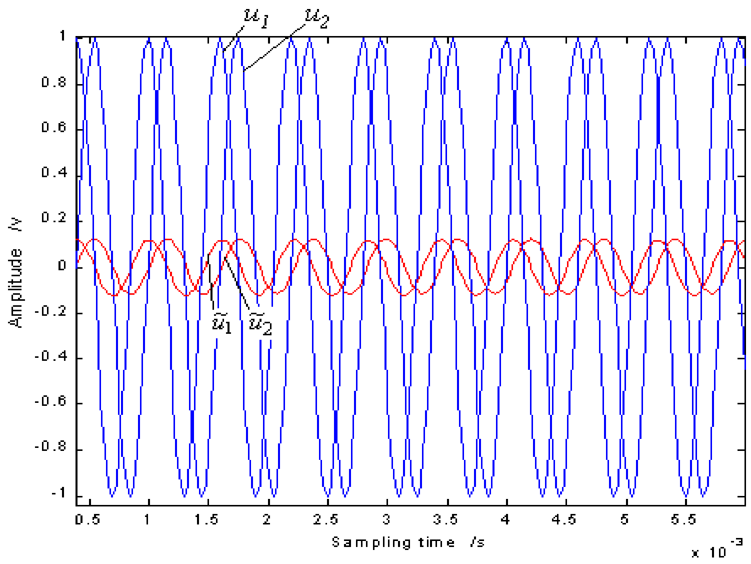


Figure 2.3: Encoder signals before and after the precompensation stage.

2.3.2 Interpolation Stage

The inputs to the interpolation stage are u_1 and u_2 from the precompensation stage. The outputs from the interpolation stage are the instantaneous values of the higher order sinusoids, i.e., r_1 and r_2. The output is dependent on the order of interpolation n. The RBFNN is used to fulfill this mapping. The network is trained offline, where the weights ω_i''s and the centers c_i''s of the RBFNN are the free parameters to be tuned. As the order of interpolation increases, the memory requirements of the network also increases accordingly, since the mapping function $\Im(.)$ will become more complicated. Thus, more computing units ϕ_i''s, subsequently more weights w_i''s to be tuned, are needed to implement the interpolation.

To reduce the memory requirements of this single-stage RBFNN, it is useful to minimize the level of redundancy within the RBFNN. To this end, it is noted that there is a strong degree of symmetry in a pure sinusoid. By considering only a quarter of the full sinusoid, the mapping function $\Im(.)$ between the absolute value of the inputs ($|u_1|$ and $|u_2|$) and the absolute value of the higher order sinusoid outputs (r_1 and r_2) can be fully represented. The sign of the higher order sinusoids can be subsequently restored by inferring the signs of u_1 and u_2, according to the schedule below.

${u_1}^i(k)$	${u_2}^i(k)$	Range	Output
${u_1}^i(k) \geq 0$	$\rightarrow {u_2}^i(k) \geq 0$	$0 \sim \pi/2$	$\rightarrow {r_1}^i(k)$
			$\rightarrow {r_2}^i(k)$
	$\rightarrow {u_2}^i(k) < 0$	$\pi/2 \sim \pi$	$\rightarrow {r_1}^i(k)$
			$\rightarrow -{r_2}^i(k)$
${u_1}^i(k) < 0$	$\rightarrow {u_2}^i(k) \geq 0$	$3\pi/2 \sim 2\pi$	$\rightarrow -{r_1}^i(k)$

		$\rightarrow {r_2}^i(k)$
$\rightarrow {u_2}^i(k) <0$	$\pi \sim 3\pi/2$	$\rightarrow -{r_1}^i(k)$
		$\rightarrow -{r_2}^i(k)$

There are many different techniques available to tune the parameters of the RBFNN. They include the 'Fixed-centers-selected-at-random', 'Self-organized selection-of-centers' and 'Supervized-selection-of-centers'. The reader is referred to ([17]-[19]) for more detailed discussions of the available tuning techniques. In this chapter, the technique adopted to tune the parameters of the RBFNN is the 'Supervized-selection-of-centers' [17], where the parameters undergo a supervized batch learning process using error-correction learning, i.e. gradient descent procedure. The main objective of the supervized learning process is to minimize the value of the cost function:

$$\xi_j^i(e) = \frac{1}{2}\sum_{i=1}^{I}(v_j^i(e))^2, \tag{2.13}$$

where I is the number of data points in one period of u_j^i used to tune the parameters in the eth epoch, and $v_j^i(e)$ is the error signal between the desired and actual output values of the RBFNN, defined as:

$$v_j^i(e) = z_j^i - r_j^i(e), \tag{2.14}$$

where z_j^i and $r_j^i(e)$ (where $i = 1, .., I$; $j = 1, 2$) are the desired output and the actual RBFNN output respectively. The desired output values z_1^i and z_2^i are obtained from the amplitudes of the ideal mathematical functions of the higher orders of sine and cosine respectively.

For the eth epoch,

$$\begin{aligned} {r_j}^i(e) &= \sum_{r=1}^{N} w'_{rj}(e)\phi^i_{rj}({u_j}^i(e)), \\ \phi^i_{rj}({u_j}^i(e)) &= exp[-\frac{\parallel {u_j}^i(e) - c'_j(e) \parallel^2}{2\sigma'_j(e)^2}], \\ W'_j(e) &= [w'_{1j}(e) \quad w'_{2j}(e) \quad .... \quad w'_{Nj}(e)]^T, \\ \Phi^i_j(e) &= [\phi^i_{1j}({u_j}^i(e)) \quad \phi^i_{2j}({u_j}^i(e)) \quad .... \quad \phi^i_{Nj}({u_j}^i(e))]^T. \end{aligned} \tag{2.15}$$

The following algorithm is used to update the parameters of the RBFNN for the interpolation stage in an offline manner. The tuning of the other parameters in the RBFNN for the interpolation stage, e.g., the centers c'_i's and the widths σ'_i's, follow a similar procedure as shown below for the weight vector W'. There is no need to adaptively update the parameters of the RBFNN for the interpolation stage because the RBFNN mapping function for this stage remains unchanged.

The updating of the weights are as follows:

$$W'_j(e) = W'_j(e-1) + \eta \frac{\sum_{i=1}^{I} {v_j}^i(e){u_j}^i(e)}{2I}, \tag{2.16}$$

where η is a constant.

To sum up the training procedure for the interpolation stage, the free parameters, i.e., weight vector W', the centers c'_i's and the width of centers σ'_i's, are updated according to (2.16). The above mentioned variables is trained offline until the error function ξ is below a preset threshold or negligibly small value. The free parameters are made to converge to the 'ideal' values. The 'ideal' values are defined as those which provide the error ξ as below a preset threshold or negligibly small value. The updating process of the free parameters of the RBFNN for the interpolation stage is done offline in a batch pattern manner.

2.3.3 Conversion to Binary Pulses

In order for the encoder signals to be received by a general purpose incremental encoder interface, the quadrature sinusoidal signals may be converted to a series of binary pulses. An analog comparator may be used to transform the high order sinusoids into pulses. As shown in Figure 2.4, the comparator will simply switch the pulse signals when the associated sinusoidal signal crosses zero. The rest of the analog information will not be used.

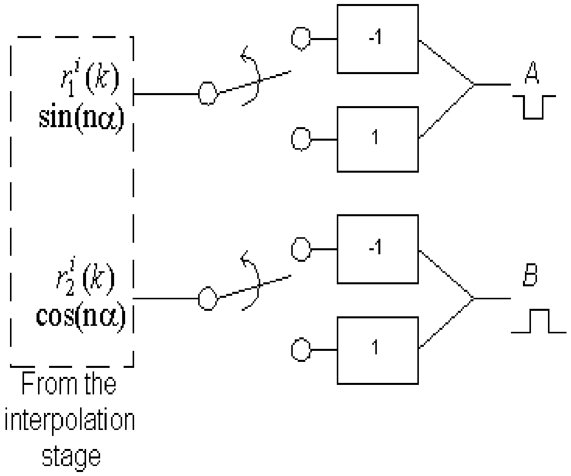


Figure 2.4: Conversion to binary pulses using a comparator.

Alternatively, this transformation can be done within the RBFNN. The $\sin(n\alpha)$ and $\cos(n\alpha)$ values can be converted into binary values A and B respectively, by training the RBFNN to conform to the following equations:

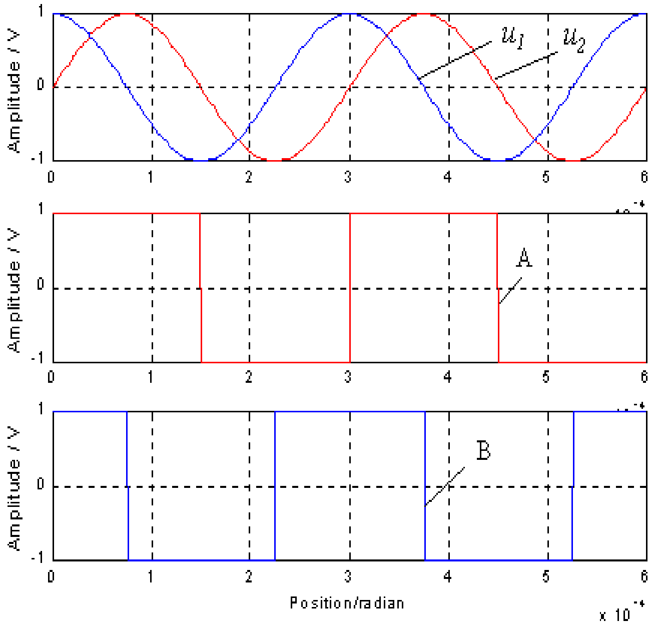


Figure 2.5: Quadrature sinusoidal signal decoding.

$$\begin{cases} A = 1, \sin(n\alpha) \geq \delta \\ A = -1, \sin(n\alpha) \leq -\delta \end{cases} \tag{2.17}$$

$$\begin{cases} B = 1, \cos(n\alpha) \geq \delta \\ B = -1, \cos(n\alpha) \leq -\delta \end{cases} \tag{2.18}$$

Thus, A and B, which are quadrature square signals, can be generated directly (Figure 2.5). δ can be 0 or a small value set according to the threshold of measurement noise.

2.3.4 Direct Conversion to Digital Position

The pulse outputs can also be easily converted into digital position values, which can be directly used for control purposes without further computations. This is especially true if the aforementioned interpolation is integrated into a general digital controller.

Alternatively, the encoder card can be made PC-bus based and the general motion controller can acquire the digital position value directly from the register or shared memory. In this case, the D/A converters for the encoder card are not required.

2.4 Simulation and Experimental Study

In this section, simulation and experimental results to illustrate the performance of the online adaptive correction and interpolation approach will be presented. In the experiment, raw data is acquired from a linear encoder (model: $Heidenhein\ LIP481A$) attached to the slide of a linear motor. A piezoelectric linear motor (Figure 2.6) manufactured by $Nanomotion$ is used as the test platform. The linear motor is mounted on a single axis linear stage manufactured by *Steinmeyer*. Table 1 shows the specifications of the stage and the motor. These raw signals are then fed to a $dSPACE$ controller with a high-speed A/D card, on which the RBF-based algorithms are implemented. The raw signals are accordingly precompensated and interpolated to higher order sinusoids. Figures 2.7 and 2.8 show the interpolation results with $n=$ 64 and 4096 respectively. Figure 2.9 shows the interpolated encoder signals converted to pulses with $n=$ 4096. The pulses are scaled to different applitudes for easy observation. The parameters of the RBFNN are initialized to random values. The RBFNNs for the precompensation and the interpolation stage are first trained offline using training data. During the normal operations of the system, the RBFNN for the precompensation stage is then fine-tuned using the online adaptation algorithm discussed in the previous section.

2.4.1 Simulation Study

To highlight the proposed approach to the precision and accuracy of position measurements, a simulation study is performed. A precise step reference signal (Figure 2.10) and a sinusoidal reference signal (Figure 2.11) is applied to the piezoelectric linear motor in two separate stages. The amplitude of the step signal and the sinusoidal signal is $1.005cm$. The well-known PID controller is used to control the piezoelectric linear motor. The positioning and tracking performances of the piezoelectric linear motor with a step reference signal and a sinusoidal reference signal are shown in Figures 2.12 and 2.14 respectively. A close-up view of Figure 2.12 is shown in Figure 2.13. It can be observed that with interpolation of encoder signals, the tracking and positioning performances of the controller are greatly improved. Interpolation of the encoder signals increases the precision and accuracy of position measurements of the system.

2.4.2 Experimental Study

For the experimental study, the same precise step reference signal (Figure 2.10) and sinusoidal reference signal (Figure 2.11) is applied to the piezoelectric linear motor in two stages. The well-known PID controller is used for this experimental study. With the step reference input signal, the positioning performance of the PID controller (with and without interpolation of the encoder signals) is shown in Figure 2.15. In a similar manner, Figure 2.16 demonstrates the tracking performance of the PID controller (with and without interpolation of the encoder signals). The error convergence rate of the RBFNN for the precompensation stage and the interpolation stage are shown in

Figures 2.17 and 2.18. As similar to the simulation study, it can be observed that with interpolation of encoder signals, the precision and accuracy of position measurements of the system are increased. Interpolation of encoder signals smoothens the tracking error and reduces the steady state error.

The parameters of the RBFNN for the interpolation stage adapt themselves continuously to changes in the raw encoder signals, in the presence of noise. As can be seen, the network converges sufficiently fast enough to provide the adaptive nature of the interpolation system. A total of about 32 and 50 basis functions (i.e., computing units) are used for the precompensation and interpolation stage, respectively. The value of η in (2.16) is preset at 0.0001.

The effectiveness of the RBFNN for interpolation is determined by the generalization property of the trained network ([24] and [25]). This is even more critical for applications involving high precision motion. A network that is designed to generalize well will produce a correct input-output mapping even when the test data is different from the data used to train the network. The network may end up fitting misleading changes due to noise in the training data, resulting in poor generalization. According to [25], as long as the standard deviation of the noise is large as compared to the sample spacing of the data points, an approximation to the mapping function $\Im$ of the RBFNN can be obtained, in the face of noise. Thus, the RBFNNs used in this chapter are fed with test data to train them adaptively. This can be seen as a recursive process in the batch updating of the free parameters of the RBFNNs.

As can be clearly seen in the simulation and experimental studies, the proposed approach improves the tracking and positioning performance of the controller, with fast speed. Although the look-up table method [26] may give similar results as the proposed approach, there is much saving in memory storage requirement using the proposed approach. It should be stressed here that given the same constraints on available memory size, the proposed approach in this chapter is superior, as compared to the look-up table method. Obtaining the sinusoidal encoder signals is integral to the successful implementation of the proposed approach. In some encoders, these sinusoidal encoder are not available due to constraints in their mechanical design. There are advantages associated with this approach when compared to the look-up table approach [26], in the use of storage memory and execution speed. Under the proposed approach, it is only necessary to reserve memory space for storing the parameters (i.e., the weights and centers) of the RBFNNs. The number of data points used to train the RBFNNs for the precompensation stage and the interpolation stage is the number of weights required in the RBFNN. In the experimental study, only seven points are required to map a complete sine or cosine function (Figure 2.19) when the redundancy present is eliminated. For order of interpolation $n = 16$, a total of 308 data points (i.e., $25.5kBytes$ of memory space) are needed for the RBF approach, while a total of $12,888$ data points (i.e., $1068kBytes$ of memory space) are needed for a look-up table method. With a lesser demand on memory storage space, the execution speed of the RBF approach is also much increased, i.e., about 42 times faster as compared to the look-up table method. Besides these, maintenance of the table is much of a hassle, especially in a common shopfloor which have many similar setups. Reconfiguring the correction and interpolation parameters of the RBFNN using the proposed approach is simple and fast.

Table 1 : Specifications of Piezoelectric Linear Motor

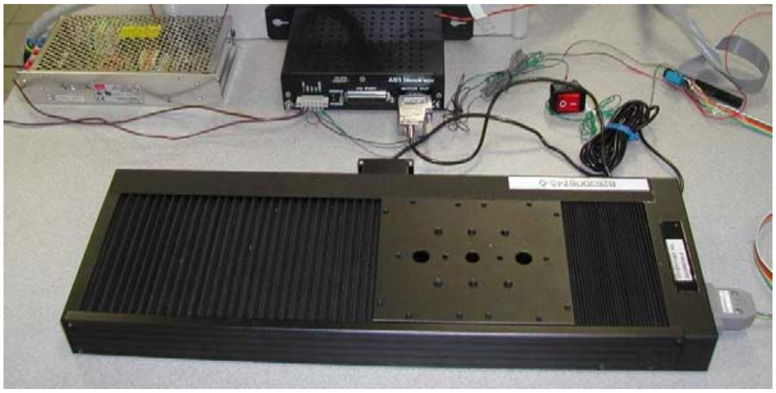


Figure 2.6: Test platform: Piezoelectric linear motor.

Travel	Velocity (Max)	Resolution	Output force (Max)
200mm	250 mm/s	$0.1\mu m$	40N

2.5 Conclusions

An adaptive online approach for the correction and interpolation of quadrature encoder signals has been developed, suitable for application to precision motion control systems. It is based on the use of a two stage double-layered RBFNN. The first RBFNN stage is used to adaptively correct for the imperfections in the encoder signals such as mean, phase offsets, amplitude deviation and waveform distortion. The second RBFNN stage serves as the inferencing machine to map the quadrature encoder signals to higher order sinusoids. Simulation and experimental results verify the effectiveness of the RBF approach, compared to a look-up table approach.

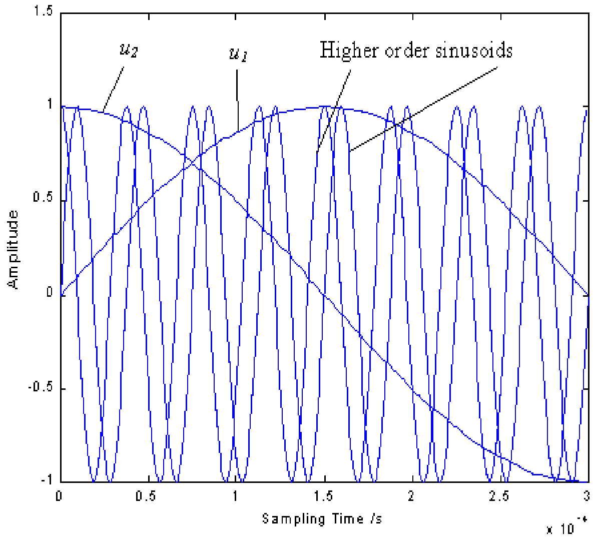


Figure 2.7: Encoder signals before and after interpolation, with $n = 64$.

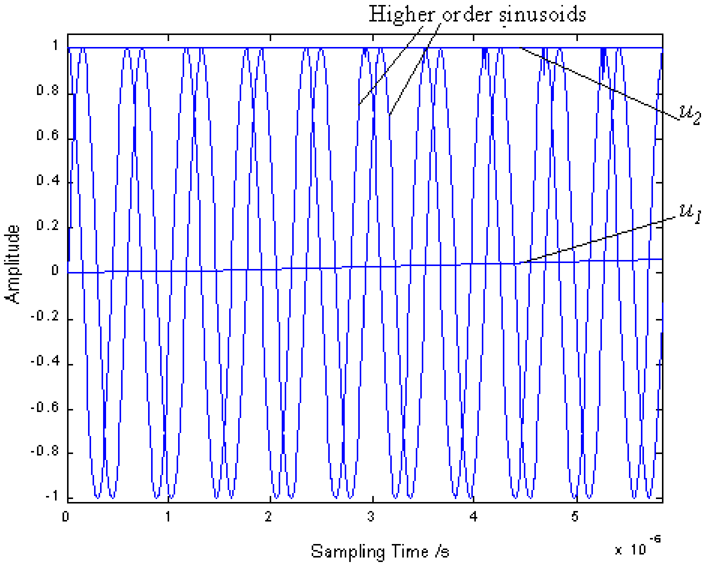


Figure 2.8: Encoder signals before and after interpolation, with $n = 4096$.

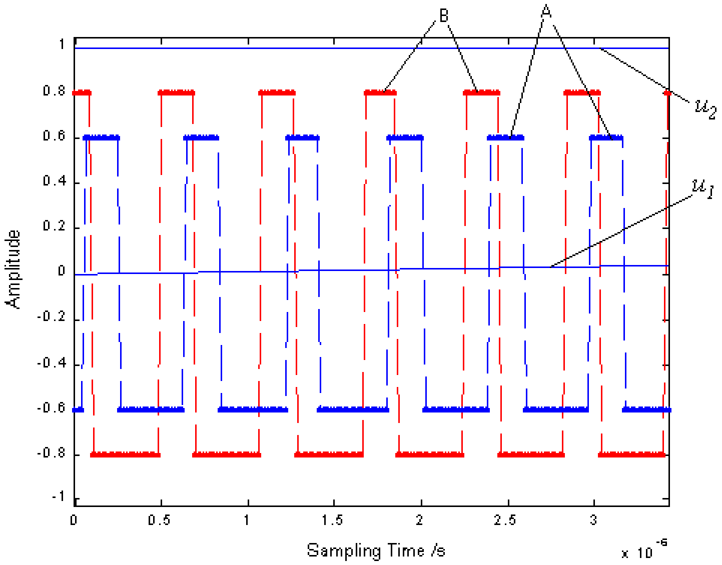


Figure 2.9: Encoder signals converted to pulses, with $n = 4096$.

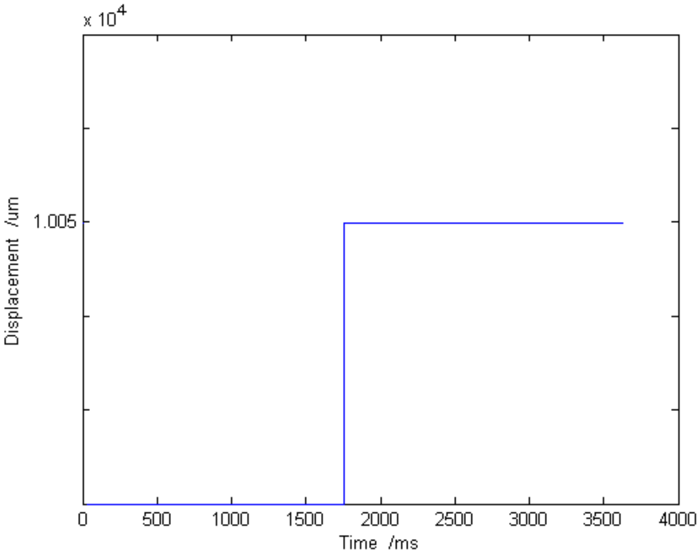


Figure 2.10: Precise step reference function.

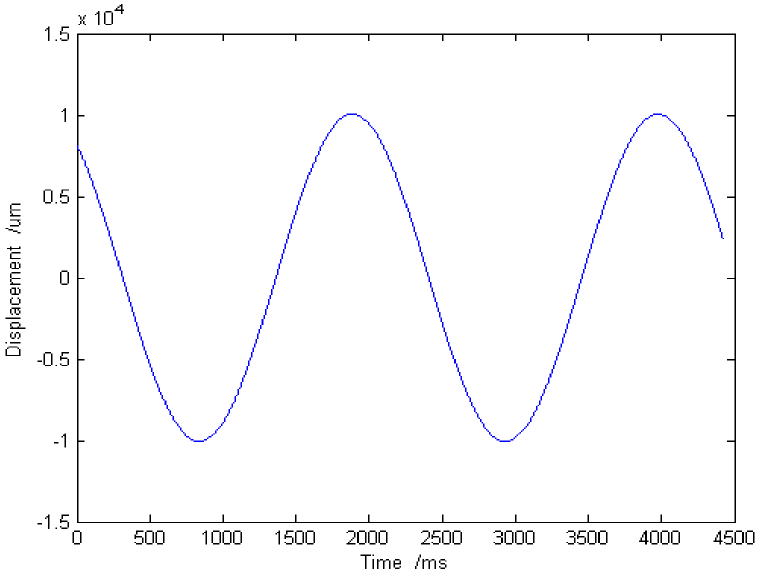


Figure 2.11: Precise sinusoidal reference function.

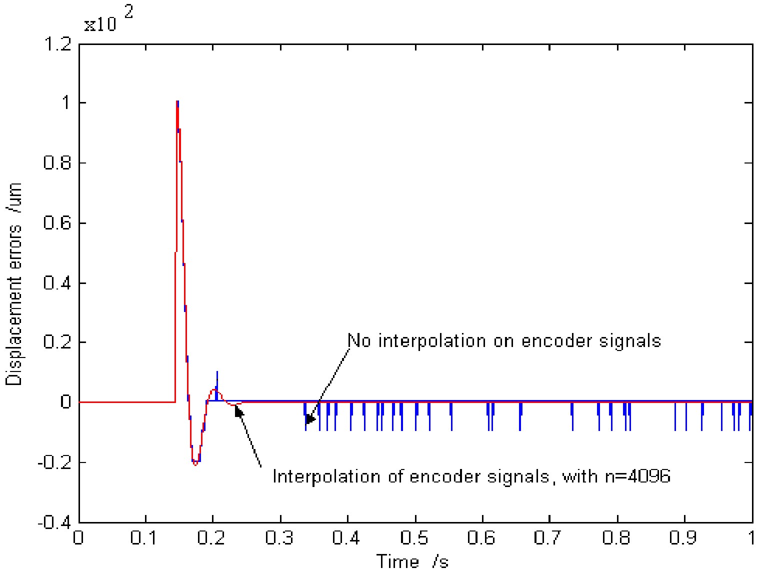


Figure 2.12: Positioning performance of the linear piezoelectric linear motor with a precise step reference input signal (Simulation study).

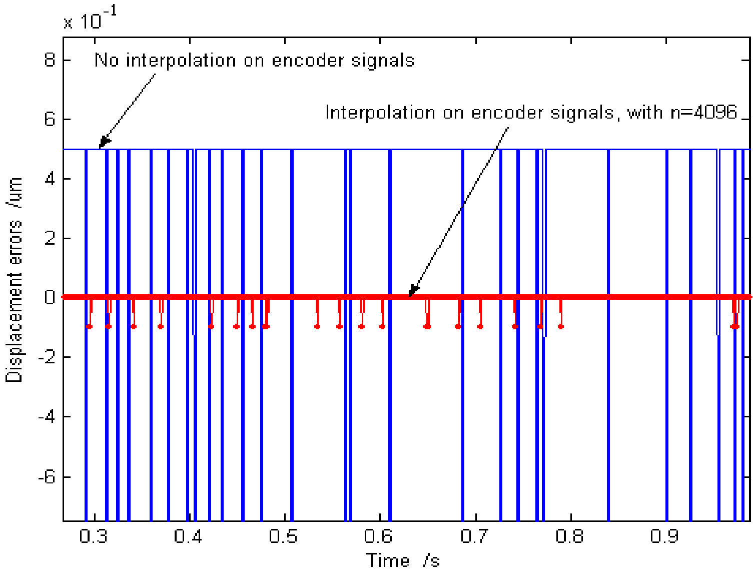


Figure 2.13: Positioning performance of the linear piezoelectric linear motor with a precise step reference input signal (More detailed figure).

