5-2006

On-Line Learning and Wavelet-Based Feature Extraction Methodology for Process Monitoring using High-Dimensional Functional Data

Olufemi Abayomi Omitaomu
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Olufemi Abayomi Omitaomu entitled "On-Line Learning and Wavelet-Based Feature Extraction Methodology for Process Monitoring using High-Dimensional Functional Data." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Adedeji B Bariru, Myong K. Jeong, Major Professor

We have read this dissertation and recommend its acceptance:

Fong-Yuen Ding, Dukwon Kim, J. Wesley Hines

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Olufemi Abayomi Omitaomu entitled "On-Line Learning and Wavelet-Based Feature Extraction Methodology for Process Monitoring using High-Dimensional Functional Data." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Adedeji B. Badiru
Major Professor

Myong K. Jeong
Co-Advisor

We have read this dissertation
and recommend its acceptance:

Fong-Yuen Ding

Dukwon Kim

J. Wesley Hines

Acceptance for the Council:

Anne Mayhew
Vice Chancellor and
Dean of Graduate Studies

(Original signatures are on file with official student records.)

On-Line Learning and Wavelet-Based Feature Extraction Methodology for Process Monitoring using High-Dimensional Functional Data

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Olufemi Abayomi Omitaomu
May 2006

DEDICATION

This dissertation is dedicated:

To the Glory of God,

To the three "Women" in my Life,
My Wife, *Remilekun Enitan*,
My Mother, *Mojisola Zainab*, and
My Late Grandma, *Rukayatu Alake*

and

To the three "Men" in my Life.
My Sons, *Oluwadamilola Ayinla* and *Oluwatimilehin Ayodeji*, and
My Father, *Mubashir Abiodun*

ACKNOWLEDGEMENTS

I give all praises to the Almighty God for giving me the resources I needed to accomplish this work. For it is through Him alone that all things are possible.

I am very grateful to Drs. Adedeji B. Badiru, Myong K. Jeong, Fong-Yuen Ding, Dukwon Kim, and Wesley Hines for serving on my dissertation committee. I am especially grateful to Professor Badiru, for his supports throughout the doctoral program. I am thankful to Dr. Jeong for providing several valuable inputs and feedbacks throughout the research efforts and his supports during the final stage of the program. His experience in applied statistics and data mining helps achieve most of the results in this dissertation. I am also thankful to Dr. Hines for providing some of the datasets used for this research. I thank Drs. Ding and Kim for their respective assistance. I am grateful to other professors in the department for creating conducive environment for learning and for doing research. I am also grateful to the staff, Louise Sexton, Jeanette Myers, and Christine Tidwell, for their respective and collective administrative supports during the program.

I thank my parents for their prayers, encouragements, and understanding over all these years. I also thank my mother-in-law, Mrs. Janet Soyinka, for her prayers and supports since I have come to know her; I thank you so much, Iyalaje. I am very grateful to my beautiful and lovely wife, Remilekun, for all her sacrifices during my search for knowledge; Eny, I am sure the outcomes worth the sacrifices. I am also very grateful to

my adorable sons, Oluwadamilola (Dammy) and Oluwatimilehin (Timmy), for their understanding when I can not be where they wanted me to be and for their inquiries during the program, thank you kids for modeling my thoughts. I also thank my brothers, sisters, my wife's siblings for their respective encouragement.

I am thankful to my friends especially Yemisi Ogunade, Kunle Olunloyo, and John Ibitayo for being there for my family and me. I also thank our Mom in America, Mrs. Wanda King, and her daughters, Jeana and Sarah King, for their generosity and supports to my family. I also thank my wife's professor and mentor, Mrs. Beverly Tune, for her encouragements and supports. I am also thankful to other Nigerians in the Knoxville area for their companionship; they include Mr. and Mrs. Samuel Cole, Rasaq and Anota Ijaduola, Tokunbo and Bukola Ojemakinde, Gozie Nsofor, Tayo Obitayo, Kemi Obitayo, and Ruby Tasie. I say thanks to all members of the International Class at Cedar Springs Presbyterian Church including Paul and Jane Slay, David and Kari Cantrell, Mac and Beth Sells, and Chucks and Nancy Bowman. I also say thanks to Dr. David and Bridget Ajueyitsi, Biodun and Foluso Adeniyi, as well as all members of the Network of LASU Engineering Alumni (NOLEA). Finally, I am thankful to my former supervisors especially Mr. Olatunji A. Ogunlade, my past professors, my former students, and several others too numerous to mention for their encouragements over the years.

Olufemi A. Omitaomu
February 2006

Abstract

The recent advances in information technology, such as the various automatic data acquisition systems and sensor systems, have created tremendous opportunities for collecting valuable process data. The timely processing of such data for meaningful information remains a challenge. In this research, several data mining methodology that will aid information streaming of high-dimensional functional data are developed.

For on-line implementations, two weighting functions for updating support vector regression parameters were developed. The functions use parameters that can be easily set *a priori* with the slightest knowledge of the data involved and have provision for lower and upper bounds for the parameters. The functions are applicable to time series predictions, on-line predictions, and batch predictions. In order to apply these functions for on-line predictions, a new on-line support vector regression algorithm that uses adaptive weighting parameters was presented. The new algorithm uses varying rather than fixed regularization constant and accuracy parameter. The developed algorithm is more robust to the volume of data available for on-line training as well as to the relative position of the available data in the training sequence. The algorithm improves prediction accuracy by reducing uncertainty in using fixed values for the regression parameters. It also improves prediction accuracy by reducing uncertainty in using regression values based on some experts' knowledge rather than on the characteristics of

the incoming training data. The developed functions and algorithm were applied to feedwater flow rate data and two benchmark time series data. The results show that using adaptive regression parameters performs better than using fixed regression parameters.

In order to reduce the dimension of data with several hundreds or thousands of predictors and enhance prediction accuracy, a wavelet-based feature extraction procedure called step-down thresholding procedure for identifying and extracting significant features for a single curve was developed. The procedure involves transforming the original spectral into wavelet coefficients. It is based on multiple hypothesis testing approach and it controls family-wise error rate in order to guide against selecting insignificant features without any concern about the amount of noise that may be present in the data. Therefore, the procedure is applicable for data-reduction and/or data-denoising. The procedure was compared to six other data-reduction and data-denoising methods in the literature. The developed procedure is found to consistently perform better than most of the popular methods and performs at the same level with the other methods.

Many real-world data with high-dimensional explanatory variables also sometimes have multiple response variables; therefore, the selection of the fewest explanatory variables that show high sensitivity to predicting the response variable(s) and low sensitivity to the noise in the data is important for better performance and reduced computational burden. In order to select the fewest explanatory variables that can predict each of the response variables better, a two-stage wavelet-based feature extraction procedure is proposed. The first stage uses step-down procedure to extract significant features for each of the curves. Then, representative features are selected out of the extracted features for all curves using

voting selection strategy. Other selection strategies such as union and intersection were also described and implemented. The essence of the first stage is to reduce the dimension of the data without any consideration for whether or not they can predict the response variables accurately. The second stage uses Bayesian decision theory approach to select some of the extracted wavelet coefficients that can predict each of the response variables accurately. The two stage procedure was implemented using near-infrared spectroscopy data and shaft misalignment data. The results show that the second stage further reduces the dimension and the prediction results are encouraging.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

The availability of advanced information technologies such as the various types of automatic data acquisitions and sensor systems has created a tremendous capability to access valuable process data. The effective processing of this data remains the backbone of intelligent process monitoring and manufacturing. The need for more practical process monitoring models continues to grow as these technologies become more sophisticated. This chapter provides an introduction to the research. Section 1.1 presents a background for the research; the motivation for the research is discussed in Section 1.2. The contributions of the research are presented in Section 1.3. The organization of the rest of this dissertation is outlined in Section 1.4.

1.1 Background

The changing world of manufacturing is shifting process monitoring and control strategies. This is necessary since timely and accurate information about incipient faults or failures of processes and machines brings about a reduction in production costs by

reducing process downtime, avoiding overstocking of spare parts, improving productivity, enhancing product quality, and improving workers' learning curves. It also increases customers' reliability on prompt delivery of products. Effective process monitoring and control depends on some reliable, prompt, and accurate data processing techniques.

A process monitoring involves the use of data provided/collected during inspections or through sensors as basis for decision making. Such data has been classified into two groups: direct and indirect data (Christer and Wang, 1995). Direct data relates process conditions directly to the collected data; this data is called primary data because it does not require further processing. It provides a "what-you-see-is-what-you-get" kind of information. An example is the thickness of a brake pad. Indirect data, on the other hand, provides associated information that is influenced by the condition of the process. Such data requires further processing for a complete understanding of its meaning and implications; hence, this data is called secondary data. Some examples of secondary data are sensor data, vibration frequency analysis, and an oil analysis. Whatever the source of available data, the objective is to predict the immediate and/or subsequent conditions of the process and to use such data as input to modeling process control and management designs. A general procedure for process monitoring can be depicted as shown in Fig. 1.1. This procedure also applies to several other real-world applications. However, both primary and secondary data are integral part of sets of information that can be used as basis for subsequent control decisions.

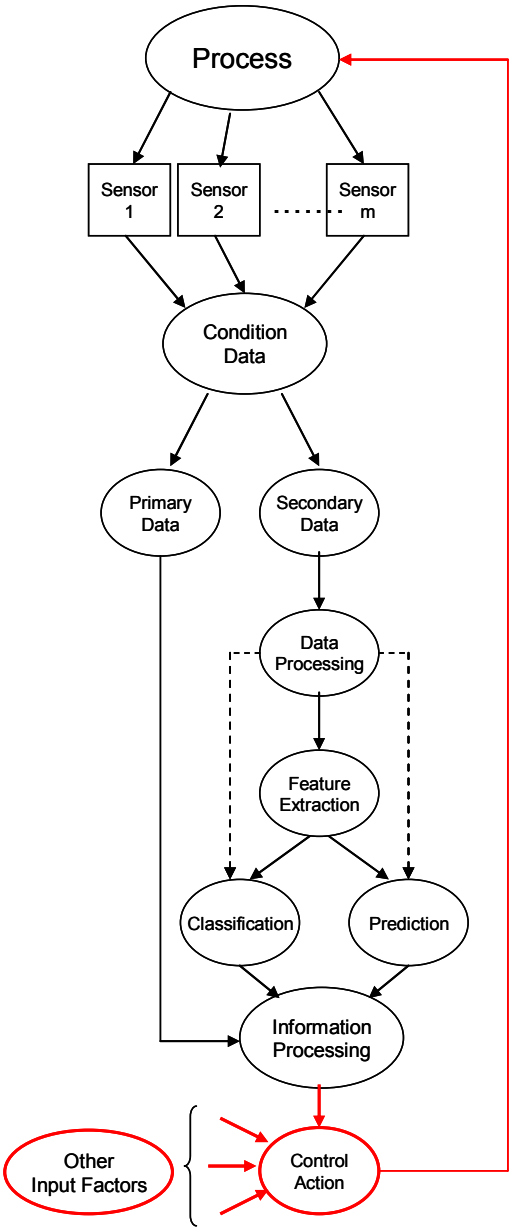

Fig. 1.1. A general procedure for process monitoring and control design.

Other factors that may influence control monitoring design and implementation include:

- The level of the required control and monitoring tasks (easy, difficult, or extremely difficult).
- The available resources (time, money, materials, and personnel).
- The significance of the affected process (downstream or upstream; high or low priority process).
- The production schedule, the anticipated downtime, and the allowable downtime.

These factors are depicted as other input factors in Fig. 1.1. Therefore, process monitoring and control design and implementation is a multidimensional process that can be represented as follows:

$$\mathbf{PDI} = f\left(\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_z\right), \tag{1.1}$$

where **PDI** is a process design and implementation factor and $\mathbf{F}_i$ represents vector of a quantitative measure for a set of **z** factors that characterize the system. Some of these factors are information processing (process monitoring) output, the available resources, the level of work required, and the anticipated and allowable downtime. In this research, one of these factors – **process monitoring** – is being considered.

Process monitoring is the tracking of all or specific events on processes, machines, or machine parts in order to deduce information that can be used to determine key performance indicators such as uptime, downtime, cycle times and in some cases replacement. Increasingly, manufacturing companies are striving to get accurate

information from production machines or processes in order to deduce factors that affect their margins and efficiencies. Given the importance of this inferred information it is vital that the processing of the secondary data is accurate and dependable.

Most real-world problems involve secondary data that are high-dimensional and functional in nature. High-dimensional problems are very challenging problems because there are more predictor variables than the sample size. The biscuit dough data discussed in this research has 256 predictors and 39 samples; the shaft misalignment problem also discussed in this research has 3072 predictors and 50 samples. These are examples of typical high-dimensional $\left(p \gg n\right)$ problems; these problems are also called small *n* large *p* problems. The technologies used for collecting data in these cases are usually expensive; this high cost constrains researchers to a few experimental units. The methods most commonly used for analyzing high-dimensional data fall into two classes: variable selection and factor-based methods (Brown et al., 2001). Two of these methods are principal components regression (PCR) (Cowe and McNicol, 1985) and partial least squares regression (PLS) (Wold et al., 1983); they are both widely used as standard approaches. The increasing power of computer has renewed interests in advanced techniques. Support vector regression is one of the new approaches for small *n* large *p* problems.

Furthermore, most of the secondary process data have observations that are curves or images. Curves and images are examples of functions and such observed curves and images are called "functional" data. Statistical methods for analyzing such data are

known as "functional data analysis" (FDA), which was coined by Ramsay and Dalzell (1991). The biscuit dough and the shaft misalignment problems are some examples of functional data. In addition, some of the high-dimensional functional data may have multiple response variables. The biscuit dough problem has four predictands (response variables) and the shaft misalignment problem has two predictands. Most classic methods were developed for single response variables. Their applications to multiple predictands require the implementation of the technique for each of the response variables without consideration for any relationship that may exist between each of the response variables and the predictors. These qualities make this type of data an important and a challenging data mining problem.

Therefore in developing the focus for this research, the following issues were considered:

a) How to *represent* secondary data in such a way that the features conserve the condition information essential for real-time decisions.
b) How to obtain a set of *parsimonious features* which are able to capture new information and also preserve the condition information in the original signals.
c) How to *reduce the uncertainty* in selecting subset features for prediction purposes.
d) How to effectively *update values of regression parameters* such that the profile of the incoming data plays a role in the computation and these values change as the profile changes.

The analysis of the secondary data can be used to determine the current/subsequent health status of a process or machine (diagnostics) and to predict the remaining useful life of a machine (prognostics). Therefore, the objective of the process control could be to prevent failure, to reduce risk, enhance productivity, or to rectify defects at a lesser cost.

Several quantitative and qualitative based models have been developed for process monitoring. The quantitative models are based on linear and nonlinear techniques including Principal Components Regression (PCR), Partial Least Squares (PLS), Artificial Neural Networks (ANN), and recently Support Vector Regression (SVR). Two of the common qualitative models are Expert Systems (ES) and Qualitative Trend Analysis (QTA). These techniques vary in their accuracy, prediction efficiency, robustness, and transparency. A comprehensive review of these techniques (except SVR) and their applications in condition monitoring is presented by Venkatasubramanian et al. (2003). In addition, an overview of PCR and PLS in condition monitoring was given by MacGregor et al. (1994), Geldi and Kowalski (1986), Hoskuldsson (1988), and Joliffe (2002). Omitaomu et al. (2006) recently present an application of support vector regression to shaft misalignment prediction using high-dimensional functional data.

1.2 Motivation

The motivations for this research are discussed in relation to the characteristics of process monitoring problems. The problems usually have the following characteristics:

1. Process data *arrive continuously in time* rather than in batches. Therefore, the data is more suitable for an on-line type of prediction rather than a batch type of prediction. This ensures availability of real time information for process control and improvement before serious damages could occur. This is true for time series data in which the prediction for time t_{n+1} is based upon the condition information obtained at time t_1 up to time t_n. It is also true for other applications where condition information at time t_n is based on information about other parameters also at time t_n. In both cases, real time information is required for prediction. The techniques mentioned in Section 1.1 are not suitable for on-line applications because the addition of a new sample to the already existing training samples requires that the technique retrain the entire training sample, which may require re-optimizing the value of the regression parameters.
2. Most real-world process data involves small sample size, high-dimensional explanatory variables, and multiple response variables. This is called multivariate small *n* large *p* problems. Most of the above techniques are applicable to problems with single response variable and their implementation for multiple response variable problems mean applying the technique to each of the response variable in sequence. This adds to the uncertainties in the developed models. Furthermore, some of the techniques require sample expansion or variable reduction in order to apply them. Sometimes, some of the explanatory variables may be highly correlated with each other and require special preprocessing.

3. The distribution of the incoming data changes over time in relation to the changes in the operating condition. Such changes are gradual in most cases and this leads to gradual changes in the relationship between the explanatory and response variables. Therefore, the prediction technique used should account for this gradual change in relationship. Most of the above mentioned techniques do not have provision for such implementation. This may be due to the fact that they were not developed for on-line prediction. This is another source of uncertainty in using those techniques.
4. Typically, process data is susceptible to noise. The use of noisy data results in biased models. The application of the above techniques requires filtering the noise in the data. The filtering process can be expensive especially for high-dimensional functional data. All the techniques mentioned above expect SVR do not have inherent property of avoiding the noise in the data. Even in SVR, an estimation of the noise in the data is necessary to guarantee correct application of this feature. However, such estimation may not be possible or accurate especially in on-line applications.

As a result of the inherent properties of SVR for small *n* large *p* problems, Ma et al. (2003) present accurate on-line support vector regression (AOSVR) algorithm. The approach combines the advantage of the conventional batch support vector regression with the capability of efficiently updating trained support vectors whenever a sample is added to or removed from the training set (Ma et al., 2003) without retraining the entire training data. The application of AOSVR technique in rotary machinery using high-

dimensional dataset was recently investigated (Omitaomu et al., 2006). However, AOSVR algorithm uses fixed value for regression parameters even though the distribution of the incoming data is not fixed. It does not take the position of the available data into consideration in predicting the response variable. That is, the same weight is use for all data samples as if they have equal role in the prediction model. Therefore, in order to use AOSVR for on-line prediction and reduce some of these uncertainties there must be a technique for updating the values of SVR parameters. There also must be a process of incorporating the non-stationary characteristics of the training data into this technique. Furthermore, there must be a procedure for selecting explanatory variables with high reliability to predict multiple response variables at the same time.

Furthermore, the sparseness property of the transformation method used would play a major role in order to reduce the dimension of the secondary data without loss of significant information. One transformation method that has been found outstanding because of its several properties is wavelet transforms. However, most of the wavelet-based feature extraction procedures in the literature are developed without a control of any error rate. Furthermore, the process of extracting significant features does not account for the relationship between these features and the response variables. Accounting for such relationship, however, in high-dimensional problems may not be feasible. Therefore, there is a need for a two-stage feature extraction procedure that will achieve dimension reduction and select only features that could explain a greater amount of the variability in each of the response variables.

1.3 Contributions of the Dissertation

Based on the issues in Section 1.1 and the motivations in Section 1.2, the contributions of this dissertation are:

1. A new methodology is developed for updating SVR parameters in relation to the changing characteristics of the training data (that is, by incorporating the non-stationary property of the training data into the prediction models).
2. The performance of the developed weighting functions is assessed in order to determine their effective and the type of data they can be used for.
3. A new on-line support vector regression algorithm that uses the weighting techniques developed in (1) above is also developed; therefore, the new algorithm uses adaptive regression parameters rather than fixed parameters.
4. The new algorithm is demonstrated using feedwater flow rate data and time-series data and the experimental results are discussed.
5. A novel wavelet-based procedure for extracting features in high-dimensional data for single curve and for multiple curves in order to achieve better computational efficiency, enhance compactness in data representation, and minimize relative similarity in variable vector is also developed.
6. An application of the new feature extraction procedures to some popular simulated signals in the literature, the biscuit dough problem, and the shaft misalignment problem is implemented. In addition, the predictions of the procedure compared with the other procedures in the literature are also discussed.

7. Finally, a new two-stage wavelet-based feature extraction methodology for high-dimensional data with multiple response variables is presented and the application of this methodology for the biscuit dough and shaft misalignment data is also discussed.

1.4 Outline of the Dissertation

The remainder of this dissertation is organized as follows: Chapter 2 reviews relevant literature on SVR, methods of computing SVR parameters, and types of kernels used in SVR. In addition, the descriptions of the shaft misalignment and biscuit dough problems are given in Chapter 2.

In Chapter 3, the two weighting functions, modified logistic weight function (MLWF) and modified Gompertz weight function (MGWF), for updating regularization constant and accuracy parameter for on-line support vector regression algorithm are presented. To use the proposed weighting functions, the on-line support vector regression algorithm with adaptive weighting parameters (AOLSVR) is also presented. Furthermore, the results of the application of these weighting functions and AOLSVR algorithm to both feedwater flow rate and time-series problems are discussed.

A wavelet-based feature extraction methodology for multivariate small n large p problems is presented in Chapter 4. The step-down thresholding (SDT) procedure using multiple hypotheses testing approach for extracting significant features for single curve and the voting selection strategy for extracting significant representative features for

multiple curves are also presented. In addition, a two-stage wavelet-based predictive modeling methodology for multivariate process data is also presented. The first stage uses SDT procedure for multiple curves and the second stage uses Bayesian decision theory approach to select some of the extracted features that can predict each of the response variables accurately. The proposed procedure was demonstrated with a shaft misalignment and biscuit dough data. The conclusions from this research and recommendations for future research are discussed in Chapter 5.

Chapter 2

Formulation of Support Vector Regression and Description of Representative Problems

An introduction to support vector machines is given in Section 2.1 and a review of support vector regression formulation is presented in Section 2.2. The types of kernel functions used in SVR implementation are discussed in Section 2.3. Section 2.4 reviews some approaches of computing SVR parameters. In Section 2.5, the descriptions of the data-sets used in this research are presented.

2.1 Introduction to Support Vector Machines

The support vector machines (SVM) technique (Vapnik, 1995; 1998) is based on statistical learning theory and it is used for learning classification and regression rules from data (Osuna et al., 1997). When used for classification problems, the algorithm is called support vector classification (SVC) and when used for regression problems, the algorithm is support vector regression (SVR). Therefore in this dissertation, as found in the literature, SVM will be used to refer to both SVC and SVR. Unlike other predictive

model, the SVM attempts to minimize the upper bound on the generalization error based on the principle of structural risk minimization (SRM) rather than minimizing the training error. This approach has been found to be superior to the empirical risk minimization (ERM) principle employed in artificial neural network (Gunn, 1998; Vapnik et al., 1996). In addition, the SRM principle incorporates capacity control that prevents over-fitting of the input data (Bishop, 1995). The SVM technique has sound orientations towards real-world applications (Smola and Schölkopf, 2004); therefore, it is applicable to condition monitoring problems (Omitaomu et al., 2006, 2005a, and 2005b).

The SVM technique continues to gain popularity for prediction because of its several outstanding properties (Muller et al., 1997; Fernandez, 1999; Cao and Tay, 2003). Some of these properties are the use of kernel function that makes the technique applicable to both linear and non-linear approximations, good generalization performance as a result of the use of only the so-called support vectors for prediction, the absence of local minima because of the convexity property of the objective function and its constraints, and the fact that it is based on structural risk minimization that seeks to minimize the upper bound of the generalization error rather than the training error. Since this research is concerned with regression problems, all the discussions henceforth will focus on SVR. Most of these discussions are also applicable to SVC. The SVR algorithm was developed after successful implementation of SVC algorithm for classification problems. The two key features in SVR implementation are mathematical programming and kernel functions. The model coefficients are obtained by solving a quadratic programming problem with linear equality and inequality constraints. The SVR technique has been

applied successfully to a wide range of pattern recognition and prediction problems (for example, Omitaomu et al., 2006, 2005a, & 2005b; Mattera & Haykin, 1999; Muller et al., 1999).

2.2 Support Vector Regression Formulation

A detailed formulation of SVR equations is provided by Vapnik (1995; 1998). Given a set of training inputs:

$$X = \{x_1, x_2, ..., x_m\} \subset \chi \tag{2.1}$$

and their corresponding outputs:

$$Y = \{y_1, y_2, ..., y_m\} \subset \gamma. \tag{2.2}$$

The training set, *T*, can then be represented by:

$$T = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}, \tag{2.3}$$

where $x \in X \subset \mathbb{R}^n$ and $y \in Y \subset \mathbb{R}$. Assume a non-linear function, $f(x)$, given by:

$$f(x) = \mathbf{w}^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}_i) + b, \tag{2.4}$$

where $\mathbf{w}$ is the weight vector, *b* is the bias, and $\mathbf{\Phi}(\mathbf{x}_i)$ is the high dimensional feature space, which is linearly mapped from the input space *x*. Assume further that the goal is

to fit the data *T* by finding a function $f(x)$ that has a largest deviation ε from the actual targets y_i for all the training data *T*, and at the same time it is as small as possible. Therefore, Eq. (2.4) is transformed into a constrained convex optimization problem as follows:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \\ &\text{subject to:} \quad \begin{cases} y_i - \left(\mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b\right) \le \varepsilon \\ y_i - \left(\mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b\right) \ge \varepsilon, \end{cases} \end{aligned} \tag{2.5}$$

where $\varepsilon(\ge 0)$ is user defined and represents the largest deviation. Eq. (2.5) can also be written as:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \\ &\text{subject to:} \quad \begin{cases} y_i - \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) - b \le \varepsilon \\ \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b - y_i \le \varepsilon. \end{cases} \end{aligned} \tag{2.6}$$

The goal of the objective function in Eq. (2.6) is to make the function as "flat" as possible; that is, to make $\mathbf{w}$ as "small" as possible while satisfying the constraints. In order to solve Eq. (2.6), slack variables are introduced to cope with possible infeasible optimization problems. One silent assumption here is that $f(x)$ actually exists; in other words, the convex optimization problem is *feasible*. However, this is not always the case; therefore, one might want to trade off errors by flatness of the estimate. This idea leads to the following primal formulations as stated in Vapnik (1995):

$$\text{minimize} \qquad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{m}\left(\xi_i^{+} + \xi_i^{-}\right)$$

$$\text{subject to:} \quad \begin{cases} y_i - \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{x}_i) - b \le \varepsilon + \xi_i^{+} \\ \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^{-} \\ \xi_i^{+}, \xi_i^{-} \qquad\qquad \ge 0, \end{cases} \tag{2.7}$$

where C (> 0) is a pre-specified regularization constant and represents the penalty weight. The first term in the objective function $\left(\mathbf{w}^{\mathbf{T}}\mathbf{w}\right)$ is the regularized term and makes the function as "flat" as possible whereas the second term $\left(C\sum_{i=1}^{m}\left(\xi_i^{+} + \xi_i^{-}\right)\right)$ is called the empirical term and measured the ε-insensitive loss function. According to Eq. (2.7), all data points whose *y*-values differ from $f(x)$ by more than ε are penalized. The slack variables, ξ_i^{+} and ξ_i^{-}, correspond to the size of this excess deviation for upper and lower deviations, respectively, as represented graphically in Fig. 2.1. The ε-tube is the largest deviation and all the data points inside this tube do not contribute to the regression model since their coefficients are equal to zero. Data points outside this tube or lying on this tube are used in determining the decision function and they are called support vectors and have non-zero coefficients. Eq. (2.7) assumes ε-insensitive loss function (Vapnik, 1995) as shown in Fig. 2.1 and defined as:

$$\left|\xi\right|_{\varepsilon} = \begin{cases} 0 & \text{if } \left|\xi\right| \le \varepsilon \\ \left|\xi\right| - \varepsilon & \text{otherwise.} \end{cases} \tag{2.8}$$

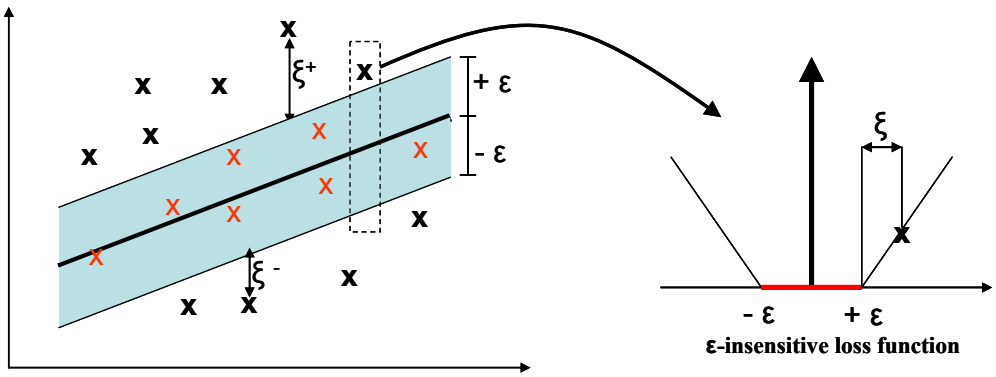

Fig. 2.1. An illustration of ε-tube for support vector regression.
(Adopted from Vapnik, 1998)

To solve Eq. (2.7), some Lagrangian multipliers $\left(\alpha_i^+, \alpha_i^-, \eta_i^+, \eta_i^-\right)$ are introduced in order to eliminate some of the primal variables. Hence, the Lagrangian of Eq. (2.7) is given as:

$$\begin{aligned} L_p = & \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{m}\left(\xi_i^+ + \xi_i^-\right) - \sum_{i=1}^{m}\left(\eta_i^+\xi_i^+ + \eta_i^-\xi_i^-\right) \\ & - \sum_{i=1}^{m}\alpha_i^+\left(\varepsilon + \xi_i^+ - y_i + \mathbf{w}^{\mathrm{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right) + b\right) \\ & - \sum_{i=1}^{m}\alpha_i^-\left(\varepsilon + \xi_i^- + y_i - \mathbf{w}^{\mathrm{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right) - b\right) \\ & s.t. \quad \alpha_i^+, \alpha_i^-, \eta_i^+, \eta_i^- \geq 0. \end{aligned} \tag{2.9}$$

Two advantages of Eq. (2.9) are that it provides the key for extending SVM to nonlinear functions and it makes solving Eq. (2.7) easier. It then follows from the saddle point condition (the point where the primal objective function is minimal and the dual objective function is maximal) that the partial derivatives of L_p with respect to the primal variables $\left(\mathbf{w}, b, \xi_i^+, \xi_i^-\right)$ have to vanish for optimality.

