REMOTE MONITORING AND FAULT DIAGNOSIS OF AN INDUSTRIAL MACHINE THROUGH SENSOR FUSION

by

HAOXIANG LANG

B. Sc., Ningbo University, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Mechanical Engineering)

The University of British Columbia

February 2008

Abstract

Fault detection and diagnosis is quite important in engineering systems, and deserves further attention in view of the increasing complexity of modern machinery. Traditional single-sensor methods of fault monitoring and diagnosis may find it difficult to meet modern industrial requirements because there is usually no direct way to measure and accurately correlate a machine fault to a single sensor output. Fusion of information from multiple sensors can overcome this shortcoming. In this thesis, a neural-fuzzy approach of multi-sensor fusion is developed for a network-enabled remote fault diagnosis system. The approach is validated by applying it to an industrial machine called the Iron Butcher, which is a machine used in the fish processing industry for the removal of the head in fish prior to further processing for canning.

An important characteristic of the fault diagnosis approach developed in this thesis is to make an accurate decision of the machine condition by fusing information from different sensors. First, sound, vibration and vision signals are acquired from the machine using a microphone, an accelerometer and a digital CCD camera, respectively. Second, the sound and vibration signals are transformed into the frequency domain using fast Fourier transformation (FFT). A feature vector from the FFT frequency spectra is defined and extracted from the acquired information. Also, a feature based vision tracking approach—the Scale Invariant Feature Transform (SIFT) —is applied to the vision data to track the object of interest (fish) in a robust manner. Third, Sound, vibration and vision feature vectors are provided as inputs to a neuro-fuzzy network for fault detection and diagnosis. A four-layer neural network

including a fuzzy hidden layer is developed in the thesis to analyze and diagnose existing faults. By training the neural network with sample data for typical faults, faults of five crucial components in the fish cutting machine are detected with high reliability and robustness. Alarms to warn about impending faults may be generated as well during the machine operation. A network-based remote monitoring architecture is developed as well in the thesis, which will facilitate engineers to monitor the machine condition in a more flexible manner from a remote site. Developed multi-sensor approaches are validated using computer simulations and physical experimentation with the industrial machine, and compared with a single-sensor approach.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

SMT	Surface Mount
FDC	Fault Detection and Classification
CCD	Charge Coupled Device
FFT	Fast Fourier Transform
DSP	Digital Signal Processing
SDP	Symmetric Dot Pattern
RMS	Root Mean Square
PSD	Power Spectrum Density
ANNs	Artificial Neural Networks
CNC	Computer Numerical Control
OES	Optical Emission Spectroscopy
RGA	Residual Gas Analysis
TSNNs	Time Series Neural Networks
NF	Neuro-fuzzy
RNN	Recurrent Neural Network
MV	Machine Vision
UGVs	Unmanned Ground Vehicles
HG	Hypothesis Generation
HV	Hypothesis Verification
LOC	Local Orientation Coding
MDT	Multivariate Decision Tree
PCA	Principal Components Analysis
LAN	Local Area Network
B/S	Browser/Server
VR	Virtual Reality
VE	Virtual Environment
VL	Virtual Laboratory
VRML	Virtual Reality Modeling Language

CS	Client-Server
DAQ	Data Acquisition
NIC	Network Interface Card
GUI	Graphic User Interface
API	Application Programming Interface
SIFT	Scale Invariant Feature Transform
BP	Backpropagation
IIS	Internet Information Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
JMF	Java Media Frame
RTT	Round Trip Time

Acknowledgement

First, I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Clarence W. de Silva whose constant supervision and guidance have enabled me to complete the master studies at UBC successfully. Particular acknowledgement should be made of the research grant held by Dr. de Silva from the Natural Sciences and Engineering Research Council (NSERC) of Canada, which funded the research reported here and provided a research assistantship for me.