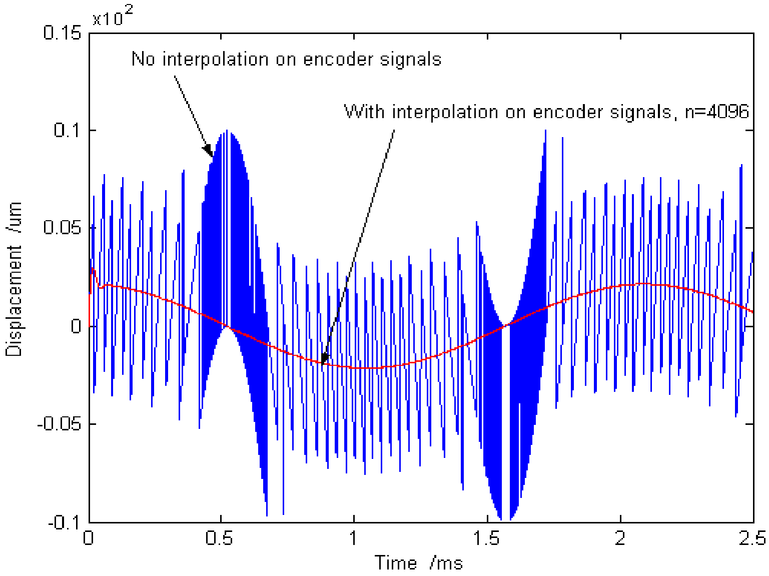


Figure 2.14: Tracking performance of the linear piezoelectric linear motor with a sinusoidal reference input signal (Simulation study).

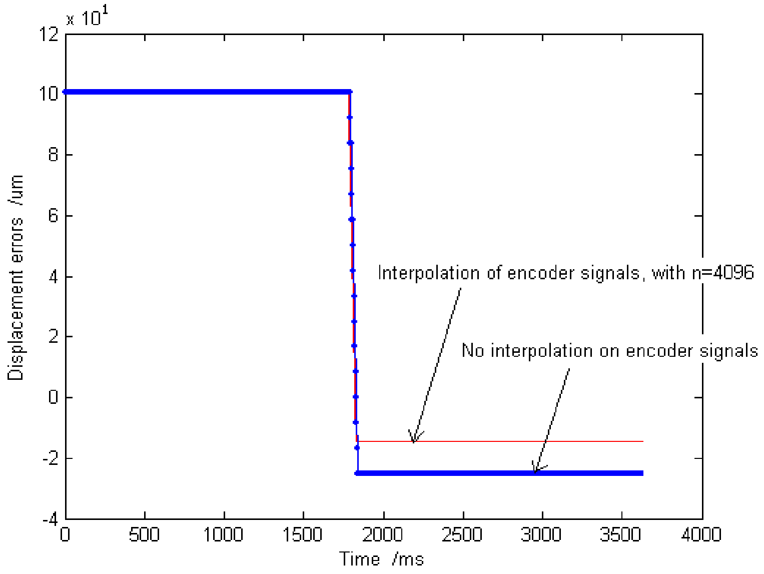


Figure 2.15: Positioning performance of the linear piezoelectric linear motor with a precise step reference input signal (Experimental study).

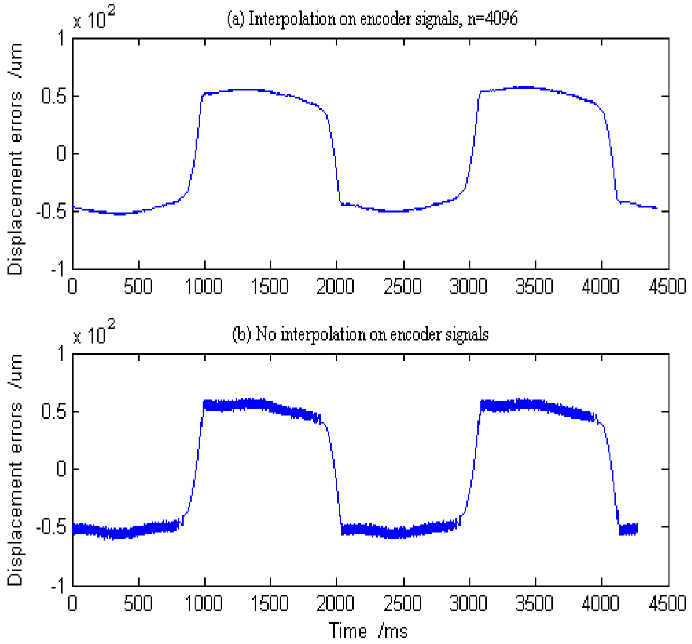


Figure 2.16: Tracking performance of the linear piezoelectric linear motor with a sinusoidal reference input signal (Experimental study).

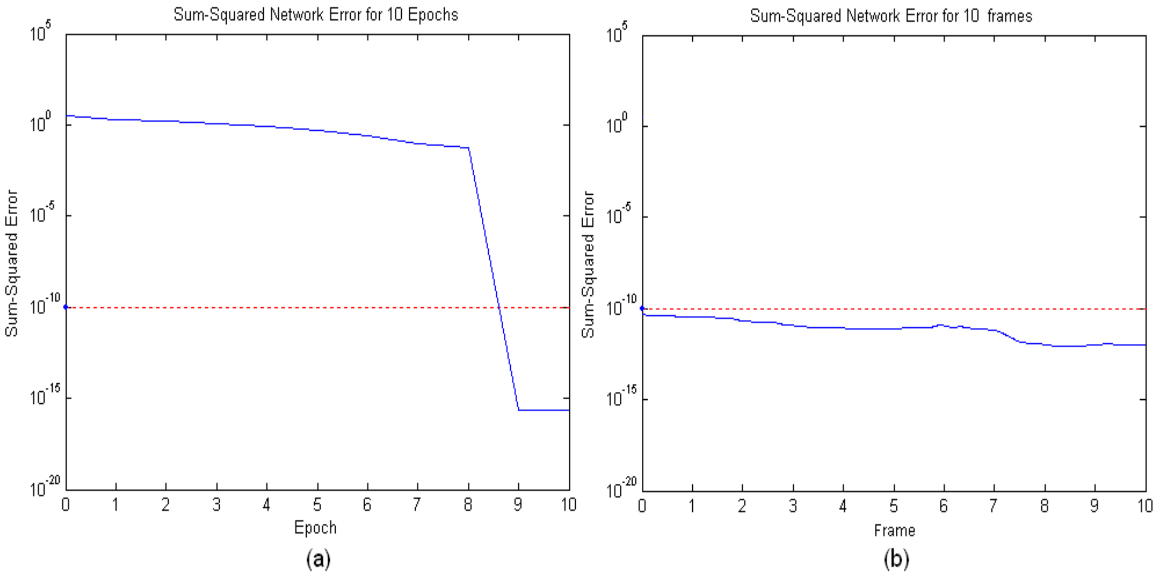


Figure 2.17: Error convergence rate of the RBFNN for the precompensation stage during the experimental study. (a) During the intial stage of the experiment (offline). (b) After 1 hour of operation of the experiment (online).

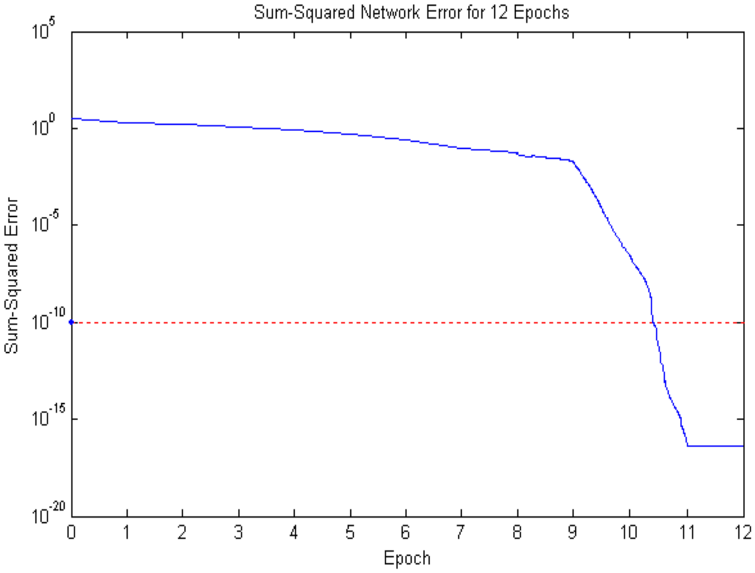


Figure 2.18: Error convergence rate of the RBFNN for the interpolation stage during the experimental study.

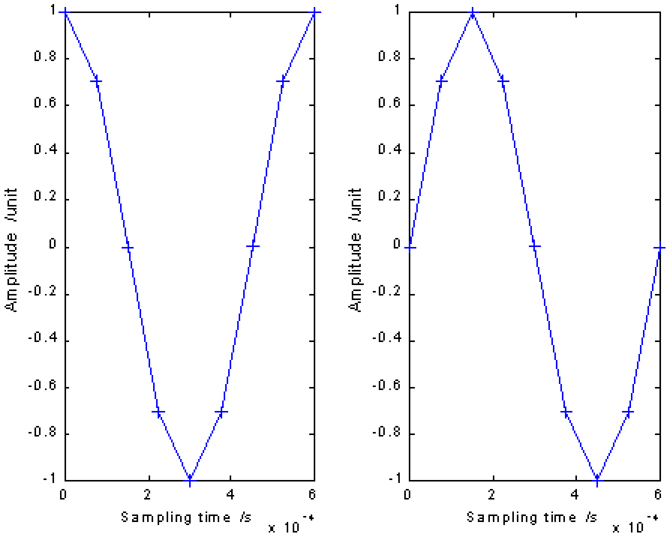


Figure 2.19: Number of data points required to model the sine and cosine function for the RBF approach.

Chapter 3

Intelligent Control: Combined PID and Adaptive Nonlinear Control for Precision Motion Systems

3.1 Introduction

The PID controller has remained, by far, as the most commonly used controller in practically all industrial control applications. The reason is that it has a simple structure which is easy to be understood by engineers. Over the years, many techniques have been suggested for tuning of the PID parameters, such as the refined Ziegler-Nichols method [27], the gain-phase margin method [28], an optimization method [29], and one based on the *Internal Model Control* [30]. Among them, the model-based tuning methods appear to be very encouraging [31]. However, the limitations of PID control rapidly become evident when applied to more complicated systems such as those with a time-delay, poorly damped, nonlinear and time-varying dynamics. For these processes, nonlinear adaptive control may be necessary to achieve good control performance. Recently, a nonlinear PD control scheme has been developed for a robot tracking application [32]. Based on a first-order model, a nonlinear PI control has also been proposed by Huang *et al.* [33].

One enabling technology which has made these and more modern applications possible is the advance and development in precision mechanisms and motion control. An increasing number of the precision motion systems today, general purpose or application specific, are based on the use of DC permanent magnet linear motors (PMLM) [34] for the main reason that among the electric motor drives available, the PMLMs are probably the most naturally akin to applications involving high speed and high precision motion control. The increasingly widespread industrial applications of PMLMs in various semiconductor processes, precision metrology and miniature system assembly are self-evident testimonies of the effectiveness of PMLMs in addressing the high requirements associated with these application areas. The main benefits of a PMLM include the high force density achievable, low thermal losses and, most importantly, the high precision and accuracy associated with the simplicity in mechanical structure. Unlike rotary machines, linear motors require no indirect coupling mechanisms as in gear boxes, chains and screws coupling. This greatly reduces the effects of contact-type nonlinearities and disturbances such as backlash and frictional forces, especially when they are used with aerostatic or magnetic bearings. However, the advantages of using mechanical transmission are also consequently reduced due to other problems that arise, such as the inherent ability to reduce the effects of model uncertainties and external disturbances [35]. An adequate reduction of these effects, either through a proper physical design or via the control system, is of paramount importance in order to achieve the end objectives of robust, high speed and high precision motion control.

There are several important challenges to the precision motion control system. First, the measurement system must be capable of yielding a very fine resolution in position measurements. Today, laser interferometers can readily yield a measurement resolution of down to one nanometer. Where cost is a concern, a high grade analog optical encoder in conjunction with an efficient interpolator can be used to provide sub-micrometer resolution measurements [8]. In the latter case, interpolation factors of up to 4096 times have been reported. This will effectively yield a resolution in the nanometer regime, given the fine scales manufacturing tolerance currently achievable. However, one should be cautious of interpolation errors associated with limited wordlength A/D operations, and imperfect analog encoder waveform with mean, phase offsets, noise as well as non-sinusoidal waveform distortion. An intelligent instrumentation approach capable of adaptive online correction and interpolation of quadrature encoder signals using neural networks is discussed in Chapter 2.

Secondly, the control electronics must have a sufficient bandwidth to cope with the high encoder count frequency associated with high speed motion on one hand, and a sufficiently high sampling frequency to circumvent anti-aliasing pits when motion is at a very low speed. Consequent of these requirements, the control algorithms must also be efficient enough to be executed within each time sample, and yet possess sufficient capacity to provide precision motion tracking and rapid disturbance suppression. This calls for a good weighted selection of efficient control components to address not only the specific dynamics of the servo system in point, but also exogenous disturbances arising from the application, including load changes, and drives-induced electro-magnetic interference.

Thirdly, the geometrical imperfections of the mechanical system should be adequately accounted for in the control system, if absolute positioning accuracy is crucial to the application concerned [36]. A 3D cartesian machine, for example, has 21 possible sources of geometrical errors (linear, angular, straightness, orthogonality errors from the 3 axes combined). Yet, many control engineers may evaluate positional accuracy solely with respect to encoder measurements, assuming ideal geometrical properties of the mechanical system. This assumption can lead to drastic and undesirable consequences when a high absolute positioning accuracy of the end object (e.g. machine tool) is required, since a very small tracking error with respect to encoder counts can be magnified many times over, when verified and calibrated in terms of absolute accuracy using a laser interferometer. These errors, arising from geometrical imperfections, can be calibrated and compensated for, if they are repeatible. The present common mode to deal with this problem is to build a look-up table model of the geometrical errors. The table maps an encoder reported position into the actual absolute position, and it can thus be used as the basis for geometrical offset compensation. Further work on this is discussed in Section 5.2.

In high precision motion control applications, vibrations induced from the mechanical system should be minimized as far as possible. Ideally, this calls for a highly rigid mechanical design, active damping and stable support structures. An intelligent vibration monitoring and control approach is elaborated in Chapter 4. This chapter will attempt to address the abovementioned challenges. The various selected control components, which constitutes the final overall strategy, will be discussed. Simulation and experimental results are provided to illustrate the effectiveness of the proposed control scheme.

3.2 Overall Control Strategy

The overall control structure is shown in Figure 3.1. The overall control signal is

$$u(t) = u_{FF}(t) + u_{PID}(t) + u_{adapt}(t), \tag{3.1}$$

where $u_{FF}(t)$, $u_{PID}(t)$ and $u_{adapt}(t)$ are the control signals provided by the feedforward controller, the PID feedback controller and the adaptive nonlinear controller respectively. $u_{adapt}(t)$ includes the compensation due to ripple and disturbance observer.

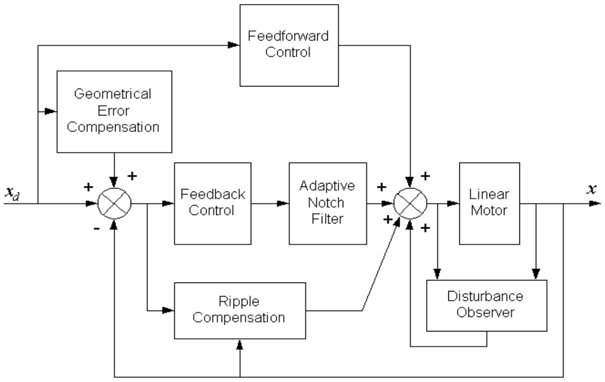


Figure 3.1: Overall structure of control system

In what follows, the purpose and design of each component depicted in Figure 3.1 will be elaborated. Since the design of several of these components will be based on a model of the PMLM, the initial part of this section will attempt to provide a concise system description of PMLM-based servo systems.

3.2.1 Mathematical Model

The dynamics of a servo-mechanical system can be described using a nonlinear mathematical model:

$$\begin{aligned} u(t) &= K_e\dot{x} + Ri(t) + Ldi(t)/dt, && (3.2) \\ f(t) &= K_f i(t), && (3.3) \\ f(t) &= m\ddot{x}(t) + \bar{f}_{fric}(\dot{x}) + \bar{f}_{load}(t), && (3.4) \end{aligned}$$

where $u(t)$ and $i(t)$ are the time-varying motor terminal voltage and armature current respectively; $x(t)$ is the motor position; $f(t)$ and $\bar{f}_{load}$ are the developed force and the applied load force respectively. $\bar{f}_{fric}$ denotes the frictional force present. The physical significance of the other physical parameters are elaborated in [37].

Since the electrical time constant is much smaller than the mechanical one, the delay of electrical response can be ignored. With this simplification, the following equation can be obtained (see [38]):

$$\ddot{x} = (-\frac{K_f K_e}{R}\dot{x} + \frac{K_f}{R}u(t) - \bar{f}_{fric} - \bar{f}_{load})/m. \qquad (3.5)$$

Let

$$\begin{aligned} a &= -\frac{K_f K_e}{mR}, && (3.6) \\ b &= \frac{K_f}{mR}, && (3.7) \\ f_{fric} &= \frac{1}{m}\bar{f}_{fric}, && (3.8) \\ f_{load} &= \frac{1}{m}\bar{f}_{load}. && (3.9) \end{aligned}$$

Thus, the following equivalent model is:

$$\ddot{x} = a\dot{x} + bu - f_{fric} - f_{load}. \qquad (3.10)$$

The PMLM model includes the dominant linear model and extraneous nonlinear effects. Among the many extraneous effects present, the two prominent nonlinear effects associated with PMLM are due to ripple and frictional forces, arising from the magnetic structure of PMLM and other physical imperfections. Figure 3.2 depicts a block diagram model of the motor, including explicitly the various exogenous disturbance signals present.

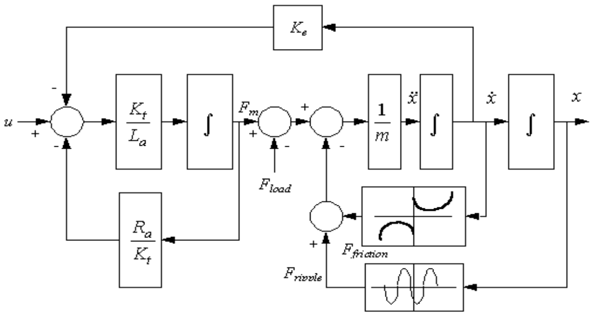


Figure 3.2: Model of PMLM.

3.2.2 Force Ripples

The thrust force transmitted to the translator of a PMLM is generated by a sequence of attracting and repeling forces between the poles and the permanent magnets when a current is applied to the coils of the translator. In addition to the thrust force, parasitic ripple forces are also generated in a PMLM due to the magnetic structure of PMLM. This ripple force exists in almost all variations of PMLM (flat, tubular, moving-magnet and etc.), as long as a ferromagnetic core is used for the windings.

The two primary components of the force ripple are the cogging (or detent) force and the reluctance force. The cogging force arises as a result of the mutual attraction between the magnets and iron cores of the translator. This force exists even in the absence of any winding current and it exhibits a periodic relationship with respect to the position of the translator relative to the magnets. Cogging manifests itself by the tendency of the translator to align in a number of prefered positions regardless of excitation states. There are two potential causes of the periodic cogging force in PMLMs, resulting from the slotting and the finite length of iron-core translator. The reluctance force is due to the variation of the self-inductance of the windings with respect to the relative position between the translator and the magnets. Thus, the reluctance force also has a periodic relationship with the translator-magnet position.

Collectively, the cogging and reluctant force constitute the overall force ripple phenomenon. Even when the PMLM is not powered, force ripples are clearly existent when the translator is moved along the guideway. There are discrete points where minimum/maximum resistance is experienced. At lower velocity, the rippling effects are more fully evident due to the lower momentum available to overcome the magnetic resistance.

Due to the direct-drive principle behind the operation of a linear motor, the force ripple has significant effects on the position accuracy achievable and it may also cause oscillations and yield stability problems, particularly at low velocities or with a light load (low momentum). The ripple periodicity has a fixed relationship with respect to position, but the amplitude can vary with velocity. Figure 3.3 shows the real-time open-loop step response of a tubular type PMLM manufacturing by Linear Drive, UK. Figure 3.4 shows the velocity-position characteristics of the PMLM with different step sizes (i.e., different steady-state velocity). Interesting observations may be inferred from these responses. First, the ripple period is independent of the step size (i.e., independent of the velocity), but is has a fixed relationship with respect to position. Secondly, the ripple amplitude is dependent on both position and velocity. At a higher velocity, the ripple amplitude decreases compared to when the motor is run at a lower velocity when the full dosage of ripple effects is experienced.

A first order model for the force ripple can be described as a periodic sinusoidal type signal:

$$F_{ripple}(x) = A(x)\sin(\omega x + \phi), \tag{3.11}$$

where $A(x)$ reflects the dependence of the ripple amplitude on x. Higher harmonics of the ripple may be included in higher order models.

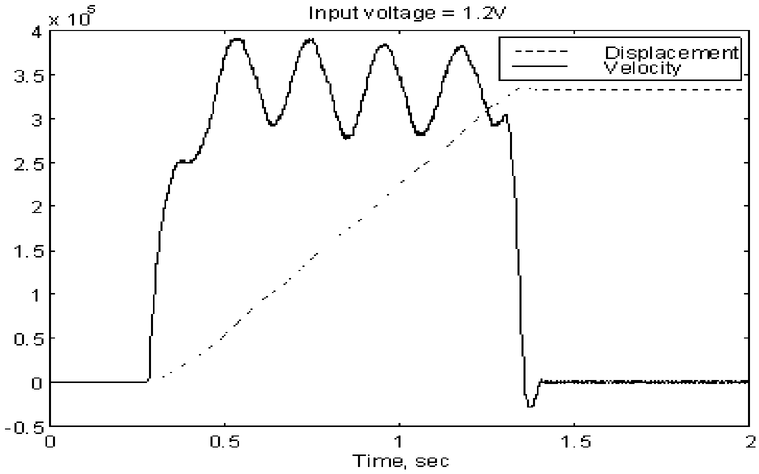


Figure 3.3: Open-loop step response of a PMLM.

3.2.3 Friction

Friction is inevitably present in nearly all moving mechanisms, and it is one major obstacle to achieving precise motion control. Several characteristic properties of friction

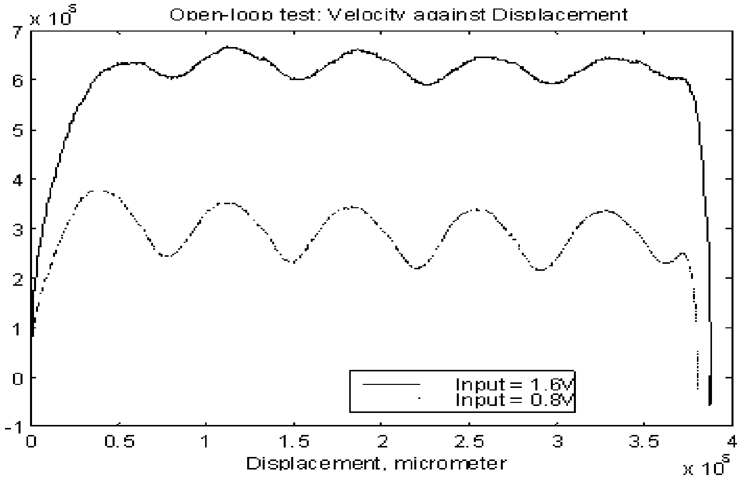


Figure 3.4: Graphs of velocity against position for different step sizes.

have been observed, which can be broken down into two categories: static and dynamic. The static characteristics of friction, including the stiction friction, the kinetic force, the viscous force, and the Stribeck effect, are functions of steady state velocity. The dynamic phenomena include pre-sliding displacement, varying breakaway force, and frictional lag. Many empirical friction models [39] have been developed which attempt to capture specific components of observed friction behavior, but generally it is acknowledged that a precise and accurate friction model is difficult to be obtained in an explicit form, especially for the dynamical component.

The frictional force affecting the movement of the translator can be modeled as a combination of Coulomb and viscous friction [39] as:

$$f_{fric} = [f_c + f_v|\dot{x}|]sgn(\dot{x}) + \delta f_{fric}, \tag{3.12}$$

where f_c is the minimum level of Coulomb friction and f_v is associated with the viscosity constant. δf_{fric} denotes possible directional bias associated with the Coulomb friction. For loading effects which are independent of the direction of motion, f_{load} can be described as:

$$f_{load} = f_l sgn(\dot{x}) + \delta f_{load}, \tag{3.13}$$

where δf_{load} denotes possible directional bias associated with the load which is the case when the load is transported in a direction aligned with gravitational force and f_l is a constant inertial friction due to the load. Cumulatively, the frictional and load force can be described as one external disturbance F, given by:

$$F = [f_1 + f_2|\dot{x}|]sgn(\dot{x}) + \delta f, \tag{3.14}$$

where $f_1 = f_l + f_c$, $f_2 = f_v$ and $\delta f = \delta f_{fric} + \delta f_{load}$. Figure 3.5 graphically illustrates the characteristics of F. The effects of friction can be greatly reduced using high quality bearings such as aerostatic or magnetic bearings.

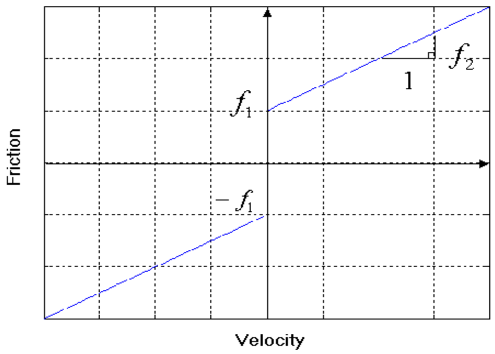


Figure 3.5: F-$\dot{x}$ characteristics.