Therefore,

$$\partial_b L_P = \sum_{i=1}^{m} \left(\alpha_i^+ - \alpha_i^- \right) \qquad = 0, \tag{2.10}$$

$$\partial_{\mathbf{w}} L_P = \mathbf{w} - \sum_{i=1}^{m} \left(\alpha_i^+ - \alpha_i^- \right) x_i \quad = 0, \tag{2.11}$$

$$\text{and} \quad \partial_{\xi_i^{(*)}} L_P = C - \alpha_i^{(*)} - \eta_i^{(*)} \qquad = 0, \tag{2.12}$$

where (*) denotes variables with + and - superscripts. Substituting (2.10) and (2.12) into (2.9) lets the terms in b and ξ vanish. In addition, Eq. (2.12) can be transformed into $\alpha_i \in [0, C]$. Therefore, substituting Eqs. (2.10) to (2.12) into (2.9) yields the following dual optimization problem:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2} \sum_{i,j=1}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right) \left(\alpha_i^+ - \alpha_i^- \right) \left(\alpha_j^+ - \alpha_j^- \right) \\ & + \varepsilon \sum_{i=1}^{m} \left(\alpha_i^+ + \alpha_i^- \right) - \sum_{i=1}^{m} y_i \left(\alpha_i^+ - \alpha_i^- \right) \\ \text{subject to} \quad & \begin{cases} \sum_{i=1}^{m} \left(\alpha_i^+ - \alpha_i^- \right) & = 0 \\ \alpha_i^+, \alpha_i^- & \in [0, C], \end{cases} \end{aligned} \tag{2.13}$$

where $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is called the kernel function. The flexibility of a kernel function allows the technique to search a wide range of solution space. The kernel function must be positive definite in order to guarantee a unique optimal solution to the quadratic optimization problem. It allows non-linear function approximations with the SVM

technique, while maintaining the simplicity and computational efficiency of linear SVM approximations. Some of the common kernel functions are polynomial kernel and Gaussian radial basis function kernel. Descriptions of the common types of kernel used in SVM are discussed in Section 2.3.

The dual problem in Eq. (2.13) has three advantages: the optimization problem is now a quadratic programming problem with linear constraints, which is easier to solve and ensures a unique global optimum. Second, the input vector only appears inside the dot product, which ensures that the dimensionality of the input space can be hidden from the remaining computations. That is, even though the input space is transformed into a high dimensional space, the computation does not take place in that space but in the linear space (Gunn, 1998). Finally, the dual form does allow the replacement of the dot product of input vectors with a non-linear transformation of the input vector. In deriving Eq. (2.13), the dual variables η_i^+, η_i^- were already eliminated through the condition in Eq. (2.12). Therefore, Eq. (2.11) can be rewritten as:

$$\mathbf{w} = \sum_{i=1}^{m} \left(\alpha_i^+ - \alpha_i^- \right) x_i. \tag{2.14}$$

Hence, Eq. (2.4) becomes:

$$f(x) = \sum_{i=1}^{m} \left(\alpha_i^+ - \alpha_i^- \right) K\left(\mathbf{x}_i, \mathbf{x}_j \right) + b. \tag{2.15}$$

This is the Support Vector Regression expansion. That is, **w** can be completely described as a linear combination of the training patterns $\mathbf{x}_i$. Some outstanding advantages of Eq. (2.15) are that it is independent of both the dimensionality of the input space χ and the sample size *m*.

Like the PCR and PLS, the SVR expansion is not suitable for on-line prediction, because the addition of a data point requires the retraining of the entire training set. As a result, Ma et al. (2003) proposed accurate on-line support vector regression (AOSVR). Several approximate on-line SVM algorithms have previously been proposed (Li and Long, 1999; Cauwenberghs and Poggio, 2001; Csato and Opper, 2001; Gentile, 2001; Graepel et al., 2001; Herbster, 2001; Kivinen et al., 2002). The procedure involved in AOSVR is that whenever a new sample is added to the training set, the corresponding coefficient is updated in a finite number of steps until it meets the KKT conditions, while at the same time ensuring that the existing samples in the training set continue to satisfy the KKT conditions at each step. A detailed presentation of AOSVR can be found in Ma et al. (2003). However, this on-line algorithm uses a fixed value for ε and *C*; even though, the characteristic of the training set is not stationary. Therefore, to use varying (adaptive) values for SVR parameters (*C* and ε), there must be some modifications to the original algorithm and there must also be some procedures for updating the values for SVR parameters at each step of training.

2.3 Types of Kernel Functions used in SVM

$K\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is defined in Eq. (2.13) as the kernel function. Its value is equal to the inner product of two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space $\mathbf{\Phi}\left(\mathbf{x}_i\right)$ and $\mathbf{\Phi}\left(\mathbf{x}_j\right)$. That is,

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{\Phi}\left(\mathbf{x}_i\right) \cdot \mathbf{\Phi}\left(\mathbf{x}_j\right). \tag{2.16}$$

Therefore, the SVM techniques use a kernel function to map the input space into a high-dimensional feature space through some non-linear mapping chosen *a priori* and used to construct the optimal separating hyperplane in the feature space. This makes it possible to construct linear decision surfaces in the feature space instead of constructing non-linear decision surfaces in the input space. There are several types of kernel function used in SVM. The type of SVM constructed is a function of the selected kernel function. This also affect the computation time of implementing the SVM. According to Hilber-Schmidt theory, $K\left(x_i, x_j\right)$ can be any symmetric function satisfying the following conditions (Courant and Hilbert, 1953):

Mercer Conditions: *To guarantee that the symmetric function* $K\left(x_i, x_j\right)$ *has the expansion*

$$K\left(x_i, x_j\right) = \sum_{k=1}^{\infty} \alpha_k \psi_k\left(x_i\right) \psi_k\left(x_j\right) \tag{2.17}$$

with positive coefficients $\alpha_k > 0$ *(that is,* $K(x_i, x_j)$ *describes an inner product in some feature space), it is necessary and sufficient that the condition*

$$\iint K(x_i, x_j) g(x_i) g(x_j) dx_i dx_j > 0 \qquad (2.18)$$

be valid for all $g \neq 0$ *for which*

$$\int g^2(x_i) dx_i < \infty. \qquad (2.19)$$

Based on this theorem, three of the popular kernel functions used in SVM are:

- A polynomial kernel function constructed using:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (x_i, x_j)^d, \ d = 1, 2, 3, \cdots. \qquad (2.20)$$

An alternative form of Eq. (2.20), which avoids computation problem encountered in using Eq. (2.20) is:

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((x_i, x_j) + 1)^d, \ d = 1, 2, 3, \cdots. \qquad (2.21)$$

- A Gaussian radial basis kernel function can be constructed using:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(x_i - x_j)^2}{\sigma^2} \right), \qquad (2.22)$$

where σ (> 0) is the kernel width.

- A sigmoid kernel function constructed using:

$$K\left(x_i, x_j\right) = \tanh\left(b\left(x_i \cdot x_j\right) - c\right). \tag{2.23}$$

The Gaussian radial basis function kernels, usually called radial basis function (RBF) kernels in SVM literature, are widely used in artificial neural networks (Haykin, 1999), support vector machines (Vapnik, 1998), and approximation theory (Schölkopf & Smola, 2002). The RBF kernel is usually a reasonable first choice because of its outstanding features: it can handle linear and non-linear input-output mapping effectively; it requires less number of hyper-parameters than polynomial kernel, which reduces computation cost in terms of tuning for optimum hyper-parameters; the kernel values for RBF ranges between 0 and 1, hence less numerical difficulties; whereas these values can range between 0 and infinity for polynomial kernel. The sigmoid kernel is not always considered because it does not always fulfill the Mercer Condition (Vapnik, 2000), which is a requirement for an SVR kernel. In addition, sigmoid kernel is similar to RBF kernel when the kernel width is a small value (Lin & Lin, 2003). Therefore, because of its outstanding advantages, the RBF kernel is adopted for all analyses and computations in this dissertation. A simple example of the kernel mapping from a two dimensional input space $\left(\mathbb{R}^2\right)$ into a six dimensional feature space $\left(\mathbb{R}^6\right)$ using polynomial kernel as defined in Eq. (2.21) is given in Example 2.1:

Example 2.1 (Quadratic features in $\mathbb{R}^2$):

$$\begin{aligned}\left(\left(x_i \cdot x_j\right)+1\right)^2 &= \left(x_{i1}x_{j1}+x_{i2}x_{j2}\right)^2 + 2\left(x_{i1}x_{j1}+x_{i2}x_{j2}\right)+1 \\ &= \left(x_{i1}x_{j1}\right)^2 + 2\left(x_{i1}x_{j1}\right)\left(x_{i2}x_{j2}\right)+\left(x_{i2}x_{j2}\right)^2 + \\ &\quad 2\left(x_{i1}x_{j1}\right)+2\left(x_{i2}x_{j2}\right)+1.\end{aligned}$$

Therefore,

$$K\left(\mathbf{x}_1,\mathbf{x}_2\right)=k\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{bmatrix}x_1^2 & x_2^2 & \sqrt{2}x_1 & \sqrt{2}x_2 & \sqrt{2}x_1x_2 & 1\end{bmatrix}^{\mathbf{T}}.$$

2.4 Methods of Computing SVR Parameters

The performance of SVR technique depends on the setting of three training parameters (kernel, *C*, and ε) for ε-insensitive loss function. However, for any particular type of kernel the values of *C* and ε are what affect the complexity of the final model. The value of ε affects the number of support vectors (SV) used for predictions. Intuitively, a larger value of ε results in a smaller number of support vectors, which leads to less complex regression estimates. On the other hand, the value of *C* is the trade off between model complexity and the degree of deviations allowed in the optimization formulation. Therefore, a larger value of *C* undermines model complexity (Cherkassky and Ma, 2004). The selection of optimum values for these training parameters (*C* and ε) that will guarantee less complex models is an active area of research. There are several existing approaches for selecting optimum value for these parameters.

The most common approach is based on users' prior knowledge or expertise in applying SVM techniques (Cherkassky and Mulier, 1998; Schölkopf et al., 1999). However, this approach could be subjective and it is not appropriate for new users of SVR. It is also not applicable for on-line application since it requires manual intervention at each step of learning. This approach constitutes a source of uncertainty when used by non-experts and experts of SVM not familiar with the characteristics of the data set under consideration. Mattera and Haykin (1999) proposed that the value of C be equal to the range of output values; but this approach is not robust to outliers (Cherkassky and Ma, 2004), especially in condition monitoring problems where data is prone to outliers due to faulty sensors or instruments. Another approach is the use of cross-validation techniques for parameter selection (Cherkassky and Mulier, 1998; Schölkopf et al., 1999). Even though this is a good approach for batch processing, it is data-intensive; hence, it is very expensive to implement in terms of computation time, especially for larger datasets. Furthermore, re-sampling techniques are not applicable to on-line applications. One more approach is that ε values should be selected in proportion to the variance of the input noise (Smola et al., 1998; Kwok, 2001); this approach is independent of the sample size and it is only suitable for batch processing where the entire data set is available. Cherkassky and Ma (2004) presented another approach based on the training data. They proposed that C values should be based on the training data without resulting to re-sampling using the following estimation:

$$C = \max\left(\left|\overline{y} + 3\sigma_y\right|, \left|\overline{y} - 3\sigma_y\right|\right), \tag{2.24}$$

where $\bar{y}$ and σ_y are the mean and standard deviation of the *y* values of the training data. One advantage of this approach is that it is robust to possible outliers. They also proposed that the value of ε should be proportional to the standard deviation of the input noise. Using the idea of Central Limit Theorem, they proposed that ε be given by:

$$\varepsilon = 3\sigma\sqrt{\frac{\ln n}{n}}, \tag{2.25}$$

where σ is the standard deviation of the input noise and *n* is the number of training samples. Since the value of σ is not known *a priori*, the following equation can be use to estimate σ using the idea of *k*-nearest-neighbor's method:

$$\hat{\sigma} = \sqrt{\frac{n^{1/5}k}{n^{1/5}k-1}\cdot\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2}, \qquad 2\le k\le 6, \tag{2.26}$$

where *n* is the number of training samples, *k* is the low-bias/high variance estimators, and $\hat{y}$ is the predicted value of *y* by fitting a linear regression to the training data to estimate the noise variance. Again, this approach is only applicable to batch processing. Cao and Tay (2003) proposed ascending regularization constant $\left(C_i\right)$ and descending tube $\left(\varepsilon_i\right)$ for batch SVR applications in financial time series data. They adopt the following definitions:

$$C_i = C\frac{2}{1+\exp\left(a-2a\left(\frac{i}{m}\right)\right)} \tag{2.27}$$

and

$$\varepsilon_i = \varepsilon \frac{1+\exp\left(b-2b\left(\frac{i}{m}\right)\right)}{2}, \tag{2.28}$$

where i represents the data sequence, C_i is the ascending regularization constant, ε_i is the descending tube, a is the parameter that controls ascending rate, and b is the parameter that controls descending rate. Even though their approach is adaptive, it is not suitable for on-line learning and its application to AOSVR will damage the on-line algorithm because the computation of the parameter that controls the ascending and the descending rates also requires re-sampling techniques (Cao and Tay, 2003). Furthermore, their approach is not flexible regarding setting the lower and upper bound of the parameter values. Some of the approaches above are very appropriate to batch processing but they are not practical for on-line learning. Therefore, there is a need for appropriate approaches for computing SVR parameters in on-line settings.

2.5 Description of the Representative Problems

This section introduces the two representative high-dimensional problems consider in this research: shaft misalignment problem and biscuit dough problem.

2.5.1 Shaft Misalignment Problem

A typical mechanical system consists of a driver machine, a driven machine, and a coupling as depicted in Fig. 2.2.

The coupling could be a rotating shaft, rigid or elastic joints, belt and gear trains. A shaft transmission system is one of the most fundamental and important parts of rotating machinery; therefore, the ability to estimate and predict shaft alignment or misalignment accurately can significantly enhance the predictive maintenance task of a production system. A proper shaft alignment is indispensable because it reduces excessive axial and radial forces on the most vulnerable parts of a machine system such as the bearings, seals, and couplings (Wowk, 2000). It also minimizes the amount of shaft bending thereby permitting full transmission of power from the driver machine to the driven machine and eliminates the possibility of shaft failure from cyclic fatigue.

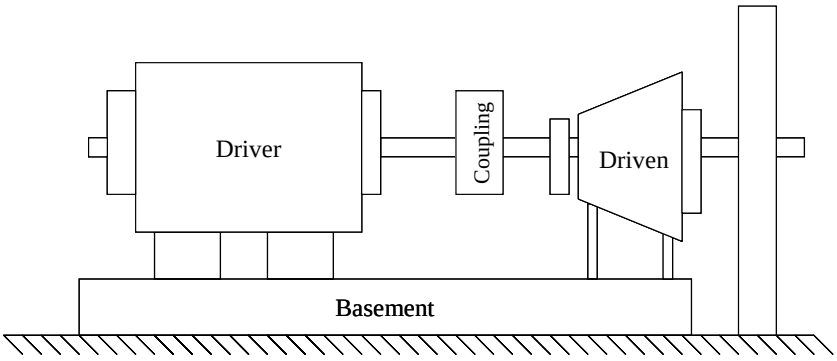

Fig. 2.2. A schematic diagram of a driver-coupling-driven system. (Adapted from Giordana et al., 1993)

In addition, it minimizes the amount of wear in the coupling components, reduces mechanical seal failure, and lowers vibration levels in machine casings, bearing housings, and rotor. Therefore, monitoring and predicting shaft alignment condition is important in order to make intelligent decisions on when to perform alignment maintenance, which plays an essential role in increasing maintenance effectiveness and reducing maintenance costs.

Shaft misalignment is one of the prevalent faults associated with rotating machines and it occurs when the shaft of the driven machine and the shaft of its driver machine do not rotate on a common axis; that is, the shafts are not coaxial. Shaft misalignment is a measure of how far apart the two centerlines are away from each other (Wowk, 2000; Kuropatwinski et al., 1997). Such shift in centers can be in parallel position (when the centerlines of the two shafts are parallel with each other, but at a constant distance apart), in angular position (when the centerlines are at an angle to each other), or a combination of these positions (Piotrowski, 1995) as depicted in Fig. 2.3.

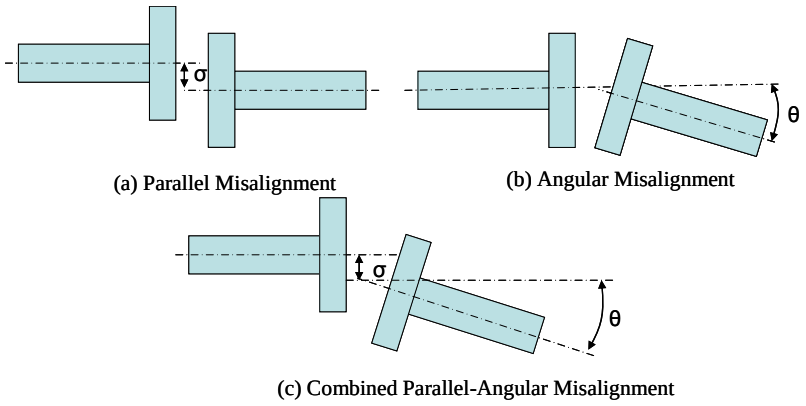

Fig. 2.3. An illustration of parallel, angular, and combined misalignments.

Even though a perfect shaft alignment is unlikely during operating cycles, there is a limit to the maximum amount of misalignment that is allowable. Therefore, shaft alignment can be classified into four grades: unsafe, poor, acceptable, and excellent. Two of these grades (poor and unsafe) are wake-up calls for maintenance actions. A poor grade results when the alignment is within the manufacturer's allowances but outside the machine's recommended limits, so it is somewhat a warning grade. An unsafe grade means the alignment is outside the manufacturer's design specification allowance and must be attended to.

Industry invests significant amount of time and money on shaft misalignment because it causes a decrease in motor efficiency and makes the machine more prone to failure due to increased loads on the shaft support devices such as the bearings, seals, and couplings. It is generally agreed that proper alignment is critical to the life of a machine, which means that coupling wear or failure, bearing failures, bent rotors, and bearing housing damage are direct evidence of poor shaft alignment (Eisenmann and Eisenmann, 1998). One way to overcome this problem is to monitor the condition of the shaft during machine operations and collect data to predict the state of the shaft. It has been estimated that more than 50% of all problems with coupled machines are due to misalignment and over 50% of all vibration troubles with coupled machines are due to shaft misalignment. Furthermore, up to 50 percent of the expected bearing life can be lost with as small as a 5 mil offset misalignment. Therefore, eliminating shaft misalignment is a major focus because it increases machine reliability, increases machine life, and reduces maintenance and cost. However, the detection and prediction of shaft misalignment requires a proven

technique with high performance capability especially in a system that alerts operators of impending faults so that corrections could be made before any damage to the motor system. Such notification would allow maintenance to be performed at scheduled shutdowns rather than creating unnecessary machine downtime.

Various quantitative based models have been used for predicting shaft misalignment conditions. The most popular techniques are Principal Components Regression (PCR), Partial Least Squares (PLS), and Artificial Neural Networks (ANN) (Kuropatwinski et al., 1997; Hines et al., 1997; 1998; 1999). These techniques vary in their accuracy, prediction efficiency, robustness, and transparency. The performance of PLS depends on the number of factors used. The degree of bias is dependent on the choice of the number of factors. A smaller number of factors result in larger bias but a large number of factors will produce high variance. In the case of non-linear relationships between predictors and response, PLS might results in a solution with large bias. Therefore, ANN is usually considered and used for non-linear relationships. However ANN has two disadvantages, a black box approach and a computationally expensive training process (Haykin, 1999), which make its acceptance and implementation in the industry difficult. The purpose of using PCR model is to find factors that have a much lower dimension than the original data set, which can be used for prediction; thereby reducing computation time and avoiding the use of correlated features. One limitation of this technique is that it is time invariant, while most of the real processes are time-varying (Venkatasubramanian, 2003).

The objective of this study is to predict motor shaft misalignment from the motor's power spectrum using a modified support vector regression approach. The motivation for using

motor power spectrum is that electric motors generate a force that turns the motor shaft through a coupling. Deviations in the motor shaft system cause immediate changes to the input power of the motor. A system based on power changes is more sensitive than one based on vibration changes because the information is immediately transferred instead of having to be passed through the different structures of the motor.

The original shaft misalignment data has 3072 predictors, 10 samples, and two response variables (parallel and angular misalignment conditions). Five sets of the original data were taken to increase the number of samples to 50. The original data set was duplicated to allow for more observations for each alignment condition. This new data set includes 10 samples of combined parallel and angular misalignment conditions and 20 samples each of parallel and angular misalignment conditions. Fig. 2.4 shows a 3D plot of the dataset for the original 10 conditions and the first 500 predictors and a 2D plot for three of the curves. The red colored signal in the 2D plot is for an angular misalignment condition, the blue colored signal is for a parallel misalignment condition, and the black colored signal is for a combined misalignment condition. These plots show that there is no significant difference between the data points; therefore, all the predictors can be said to be significant. However, there may be internal characteristics that are not feasible in this domain. Transforming the data into another domain may reveal these internal characteristics.

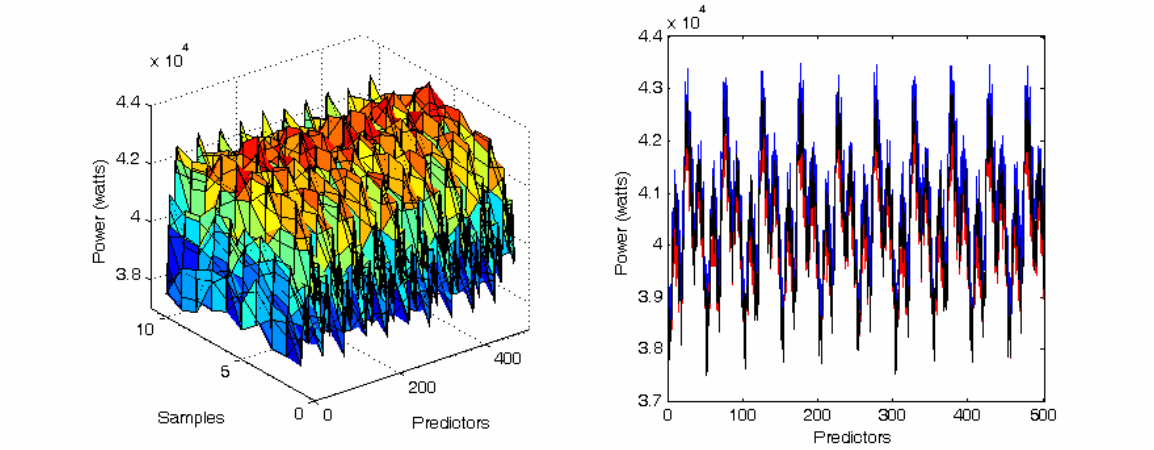

Fig. 2.4. Plots of shaft misalignment data.

2.5.2 Chemical Composition Problem

Quantitative near-infrared (NIR) spectroscopy is used to analyze the chemical composition or biological properties of samples and has applications in several industries including food and drink, medicine, geology, paper, petrochemical, pharmaceutical, and biotechnology (Osborne et al., 1993). The example studied in this dissertation arises from an experiment to measure the composition of biscuit dough pieces for possible on-line implementation (Brown et al., 2001). The purpose of the work was to investigate if NIR would be of practical benefit in the monitoring of automatic metering equipment used for fat, dry flour, sugar, and water in short dough biscuit production (Osborne et al., 1984). Two similar sample sets (training and testing sets) with the standard recipe varied to provide a large range for each of the four constituents under investigation: fat, sucrose, fry flour, and water. Thus, there are four predictands and 39 samples for training and

testing. For each sample, there are 700 points measured from 1100 to 2498 nanometers (nm) in steps of 2 nm. However, for analysis, only data points from 1380 nm and 2400 nm in steps of 4 nm are used. Therefore, the number of predictors is 256. Fig. 2.5 shows a 3D plot of the dataset and a 2D plot for one of the curves. The red colored signal in the 2D plot is for the first sample, the blue is for the 15th sample, and the black colored signal is for the 34th sample. Again, these curves have slight differences in the original domain and it may be difficult to select the significant samples in this domain.

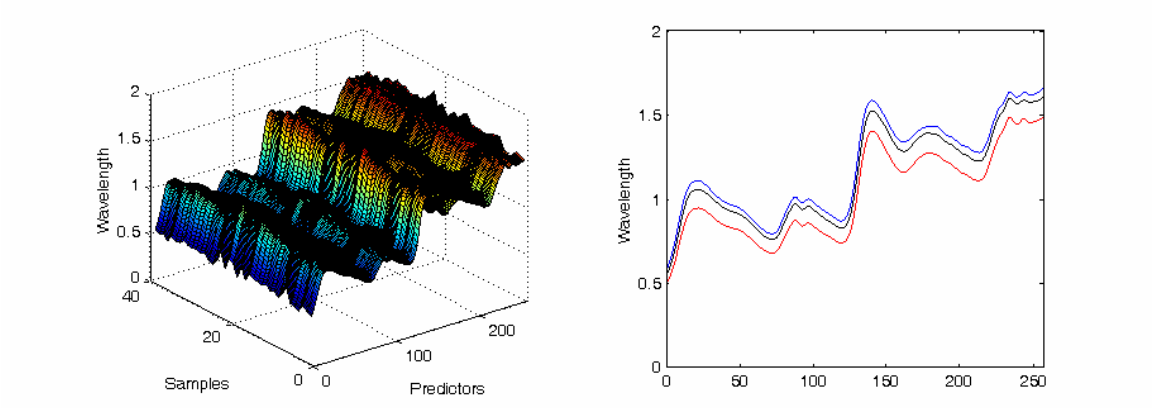

Fig. 2.5. Plots of biscuit dough spectroscopy data.