In the same vein, I also want to thank my colleagues, Mr. Ying Wang, Mr. Jason Zhang, Dr. Tao Fan, and Mr. Mohammed Tahir Khan, at the Industrial Automation Laboratory (IAL) for their friendship and help. For those who have been working with me, and have paved the way to my success, please allow me to say a sincere "thank you." I also want to thank my friends who have supported me and given me their coaching and mentoring in the past two years.

Finally, I would like to express my deepest thankfulness to my father Junsheng Lang and my mother Xiutian Zheng who have given me life.

Chapter 1

Introduction

Fault detection and diagnosis is important in engineering systems, and deserve further attention in view of the increasing complexity and performance requirements of modern machinery. For fairly simple systems, simple sensing or observation and one's intuition usually provides sufficient diagnosis of a faulty condition fairly quickly. However, modern machines with quite complicated structural configurations; for example, chemical plants, power stations, and surface mount machines (SMT), have higher probabilities of faults, which often cannot be correctly diagnosed with traditional ways because these machines may incorporate advanced technologies and their hardware and software architectures may be very complex.

Machine failure will have product quality implications, and may cause loss of throughput, and significant financial losses. Therefore, to improve productivity and cost-effectiveness, it may become necessary to develop intelligent systems that will provide accurate and reliable fault detection, diagnosis and prediction and can suggest preventive actions, for modern machinery.

A modern automated factory environment has to be effective, flexible and reliable. Machines can be situated in various locations in the factory, and hazardous facilities should be isolated from densely populated areas. However, engineers, managers, operators and consultants still should be able to work together in a coordinated, interactive and efficient manner without being physically present at one location. For this purpose, a reliable and robust system is required that can monitor from any remote geographic location of the industrial network the health of a machine.

The field of remote sensing and intelligent fault diagnosis is an active research area, and some of the research outcomes have been applied to industrial systems such as power generation plants, fabric processing machines, food processing machines and semiconductor manufacturing equipment. Figure 1.1 shows the generic architecture of e-Mfg with real-time data which was developed by General Motors (GM®). Specifically, a typical e-Mfg system consists of target machines, a local monitoring and control/diagnosis system, a remote monitoring and control/diagnosis system and an intelligent decision making system, which are all interconnected through a common communication network. In this architecture, different sensors (counters, accelerometers, strain gauges, ultrasonic sensors, infra-red sensors, and so on) are mounted on machinery, depending on the application. A group of powerful computers acquires the sensor signals and stores the raw data in local hard drives. By preprocessing the raw data in local computers, a cluster of recognizable data is generated and distributed via Ethernet to certain remote clients who are at different locations of the network. In remote sites, customized software, which run on the clients with permission to access the servers, acquires, stores and displays on a computer screen the data for subsequent analysis. In addition, for automated manufacturing, an intelligent decision making system is included in the architecture, which is used to achieve automatic control or fault diagnosis without human interruption intervention and interruption of the machine operation.