3.2.4 Feedforward Control

The design of the feedforward control component is relatively straightforward, as compared to the other components in the overall control configuration (Figure 3.1). One point to note is that the reference position trajectory must be continuous and twice differentiable, otherwise a pre-compensator to filter the reference signal will be necessary. The only parameters required for the design of the feedforward control are the parameters of the second-order linear model.

Additional feedforward terms may be included for direct compensation of the nonlinear effects, if the appropriate models are available. For example, if a good signal model of the ripple force is available (3.11), then an additional static term in the feedforward control signal $u_{FF} = \frac{R}{K_f} F_{ripple}(x_d)$ can effectively compensate for the ripple force.

In the same way, a static friction feedforward pre-compensator can be installed if a friction model is available. In [40], an efficient way of friction modeling using relay feedback is proposed where a simple friction model (incorporating Coulomb and viscuous friction components) can be obtained automatically. This model can be used to construct an additional feedforward control signal, based only on the reference trajectories. In addition, if the motion control task is essentially repetitive, an iteratively refined additional feedforward signal can further reduce any control-induced tracking error. A possible scheme based on iterative learning control (ILC) can be found in [41] and [42]. The basic idea in ILC is to exploit the repetitive nature of the tasks as experience gained to compensate for the poor or incomplete knowledge of the system model and the disturbances. Essentially, the ILC structure includes a feedforward control component which refines the feedforward signal to enhance the performance of the next cycle based on previous cycles. A block diagram of the ILC scheme is depicted in Figure 3.6.

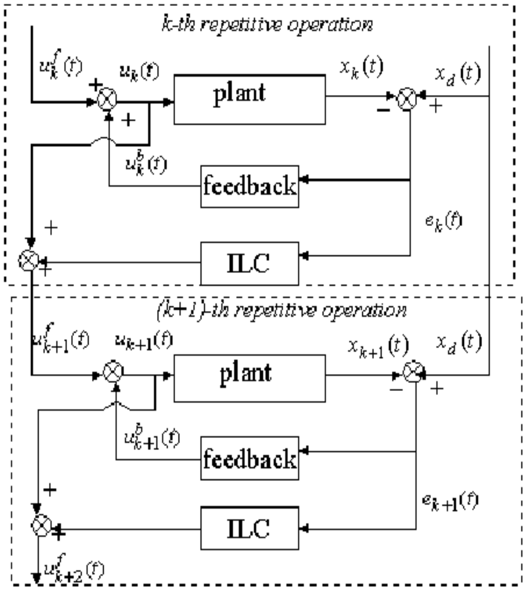


Figure 3.6: Iterative Learning Control

Characteristic of all feedforward control schemes, the performance is critically dependent on the accuracy of the model parameters. Therefore, feedforward control is usually augmented with suitable feedback control schemes, for example PID control.

3.2.5 PID Feedback Control

In spite of the advances in mathematical control theory over the last fifty years, industrial servo control loops are still essentially based on the three-term PID controller. The main reason is due to the widespread field acceptance of this simple controller which has been effective and reliable in most situations when adequately tuned. Some complex and advanced controllers have fared less favourably under practical conditions, despite the higher costs associated with implementation and the higher demands in control tuning, as compared to PID controller. It is very difficult for operators unfamiliar with advanced control to adjust the control parameters. Given these uncertainties, there is little surprise that PID controllers continue to be manufactured by the hundred thousands yearly and still increasing. In the composite control system, PID is used as the feedback control term. While the simplicity in a PID structure is appealing, it is also often proclaimed as the reason for poor control performance whenever it occurs. In this design, advanced optimum control theory is applied to tune PID control gains. The PID feedback controller is designed using the Linear Quadratic Regulator (LQR) technique for optimal and robust performance of the nominal system. The feedforward plus feedback configuration is often also referred to as a two-degree-of-freedom (2-DOF) control.

The nominal portion of the system (without uncertainty) is given by:

$$\dot{x}(t) \quad = \quad ax(t) + bu_{PID}(t), \tag{3.15}$$

where a and b are the system's parameters defined in (3.6) and (3.7), and

$$u_{PID} = kx_1 + k_{d1}x_2 + k_{d2}x_3, \tag{3.16}$$

where k, k_{d1} and k_{d2} are constants; x_1, x_2 and x_3 are the state variables. This is a PID control structure which utilizes a full-state feedback. The optimal PID control parameters are obtained using the LQR technique that is well known in modern optimal control theory and it has been widely used in many applications. It has a very nice robustness property, i.e., if the process is of single-input and single-output, then the control system has at least a phase margin of 60 degree and a gain margin of infinity. Under mild assumptions, the resultant closed-loop system is always stable. This attractive property appeals to the practitioners. Thus, the LQR theory has received considerable attention since 1950s.

The PID control is given by:

$$u_{PID} = -(r_0 + 1)B^T P x(t), \tag{3.17}$$

where P is the positive definite solution of the Riccati equation:

$$A^T P + PA - PBB^T P + Q = 0, \tag{3.18}$$

and $Q = H^T H$ where H relates to the states weighting parameters in the usual manner. A and B refer to the system matrices [43]. Note that r_0 is independent of P and it is introduced to weigh the relative importance between control effort and control errors. Note for this feedback control, the only parameters required are the parameters of the second-order model and a user-specified error weight r_0.

Where other state variables are available (e.g. velocity, acceleration and etc.), a full state feedback controller may also be used for the feedback control component. The implementation of such a scheme on PMLMs can be found in [43]. Adaptive and robust control has also been investigated in a previous study as an alternative to the PID feedback control, where the feedback control signal is adaptively refined based on parameter estimates of the nonlinear system model, using prevailing input and output signals. The achievable performance is highly dependent on the adequacy of the model, and the initial parameter estimates. Furthermore, full adaptive control schemes can greatly drain the computational resources available. More details on adaptive and robust control schemes for precision motion control are provided in [44].

3.2.6 Ripple Compensation

From motion control viewpoints, force ripples are highly undesirable, but yet they are predominantly present in PMLMs. They can be minimized or even eliminated by an alternative design of the motor structure or spatial layout of the magnetic materials such as skewing the magnet, optimizing the disposition and width of the magnets and etc. These mechanisms often increase the complexity of the motor structure. PMLM, with a slotless configuration is a popular alternative since the cogging force component due to the presence of slots is totally eliminated. Nevertheless, the motor may still exhibit significant cogging force owing to the finite length of the iron-core translator. Finite element analysis confirms that the force produced on either end of the translator is sinusoidal and unidirectional. Since the translator has two edges (leading and trailing edges), it is possible to optimize the magnet length so that the two sinusoidal force waveform of each edge cancel out each other. However, this would again contribute some degree of complexity to the mechanical structure. A more practical approach to eliminate cogging force would be to adopt a sleeve-less or an iron-less design in the core of the windings. However, this approach results in a highly inefficient energy conversion process with a high leakage of magnetic flux due to the absence of material reduction in the core. As a result, the thrust force generated is largely reduced (typically by 30 % or more). This solution is not acceptable for applications where high acceleration is necessary. In addition, iron-core motors, which produce high thrust force, are ideal for accelerating and moving large masses while maintaining stiffness during the machining and processing operations.

In this section, a simple approach will be developed which is based on the use of a dither signal as a “trojan horse” to cancel the effects of force ripples. The construction of dither signal requires knowledge of the characteristics of force ripples which can be obtained from simple step experiments. For greater robustness, real-time feedback of motion variables can be used to adaptively refine the dither signal characteristics.

It is assumed that the force ripple can be equivalently viewed as a response to a virtual input described in the form of a periodic sinusoidal signal:

$$u_{ripple} = A(x)\sin(\omega x + \phi) = A_1(x)\sin(\omega x) + A_2(x)\cos(\omega x). \tag{3.19}$$

The dither signal is thus designed correspondingly to eradicate this virtual force as:

$$u_{AFC} = a_1(x(t))\sin(\omega x) + a_2(x(t))\cos(\omega x). \tag{3.20}$$

Perfect cancellation will be achieved when

$${a_1}^*(x) = -A_1(x), {a_2}^*(x) = -A_2(x). \tag{3.21}$$

Feedforward compensation schemes are well-known to be sensitive to modeling errors which inevitably result in significant remnant ripples. An adaptive approach is thus adopted so that a_1 and a_2 will be continuously adapted based on desired trajectories and prevailing tracking errors.

Let

$$a = \begin{bmatrix} a_1(x) \\ a_2(x) \end{bmatrix}, \theta = \begin{bmatrix} \sin(\omega x) \\ \cos(\omega x) \end{bmatrix}, a^* = \begin{bmatrix} -A_1(x) \\ -A_2(x) \end{bmatrix}. \tag{3.22}$$

The system output due to AFC is then given by:

$$x_a = P[a - a^*]^T\theta, \tag{3.23}$$

where P denotes the system.

(3.23) falls within the standard framework of adaptive control theory. Possible update laws for the adaptive parameters will therefore be:

$$\dot{a_1}(x(t)) = -ge\sin(\omega x), \tag{3.24}$$

$$\dot{a_2}(x(t)) = -ge\cos(\omega x), \tag{3.25}$$

where $g > 0$ is an arbitrary adaptation gain.

Differentiating (3.24) and (3.25) with respect to time, it follows

$$\dot{a_1}(t) = -ge\dot{x}_d\sin(\omega x), \tag{3.26}$$

$$\dot{a_2}(t) = -ge\dot{x}_d\cos(\omega x). \tag{3.27}$$

In other words, the adaptive update laws (3.26) and (3.27) can be applied as an adjustment mechanism such that $a_1(t)$ and $a_2(t)$ in (3.20) converge to their true values. Full details on this adaptive ripple compensation scheme are provided in [45].

3.2.7 Disturbance Observer

The achievable performance of PMLMs is also unavoidably limited by the amount of disturbances present. These disturbances may arise due to load changes, system parameter perturbation owing to prolonged usage, measurement noise and high frequencies generated from the amplifiers (especially when a *Pulse Width Modulated* (PWM) amplifier is used), or inherent nonlinear dynamics such as the force ripples and frictional forces mentioned. Incorporating a higher resolution in the measurement system via the use of high interpolation electronics on the encoder signals can only achieve improvement in positioning accuracy to a limited extent. Thereafter, the amount of disturbances present will ultimately determine the achievable performance. In this subsection, this important issue of disturbance compensation for precision motion control systems will be addressed.

Figure 3.7 shows the block diagram of the "Disturbance Observer" part of the proposed control system which uses an estimate of the actual disturbance, deduced from a disturbance observer, to compensate for the disturbances. r, u, ξ, y, d and $\hat{d}$ denote the reference signal, control signal, measurement noise, system output, actual and estimated disturbance respectively. The disturbance observer, shown demarcated within the dotted box in Figure 3.7, estimates the disturbance based on the output x and the control signal u. P denotes the actual system. P_n denotes the nominal system which can be generally described by:

$$P_n = \frac{a_0}{s^l(s^{m-l} + a_1 s^{m-l-1} + ... + a_{m-l-1}s + a_{m-l})}, \tag{3.28}$$

where P_n is a m-th order system and has l poles at the origin. For example, a third order model, i.e., $l = 1, m = 3$, will be used

$$P_n = \frac{a_0}{s(s^2 + a_1 s + a_2)}. \tag{3.29}$$

The disturbance observer incorporates the inverse of the nominal system, and thus a low pass filter F is required to make the disturbance observer proper and practically realizable. For the choice of a third order model P_n, a suitable filter is

$$F(s) = \frac{f_3}{s^3 + f_1 s^2 + f_2 s + f_3}, \tag{3.30}$$

where f_1, f_2 and f_3 can be adjusted to satisfy a satisfactory compromise between tracking and disturbance rejection. [46] and [47] give the full details of the disturbance observer scheme.

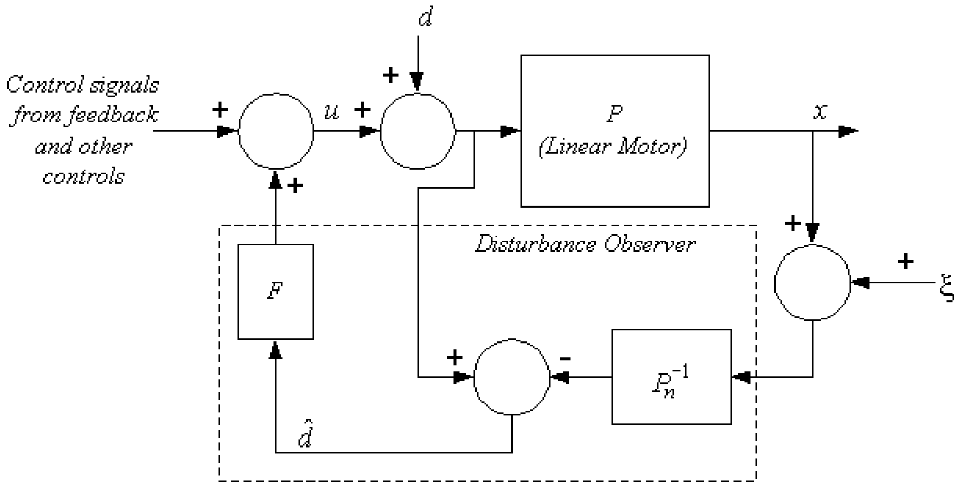


Figure 3.7: Control system with disturbance observer.

3.2.8 Vibration Control and Monitoring

Mechanical vibration in machines and equipment can occur due to many factors, such as unbalanced inertia, bearing failures on turbines, motors, generators, pumps, drives, turbofans, etc, poor kinematic design resulting in a non-rigid support structure, component failure and/or operations outside prescribed load ratings. The machine vibration signal can be typically characterised as a narrow-band interference signal anywhere in the range from 1 Hz to 500kHz. To prevent equipment damage from the severe shaking that occurs when machines malfunction or vibrate at resonant frequencies, a filter which terminate signal transmission at these frequencies will be very useful. When the machine is used to perform highly precise positioning functions, undue vibrations can lead to poor repeatibility properties, impeding any systematic error compensation effort. This results directly in a loss of precision and accuracy achievable. Two different approaches to control and monitor the undesirable vibrations are elaborated in Chapter 4. One possible approach is to design an adaptive notch filter to suppress the vibrations.

3.3 Robust Nonlinear PID Control

In this section, the development of the proposed control scheme will be described systematically in detail. Firstly, define the tracking error $e(t) = x_d(t) - x(t)$ and write (3.10) as

$$\ddot{e} = a\dot{e} - bu + f_{fric} + f_{load} + \ddot{x}_d - a\dot{x}_d. \tag{3.31}$$

Let $d = -(f_{load} + \ddot{x}_d - a\dot{x}_d)/b$ and $f(x, \dot{x}) = -(f_{fric} + f_{ripple})/b$ and

$$\ddot{e} = a\dot{e} - bu - b[f(x, \dot{x}) + d]. \tag{3.32}$$

Assumption 2.1: It is assumed that the desired trajectories $x_d, \dot{x}_d, \ddot{x}_d$ are bounded. Thus, it can be concluded that d is bounded, i.e.,

$$|d| < d_M, \tag{3.33}$$

since d_M is bounded.

Since $\frac{d}{dt}\int_0^t e(\tau)d\tau = e$, let the system's variables be $z = [\int_0^t e(\tau)d\tau \;\; e \;\; \dot{e}]^T$. Then, (3.32) can be put into the following equivalent state equation:

$$\dot{z} = Az + Bu + B[f(x, \dot{x}) + d], \tag{3.34}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & a \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ -b \end{bmatrix}. \tag{3.35}$$

For the nominal system

$$\dot{z} = Az + Bu, \tag{3.36}$$

the nominal control law $u_{nom} = Kz$ which is the standard PID control to guarantee uniform stability for the nominal system is proposed.

However, in order to compensate the effects of $f(x, \dot{x})$ which may possibly induce instability problems, it is possible to augment the nominal control signal with an additional signal to cancel the nonlinear terms. To this end, the RBFNN may be used to model $f(x, \dot{x})$. An additional control signal is thus provided by the RBFNN to compensate for the effects of $f(x, \dot{x})$. The main property of a RBFNN used here for estimation purposes is the function approximation property ([48]-[50]).

Since $f(x, \dot{x})$ is a nonlinear smooth function (unknown), it may be represented by a RBFNN with constant 'ideal' weights $w_i, i = 0, 1, 2, ..m$ and a sufficient number of basis functions $\phi(\cdot)$ on the compact set $\Omega = \{X| \; ||X_d - X|| \leq M\}$ where $X_d = [x_d, \dot{x}_d]^T$. Thus,

$$f(x, \dot{x}) = \sum_{i=0}^{m} w_i \phi_i(c_i X) + \epsilon, \tag{3.37}$$

where ϵ is the RBFNN approximation error satisfying $|\epsilon| \leq \epsilon_M$ with constant ϵ_M, and $\phi_i(c_i X)$ is given by

$$\phi_i(c_i X) = exp(-\frac{||X - c_i||^2}{2\sigma_i^2}), \tag{3.38}$$

where c_i is a 2-dimensional vector representing the center of the ith basis function, and σ_j is the variance representing the spread of the basis function. In general, the RBF basis parameters c_i and ideal RBF weights w_i^* are unknown and need to be estimated in the control design. Note that the set Ω and the bounding constant ϵ_M can be arbitrarily large.

$$f(x, \dot{x}) = W^T \Phi(C^T \bar{X}) + \epsilon, \tag{3.39}$$

where $W = [w_0, w_1, ..., w_m]^T, \Phi = [\phi_0, \phi_1, ..., \phi_m]^T, \bar{X} = [X^T, 1]^T$.

Let $\hat{W}, \hat{C}$ be estimates of the ideal W and C. Define the estimation errors as

$$\tilde{W} = W - \hat{W}, \;\; \tilde{C} = C - \hat{C}. \tag{3.40}$$

Then, applying the same approach of [51] and [52]:

$$\tilde{\Phi} = \Phi - \hat{\Phi} = \Phi(C^T \bar{X}) - \Phi(\hat{C}\bar{X}). \tag{3.41}$$

The Taylor series expansion for a given $\bar{X}$ may be written as:

$$\Phi(C^T \bar{X}) = \Phi(\hat{C}^T \bar{X}) + \Phi'(\hat{C}^T \bar{X})\tilde{C}^T \bar{X} + O(\tilde{C}^T \bar{X})^2. \tag{3.42}$$

Then

$$\begin{aligned} W^T \Phi(C^T \bar{X}) - \hat{W}^T \Phi(\hat{C}\bar{X}) = \tilde{W}^T [\Phi(\hat{C}^T \bar{X}) - \Phi'(\hat{C}^T \bar{X})\hat{C}^T \bar{X}] \\ + \hat{W}^T \Phi'(\hat{C}^T \bar{X})\tilde{C}^T \bar{X} + d_u, \end{aligned} \tag{3.43}$$

where

$$d_u = \tilde{W}^T \Phi'(\hat{C}^T \bar{X}) C^T \bar{X} + W^T O(\tilde{C}^T \bar{X})^2. \tag{3.44}$$

Since Φ is the RBF function, every element of $\Phi(C^T\bar{X}) - \Phi(\hat{C}^T\bar{X})$ is bounded by $2M$. Thus,

$$|d_u| \leq ||C||_F||\bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})||_F + ||W||||\Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|| + 2M|W|_1 \tag{3.45}$$

In this subsection, a combined control law constituting PID control and an adaptive control (provided by the RBFNN) is proposed. The control structure is as shown in Figure 3.8. The nominal system can be made asymptotically stable by properly choosing the PID parameters while the uncertain term can be compensated by the RBFNN. Thus, considering the system (3.34), the control input is given by

$$u = Kz - \hat{f}(x,\dot{x}) - sgn(z^TPB)\hat{d}, \tag{3.46}$$

where Kz is standard PID control and $\hat{f}(x,\dot{x})$ is the RBF functional estimate of $f(x,\dot{x})$ given by

$$\hat{f}(x,\dot{x}) = \sum_{i=0}^{m} \hat{w}_i\phi_i(x,\dot{x}), \tag{3.47}$$

where $\hat{w}_i$ is the estimate of the ideal weighting w_i^*, and $sgn(z^TPB)\hat{d}$ is a robustification term which provides robustness in the face of bounded disturbances.

For PID control, much work has been done in this area. In principle, the existing PID tuning methods, such as the gain and phase margin method, dominant pole method [53], and many other methods, can be employed for this purpose. Since the PID controller can be designed to ensure nominal stability for the dominant model, the following Lyapunov equation holds:

$$(A+BK)^T P + P(A+BK) = -I. \tag{3.48}$$

Let $\bar{A} = A + BK$. The following controller is proposed

$$u = Kz - \hat{W}^T \Phi(\hat{C}^T \bar{X}) + u_c, \tag{3.49}$$

where u_c is a compensator used to reject the effect of disturbances. Here, u_c is designed as

$$u_c = -[\hat{c}_f ||\bar{X}\hat{W}^T \Phi'(\hat{C}^T \bar{X})||_F + \hat{c}_w ||\Phi'(\hat{C}^T \bar{X})\hat{C}^T \bar{X}|| + \hat{c}_d] sgn(z^T PB). \tag{3.50}$$

For the weights of the RBFNN function and robustifying term, consider the following tuning rules

$$\begin{aligned} \dot{\hat{W}} &= \Gamma_w [\Phi(\hat{C}^T \bar{X}) - \Phi'(\hat{C}^T \bar{X})\hat{C}^T \bar{X}] z^T PB, & (3.51)\\ \dot{\hat{C}} &= \Gamma_c \bar{X} \hat{W}^T \Phi'(\hat{C}^T \bar{X}) z^T PB, & (3.52)\\ \dot{\hat{c}}_f &= r||\bar{X}\hat{W}^T \Phi'(\hat{C}^T \bar{X})||_F |z^T PB|, & (3.53)\\ \dot{\hat{c}}_c &= r||\Phi'(\hat{C}^T \bar{X})\hat{C}^T \bar{X}|| |z^T PB|, & (3.54)\\ \dot{\hat{c}}_d &= r|z^T PB|, & (3.55) \end{aligned}$$

where Γ_w, Γ_c, and $r > 0$.

With this controller, the system (3.34) can be written as

$$\begin{aligned} \dot{z} &= \bar{A}z + B[W^T \Phi(C^T \bar{X}) - \hat{W}^T \Phi(\hat{C}\bar{X})] + B(u_c + d + \epsilon) \\ &= \bar{A}z + B\{\tilde{W}^T [\Phi(\hat{C}^T \bar{X}) - \Phi'(\hat{C}^T \bar{X})\hat{C}^T \bar{X}] + \hat{W}^T \Phi'(\hat{C}^T \bar{X})\tilde{C}^T \bar{X}\} \\ &\quad +B(u_c + d_u + d + \epsilon). \end{aligned} \tag{3.56}$$

Theorem 3.1. The plant (3.34) with controller (3.49), PID control and adaptive laws (3.51)-(3.55), is stable in that all the signals in the closed-loop system are bounded. In addition, $\lim_{t\to\infty}||z(t)|| = 0$.

Proof. Consider the following Lyapunov function candidate:

$$V = V_0 + V_1 = z^T P z + \tilde{W}\Gamma_w^{-1}\tilde{W} + tr(\tilde{C}^T\Gamma_c^{-1}\tilde{C}) + \frac{1}{r}[\tilde{c}_f^2 + \tilde{c}_w^2 + \tilde{c}_d^2], \tag{3.57}$$

where

$$\begin{aligned} V_0 &= z^T P z, & (3.58)\\ V_1 &= \tilde{W}\Gamma_w^{-1}\tilde{W} + tr(\tilde{C}^T\Gamma_c^{-1}\tilde{C}) + \frac{1}{r}[\tilde{c}_f^2 + \tilde{c}_w^2 + \tilde{c}_d^2]. & (3.59) \end{aligned}$$

Taking the time derivative of V along the solution of (3.56), it can be shown that

$$\begin{aligned} \dot{V}_0 &= -z^T z + 2z^T PB\tilde{W}^T[\Phi(\hat{C}^T\bar{X}) - \Phi'(\hat{C}^T\bar{X})\hat{C}^T\bar{X}] + 2z^T PB\hat{W}^T\Phi'(\hat{C}^T\bar{X})\tilde{C}^T\bar{X} \\ &\quad +2z^T PB(u_c + d_u + d + \epsilon). \end{aligned} \tag{3.60}$$

Note that

$$\begin{aligned} 2z^T PB(u_c + d_u + d + \epsilon) &\leq 2z^T PBu_c + 2|z^T PB||d_u| + 2|z^T PB|(d_M + \epsilon_M) \\ &\leq 2z^T PBu_c + 2|z^T PB|||C||_F||\bar{X}\hat{W}^T\Phi'(\hat{C}^T\bar{X})||_F \\ &\quad +2|z^T PB|||W||||\Phi'(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|| \\ &\quad +2|z^T PB|(d_M + \epsilon_M + 2|W|_1). \end{aligned} \tag{3.61}$$

Let $c_f = ||C||_F, c_w = ||W||$ and $c_d = d_M + \epsilon_M + 2|W|_1$. Then, applying the compensator u_c,

$$\begin{aligned} 2z^T PB(u_c + d_u + d + \epsilon) &\leq 2z^T PBu_c + 2|z^T PB|c_f||\bar{X}\hat{W}^T\Phi'(\hat{C}^T\bar{X})||_F \\ &\quad +2|z^T PB|c_w||\Phi'(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|| + 2|z^T PB|c_d \end{aligned}$$

$$\begin{aligned}
&\leq && 2|z^T PB|(\tilde{c}_f||\bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})||_F \\
& && +\tilde{c}_w||\Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|| + \tilde{c}_d) \\
&= && 2[\tilde{c}_f||\bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})||_F + \tilde{c}_w||\Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|| + \\
& && \tilde{c}_d]|z^T PB|.
\end{aligned} \tag{3.62}$$

Thus,

$$\begin{aligned}
\dot{V}_0 \leq\; & -z^Tz + 2z^TPB\tilde{W}^T[\Phi(\hat{C}^T\bar{X}) - \Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}] + 2z^TPB\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})\tilde{C}^T\bar{X} \\
& +2[\tilde{c}_f||\bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})||_F + \tilde{c}_w||\Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|| + \tilde{c}_d]|z^TPB|.
\end{aligned} \tag{3.63}$$

Note that z^TPB is a scalar. For $\dot{V}$,

$$\begin{aligned}
\dot{V} \leq\; & -||z||^2 - 2\tilde{W}^T\Gamma_w^{-1}\dot{\hat{W}} + 2\tilde{W}^T[\Phi(\hat{C}^T\bar{X}) - \Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}]z^TPB \\
& -2tr(\tilde{C}^T\Gamma_c^{-1}\dot{\hat{C}}) + 2\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})\tilde{C}^T\bar{X}z^TPB \\
& -\frac{2}{r}\tilde{c}_f\dot{\hat{c}}_f + 2\tilde{c}_f||\bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})||_F|z^TPB| - \frac{1}{r}2\tilde{c}_w\dot{\hat{c}}_w \\
& +2\tilde{c}_w||\Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|||z^TPB| - \frac{2}{r}\tilde{c}_d\dot{\hat{c}}_d + \tilde{c}_d|z^TPB| \\
\leq\; & -||z||^2 + 2\{-\tilde{W}^T\Gamma_w^{-1}\dot{\hat{W}} + [\Phi(\hat{C}^T\bar{X}) - \Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}]z^TPB\} \\
& +2tr\tilde{C}^T[-\Gamma_c^{-1}\dot{\hat{C}} + \bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})z^TPB] \\
& +2\tilde{c}_f[-\frac{1}{r}\dot{\hat{c}}_f + ||\bar{X}\hat{W}^T\Phi^{'}(\hat{C}^T\bar{X})||_F|z^TPB|] + 2\tilde{c}_w[-\frac{1}{r}\dot{\hat{c}}_w \\
& +||\Phi^{'}(\hat{C}^T\bar{X})\hat{C}^T\bar{X}|||z^TPB|] \\
& +2\tilde{c}_d[-\frac{1}{r}\dot{\hat{c}}_d + |z^TPB|].
\end{aligned} \tag{3.64}$$

Substituting the RBFNN weights' and parameters' update laws (3.51)-(3.54) into (3.64) yields

$$\dot{V} \leq -||z||^2. \tag{3.65}$$

This implies that all the signals of the closed-loop system are uniformly bounded.