Chapter 3

On-Line Support Vector Regression with Adaptive Weighting Parameters

An introduction to weighting functions for on-line prediction is presented in Section 3.1. The proposed weighting functions for on-line prediction, modified logistic weight function (MLWF) and modified Gompertz weight function (MGWF), are presented in Sections 3.2 and 3.3 respectively. Descriptions of some potential areas of application of the weighting functions are presented in Section 3.4. The modified AOSVR algorithm called on-line support vector regression with adaptive weighting parameters (AOLSVR) algorithm is presented in Section 3.5. Applications of AOLSVR using MLWF and MGWF for two spectra problems and two time-series benchmark data are presented in Section 3.6.

3.1 Introduction to Weighting Functions for On-line Prediction

For on-line learning, one approach of selecting C and ε is to vary their values with respect to the relative importance of the training samples. In general, for on-line

predictions, recent data points provide more quality information than past data points. Furthermore, recent data points are more critical to prediction, especially for condition monitoring applications and financial time series data. Process condition information increases monotonically over time starting from a zero level. Therefore, recent data should be given more attention (more weight) than distant data points. To extend the application of adaptive regularization constant and tube to on-line learning and to enhance the application of AOSVR, two simple weighting functions for computing SVR parameters for on-line applications are proposed.

Another motivation for these weighting schemes is that the rate at which learning evolves can be divided into three phases: an initialization phase, a progress phase, and a stable phase. The initialization phase is the starting phase when we just start gathering data but we have not enough data for stable learning. The prediction error for this phase is usually very high. After some rounds of learning, we have the progress phase. This is when learning picks up as a result of the availability of more data and we expect the prediction error to start decreasing at a faster rate. That is, the addition of one sample of data brings about a significant reduction in prediction error. The last phase is the stable phase and this occurs at a point when we have had enough data for learning and can confidently say that the learning is somewhat stable. At this phase, the prediction error is also stable and there is no significant improvement in prediction error for additional training samples. These phases are as depicted in Fig. 3.1, where $i = 1, \ldots, n$ and n is the sample size. From Fig. 3.1, as the sample size increases the prediction error decreases until the learning becomes stable. In on-line learning, we want to achieve a stable prediction as soon as

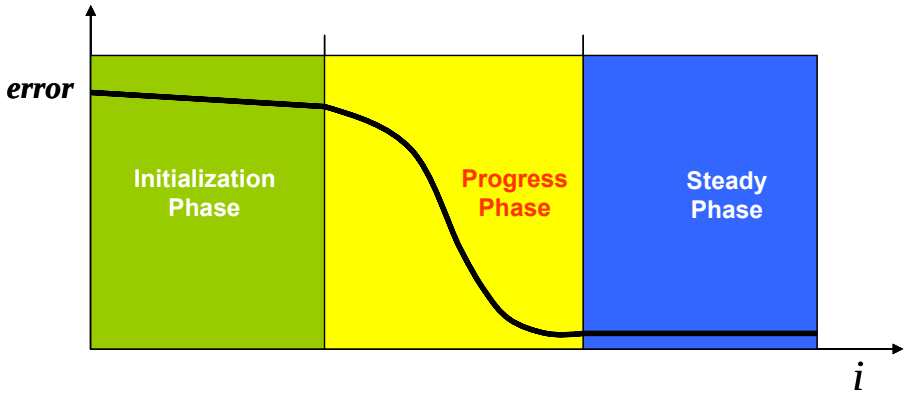

Fig. 3.1. An illustration between training error and sample size

possible since training data is usually sparse. Using a weight function may help achieve this objective without using a large training set. Based on this description, if recent samples are more important than past samples the value of the regularization constant should increase over time as the number of data sequence increases. On the other hand, the accuracy parameter should decrease over time as the number of data sequence increases. In addition, in using SVM algorithms the value of the regularization constant should be much greater, possibly tending towards infinity and the value of the accuracy parameter should be much smaller, possibly tending towards zero depending on the estimate of noise in the data. Therefore, two sigmoidal weight functions for on-line learning are proposed: modified Logistic weight function (MLWF) and modified Gompertz weight function (MGWF).

3.2 Modified Logistic Weight Function

One of the most popular classical symmetric functions that use only one equation is the Logistic function. It has wide applications in several areas including engineering, natural sciences, and statistics. It combines two characteristic exponential growth (exponential and bounded exponential). Logistic function has been widely used in neural network as the preferred activation function, because it has good properties for updating training weights. However, the standard form of Logistic function is not flexible in setting lower and upper bounds on weights. For time-dependent predictions, it is reasonable to have a certain initial weight at the beginning of the training. Therefore, in order to extend the properties of Logistic function to estimating SVR parameters, we propose modified Logistic weight function (MLWF) equations for adaptive SVR parameters. The adaptive regularization constant (C_i) is defined as:

$$C_i = C_{\min} + \left[\frac{C_{\max}}{1 + \exp\left(-g \times \left(i - m_c\right)\right)} \right] \tag{3.1}$$

and the MLWF equation for adaptive accuracy parameter (ε_i) is defined as:

$$\varepsilon_i = \varepsilon_{\min} + \left[\frac{\varepsilon_{\max}}{1 + \exp\left(g \times \left(i - m_c\right)\right)} \right], \tag{3.2}$$

where $i = 1, \ldots, m$, m is the number of training samples, m_c is the changing point, $C_{\min}$ and $\varepsilon_{\min}$ are the desired lower bound for the regularization constant and the accuracy

parameter respectively, $C_{\max}$ and $\varepsilon_{\max}$ are the desired upper bound for the regularization constant and the accuracy parameter respectively, and g is an empirical constant that controls the curvature (slope) of the function; that is, it represents the factor for the relative importance of the samples. The essence of the lower bound is to avoid underestimation and the upper bound avoids overestimation of the parameters. The value of g could range from zero to infinity but we consider only four special cases in this dissertation. The behaviors of these four cases are summarized as follows:

i. **Constant weight**: This is the case with conventional AOSVR in which all data points are given the same weight. This is more suitable for data from the same process. This can be achieved in Eqs. (3.1) and (3.2) when $g = 0$, therefore $C_i \cong C_{\min} + C_{\max}/2$ and $\varepsilon_i \cong \varepsilon_{\min} + \varepsilon_{\max}/2$.

ii. **Linear weight**: This is applicable to cases in which the weight is linearly proportional to the size of the training set. This is the case when $g = 0.005,$ then the value of C_i is a linearly increasing relationship and the value of ε_i is a linearly decreasing relationship.

iii. **Sigmoidal weight**: Different sigmoidal pattern can be achieved using different values of g in relation to the number of training set. One possibility is when $g = 0.03,$ the weight function follows a sigmoidal pattern. The value of g can also be set to achieve a pattern with a zero slope at the beginning and at the end of the training.

iv. **Two distinct weights**: In this case, the first one-half of the training set is given one weight and the second one-half is given another weight. A possible application is the case of data from two different processes. This is possible when $g = 5,$ then,

$$C_i \cong \begin{cases} C_{\min}, & i < m_c \\ C_{\min} + C_{\max}, & i \geq m_c \end{cases} \quad \text{and} \quad \varepsilon_i \cong \begin{cases} \varepsilon_{\min} + \varepsilon_{\max}, & i < m_c \\ \varepsilon_{\min}, & i \geq m_c. \end{cases}$$

The pictorial representations of the different weights for these *g* values are shown in plots *a* and *b* in Fig. 3.2. Both plots a & b show that the profile for MLWF is symmetric around the mid point (m_c) of the total training set. For the plots in Fig. 3.2, the $C_{\min}$ and $C_{\max}$ are set to 5.0 and 60.0 respectively and $\varepsilon_{\min}$ and $\varepsilon_{\max}$ are set to 0.01 and 0.45 respectively for *m* = 300 and m_c = 150.

3.3 Modified Gompertz Weight Function

In order to generate asymmetric profiles for non-stationary process data, we present modified Gompertz weight function (MGWF) equations for SVR parameters. The asymmetric property is useful if we do not want a balanced weight profile as seen in case of MLWF. This is a double exponential function, but its behavior is similar to MLWF. The MGWF equation for adaptive regularization constant (C_i) is defined as:

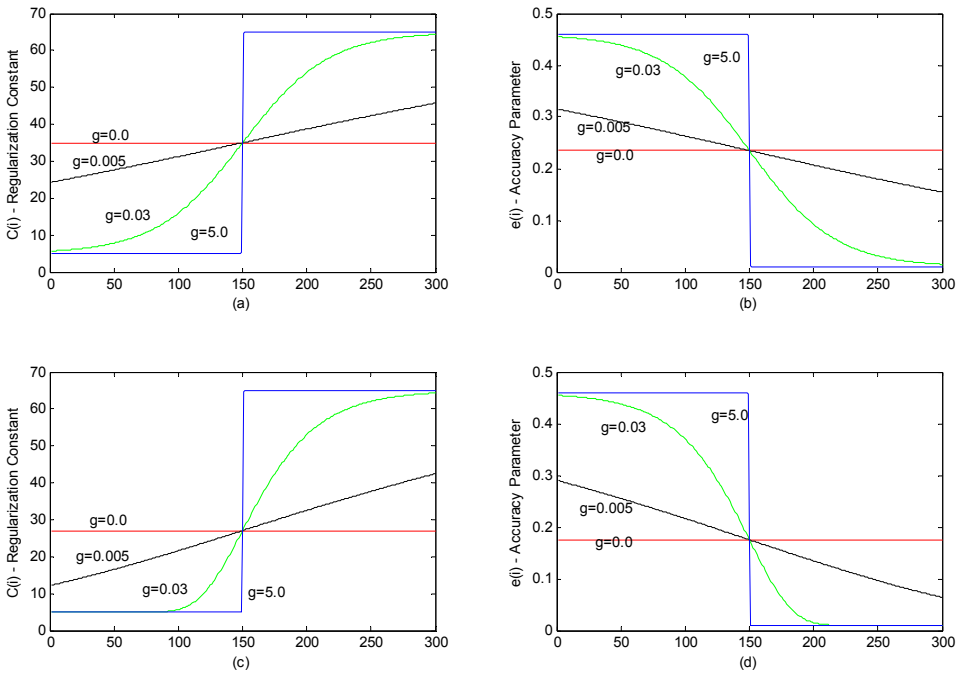

Fig. 3.2: The pictorial representations of MLWF and MGWF with different values of g.

$$C_i = C_{\min} + C_{\max}\left(\exp\left(-\exp\left(-g\times\left(i-m_c\right)\right)\right)\right) \tag{3.3}$$

and the MGWF function for adaptive accuracy parameter $\left(\varepsilon_i\right)$ is defined as:

$$\varepsilon_i = \varepsilon_{\min} + \varepsilon_{\max}\left(\exp\left(-\exp\left(g\times\left(i-m_c\right)\right)\right)\right), \tag{3.4}$$

where i, m_c, g, $C_{\min}$, $C_{\max}$, $\varepsilon_{\min}$, and $\varepsilon_{\max}$ have the same definitions and explanations as in Eqs. (3.1) and (3.2). As in Section 3.2, we also consider the same values of g for the four cases considered. The behaviors of Eqs. (3.3) and (3.4) for the four cases considered

is summarized as follows and their pictorial representations are shown in plots *c* and *d* in Fig. 3.2.

1. **Constant weight**: When $g = 0$, $C_i \cong C_{\min} + C_{\max}/e$ and $\varepsilon_i \cong \varepsilon_{\min} + \varepsilon_{\max}/e$. That is, a fixed value is used for all data points.
2. **Linear weight**: When $g = 0.005$, the value of C_i is a linearly increasing relationship and the value of ε_i is a linearly decreasing relationship.
3. **Sigmoidal weight**: When $g = 0.03$, the weight function follows a sigmoidal pattern.
4. **Two distinct weights**: When $g = 5$,

$$C_i \cong \begin{cases} C_{\min}, & i < m_c \\ C_{\min} + C_{\max}, & i \geq m_c \end{cases} \quad \text{and} \quad \varepsilon_i \cong \begin{cases} \varepsilon_{\min} + \varepsilon_{\max}, & i < m_c \\ \varepsilon_{\min}, & i \geq m_c. \end{cases}$$

Both plots *c* and *d* in Fig. 3.2 show that the MGWF profile is asymmetric around the mid point (m_c) of the total training set.

For these applications, if the recent samples are more important than past samples, *g* must be greater than zero $(g > 0)$ as shown in Sections 3.2 and 3.3. Other values can also be used depending on the objectives of each problem. However, if past samples are more important than recent samples, then *g* must be less than zero $(g < 0)$ for all equations described in Sections 3.2 and 3.3; these cases are not considered in this dissertation.

Some of the advantages of these functions include:

- Selection of C and ε parameters directly in relation to the relative importance of data samples and/or their respective position in the data sequence without using re-sampling methods.
- Flexibility in setting lower and upper bound for the parameters without using re-sampling techniques.
- The use of a curvature control parameter that has only fewer possible values that will enhance *a priori* settings.
- The use of parameters that can be set with the slightest knowledge of the characteristics of the incoming data.

In this dissertation, the lower and the upper bounds are set empirically; however, there are techniques in the literature that can be used to compute these values. For example, Cherkassky and Ma (2004) presented an approach that is data dependent and robust to outliers.

3.4 Potential Areas of Application of the Weight Functions

The on-line weighting functions (MLWF and MGWF) can be used in several modeling cases; we generalized those cases into three:

1. The functions can be used in time series prediction where the prediction for time t_{n+1} is based upon the condition information obtained at the past times up to time t_n. In such cases, the most recent condition information is more valuable than distant condition information. Therefore, the most recent data has more weight than distant data in predicting the next condition. One possible example is in machine wear condition monitoring, the future wear of a machine part is dependent upon all past condition information and such dependency increases monotonically over time from somewhat perfect condition.
2. These functions are also applicable to other situations where condition information at time t_n is based on information about other parameters also at time t_n. In such cases, the volume of available data enhances the performance of the prediction. The more the data points available for prediction, the greater the accuracy of the prediction. Therefore, as the volume of available data increases, more weight is given to the data until the prediction is somewhat stable and the value of the parameters become constant.
3. They are also applicable to cases where there is enough data for prediction. In such cases, the learning rate is set so small so that all data points are given equal weight in the prediction. Instead of a sigmoidal pattern of weighting profile, the learning rate can also be set to achieve linear weighting profiles as shown in Fig. 3.2.

3.5 On-line SVR with Adaptive Weighting Parameters

Two weight functions are proposed in Section 3.3 for computing adaptive regularization constant and adaptive accuracy parameter. In order to use any of these proposed functions for on-line predictions, we modify both the empirical error (risk), which is measured by the ε-insensitive loss function, and the constraints of the AOSVR formulation, which will lead to a new set of KKT conditions. Therefore, the regularized constant adopts adaptive regularization constant C_i and every training sample uses adaptive accuracy parameter (different tube size) ε_i. The modified algorithm will compute the SVR parameters, $\left(C_i \text{ and } \varepsilon_i\right)$, as explained in Section 2; while it avoids re-training of the training set.

For AOLSVR, Eq. (2.6) of SVR becomes:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \\ &\text{subject to:} \quad \begin{cases} y_i - \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) - b \le \varepsilon_i \\ \mathbf{w}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{x}_i) + b - y_i \le \varepsilon_i, \end{cases} \end{aligned} \tag{3.5}$$

where ε_i is varying (adaptive) accuracy parameter. This idea leads to the following primal formulations:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \sum_{i=1}^{m} C_i\left(\xi_i^{+} + \xi_i^{-}\right) \\ &\text{subject to:} && \begin{cases} y_i - \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{x}_i) - b \le \varepsilon_i + \xi_i^{+} \\ \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{x}_i) + b - y_i \le \varepsilon_i + \xi_i^{-} \\ \xi_i^{+}, \xi_i^{-} \qquad\qquad \ge 0, \end{cases} \end{aligned} \tag{3.6}$$

where C_i is the varying (adaptive) regularization constant. Therefore, the Lagrangian of Eq. (3.6) is given as:

$$\begin{aligned} &\frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \sum_{i=1}^{m} C_i\left(\xi_i^{+} + \xi_i^{-}\right) - \sum_{i=1}^{m}\left(\eta_i^{+}\xi_i^{+} + \eta_i^{-}\xi_i^{-}\right) \\ &\quad - \sum_{i=1}^{m}\alpha_i^{+}\left(\varepsilon_i + \xi_i^{+} - y_i + \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right) + b\right) \\ &\quad - \sum_{i=1}^{m}\alpha_i^{-}\left(\varepsilon_i + \xi_i^{-} + y_i - \mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right) - b\right) \\ &s.t. \quad \alpha_i^{+}, \alpha_i^{-}, \eta_i^{+}, \eta_i^{-} \ge 0. \end{aligned} \tag{3.7}$$

Therefore, the necessary conditions for α to be a solution to the original optimization problem, Eq. (3.6), are given by the following:

$$\begin{aligned} \partial_b &= \sum_{i=1}^{m}\left(\alpha_i^{+} - \alpha_i^{-}\right) = 0 \\ \partial_{\mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{m}\left(\alpha_i^{+} - \alpha_i^{-}\right)x_i = 0 \\ \partial_{\xi_i^{+}} &= C_i - \eta_i^{+} - \alpha_i^{+} = 0 \\ \partial_{\xi_i^{-}} &= C_i - \eta_i^{-} - \alpha_i^{-} = 0. \end{aligned} \tag{3.8}$$

We can rewrite Eq. (3.7) as follows:

$$\begin{aligned}
&\frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w}+\sum_{i=1}^{m}\xi_i^{+}\left(C_i-\eta_i^{+}-\alpha_i^{+}\right)+\sum_{i=1}^{m}\xi_i^{-}\left(C_i-\eta_i^{-}-\alpha_i^{-}\right)\\
&\quad+\sum_{i=1}^{m}\varepsilon_i\left(\alpha_i^{+}+\alpha_i^{-}\right)-\sum_{i=1}^{m}y_i\left(\alpha_i^{+}-\alpha_i^{-}\right)-\sum_{i=1}^{m}\mathbf{w}^{\mathbf{T}}\boldsymbol{\Phi}\left(\mathbf{x}_i\right)\left(\alpha_i^{+}-\alpha_i^{-}\right) \qquad (3.9)\\
&\quad-b\sum_{i=1}^{m}\left(\alpha_i^{+}-\alpha_i^{-}\right).
\end{aligned}$$

Substituting Eq. (3.8) into Eq. (3.9), in order to eliminate $\mathbf{w}, b, \xi_i^{+}$, and ξ_i^{-}, results in the following dual optimization problem:

$$\begin{aligned}
\text{maximize}\quad & \frac{1}{2}\sum_{i,j=1}^{m}K\left(x_i,x_j\right)\left(\alpha_i^{+}-\alpha_i^{-}\right)\left(\alpha_j^{+}-\alpha_j^{-}\right)\\
&+\sum_{i=1}^{m}\varepsilon_i\left(\alpha_i^{+}+\alpha_i^{-}\right)-\sum_{i=1}^{m}y_i\left(\alpha_i^{+}-\alpha_i^{-}\right) \qquad (3.10)\\
\text{subject to}\quad & \begin{cases}\sum_{i=1}^{m}\left(\alpha_i^{+}-\alpha_i^{-}\right)=0\\ 0\le\alpha_i^{+},\alpha_i^{-}\le C_i.\end{cases}
\end{aligned}$$

Following the approach by Ma et al. (2003), the Lagrange of Eq. (3.10) can be written as:

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{m}K\left(\mathbf{x}_i,\mathbf{x}_j\right)\left(\alpha_i^{+}-\alpha_i^{-}\right)\left(\alpha_j^{+}-\alpha_j^{-}\right)+\varepsilon\sum_{i=1}^{m}\left(\alpha_i^{+}+\alpha_i^{-}\right)\\
&\quad-\sum_{i=1}^{m}y_i\left(\alpha_i^{+}-\alpha_i^{-}\right)-\sum_{i=1}^{m}\left(\delta_i^{+}\alpha_i^{+}+\delta_i^{-}\alpha_i^{-}\right)+\zeta\sum_{i=1}^{m}\left(\alpha_i^{+}-\alpha_i^{-}\right) \qquad (3.11)\\
&\quad+\sum_{i=1}^{m}\left[u_i^{+}\left(\alpha_i^{+}-C\right)+u_i^{-}\left(\alpha_i^{-}-C\right)\right],
\end{aligned}$$

where $\delta_i^{+},\delta_i^{-},u_i^{+},u_i^{-},\zeta$ are Lagrange multipliers. Optimizing Eq. (3.11) leads to the following Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial}{\partial \alpha_i^+} = \sum_{j=1}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right)\left(\alpha_j^+ - \alpha_j^-\right) + \varepsilon - y_i - \delta_i^+ + \zeta + u_i^+ = 0, \tag{3.12}$$

$$\frac{\partial}{\partial \alpha_i^-} = -\sum_{j=1}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right)\left(\alpha_j^+ - \alpha_j^-\right) + \varepsilon + y_i - \delta_i^+ - \zeta + u_i^+ = 0, \tag{3.13}$$

$$\delta_i^{(*)} \geq 0,\ \delta_i^{(*)}\alpha_i^{(*)} = 0, \tag{3.14}$$

$$\text{and} \quad u_i^{(*)} \geq 0,\ u_i^{(*)}\left(\alpha_i^{(*)} - C\right) = 0. \tag{3.15}$$

Using the following definitions,

$$Q_{ij} = K\left(\mathbf{x}_i, \mathbf{x}_j\right), \tag{3.16}$$

$$\theta_i = \alpha_i^+ - \alpha_i^- \text{ and } \theta_j = \alpha_j^+ - \alpha_j^-, \tag{3.17}$$

$$\text{and} \quad h\left(x_i\right) \equiv f\left(x_i\right) - y_i = \sum_{j=1}^{m} Q_{ij}\theta_j - y_i + b. \tag{3.18}$$

where $h\left(x_i\right)$ is the error of the target value for vector *i*. The KKT conditions in Eqs. (3.12), (3.13), (3.14), and (3.15) can be rewritten as:

$$\begin{aligned} &\frac{\partial L_D}{\partial \alpha_i^+} = h\left(x_i\right) + \varepsilon_i = \psi_i^+ = 0 \\ &\frac{\partial L_D}{\partial \alpha_i^-} = -h\left(x_i\right) + \varepsilon_i = \psi_i^- = -\psi_i^+ + 2\varepsilon_i = 0 \\ &\frac{\partial L_D}{\partial b} = \sum_{i=1}^{m} \theta_i = 0, \end{aligned} \tag{3.19}$$

where $\psi_i^{(*)}$ is the adaptive margin function and can be described as threshold for error on both sides of the adaptive $\varepsilon-$tube. Modifying the approach by Ma et al. (2003), these KKT conditions lead to five new conditions for AOLSVR:

$$\begin{array}{llll} 2\varepsilon_i < \psi_i^+ \rightarrow & \psi_i^- < 0, & \theta_i = -C_i & i \in E^- \\ \psi_i^+ = 2\varepsilon_i \rightarrow & \psi_i^- = 0, & -C_i < \theta_i < 0 & i \in S \\ 0 < \psi_i^+ < 2\varepsilon_i \rightarrow & 0 < \psi_i^- < 2\varepsilon_i, & \theta_i = 0 & i \in R \\ \psi_i^+ = 0 \rightarrow & \psi_i^- = 2\varepsilon_i, & 0 < \theta_i < C_i & i \in S \\ \psi_i^+ < 0 \rightarrow & \psi_i^- > 2\varepsilon_i, & \theta_i = C_i, & i \in E^+. \end{array} \tag{3.20}$$

These conditions can be used to classify the training set into three subsets defined as follows:

$$\begin{array}{l} \text{The } E \text{ set: Error support vectors: } E = \left\{ i \middle| |\theta_i| = C_i \right\} \\ \text{The } S \text{ set: Margin support vectors: } S = \left\{ i \middle| 0 < |\theta_i| < C_i \right\} \\ \text{The } R \text{ set: Remaining samples: } R = \left\{ i \middle| \theta_i = 0 \right\}. \end{array} \tag{3.21}$$

Based on these conditions, we modify the AOSVR algorithm appropriately and incorporate the algorithms for computing adaptive SVR parameters for the on-line training as described in Sections 3.2 and 3.3.

3.5.1 Initializing and Updating the Algorithm

To initialize the algorithm, we adapt a two-sample solution approach (Ma et al., 2003) in which the coefficients are given by:

$$\begin{aligned}\theta_1 &= \max\left(0, \min\left(C_{12}, \frac{y_1 - y_2 - 2\varepsilon_{12}}{2\left(K_{11} - K_{12}\right)}\right)\right) \\ \theta_2 &= -\theta_1 \\ b &= \left(y_1 + y_2\right)/2.\end{aligned} \tag{3.22}$$

The subscript 1 and 2 denotes the parameter for the first two samples; whereas subscript 1 or 2 denotes the parameter for sample 1 or 2 respectively. The *E*, *S*, and *R* sets are initialized from these two points using equations in (3.21) if the *S* set is empty. However, if the *S* set is nonempty, the variation relations will have to be used to initialize the sets.

Variation Relations: Following the approach of AOSVR, the idea for adding a new sample (c) to the training set is to change its coefficient θ_c in a finite time until it meets the KKT conditions, whereas the existing samples in the training set must continue to satisfy the KKT conditions during each step of training. Using the KKT conditions and Eq., (3.20), the variation relations between $\Delta\psi_i^{(*)}$, Δb, $\Delta\varepsilon$, and $\Delta\theta_i$ are given by:

$$\Delta\psi_i^{+} = Q_{ic}\Delta\theta_c + \sum_{i=1}^{m} Q_{ij}\Delta\theta_j + \Delta b + \Delta\varepsilon_i \tag{3.23}$$

$$\Delta\theta_c + \sum_{i=1}^{m} \Delta\theta_i = 0 \tag{3.24}$$

$$\Delta\psi_i^- = -\Delta\psi_i^+ + \Delta 2\varepsilon_i \tag{3.25}$$

If the added sample must remain in *S*, then $\Delta\varepsilon = 0$ and $\Delta\psi_i^+ \equiv 0$ since $i \in S$. Therefore, Eqs. (3.23) and (3.25) becomes:

$$\sum_{j\in S} Q_{ij}\Delta\theta_j + \Delta b = -Q_{ic}\Delta\theta_c \tag{3.26}$$

$$\sum_{j\in S} \Delta\theta_j = -\Delta\theta_c. \tag{3.27}$$

Assuming $S = \{S_1, S_2, \ldots, S_{m_s}\}$, Eq. (3.26) can be represented in matrix form as:

$$Q \cdot \begin{bmatrix} \Delta b \\ \Delta\theta_{S_1} \\ \vdots \\ \Delta\theta_{S_m} \end{bmatrix} = -\begin{bmatrix} 1 \\ Q_{S_1 c} \\ \vdots \\ Q_{S_m c} \end{bmatrix} \Delta\theta_c, \tag{3.28}$$

where *Q* is defined as:

$$Q = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & Q_{S_1S_1} & \cdots & Q_{S_1S_m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & Q_{S_mS_1} & \cdots & Q_{S_mS_m} \end{bmatrix} \tag{3.29}$$

Therefore, from (3.28),

$$\begin{bmatrix} \Delta b \\ \Delta\theta_{S_1} \\ \vdots \\ \Delta\theta_{S_m} \end{bmatrix} = -Q^{-1} \cdot \begin{bmatrix} 1 \\ Q_{S_1 c} \\ \vdots \\ Q_{S_m c} \end{bmatrix} \Delta\theta_c \tag{3.30}$$

and thus the variation in the θ_c value of a new vector *c* influences θ_i values of vector $i \in S$ through the following equations:

$$\Delta b = \delta \Delta\theta_c \tag{3.31}$$

$$\Delta\theta_j = \delta_j \Delta\theta_c \tag{3.32}$$

where

$$\begin{bmatrix} \delta \\ \delta_{S_1} \\ \vdots \\ \delta_{S_m} \end{bmatrix} = -\mathbf{R} \begin{bmatrix} 1 \\ Q_{S_1 c} \\ \vdots \\ Q_{S_m c} \end{bmatrix} \tag{3.33}$$

and $\mathbf{R} = Q^{-1}$.