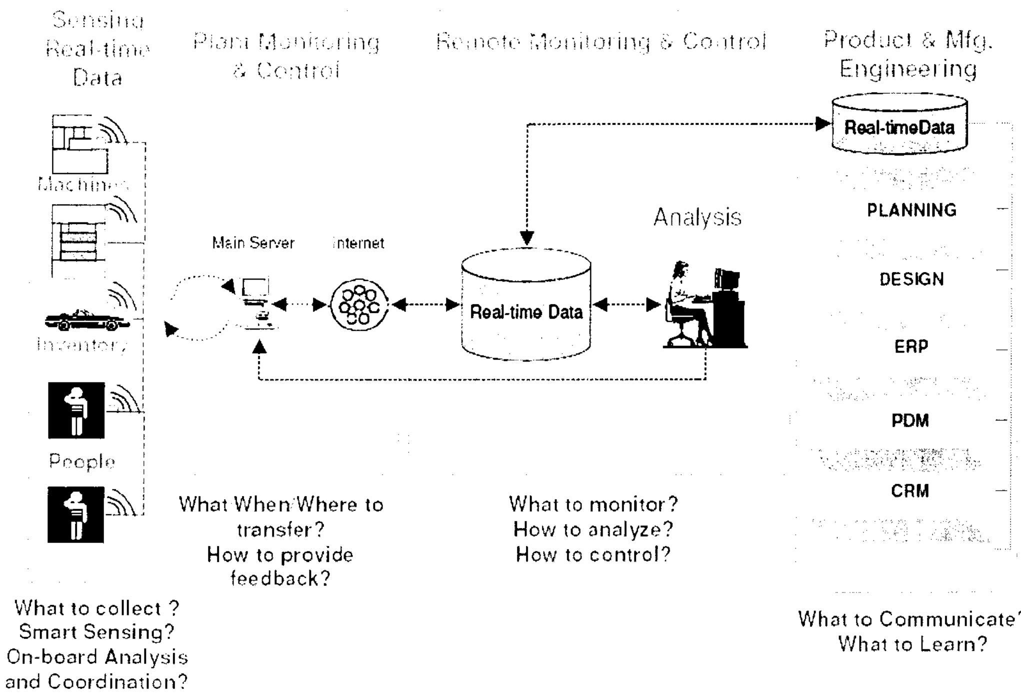


Figure 1.1: General e-Mfg Architecture with Real-time Data. (@ GM®)

Figure 1.2 shows a real-time fault detection and classification (FDC) system which is called FabStat, developed by Umetrics company together with IBM. It is commonly known that proper tool monitoring and fault detection are crucial in semiconductor manufacturing. FabStat is employed to monitor the entire semiconductor production line. When a fault occurs, it instantly identifies the out-of-control tool, and points to the leading culprit signals. The system reduces downtime of the manufacturing system by quickly identifying faults, their causes, and their solutions.