From (3.56) and the fact that $\bar{X}, \Phi, \epsilon, d$ are bounded and the system parameters are bounded, it follows that $||\dot{z}||_2$ is bounded. (3.65) and the definiteness of V imply that

$$\int_0^\infty ||z|^2 dt \leq \int_0^\infty -\dot{V}(\tau) d\tau + const. \tag{3.66}$$

This implies that $||z||_2 \in L_2$. Applying Barbalat's lemma, it can be obtained that

$$\lim_{t \to \infty} ||z||_2 = 0. \tag{3.67}$$

The proof is completed.

Remark 3.1. The RBFNN reconstruction error ϵ is a critical quality to meet Theorem 2.1, representing the minimum possible deviation between the unknown function $f(x, \dot{x})$ and the function estimation $\hat{f}(x, \dot{x})$. In general, increasing the RBFNN node number reduces the RBF reconstruction error.

Remark 3.2. The proposed controller differs from the controller of Lewis *et al.* [51] and Zhang *et al.* [52]. This controller can achieve an error approaching zero, while their controllers can achieve the error approaching to a small region.

Theorem 3.1 relates only to the asymptotic performance requirement in the closed-loop system, no transient performance is discussed. In practical applications, transient performance can be even more important. To this end, the following theorem is proposed.

Theorem 3.2. For the closed-loop system (3.56), then the tracking error bound on the L_2 norm is

$$
\begin{aligned}
||z||_2 \;\; &\leq \;\; \sqrt{\lambda_{max}(P)}||z(0)|| + \frac{1}{\sqrt{\lambda_{min}(\Gamma_w)}}||\tilde{W}(0)|| + \frac{1}{\sqrt{\lambda_{min}(\Gamma_c)}}||\tilde{C}(0)|| \\
& \quad + \frac{1}{\sqrt{r}}(|\tilde{c}_f(0)| + |\tilde{c}_w(0)| + |\tilde{c}_d(0)|). \qquad (3.68)
\end{aligned}
$$

Proof. From (3.65),

$$\dot{V} \leq -||z||^2. \qquad (3.69)$$

It follows that

$$||z||^2 \leq -\dot{V}. \qquad (3.70)$$

Since $V = z^T P z + \tilde{W}\Gamma_w^{-1}\tilde{W} + \tilde{C}^T\Gamma_c^{-1}\tilde{C} + \frac{1}{r}[\tilde{c}_f^2 + \tilde{c}_w^2 + \tilde{c}_d^2]$ is non-increasing and bounded from below by zero, it will be limited as $t \to \infty$, so that

$$
\begin{aligned}
||z||^2 = \int_0^\infty ||z(\tau)||^2 d\tau \;\; &\leq \;\; -\int_0^\infty \dot{V}(\tau) d\tau \\
&= \;\; V(0) - V(\infty) \leq V(0) \\
&= \;\; z^T(0)Pz(0) + \tilde{W}(0)\Gamma_w^{-1}\tilde{W}(0) + \tilde{C}^T(0)\Gamma_c^{-1}\tilde{C}(0) \\
& \quad\;\; + \frac{1}{r}[\tilde{c}_f(0)^2 + \tilde{c}_w(0)^2 + \tilde{c}_d(0)^2] \\
&\leq \;\; \lambda_{max}(P)||z(0)||^2 + \frac{1}{\lambda_{min}(\Gamma_w)}||\tilde{W}(0)||^2 + \frac{1}{\lambda_{min}(\Gamma_c)}||\tilde{C}(0)||^2 \\
& \quad\;\; + \frac{1}{r}[\tilde{c}_f(0)^2 + \tilde{c}_w(0)^2 + \tilde{c}_d(0)^2], \qquad (3.71)
\end{aligned}
$$

which implies (3.68).

Remark 3.3. Theorem 3.2 provides some methods for improving the transient performance. 1) The non-zero initial parameter error of $W_i^* - W_i(0)$ may increase the error bound. However, with a NN trained offline, the initial error can be reduced. 2) Large adaptation gains Γ_w, Γ_c, r which attenuate the effects of initial parameter error is vital for good L_2 performance.

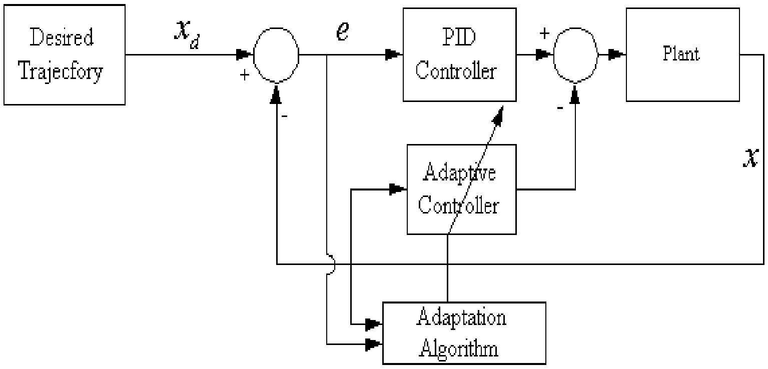


Figure 3.8: Control structure.

3.4 Simulation and Experimental Study

In this section, a simulation and experimental study is provided to demonstrate the performance of the proposed algorithms on mechanical systems. A piezoelectric linear motor (Figure 2.6) as similar to the one used in the previous chapter is used as the test platform for this experimental study. The motor is modeled as a nonlinear differential equation as given in (3.5), with $K_f = 8N/Volt$, $K_e = 144Ns/m$, $M = 5.3kg$, and $R = 1.5\Omega$.

The nominal plant model (ignoring frictional force f_{fric} in (3.12)) in (3.5) may also be expressed as

$$G_p(s) = \frac{k_p}{s(T_p s + 1)}, \tag{3.72}$$

where k_p, T_p are the model parameters.

The nominal model of the plant is first identified offline using a relay feedback method [50]. The model parameters in (3.72) are identified as $k_p = 4.3823e5$, $T_p = 0.009$. Using the dual relay feedback method ([35] and [54]), the friction parameters in (3.12) are identified as $f_1 = 0.0064$, $f_2 = 8.2876e - 5$ and $\delta f = 0.381$. Using these system parameters, a PID feedback controller can be commissioned. The controller can be designed based on a wide range of well-developed PID tuning methodologies [50]. The desired displacement trajectory is shown in Figure 3.9. A main consideration of the trajectory selected is that it has to be first-order integrable and differentiable. One of the strengths of the proposed method is that only a stable set of PID parameters (which can be easily obtained using the norminal plant model) is required. In the simulation example, only one or two relay experiments are performed to obtain the PID parameters.

For simulation purposes, three cases (i.e., *Case 1*: PID controller on the nominal plant (without considering the frictional force); *Case 2*: PID controller on the full nonlinear plant; *Case 3*: Combined PID/adaptive controller on the full nonlinear plant) are considered. The simulation results of the tracking performance in all three cases are shown in Figure 3.10. For the adaptive controller, 40 nodes are used in the RBFNN. As seen in Figure 3.10, the combined PID and adaptive controller exhibits good learning capabilities and convergence properties while performing the tracking tasks. The displacement error is much reduced when the adaptive controller is commissioned in the full nonlinear system. The effect of including the frictional force in the simulation model is evident in that the displacement error is increased, as compared to the case with just the nominal plant. The PID controller alone is not adequate for the motion tracking purposes. The adaptive controller provides the major portion of the control action.

An actual experimental study is conducted on the actual piezoelectric platform. The same desired displacement trajectory as shown in Figure 3.9 is used in this experimental study. The tracking performance of the PID controller alone is shown in Figure 3.11. With the additional adaptive controller applied to the full nonlinear system, the tracking performance of the combined PID and adaptive controller is shown in Figure 3.12.

As the desired trajectory is repetitive in nature, the forward and backward linear motion of the moveable translator along the motor stage is shown in Figures 3.11 and 3.12. With PID controller alone, the amplitude of the displacement error remains fairly constant at about 120 μm. There is just a change of sign in the displacement error as the translator moves from the forward motion and back. It can be observed that there is a step-like increase in the displacement error as the translator changes its direction of motion. This is due largely to the frictional forces [39], which includes viscous and Coulomb friction, at work. Combining the efforts of the RBFNN and PID controller, the displacement error (Figure 3.12) is greatly reduced. There is an initial spike-like increase in the displacement error as the translator changes its direction of motion. But the RBFNN is able to clamp down on the displacement error for the rest of the motion, until the translator changes its direction of motion again. For the adaptive controller, 30 nodes are used in the RBFNN in this experimental study. The results of the experimental study is consistent with that of the simulation study.

The combined PID and adaptive controller achieves good tracking performance, in the face of strong nonlinearities in the system.

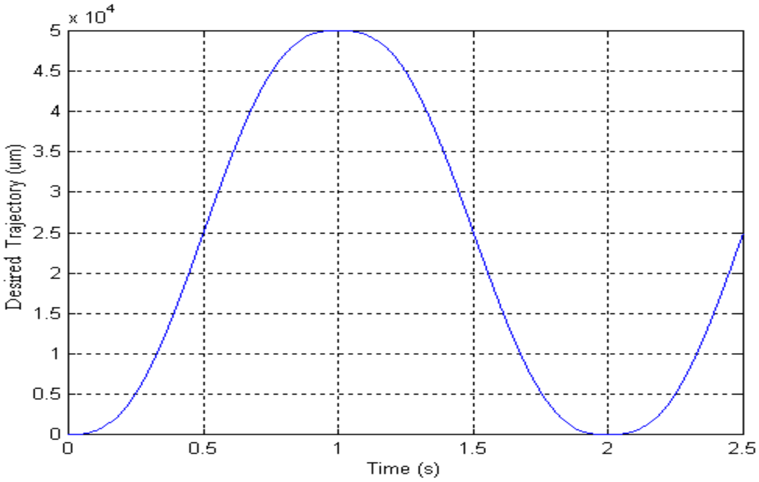


Figure 3.9: Desired trajectory.

3.5 Conclusions

This chapter has considered the development of a new PID/adaptive controller for precision motion systems. The second-order model is used as the nominal dominant model for the design of the PID controller, and an adaptive component designed based on a RBFNN provides for the possibility of performance enhancement when the feedback control alone is inadequate to cope with uncertain nonlinear phases. The effectiveness of the proposed control scheme is highlighted in the simulation and experimental study.

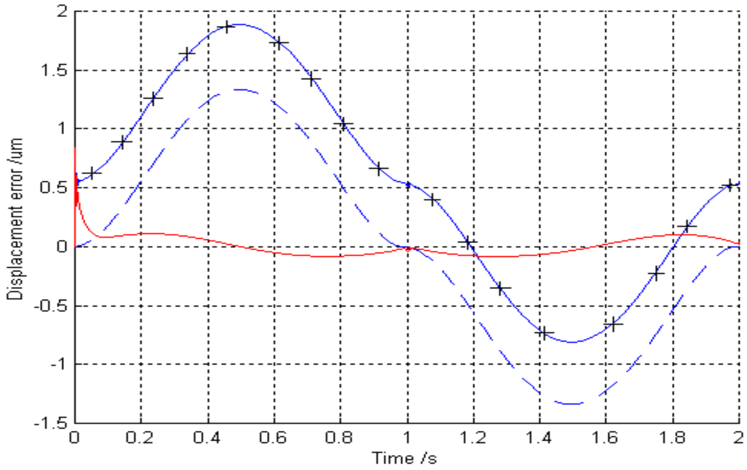


Figure 3.10: Comparison of the displacement error in all 3 cases. (*dash line*)Case 1: PID controller on the nominal plant; (+)Case 2: PID controller on the full nonlinear system; (*full line*)Case 3: Combined PID/adaptive controller on the full nonlinear system.

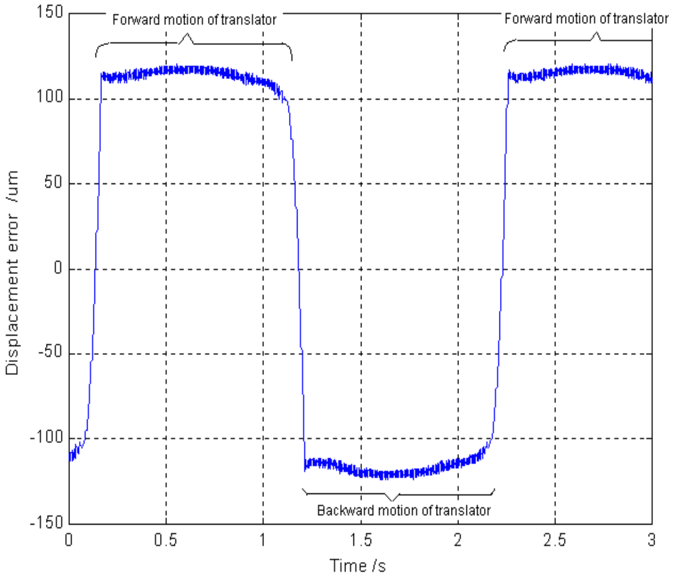


Figure 3.11: Tracking performance of the PID controller on the actual piezoelectric motor.

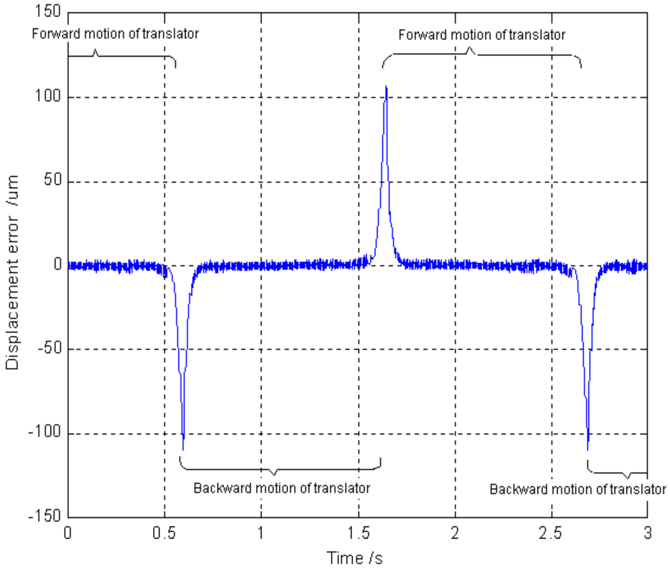


Figure 3.12: Tracking performance of the combined PID/adaptive controller on the actual piezoelectric motor.

Chapter 4

Intelligent Monitoring: Monitoring and Suppression of Vibration in Precision Motion Systems

4.1 Introduction

Mechanical vibration in machines and equipment can occur due to many factors, such as unbalanced inertia, bearing failures in rotating systems such as turbines, motors, generators, pumps, drives and turbofans, poor kinematic design resulting in a non-rigid and non-isolating support structure, component failure and/or operation outside prescribed load ratings. The machine vibration signal can be typically characterized as a narrow-band interference signal anywhere in the range from 1 Hz to 500kHz. A real-time monitoring and control device will be very useful to prevent equipment damage from the severe shaking that occurs when a machine malfunctions or vibrates at a resonant frequency. When the machine is used to perform highly precise positioning functions, undue vibrations can lead to poor repeatibility properties, impeding any effort for systematic error compensation. This results directly in a loss of achievable precision and accuracy.

This chapter addresses two approaches to deal with mechanical vibrations in precision motion systems. The first approach utilizes an adaptive notch filter (narrow-bandstop filter) to identify the resonant frequencies and suppress any signal transmission into the system at these frequencies. The adaptive notch filter can be directly incorporated into the control system. The second approach uses a real-time analyzer to detect excessive vibration based on which appropriate actions can be taken, say to provide a warning or corrective action. This second approach can be implemented independently of the control system and as such can be applied to existing equipment without modification of the normal mode of operation. A signature is derived from the vibration signal acquired using an accelerometer that is attached to the machine under normal operating conditions. A pattern recognition template is used to compare real-time vibration signal against the normal-condition signature and an alarm can be activated when the difference deviates beyond an acceptable threshold. Rectification actions can be invoked before damage is done to the machine.

4.2 Adaptive Notch Filter

The task of eliminating/suppressing undesirable narrow-band frequencies can be efficiently accomplished using a notch filter (also known as a narrow band-stop filter). The filter highly attenuates a particular frequency component and leaves the rest of the spectrum relatively unaffected. An ideal notch filter has a unity gain at all frequencies except in the so-called null frequency band, where the gain is zero. A single-notch filter is effective in removing a single frequency or a narrow-band interference; a multiple-notch filter is useful for the removal of multiple narrow-bands which would be necessary in applications requiring cancellation of harmonics. Digital notch filters are widely used to retrieve sinusoids from noisy signals, eliminate sinusoidal disturbances, and track and enhance time-varying narrow-band signals with wide-band noise. They have found extensive use in the areas of radar, signal processing, communications, biomedical engineering, and control/instrumentation systems.

To create a null band in the frequency response of a digital filter at a normalized frequency β_0, a pair of complex-conjugate zeros can be introduced on the unit circle at phase angles $\pm\beta_0$ respectively. The zeros are defined as:

$$z_{1,2} = e^{\pm j\beta_0} = \cos\beta_0 \pm j\sin\beta_0, \tag{4.1}$$

where the normalized null frequency β_0 is defined as:

$$\beta_0 = 2\pi\frac{f_0}{f_s}. \tag{4.2}$$

Note that f_s is the sampling frequency in Hz (or rad/s) and f_0 is the notch frequency in Hz (or rad/s). This yields a *Finite Impulse Response* (FIR) filter given by the z-transform transfer function:

$$H(z) = 1 - 2\cos\beta_0 z^{-1} + z^{-2}. \tag{4.3}$$

A FIR notch filter has a relatively large notch bandwidth, which means that the frequency components in the neighbourhood of the desired null frequency are also severely attenuated as a consequence. The frequency response can be improved by introducing a pair of complex-conjugate poles. The poles are placed inside the circle with a radius of α at phase angles $\pm\beta_0$. The poles are defined as

$$p_{1,2} = \alpha e^{\pm j\beta_0} = \alpha(\cos\beta_0 \pm j\sin\beta_0), \tag{4.4}$$

where one requires $\alpha \leq 1$ for filter stability, and $(1-\alpha)$ is the distance between the poles and the zeros.

The poles introduce a resonance in the vicinity of the null frequency, thus reducing the bandwidth of the notch. The transfer function of the filter is given by:

$$H(z) = \frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)}. \tag{4.5}$$

Substituting the expression for z_i and p_i, and dividing throughout by z^2, the resulting filter has the following transfer function:

$$\begin{aligned} H(z) &= \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}, && (4.6) \\ &= \frac{1 - 2\cos\beta_0 z^{-1} + z^{-2}}{1 - 2\alpha\cos\beta_0 z^{-1} + \alpha^2 z^{-2}}. && (4.7) \end{aligned}$$

Digitally, the filtered signal y is thus obtained from the raw signal u via the recursive formula in the discrete time domain as follows:

$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) - b_1 y(n-1) - b_2 y(n-2), \tag{4.8}$$

where the coefficients a_i and b_i are the same as those in (4.6) because z^{-1} corresponds to the time-shift (delay through sampling period) operator.

The bandwidth and the Q-factor of the notch filter are respectively given by Ferdjallah *et al.* [55]:

$$BW = \frac{2\sqrt{2}(1-\alpha^2)}{[16-2\alpha(1+\alpha)^2]^{\frac{1}{2}}}, \tag{4.9}$$

$$Q = \omega_0 \frac{[16-2\alpha(1+\alpha)^2]^{\frac{1}{2}}}{2\sqrt{2}(1-\alpha^2)}. \tag{4.10}$$

The filter transfer function $H(z)$ has its zeros on the unit circle. This implies a zero transmission gain at the normalized null frequency β_0. It is interesting to note that the filter structure (4.7) allows independent tuning of the null frequency and the 3-dB attenuation bandwidth by adjusting β_0 and α, respectively. The performance of the notch filter depends on the choice of the constant α, which controls the bandwidth BW according to (4.9). The bandwidth, which is a function of the distance of the poles and zeros $(1-\alpha)$, narrows when α approaches unity. Clearly, when α is close to 1, say $\alpha = 0.995$, the corresponding transfer function behaves virtually like an ideal notch filter.

Complete narrow-band disturbance suppression requires an exact adjustment of the filter parameters to align the notches with the resonant frequencies. If the true frequency of the narrow-band interference that is to be rejected is stable and known *a priori*, a notch filter with fixed null frequency and fixed bandwidth can be used. However, if no information is available *a priori* or when the resonant frequencies drift with time, the fixed notch may not coincide exactly with the desired null frequency, particularly if the bandwidth is too narrow (i.e. $\alpha \approx 1$). In this case, a tunable/adaptive notch filter is highly recommended. In [56] and [57], it is proposed to adapt the null bandwidth of the filter to accommodate the drift in frequency. In [58], it is suggested that an active compensator be used to suppress the vibration signals. Kwan and Martin [59] adapt the null frequency β_0, while keeping the pole radii α constant. In other words, the parameters a_1 and b_1 of (4.6) are adjusted such that the notch will center at the unwanted frequency while retaining the null bandwidth of the notch filter.

4.2.1 Fast Fourier Transform (FFT)

The *Discrete Fourier Transform* (DFT) is a tool that links the discrete-time domain to the discrete-frequency domain. It is a popular off-line approach widely used to obtain the information about the frequency distribution required for the filter design. However, the direct computation of DFT is prohibitively expensive in terms of required computation effort. Fortunately, FFT is mathematically equivalent to DFT, but it is a more efficient alternative for implementation purposes (with a computational speed that is exponentially faster) and can be used when the number of samples n is a power of two (which is not a serious constraint). For vibration signals where the concerned frequencies drift with time, FFT can be continuously applied to the latest n samples to update the signal spectrum. Based on the updated spectrum, the filter characteristics can be continuously adjusted for notch alignment. The block diagram of the adaptive notch filter which has been developed in the present work, with its adjusting mechanism, is shown in Figure 4.1.

4.2.2 Simulation

There are many issues to consider in designing the notch filter. One of the main consideration is system stability. The introduction of the notch filter should not affect the overall stability of the system. Other considerations include filter consistency, i.e., filter performance should not degrade when the filter coefficients change slightly as

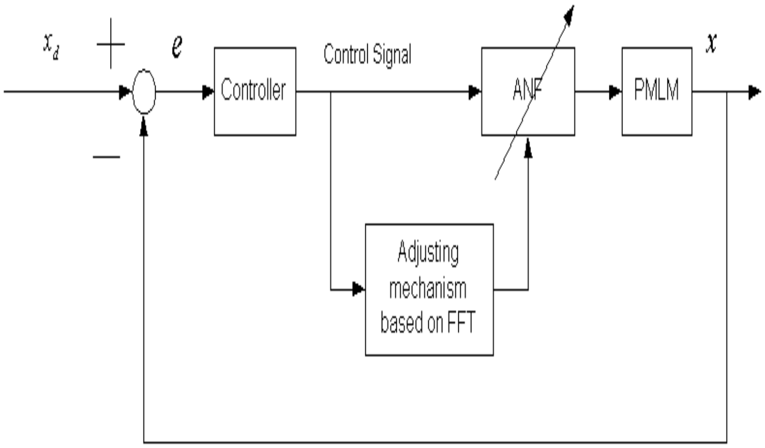


Figure 4.1: Block diagram of the adaptive notch filter with adjusting mechanism.

a result of quantization and noise in the signal should not hurt the performance of the filter, and filter performance, i.e., the filter should provide sufficient signal discrimination or noise cancellation. In the case of the simulation and experimental study here, the PID controller has to be retune to with the introduction of the notch filter. It is shown in the simulation and experimental results that the introduction of the notch filter does not affect the overall system stability.

Computer simulations have been carried out to explore the application of the adaptive notch filter in suppressing undesirable frequency transmission in the control system for a precision positioning system that uses permanent magnet linear motors (PMLM). In one simulation, a sinusoidal trajectory is closely followed and an undesirable vibration signal is simulated, which drifts from a frequency of 500Hz in the first cycle to a frequency of $1-5Hz$ in the second cycle of the trajectory. Figure 4.2 shows the tracking performance of the precision machine without a notch filter. Figure 4.3 shows the performance when a fixed notch filter is used and Figure 4.4 shows the performance with an adaptive notch filter. It is clearly evident that a time-invariant

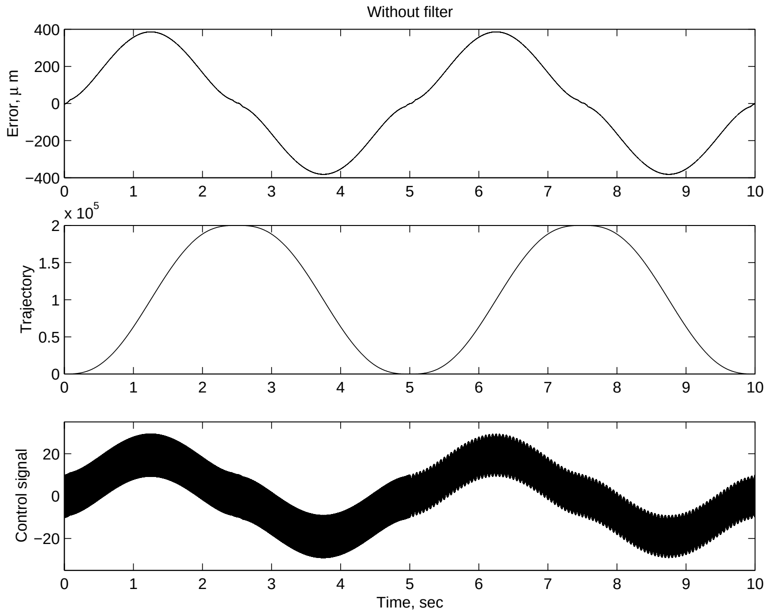


Figure 4.2: Simulation results without a notch filter: (a) Error (μm); (b) Desired trajectory (μm); (c) Control signal (V).

narrow-band vibration signal can be effectively eliminated using just a fixed notch filter. However, when the vibration frequencies drift, an adaptive notch filter is able to detect the drift and align the notch to remove the undesirable frequencies with only a short transient period.