For vectors $i \notin S$, the variation relation is obtained by replacing $\Delta\theta_j$ and Δb by their equivalence in Eqs. (3.31) and (3.32) and noting that $\Delta\varepsilon \neq 0$. Therefore,

$$\begin{aligned}\Delta\psi_i^+ &= Q_{ic}\Delta\theta_c + \sum_{j\in S} Q_{ij}\Delta\theta_j + \Delta b + \Delta\varepsilon_i \\ &= Q_{ic}\Delta\theta_c + \sum_{j\in S} Q_{ij}\delta_j\Delta\theta_c + \delta\Delta\theta_c + \Delta\varepsilon_i \\ &= \left(Q_{ic} + \sum_{j\in S} Q_{ij}\delta_j + \delta\right)\Delta\theta_c + \Delta\varepsilon_i \\ &= \gamma_i\Delta\theta_c + \Delta\varepsilon_i \end{aligned} \tag{3.34}$$

where

$$\gamma_i = Q_{ic} + \sum_{j\in S} Q_{ij}\delta_j + \delta \qquad \forall i \notin S. \tag{3.35}$$

The γ values are defined for only $i \notin S$ because for $i \in S$, $\Delta\psi_i^+ = 0$; therefore, $\gamma_i \equiv 0$ for $i \in S$. In summary, Eq. (3.34) shows how ψ_i^+ values change as θ_c changes for vector not in *S*; Eq. (3.32) shows how θ_i values change as θ_c changes for vector $i \in S$. Furthermore, Eq. (3.31) shows how *b* varies as θ_c changes. These equations are valid if vectors do not migrate from one set to the other while maintaining the KKT conditions. In some other situations, in order to reach the KKT conditions for the new vector, it may be necessary for some of the vectors to change membership. In these cases, θ_c is initially set to zero and then incrementally increase or decrease its value under the KKT conditions and update the **R** matrix.

The **R** matrix is defined in Eq. (3.33) as the inverse of *Q* but we only need **R** to update θ. We adapt the efficient approach of updating the **R** matrix (Ma et al., 2003). When the *k*th sample $\mathbf{x}_{sk}$ in the *S* set is removed from the *S* set, the new **R** can be obtained using the following equation:

$$\mathbf{R}_{ij} = \mathbf{R}_{ij} - \frac{\mathbf{R}_{ik}\mathbf{R}_{kj}}{\mathbf{R}_{kk}} \qquad \forall_{j,i} \neq k \in [0 \ldots m] \tag{3.36}$$

where the index 0 refers to the *b*-term. On the other hand, when a new sample is added to *S* set, the new **R** can be updated as follows:

$$\mathbf{R}^{new} = \begin{bmatrix} & & & 0 \\ & \mathbf{R} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\gamma_c} \begin{bmatrix} \delta \\ \delta_{S_1} \\ \vdots \\ \delta_{S_m} \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \delta & \delta_{S_1} & \cdots & \delta_{S_m} & 1 \end{bmatrix}, \tag{3.37}$$

Therefore, the new algorithm for adding a new data point, *c*, to the training set is shown in Algorithm 3.1. Similarly, the algorithm for removing an old data sample *c* from the training set is given in Algorithm 3.2.

The two algorithms can also be used efficiently for updating an old, possibly incorrect, sample using these two steps:

Algorithm 3.1: Algorithm for Adding a New Sample

1. Set θ_c to 0
2. **Compute** C_c and ε_c for the new sample using the appropriate weighting function
3. **Compute** ψ_c^+ and ψ_c^- for the new sample
4. **If** $\psi_c^+ > 0$ and $\psi_c^- > 0$, **Then** add *x* to *R* and **exit**
5. **If** $\psi_c^+ \le 0$, **Then**

   Increase θ_c, update θ_i for $i \in S$ and ψ_i^+, ψ_i^- for $i \notin S$, until one of the following conditions holds:

   - $\psi_c^+ = 0$: add *c* to *S*, update *R* and **exit**
   - $\theta_c = C_c$: add *c* to E^+ and **exit**
   - *transfer vector between neighbor sets and update set memberships and R matrix.*

   Else $\{\psi_c^- \le 0\}$

   Decrease θ_c, updating θ_i for $i \in S$ and ψ_i^+, ψ_i^- for $i \notin S$, until one of the following conditions holds:

   - $\psi_c^- = 0$: add *c* to *S*, update *R* and **exit**
   - $\theta_c = -C_c$: add *c* to E^- and **exit**
   - *transfer vector between neighbor sets and update set memberships and R matrix.*

Return to 2.

Algorithm 3.2: Algorithm for Removing an Old Sample

1. **Determine** ψ_c^+ and ψ_c^- for the affected sample
2. **If** $\psi_c^+ > 0$ and $\psi_c^- > 0$, **Then** remove *c* from *R* and **exit**
3. **If** $\psi_c^+ \le 0$, **Then**

   Decrease θ_c, updating θ_i for $i \in S$ and ψ_i^+, ψ_i^- for $i \notin S$, until one of the following conditions holds:

   - $\theta_c = 0$: remove *x* from *R* and **exit**
   - *transfer vector between neighbor sets and update set memberships and R matrix.*

   Else $\{\psi_c^- \le 0\}$

   Increase θ_c, updating θ_i for $i \in S$ and ψ_i^+, ψ_i^- for $i \notin S$, until one of the following conditions holds:

   - $\theta_c = 0$: remove *c* from *R* and **exit**
   - *transfer vector between neighbor sets and update set memberships and R matrix.*

Return to 1.

1. The removal of the old incorrect sample using algorithm 3.2 for removing old samples.
2. The addition of the correct sample using algorithm 3.1 for adding new samples. However, only algorithm 3.1 is implemented in our experimental studies.

3.6 Decisions Based on the Weighting Functions and AOLSVR

In this section, we apply the proposed AOLSVR to real-world data sets: feedwater flow rate data and time-series data. For the implementations, we used a typical on-line time series prediction scenario as presented by Tashman (2000) and used a prediction horizon of one time step. The procedure used is, consider given a time series $\{x(t),\ t=1,2,\ldots\}$ and prediction origin *O*, time from which the prediction is generated, we construct a set of training samples, $\mathbf{A}_{O,B}$, from the segment of time series $\{x(t),\ t=1,\ldots,O\}$ as

$$\mathbf{A}_{O,B}=\{\mathbf{X}(t),y(t),t=B,\ldots,0-1\},$$

where $\mathbf{X}(t)=\left[x(t),\ldots,x(t-B+1)\right]^T$, $y(t)=x(t+1)$, and B is the embedding dimension of the training set $\mathbf{A}_{O,B}$, which in this dissertation is taken to be five. We train the predictor $P(\mathbf{A}_{O,B};\mathbf{X})$ from the training set $\mathbf{A}_{O,B}$. Then, predict $x(O+1)$ using $\hat{x}(O+1)=P(\mathbf{A}_{O,B};\mathbf{X}(O))$. When $x(O+1)$ becomes available, we update the prediction origin; that is, $O=O+1$ and repeat the procedure. As the origin increases, the

training set keeps growing and this can become very expensive. However, on-line prediction take advantage of the fact that the training set is augmented one sample at a time and continues to update and improve the model as more data arrive.

3.6.1 Application for Inferential Sensing

In nuclear power plant, the accurate prediction of an important variable, such as feedwater flow rate, can reduce periodic monitoring. Such prediction can be used to assess sensor performance thereby reducing maintenance costs and increasing reliability of the instrument. Feedwater flow rate directly estimates the thermal power of a reactor. Nuclear power plants use venturi meters to measure feedwater flow rate. These meters are sensitive to measurement degradation due to corrosion products in the feedwater (Gribok et al., 2000). Therefore, measurement error due to feedwater fouling results in feedwater flow rate overestimation. As a result, the thermal power of the reactor is also overestimated and the reactor must be adjusted to stay within regulatory limits, which is an unnecessary action and involves unnecessary costs. To overcome this problem, several on-line inferential sensing systems have been developed to infer the "true" feedwater flow rate (Kavaklioglu and Upadhyaya, 1994; Gross et al., 1997; Gribok et al., 1999; Gribok et al., 2000; Hines et al., 2000). Inferential sensing is the use of correlated variables for prediction. Inferential measurement is different from conventional prediction where a parameter value is estimated at time t_{n+1}, based on information about other parameters at time t_n. In inferential measurements, a parameter is estimated at time

t_n based on information about other parameter also at time t_n. A detailed description of this problem is available in (Gribok et al., 1999; Gribok et al., 2000). Like other non-stationary processes, inferential sensing is an ill-posed problem and SVR has been found useful for ill-posed problems. We think that the AOLSVR approach can further enhance prediction accuracy of inferential sensing problems.

To infer feedwater flow rate, twenty-four variables were selected as predictors based on engineering judgment and on their high correlation with feedwater flow (Gribok et al., 1999). The difference between the estimated (inferred) flow rate and the measured flow rate is called drift and the mean of the drift is used to quantify the prediction performance. The data set has 700 samples and 24 variables. The objective here is to determine if we can estimate the feedwater flow rate at any point in the power cycle. The results of the experiments are shown in Table 3.1. For this implementation, we use $C_{\min} = 2.0,\ C_{\max} = 20.0,\ \varepsilon_{\min} = 0.01,\ \varepsilon_{\max} = 0.45,$ and RBF kernel with p = 1. We implemented the algorithm for both weight functions and for the four cases of weight patterns described in Section 3.2. The g values for these cases are 0.0 for constant weight, 0.005 for linear weight, 0.3 for sigmoidal weight, and 5.0 for two distinct weights.

Table 3.1: Drift Performance for the Feedwater Flow Rate Data

Weight Function	**Mean Drift (klb/hr)**			
	AOSVR	**AOLSVR**		
	***g* = 0.0**	***g* = 0.005**	***g* = 0.3**	***g* = 5.0**
MLWF	98.0619	70.8623	4.1354	4.0713
MGWF	67.7703	28.3342	4.0732	4.0713

The results in Table 3.1 show that AOLSVR performs better than AOSVR for this data using both weight functions. We observe significant difference in prediction performance between AOSVR and AOLSVR. Both sigmoidal weight and two distinct weights models achieve the smallest mean drift; using linear weight also achieves smaller mean drift than AOSVR but its value is on the high side, which indicates that the linear weight is not very useful for this application. The type of the weight function used becomes irrelevant when $g = 5.0$ because they both have the same characteristic behavior. The difference in the performance between MGWF and MLWF in other cases is due to their different properties: MLWF is symmetric and MGWF is asymmetric.

3.6.2 Time-Series Informatics

We also demonstrate the application of the weighting function and the AOLSVR algorithm and compare its performance to the existing AOSVR using two of the widely used benchmark data in time-series predictions. The two time series benchmark data used are the Mackey-Glass equation with $\tau = 17$ (Mackey and Glass, 1977) and the Santa

Fe Institute Competition time series A (Weigend and Gershenfeld, 1994). The Mackey-Glass equation (MG17) data has 1500 data points; whereas the Santa Fe Institute Competition time series A (SFIC) data has 1000 data points. The results of the experiments for MG17 and SFIC are summarized in Tables 3.2 and 3.3 respectively.

Here, we used the value of $C_{\min} = 5.0$, $C_{\max} = 60.0$, $\varepsilon_{\min} = 0.01$, and $\varepsilon_{\max} = 0.45$ and a Gaussian radial basis function (RBF) kernel, $\exp\left(-p\left|\mathbf{x}_i - \mathbf{x}_j\right|^2\right)$, with p = 1. We also implemented the algorithm for both weight functions and for the four cases of weight patterns described in Section 3.2. The g values for these cases are 0.0 for constant weight, 0.005 for linear weight, 0.3 for sigmoidal weight, and 5.0 for two distinct weights.

Table 3.2: Performance Comparison for the Mackey-Glass Equation Data

Weight Function	**MSE (MAE)**			
	AOSVR	**AOLSVR**		
	***g* = 0.0**	***g* = 0.005**	***g* = 0.3**	***g* = 5.0**
MLWF	0.0216 (0.1283)	0.0071 (0.0736)	4.89*E*-05 (0.0058)	4.63*E*-05 (0.0058)
MGWF	0.0121 (0.0959)	0.0011 (0.0294)	4.65*E*-05 (0.0058)	4.63*E*-05 (0.0058)

Table 3.3: Performance Comparison for the Santa Fe Institute Competition Data

Weight Function	**MSE (MAE)**			
	AOSVR	**AOLSVR**		
	***g* = 0.0**	***g* = 0.005**	***g* = 0.3**	***g* = 5.0**
MLWF	0.0238 (0.1200)	0.0226 (0.1166)	0.0037 (0.0195)	0.0037 (0.0186)
MGWF	0.0164 (0.0985)	0.0149 (0.0937)	0.0037 (0.0186)	0.0037 (0.0186)

As shown in Tables 2 and 3, AOLSVR performs better than AOSVR for both data sets considered, which further confirms that using varying parameters capture more of the properties of the data than using fixed parameters. These results show that better test prediction errors are achieved as a result of updating the values of the regression parameters for the incoming training data. Furthermore, there is no significant difference between using two distinct weights and sigmoidal weight. As expected, the type of weight function used with $g = 5.0$ does not matter because both C_i and ε_i have the same characteristics.

Chapter 4

Wavelet-Based Feature Extraction Procedures

Introduction to feature extraction and wavelet transform are given in Sections 4.1 and 4.2 respectively. A review of wavelet transform in process monitoring is presented in Section 4.3. The step-down thresholding (SDT) procedure is described in Section 4.4; in addition, the results of the application of the SDT procedure are also presented in Section 4.4. A two-stage wavelet-based feature extraction methodology and the application of the methodology to shaft misalignment and biscuit dough data are presented in Section 4.5.

4.1 Introduction to Feature Extraction

Data pre-processing is an integral part of an effective decision-making process. The output of the pre-processing steps affect the time spent in coming up with final decisions and the quality of such decisions. In implementing data mining procedures, data pre-processing steps take considerable amount of time and this has been estimated to be about

50 to 70 per cent of the total time spent in implementing any particular procedure (Liu, 2003). The estimated time may even be more if the data involve has several thousands of predictor variables as the case with several real-world applications including medical data, condition monitoring data, semiconductor fabrication data, and chemical manufacturing data (Gardner et al., 1997; Bakshi, 1998; Jin and Shi, 1999; Ganesan et al., 2003; Lada et al., 2002, Omitaomu et al., 2006, & 2005a). There are several data pre-processing techniques including techniques for feature selection and/or feature extraction. In this dissertation, we will use the term feature extraction process to represent both feature selection process and feature extraction process. Feature selection is the selection of significant features in the original (primary) data domain; whereas feature extraction process selects significant features in a derived (secondary) data domain (Webb, 1999). Suppose $\mathbf{y}$ a variable of interest and $x_1,\ldots,x_p$ a set of potential explanatory variables or predictors are vectors of n observations. The problem of feature extraction arises when one wants to model the relationship between $\mathbf{y}$ and a subset of $x_1,\ldots,x_p$. There is uncertainty about which subset contains most of the information in the original set. In addition, as stated in Section 1.1, the secondary data should be represented in such a way that it conserves the essential data required for effective decisions and preserve the information contained in the original signals. This situation is particularly of interest when p is large, n is small, and $\mathbf{y}$ is multivariate $\left(\mathbf{y}_1,\ldots\mathbf{y}_q\right)$. Therefore, feature extraction is a transformation process in which the reduced set of features is more manageable and conserves the information in the original signals. The transformed dataset $\mathbf{y}_{ik}$ is given by:

$$\begin{aligned} &\mathbf{y}_{ik} = \beta\left(x_{ij}\right), \\ &\text{where } i = 1,...n \\ &\qquad j = 1,..., p \\ &\qquad k = 1,..., m \\ &\qquad m < p \end{aligned} \tag{4.1}$$

Such that β is the transformation factor, which can be a scalar or a vector, x_{ij} is the original dataset, *n* is the number of object, *p* is the number of original features, and *m* is the number of extracted features.

The feature extraction can be performed by selecting a subset of the available variables to improve the prediction performance. The number of possible combinations to select the best subset of size *m* from a given set of measurements on *p* variables is given by the following equation (Webb, 1999):

$$N_m = \frac{p!}{m!(p-m)!}, \tag{4.2}$$

which can be very large even for small values of *p* and *m*. For example, selecting the best 2 features out of 3000 original features (that is, $p = 3000$ and $m = 2$) means that 4,498,500 features sets must be considered. Therefore, this option is not feasible in problems where p is even larger than 3000. There are several other techniques available for reducing the number of variables such as correlation analysis (Hastie et al., 2001), all possible regressions, stepwise regression, wavelet-based procedures (Jeong et al., 2006), and principal components analysis. Other feature extraction procedures are available in the literature (Guyon and Elisseeff, 2003; Bedrick and Tsai, 1994; Webb, 1999). The

objectives of the processes are to reduce the number of features in order to eliminate redundancy in the prediction model, reduce prediction model instability, and enhance generalization performance of the prediction model (Webb, 1999), which will lead to a better understanding of the conditions that generated the data (Guyon and Elisseeff, 2003). These feature extraction (data-reduction) procedures have been classified into sampling approaches, modeling and transformation techniques, and data splitting methods (Lu, 2001). The application of these procedures to complicated functional or spatial data with nonstationary and correlated variables are difficult to handle (Jeong et al., 2006). The major problem is that the selection of significant features in the original domain is usually expensive or infeasible especially for high-dimensional data. In most cases, it may be better to select significant features in derived domains such as fast Fourier transform and discrete wavelet transform. One advantage of wavelet transforms over fast Fourier transform is its "universality;" that is, functions from a wide range of problems have a parsimonious representation in wavelet series (Abramovich et al., 1998). This sparseness property means that only few wavelet coefficients are actually "significant."

4.2 Discrete Wavelet Transforms

Wavelets are basis functions that allow transformation of signals from their original domain to another domain in which some operations can be performed in an easier way. The wavelet transform (WT) resembles the fast Fourier transform (FFT); however, FFT

uses sine and cosine functions; whereas WT uses basic wavelet types (basic building blocks). There are different types of wavelets that can be used as basic building blocks. Some of the popular types are shown in Fig. 4.1.

Furthermore, FFT are local in frequency domain but global in time domain; wavelets are well localized in both time and frequency domain. The local character of wavelet basis function means that they differ from zero only in a limited time domain; which makes WT applicable for condition monitoring problems. The major issue for condition monitoring is that the FFT-based methods are not suitable for non-stationary signals; therefore, not able to reveal the inherent information in such signals (Peng and Chu, 2004).

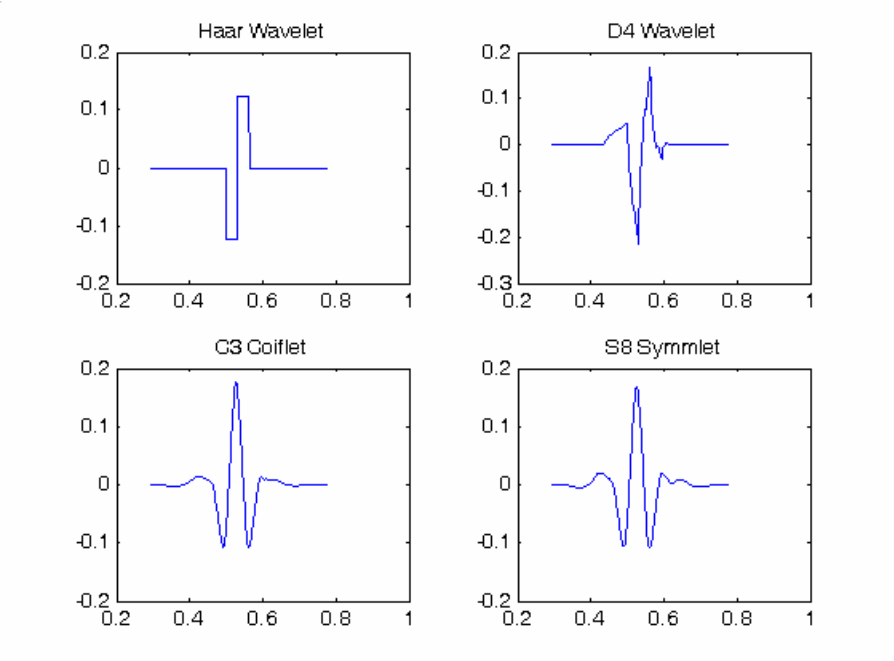

Fig. 4.1. Some basic types of wavelets.

Non-stationary characteristic of the signals is important in condition monitoring because it reveals several information such as changes in the operating environment and faults in the machines. There are several other transformation techniques suitable for analyzing non-stationary signals such as Wigner-Ville distribution (WVD) (Russell et al., 1998), and the short time Fourier transform (STFT) (Koo and Kim, 2000). These methods map one-dimensional signal $x(t)$ to a two-dimensional function of time and frequency $TFR(x:t,\omega)$. Therefore, they are suitable for non-stationary signals. However for WVD, the support areas of the signal do not overlap each other, which will mislead the signal analysis (Peng and Chu, 2004). The major problem with STFT is that there exist no orthogonal bases, which makes it difficult to find a fast and effective algorithm to calculate STFT (Peng and Chu, 2004). The wavelet transform can be used for multi-scale analysis of a signal through dilation and translation, so it can extract time-frequency features of a signal effectively. Therefore, it is suitable for the analysis of non-stationary signals (Francois and Patrick, 1995).

Mathematically, a wavelet is a function $\psi(t) \in L^2(\mathbb{R})$ with the following basic properties:

$$\int_{\mathbb{R}} \psi(t)\,dt = 0 \text{ and } \int_{\mathbb{R}} \psi^2(t)\,dt = 1, \tag{4.3}$$

where $L^2(\mathbb{R})$ space is the space of all square-integrable functions defined on the real line $\mathbb{R}$. Wavelets can be used to create a family of time-frequency atoms,

$\psi_{s,u}(t) = s^{1/2}\psi(st-u)$, using both the dilation (s) and the translation (u) factors. A scaling function is defined as $\phi(t) \in L^2(\mathbb{R})$ and satisfies the following equations:

$$\int_{\mathbb{R}} \phi(t)\,dt \neq 0 \text{ and } \int_{\mathbb{R}} \phi^2(t)\,dt = 1. \tag{4.4}$$

Selecting the scaling and wavelet functions as:

$$\left\{\phi_{L,k}(t) = 2^{L/2}\phi\left(2^L t - k\right);\ \ k \in \mathbb{Z}\right\}$$

and

$$\left\{\psi_{j,k}(t) = 2^{j/2}\psi\left(2^j t - k\right);\ \ j \geq L, k \in \mathbb{Z}\right\}$$

respectively, one can form an orthonormal basis to represent a signal function:

$$f(t) = \sum_{k\in\mathbb{Z}} c_{L,k}\phi_{L,k}(t) + \sum_{j\geq L}\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(t), \tag{4.5}$$

where $\mathbb{Z}$ denote the set of all integers $\{0, \pm 1, \pm 2, \ldots\}$, and the coefficients $c_{L,k} = \int_{\mathbb{R}} f(t)\phi_{L,k}(t)\,dt$ are considered to be the coarser-level coefficients characterizing smoother data patterns, and $d_{j,k} = \int_{\mathbb{R}} f(t)\psi_{j,k}(t)\,dt$ are viewed as the finer-level coefficients describing (local) details of data patterns. The following version of Eq. (4.5) is used in practice:

$$\tilde{f}(t) = \sum_{k=0}^{2^L-1} c_{L,k}\phi_{L,k}(t) + \sum_{j=L}^{J}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t), \tag{4.6}$$

where $J > L$ and L corresponds to the lowest decomposition level. An example of scale families of wavelets is shown in Fig. 4.2. At time t, it is assumed that signal $y(t)$ can be decomposed into signal plus noise: $y(t) = f(t) + \varepsilon_t$, where $f(t)$ is the true signal space and ε_t is the random noise that are iid $N(0, \sigma^2)$. In the sequence of data $\mathbf{y} = (y(t_1), \cdots, y(t_N))^T$ taken from $\mathbf{f} = (f(t_1), \ldots, f(t_N))^T$ at equally spaced discrete time

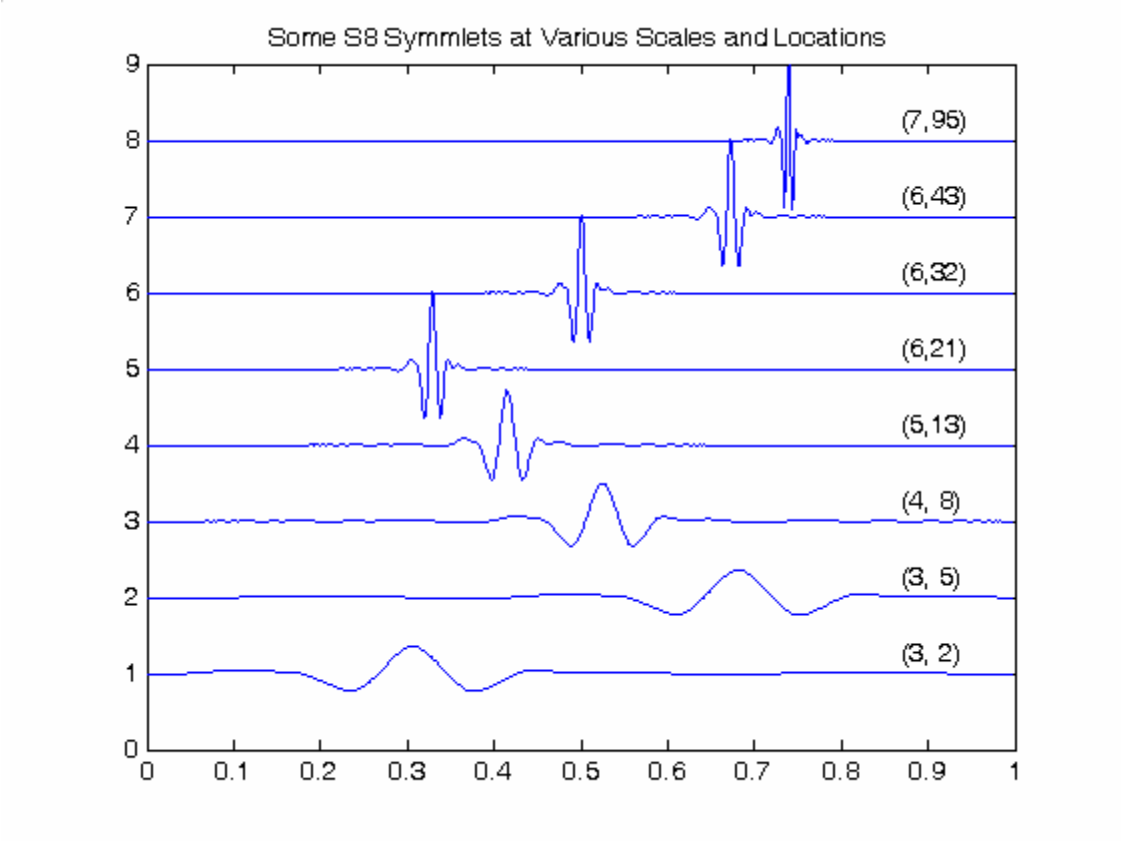

Fig. 4.2. An example of wavelet family.

points, where the superscript *T* denotes the vector transpose, $y(t_i)$ is assumed to be iid $N\left(f(t_i), \sigma^2\right)$, $i = 1, \ldots, N$. The discrete wavelet transform (DWT) of $\mathbf{y}$ is defined as:

$$\mathbf{d} = \mathbf{W}\mathbf{y} = \mathbf{W}\mathbf{f} + \mathbf{W}\boldsymbol{\varepsilon}, \tag{4.7}$$

where $\mathbf{W}$ is the orthonormal $(N \times N)$ matrix corresponding to the discrete wavelet transform and $\mathbf{d}$ is a $(N \times 1)$ vector of wavelet coefficients describing features of the original function. Equation (4.7) is achieved in only $O(N)$ operations; hence, the wavelet transforms can be computed very quickly and have good compression properties. A similar procedure for principal component analysis (PCA) requires $O(n^3)$ operations and FFT requires $O(n \log n)$ operations. The wavelet transforms have been found to be effective in several applications including noise removal (denoising), baseline removal, zero crossing to find second derivatives, signal compression, and wavelet regression. In Eq. (4.6), by letting

$$\mathbf{d} = \left(\mathbf{c}_L, \mathbf{d}_L, \mathbf{d}_{L+1}, \ldots, \mathbf{d}_J\right)^T, \tag{4.8}$$

where $\mathbf{c}_L = \left(c_{L,0}, \ldots, c_{L,2^L-1}\right)^T$, $\mathbf{d}_L = \left(d_{L,0}, \ldots, d_{L,2^L-1}\right)^T$, $\mathbf{d}_{L+1} = \left(d_{L,+1,0}, \ldots, d_{L+1,2^{L+1}-1}\right)^T$, $\ldots, \mathbf{d}_J = \left(d_{J,0}, \ldots, d_{J,2^J-1}\right)^T$ are wavelet coefficients at various subbands (scales). The total number of wavelet coefficients is equal to the number of signal measurements; that is, $\left(N = 2^{J+1}\right)$. The $c_{L,k}s$ capture low frequency oscillations that represent the coarsest

(smoothest) scale whereas the $d_{J,k}s$ capture the high frequency oscillations that represent the finest (detailed) scale (Morettin, 1997). To simplify notation, **d** in Eq. (4.8) can be written as:

$$\mathbf{d} = \left(d_1, d_2, \ldots, d_N\right)^T. \tag{4.9}$$

Using the inverse DWT, the $N \times 1$ vector **y** of the original signal can be reconstructed using:

$$\mathbf{y} = \mathbf{W}^{-1}\mathbf{d}. \tag{4.10}$$

Wavelets exist in an abundant variety as shown in Fig. 4.1; therefore, we can choose a wavelet that satisfies special signal properties. This is an additional strength for wavelet transforms. One problem, however, is deciding which wavelet will produce the best result for a particular application. Additional properties that have made wavelet transforms popular include localization, sparseness, multi-resolution, non-stationary, de-correlation, and quicker computation.