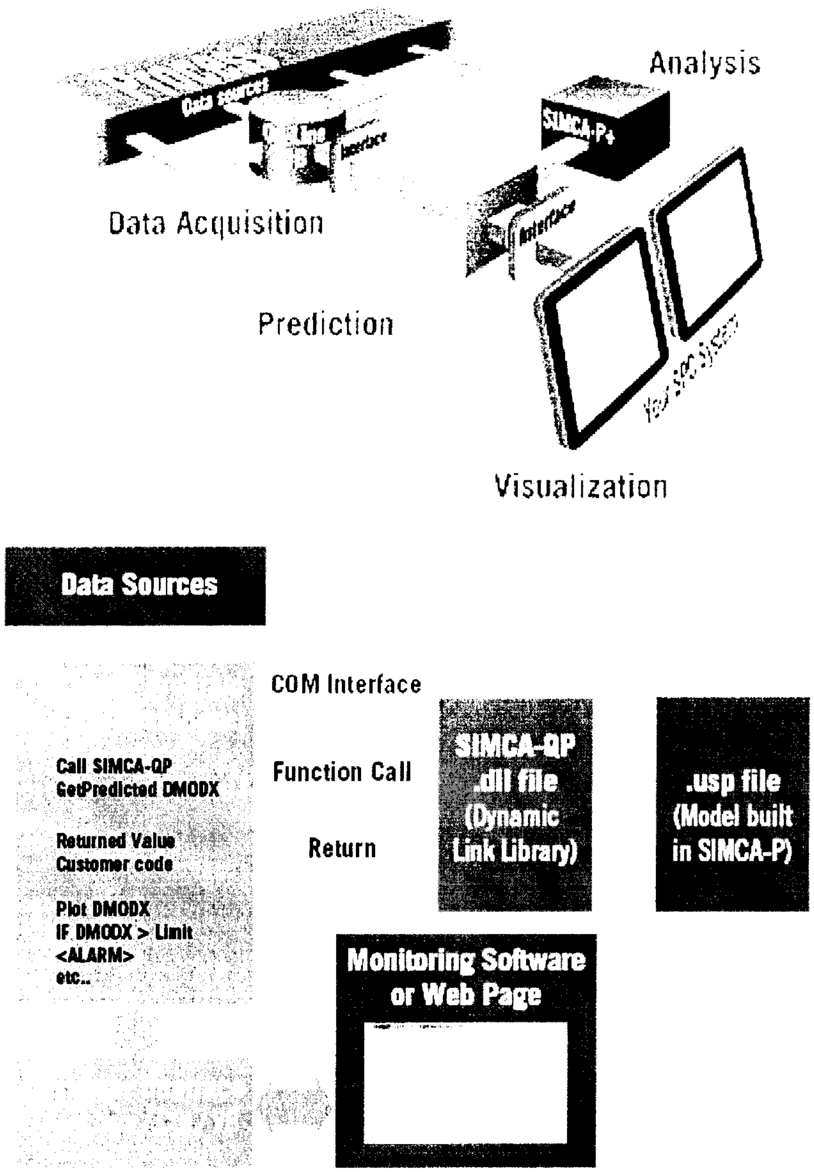


Figure 1.2: FabStat – Real-Time Monitoring, Fault Detection and Prediction in Semiconductor Manufacturing. (@ Umetrics®)

1.1 Goals of the Research

For a system with remote monitoring and fault diagnosis capabilities, the main challenges include the following:

- Redundant sensors
- Multi-sensor fusion
- Accuracy and robustness
- Time delay

The main objective of the research reported in this thesis is to design an intelligent sensor fusion architecture with neural networks and fuzzy logic, for a robust and reliable multi-sensor fault detection and diagnosis system. Three types of sensors are incorporated: sound, vibration and vision. After a data pre-processing operation, the signatures of the three signals are generated and sent to a diagnosis network. The purpose of the neuro-fuzzy fault diagnosis network is to map the preprocessed data onto specific faults of the machine. The developed architecture will be implemented and validated on a prototype industrial machine, and accuracy and reliability will be used as the criteria to evaluate the performance of the diagnosis system.

The second objective of the thesis is to develop an effective and universal network architecture for remote monitoring, and optimize the efficiency of the network communication so as to increase the real-time performance of the monitoring system. For these purposes, a web-based network architecture is proposed and developed in this thesis, which allows customers to monitor the machine in remote areas on a common PC via Internet Explorer (IE). Different data transfer strategies and data flow formats will be addressed based on different networking conditions.

An experiment setup will be developed using an industrial machine—fish cutting machine—to validate the developed neuro-fuzzy approach and web-based remote monitoring architecture.

1.2 The Fish Cutting Machine (Iron Butcher)

Fish processing is a major resource-based industry in the province of British Columbia, Canada. The annual wastage of fish during processing reaches up to millions of dollars in British Columbia alone. For the purpose of decreasing the meat wastage in the fishing processing industry, a fish cutting machine, called the Iron Butcher (de Silva, 1992), is designed in the Industrial Automation Laboratory at the University of British Columbia (UBC). This machine not only improves the recovery of useful meat in a significant way, it also helps replace the manual labor of human fish processors thereby moving them away from hazardous and unpleasant working conditions of a fish processing plant.

A view of the fish cutting machine is shown in Figure 1.3. There are three crucial sub-systems in the machine:

(1) Conveyer system

(2) Pneumatic system

(3) Hydraulic system

The main functions of the above three subsystems are to generate the motions of: the fish; the cutter in the horizontal positioning plane; and the cutter in the vertical direction, respectively.