4.2.3 Experiments

A notch filter is implemented in the control system of a Linear Drive tubular linear motor (LD3810) equipped with a Renishaw optical encoder having an effective resolution of $1\mu m$. The popular PID control is used in the controller. Figure 4.5 shows the performance of the PMLM when no filter is used. Figure 4.6 shows the improvement in the control performance when the notch filter is incorporated into the control system. Again, it is noted how a notch filter can be used in a practical

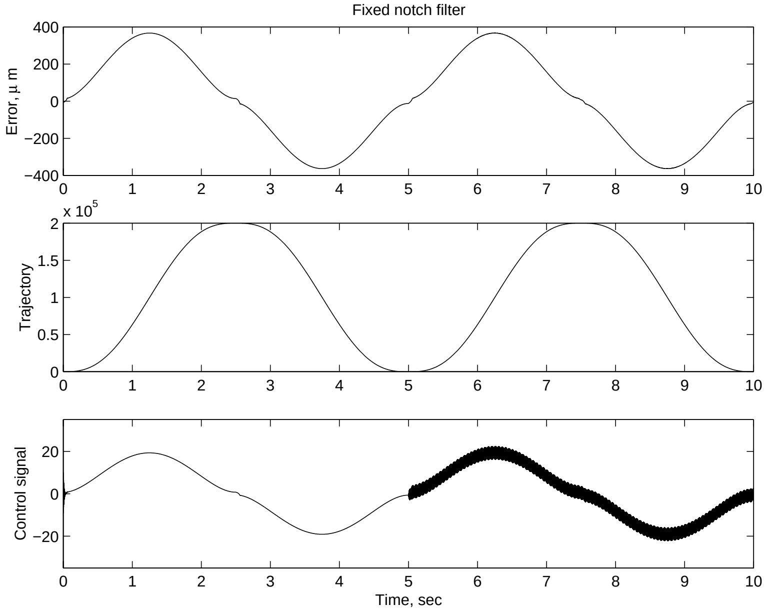


Figure 4.3: Simulation results using a fixed notch filter: (a) Error (μm); (b) Desired trajectory (μm); (c) Control signal (V).

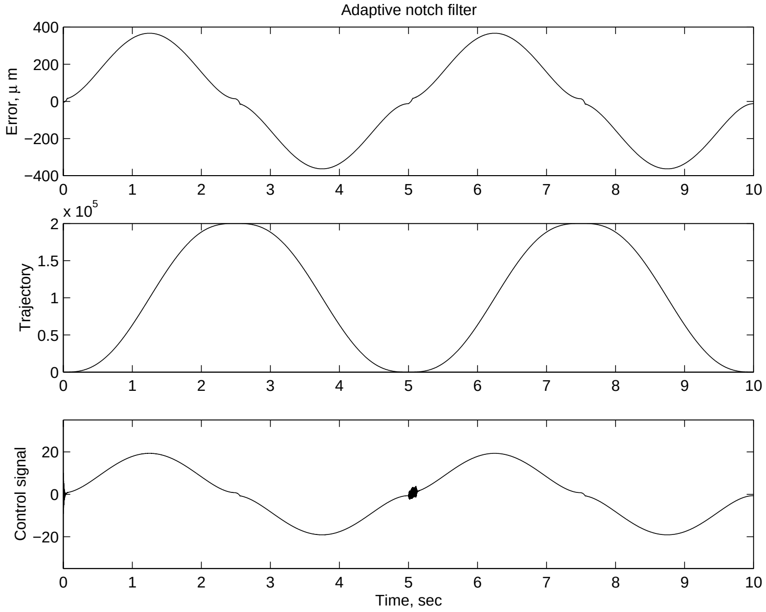


Figure 4.4: Simulation results using an adaptive notch filter: (a) Error (μm); (b) Desired trajectory (μm); (c) Control signal (V).

situation of motion control.

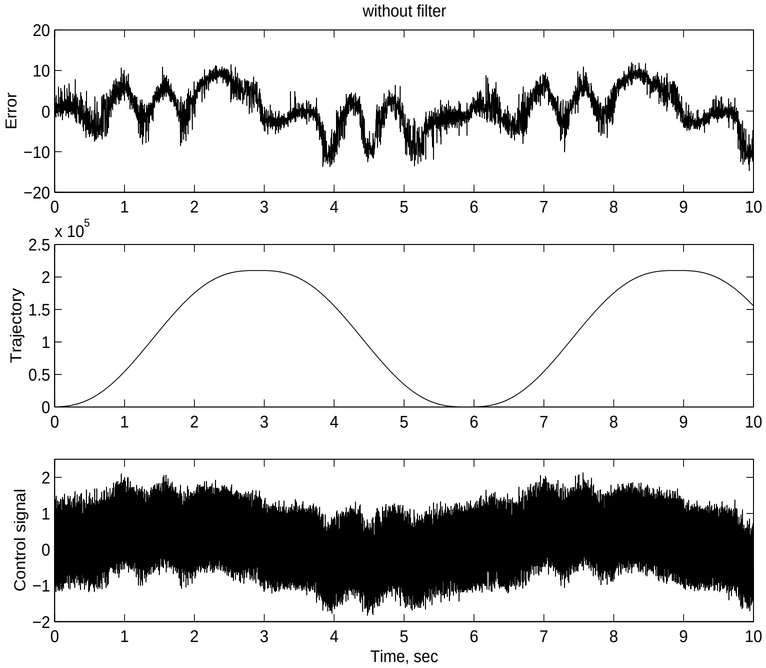


Figure 4.5: Experimental results without a notch filter: (a) Error (μm); (b) Desired trajectory (μm); (c) Control signal (V).

4.3 Real Time Vibration Analyzer

Another approach towards real time monitoring and analysis of machine vibration ([60] and [61]) is described in this section. The main idea behind this approach is to construct a vibration signature based on pattern recognition of ‘acceptable’ or ‘healthy’ vibration patterns. The vibration analyzer can operate in three modes: learning, monitoring, or diagnostic. The learning mode, to be initiated first, will yield a set of vibration signatures based on which the monitoring and diagnostic modes will operate. In the monitoring mode, with the machine under normal closed-loop control, the analyzer only uses a naturally occuring vibration signal to deduce the condition

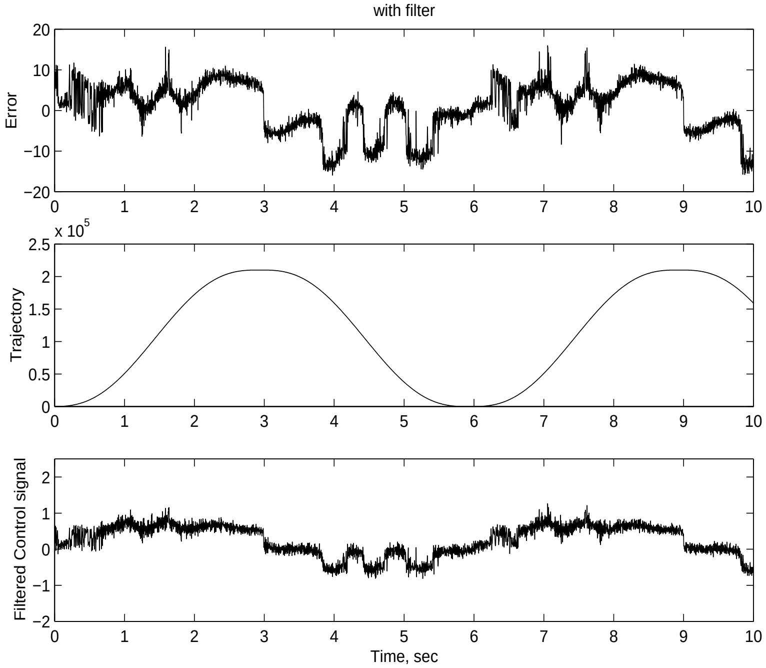


Figure 4.6: Experimental results using a notch filter: (a) Error (μm); (b) Desired trajectory (μm); (c) Control signal (V).

of the machine. No test excitation is deliberately added to the input signal of the machine. More than one criterion may be used in the evaluation of the condition of the machine, and in which case, a fusion approach would generate a combined output (machine condition) based on the multiple inputs. In the diagnostic mode, explicit input signals are applied to the machine and the output signal (vibration) is logged for analysis with respect to the associated vibration signature. In what follows, the details of the various components/functions of the analyzer will be described systematically.

The block diagram of the real-time vibration analyzer system that has been developed in the present work is shown in Figure 4.7. It consists of an accelerometer, which is mounted on the machine to be monitored. The accelerometer measures a multi-frequency vibration signal and transmits it to an intelligent DSP module, after performing appropriate signal conditioning. This module can be a standalone device, or one integrated to a PC host. The vibration analysis algorithm is downloaded to this DSP module. With this algorithm, it can now be established as to whether the condition of the machine is within a pre-determined acceptable threshold. If the condition is determined to be poor, the DSP module will trigger an alarm to the operator who would enable a corrective action, or automatically activate a corrective action (e.g., change the operating conditions of the machine, modify the paramenters of the controller or shut down the machine).

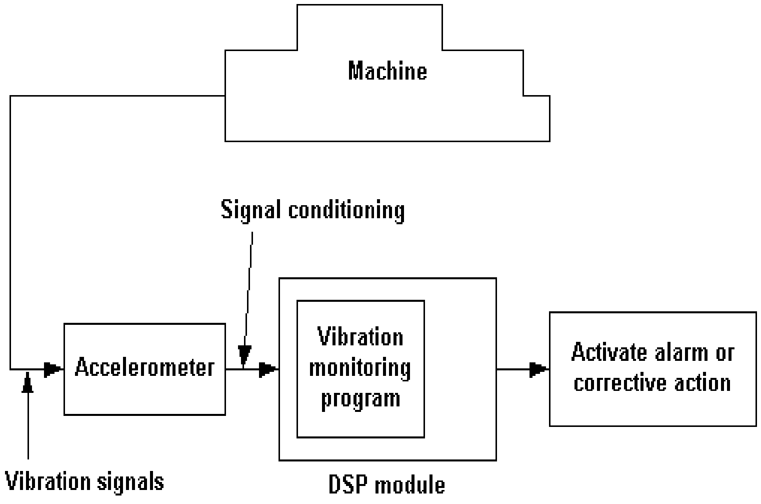


Figure 4.7: Schematic diagram of the real-time vibration analyzer.

The construction of the real-time vibration analyzer is inexpensive and requires only commercially available, low cost components. The installation can be easy and simple, as the accelerometer is able to gather vibration signals, independent of the machine's own control system. Thus, there is no need to disrupt the operation of the machine. In the prototype reported here, a DSP emulator board (TMS320C24x model) [62] from Texas Instruments is used as the standalone DSP module. This C24x series emulator board is built around the F240 DSP controller, operating at 20MIPS with an instruction cycle time of 50ns. It is optimized for digital motor control and conversion applications. Other key components supported on this DSP module are ADCs, dual access RAM (DARAM), on-chip flash memory and RS-232 compatible serial port. The vibration analysis algorithm, which is described in the sequel, will be downloaded to the DSP board after satisfactory evaluation and tests on the PC. This DSP module and the accelerometer unit (with signal conditioning) constitute the only hardware requirements of the real time vibration analyzer (see Figure 4.7).

4.3.1 Learning Mode - Extracting the Vibration Signature

In the learning mode, the vibration signals, with the machine operating under normal conditions, are acquired by the accelerometer and stored in the DSP module. A suitable vibration signature [63] is then extracted from the vibration signals. There are many types of vibration signatures that are adequate for the purpose of machine monitoring. For example, one form of vibration signature may be based on the amplitude of the vibration; another form may use a time series analysis of the vibration; yet another form may employ the spectrum of the vibration which can be efficiently obtained using the FFT algorithm. Regardless of the type, these vibration signatures are dependent on the nature of the input signals driving the machine. For example, a square wave input will produce a vibration spectrum which can be quite different from that resulting from an input of a chirp signal (i.e., repeating sine wave of increasing frequency) or a pure sinusoid. Thus, a particular input signal will produce a unique spectrum based on which a unique vibration signature can be derived. Multiple vibration signatures corresponding to the natural vibrations of the machine (useful for the monitoring mode), or corresponding to different input signals (useful for the diagnostic mode) can thus be captured for subsequent diagnosis and monitoring of the machine.

4.3.2 Monitoring Mode

In the monitoring mode, the vibration signals are sampled periodically from the machine to monitor the condition of the machine. No deliberate or additional input signal is required, so the machine operation is not disrupted. The updated spectra are analyzed against the relevant vibration signatures. The analysis and comparison may be done in terms of the shift in frequency or amplitude of the spectrum, or a combination of the two. For example, one evaluation criterion (`EV`) may be based on the mean-square (ms) value of the error [63] between the current real-time vibration spectrum and the vibration signature:

$$\mathtt{EV_1} = \frac{\sum_{q=1}^{N}(S_q - S_q^*)^2}{M}, \tag{4.11}$$

where S_q is the discretized current real-time vibration spectrum, S_q^* is the corresponding vibration signature, q is the index for the data points, and M is the total number of data points. Another `EV` may be formulated based on the difference in the amplitude of the current time series vibration pattern and its corresponding vibration signature:

$$\mathtt{EV_2} = \frac{\max|T_q| - \max|T_q^*|}{M}, \tag{4.12}$$

where $\max|T_q|$ represents the highest amplitude of the current time series vibration pattern T_q, $\max|T_q^*|$ is the highest amplitude of its corresponding vibration signature and $|.|$ is the modulus operator. The amplitude is normalised here to be consistent with the other EVs.

More than one evaluation criterion may be used in the determination of the machine condition. In this case, a fusion technique would be necessary. The key idea of fusion is to associate the machine with a `HEALTH` attribute which is computed from multiple evaluation criteria. These criteria are expected to influence, to a varying degree, the `HEALTH` of the machine. The `HEALTH` attribute is thus an appropriate function $\Im$ of the various criteria ($\mathtt{EV_i}$s); i.e.,

$$\mathtt{HEALTH} = \Im(\mathtt{EV_1}, \mathtt{EV_2}, ..., \mathtt{EV_n}), \tag{4.13}$$

where n refers to the number of criteria being evaluated.

A fuzzy weighted approach may be used to realize the $\Im$ function as follows:

The `HEALTH` attribute is treated as a fuzzy variable (i.e., $\mathtt{HEALTH} \in [0, 1]$). $\mathtt{HEALTH} = 0$ will represent absolute machine failure while $\mathtt{HEALTH}= 1$ represents a perfectly normal machine condition. This attribute may be computed from a fuzzy operation on a combination of the evaluation criteria ($\mathtt{EV_i}$s) obtained via an analysis of the vibration signals against their signatures. The final decision on the condition of the machine will be derived from the `HEALTH` attribute. Zadeh [64] provides a comprehensive review on fuzzy logic.

A Takagi and Sugeno [65] type of fuzzy inference is used in this chapter. Consider the following p rules governing the computation of an attribute:

$$\textbf{IF}\ \ \mathtt{EV^i_1}\ \ \textbf{IS}\ \ \mathtt{F^i_1} \otimes ... \otimes \mathtt{EV^i_n}\ \ \textbf{IS}\ \ \mathtt{F^i_n}\ \ \textbf{THEN}\ \ u^i = \alpha^i, \quad i = 1...p. \tag{4.14}$$

where $u^i \in (0, 1]$ is a crisp variable output representing the extent to which the ith evaluation rule affects the final outcome. Thus, α_i represents the weight of the ith rule, with $\sum_i \alpha^i = 1$, $\mathtt{F^i_j}$ represents the fuzzy sets in which the input linguistic variables ($\mathtt{EV_i}$s) are evaluated, and $\otimes$ is a fuzzy operator which combines the antecedents into premises.

The value of the attribute is then evaluated as a weighted average of the u^is:

$$\mathtt{HEALTH} = \frac{\sum_{i=1}^{p} \omega^i u^i}{\sum_{i=1}^{p} \omega^i}, \tag{4.15}$$

where the weight ω^i implies the overall truth value of the premise of rule i for the input and it is computed as:

$$\omega^i = \Pi_{j=1}^{n} \mu_{\mathtt{F^i_j}}(\mathtt{EV^i_j}). \tag{4.16}$$

where $\mu_{\mathtt{F^i_j}}(\mathtt{EV^i_j})$ is the membership function for the fuzzy set $\mathtt{F^i_j}$ related to the input linguistic variable $\mathtt{EV^i_j}$ (for the ith rule). For example, in this application, the evaluation criterion ($\mathtt{EV_i}$) may be the maximum error (`MAX_ERR`) and $\mathtt{F^i_j}$ may be the fuzzy set HIGH.

The membership function $\mu_{\texttt{HIGH}}(\texttt{MAX\_ERR})$ may have the characteristic shown in Figure 4.8. The decision as to whether any corrective action that might be necessary could then be based on a simple **IF-THEN-ELSE** formulation as follows:

$$\textbf{IF } \texttt{HEALTH} \leq \gamma, \textbf{ THEN } \texttt{STRATEGY=TRIGGER ALARM}$$
$$\textbf{ELSE } \texttt{STRATEGY=CONTINUE TO MONITOR}.$$

Here γ is interpreted as a threshold value. Suitable values for γ may be in the range $0.6 \leq \gamma \leq 0.9$. Here, STRATEGY is stated to trigger an alarm to the operator who will enable a corrective action, or automatically activate a corrective action (e.g., change the operating conditions of the machine, modify the paramenters of the controller or shut down the machine).

Under this framework, it is relatively easy to include additional criteria for analysis and decision making on the system. The procedure will involve setting up of the membership functions for the criterion, formulating the additional fuzzy rules required, and adjusting the scaling parameters (the α terms in (4.14)) to reflect the relative weight of the new criterion over the existing ones. In this manner in the monitoring mode, foreboding trends can often be spotted long before the vibration reaches a level that is seriously detrimental to the machine.

4.3.3 Diagnostic Mode

In the diagnostic mode, the current vibration signal corresponding to each input signal (with standardized amplitude and frequency) is analyzed against the associated

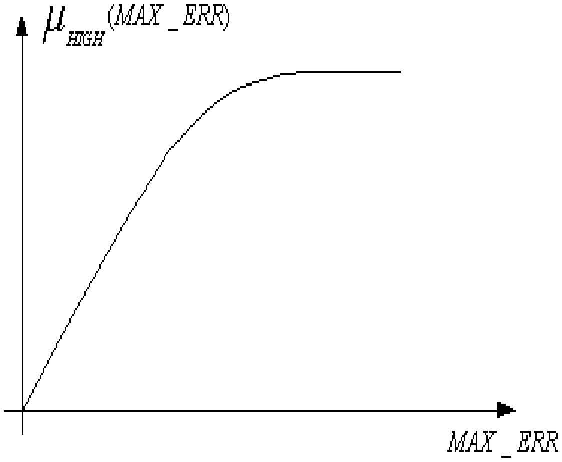


Figure 4.8: Membership function for the the input MAX_ERR, $\mu_{\texttt{HIGH}}(MAX\_ERR)$.

signature (obtained earlier in the learning mode), depending on the type of machine. Similar to the monitoring mode, there can be multiple evaluation criteria to be used in the diagnostic mode, so that the fusion technique described earlier is also applicable.

The input signals applied to the machine have to be designed carefully so as to yield as much information of the machine condition as possible in the operational regime of interest. Two important considerations are in the choice of amplitude and frequency.

Machines may have constraints in relation to the amount of travel that is possible. Too large an amplitude for the input signal may be not be viable for the machine due to the limit of travel or may even damage the machine. Also, the frequency range of the input should be chosen so that it has most of its energy in the frequency bands that are important for the system. Where input signals cannot be applied to the system in the open-loop, the setpoint signal will serve as the input for the closed-loop system since it may not be possible to directly access the system under closed-loop control. Careful considerations of the mentioned issues will ensure that significant information can be obtained from the machine.

In this chapter, the input signals considered are square wave input (Figure 4.9), chirp input (Figure 4.11) and sine wave input (Figure 4.13), standardized in amplitude to 1V and frequency to 5 Hz. The corresponding vibration signatures are shown in figures 4.10, 4.12 and 4.14, respectively.

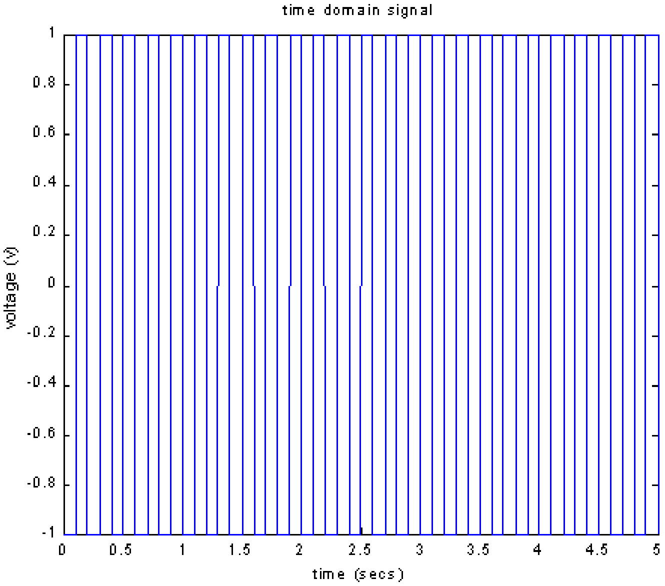


Figure 4.9: Square wave input, with standardized amplitude of 1V and frequency of 5Hz.

4.3.4 Experiments

A shaker table (Figure 4.15) is used as the test platform for the experiments presented here. The shaker table can be used to simulate machine vibrations and evaluate the performance of for example, active inertial dampers. The shaker table is driven by a high torque direct drive motor (which has a maximum torque of 1.11Nm, a maximum design load of 11Kg and generates a maximum force of 175N). The maximum linear

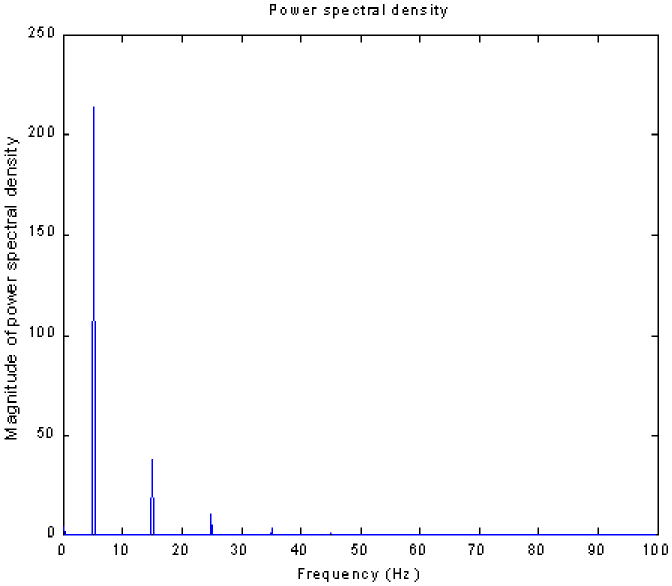


Figure 4.10: Vibration signature of the square wave input, with standardized amplitude of 1V and frequency of 5Hz.

travel of the table is +/- 2cm.

The learning mode is first initiated to obtain the vibration signals with the shaker table operating under normal conditions. It is assumed in the experiments that the normal condition corresponding to the input is a square wave signal (with standardized amplitude of 1V and frequency of 5Hz). For the purpose of implementing the diagnostic mode, the vibration signals are also obtained for the input signals of the sinusoidal and chirp type, with standardized amplitude of 1V and frequency of 5Hz (See figures 4.9 through 4.14 for the inputs and their corresponding vibration signatures).

A. Input Variables - Evaluation Criteria

Different types of `EVs` can be used as input variables for the determination of the machine condition. For the present vibration analysis application, the input variables

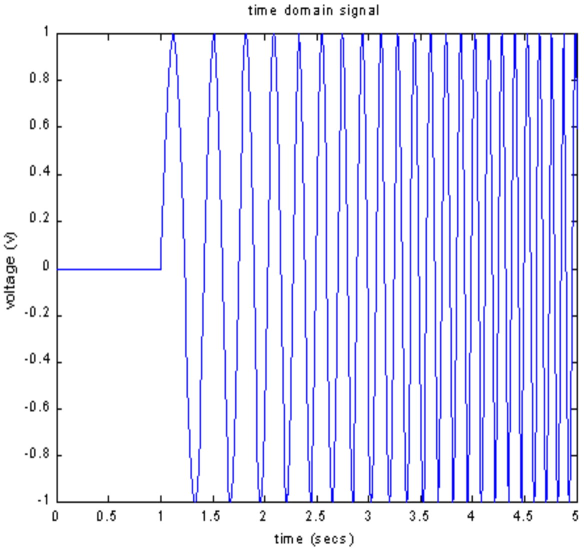


Figure 4.11: Chirp wave input, with standardized amplitude of 1V and starting frequency of 5Hz.

chosen for the computation of the `HEALTH` attribute are:

A.I. Monitoring Mode

$$\begin{aligned}\texttt{EV}_1 &= \frac{\sum_{q=1}^{N}(S_{sq,q}-S^*_{sq,q})^2}{M},\\ \texttt{EV}_2 &= \frac{(\max|T_{sq,q}|-\max|T^*_{sq,q}|)^2}{M},\\ \texttt{EV}_3 &= \frac{\sum_{q=1}^{M}(T_{sq,q}-T^*_{sq,q})^2}{M},\end{aligned} \tag{4.17}$$

where $S_{sq,q}$ and $T_{sq,q}$ represent the vibration spectrum and the time-domain signal, respectively, corresponding to a square wave input, and M is the number of time series data points over an operational cycle. Hence, $\texttt{EV}_1$ refers to the mean-square deviation between the vibration spectrum and its signature, $\texttt{EV}_2$ refers to the square of the difference between the amplitude of the vibrational signal over one operational cycle compared to its signature, and $\texttt{EV}_3$ refers to the mean-square deviation between

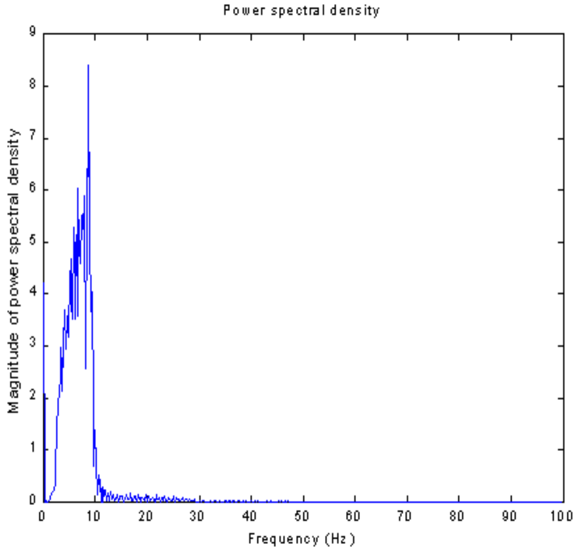


Figure 4.12: Vibration signature of the chirp wave input, with standardized amplitude of 1V and starting frequency of 5Hz.

the vibration signal and its signature (time domain) over one operational cycle.