4.3 Wavelet Transforms in Process Monitoring

In the last 15 years, the application of wavelet transforms for machine fault diagnostics has been at a very rapid rate especially for small *n* large *p* problems. This section reviews the application of wavelet in condition monitoring. Leducq (1990) used wavelet to

analyze the hydraulic noise of a centrifugal pump. Wang and McFadden (1993) applied wavelet transform to analyze gear vibration signals. Wavelets have also been used for crack detection including edge cracks in cantilever beams (Zhang et al., 2001), cracks in rotors (Zou et al., 2002), and cracks in metallic structures (Bieman et al., 1999). Wavelets are also used for feature extraction in condition monitoring problems because they have good energy concentration properties (Peng and Chu, 2004). Momoh and Dias (1996) used both the FFT and the wavelet transform as feature extractors to diagnose the type and location of faults in a power distribution system and concluded that feature extracted from wavelet transforms gave better results. Ye et al. (2000) used wavelet coefficients to detect the induction motor rotor bar breakage. Momoh et al. (1995) compared the performances of feature extractors for DC power system faults using the FFT, the Hartley transform, and the wavelet transform. Their conclusion is that the wavelet extractors exhibited superior performance. Wavelets have also been used in other areas such as singularity detection (Tang and Shi, 1997), denoising and extraction of the weak signals (Donoho, 1995; Zheng et al., 2000; Altmann and Mathew, 2001; Littler and Morrow, 1996). A comprehensive review of application of wavelet transform in condition monitoring and fault diagnostics is presented by Peng and Chu (2004). Despite the extensive use of wavelet transform in condition monitoring, its applications have not achieve a standard status. This may be attributed to the fact that unlike the FFT, the results of the wavelet transform have no straightforward physical implication; therefore, it is difficult to extract useful information directly from the results of the wavelet transform (Peng and Chu, 2004). This is not a major problem in this research since we are concern with secondary data that requires further processing and our

objective is to enhance the further processing of the secondary data and wavelet transforms have been shown to have the qualities to achieve such enhancement.

4.4 Step-Down Thresholding Procedure for Single Curve

As a result of the many properties of wavelet transforms, several wavelet thresholding methods have been developed for data denoising or data reduction applications. Once the original data is transformed into wavelet coefficients, the more important of the wavelet coefficients are selected for prediction purposes. One of the popular nonlinear approximation method used to achieve this selection is the shrinkage technique. This method selects important wavelet coefficients (usually the largest in magnitude) and set to zero the unimportant coefficients (usually those representing noise). In this scheme, wavelet coefficients are set to zero if their absolute values are below or equal to a certain threshold level, λ. The thresholding process has the effect of removing data noises; hence, the shrinkage methods are also called data denoising methods. The major issue in applying this technique is the determination of the threshold value. A large threshold value will results in over-smoothing of the data curves by setting more data points to zero. A smaller threshold value, on the other hand, will allow many coefficients to be included in the reconstruction, giving a result closer to the original noisy data. A comprehensive overview of threshold selection process is given by Antoniadis et al. (1997).

From a statistical point of view, thresholding is closely related to multiple hypotheses testing, where each coefficient is tested whether it is a zero or not (Donoho and Johnstone, 1994). Only coefficients that are "significantly different from zero" are used in model building. If the results of a hypothesis testing should guide the choice of significant coefficients, a stronger control of error is needed so that no truly zero coefficients are used in the model (Abramovich and Benjamini, 1996). Some of the thresholding methods in the literature shrink insignificant coefficients without a control of any error rate; furthermore some of the methods assumed that the noise level (σ) in the data is known or can be estimated. The step-down procedure borrows its main idea from Venter and Steel's approach (Venter and Steel, 1998) for identifying active contrasts from unreplicated fractional factorial experiments. The use of experimental design approach for wavelet thresholding has many features that make it attractive from a practical point of view: the approach is simple and easily interpretable in that it involves only two quantities; assumed least number of insignificant coefficients and significant test level. Furthermore, the approach provides a flexible guidance for checking the sparseness property of wavelet coefficients by computing the *p*-values and comparing with the significant level. Corresponding with the need of aggressively shrinking data dimension for large data sets (as in the shaft misalignment example), the procedure suggested in this dissertation facilitates controlling the shrinkage ratio through the user-specified error rate.

4.4.1 Review of Thresholding Methods

The three most popular thresholding procedures are *VisuShrink* (Donoho and Johnstone, 1994), *RiskShrink* (Donoho and Johnstone, 1995), and *SURE* (Donoho and Johnstone, 1995). The VisuShrink threshold method, usually called a universal threshold method, requires an estimation of the standard deviation (σ) for calculating the threshold value as given by the following:

$$\lambda_{VisuShrink} \sim \sigma\sqrt{2\ \log N}. \tag{4.11}$$

Therefore, different estimates of σ may give different thresholds and different number of wavelet coefficients. The RiskShrink is a minimax threshold method and minimizes a theoretical upper bound on the asymptotic risk. The SURE thresholding is based on minimizing Stein's Unbiased Risk Estimate (Stein, 1981) at each resolution level. The SURE threshold for $\mathbf{d}$ with K coefficients is defined as:

$$\lambda_{SURE} = \arg\min_{t\geq 0} SURE(\mathbf{d}, t) \tag{4.12}$$

where

$$SURE(\mathbf{d}, t) = K - 2\sum_{k=1}^{K} 1_{\left[|\mathbf{d},k|\leq t\sigma\right]} + \sum_{k=1}^{K} \min\left\{(\mathbf{d}, k/\sigma)^2, t^2\right\} \tag{4.13}$$

The SURE threshold can perform poorly if the coefficients are very sparse. Of the three methods, the VisuShrink method gives smoother estimates than RiskShrink and SURE but has higher bias. Other data-driven thresholding rules in the literature include the use of adjusted cross-validation approach for choosing the threshold level (Nason, 1995, 1996). Abramovich and Benjamini (1995, 1996) and Ogden and Parzen (1996a, b) use

multiple-hypothesis testing procedure to determine threshold level. Johnstone and Silverman (1997) developed a level-dependent threshold procedure for data with correlated noise. Chipman et al. (1997), Clyde et al. (1998), Vidakovic (1998), and Abramovich et al. (1998) use a Bayesian approach to determine wavelet thresholding level.

Some of the wavelet-based data reduction techniques include Approximate Minimum Description Length (*AMDL*) method proposed by Saito (1994). The method selects N_s to minimize the following objective function:

$$AMDL\left(N_s\right) = 1.5N_s \log_2 N + 0.5N \log_2 \left[\sum_{i=1}^{N}\left(y_i - \hat{y}_{i,N_s}\right)^2\right], \qquad (4.14)$$

where $\hat{y}_{i,N_s}$ is the approximation model constructed from the N_s largest-magnitude wavelet coefficients and the data y_i is $y\left(t\right)$ evaluated at $t = t_i$. Two recently developed data-reduction methods are RRE_h and RRE_s (Jeong et al., 2006). The RRE_h is based on hard-thresholding policy and balances two ratios (the relative data-energy in the approximation model and the relative number of coefficients used). The thresholding equation is defined as:

$$RRE_h\left(\lambda\right) = \frac{E\left\|d - \hat{d}_h\left(\lambda\right)\right\|^2}{E\left\|d\right\|^2} + \omega\frac{E\left\|\hat{d}_h\left(\lambda\right)\right\|_0}{N}, \qquad (4.15)$$

where $\left\|\hat{d}_h(\lambda)\right\|_0 = \sum_{i=1}^{N}\left|\hat{d}_{h,i}(\lambda)\right|_0$ is the number of coefficients selected and $\left|\hat{d}_{h,i}(\lambda)\right|_0 = 1$, if $\hat{d}_{h,i}(\lambda) \neq 0$; $\left|\hat{d}_{h,i}(\lambda)\right|_0 = 0$, otherwise. The RRE$_s$, on the other hand, is based on the soft-thresholding policy defined as:

$$RRE_s(\lambda) = \frac{E\left\|d - \hat{d}_s(\lambda)\right\|^2}{\left(E\|d\|^2\right)^{\frac{1}{2}}} + \omega\frac{E\left\|\hat{d}_s(\lambda)\right\|_1}{\left(E\|d\|_1\right)^{\frac{1}{2}}}, \tag{4.16}$$

where $\left\|\hat{d}_s(\lambda)\right\|_1 = \sum_{i=1}^{N}\left|\hat{d}_{s,i}(\lambda)\right|$. These thresholding and data-reduction methods shrink data dimension without a control of any error rate. However, if we seek to aggressively shrink data dimension while reconstructing the original signal effectively for large data set, it is essential for users to control the resolution level of wavelet-transformed signal by specifying an error rate. In the following section, we present a wavelet thresholding procedure that controls a user specified error rate in order to guide against shrinking "significant" coefficients.

4.4.2 The Step-Down Thresholding Procedure

In this section, a wavelet thresholding procedure that designates a user-specified error rate in order to control the risk of shrinking significant coefficients is proposed. A user specifies a lower bound on the number of insignificant coefficients. The procedure starts testing the wavelet coefficients farthest from zero and proceed inwards (hence, step-down) until some criteria are met.

Rewrite the DWT equation, Eq. (4.7), as

$$\mathbf{d} = \theta + \boldsymbol{\varepsilon}', \tag{4.17}$$

where $\theta \equiv (\theta_1, \ldots, \theta_N)^T = \mathbf{W}\mathbf{f}$ and $\varepsilon' \equiv (\varepsilon_1, \ldots, \varepsilon_N)^T = \mathbf{W}\boldsymbol{\varepsilon}$. Due to orthogonality of $\mathbf{W}, \varepsilon_i'$ has an identical structure with ε_i as iid $N(0, \sigma^2)$; hence, d_i is iid $N(\theta_i, \sigma^2)$ for $i = 1, \ldots, N$. One can obtain the signal $\mathbf{f}$ from inverse wavelet transformation of $\boldsymbol{\theta}$. However, the true value $(\theta_1, \ldots, \theta_N)^T$ and σ^2 are unknown and must be estimated from the wavelet coefficients $\mathbf{d}$ only; that is, no estimation of σ^2 independent of $\mathbf{d}$ is available. Different estimates of σ will lead to distinct threshold, different shrinkage schemes of wavelet coefficients, thus different amounts of data reduction. In general, small-valued coefficients are contributed from noise data; hence, thresholding out these coefficients has an effect of "removing data noises." Relatively few of large-valued coefficients can effectively contribute to reconstruction of original signals ("sparseness property of wavelet coefficients"). In using any type of wavelet thresholding procedure, the main issue is how to choose the threshold value. The thresholding rule is intimately associated with identifying active (that is, "non-zero") contrasts in experimental design problems.

Consider a linear model for an experimental design

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{4.18}$$

where $\mathbf{Y}$ is a $(N \times 1)$ vector of responses, design matrix $\mathbf{X}$ is orthogonal, and $\boldsymbol{\varepsilon}$ is a $(N \times 1)$ vector of random errors and assumed to be iid $MVN\left(0, \sigma^2 \mathbf{I}_N\right)$. Then an ordinary least squares estimator $\hat{\beta} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{Y}$ becomes $\hat{\beta} = \mathbf{X}^T \mathbf{Y}$, the orthogonal transformation of the observations. The stepwise elimination of inactive constrasts in the model corresponds to elimination of the components with the smallest absolute value of *t*-statistics in $\hat{\beta}$. This amounts to eliminating the components in $\hat{\beta}$ with the smallest absolute value. Since the design is orthogonal, the values of the remaining $\hat{\beta}$ do not change in the process of elimination. This step-down procedure characterizing a multiple hypotheses-testing approach is considered as the hard-thresholding rule as the case of orthogonal wavelet transforms (Vidakovic, 1999 p. 177); that is,

$$\hat{\beta}_i^* = \begin{cases} \hat{\beta}_i, & \text{if } \left|\hat{\beta}_i\right| > \lambda \\ 0, & \text{if } \left|\hat{\beta}_i\right| \le \lambda, \end{cases} \tag{4.19}$$

where $\hat{\beta}^*$ is a chosen contrast as active for the threshold $\lambda(>0)$.

Introducing the multiple hypotheses-testing approach to wavelet thresholding problems, we seek a test of the overall null hypotheses that all $d_i s$ are zero (referred to as H_0), and if the H_0 is rejected, the non-zero $d_i s$ that cause the rejection are identified. The procedure is designed to control the probability of mistakenly declaring at least one of the insignificant coefficients as significant using their *p*-values. The step-down procedure proceeds downwards from the largest absolute coefficient, declaring the corresponding

wavelet-coefficients active if its test statistics value exceeds the corresponding critical value, stopping on reaching the first insignificant test statistic value or when only pre-assumed number of inactive coefficients remain. The wavelet step-down thresholding (SDT) procedure involves the following steps:

1. Sort absolute wavelet coefficients $|\mathbf{d}|$ so that $\left\{\left|d_{(N)}\right| < \left|d_{(N-1)}\right| < \cdots < \left|d_{(1)}\right|\right\}$ be the order statistics $|\mathbf{d}|$, where $\left|d_{(N)}\right|$ is the smallest absolute wavelet coefficient.

2. Define a scale invariance ratio T_q as:

$$T_q = \frac{\left|d_{(q)}\right|}{\left(\frac{1}{l-1}\sum_{i=1}^{l-1}\left|d_{(q)}\right|^2\right)^{1/2}}, \tag{4.20}$$

   where l is a specified lower bound on the number of wavelet coefficients with approximately zero mean (insignificant coefficients) and q is an assumed number of non-zero wavelet coefficients (significant coefficients) for $q = l+1,\ldots,N$. The denominator is a fixed scale-equivalent function that depends only on the pre-determined value *l*. The scale ratio is a form of standardization that will guarantee evaluating the coefficients on the same scale (Montgomery, 2005, p. 90). Section 4.4.3 discusses guidelines for determining the lower bound l and investigates their effect on the overall performance of the SDT procedure.

3. The testing of the statistic to determine the number of insignificant coefficients, is achieved using *p*-value:

$$P\left(T_j \geq c_j(\kappa) \middle| H_j\right) = \kappa, \tag{4.21}$$

where H_j denotes the parameter configuration in which exactly j of the wavelet coefficients are zero and κ is a false discovery error rate which controls the expected proportion of falsely rejected hypotheses (Benjamini and Hochberg, 1995). $c_j(\kappa) = (1-\kappa)th$ quantile of the distribution of the ratio of the largest absolute value statistic of a sample of size j from an $N(0,\sigma^2)$ to the scaling function based on $(j-1)$ smallest of the order statistics. By scale-invariance of the ratio, its distribution does not depend on σ^2 (Venter and Steel, 1998) and we will take $\sigma^2 = 1$ for convenience when calculating $c_j(\kappa)$. The exact computation of the *p*-value (Eq. (4.21)) may be difficult; therefore, an approximate method using simulation approach will be used in this dissertation. The approximate method is described in Section 4.4.3.

4. Test the statistic by proceeding downwards from the largest absolute coefficient. A coefficient is significant if its *p*-value is less or equal to the false discovery error rate (κ). The first absolute wavelet coefficient whose *p*-value is less or equal to κ, denoted by $|d_\tau|$, becomes the threshold value (λ); that is,

$$\lambda_{SDT} = |d_\tau| \tag{4.22}$$

5. Extract the significant wavelet coefficients using either hard or soft thresholding criterion. The soft thresholding criterion is defined by:

$$d_i^{s^*} = \begin{cases} sign(d_i)\left(|d_i| - \lambda_{SDT}\right), & if\ |d_i| > \lambda_{SDT} \\ 0, & if\ |d_i| \le \lambda_{SDT}. \end{cases} \tag{4.23}$$

This is a "kill or shrink" rule, where absolute coefficients less or equal to the threshold value are set to zero (killed) and absolute coefficients greater than the threshold value are shrunk towards zero. Alternatively, we defined the hard thresholding criterion by:

$$d_i^{h^*} = \begin{cases} d_i, & if\ |d_i| > \lambda_{SDT} \\ 0, & if\ |d_i| \le \lambda_{SDT}. \end{cases} \tag{4.24}$$

The hard thresholding criterion is a "kill or keep" rule in which the retained coefficients are kept. The procedure based on the soft thresholding is called *SDTs* and the procedure based on hard thresholding is *SDTh*. In reconstructing the wavelet coefficients, Donoho and Johnstone (1994) suggested that the coefficients of the first coarse levels should always be included even if these coefficients do not pass the thresholding level. We adopt this suggestion in this study. However, more flexibility options are also possible depending on applications.

4.4.3 Approximate Method of Solution

A simulation-based approach provides simple but effective alternative of calculating *p*-value. Let $\tilde{T}_q$ be a random variable whose distribution is the same as that of T_q under H_q. Then the procedure used is:

a) A realization of $\tilde{T}_q$ is generating by iid $N(0,1)$-distributed random variables $\tilde{X}_1, \tilde{X}_2 \ldots, \tilde{X}_N$.

b) Set $\tilde{T}_q = \left|\tilde{X}_{(q)}\right| \Big/ \left(\sum_{q=1}^{l-1} \left|\tilde{X}_{(q)}\right|^2 \Big/ l-1 \right)^{1/2}$, where $\left\{ \left|\tilde{X}_{(N)}\right| < \ldots < \left|\tilde{X}_{(1)}\right| \right\}$ are order statistics of $\left\{ \left|\tilde{X}_1\right|, \ldots, \left|\tilde{X}_N\right| \right\}$. Then, $\tilde{P}\left(\tilde{T}_q \geq c_q(\kappa)\right) = \kappa$ with $\tilde{P}_q$ denoting probability computed under the distribution of $\tilde{T}_q$. By defining the *p*-value associated with T_q as the solution of the equation $c_q(P_q) = T_q$ and taking $\kappa = P_q$, $P_q = \tilde{P}\left(\tilde{T}_q \geq T_q\right)$.

c) Generate realizations $\tilde{T}_q^{(1)}, \ldots, \tilde{T}_q^{(B)}$ of $\tilde{T}_q$, where *B* is the number of replications. Using $\tilde{T}_q^{(1)}, \ldots, \tilde{T}_q^{(B)}$, approximate P_q by the fraction of $\tilde{T}_q^{(b)}$s that exceed or equal to T_q as:

$$P_q = \left(\frac{1}{B}\right) \sum_{b=1}^{B} I\left(\tilde{T}_q \geq T_q\right). \tag{4.25}$$

4.4.4 Estimation of the Hyperparameters

To execute the proposed step-down thresholding procedure effectively, it is necessary to determine values of the hyperparameters: assumed number of insignificant wavelet coefficients (l), false discovery error rate (κ), and the number of simulation replications

(*B*) in Eq. (4.20), (4.21), and (4.25) respectively. Our suggestions for determining these values are as follows.

Actually, the value of *l* should be less than the anticipated number of insignificant coefficients in order to safeguard against eliminating some of the significant coefficient from the SDT procedure. If the number of extracted significant coefficients equals the number of tested coefficients in process, we decrease the *l* value and implement the procedure again. To determine the value of *l*, denote a normalized energy at position *j* as $\breve{d}_j^2 = d_j^2 \big/ \|\mathbf{d}\|^2$ and sort them to be ordered normalized energies so that $\left\{\breve{d}_{(N)}^2, \breve{d}_{(N-1)}^2, \ldots, \breve{d}_{(1)}^2\right\}$. Using a cumulative ordered normalized energy for *l* smallest $E_d(l) = \sum_{j=1}^{l} \breve{d}_{(j)}^2 \big/ \|\mathbf{d}\|^2$, the criterion for selecting the value of l is defined as

$$l = N - \left(\sum_{j=1}^{N} I\left(E_{\mathbf{d}}(l)\right) \geq (1-\delta) \right), \tag{4.26}$$

where δ is a cumulative energy level of interest and $(1-\delta)$ is the energy cut-off point.

Because wavelet coefficients have sparseness property, much greater cumulative normalized energy are compacted into fewer coefficients. For three different curves of antenna data from Jeong et al. (2006), when we give an instance of energy cut-off as 0.2, approximately 40% of the coefficients have cumulative energy greater than the cut-off value for curve 1, approximately 30% for curve 2, and about 20% for curve 6 as shown in Fig. 4.3. A pictorial explanation of notations used in developing the criteria for determining the value of *l* is shown in Fig. 4.4.

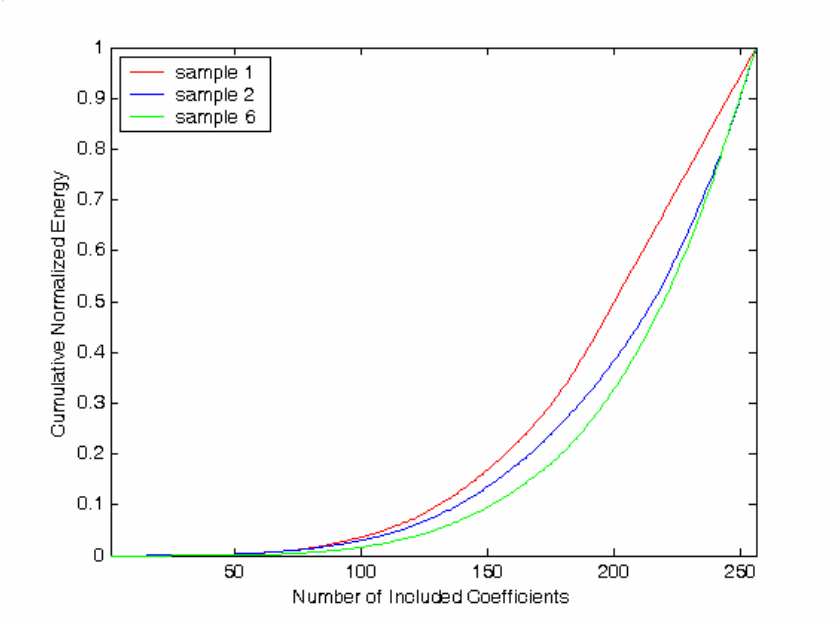

Fig. 4.3. A plot of cumulative normalized energies for three samples.

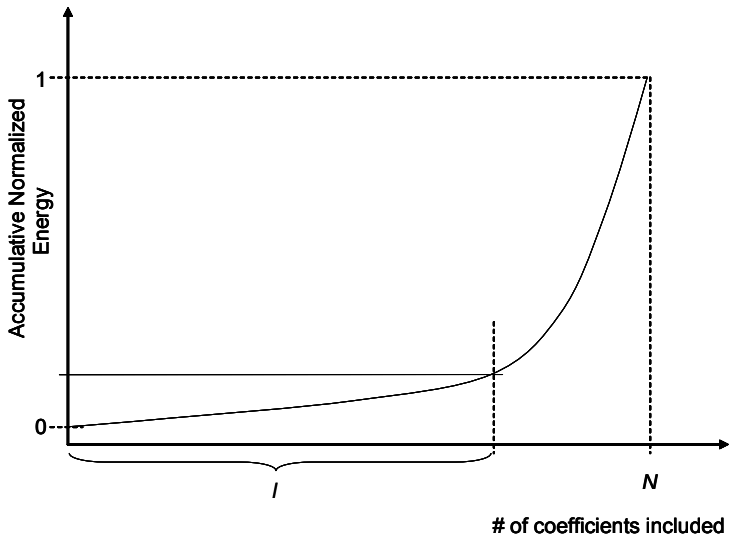

Fig. 4.4. An explanation of notations for determining the value of *l*.

The choice of κ affects the threshold level and the number of significant coefficients; therefore, different values of κ have to be implemented in order to investigate their impacts on the extracted wavelet features. However, we fix κ to either 0.01 or 0.05 to secure 95% or 99% confidence level in real world applications. The number of simulation replicates can also affect the overall thresholding procedure. It should be large enough to achieve desirable accuracy of approximation with consistent results. One way to guide against inaccurate approximation is to try different values of *B*, but if the values of *B* are large enough, they should give almost the same results.

4.4.5 Applications and Comparisons

In this section, we compare the performance of our procedure with other existing thresholding procedures using simulated examples and three real-world applications. The simulated data patterns used are four well-known testing signals from the wavelet literature and shown in Fig. 4.5. In Section 4.4.5.1, we used noise-free simulated examples and in Section 4.4.5.2, we used noisy simulated examples. The results obtained using real-world data are discussed in Section 4.4.5.3. In all the methods, the wavelet coefficients on the five coarsest levels were not thresholded and in all cases, except for SDT and RRE, the soft thresholding was applied. The "s8" wavelet was used for Bumps, HeaviSine, and Doppler signals and the "haar" wavelet was used for Blocks signal. The goodness of fit of each estimator was measured by the number of selected coefficients

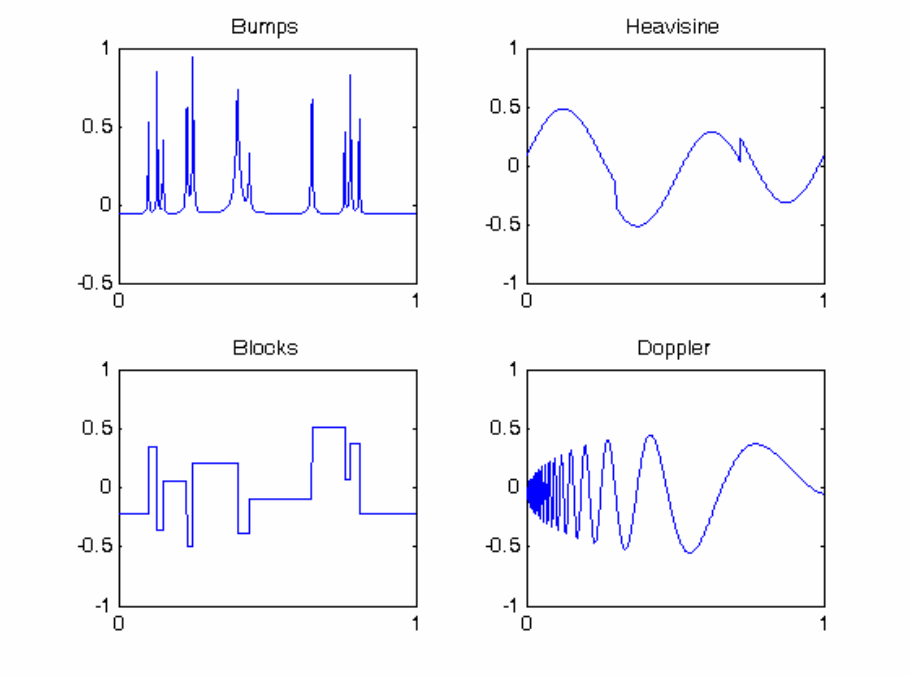

Fig. 4.5. Four noise-free testing signals from the literature.