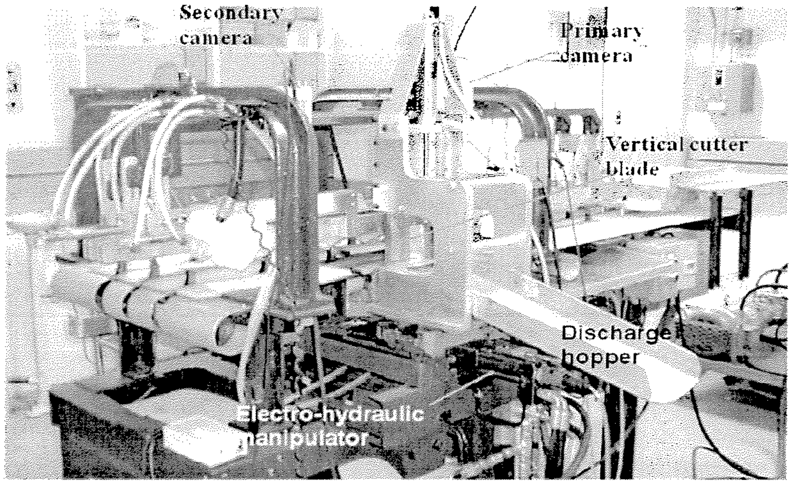


Figure 1.3: The fish cutting machine (Iron Butcher)

The normal operation of the fish processing machine is as follows: A worker places a salmon on the feeding table, which is located at one end of the machine, with its head pointed towards the cutter and its belly facing away from the direction of the conveyer motion. The feeding table allows the salmon to slide down to the conveyor bed, which transports the fish from the feeding table end to the cutter. A charge coupled device (CCD) camera snaps an image of the fish and on this basis the cutter controller moves the position of the cutter to the proper cutting location on the body of the fish. The cutting blade is specially shaped to remove both the fish head and the pectoral fin while minimizing the wastage of meat. Another CCD camera images the processed fish, and cutting quality is determined by analyzing the image. The cutting performance as determined in this manner may be used to adjust the parameters of the machine and its controller so as to improve the cutting performance.

It is clear that the Iron Butcher is a complex, nonlinear, and coupled mechatronic system (de Silva, 2005a), which contains different sub-systems that involve mechanical, electrical, electronic and information technologies. The failure of a sub-system can cause significant degradation of the machine performance and also incur financial losses. Therefore, an accurate and reliable fault detecting and diagnosis system is necessary for detecting and diagnosing potential faults and providing suitable corrective actions. The fish processing plant environment is not particularly suitable for human workers. Skilled operators and professionals should not be required to spend extensive periods of time in such an environment. It may not be logistically possible as well, to have all the human skills required for proper operation of a plant to be readily available at a specific location as needed. To address this problem, a remote monitoring system is developed implemented in the present work, which can provide operating condition of a machine in a real-time at a remote location. Furthermore, the technologies developed in the present thesis can help reduce the requirement of skilled human labor, leading to improvements in the cost-effectiveness of fish processing industry.

1.3 Scope of the Study

The scope of the study reported in this thesis includes two main areas: 1. Fault diagnosis with intelligent sensor fusion, 2. Remote monitoring of machinery. For the purpose of including a practical engineering project, several branches of technologies have to be considered and addressed:

- Acquisition and processing of multiple sensory data
- An intelligent sensor fusion architecture
- An effective training strategy
- Accurate and reliable mapping between machine signals and potential faults
- Real-time monitoring of machines in remote locations

1.4 Related Work

An extensive literature review has been carried out in order to establish the state of the art and benchmark to develop an effective diagnosis system, by investigating the technologies and challenges mentioned above. The following sections survey the particularly relevant work that has appeared in the past 10 years.

1.4.1 Fault Diagnosis System

Fault detection and diagnosis is the central component of abnormal event management (AEM) in industrial processes, which has received increased attention. Detection and diagnosis of process faults in an early stage when the plant is still operating in a controllable condition can greatly help in avoiding abnormal events and thereby improving the performance and reducing the loss of productivity. It is reported that the petrochemical industry alone loses approximately 20 billion dollars every year due to machinery failure. There is considerable interest in this field from industrial practitioners as well as academic researchers. There is a wealth of literature on process fault diagnosis ranging from analytical methods to approaches that use artificial intelligence and statistics, based on different sources of signals. The literature review in this thesis focuses on fault diagnosis systems and approaches that use vibration, sound and vision signals.