A.II. Diagnostic Mode

$$\begin{aligned}\mathtt{EV}_4 &= \frac{\sum_{q=1}^{N}(S_{sq,q}-S^*_{sq,q})^2}{M},\\ \mathtt{EV}_5 &= \frac{\sum_{q=1}^{N}(S_{cp,q}-S^*_{cp,q})^2}{M},\\ \mathtt{EV}_6 &= \frac{\sum_{q=1}^{M}(S_{sn,q}-S^*_{sn,q})^2}{M}.\end{aligned} \tag{4.18}$$

Here *cp* denotes a chirp input signal and *sn* denotes a sine input signal.

For the monitoring mode, the input attributes are related only to the square input, due to the assumption that the input signal, under normal operating conditions, is the square wave signal (with standardized amplitude of 1V and frequency of 5Hz).

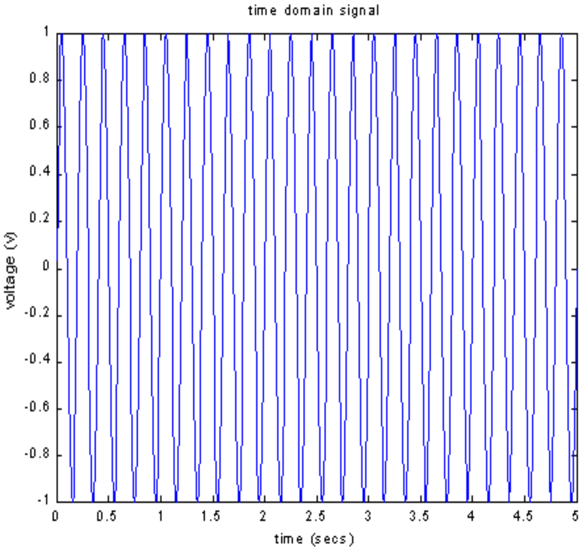


Figure 4.13: Sine wave input, with standardized amplitude of 1V and frequency of 5Hz.

B. Evaluation Rules

The three rules for the computation of the `HEALTH` attribute are:

B.I. Monitoring Mode

IF EV_1 **IS** `LOW`, **THEN** u=μ_1,

IF EV_2 **IS** `SHORT`, **THEN** u=μ_2,

IF EV_3 **IS** `LOW`, **THEN** u=μ_3.

The values of the scaling parameters, i.e., α terms in (4.14), reflect the relative importance of the fuzzy rules in the determination of the `HEALTH` of the machine. The scaling values used are:

$$\alpha_1 \;=\; 0.7,$$

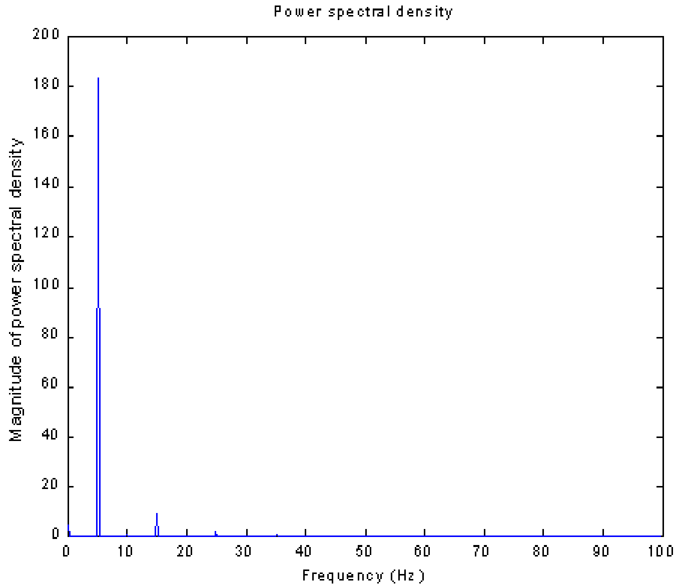


Figure 4.14: Vibration signature of the sine wave input, with standardized amplitude of 1V and frequency of 5Hz.

$$\alpha_2 = 0.2,$$

$$\alpha_3 = 0.1.$$

The respective membership functions are:

$$\mu_i(\mathtt{EV_i}) = e^{-n(\mathtt{EV_i})^{\beta}}, i = 1...6.$$

where n and β are scaling factors for normalization of EV_i. In this application, they are selected to be $n = 10$ and $\beta = 0.5$.

B.II. Diagnostic Mode

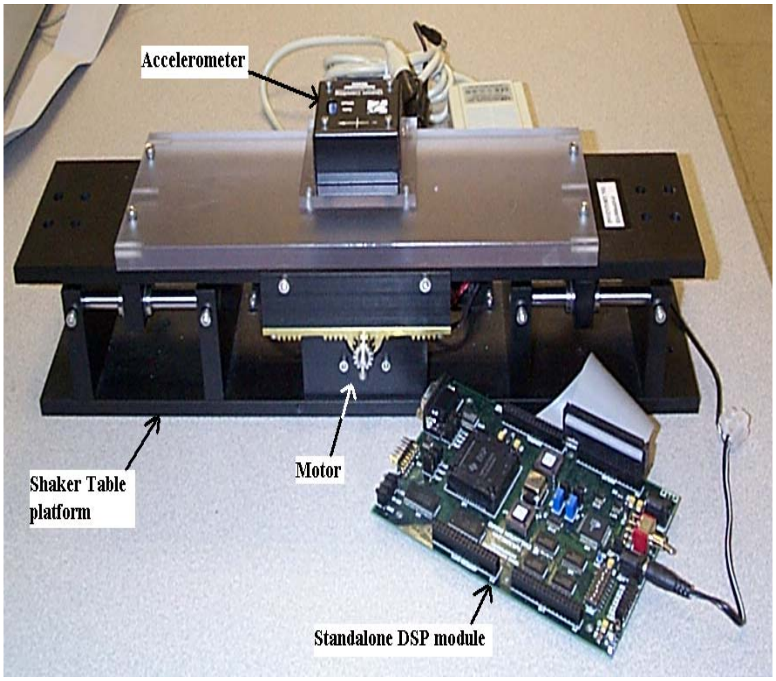


Figure 4.15: Test platform: the shaker table.

IF EV_4 **IS** `LOW`, **THEN** u=μ_4,

IF EV_5 **IS** `LOW`, **THEN** u=μ_5,

IF EV_6 **IS** `LOW`, **THEN** u=μ_6.

The scaling values used are:

$$\begin{aligned}\alpha_4 &= 0.4,\\ \alpha_5 &= 0.2,\\ \alpha_6 &= 0.4.\end{aligned}$$

Similar membership functions are used as for the monitoring mode.

The machine condition attribute `HEALTH` is then computed as in (4.15).

C. Tests

C.I. Monitoring Mode

In the monitoring mode, the normal input signal (i.e., the square wave with standardized amplitude of 1V and frequency of 5Hz) is applied to the shaker table system. At $t = 5s$, a sinusoidal signal (with amplitude 0.4V and frequency $f = 5Hz$) is also applied to the system to simulate a fault arising in the machine. The time domain signal of the machine (corresponding to the square input) is shown in Figure 4.16. The spectrum of the machine before and after $t = 5s$ are shown in figures 4.17 and 4.18, respectively. The vibration analysis algorithm is able to detect the fault in the machine. Before the introduction of the fault, the `HEALTH` attribute of the shaker table is found to be 0.98. After the introduction of the fault, the `HEALTH` attribute falls to 0.63 which is below the threshold value (which is set at 0.7). As a result, the alarm is triggered.

C.II. Diagnostic Mode

In the diagnostic mode, three input signals (i.e., sine, square, and chirp wave with standardized amplitude and frequency) are selected to be applied to the shaker table system. To simulate a fault arising at $t = 5s$, the input gain is increased by a factor of 1.4 times at $t = 5s$. The time domain vibration signal of the machine (corresponding to the chirp signal, with standardized amplitude of 1V and starting frequency of 5Hz) is shown in Figure 4.19. The spectrum (corresponding to the chirp signal) of the machine before and after $t = 5s$ are shown in figures 4.20 and 4.21, respectively. The time domain vibration signal of the machine (corresponding to the sinusoidal wave input, with standardized amplitude of 1V and frequency of 5Hz) is shown in Figure 4.22. The spectrum (corresponding to the sinusoidal input) of the machine before and after $t = 5s$ are shown in figures 4.23 and 4.24, respectively.

The vibration analysis algorithm is able to detect the fault in the machine. Before the introduction of the fault, the `HEALTH` attribute of the shaker table is found to be about 0.97. After the introduction of the fault, the `HEALTH` attribute falls to 0.58 which is below the threshold value (which is set at 0.7). The alarm is triggered as a result.

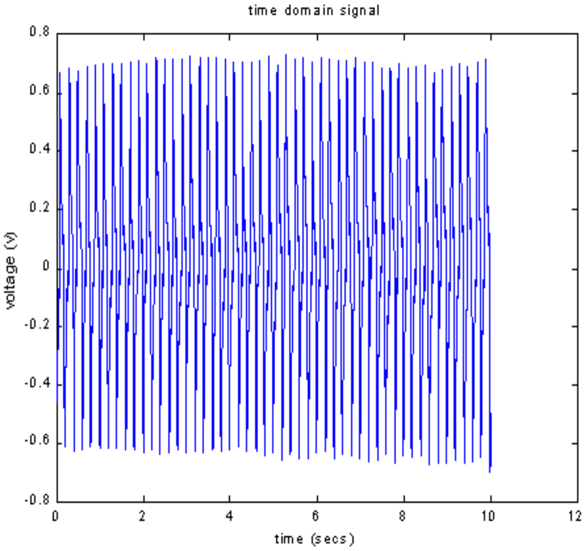


Figure 4.16: Time domain vibration signal corresponding to the square input, with standardized amplitude of 1V and frequency of 5Hz (at t=5s, a fault is simulated).

4.3.5 Remote Monitoring and Control

Knowledge-Based (KB) control has become an important approach towards the realization of intelligent control and expert systems. It has found wide ranging applications, including robot path planning ([66] and [67]), process supervision [68],

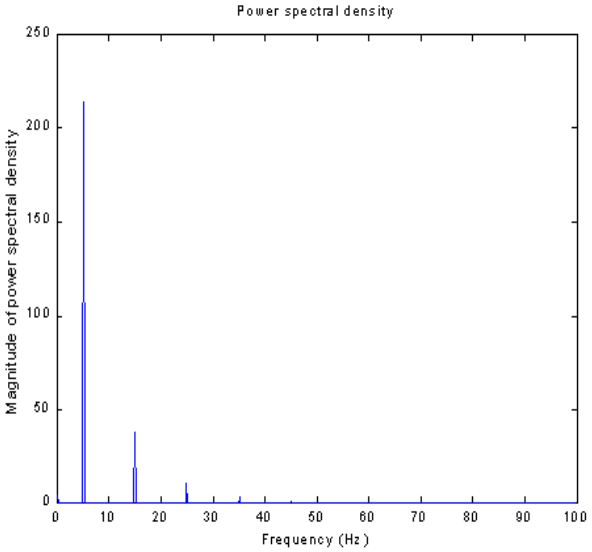


Figure 4.17: Vibration signature corresponding to the square input, with standardized amplitude of 1V and frequency of 5Hz.

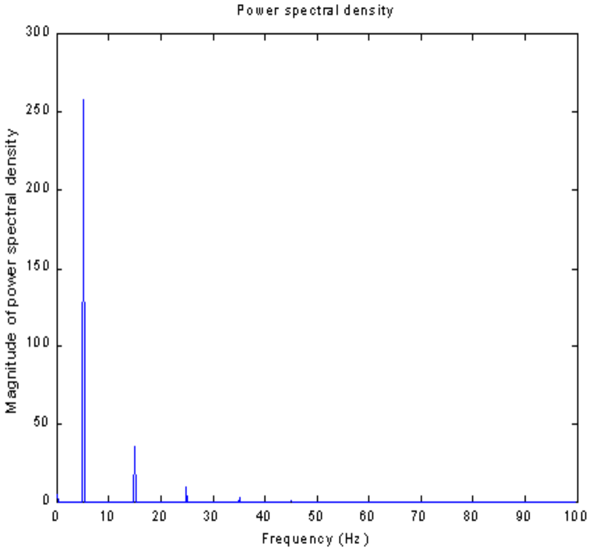


Figure 4.18: Spectrum of machine corresponding to the square input (with standardized amplitude of 1V and frequency of 5Hz) after fault occurs.

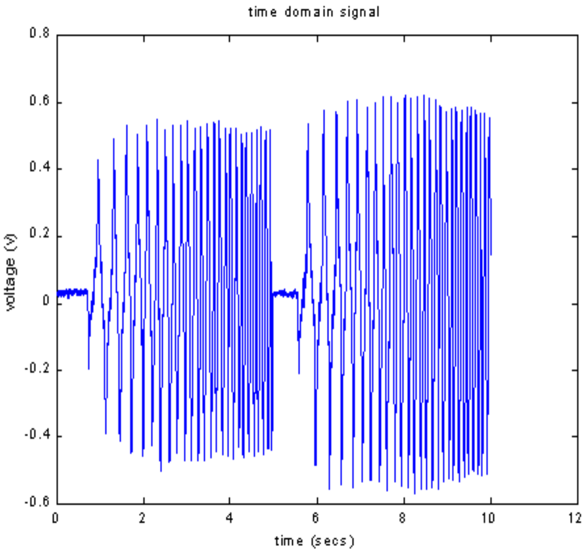


Figure 4.19: Time domain vibration signal corresponding to the chirp input, with standardized amplitude of 1V and starting frequency of 5Hz (at t=5s, a fault is simulated).

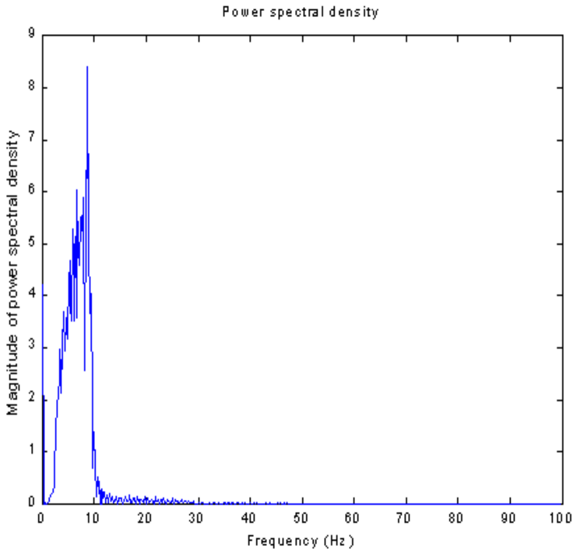


Figure 4.20: Vibration signature corresponding to the chirp input, with standardized amplitude of 1V and starting frequency of 5Hz.

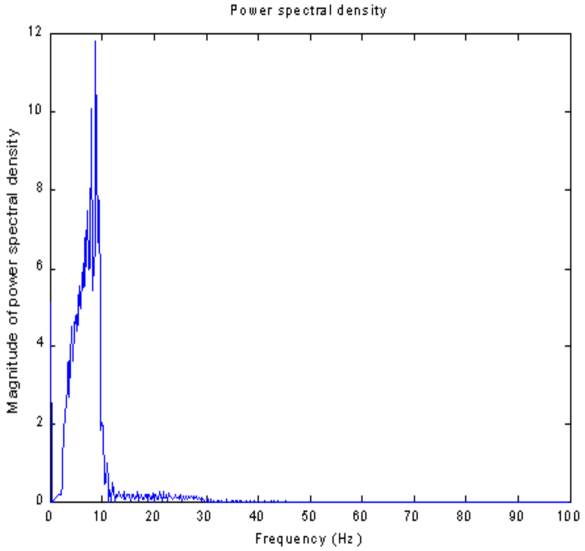


Figure 4.21: Spectrum of machine corresponding to the chirp input (with standardized amplitude of 1V and starting frequency of 5Hz) after fault occurs.

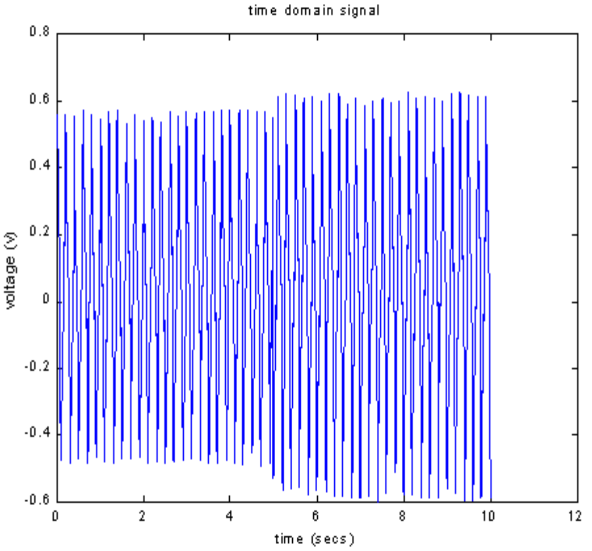


Figure 4.22: Time domain vibration signal corresponding to the sinusoidal input, with standardized amplitude of 1V and frequency of 5Hz (at t=5s, a fault is simulated).

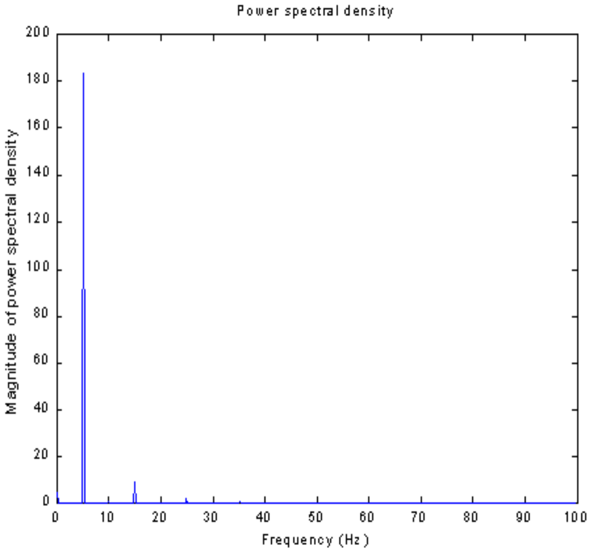


Figure 4.23: Vibration signature corresponding to the sinusoidal input, with standardized amplitude of 1V and frequency of 5Hz.

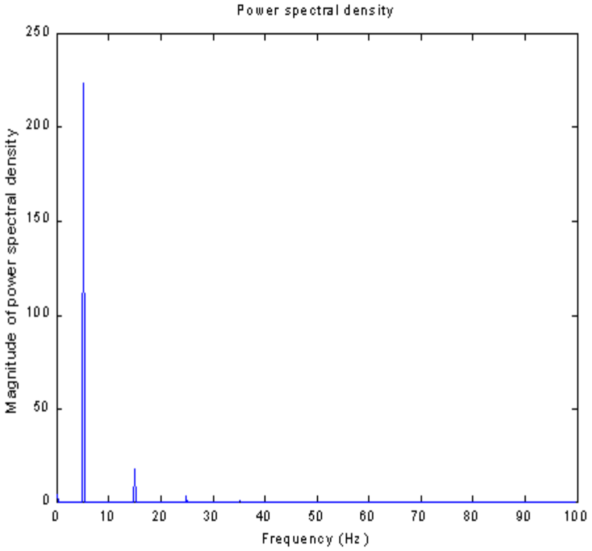


Figure 4.24: Spectrum of machine corresponding to the sinusoidal input (with standardized amplitude of 1V and frequency of 5Hz) after fault occurs.

automation of manufacturing systems [69], fault detection and diagnosis [70], flight control [71], control of biomedical systems [72], and intelligent instrumentation [73]. KB control systems can be viewed as intelligent control systems for complex dynamical systems which cannot be described crisply using mathematical models [74]. The knowledge-base is usually a rules-base, developed using knowledge-representation mechanisms, such as frames, semantic nets, and causal diagrams [75]. The inference mechanism in the KB controller uses sophisticated matching strategies to determine which rules should be allowed to fire, including refraction, recency and priority-based schemes, to reach a final decision. The knowledge-base essentially captures the experience of skilled operators, which is then used in the inferencing process so as to emulate, in an automatic manner, the operations of a complex system as closely as possible to what the most consistent and experienced experts would have done under the same situations.

A KB control system comprises of a central intelligent controller (usually implemented on a workstation or a high performance PC) with a knowledge-base, instrumented via a digital bus (such as the General Purpose Instrumentation Bus) to front-end microprocessor-based controllers [1]. This central controller will be referred to as the expert controller henceforth in the chapter. The front-end controllers gather real-time information of the system, and transmit them to the expert controller. Through the knowledge-base, the expert controller will execute the inference procedures to determine the best course of actions necessary. Commands will then be transmitted to the front-end devices which will carry out the control actions. Clearly, the main intelligence lies in the expert controller which is usually also the most costly and sophisticated component of the overall system. The maximum operating distance from the expert controller to the front-end devices, depends on the type of bus used, and the amount of noise interference affecting signal transmission [76]. Using a serial bus (instead of a parallel bus), this distance can be increased, albeit still within the local vicinity of the plant [77].

Many manufacturing processes are now widely distributed geographically, due to economy-related factors in manufacturing and distribution. The layout of an entire plant can now be rather extensive, spreading across continents in certain cases. Therefore, it has become an important challenge to be able to optimize any synergy opportunities in the operations of these distributed systems. In many cases, the same set of processes to manufacture the same product (or to monitor the same process) can be cloned over different plants. This requires close coordination and synchronization of the distributed operations, as well as an efficient remote monitoring and control facility in place. Thus, an extensive and 'borderless' approach towards the dissemination of expert knowledge and coordination efforts to distributed points and seamless integration of control strategies applied to distributed yet identical systems is crucial to enhance overall efficiency and operational costs. The work in this chapter is motivated by this observation.

In the proposed approach, the basic control configuration of the KB control system remains the same. However, instead of the usual local bus instrumentation, the expert controller is now connected via the Internet to the clients. The clients will carry out the real-time operations necessary. The role of the expert controller is more supervisory in nature. It makes the crucial decisions to operate the remote pawn pieces to maximize a plantwide efficiency. Multiple front-end clients may be connected to the expert controller at any one time, tapping on the widespread existing network of the Internet. In this chapter, the multi-access issue is resolved using the datasocket technology [78]. DataSocket is an Internet programming technology that simplifies data exchange between computers and applications and solves the common problems of client-server applications. With the DataSocket, passing data over the Internet, and responding to multiple users (client) can be achieved more efficiently, without incurring the complexity of low-level TCP/IP programming. Moreover, the speed of the data exchange between the various computers via datasocket technology is faster and more stable, compared to the usual client-server data exchange via a web browser.

An exemplary application in the remote monitoring and control of machines distributed over different locations, using a KB controller via the Internet, will be presented to further illustrate the principles of the proposed configuration. The hardware architecture of the KB control system is shown in Figure 4.25. The knowledge-base and the inference mechanism is located in the expert controller. The remote front-end microprocessor-based controller is responsible for the real-time algorithmic control. The real-time data of the system is collected by these front-end controllers and transmitted via the Internet to the expert controller. These data are monitored and used by the expert controller in conjunction with the knowledge-base. When an analysis of the collected data is completed, the decisions of the expert controller are sent via the Internet to update the front-end controllers.

The expert controller uses a web page with Datasocket reader components which connect to the datasocket server, and which read the system information from the front-end controllers. The operator can restart the front-end data acquisition applications or reboot the front-end controllers without having to reconnect to the expert

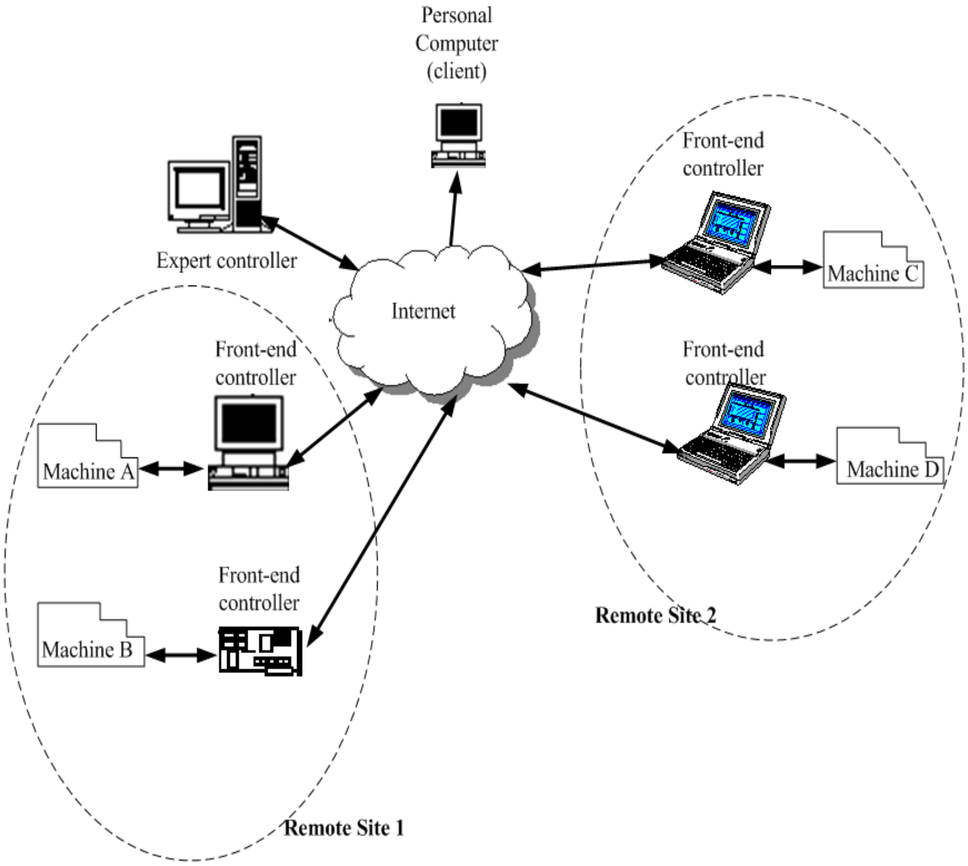


Figure 4.25: KB control system via the Internet.

controller. The expert controller remains connected to the front-end controllers and receives new data as soon as the front-end applications restart via the datasocket server.