(N_s), the compression ratio (CR), *AMDL*(N_s) measure, and average mean-square error (AMSE) defined as $N^{-1}\sum_{i=1}^{N}\left(f_i - \hat{f}_i\right)^2$ for 100 runs.

4.4.5.1 Simulation Study using Noise-Free Signals

The four "noise-free" testing signals (Fig. 4.5) characterize different types of features usually seen in imaging, seismography, manufacturing, and other engineering fields (Donoho and Johnstone, 1994; 1995). Each of the testing signals has 1024 data points (wavelet coefficients).

Table 4.1 shows a summary of the performance measures. From the results, we can see that SDT methods achieve smaller extracted features than other soft thresholding methods except RREh. This is true in all cases; therefore, the SDT methods have the highest compression ratio than other methods except for RREh in case of Bumps and Doppler signals. For all implementations in this dissertation, the value of α (Eq. (4.21)) is 0.05 and the value of δ (Eq. (4.26)) for SDT is 0.99. However, the SDT methods have the higher AMSE because it extracted smaller number of features. Both SDTh and SDTs achieve smoother reconstruction and capture the peaks as shown in Figs. 4.6 to 4.8 for the Doppler, Bumps, and HeaviSine signals respectively. These figures confirm that smaller AMSE alone is not enough in determining a better method. The probability that the reconstructed signal is as smooth as the original signal should be very high. We also notice that SDTh and SDTs achieve smooth reconstruction error with a high probability since we have 95% confidence level about the selected coefficients. In conclusion, the SDT methods achieve smooth reconstruction signals using the minimum number of features possible.

4.4.5.2 Simulation Study using Noisy Signals

The performance of the *SDT* procedure was also compared using noisy signals. In a series of experiments, various amounts of random normal noises were added to the original testing signals discussed in Section 4.4.5.1. Fig. 4.9 shows some noisy Bumps

Table 4.1: Results for the Noise-Free Signals

Method	*Measures*	***Bumps***	***HeaviSine***	***Doppler***
VisuShrink	*Ns*	646	288	600
	CR (%)	37	72	41
	AMSE	8.50	4.25	7.93
RiskShrink	*Ns*	664	314	618
	CR (%)	36	69	40
	AMSE	0.07	0.67	0.18
SURE	*Ns*	722	422	707
	CR (%)	29	59	31
	AMSE	1.25*E*+06	8.05*E*+06	4.56*E*+08
AMDL	*Ns*	1023	194	619
	CR (%)	0.09	81	40
	AMSE	2.63*E*-08	32.5	0.18
RREh	*Ns*	68	29	38
	CR (%)	93	97	96
	AMSE	1.74*E*+15	1.53*E*+21	9.04*E*+20
RREs	*Ns*	402	143	271
	CR (%)	61	86	73
	AMSE	1.05*E*+08	1.30*E*+11	1.30*E*+11
SDTh	*Ns*	111	32	265
	CR (%)	89	97	74
	AMSE	5.04*E*+14	9.43*E*+20	7.40*E*+09
SDTs	*Ns*	111	32	265
	CR (%)	89	97	74
	AMSE	2.39*E*+15	9.43*E*+20	1.48*E*+11

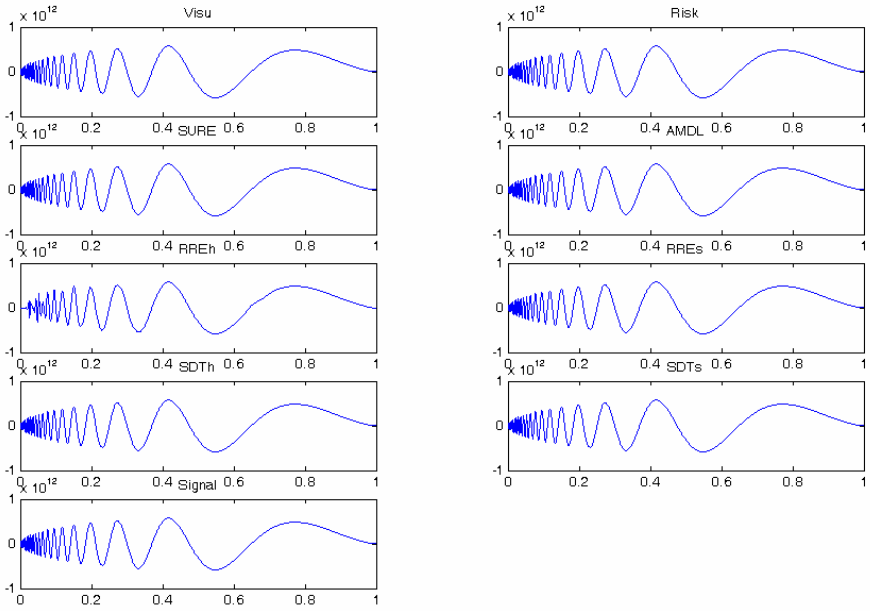

Fig. 4.6. Reconstruction of the Doppler signal.

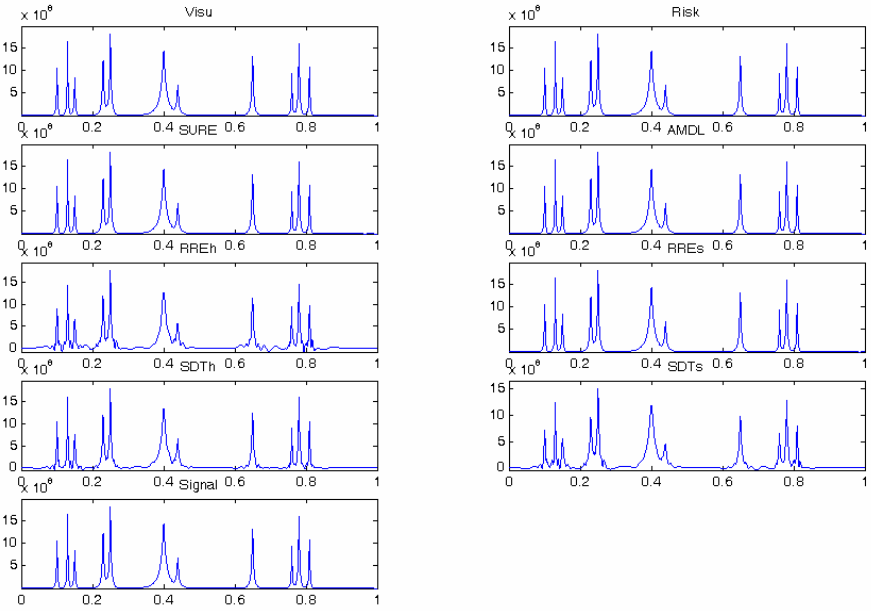

Fig. 4.7. Reconstruction of the Bumps signal.

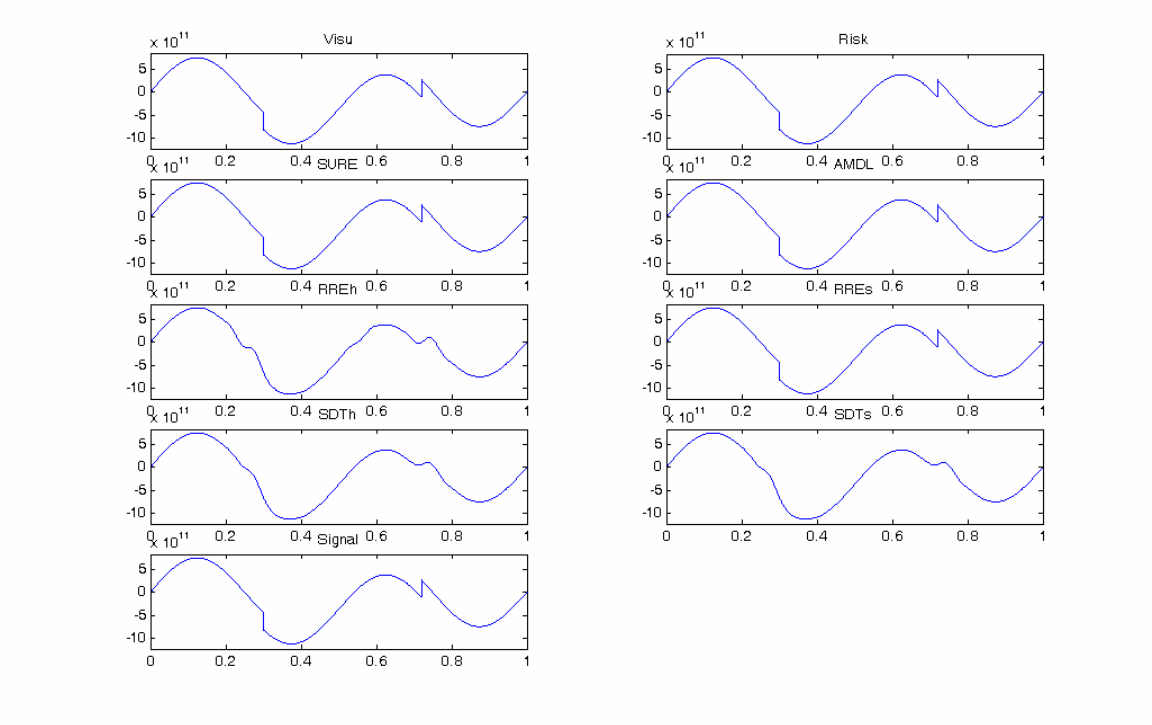

Fig. 4.8. Reconstruction of the HeaviSine signal.

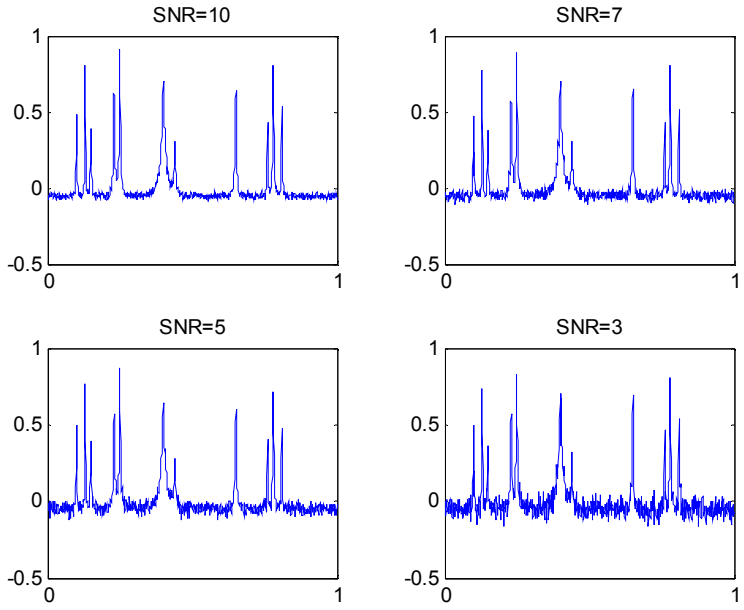

Fig. 4.9. Noisy Bumps signal at various SNR values.

signals at different signal-to-noise ratio (SNR). Table 4.2 gives a summary of the performance measures for the four SNR cases. The results in Table 4.2 indicate that, for other methods, as the SNR value gets larger (less noisy), the value of AMSE also gets larger. This indicates that denoising methods are less effective for noise-free signals. This is noted in case of VisuShrink, RiskShrink, and SURE.

However, the reduction ratio for SDT methods is somewhat consistent irrespective of the value of SNR, which indicates that SDT is applicable for data-denoising as well as for data-reduction without any concern about the amount of noise in the data. The results also show that the difference in AMSE for all the methods is smaller in case of noisy data than in case of less noisy data. The results show in Table 4.2 has different meaning with respect to the objectives of the chosen method. As stated in Section 4.1, the AMSE alone is not sufficient for final decision because these methods are based on different criteria. For example, Fig. 4.10 is the reconstruction plot for Bumps signal with SNR equals 3. The N_s for these methods are 84, 137, 221, 44, 68, 245, 104, and 104 respectively. From the figure, we can see that SDTh and SDTs achieve a smooth reconstruction without any element of the random noise. However, RiskShrink, SURE, and RREs still has some random noise present in the reconstructed signal.

Table 4.2: AMSE for the Noisy Signals

Functions	*SNR*	**Visu**	**Risk**	**SURE**	**RREh**	**SDTh**	**SDTs**
Bumps	10	2.48	0.71	0.45	3.92	2.65	3.35
	7	2.25	0.70	0.49	2.33	2.25	2.39
	5	2.07	0.70	0.46	1.61	1.38	1.95
	3	1.78	0.72	0.57	1.09	0.89	1.40
Blocks	10	1.46	0.66	0.48	2.61	2.75	2.41
	7	1.45	0.67	0.48	1.69	2.73	2.32
	5	1.41	0.67	0.48	1.25	2.76	2.30
	3	1.28	0.68	0.48	1.02	1.46	2.08
HeaviSine	10	1.01	0.72	0.70	1.26	0.72	0.88
	7	0.96	0.73	0.77	1.00	0.76	0.88
	5	0.93	0.73	0.81	0.93	0.89	0.91
	3	0.90	0.74	0.79	0.86	0.86	0.89
Doppler	10	1.33	0.72	0.65	1.68	0.68	0.94
	7	1.25	0.72	0.71	1.24	0.71	0.93
	5	1.18	0.71	0.74	1.03	0.81	0.99
	3	1.09	0.71	0.80	0.90	0.80	0.95

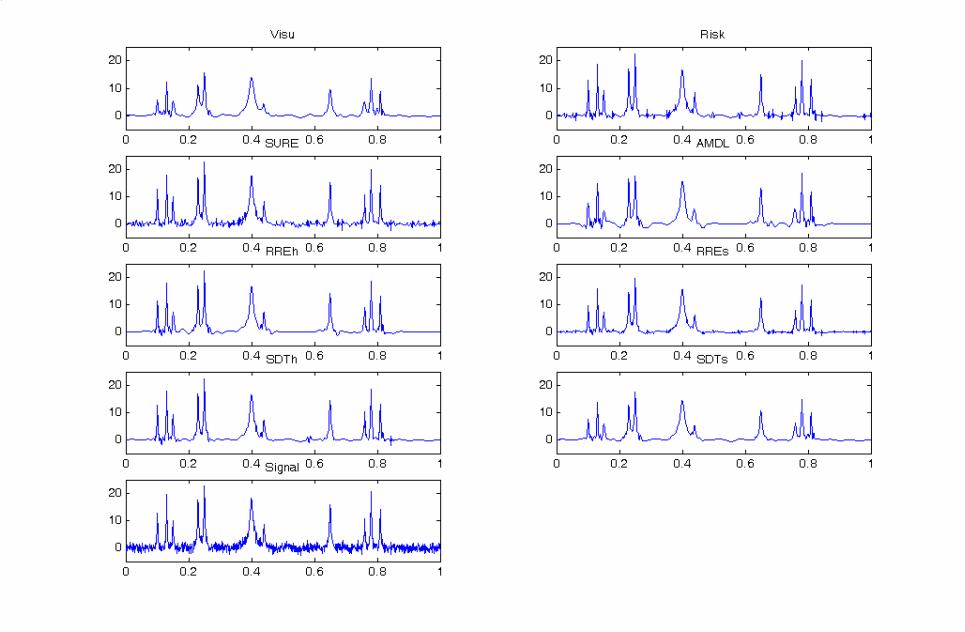

Fig. 4.10. Reconstruction of the Bumps noisy signal (SNR = 3).

4.4.5.3 Applications to Process Monitoring Problems

Three real-world datasets are also used to compare the SDT procedure with six other methods. The first data set is a data collected for developing a procedure to monitor antenna manufacturing quality (Jeong et al., 2006). The original data set has 256 wavelet coefficients. Fig. 4.11 shows a reconstruction of the data curve.

The plots show that *SDT* provides a reasonable fitting and also captures the peaks. The fitting is also smoother than the case for *RREh* and *RREs*. The results for the antenna data are summarized in Table 4.3. The results show that *SDT* uses the smallest number of features with the smoothest fitting and achieves 88% reduction ratio, which is better than other methods. *RREh* uses the smallest number of features but the fitting is not smooth

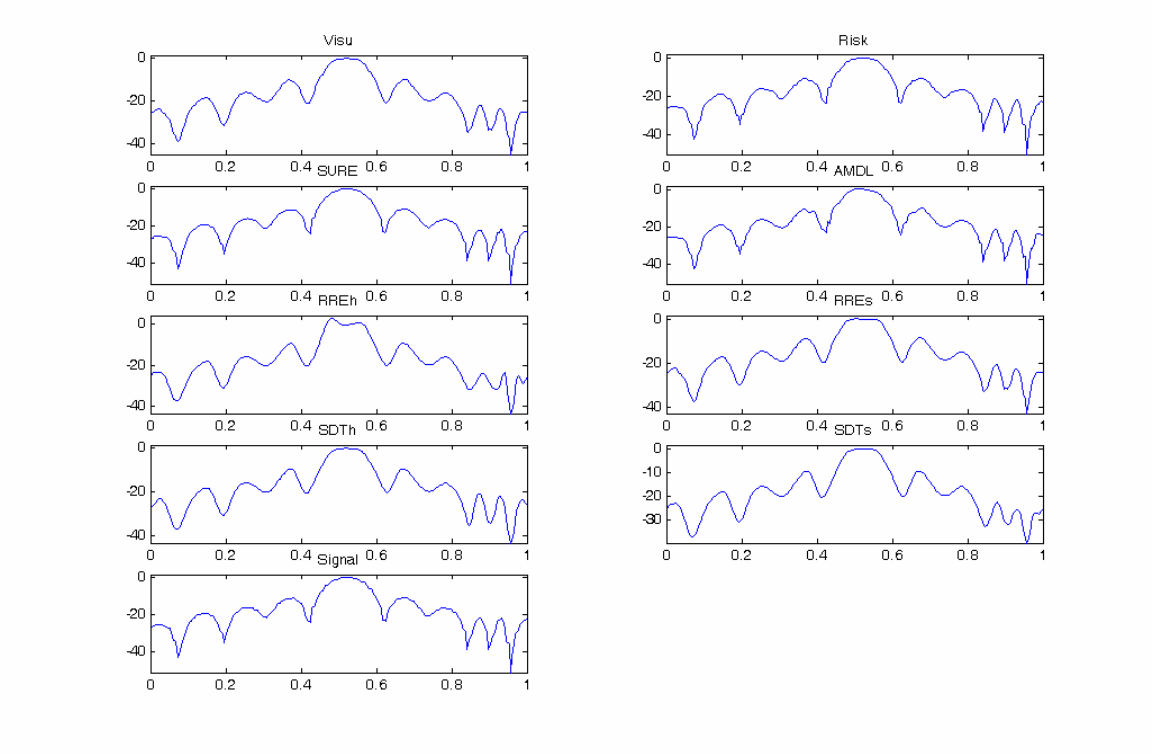

Fig. 4.11. Reconstruction of the antenna data

Table 4.3: Results for the Antenna Data

Method	N_s	*CR (%)*	*AMSE*	*AMDL*(N_s)
VisuShrink	55	79	1.55	1765
RiskShrink	73	71	0.19	1597
SURE	185	28	7196.3	4884
AMDL	55	79	0.52	1562
RRE_h	29	89	5.85	1698
RRE_s	50	81	3.56	1858
SDTh	36	86	3.02	1660
SDTs	36	86	4.31	1725

especially the center peaks. Therefore, we can say that *SDT* produces the smoothest reconstruction curve using the smallest possible coefficients needed to achieve that.

Another important application of this procedure is the biscuit dough problem (Brown et al., 2001) and has 256 spectra points. The reconstruction of the data is shown in Fig. 4.12 and the results are shown in Table 4.4. Again, we can see that SDT has the smallest AMSE and uses only 35 coefficients. One interesting observation, however, is that RREs also uses 32 wavelet points but has different RelErr, AMSE, and AMDL. One implication of such a significant reduction in spectra features is that several standard prediction techniques such as Partial Least Squares (PLS) can easily be used to predict the sample composition.

The shaft misalignment data was also used to demonstrate the application of this procedure. For this analysis, the shaft misalignment data described in Chapter 2 was used. The data set used has 1024 points by sampling every other data points. The reconstructed data is shown in Fig. 4.13 and the results are summarized in Table 4.5. The results show that the SDT procedure, like the RREh procedure, is more aggressive than the other methods for this data. However, comparing the original pattern and the reconstructed patterns by SDT and RREh shows that these methods capture the extreme outer peaks and all the inner peaks in the patterns. Comparing the number of extracted features to the original features shows that we can easily used any of the classical prediction techniques to predict misalignment conditions.

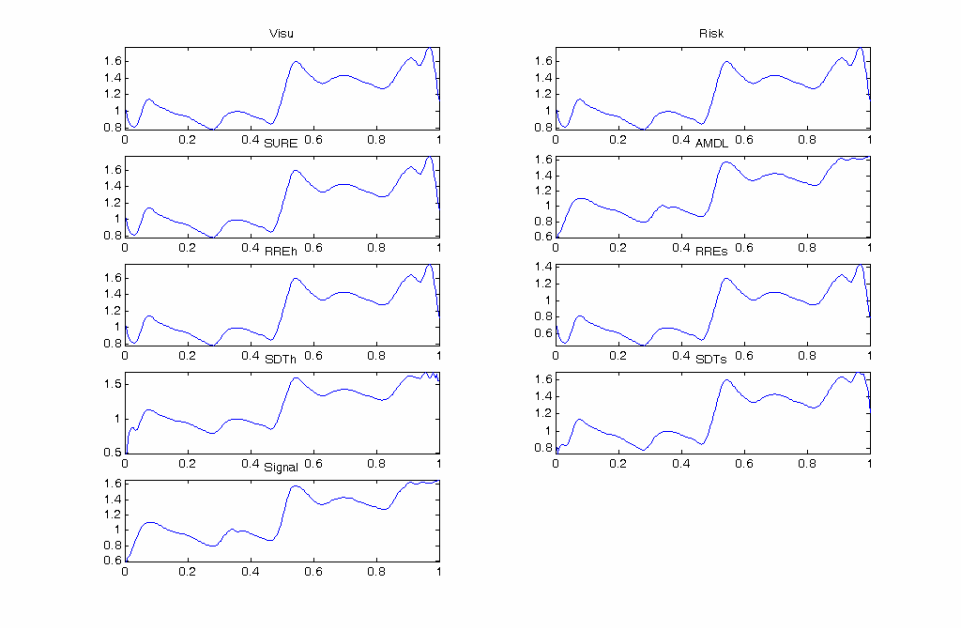

Fig. 4.12. Reconstruction of the biscuit dough data

Table 4.4: Results for the Biscuit Dough Data

Method	N_s	*CR (%)*	*AMSE*	*AMDL*(N_s)
VisuShrink	32	88	4.69*E*-03	418.08
RiskShrink	32	88	4.69*E*-03	418.08
SURE	187	27	1.90*E*+07	6362.97
AMDL	100	61	8.29*E*-08	-786.98
RRE_h	32	88	4.69*E*-03	418.08
RRE_s	32	88	1.12*E*-01	1004.76
SDTh	33	86	6.55*E*-04	90.19
SDTs	33	86	1.96*E*-03	293.16

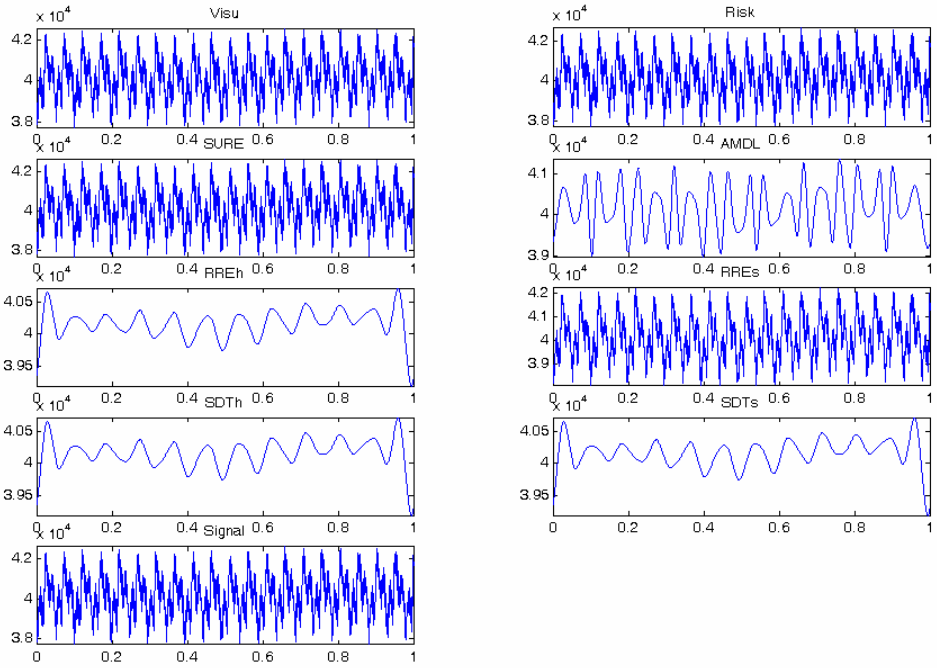

Fig. 4.13. Reconstruction of the misalignment data

Table 4.5: Results for the Misalignment Data

Method	N_s	*CR (%)*	*AMSE*	*AMDL*(N_s)
VisuShrink	1022	0.19	1.33*E+01*	22364.2
RiskShrink	1023	0.10	2.59*E*-03	16067.1
SURE	237	77	1.61*E*+09	24335.5
AMDL	36	96	1.28*E*+06	16049.5
RRE_h	32	97	1.39*E*+06	16050.1
RRE_s	838	18	4.48*E*+04	25602.2
SDTh	32	97	1.39*E*+06	16050.1
SDTs	32	97	1.39*E*+06	16050.1

4.5 Two-Stage Wavelet-Based Feature Extraction Methodology

The SDT procedure presented in Section 4.4.2 and other thresholding procedures discussed in Section 4.4.1 are only applicable to single curve. In order to apply the procedure for process monitoring problem in which there are multiple curves and multiple response variables, we present a two-stage wavelet-based feature extraction procedure for process monitoring. This two-stage procedure is also applicable to problems with single response variable.

The wavelet-based procedure for monitoring process conditions and detecting process faults involves two stages:

1. The extraction of significant wavelet coefficients for each data curve using the step-down thresholding procedure as discussed in Section 4.4 and the selection of representative significant wavelet coefficients for all the curves. The second step of this first stage is discussed in this section.
2. The application of Bayesian decision theory approach for selecting the extracted features that can predict the response variables accurately.

4.5.1 Selection of Representative Significant Wavelet Positions for all Data Curves

The number of wavelet coefficients extracted for each curve based on any of the data-reduction procedures can be different from one curve to another. In addition, if the number of extracted features is the same, the position of the extracted features may vary

from one curve to another. Therefore, the challenge is in determining the wavelet positions to represent adequately the overall structure of the reduced data. The application of wavelet data-reduction procedures for multiple data curves involves using the union or intersection strategy to select wavelets position to construct a representative of the original data curves. Lada et al. (2002) present an application of these two selection strategies.