Vibration-based fault diagnosis has a long history. It is a rather effective method in machine monitoring and diagnosis, especially for rotating machinery. Each machine defect produces vibrations with distinctive characteristics that can be measured and compared with reference ones. Both time domain and frequency domain methods can be used to analyze vibration signals. However, the time domain approaches, which provide insight into the physical, time-evolutionary nature of vibration, can become practically impossible in modern complex machinery. For this reason, and in view of the frequency connotation of mechanical faults, particularly in machines with rotatory components, frequency domain analysis of vibration signals has become more popular in recent years. Fast Fourier Transform (FFT) and Wavelet analysis are two well known methods which help to transform time domain signals into the frequency domain and another useful variant (de Silva, 2005b).

Betta et al. (1998, 2002) proposed a digital signal processing (DSP)-based FFT analyzer for the fault diagnosis of rotating machinery using vibration analysis. The hardware and software of the DSP-based FFT-analyzer were introduced in their papers, and parametric non-fault and fault models, which were estimated by extracting specific characteristics in vibration spectrums, were integrated with rule-based reasoning for fault detection. It was claimed that the system could diagnose 99% of the situations. Tse et al. (2004) designed a novel wavelet transform called exact wavelet analysis, to minimize the effect of redundant information generated by overlapping and to enhance the accuracy of fault detection by extracting an adaptive daughter wavelet to match the inspected signal as exactly as possible. The results obtained from computer simulations and experiments showed that the fatal breakdown of machinery could be avoided by this approach.

Diagnostic methods using sound signals have existed for many years where FFT is applied to segments of time signals for analyzing the changes in Fourier spectra. These techniques are effective in cases where a specific "signature" of sound signal changes when a machine component fails or malfunctions.

Shibata et al. (1999) proposed a symmetric dot pattern (SDP) method for visually expressing the changes in the amplitude and frequency of sound signals. SDP involves the transformaion of a sound signal into a set of dots having mirror symmetry, in which the time waveform of the sound signal is visualised as a snowflake-shaped pattern of six-fold symmetry. It was claimed that this pattern could potentially be applied to the detection and characterisation of significant features of any sound signals. Benko et al. (2003) developed a sound-based fault detection and diagnosis system for vacuum cleaner motors. Root mean square (RMS), power spectrum density (PSD), short-time Fourier transform and Hilbert transform were applied for feature extraction. Three case studies were presented in the paper to illustrate how fault-free motors could be distinguished using sound analysis.

Vision-based approaches are among the most powerful methods of fault detection and diagnosis, especially in the field of industrial quality control. Ahmed Abouelela et al. (2004) discussed an automated visual inspection system for detecting textile defects. Simple statistical features (mean, variance, median) were employed to guarantee the real-time performance. The results indicated that the approach had a very low false detection rate. Verma et al. (2005) presented an algorithm that provided a computationally tractable approach for robot fault diagnosis. Vision information was used to monitor the states of robots so that anomalous situations could be detected in a timely manner. A dynamic simulation of a six-wheel rocker-bogie rover showed very good performance over classical approaches.

Sometimes, a single sensor is not adequate to detect all possible faults that may occur in complex machinery within practical environments. In view of this, multi-sensor approaches with data fusion are considered to improve the diagnosis accuracy under these conditions. A review of current research in sensor fusion/ data fusion will be presented in the next section.

1.4.2 Sensor Fusion

Multi-sensor systems are designed to exploit several signature-generation phenomena and to gather different types of information about objects and scenes of interest. Design of a multi-sensor system involves optimization of sensors, data processing, and communication, and particularly the use of an appropriate fusion strategy; e.g., Bayesian and Dempster-Shafer inference; fuzzy logic; pattern recognition using signal processing algorithms, and artificial neural networks. The backbone of a multi-sensor system concerns how to utilize various data by using an effective sensor /data fusion approach.