Web interactivity is an essential issue to be addressed for this remote monitoring application, as the communication and data exchanges between the local controller (from the remote user viewpoint) and the remote front-end controllers is an integral aspect of the whole system. For this purpose, many different software architectures (which provide a common linking platform for the local and remote front-end controllers) can be feasible. One possible approach is to share real-time data between the local controller and the remote front-end controllers using the Datasocket technology (Figure 4.26). Datasocket is a programming tool that enables the user to read, write, and share data between applications and/or different data sources and targets. Datasocket can access data in local files and data on HTTP and FTP servers. If general purpose file I/O functions, TCP/IP functions, and FTP/HTTP requests are used to transfer data, separate code for each protocol has to be written. However, datasocket provides a unified API for these low-level communication protocols. Different data sources can be accessed without writing different codes to support different data formats and protocols, since the datasocket control converts data for transfer and passes the actual values to the applications. To connect to a data source location, a URL has to be specified. Datasocket also recognizes several existing schemes, including HTTP, FTP, file (local files), and OPC. The Datasocket also has a scheme, i.e., DSTP, for sharing live data through the Datasocket servers. Thus real-time process monitoring, supervisory analysis of data and control can be achieved in this configuration. In some cases, whereby the workload of a single expert controller for the many front-end controllers may be too heavy, two or more expert controllers can be implemented to distribute the workload. The expert controllers and the front-end controllers will be all connected through the datasocket server.

Another possible way for data transfer across the Internet is to write applications using low-level protocols, such as TCP/IP and *User Datagram Protocol* (UDP). Yet another possible approach is to publish data through the web using a web-server (Figure 4.27). The method chosen for this monitoring application is using the *Laboratory Virtual Instrument Engineering Workbench* (LabVIEW) web-server to publish data on the web.

The synergy of KB control with the Internet realizes all the main benefits associated with an Internet application for the KB control system. The operational span of the KB control system is now virtually without constraints. The clients can operate from anywhere with an Internet access. The intelligence concentrated in the expert controller may be shared among multiple and distributed clients. Redundant expert controllers for contingency purposes can be set up at different locations.

With the Internet, it also means that it is not necessary to set up a local bus system to enjoy the benefits of a distributed control system. The information distribution, coordination and centralized supervisory control features can be realised as long as there are facilities to tap onto the Internet. Direct upgrading of the overall system can be done on the expert controller, effecting all downstream applications efficiently. With the Internet connection, it is also easy to tap on the machines manufacturers' resources directly. It is possible to expand the interface for the front-end clients or expert controller to wireless devices, including handphones and PDAs.

These advantages have direct positive implications in improving the efficiency of distributed operations, thereby enabling plantwide optimization and costs savings.

4.4 Application Example: Expert Vibration Monitoring System

To illustrate the effectiveness of the KB control via the Internet, an expert vibration monitoring system for monitoring and control of machine vibrations is presented in this section.

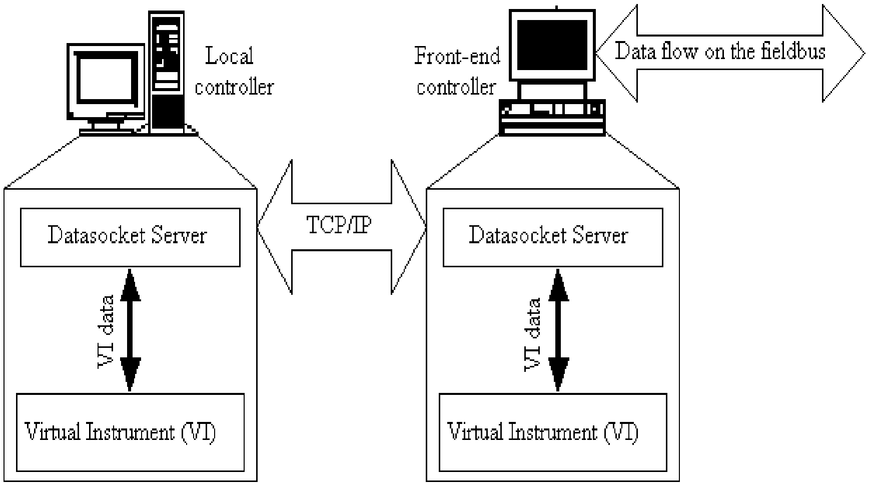


Figure 4.26: Datasocket transfer method.

4.4.1 Operational Principles

The expert vibration monitoring system is able to continuously monitor the real-time vibration patterns from multiple machines connected through the Internet. The system operates in two modes: the learning and the monitoring mode. In the learning mode, vibration signatures ([60] and [61]), representative of the health of the machines to be monitored, are first derived from the vibration signals of the machines when operating under normal conditions. They are stored in the knowledge-base of the expert controller. Accelerometers mounted on the machine directly provide measurements of the vibration signals. Thus, this expert monitoring system may be applied to existing machines, without much modifications necessary.

In the monitoring mode, pattern recognition templates are used to compare the real-time vibration patterns measured from the monitored machines against the vibration signatures stored in the knowledge-base. The expert controller is able to generate decisions on the well-beings of the machines, taking into considerations various criteria

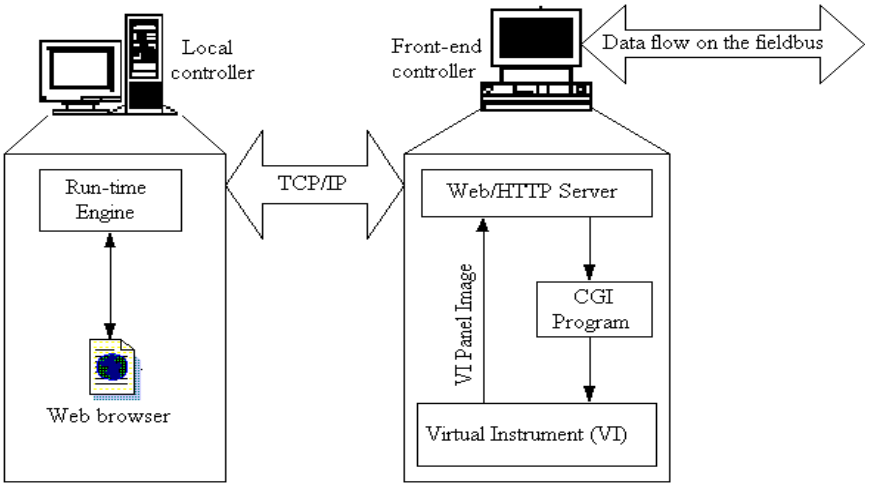


Figure 4.27: Web-server transfer method.

based on a fuzzy fusion technique. An alarm is activated when the difference deviates beyond an acceptable threshold. Subsequently, rectification actions, to provide a warning or automatic corrective action (e.g., changing the operating conditions of the machine, modifying the parameters of the controller or shutting down the machine), may be invoked before extensive damage is caused to the machine.

4.4.2 System Configuration

The configuration of the expert vibration monitoring system is shown in Figure 4.28. It consists of accelerometers, which are mounted on the machines to be monitored. The accelerometers, which serve as the interface between the machine and the front-end controllers, measure the multi-frequency vibration signals and transmit these signals to the data acquisition (DAQ) cards installed in the front-end controllers which may be in the form of *PC*s, laptops, or even standalone DSP devices.

The expert controller contains the main intelligence and coordinates the entire monitoring and control process. It can initiate the learning mode, when the machines

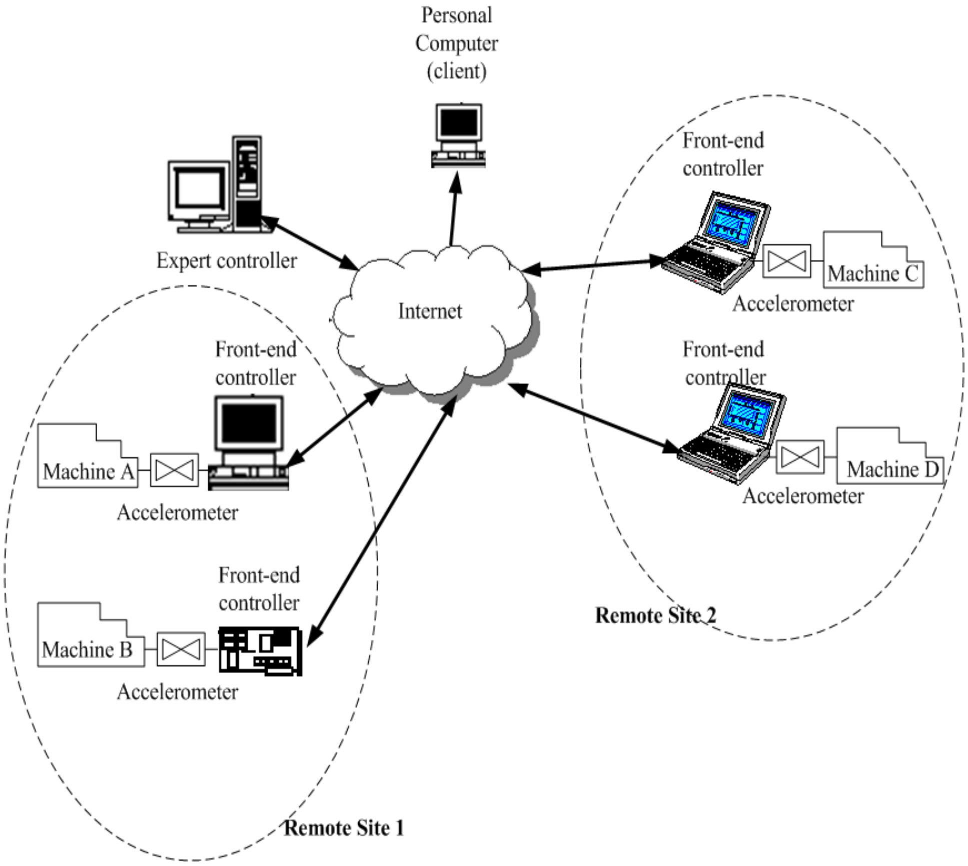


Figure 4.28: Expert vibration monitoring system.

are operating normally. With the vibration signals acquired by the accelerometers and transmitted to the server, it then derives suitable vibration signatures. Different types of vibration signatures can be used. For example, one form of vibration signature may be based on the amplitude of the vibration; another form may use a time series analysis of the vibration; yet another form may employ the spectrum of the vibration which can be efficiently obtained using the FFT algorithm. In the monitoring mode, the expert controller will decide, based on comparing the actual vibration patterns with the signatures of the machines, if an alarm should be raised. No deliberate or additional input signal is required, so the machine operations are not disrupted.

4.4.3 Inferencing Process

This inferencing process will be based on the rules in the knowledge-base. Here, the actual vibration pattern is compared against the relevant vibration signatures. The analysis and comparison may be done in terms of different evaluation criteria, such as the shift in frequency or amplitude of the spectrum, or a combination of the two. The same fusion technique as mentioned in Section 4.3.2 is used in the determination of each of the machine's condition, i.e., `HEALTH` attribute which is computed from multiple evaluation criteria. A fuzzy weighted approach, similar to that mentioned in Section 4.3.2, is used to realize the $\Im$ function.

4.4.4 Experiments

Shaker tables (Figure 4.15) are used as the test platforms representing the machines. Two shaker tables (named $ShakerTableA$ and $ShakerTableB$) will be used for illustration in the experiments.

A. Generation of the Vibration Signature

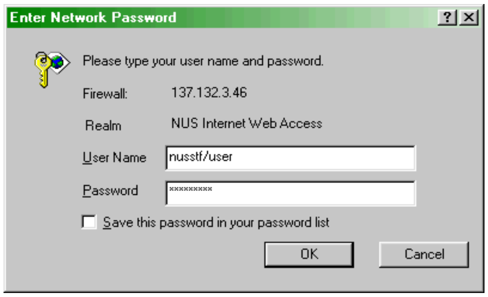
The learning mode is first initiated by the expert controller to obtain the vibration signatures of the machines to be monitored under normal conditions. The normal conditions are emulated as corresponding to specific input signals to the shaker. To illustrate the different natures of machines, the normal input to $ShakerTableA$ is assumed to be pulse trains, and the normal input to $ShakerTableB$ is assumed to be chirp signals.

B. Inferencing Process

Three evaluation criteria, as similar to (4.17), are used. The rules used for the computation of the `HEALTH` attribute are similar to that used in Section 4.3.4. The machine condition attribute `HEALTH` is then computed as in (4.15).

C. Tests

A typical session begins at the expert server's end. Upon successful authentication (Figure 4.29) of the user, the control panel at the server side of the expert vibration analyzer as shown in Figure 4.30 will appear. The top half and bottom half of the control panel shows the signals for *ShakerTableA* and *ShakerTableB* respectively. The user may initiate the learning or monitoring mode by clicking on one of the push buttons on the top left side of the panel. Clicking on the 'Start' button will begin any mode that has been selected. Whenever an alarm is triggered in the system, the 'Alarm' LED near the right side of the panel will be lighted up. The chart area on the left side of the panel displays the vibration signals (in time domain) that are transmitted to the analyzer; whereas the chart area on the right side of the panel displays the spectra of the vibration signals. Results and short messages to the user will be displayed in the message box.



Please type your user name and password.

Firewall: 137.132.3.46

Realm NUS Internet Web Access

User Name nusstf/user

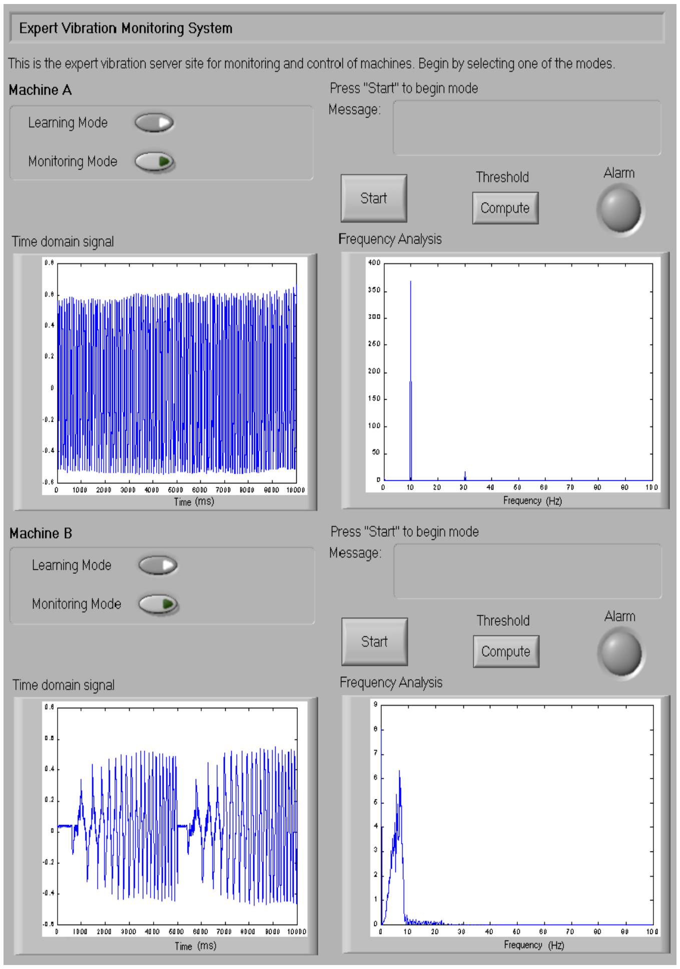
Password **********

□ Save this password in your password list

Figure 4.29: Authentication of the user for entry into the expert monitoring system.

C.I. Learning Mode

The learning mode must first be invoked to obtain the vibration signatures associated with the machines. These vibration signatures are then stored in the knowledge-base of the expert vibration monitoring system. A snapshot of the control panel executing the learning mode of the vibration analysis (to obtain the different signatures) is given in Figure 4.30.



Expert Vibration Monitoring System

This is the expert vibration server site for monitoring and control of machines. Begin by selecting one of the modes.

Machine A

Press "Start" to begin mode

Message:

Machine B

Press "Start" to begin mode

Message:

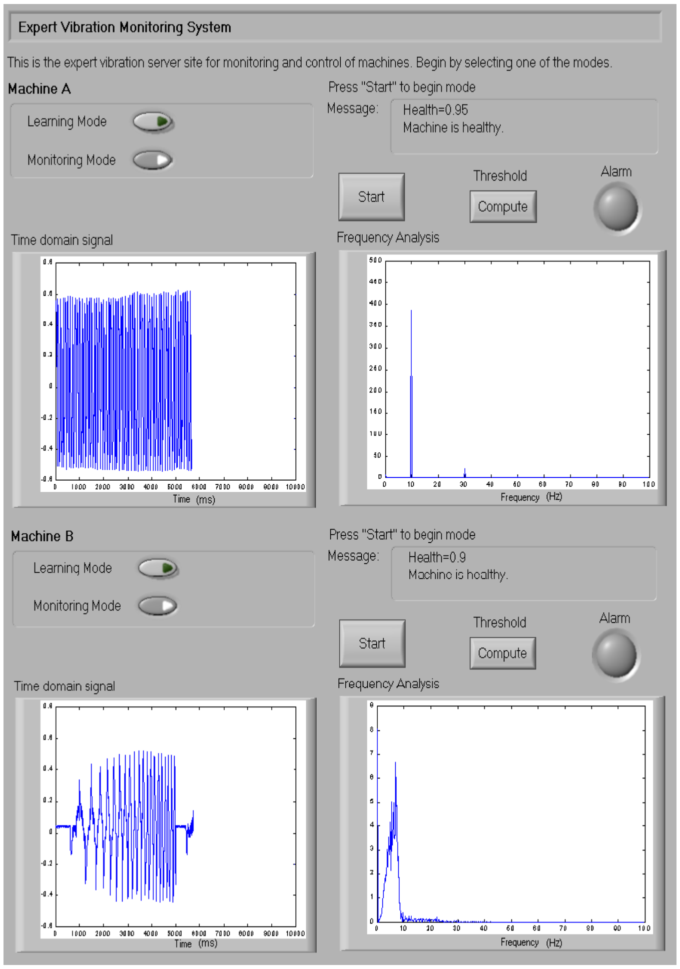
Figure 4.30: Learning mode - Vibration signatures of $ShakerTableA$ and B.

C.II. Monitoring Mode

At time $t = 5s$, the input gain for $ShakerTableA$ is increased by a factor of 1.5 times and the pulse input frequency is increased slightly to emulate a fault arising in the machine (*ShakerTableB* is left unchanged). The snapshot (Figure 8) of the control panel before $t = 5s$ shows that both machines are 'healthy'. Following the fault, as shown in Figure 9, the snapshot shows that the vibration analysis algorithm (in the expert controller end) is able to detect the fault occurring in *ShakerTableA*. Before the introduction of the fault, the `HEALTH` attribute of *ShakerTableA* is 0.95. After the introduction of the fault, the `HEALTH` attribute of *ShakerTableA* falls to 0.55 which is below the threshold value (which is set at 0.6 for the monitoring mode). As a result, the alarm for *ShakerTableA* is triggered. *ShakerTableB*, however, continues to operate normally.

At time $t = 14s$, a sinusoidal input signal is used instead of the chirp input signal for *ShakerTableB* to emulate an abnormal condition. Following the inference procedures, the expert system concludes that a fault has also occurred at *ShakerTableB* and the alarm for *ShakerTableB* is raised accordingly (Figure 4.33). The `HEALTH` attribute of *ShakerTableB* falls from about 0.9 (before the fault is introdued) to 0.53 (after the fault is introduced). This value of `HEALTH` attribute is below the threshold value (which has been set at 0.6).

Comparing Figures 4.31 and 4.33, it is shown that the vibration pattern of *Shaker TableB* (in the frequency domain) is changed from an envelope shape (as shown in Figure 4.31 before the fault is introduced) to a pure dominant spike (as shown in Figure 4.33 after the fault is introduced). The fault is also reflected in the time domain signals for *ShakerTableB* before (Figure 4.31) and after (Figure 4.33) the fault is introduced. The vibration analysis algorithm (in the expert controller end) is also able to detect the fault in *ShakerTableB*. *ShakerTableA* remains in the ‘unhealthy’ range.



Expert Vibration Monitoring System

This is the expert vibration server site for monitoring and control of machines. Begin by selecting one of the modes.

Machine A

Press "Start" to begin mode

Message: Health=0.95
Machine is healthy.

Machine B

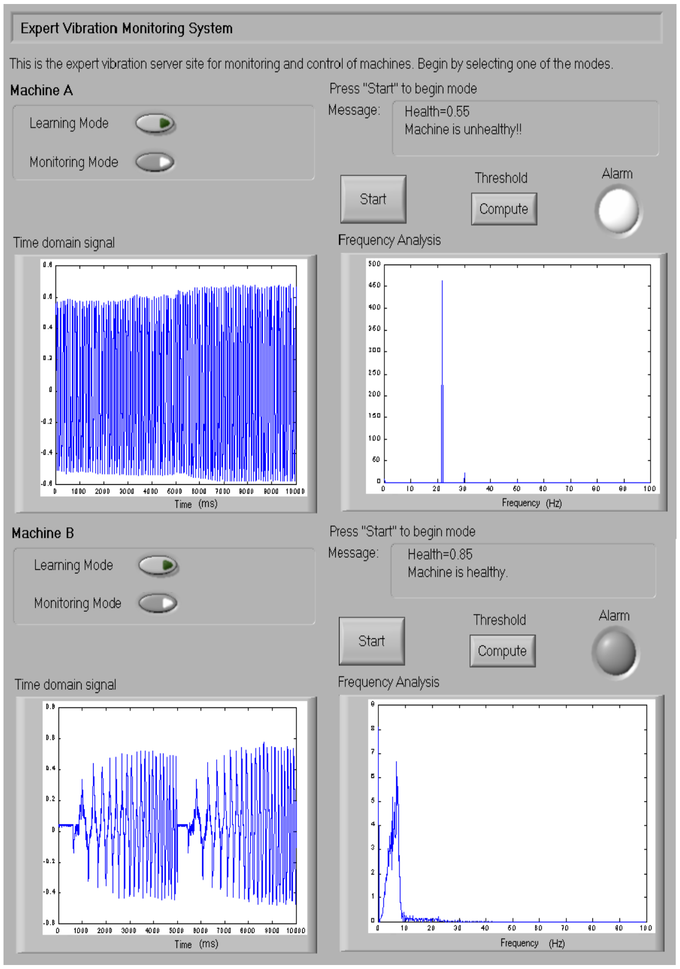
Press "Start" to begin mode

Message: Health=0.9
Machino is hoalthy.

Figure 4.31: Monitoring mode - Snapshot of the expert vibration control panel before any fault is emulated.

4.5 Conclusions

Two approaches to mitigate the damage caused by the mechanical vibrations in precision motion systems are developed. In the first approach, an adaptive notch filter is employed to detect online the resonant frequencies in the vibration signal and to adaptively position its notch so as to suppress the signals at these frequencies. In the second approach, a real-time analyzer, based on a fuzzy fusion technique, is used to continuously monitor and compare the actual vibration patterns against a set of vibration signatures. The vibration signal is acquired using an accelerometer mounted on the machine. The approach can be applied to existing machines without modification, or disrupting the operation. Both computer simulations and experiments using a prototype machine are carried out. The results illustrate the satisfactory performance and the advantages associated with these approaches. The development of a KB control system for remote monitoring and fault diagnosis of machines which operates via the Internet has been developed in the chapter.



Expert Vibration Monitoring System

This is the expert vibration server site for monitoring and control of machines. Begin by selecting one of the modes.

Machine A

Press "Start" to begin mode

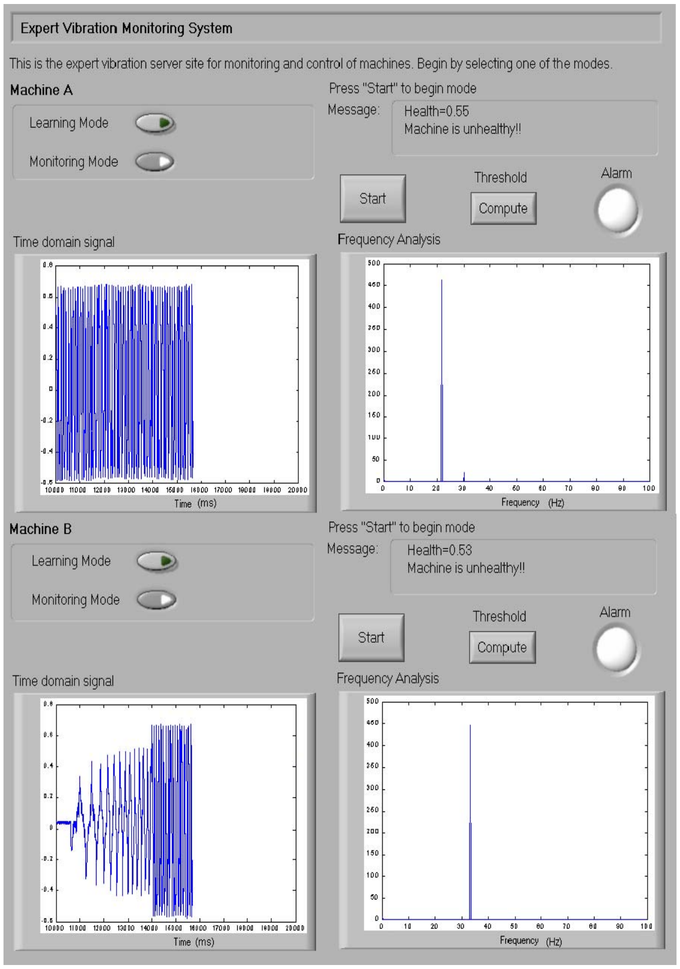
Message: Health=0.55
Machine is unhealthy!!