For *N* wavelet positions and *M* curves, the union strategy selects a position if the position is identified significant for at least one curve. One advantage of this strategy is that it captures the most important features of each curve. However, the strategy usually results in larger sets of features, which is not consistent with data reduction objectives. The intersection strategy, on the other hand, selects a wavelet position if that position is identified significant for all the *M* curves; it usually results in fewer sets of features. Therefore, this strategy may ignore some active positions and make the model to be over-smooth. In this dissertation, we present a more general procedure, the voting selection strategy, in which a wavelet position is selected if that position is identified significant for *C*-out-of-*M* curves. Therefore, when the value of *C* is *one*, this strategy reduces to union strategy and if the value of *C* is equal to *M*, it becomes the intersection strategy. For the voting method, the value of *C* ranges from 2 to *M*-1.

Let $\mathbf{y}_i = \left[y_{i1}, y_{i2}, \ldots, y_{iN} \right]^{\mathrm{T}}$ be a vector of *N* equally-spaced data points from a signal curve where $N = 2^J$ with some positive integer *J* (resolution or scale level) and $i = 1, 2, \ldots, M$. Let $\mathbf{Y} = \left[\mathbf{y}_1^{\mathrm{T}}, \mathbf{y}_2^{\mathrm{T}}, \ldots, \mathbf{y}_M^{\mathrm{T}} \right]^{\mathrm{T}}$ be the collection of *M* multiple curves. When discrete

wavelets transform (DWT) **W** is applied to a data set, the matrix of wavelet coefficients obtained from this transformation is $\mathbf{D} = \mathbf{YW}$, where $\mathbf{D} = \left[\mathbf{d}_1^{\mathrm{T}}, \mathbf{d}_2^{\mathrm{T}}, \ldots, \mathbf{d}_M^{\mathrm{T}}\right]^{\mathrm{T}}$, $\mathbf{d}_i = \left[d_{i1}, d_{i2}, \ldots, d_{iN}\right]^{\mathrm{T}}$, and d_{im} is the wavelet coefficient at the *m*th wavelet-position for the *i*th data curve. When **W** is orthonormal, the original observations **Y** can be recovered using the inverse DWT; that is, $\mathbf{Y} = \mathbf{DW}^{\mathrm{T}}$. Therefore, the voting selection strategy for the SDT procedure can be stated as:

$$\Psi_{voting}\left(d_{im}\right) = \begin{cases} 1, & \text{if } I\left(\tilde{T}_{\tilde{q}} \geq T_{\tilde{q}}\right) \geq C \\ 0, & \text{otherwise} \end{cases} \tag{4.27}$$

where *C* is the number of reference curves; that is, the minimum number of experimental replicates necessary for sufficient information about the process. Different approach can be used to determine the value of *C* for the voting strategy. A better approach will be to use different values of *C* and compare their respective reduced data in order to make the final decision. In this dissertation, we assume "pooling" 80 percent of the number of curves; therefore, *C* equal 0.8*M*.

Using the same approach, the union strategy for the SDT procedure can be stated as:

$$\begin{aligned} \Psi_{union}\left(d_{im}\right) &= \max\left(I\left(\tilde{T}_{1\tilde{q}} \geq T_{1\tilde{q}}\right), I\left(\tilde{T}_{2\tilde{q}} \geq T_{2\tilde{q}}\right), \ldots, I\left(\tilde{T}_{M\tilde{q}} \geq T_{M\tilde{q}}\right)\right) \\ &= \prod_{i=1}^{M} I\left(\tilde{T}_{i\tilde{q}} \geq T_{i\tilde{q}}\right) = 1, \end{aligned} \tag{4.28}$$

and the intersection selection strategy for the SDT procedure can be given by the following equations:

$$\Psi_{\text{intersection}}\left(d_{im}\right) = \min\left(I\left(\tilde{T}_{1\tilde{q}} \geq T_{1\tilde{q}}\right), I\left(\tilde{T}_{2\tilde{q}} \geq T_{2\tilde{q}}\right), \ldots, I\left(\tilde{T}_{M\tilde{q}} \geq T_{M\tilde{q}}\right)\right)$$
$$= \prod_{i=1}^{M} I\left(\tilde{T}_{i\tilde{q}} \geq T_{i\tilde{q}}\right) = M. \tag{4.29}$$

These selection strategies are depicted pictorially in Fig. 4.14 for *M* number of curves.

4.5.2 Applications and Comparisons of the Multiple Curve Procedure

In order to compare the data reduction methods, we implement the selection strategies using four different thresholding methods (SDT, SURE, RRE, and AMDL). The soft thresholding criterion is used in each case. For this analysis, we used the shaft misalignment data described in Chapter 2. The original data has 3072 patterns and 50 curves. The data set used in the dissertation has 1024 patterns obtained by taking every other pattern. The results of the number of coefficients extracted are given in Table 4.6.

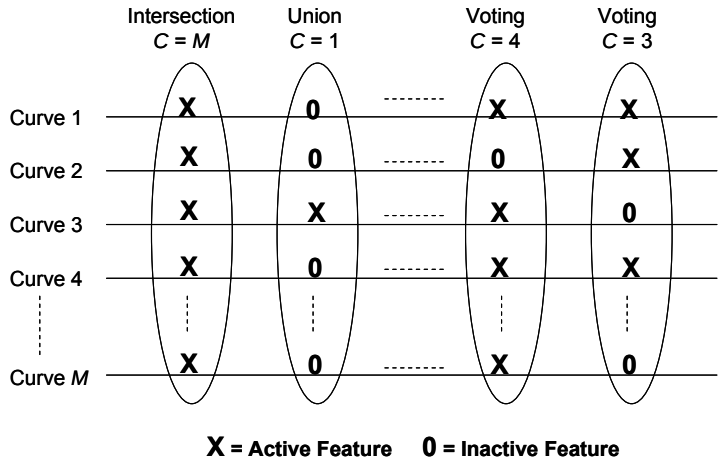

Fig. 4.14. An illustration of the selection strategies.

Table 4.6: Number of Coefficients Extracted for the Shaft Misalignment Data

Data Reduction Method	**Selection Strategies**		
	Voting	Intersection	Union
SDT	32	32	32
SURE	201	109	258
RRE	32	32	32
AMDL	40	32	49

The results in Table 4.6 indicate that the SDT method extracts the smallest number of coefficients for each of the selection strategy. In addition, SDT consistently extracts the same wavelet coefficients, which indicates that the method extracts about the same coefficients for each curve. The consistency is due to the testing approach used by the method. The SURE method returns almost the same number of wavelet coefficients for two of the strategies. This is not a surprise since the method was not initially developed for data reduction problem but for data denoising problems. The method fails since this data contains no noise. The AMDL method also performs very well compare to SDT. The method extracts the same number of coefficients as SDT using the voting selection strategy but slightly higher numbers for others. The RRE method also gives reduced coefficients but the number of extracted coefficients is still very large compare to the number of curves. These results indicate that the shaft misalignment problem is a very difficult problem and most of the procedures perform poorly extracting significant features.

We also test the performance of the proposed procedure using a set of biscuit dough spectroscopy data collected to measure the composition of formed biscuit dough pieces (Brown et al., 2001) and described in Chapter 2. The original data set has 256 wavelet coefficients and 39 samples (curves). The results of the experiment are shown in Table 4.7.

The results again show that SDT, like SURE and RRE in this case, is consistent in the number of extracted coefficients. The coefficients extracted in case of RRE and SURE are the coefficients in the coarser level; SDT extracted seven more coefficients in addition to those in the coarser level making 39. The AMDL method extracts the highest number of coefficients of the four methods for each of the selection strategy.

Table 4.7: Number of Coefficients Extracted for the Biscuit Dough Data

Data Reduction Method	**Selection Strategies**		
	Voting	Intersection	Union
SDT	33	32	33
SURE	151	110	250
RRE	56	52	68
AMDL	92	83	133

4.5.3 Multivariate Bayesian Decision Theory Approach for Process Monitoring

In this section, we present the Bayesian decision theory procedure (the second stage of the feature extraction methodology) for process monitoring. The choice of explanatory variables in linear regression has attracted considerable attention in the literature, from backward and forward stepwise regression (for example, see Hrushka, 1987), model selection criteria such as Akaike's information criterion, to Bayesian techniques (Brown et al., 1999). The idea behind these techniques is to identify a subset from either the actual variables or derived variables (such as, wavelet coefficients) that will produce the smallest possible prediction errors for future samples. In this research, we focus on Bayesian framework for feature extraction using wavelet coefficients. Several researchers have developed Bayesian based methodology for extracting features in wavelet domain. Brown et al. (1998a, b) present Bayesian variable selection methods that use mixture priors for multivariate data. Brown et al., (2001) investigated using mixing priors and Markov Chain Monte Carlos (MCMC) methods, an approach which led to model averaging. Vannucci et al. (2003) present a feature selection approach for high-dimensional data using conjugate Bayesian decision theory approach. In this research, we apply a methodology for selecting extracted wavelet coefficients using non-conjugate Bayesian decision theory approach. The previous application of this methodology for variable selection is using the original data domain (see Brown et al., 1999; Fang and Dawid, 2002). We extend this idea to feature selection in the wavelet domain. In this stage, we omit some of the extracted coefficients in the final model not because we believe their coefficients are zero but because they cost too much relative to

their predictive benefit. This approach has a number of features including a multivariate response, a proper non-conjugate prior distribution avoiding determinism, and simulated annealing search technique.

4.5.3.1 Non-Conjugate Bayesian Decision Theory

Bayesian inference is widely used for feature selection because it does not place any constraints on the number of features that can be used. When the number of features becomes large, the inference becomes very sensitive to the prior distribution (Fang and Dawid, 2002). Different prior distribution has been investigated in the literature. Dawid (1988) investigates the implications and consequences of using the conjugate prior distribution. The use of conjugate prior implies a strong belief in the possibility of deterministic prediction, which means that the response variable can be predicted with arbitrarily high accuracy by using a sufficiently large number of predictors. This idea is reasonable in certain applications such as pattern recognition. However, in most other contexts such deterministic assumption may not be reasonable because it leads to predictions which may be too simply in most realistic contexts. Therefore, the use of conjugate prior in such contexts is inappropriate. As a result, non-conjugate prior has been investigated (Brown et al., 1999; Fang and Dawid, 2002). The use of non-conjugate prior has several advantages including easy of implementation and indeterminism while avoiding overfitting (Fang and Dawid, 2002). In this research, we apply non-conjugate Bayesian decision theory to feature selection in the wavelet domain.

The model considered in this research follows notation introduced by Dawid (1981) for matrix-variate distribution. This notation makes Bayesian manipulations easier and preserves the matrix structure (Brown et al., 2001). A detailed formulation of non-conjugate Bayesian decision theory is provided by Fang and Dawid (2002) and Brown et al. (1999) and this section is based on these papers. To construct a non-conjugate prior, suppose that the response $\mathbf{Y}$ is the sum of an unobservable variable η and an error α, independent of each other; the joint distribution of η and the explanatory variables $\mathbf{X}_q$ is normal; α has an independent normal distribution. Therefore, the model is:

$$\mathbf{Y} = \eta + \alpha, \tag{4.30}$$

$$\left(\eta, \mathbf{X}_q\right) \sim N\left(1, \mathbf{\Sigma}_{r+q}\right), \qquad \alpha \sim N\left(1, \Phi\right), \tag{4.31}$$

with $\left(\eta, \mathbf{X}_q\right)$ and α conditionally independent, given $\left(r+q\right)\times\left(r+q\right)$ and $r \times r$ covariance matrices $\mathbf{\Sigma}_{r+q}$ and $\mathbf{\Phi}$, respectively. The means of all these variables are zero since both $\mathbf{X}$ and $\mathbf{Y}$ are assumed centered. If γ is a binary q-vector that identifies subsets,

$$\gamma_i = 1 \leftrightarrow x_i \text{ included}, \tag{4.32}$$

that is, where the ith feature is selected if the ith entry of γ is a 1 and not if 0. A typical selected model will have $p = \left|\gamma\right| < q$ features. A particular submodel γ involving p of the explanatory variables has the distribution in Eqs. (4.30) and (4.31) with $\mathbf{X}_q$ and $\mathbf{\Sigma}_{r+q}$ replaced by $\mathbf{X}_\gamma$ (the row vector of p variables) and $\mathbf{\Sigma}_{r+\gamma}$ (the appropriate

$(r+p)\times(r+p)$ submatrix of $\mathbf{\Sigma}_{r+q}$) respectively. The joint normality of $\mathbf{Y}$ and $\mathbf{X}_q$ implies that

$$\mathbf{Y}\left|\mathbf{X}_\gamma \sim \mathbf{X}_\gamma \mathbf{B}_\gamma + N\left(1, \Delta_\gamma\right),\right. \tag{4.33}$$

$$\mathbf{X}_\gamma \sim N\left(1, \mathbf{\Sigma}_{\gamma\gamma}\right), \tag{4.34}$$

where the joint covariance matrix $\mathbf{\Sigma}_{r+\gamma}$ is partitioned as the $(r+p)\times(r+p)$ matrix and

$$\mathbf{\Sigma}_{r+\gamma} = \begin{pmatrix} \mathbf{\Sigma}_{00} & \mathbf{\Sigma}_{0\gamma} \\ \mathbf{\Sigma}_{\gamma 0} & \mathbf{\Sigma}_{\gamma\gamma} \end{pmatrix}. \tag{4.35}$$

If $\mathbf{B}_\gamma = \mathbf{\Sigma}_{\gamma\gamma}^{-1}\mathbf{\Sigma}_{\gamma 0}\left(p\times r\right)$ and $\mathbf{\Sigma}_{00.\gamma} = \mathbf{\Sigma}_{00} - \mathbf{\Sigma}_{0\gamma}\mathbf{\Sigma}_{\gamma\gamma}^{-1}\mathbf{\Sigma}_{\gamma 0}$, then $\Delta_\gamma = \mathbf{\Sigma}_{00.\gamma} + \mathbf{\Phi}\left(r\times r\right)$. An inverse Wishart prior distribution is assigned for $\mathbf{\Sigma}_{r+q}$, which by implication also assigns for $\mathbf{\Sigma}_{r+\gamma}$, as:

$$\mathbf{\Sigma}_{r+q} \sim IW\left(\delta; \mathbf{Q}_{r+q}\right) \tag{4.36}$$

where $\delta > 0$ is the shape parameter and $\mathbf{Q}_{r+q}$ is an $(r+q)\times(r+q)$ positive definite scale matrix. The assumption used in this formulation is that the error covariance matrix $\mathbf{\Phi}$ is proportional to the residual explainable covariance matrix $\mathbf{\Sigma}_{00.q}$. Let

$$w_q \mathbf{I}_r = \mathbf{\Sigma}_{00.q}\left(\mathbf{\Sigma}_{00.q} + \mathbf{\Phi}\right)^{-1} \tag{4.37}$$

for some scalar w_q.

The training data consist of *n* independent realizations from Eqs. (4.30) and (4.31) leading to $\mathbf{Y}^l\left(n\times r\right)$ and $\mathbf{X}_q^l\left(n\times q\right)$. In this section, the superfix *l* is used to note explicitly that *n* observations of the training data are involved, whereas *f* as a superfix

denotes a future observation. The interest is on predicting $\mathbf{Y}^f$ for a future case with $\mathbf{Y}^f = \eta^f + \alpha^f$, and $\left(\eta^f, \mathbf{X}_q^f\right)$ and α^f independent realizations of model (4.30) and (4.31) conditional on the covariance matrices $\left(\mathbf{\Sigma}_{r+q}, \mathbf{\Phi}\right)$.

Using a *p*-variate subset γ of the *q* regressor variables for prediction and considering the quadratic prediction loss defined as

$$L\left(\mathbf{Y}^f, \hat{\mathbf{Y}}^f\right) = tr\left\{\ell\left(\mathbf{Y}^f - \hat{\mathbf{Y}}^f\right)\left(\mathbf{Y}^f - \hat{\mathbf{Y}}^f\right)'\right\}, \tag{4.38}$$

with ℓ any $r \times r$ positive definite matrix of weight constants. The Bayes predictor $\hat{\mathbf{Y}}^f$ is the predictor of $\mathbf{Y}^f$ assuming all variables have been measured in the learning data $\mathbf{Y}^l, \mathbf{X}_q^l$ but that only the selection γ of the $\mathbf{X}_q^f$ is available for prediction and is given as:

$$\hat{\mathbf{Y}}^f = E\left(\mathbf{Y}^f \left| \mathbf{X}_\gamma^f, \mathbf{X}_q^l, \mathbf{Y}^l\right.\right) = E\left(\eta^f \left| \mathbf{X}_\gamma^f, \mathbf{X}_q^l, \mathbf{Y}^l\right.\right), \tag{4.39}$$

since $\mathbf{Y}^f = \eta^f + \alpha^f$ and $E\left(\alpha^f \left| \mathbf{X}_\gamma^f, \mathbf{X}_q^l, \mathbf{Y}^l\right.\right) = 0$. Conditioning the right-hand side of Eq. (4.39) on the unobserved 'error-free' variables $\eta^l\left(n \times r\right)$, the right-hand side can then be simplified to

$$E\left(\eta^f \left| \mathbf{X}_\gamma^f, \eta^l, \mathbf{X}_q^l\right.\right). \tag{4.40}$$

Evaluating Eq. (4.40) simplifies Eq. (4.39) to

$$\hat{\mathbf{Y}}^f = E\left(\mathbf{Y}^f \left| \mathbf{X}_\gamma^f, \mathbf{X}_q^l, \mathbf{Y}^l\right.\right) = \mathbf{X}_\gamma^f \mathbf{P}_{\gamma\gamma}^{-1} \mathbf{P}_{0\gamma}\left(q\right), \tag{4.41}$$

where

$$\mathbf{P}_{0\gamma}\left(q\right) = \mathbf{Q}_{0\gamma} + E\left(\eta^l \left| \mathbf{X}_q^l, \mathbf{Y}^l\right.\right)' \mathbf{X}_\gamma^l. \tag{4.42}$$

Therefore, the Bayes predictor for quadratic loss is given by Eq. (4.41). Factorizing the inverse Wishart prior distribution (Eq. (4.36)) for $\mathbf{\Sigma}_{r+q}$ and given $w_q, \mathbf{\Sigma}_{00.q}, \mathbf{Y}^l, \mathbf{X}_q^l$, the posterior distribution of η^l is given as:

$$\eta^{**} + N\left(\mathrm{I}_n + \left(1-w_q\right)\mathbf{X}_q^l \left\{ \mathbf{Q}_{qq} + w_q \left(\mathbf{X}_q^l\right)' \mathbf{X}_q^l \right\}^{-1} \left(\mathbf{X}_q^l\right)', \left(1-w_q\right)\mathbf{\Sigma}_{00.q} \right), \quad (4.43)$$

with

$$\eta^{**} = w_q \mathbf{Y}^l + \left(1-w_q\right)\mathbf{X}_q^l \mathbf{B}_q^* \quad (4.44)$$

and

$$\mathbf{B}_q^* = \left\{ \mathbf{Q}_{qq} + w_q \left(\mathbf{X}_q^l\right)' \left(\mathbf{X}_q^l\right) \right\}^{-1} \left\{ \mathbf{Q}_{q0} + w_q \left(\mathbf{X}_q^l\right)' \left(\mathbf{Y}^l\right) \right\}. \quad (4.45)$$

Since the posterior mean η^{**} depends on w_q, the value of w_q will be specify *a priori* in this application. Therefore, the quantity $\eta^{**} = E\left(\eta^l \middle| \mathbf{X}_q^l, \mathbf{Y}^l\right)$ needed in Eq. (4.42) to evaluate $\hat{\mathbf{Y}}^f$ (the Bayes predictor of $\mathbf{Y}^f$ given in Eq. (4.41)) is found in Eq. (4.44). The hyperparameters that need to be specified are the matrices $\mathbf{Q}_{r+q}$ in Eq. (4.36) and the scalar w_q in Eq. (4.37), which is assumed fixed. In addition, $\mathbf{Q}_{q0} = 0$ and $\mathbf{Q}_{00}$ is not needed to evaluate Eq. (4.41) The simplest prior structure used is to take $\mathbf{Q}_{qq} = k\mathrm{I}_q$, where *k* is a scalar to be specified (Brown et al., 1999).

4.5.3.2 Simulated Annealing Search Method

Since there are q wavelet coefficients, 2^q possible subsets are possible. Simple forward and backward stepwise algorithms could be used but with so many possible subsets they tend to get easily trapped into local minima; therefore, an optimization method is employed. Several stochastic optimization methods have been in the literature for feature extraction. Kalivas et al. (1989) used simulated annealing; Leardi et al. (1992) used genetic algorithm; and Brown et al. (2001) used metropolis search. In this research, we use simulated annealing to optimize the expected utility for the selection of final wavelet coefficients. One motivation for using this approach is that it is easy to implement.

Simulated annealing (SAN) is a searching technique that moves sequentially through the space of all possible subsets with a good chance of finding at least some of the best (that is, low cost subsets). It is a simple stochastic technique that combines a simulation technique with an annealing process, which has analogy to the cooling of a liquid or solid. "An annealing is a process in which a solid in a heat bath is melted by increasing the temperature to a high value and then reducing it slowly until the system eventually freezes down to a configuration of minimum energy" (Vannucci et al., 2003).

With the use of the binary q-vector γ that identifies subsets, the search moves sequentially through the space of all possible binary vectors trying to find good ones, that is, low cost ones. The cost function used is defined as

$$C(\gamma) = tr\{R(\gamma)\} + cp, \tag{4.46}$$

where c is the common cost value, p is the number of nonzero components of γ, and $R(\gamma)$ is the terminal cost of using a particular γ subset of the q regressors. At each step the algorithm constructs γ^{new} from γ^{old} by choosing at random between three types of move:

- Move 1: Add a variable by choosing at random a 0 in γ^{old} and changing it to a 1. Move chosen with probability P_A.
- Move 2: Delete a variable by choosing at random a 1 in γ^{old} and changing it to a 0. Move chosen with a probability P_D.
- Move 3: Swap two variables by choosing independently at random a 0 and a 1 in γ^{old} and changing both of them. Move chosen with probability 1 - P_A – P_D.

At the boundaries, with all variables included or no variable present, only deletion or addition is possible respectively and then we choose this move with probability 1. At each step $d = C(\gamma^{new}) - C(\gamma^{old})$ is calculated. If d < 0, γ^{new} is accepted. Otherwise, it is accepted with probability $\exp(-d/T)$, where T is a control parameter called temperature. We chose a cooling schedule of the form $T_i = \rho T_{i-1} (0 < \rho < 1)$, reducing temperature at each iteration i. Allowing moves to 'worse' subsets may help to avoid local minima. The starting configuration described involves specifying parameter θ and a random set of chosen features with expected size $q\theta$. We stop when the temperature becomes so low that the system essentially stops moving. Every m steps we calculate an acceptance ratio AR, the proportion of m steps that have been accepted and stop if $AR \leq \tau$. If $\tilde{\gamma}$ is the

vector with minimum cost given by the search, a good practice is to 're-heat' by starting a new annealing with $\gamma^0 = \tilde{\gamma}$. This will allow a 'jump' from $\tilde{\gamma}$ in an attempt to avoid being trapped in a local minimum. The implementation of SAN requires a number of choices, starting temperature, cooling schedule, and stopping criteria. There are no prescriptive rules for setting these hyperparameters. However, a lot can be learn by guessing some sensible values and watch the outcomes of the search, which is what we did in this research.

4.5.4 Applications of the Two-Stage Procedure to Functional Data

In order to implement the second stage procedure, we need to specify the inverse Wishart prior distribution (Eq. (4.36)) for $\mathbf{\Sigma}_{r+q}$ and the value of *c* in Eq. (4.46). We use the following simple choices: $\delta = 3$ for minimally informative prior knowledge; $k = 0.009 \wedge 2$ to provide enough shrinkage to prevent numerical problems occurring when the search investigates subsets with many variables, but this would likely have little effect on the relatively small subsets that we are interested in finding since the first stage procedure has decrease the available features considerably; $w_q = 0.5$ for indeterminism parameter; the value of *c* is estimated using the following $1/4 \times r \times 0.1$; this implies that we want to reduce the number of additional variables by 25% and expect to reduce the variance to around 0.1. Therefore, the value of *c* for the biscuit dough data is 0.0125 and for the misalignment data *c* is 0.05.

For the optimization, we ran the simulated annealing sampling with γ^0 all ones. The initial temperature was $T_0 = 300$. From the many types of cooling schedule that have been proposed, (see for example Dowsland (1995)), we choose a geometric schedule in which the temperature is reduced by a factor of $\rho = 0.999$ after each accepted move. For all the searches the three types of moves were chosen with equal probabilities of $\frac{1}{3}$; that is, adding and deleting steps were chosen with probabilities $P_A = P_D = \frac{1}{3}$ and the swapping with a probability $1 - P_A - P_D = \frac{1}{3}$. The acceptance ratio, *AR*, was calculated every 500 iterations, $m = 500$, and the search stopped when *AR* = 0, that is, $\tau = 0$. With the combination of large *m* and *AR* = 0, the annealing search can confidently be said to have frozen.

We exemplify the two-stage procedure using the biscuit dough and the shaft misalignment data. As stated in Section 4.5.2, we apply the SDT procedure to each of the data curve and select representative features using the voting method. From Table 4.7, the 33 features selected using the SDT procedure are then passed on for the second stage procedure using non-conjugate Bayesian approach and simulated annealing. For the prediction, the data was divided into training and testing sets. Twenty of the 39 samples were used for training and 19 samples used for testing. The simulated annealing procedure selected 19 features (out of the 33 features) with the minimum cost that can be used for predicting each of the four response variables (fat, flour, sugar, and water content). The selected features are 1, 5 – 9, 15 – 19, 24 – 27, and 32 – 35. These selected features represent 1380nm, 1396 – 1412nm in increment of 4nm, 1436 – 1452 nm in

increment of 4nm, 1472 – 1484nm in increment of 4nm, and 1504 – 1516nm in increment of 4nm. The simulation annealing search stopped after 15,001 iterations giving a vector $\tilde{y}$ with a minimum cost of 1.5199. The selected coefficients are used for prediction using Bayes model and partial least squares. In order to ensure fair comparisons, all the extracted features are used for prediction using partial least squares. The prediction results are summarized in Tables 4.8. From the Table, we can see that the Bayes model performs better than PLS for all the response variables. One advantage of this procedure for this type of samples is that data can be collected only at this selected wavelength for predicting the composition of the response variables rather than collecting the samples over a wider range of wavelength. This approach thereby reduces the cost of data collection and improves data prediction.

The two-stage procedure was also applied to the shaft misalignment problem. As stated in Section 2.5.1, the original data of 10 samples was duplicated to have a data set of 50 samples. This is to allow for more observations for each alignment conditions. Therefore, dividing the data set into training and testing sets as we did for the biscuit

Table 4.8: Prediction Results using the Two-Stage Procedure for the Biscuit Dough Data

Response Variable	**MSE**	
	Bayes Model	**PLS**
Fat	0.0998	0.1354
Flour	0.1273	2.3449
Sugar	1.2401	1.7543
Water content	0.0986	0.1196

dough problem is not a practically correct option. Since the 50 samples are made up of 5 sets of each sample, we predict the response variables using a leave-five-out cross-validation method. Therefore, we developed 10 different models for the shaft misalignment problem. For each model, the 32 representative features selected using SDT procedure for multiple curves and voting selection strategy were used for the second stage. We notice that the simulated annealing search method selects either 25 or 26 features for each model. In cases where 25 features are selected, the selected features are 1-3, 5-15, 20-27, 30-32. In other cases where 26 features are selected, all the 25 features listed above are also selected in addition to feature number 16. The selected features for each model were used to predict the two response variables (parallel and angular misalignment conditions) using Bayes model and partial least squares. The averages of the mean squared error (MSE) for the 10 models are shown in Table 4.9. Again, the Bayes model performs better than the PLS model. However, the results show that the problem is a difficult and challenging problem. The use of Bayes model has improved the prediction error but there is a need for further improvement in the prediction errors.