Sensor fusion first appeared in the literature in 1960s. Today, application of sensor fusion has expanded into a wide range of areas: maintenance engineering, robotics, pattern recognition, object tracking, and so on. Figure 1.4 shows a general architecture of a sensor fusion system which includes low-level data acquisition and processing and high level data fusion processing. Low-level processing concerns the hardware level and the raw data processing; e.g., sensor selection and arrangement and data acquisition, while the high-level processing focuses on extracting the raw data of interest and transforming it into utilizable forms. A decision making system that uses the pre-processed information is also embedded in the high-level processing layer.

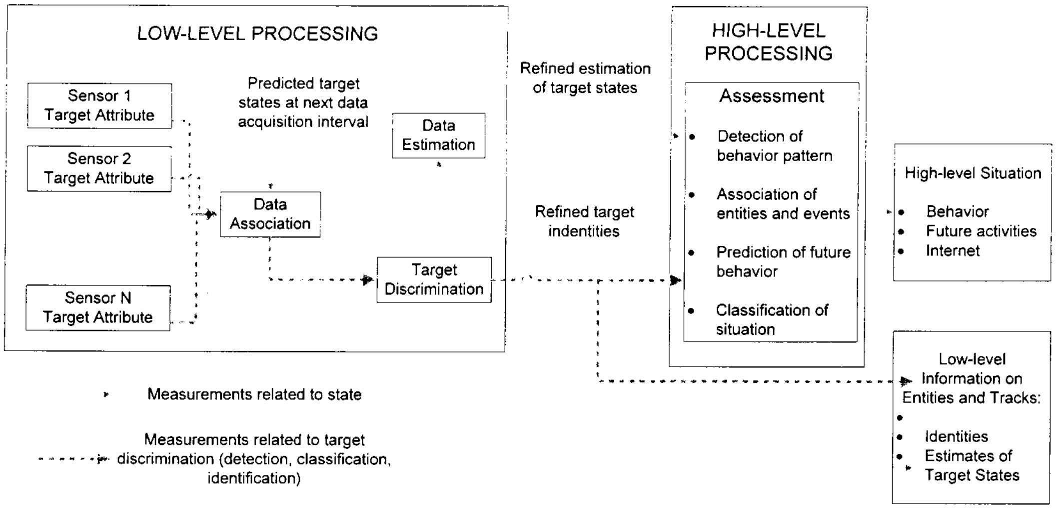


Figure 1.4: General architecture of data fusion processing.

The use of artificial neural networks (ANNs) is a commonly applied approach to solve the data fusion problem. It was introduced by Posner (1989) with the objective of understanding the functioning of the human brain. He built models of natural neural networks in the brain and carried out simulation studies. The general idea of ANNs is to make a nonlinear transformation from a d-dimensional input space to a h-dimensional output space through an appropriate number of hidden layers. Through appropriate training, the network weights are adjusted, establishing a reasonably accurate nonlinear relationship between inputs and outputs.

Ghosh et al. (2007) proposed a neural network-based sensor fusion architecture for the estimation of tool wear of a computer numerical control (CNC) milling machine. Monitoring of tool wear is crucial in preventing degradation of the machining quality. Unfortunately, there is no direct way of measuring the process variables related to tool wear. An ANN based sensor fusion approach has been proposed by Ghosh et al. (2007) to fuse the data of cutting force, spindle vibration, spindle current, and sound level from different sensors. The approach had been validated by both laboratory and industrial implementation. Hong et al. (2005) developed a neural network based sensor fusion system for real-time fault detection of reactive ion etching. The target of this project was to guarantee the system accuracy and real-time performance. Two in-situ sensors: optical emission spectroscopy (OES) and residual gas analysis (RGA) were used and the generated signals were sent to a time series neural networks (TSNNs) for fusing as well as predicting the process parameters. Simulated fault processing data was used to train the NN and the author claimed that this approach could potentially contribute to maintaining a consistent etching process by increasing the probability of identifying incipient faults.