Machine B

Press "Start" to begin mode

Message: Health=0.85
Machine is healthy.

Figure 4.32: Monitoring mode - Snapshot of the expert vibration control panel after a fault is emulated on $ShakerTableA$.



Expert Vibration Monitoring System

This is the expert vibration server site for monitoring and control of machines. Begin by selecting one of the modes.

Machine A

Press "Start" to begin mode

Message: Health=0.55
Machine is unhealthy!!

Machine B

Press "Start" to begin mode

Message: Health=0.53
Machine is unhealthy!!

Figure 4.33: Monitoring mode - Snapshot of the expert vibration control panel after a fault is emulated on $ShakerTableB$.

Chapter 5

Conclusions

5.1 General Conclusions

With the progress of precision manufacturing, the increasing complexity of precision motion systems coupled with the increasing demands in closed loop performance specifications necessitates the use of more complex and sophisticated controllers. There are much benefits to be gained by combining intelligent control with the well-established tools in control theory to develop robust, high speed and high precision motion systems. In this thesis, intelligent control is combined with the well-established tools in control theory to achieve advancements in intelligent instrumentation, control and monitoring of precision motion systems.

Firstly, the thesis has presented an intelligent controller that performs adaptive online correction and interpolation of quadrature encoder signals. In this chapter, a two-stage RBFNN is employed to carry out concurrently the correction and interpolation of encoder signals in real-time.

Subsequently, a new PID/adaptive controller for precision motion systems is developed. The second-order model is used as the nominal dominant model for the design of the PID controller, and an adaptive component designed based on a RBFNN provides for the possibility of performance enhancement when the feedback control alone is inadequate to cope with uncertain nonlinear phases.

Lastly, two approaches to monitor and suppress mechanical vibrations in precision motion systems are covered. The first approach utilizes an adaptive notch filter to identify the resonant frequencies and suppress any signal transmission into the system at these frequencies. The second approach uses a real-time analyzer to detect excessive vibration based on which appropriate actions can be taken. To expand the scope of vibration monitoring, an exemplary application concerning the remote monitoring and control of machines distributed over different locations, using a knowledge-based controller via the Internet, is presented.

The effectiveness of the proposed approaches is highlighted in the simulation and experimental studies.

5.2 Recommendations for Future Work

As mentioned in Chapter 1, much benefits could be gained by combining intelligent control with the well-established tools in control theory. In this perspective, advancements in system design and implementation can be made by exploring different combinations of control theory and artificial intelligence. The aim is to design robust and high performance precision motion systems, in the face of disturbances and other uncertainties.

5.2.1 Improvements in Intelligent Controllers

Considering the many works in the literature [1], it was shown that NNs and fuzzy logic are complementary methodologies in the design and implementation of intelligent systems. Hybrid controllers, involving NNs and fuzzy logic, have much potential but yet to be explored for many purposes. It is known that NNs are useful as function approximators because they do not require prior knowledge of the input data distribution or the system. Thus, NNs can be used for the design of intelligent controllers in precision motion systems, operating in uncertain and time-varying environments. From a systems theoretic point of view, NNs can be considered as practically implementable parametrizations of nonlinear maps from one finite-dimensional space to another. Such networks are ideally suited to cope with all three categories of difficulties encountered in complex control systems (i.e., computational complexity, nonlinearity, and uncertainty). It is also known that they can approximate arbitrary nonlinear maps. In certain situations, the knowledge of the plant is expressed in linguistic terms. Fuzzy logic is thus a viable candidate. With a good understanding of motion systems and its environment, many control methodologies are available in the literature [35]. However, in some instances, the plant to be controlled may be too complex and the physical processes in the system may not be fully understood. Thus, the controller has to adapt itself to improve overall system's performance while maintaining stability. System identification needs to be commissioned in the system to obtain a better understanding of the plant and its environment.

In view of the above, one of the future direction of research is to develop an adaptive hierarchical neuro-fuzzy controller, with learning capabilities, for the identification and control of precision motion systems. The higher levels of the network will provide supervisory control of the lower levels to tackle the various control tasks of the system.

5.2.2 Intelligent Geometrical Compensation using Support Vectors

As precision engineering is poised to enter a nanotechnology regime, one of the key challenges in achieving sub-micrometer technology positioning accuracy is to have efficient geometrical error compensation. The key importance in achieving these results is to have a good model for describing the overall errors, an accurate method for reading the positioning errors, and a sound scheme to implement the control/compensation algorithm. With a suitable error model structure on hand, the next issue in compensating geometrical errors precision machine is on the actual implementation scheme of the compensation software. Since it is virtually impossible to locate the geometrical error at every particular location in the whole work space of the machine, only representative points are calibrated. The intermediate points would have to be calculated. The employed method must be fast since the control algorithm usually pass down a stringent requirement on the sampling time. On the other hand, it must be accurate to meet the minimum specifications of compensation. Till now, this is achieved by means of a look-up table storing the positional errors at calibrated points. To calculate the geometrical errors in between the points, linear interpolation is carried out. For sub-micrometer positioning accuracy, a massive amount of memory is required to store the look-up table which is not permissible in most servo controllers available in the market. To solve this issue, mathematical modeling can be applied to derive a parametric error model offline from the calibrated points. Taking into consideration the required precision and accuracy of the precision motion systems, the calculations involved would be too rigorous.

One avenue for future research is to develop a support vector methodology ([79]-[81]) to perform geometrical error compensation. The proposed approach is motivated by the abovementioned problems with the look-up table and the other approaches in the literature ([82]-[84]). A support vector regression (SVR) method can be used as the model and thus, arbitrary nonlinear functions can be fitted to the calibrated points. This model will serve as the basis for error compensation, thus dispensing with the need for look-up tables. Furthermore, inter-point interpolation can be a nonlinear one. The support vector machine (SVM), originated from the Statistic Learning Theory ([79] and [80]), is mostly used in regression and classification applications. The SVM is able to select the number of the basis functions systematically without the curse of dimensionality and the number of data points available. The common optimization problem of being trapped in local minimas is also avoided in SVM applications due to its fundamental Structural Risk Minimization (SRM) principle [80]. SVMs are believed to be able to generalize well on unseen data and overcome the problem of overfitting, considering the many outstanding results reported in the literature ([82]-[84]). All these attractive features suggest that SVMs are strong candidates for regression purposes.

5.2.3 Improvements in Learning Capabilities of NN

The thesis has earlier presented an application of the RBFNN for the purpose of adaptive online correction and interpolation of quadrature encoder signals. For this particular architecture of NN, the learning capability of the NN plays an important role in its overall effectiveness. Current learning algorithms reported in the literature [17] have their strengths and weaknesses. There is a need to improve on the learning capabilities of the RBF network. One of the future focus of research is to design an improved online backpropagation algorithm. It is well-known that the standard error backpropagation training of a multilayer perceptron [17] may converge very slowly (if at all) to a good local optimum. The convergence can be improved by properly controlling the learning rate during the course of the training [85]. The learning rate can be controlled in many ways, i.e., introducing a momentum term in the weight update rule [86], normalizing the inputs before presenting them to the network [87], using learning rates that are inversely proportional to the fan-in of the node [88], adopting a conjugate gradient search [89] and etc. These techniques rely on presumed properties of the cost function in the space of the network weights (e.g., a quadratic form with a positive definite Hessian [90]). In applications where the multilayer perceptrons of a few hundred thousand weights have to be trained on millions of examples, the central processing unit (CPU) time may become excessively large. However, the convergence rate can be increased dramatically by simply moving to online error backpropagation [91]. In view of these current literature on the learning capability of NNs [1], there is a need to design a robust learning algorithm.

Author's Publications

K.Z. Tang, K.K. Tan, C.W. De Silva, T.H. Lee and S.J. Chin, "Monitoring and Suppression of Vibration in Precision Machines", *Journal of Intelligent and Fuzzy Systems*, 10, IOS Press, pp. 33-52, 2002.

K.K. Tan, K.Z. Tang, H. Dou and S. Huang, "Development of an integrated and open-architecture precision motion control system", *Control Engineering Practice*, 10(7), pp. 757-772, 2002.

K.K. Tan and K.Z. Tang, "Interpolation of Quadrature Encoder Signals Using Radial Basis Function", *IEEE Trans. on Control Systems Technology*, accepted for publication, June 2004.

K.Z. Tang, H.L. Goh, K.K. Tan and T.H. Lee, "Knowledge-Based Control Via the Internet", *International Journal of Control, Automation and Systems*, Vol 2, No. 2, June 2004, pp. 1-13.

K.Z. Tang, S.N. Huang, K.K. Tan and T.H. Lee, "Combined PID and Adaptive Nonlinear Control for Servo Mechanical Systems", *Mechatronics-The Science of Intelligent Machines*, Vol. 14, No. 6, June 2004, pp. 701-714.

K.K. Tan, K.Z. Tang, C.W. De Silva, T.H. Lee and K.C. Tan, "Application of Vibration Sensing in Monitoring and Control of Machine Health", *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '01)*, Como, Italy, 2001.

K.K. Tan, K.N. Wang and K.Z. Tang, "Mechatronic Experiment on Remote Vibration Monitoring and Fault Diagnosis via the Internet", *International Journal of Engineering Education*, 19(3), pp. 503-511, 2003.

T.H. Lee, S.N. Huang, K.Z. Tang, K.K. Tan and A. Al Mamun, "PID Control Incorporating RBF-Neural Network for Servo Mechanical Systems", Proceedings on the *29th Annual Conference of the IEEE Industrial Electronics Society*, November 2-6, Virginia, USA, pp. 2789-2793, 2003.

K.Z. Tang, K.K. Tan, T.H. Lee and C.S. Teo, "Neural-based Correction and Interpolation of Encoder Signals for Precision Motion Control", Proceedings of *8th IEEE International Workshop on Advanced Motion Control*, Kawasaki (Japan), pp. 499-504, July 2003.

Bibliography

[1] Samad, T., *Perspectives in Control Engineering : Technologies, Applications, and New Directions*, New York: IEEE Press, 2001.

[2] Vincent, T.L. and Grantham, W.J., *Nonlinear and Optimal Control Systems*, New York: Wiley, 1997.

[3] Ge, S.S., Li, G.Y. and Lee, T.H., "Adaptive Control for a Class of Nonlinear Discrete-Time Systems Using Neural Networks", Proceedings of the *2001 IEEE International Symposium on Intelligent Control*, Mexico City, pp. 97-102, September 2001,

[4] Bellman, R., *Adaptive Control Processes - A Guided Tour*, Princeton, NJ: Princeton University Press, 1961.

[5] Narendra, K.S. and Koditschek, D.E., "Intelligent control systems design", *NSF Intelligent Control Initiative*, 1992.

[6] Emerson Process Management homepage, *http : //www.emersonprocess.com*, 2004.

[7] GE Fanuc Automation homepage, *http : //www.geindustrial.com/cwc/gefanuc/index.html*, 2004.

[8] Heydemann, P.L.M., "Determination and correction of quadrature fringe measurement errors in interferometers", *Applied Optics*, 20(19), pp. 26-38, October 1981.

[9] Birch, K.P., "Optical fringe subdivision with nanometric accuracy", *Precision Engineering*, 12(4), pp. 33-47, October 1990.

[10] Watanabe, K. and Yokote H., "A microstep controller of a DC servomotor", *IEEE Trans. on Instrumentation and Measurement*, 39(6), pp. 95-106, December 1990.

[11] Hagiwara, N. and Murase, H., "A method of improving the resolution and accuracy of rotary encoders using a code compensation technique", *IEEE Trans. on Instrumentation and Measurement*, 41(1), pp. 78-84, February 1992.

[12] Yokote, H. and Watanabe, K., "A hybrid digital and analog controller for DC and brushless servomotors", *IEEE Trans. on Instrumentation and Measurement*, 39(1), pp. 103-109, February 1990.

[13] Rene Mayer, J.R., "High resolution of rotary encoder analog quadrature signals", *IEEE Trans. on Instrumentation and Measurement*, 43(3), pp. 82-89, June 1994.

[14] Cheung, N.C. , "An innovative method to increase the resolution of optical encoders in motion servo systems", Proceedings of *IEEE 1999 International Conference on Power ELectronics and Drive Systems*, PEDS '99, Hong Kong, pp. 797-800, July 1999.

[15] Madni, A.M., Jumper, M. and Malcolm, T., "An absolute high performance, self calibrating optical rotary positioning system", IEEE Proceedings of Aerospace Conference 2001, 5, pp. 2363-2373, March 2001.

[16] ServoStar drives catalog and product updates (http://*www.motionvillage.com/training/handbook/feedback/sineencoders.html*), 2003.

[17] Haykin, S., *Neural Networks, A Comprehensive Foundation*, USA: Prentice Hall International Edition. 1994.

[18] Khana, T., *Foundation of Neural Networks*, New York: Addison Wesley, 1995.

[19] Wasserman, P., *Neural Computing: Theory and Practice*, New York: Van Nostrand, 1989.

[20] Zurada, J.M., *Introduction to Artificial Neural Systems*, USA: West Publishing Company, 1992.

[21] Ljung, L., *System Identification: Theory for the User*, London: Prentice Hall Inc., 2nd Edition, 1999.

[22] Huang, S., Tan, K.K., Tang, K.Z., *Neural Network Control: Theory and Applications*, England: Research Studies Press Ltd, 2004.

[23] Maciejowski, J.M., *Multivariable Feedback Design*, USA: Addison-Wesley, 1989.

[24] Broomhead, D.S. and Lowe, D., "Multivariable functional interpolation and adaptive networks", *Complex Systems 2*, pp. 321-355, 1988.

[25] Webb, A.R., "Functional approximation by feedforward networks: a least-squares approach to generalization", *IEEE Trans. on Neural Networks*, 5(3), pp. 363-371, May 1994.

[26] Tan, K.K. and Zhou, H.X., "New interpolation method for quadrature encoder signals", *IEEE Trans. on Instrumentation and Measurements*, 51(5), pp. 1073-1079, 2002.

[27] Hang, C.C., Astrom, K.J. and Ho, W.K., "Refinements of the Ziegler-Nichols tuning formula," in *I*EE Proceedings Part D, 138(2), pp. 111-118, 1991.

[28] Astrom, K.J. and Hagglund T., *Automatic Tuning of PID Controllers*, USA: Research Triangle Park, Instrument Society of America, 1988.

[29] Shinskey, F.G., *Process Control System: Application, Design and Tuning*, 3rd Edition. New York: McGraw-Hill, 1988.

[30] Rivera, D.E., Morari, M. and Skogestad S., "Internal model control for PID controller design", in *Ind. Eng. Chem. Process Des. Dev.*, 25, pp. 252-265, 1986.

[31] Huang, H.P., Chen, C.L., Lai, C.W. and Wang, G.B., "Auto-tuning for model based PID controllers", in *AIChE Journal*, 42(9), pp. 2687-2691, 1996.

[32] Xu, Y., Hollerbach, J.M. and Ma, D., "A nonlinear PD controller for force and contact transient control", *IEEE Control Systems*, 15(1), pp. 15-21, 1995.

[33] Huang, S., Tan, K.K. and Lee, T.H., "A combined PID/adaptive controller for a class of nonlinear systems", *Automatica*, 37(4), pp. 611-618, 2001.

[34] Basak, A., *Permanent-magnet DC Linear Motors*, Monographs in Electrical and Electronic Engineering, Oxford: Clarendon Press, 1996.

[35] Tan, K.K., Lee, T.H., Dou, H.F. and Huang, S.N., *Precision Motion Control*, London: Springer-Verlag London Limited, 2001.

[36] Tan, K.K., Huang, S.N., and Seet, H.L., "Geometrical error compensation of precision motion systems using radial basis functions", *IEEE Transactions on Instrumentation and Measurement*, 49(5), pp. 984-991, 2000.

[37] Tan, K.K, Lim, S.Y. and Huang, S.N., "Two-degree-of-freedom controller incorporating RBF adaptation for precision motion control applications", in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (AIM '99), (Altanta, USA), pp. 848-853, 1999.

[38] Fujimoto, Y. and Kawamura, A., "Robust servo-system based on two-degree-of-freedom control with sliding mode", in *IEEE Trans. on Industrial Electronics*, 42(3), pp. 272-280, 1995.

[39] Armstrong-Helouvry, B., Dupont, P. and de Wit, C.C., "A survey of models, analysis tools and compensation methods for the control of machines with friction", *Automatica*, 30(7), pp. 1083-1138, 1994.

[40] Tan, K.K., Lee, T.H., Huang, S. and Jiang, X., "Friction modeling and adaptive compensation using a relay feedback Approach", *IEEE Transactions on Industrial Electronics*, 48(1), pp. 169-176, 2001.

[41] Tan, K.K., Dou, H.F., Chen, Y.Q. and Lee, T.H., "High precision linear motor control via artificial relay tuning and zero-phase filtering based iterative learning", *IEEE Transactions on Control Systems Technology*, 9(2), pp. 244-253, 2001.

[42] Longman, R.W., "Designing iterative learning and repetitive controllers". In Z. Bien and Xu J.-X. eds, "*Iterative Learning Control- Analysis, Design, Integration and Application*", Kluwer Academic Publishers, pp. 107-145, 1998.

[43] Tan, K.K., Lim, S.Y., Lee, T.H. and Dou, H.F., “High precision control of linear actuators incorporating acceleration sensing”, *Robotics and Computer-Integrated Manufacturing*, 16, pp. 295-305, 2000.

[44] Tan, K.K., Huang, S.N., Lee, T.H., Chin, S.J., and Lim, S.Y., “Adaptive robust motion control for precise trajectory tracking applications”, *ISA Transactions*, 40(1), pp. 57-71, 2001.

[45] Tan, K.K., Lee, T.H., Dou, H.F., and Lim, S.Y., “An adaptive ripple suppression/compensation apparatus for permanent magnet linear motors”, Technical Report- Department of ECE, National University of Singapore, 2000.

[46] Yamada, K., Komada, S., Ishida, M., and Hori, T., “Analysis and classical control design of servo system using high order disturbance observer”, *23rd International Conference on Industrial Electronics, Control and Instrumentation IECON 97*, Vol. 1, pp.4-9, 1997.

[47] Tan, K.K., Lee, T.H., Dou, H.F., and Chin, S.J., “PWM modeling and application to disturbance observer-based precision motion control”, *PowerCon 2000*, Perth, Australia, CD-format, 2000.

[48] Hornik, K., Stinchcombe, M. and White, H., “Multilayer feedforward networks are universal approximators”, *Neural Networks*, 2, pp. 359-366, 1989.

[49] Fabri, S. and Kadirkamanathan, V., “Dynamic structure neural networks for stable adaptive control of nonlinear systems”, *IEEE Trans. on Neural Networks*, 7(5), pp. 1151-1167, 1996.

[50] Tan, K.K., Wang, Q.G., Hang, C.C., and Hagglund, T., *Advances in PID control*, London: Springer-Verlag London Limited, 1999.

[51] Lewis, F.L., Yesildirek, A., and Liu, K., "Multilayer neural-net robot controller with guaranteed tracking performance", *IEEE Trans. on Neural Networks*, 7(2), pp. 388-398, 1996.

[52] Zhang, T., Ge, S.S. and Hang, C.C., "Design and performance analysis of a direct adaptive controller for nonlinear systems", *Automatica*, 35, pp. 1809-1817, 1999.

[53] Astrom, K.J. and Hagglund, T., *PID controllers: Theory, Design and Tuning*, 2nd edition. USA: Research Triangle Park, Instrument Society of America, 1995.

[54] Friman, M. and Waller, K.V., "A two-channel relay for autotuning", *Ind. Eng. Chem. Res.*, 36, pp. 2662-2671, 1997.

[55] Ferdjallah, M. and Barr, R.E., "Adaptive digital notch filter design on the unit circle for the removal powerline noise from biomedical signals", *IEEE Trans. on Biomedical Engineering*, 41(6), pp. 529-536, 1994.

[56] Ahlstrom, M.L. and Tompkins, W.J., "Digital filters for real-time ECG signal processing using microprocessors", *IEEE Trans. on Biomedical Engineering*, 32, pp. 708-713, 1985.

[57] Glover, J.R. Jr., "Comments on digital filters for real-time ECG signal processing using microprocessors", *IEEE Trans. on Biomedical Engineering*, 34, pp. 962-963, 1987.

[58] Bertran, E. and Montoro, G., “Adaptive suppression of narrow-band vibrations”, 5th International Workshop on *Advanced Motion Control*, pp. 288-292, 1998.

[59] Kwan, T. and Martin, K., “Adaptive detection and enhancement of multiple sinusoids using a cascade IIR filter”, *IEEE Trans. on Circuits and Systems*, 36(7), pp. 937-947, 1989.

[60] Vierck, Robert K., *Vibration Analysis*, New York: Crowell, 1979.

[61] de Silva, Clarence W., *Vibration: Fundamentals and Practice*, USA: CRC Press LLC, 2000.

[62] *TMSS320C24x DSP Controllers Evaluation Module Technical References*, USA: Texas Instruments Inc., 1997.

[63] Ramirez, Robert W., *The FFT Fundamentals and Concepts*, Englewood Cliffs: Prentice-Hall Inc, 1985.

[64] Zadeh, L.A.,“Outline of a new approach to the analysis of complex systems and decision process”, *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 3, pp. 28-44, 1973.

[65] Takagi, T. and Sugeno, M.,“Fuzzy identification of systems and its applications to modelling and control”, *IEEE Transactions on Systems, Man, and Cybernetics*, 15, pp. 116-132, 1985.

[66] de Silva, C.W. and MacFarlane, A.G.J., “Knowledge-based control approach for robotic manipulators”, *Int. J. Control*, Vol. 50, pp. 249-273, 1989.

[67] Luo, R.C., Lin, M.H., and Shen, S.H., "The developmment of object-oriented knowledge base and adaptive motion planning for autonomous mobile robots", *IEEE/RSJ Int. Conf. Intelligent Robots and Syst.*, USA, pp. 108-115, 2001.

[68] Wu, Q.M.J., Lee, M.F.R., and de Silva, C.W., "Intelligent 3-D sensing in automated manufacturing processes", *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vol. 1, pp. 334-339, 2001.

[69] Wang, X.G., Liu, W., Gu, L., Sun, C.J., Gu, C.E. and de Silva, C.W., "Development of an intelligent control system for wood drying processes", *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Vol. 1, pp. 371-376, 2001.

[70] Vachtsevanos, G. and Davey, K., "Fault diagnosis for the space station thermal control system using a hybrid analytic/intelligent approach", *Proc. IEEE ISIC*, pp. 54-58, 1987.

[71] Lea, R.K., Allen, R. and Merry, S.L., "A comparative study of control techniques for an underwater flight vehicle", *International Journal of Systems Science*, Vol. 30(9), Sept. 1999.

[72] Alhady, S.S.N., Venkatachalam, P.A. and Sulaiman, M., "Noiseless ECG monitoring system with integrated expert system", *1st International Conference on IEE Conf. Publ. No. 476*, pp. 79-87, 2000.

[73] Ahlrichs, U., Paulus, D. and Niemann, H., "Integrating aspects of active vision into a knowledge-based system", *15th International Conference on Pattern Recognition*, Vol. 4, pp. 579-582, 2000.

[74] de Silva, C.W. and MacFarlane, A.G.J., *Knowledge-based Control Approach with Application to Robots*, Berlin: Springer-Verlag, 1989.

[75] Van Eck, P., Engelfriet, J., Fensel, D., Van Harmelen, F., Venema, Y. and Willems, M., “A survey of languages for specifying dynamics: a knowledge engineering perspective”, *IEEE Trans. on Knowledge and Data Eng.*, 13(3), June 2001.

[76] Berge, J., *Fieldbuses for Process Control*, The Instrumentation, Systems and Automation Society, 2002.

[77] Jordan, J.R., *Serial Networked Field Instrumentation*, Chichester, W. Sussex, Eng., New York: John Wiley, 1995.

[78] National Instruments LabVIEW Internet Developers Toolkit Webpage, http://www.natinst.com/labview/internet/, 2003.

[79] Girosi, F., “An equivalence between sparse approximation and support vector machines”, *Neural Computation*, Vol. 10, pp. 1455-1480, 1998.

[80] Vapnik, V., *The Nature of Statistical Learning Theory*, Berlin: Springer-Verlag, 1995.

[81] Burges, C.J.C., “A tutorial on support vector machines for pattern recognition”, *Data Mining and Knowledge Discovery*, Vol. 2, pp. 121-167, 1998.

[82] Drucker, H., Burges, C., Kaufman, L., Smola, A., and Vapnik, V., “Support vector regression machines”, *Advances in Neural Information Processing Systems*, Vol. 9, pp. 151-161, 1997.

[83] Muller, K.R., Smola, A., Ratsch, G., Scholkopf, B., Kohlmorgen, J., and Vapnik, V., “Predicting time series with support vector machines”, in Proceedings of *Artificial Neural Networks*- ICANN’97, pp. 999-1004, 1997.

[84] Mukherjee, S.O., Osuna, E., and Girosi, F., “Nonlinear prediction of chaotic time series using support vector machines”, Proceedings of the 1997 IEEE Signal Processing Society Workshop, *Neural Networks for Signal Processing*, pp. 511-520, 1997.

[85] Chan, L. and Fallside, F., “An adaptive training algorithm for backpropagation networks”, *Comput. Speech, Language*, Vol. 2, pp. 205-218, 1987.

[86] Rumelhart, D., Hinton, G. and Williams, R., *Parallel Distributed Processing: Exploration of the Micro-Structure of Cognition*, Cambridge, MA: MIT Press, 1986.

[87] Leung, H. and Zue, V., “Phonetic classification using multi-Layer perceptrons”, *IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1, Albuquerque, NM, pp. 525-528, April 1990.

[88] Plaut, D. and Hinton, G., “Learning sets of filters using back-propagation”, *Comput., Speech, Language*, Vol. 2, pp. 35-61, 1987.

[89] Polak, E., *Computational Methods in Optimization*, New York: Academic, 1971.

[90] Golden, R., *Mathematical Methods for Neural Network and Design*, Cambridge, MA: MIT Press, 1996.

[91] Bourlard, H. and Wellekens, C., “Links between markov models and multilayer perceptrons”, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, pp. 1167-1178, 1990.