Some of the choices in this implementation are made arbitrarily but the objective is to identify a much smaller set of variables than did from the first stage. We cannot claim to have found the optimum subset since not all the original features are used in the second stage. Moreover, only an exhaustive search technique will justify such a claim. We have, however, found some very good subsets of the wavelet coefficients that can predict each of the response variables better.

Table 4.9: Prediction Results using the Two-Stage Procedure for the Shaft Misalignment Data

Response Variable	**MSE**	
	Bayes Model	**PLS**
Parallel	88.09	94.91
Angular	61.46	71.16

Chapter 5

Conclusions and Future Research

This chapter gives the overall summary of the research in Section 5.1 and provides some areas for future research in Section 5.2.

5.1 Summary of Results

In this dissertation, we have proposed and implemented two weight functions (MGWF and MLWF) for updating SVR parameters for on-line predictions. Based on experimental results, MGWF is more applicable to problems in which past samples should be given much lesser weight; whereas, MLWF is more applicable to data that should be given more weight than what MGWF gives under the same relative importance conditions. In order to apply the proposed functions for on-line predictions, we presented a modified on-line SVR algorithm called on-line SVR with adaptive weighting parameters (AOLSVR) in order to incorporate the proposed function into the regression formulations.

We compared the performance of the proposed procedure with conventional AOSVR based on two spectra data and two benchmark time-series data. As demonstrated in the experiments, AOLSVR predicts better than AOSVR in all cases. The proposed two weighting functions, MLWF and MGWF, can also be used in cases where past data is more important than recent data. In such cases, the relative importance parameter (g) must be less than zero ($g < 0$) for all the equations. For batch implementations in which all data samples are given equal weight, g is set to zero ($g = 0$).

We also presented step-down thresholding (SDT) procedures for single and multiple curves. The single curve procedure uses multiple hypothesis testing approach and controls false discovery error rate. Our procedure was exemplified using common simulated signals in the literature and real-world data. The results show that our procedure performs better than some of the common techniques in the literature and gives the same performance as several others. In order to use the procedure for process data, we propose a voting selection strategy for selecting representative features for multiple curves. The procedure is to apply the SDT procedure for each of the curves and use the voting selection strategy for extracting representative features. This feature extraction was achieved without any consideration for relationships between the predictors and the response variable(s). Therefore, we also developed a two-stage wavelet-based feature extraction procedure. The first stage is the same procedure for applying the SDT procedure for multiple curves. The second stage uses Bayesian decision theory approach in order to minimize the uncertainty in extracting features for prediction problems. The non-conjugate Bayes approach uses simulated annealing optimization technique to search

the data spaces. The selected features based on the two-stage procedures were used for prediction using Bayes model and PLS model. The results show that the extracted features accounts for most of the variability in each of the response variables. Furthermore, there is a significant reduction in the number of extracted features.

5.2 Future Research

Future work is needed to explore the strengths and weaknesses in other areas of applications (for example, classification/data clustering in data mining) and to extend the proposed idea to other areas of condition monitoring (for example, spatial image data in process monitoring, bioengineering data, and medical data). We will also consider extending the feature extraction procedure for on-line predictions.

To further reduce uncertainty surrounding the implementation of the proposed procedure, we will develop Bayesian approach to compute prediction intervals rather than point predictions since these prediction intervals are more useful in process monitoring. The choice of wavelets, decomposition level, and type of error rate can affect the performance of the SDT procedure; therefore, we need further research on how to reduce these sources of uncertainty in order to further improve predictions and reduce computation time.

My research procedures have wide applications in several other areas including nano-machining process, semiconductor fabrication, automobile industry, and chemical

industry. Therefore, I will further explore these other applications areas in order to enhance the performance of the procedures.

LIST OF REFERENCES

List of References

[1] Abramovich, F. and Benjamini, Y. (1995). Thresholding of wavelet coefficients as multiple hypotheses testing procedure. *Lecture Notes in Statistics*, **103**: 5-14.

[2] Abramovich, F. and Benjamini, Y. (1996). Adaptive thresholding of wavelet coefficients. *Computational Statistics and Data Analysis*, **22**: 351-361.

[3] Abramovich, F., Sapatinas, T., and Silverman, B.W. (1998). Wavelet thresholding via a Bayesian approach. *Journal of Royal Statistical Society* **B**, **60**: 725-749.

[4] Altmann, J. and Mathew, J. (2001). Multiple band-pass autoregressive demodulation for rolling-element bearing fault diagnostics. *Mechanical Systems and Signal Processing*, **80**: 1535-1549.

[5] Antoniadis, A., Gijbels, I., and Grégoire, G. (1997). Model selection using wavelet decomposition and applications. *Biometrika* **84** (4): 751-763.

[6] Bakshi, B. R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChe Journal* **44** (7): 1596 – 1610.

[7] Bedrick, E.J. and Tsai, C-L. (1994). Model selection for multivariate regression in small samples. *Biometrics* **50**: 226 – 231.

[8] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of Royal Statistical Society* **B**, **57** (1): 289-300.

[9] Bieman, C. Staszewski, W.J., Boller, C., and Tomlinson, G.R. (1999). Crack detection in metallic structures using piezoceramic sensors. *Key Engineering Materials*, **167**: 112-121.

[10] Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

[11] Brown, P.J., Fearn, T., and Vannucci, M. (1999). The choice of variables in multivariate regression: a Bayesian non-conjugate decision theory approach. *Biometrika*, **86**: 635-648.

[12] Brown, P.J., Fearn, T., and Vannucci, M. (2001). Bayesian wavelet regression on curves with application to a spectroscopic calibration problem. *Journal of the American Statistical Association* **96** (454): 398-408.

[13] Brown, P.J., Vannucci, M., and Fearn, T. (1998a). Bayesian wavelength selection in multicomponent analysis. *Journal of Chemometrics* **12**: 173-182.

[14] Brown, P.J., Vannucci, M., and Fearn, T. (1998b). Multivariate Bayesian variable selection and prediction. *Journal of the Royal Statistical Society, Ser. B* **60**: 627-641.

[15] Cao, L. and Tay, F.E.H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks* **14** (6): 1506-1518.

[16] Cauwenberghs, E. and Poggio, T. (2001). Incremental and decremental support vector machine learning. In T.K. Leen, T.G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems* **13**: 409-523. Cambridge, MA: MIT Press.

[17] Cherkassky, V. and Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, **17** (1): 113-126.

[18] Cherkassky, V. and Mulier, F. (1998). *Learning from Data: Concepts, Theory, and Methods.* New York: John Wiley.

[19] Chipman, H.A., Kolaczyk, E.D., and McCulloch, R.E. (1997). Adaptive Bayesian wavelet shrinkage. *Journal of the American Statistical Association*, **92**: 1413-1421.

[20] Christer, A.H. and Wang, W. (1995). A simple condition monitoring model for a direct monitoring process. *European Journal of Operational Research*, **82**(2): 258-269.

[21] Clyde, M., Parmigiani, G., and Vidakovic, B. (1998). Multiple shrinkage and subset selection in wavelets. *Biometrika*, **85 (2)**: 391-401.

[22] Courant, R. and Hilbert D. (1953). *Methods of Mathematical Physics*. vol. 1, Berlin: Springer Verlag.

[23] Cowe, I.A. and McNicol, J.W. (1985). The use of principal components in the analysis of near-infrared spectra. *Applied Spectroscopy*, **39**: 257-266.

[24] Csato, L. and Opper, M. (2001). Sparse representation for Gaussian process models. In T.K. Leen, T.G. Dietterich, V. Tresp (Eds.), *Advances in Neural Information Processing Systems*, **13**: 444-450. Cambridge, MA: MIT Press.

[25] Dawid, A.P. (1981). Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, **68**: 265-274.

[26] Dawid, A.P. (1988). The infinite regress and its conjugate analysis (with discussion). In J.M. Bernardo, M.H. DeGroot, D.V. Lindley, A.F.M. Smith (Eds.), *Bayesian Statistics*, **3**: 95-110. Oxford: Oxford University Press.

[27] Donoho, D.L. (1995). De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, **41**: 613-627.

[28] Donoho, D.L. and Johnstone, I.M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81(4)**: 425-455.

[29] Donoho, D.L. and Johnstone, I.M. (1995). Adapting to unknown smoothness in wavelet shrinkage. *Biometrika,* **81** (4): 425-455.

[30] Dowsland, K.A. (1995). Simulated annealing. In C.R. Reeves (Ed.), *Modern Heuristic Techniques for Combinatorial Problems*: 377-419. London: McGraw-Hill.

[31] Eisenmann, R. C., Sr. and Eisenmann, R. C., Jr. (1998). *Machinery Malfunction Diagnosis and Correction*. New Jersey: Prentice Hall PTR.

[32] Fang, B.Q. and Dawid, A.P. (2002). Non-conjugate Bayesian regression on many variables. *Journal of Statistical Planning and Inference*, **103**: 245-261.

[33] Fernandez, R. (1999). Predicting time series with a local support vector regression machine. In *Advanced Course on Artificial Intelligence* (ACAI '99). Available on-line at: http://www.iit.demokritos.gr/skel/eetn/acai99/ (Downloaded on January 15, 2005).

[34] Francois, A. and Patrick, F. (1995). Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE Transactions on Signal processing*, **43**: 1068-1089.

[35] Ganesan, R., Das, T.K., Sikder, A.K., and Kumar, A. (2003). Wavelet based identification of delamination emission signal. *IEEE Trans. on Semiconductor Manufacturing* **16** (4): 677–685.

[36] Gardner, M. M., Lu, J. C., Gyurcsik, R. S., Wortman, J. J., Hornung, B. B., Heinisch, H. H., Rying, E. A., Rao,S., Davis, J. C., and Mozumder, P. K. (1997). Equipment fault detection using spatial signatures. *IEEE Transaction on Components, Packaging, and Manufacturing Technology* - Part C, **20**: 295–303.

[37] Geldi, P. and Kowalski, B. (1986). Partial least squares regression: A Tutorial. *Analytica Chemica Acta*, **185**: 1-7.

[38] Gentile, C. (2001). A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, **2**: 213-242.

[39] Giordana, A., Saitta, L., Bergadano, F., Brancadori, F., and De Marchi, D. (1993). ENIGMA: A system that learns diagnostic knowledge. *IEEE Transactions on Knowledge and Data Engineering*, **50** (1), 15-28.

[40] Graepel, T., Herbrich, R., and Williamson, R.C. (2001). From margin to sparsity. In T.K. Leen, T.G. Dietterich, V. Tresp (Eds.), *Advances in Neural Information Processing Systems*, **13**: 210-216. Cambridge, MA: MIT Press.

[41] Gribok, A.V., Attieh, I., Hines, J.W., and Uhrig, R.E. (1999). Regularization of feedwater flow rate evaluation for venture meter fouling problem in nuclear power plants. *Ninth International Meeting on Nuclear Reactor Thermal Hydraulics* (NURETH-9), San Francisco, CA, Oct 3-8.

[42] Gribok, A. V., Hines, J.W., and Uhrig, R.E. (2000). Use of kernel based techniques for sensor validation in nuclear power plants. *International Topical*

Meeting on Nuclear Plant Instrumentations, Controls, and Human-Machine Interface Technologies, Washington DC.

[43] Gross, K.C., Singer, R.M., Wegerich, S.W., Herzog, J.P., Alstine, R.V., and Bockhorst, F.K. (1997). Application of a model-based fault detection system to nuclear plant signals. *Proceedings of the International Conference on Intelligent System Application to Power Systems*, pp. 60-65, Seoul, Korea.

[44] Gunn, S. R. (1998). Support vector machines for classification and regression. *Technical Report*, Image Speech and Intelligent Systems Research Group, University of Southampton, UK. Available at http://www.isis.ecs.soton.ac.uk/isystems/kernel/ (Downloaded on December 5, 2004).

[45] Guyon, I. and Elisseeff, A. (2003). An Introduction to Variable and Feature extraction. *Journal of Machine Learning Research* **3**: 1157-1182.

[46] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin: Springer.

[47] Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, (2nd ed.). Prentice Hall, NJ: Upper Saddle River.

[48] Herbster, M. (2001). Learning additive models online with fast evaluating kernels. In D.P. Helmbold, B. Williamson (Eds.), *Proceedings of the 14th Annual Conference on Computational Learning Theory*: 444-460, New York: Springer-Verlag.

[49] Hines, J.W., Gribok, A.V., Attieh, I., and Uhrig, R.E. (2000). Regularization methods for Inferential Sensing in Nuclear Power Plants. In D. Ruan (Ed.), *Fuzzy*

Systems and Soft Computing in Nuclear Engineering, pp. 285-310, New York: Springer-Verlag.

[50] Hines, J. Wesley, Jesse, S., Kuropatwinski, J., Carley, T., Kueck, J., Nower, D., and Hale, F. (1997). Motor shaft alignment versus efficiency analysis. *P/PM Technology*, 10-13, October.

[51] Hines, J. Wesley, Jesse, S., Edmondson, A., and Nower, D. (1998). Effects of motor misalignment on rotating machinery. *Proceedings of the Maintenance and Reliability Conference (MARCON 98)*, May 12-14.

[52] Hines, J. W., Jesse, S. Edmondson, A., and Nower, D. (1999). Motor shaft misalignment versus bearing load analysis. *Maintenance Technology*, 11-17, May.

[53] Hoskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics*, **2**: 211-228.

[54] Hrushka, W.R. (1987). Data analysis: wavelength selection methods. In P. Williams and K. Norris (Eds.), *Near-Infrared Technology in the Agricultural and Food Industries*: 35-55. St Paul, MN: American Association of Cereal Chemists.

[55] Jeong, M. K., Lu, J. C., Huo, X., Vidakovic, B., and Chen, D. (2006). Wavelet-based data reduction techniques for process fault detection. *Technometrics* **48**(1): 26-40.

[56] Jin, J. and Shi, J. (1999). Feature-preserving data compression of stamping tonnage information using wavelets. *Technometrics* **41** (4): 327–339.

[57] Johnstone, I.M. and Silverman, B.W. (1997). Wavelet threshold estimators for data with correlated noise. *Journal of Royal Statistical Society* **B**, **59**: 319-351.

[58] Joliffe, I.T. (2002). *Principal Component Analysis* (2nd ed.). New York: Springer-Verlag.

[59] Kalivas, J.H., Roberts, N., and Sutter, J.M. (1989). Global optimization by simulated annealing with wavelength selection for ultraviolet-visible spectrophotometry. *Analytical Chemistry*, **61**: 2024-2030.

[60] Kavaklioglu, K. and Upadhyaya, Belle R. (1994). Monitoring feedwater flow rate and component thermal performance of pressurized water reactors by means of artificial neural networks. *Nuclear Technology*, vol. 107, pp. 112-123.

[61] Kivinen, J., Smola, A.J., and Williamson, R.C. (2002). Online learning with kernels. In T.G. Dietterich, S. Becker, R.C. Williamson (Eds.), *Advances in Neural Information Processing Systems*, **14**: 785-792. Cambridge, MA: MIT Press.

[62] Koo, I.S. and Kim, W.W. (2000). Development of reactor coolant pump vibration monitoring and a diagnostic system in the nuclear power plant. *ISA Transactions*, **39**: 309-316.

[63] Kuropatwinski, J. J., Jesse, S., Hines, J. W., Edmondson, A., and Carley, J. (1997) Prediction of motor misalignment using neural networks, *Proceedings of Maintenance and Reliability Conference (MARCON 97)*, Knoxville, TN, May 20-22.

[64] Kwok, J. T. (2001). Linear dependency between ε and the input noise in ε-support vector regression. In G. Dorffner, H. Bishof, and K. Hornik, (Eds.), *ICANN 2001, LNCS 2130*, 405-410.

[65] Lada, E. K., Lu, J. C., and Wilson, J. R. (2002). A Wavelet-based procedure for process fault detection. *IEEE Transactions on Semiconductor Manufacturing* **15** (1): 79–90.

[66] Leardi, R., Boggia, R., and Terrile, M. (1992). Genetic algorithms as a strategy for feature selection. *Journal of Chemometrics*, **6**: 267-281.

[67] Leducq, D. (1990). Hydraulic noise diagnostics using wavelet analysis. *Proceedings of the International Conference on Noise Control Engineering*: 997-1000.

[68] Li, Y. and Long, P.M. (1999). The relaxed online maximum margin algorithm. In S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems*, **12**: 498-504, Cambridge, MA: MIT Press.

[69] Lin, H.T. and Lin, C.J. (2003). A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods, *Technical Report*. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf (Downloaded on January 25, 2005).

[70] Liu, Xiaohui (2003). Systems and Applications. In Michael Berthold & David J. Hand (Eds.), *Intelligent Data Analysis: An Introduction* (2nd revised ed.), 429-443. Berlin: Springer.

[71] Littler, T.B. and Morrow, D.J. (1996). Signal enhancement and de-noising of power system disturbances using the wavelet method. *Proceedings of the Universities Power Engineering Conference* **2**: 590-593.

[72] Lu, J.-C. (2001). Methodology of mining massive data set for improving manufacturing quality/efficiency. In D. Braha (Ed.). *Data Mining for Design and*

Manufacturing: Methods and Applications. New York: Kluwer Academic Publishers.

[73] Ma, J., James, T., and Simon, P. (2003). Accurate on-line support vector regression. *Neural Computation*, **15**: 2683-2703.

[74] MacGregor, J.F., Jacckle, C., Kiparissides, C., and Koutondi, M. (1994). Process monitoring and diagnosis by multiblock PLS methods. *American Institute of Chemical Engineers Journal*, **40** (5): 826-838.

[75] Mackey, M.C. and Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, **197**: 287-289.

[76] Mallat, S. G. (1998). A Theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **11**: 674-693.

[77] Mattera D. and Haykin, S. (1999). Support vector machines for dynamic reconstruction of a chaotic system. In B. Schölkopf, J. Burges, A. Smola (Eds), *Advances in Kernel Methods: Support Vector Machine*. Cambridge, MA: MIT Press.

[78] Momoh, J.A. and Dias, L.G. (1996). Solar dynamic power system fault diagnostics, *NASA Conference Publication*, **10189**: 19.

[79] Momoh, J.A., Oliver, W.E.J., and Dolce, J.L. (1995). Comparison of feature extractors on DC power system faults for improving ANN fault diagnostics accuracy. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, **4**: 3615-3623.

[80] Montgomery, D.C. (2005). *Design and Analysis of Experiments*. 6th Edition. New Jersey: John Wiley and Sons, Inc.

[81] Morettin, P. (1997). Wavelets in Statistics. *Resenhas*, **3**(2): 211-272.

[82] Muller, K.-R., Smola, A.J., Ratsch, G., Scholkopf, B., Kohlmorgen, J., and Vapnik, V. (1997). Predicting time series with support vector machines. In W. Gerstner (Ed.), *Artificial Neural Networks* – ICANN '97. 999-1004. Berlin:Springer-Verlag.

[83] Muller, K.-R., Smola, A., Ratsch, G., Schölkopf, B., Kohlmorgen, J., and Vapnik, V. (1999). Using support vector machines for time series prediction. In B. Schölkopf, J. Burges, A. Smola, (Eds), *Advances in Kernel Methods: Support Vector Machine*. Cambridge, MA: MIT Press.

[84] Nason, G.P. (1995). Choice of the threshold parameter in wavelet function estimation. *Lecture Notes in Statistics*, **103**:261-280.

[85] Nason, G.P. (1996). Wavelet shrinkage using cross-validation. *Journal of Royal Statistical Society* **B**, **58**, 463-479.

[86] Ogden, T. and Parzen, E. (1996a). Data dependent wavelet thresholding in nonparametric regression with change-point applications. *Computational Statistics & Data Analysis*, **22**: 53-70.

[87] Ogden, T. and Parzen, E. (1996b). Change-point approach to data analytic wavelet thresholding. *Statistical Computation*, **6**: 93-99.

[88] Omitaomu, O. A., Jeong, M.K., Badiru, A.B., and Hines, J.W. (2006). On-line prediction of motor shaft misalignment using FFT generated spectra data and

support vector regression. *ASME Transactions on Journal of Manufacturing Science and Engineering* (In press).

[89] Omitaomu, O. A., Jeong, M.K., Badiru, A.B., and Hines, J.W. (2005a). On-line support vector regression for machine condition monitoring with applications to motor shaft misalignment prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Part C*. (Accepted, to appear).

[90] Omitaomu, O. A., Jeong, M.K., Badiru, A.B., and Ma, J. (2005b). On-line support vector regression with adaptive weighting parameters for Inferential Sensing. *IEEE Transactions on Systems, Man, and Cybernetics: Part B.* (Under review).

[91] Osborne, B.G., Fearn, T., and Hindle, P.H. (1993). *Practical NIR Spectroscopy*. Harlow, U.K.: Longman.

[92] Osborne, B.G., Fearn, T., Miller, A.R., and Douglas, S. (1984). Application of Near Infrared Reflectance Spectroscopy to the Compositional Analysis of Biscuit and Biscuit Doughs. *Journal of the Science of Food and Agriculture*, **35**: 99-105.

[93] Osuna, E., Freund, R., and Girosi, F. (1997). An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, (Eds.), *Neural Networks for Signal Processing VII – Proceedings of the 1997 IEEE Workshop*, pp. 276-285.

[94] Peng, Z.K. and Chu, F.L. (2004). Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. *Mechanical Systems and Signal Processing*, **18**: 199-221.

[95] Piotrowski, J. (1995). *Shaft Alignment Handbook*. Marcel Dekker, Inc., New York.

[96] Raftery, A.E., Madigan, D., and Hoeting, J.A. (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* **92**: 179-191.

[97] Ramsay, J. O. and Dalzell, C. (1991). Some tools for functional data analysis (with discussion). *Journal of the Royal Statistical Society, Series B* **60**: 351-363.

[98] Russell, P.C. Cosgrave, J., Tomtsis, D., Vourdas, A., Stergioulas, L., and Jones, G.R. (1998). Extraction of information from acoustic vibration signals using Gabor transform type devices. *Measurement Science and Technology*, **9**: 1282-1290.

[99] Saito, N. (1994). Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion. In E. Foufoula-Georgiou and P. Kumar (Eds.), *Wavelets in Geophysics*, 299-324. New York: Academic Press.

[100] Schölkopf, B., Burges, J., and Smola, A. (1999). *Advances in Kernel Methods: Support Vector Machine*. Cambridge, MA: MIT Press.

[101] Schölkopf, B. and Smola, A. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press.

[102] Smola, A.J., Murata, N., Schölkopf, B., and Muller, K. (1998). Asymptotically optimal choice of ε-loss for support vector machines. In L. Niklasson, M. Boden, and T. Ziemke, (Eds.), *Proceedings of the International Conference on Artificial*

Neural Networks (ICANN 1998), Perspectives in Neural Computing, Berlin: Springer, 105-110.

[103] Smola, A.J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, **14**: 199-222.

[104] Stein, C. (1981). Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, **9(6)**: 1135-1151.

[105] Tang, W. and Shi, Y.W. (1997). Detection and classification of welding defects in friction-welded joints using ultrasonic NDT methods. *Insight: Non-Destructive Testing and Condition Monitoring*, **39**: 88-92.

[106] Tashman, L.J. (200). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, **16**: 437-450.

[107] Vannucci, M., Brown, P.J., and Fearn, T. (2003). A decision theoretical approach to wavelet regression on curves with a high number of regressors. *Journal of Statistical Planning and Inference*, **112**: 195-212.

[108] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York, NY: Springer-Verlag.

[109] Vapnik, V. (1998). *Statistical Learning Theory*. New York, NY: John Wiley.

[110] Vapnik, V. (2000). Support-vector networks. *Machine Learning*, **20** (3): 273-297.

[111] Vapnik, V., Golowich, S.E., and Smola, A. (1996). Support vector method for prediction, regression estimation, and signal processing. *Advances in Neural Information Processing Systems.* San Mateo, CA: Morgan Kaufmann Publishers.

[112] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., and Yin, K. (2003). A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers and Chemical Engineering*, **27**: 327-346.

[113] Venter, J.H. and Steel, S.J. (1998). Identifying active contrasts by stepwise testing. *Technometrics*, **40(4)**: 304-313.

[114] Vidakovic, B. (1998). Nonlinear wavelet shrinkage with Bayes rules and Bayes factors. *Journal of the American Statistical Association*, **93**: 173-179.

[115] Vidakovic, B. (1999). *Statistical Modeling by Wavelets*. New York: Wiley.

[116] Wang, W.J. and McFadden, P.D. (1993). Application of the wavelets transform to gearbox vibration analysis. *ASME, Petroleum Division Publication*, **PD 52**: 13-20.

[117] Webb, A. (1999). *Statistical Pattern Recognition*. New York: Arnold.

[118] Weigend, A.S. and Gershenfeld, N.A. (1994). *Time-series prediction: Forecasting the future and understanding the past*. Reading, MA: Addison-Wesley.

[119] Wold, S., Martens, H., and Wold, H. (1983). The multivariate calibration problem in chemsitry solved by PLS. In A. Ruhe and B. Kagstrom (Eds.), *Matrix Pencils*, 286-293. Heidelberg: Springer.

[120] Wowk, Victor (2000). *Machine Vibration: Alignment*. New York: McGraw Hill.

[121] Ye, Z., Wu, B., Zargari, N. (2000). Online mechanical fault diagnostics of induction motor by wavelet artificial neural network using stator current. *IECON Proceedings*, **2**: 1183-1188.

[122] Zhang, L.X., Li, Z., and Su, X.Y. (2001). Crack detection in beams by wavelet analysis. *Proceedings of SPIE 4537*: 229-232.

[123] Zheng, Y., Tay, D.B.H., and Li, L. (2000). Signal extraction and power spectrum estimation using wavelet transform scale space filtering and Bayes shrinkage. *Signal Processing*, **80**: 1535-1549.

[124] Zou, J., Che, J., Pu, Y.P., and Zhong, P. (2002). On the wavelet time-frequency analysis algorithm in identification of a cracked rotor. *Journal of Strain Analysis for Engineering Design*, **37**: 239-246.

VITA

Olufemi Abayomi Omitaomu received a B.S. degree in Mechanical Engineering from Lagos State University, Nigeria in 1995 and an M.S. degree in Mechanical Engineering from University of Lagos, Nigeria in 1999. He won five prizes for academic excellence during his undergraduate program. After his B.S., he worked as a project engineer for Mobil Producing Nigeria between 1995 and 2001. He started a Ph.D. program in Industrial Engineering at the University of Tennessee in 2001. He is listed in the 2006 Edition of Who's Who in America.

During his Ph.D. program, he taught an undergraduate engineering economic analysis course with full responsibilities for four semesters and a graduate data mining course with partial responsibilities for one semester. He was also involved in various research projects in the department of industrial and information engineering. Olufemi has published or submitted for publication more than 8 journal articles, four book chapters, and several papers in various conference proceedings. His areas of research include data mining, wavelets, optimization modeling, quality and reliability engineering, signal and image processing, and economic analysis and financial modeling. He has jointly published two computer software including the ENGINEA for analyzing problems in engineering economic analysis. He was webmaster for several professional and private organizations including the Department of Industrial and Information Engineering and the Engineering Economy Division of the Institute of Industrial Engineers (IIE). He won the outstanding Ph.D. student award in 2004. Olufemi is a member of several

professional bodies including the Institute of Industrial Engineers (IIE), Institute of Electrical and Electronic Engineers (IEEE), Institute for Operations Research and the Management Sciences (INFORMS), American Society of Mechanical Engineers (ASME), and American Society for Engineering Education (ASEE). He has been a board member of the Engineering Economy Division of the Institute of Industrial Engineers (IIE) since May 2002. He can be reached at femiomit@gmail.com. He recently accepted a post-doc fellowship position at McMaster University, Hamilton, Ontario.

Olufemi is married to his college sweetheart, beautiful Remilekun (Nee Soyinka); they both have two adorable sons, Oluwadamilola (Dammy) and Oluwatimilehin (Timmy).