Considerable amount of work has been done in sensor fusion where fuzzy logic is implemented as the fusing method. Lotfi Zadeh (1978) developed the fuzzy set theory in 1965. Zadeh reasoned that the rigidity of the conventional set theory made it impossible to account for vagueness, imprecision, and shades of gray that are commonplace in real-world events. Consequently, fuzzy logic is valuable where the boundaries between sets of values are not sharply defined or there is partial occurrence of an event.

Although fuzzy logic and neural networks are structurally different, they share a rather complementary nature as far as strengths and weaknesses are concerned (Karray and de Silva, 2004). Applying fuzzy methods into the workings of neural networks constitutes a major thrust of neuro-fuzzy (NF) computing. Wang et al. (2004) proposed a neuro-fuzzy system to forecast damage propagation trend in rotary machinery and to provide an alarm before a fault reaches critical levels. After proper training, the performance of the NF was compared with the performance of a recurrent neural network (RNN). It showed that the NF was a reliable and robust machine health predictor which could capture the system dynamic behavior quickly and accurately. Palluat et al. (2006) designed a neuro-fuzzy based intelligent monitoring aid. The system contained a neural-network detection tool and a neuro-fuzzy diagnosis tool. Four sensors were used in gathering information. After training, the NF demonstrated industrial usefulness in an application of monitoring a flexible production system.

1.4.3 Machine Vision

Machine vision (MV) involves the application of image processing and interpretation (computer vision), which has numerous applications in industry. It requires digital input/output devices and computer networks, combined with image processing techniques, to control manufacturing equipment including robotic arms. Machine vision is a subfield of engineering that encompasses computer science, optics, mechanical engineering, and industrial automation. A common application of machine vision is the inspection of manufactured goods such as semiconductor chips, automobiles, food products, and pharmaceuticals. Just as human inspectors in assembly lines visually inspect parts to judge the quality of workmanship, a machine vision system can carry out quality assessment task with good accuracy and repeatability. Another recent application is vision-based object detection and tracking of UGVs (Unmanned Ground Vehicles) and path planning and following of mobile robots (visual servoing). This technology can also be implemented in security and transportation applications such as video surveillance and traffic control.

The first step of correctly tracking or accurately measuring an object by a vision system is to correctly detect the object. Vision-based automated object detection has played a significant role in industrial and service applications. Studies (Sun et al. (2006) and Park et al. (2004)) have focused on detecting objects efficiently by using color, shape, size, and texture features. However, there are a number of problems that arise when using these methods to process real world images because the detection methods based on low-level pixel features cannot meet requirements in different conditions and environments. Although it is possible to improve the detection accuracy by combining these approaches with human vision concepts, it is difficult to automate the process and therefore inconvenient. Thus, although various methods have been considered for processing real-world images, their performance is not yet sufficient for common practical use.

Fortunately, the computer vision technology has dramatically improved in the past several years and has been demonstrated in a number of successful applications. A well-known project is the DARPA Grand Challenge. It involved an Unmanned Ground Vehicles (UGVs) racing in a desert terrain. Object detection technology helped to solve problems of both environmental modeling and obstacle detection. Other applications such as defect detection on textured surfaces (e.g., ceramics, textile) use the detection technology in combination with knowledge-based systems in automated production lines in industry.

There are two steps in object detection: 1) Hypothesis Generation (HG) where the locations of the possible objects in an image are hypothesized, and 2) Hypothesis Verification (HV) where tests are performed to verify the presence of the objects in an image. These two steps are described next.

HG (Hypothesis Generation) Method

Several useful methods have been developed in the field of HG, such as the symmetry-based method, color-based method, corner-based method and vertical/horizontal edges-based method, which help quickly find candidate objects from an image for further exploration.

Seelen et al. (2000) proposed a symmetry-based method that used three steps to solve the object detection problem. Firstly, a fusion process of simple features was used to reduce the amount of information. By using a neural network and LOC (Local Orientation Coding) method, an image with reduced information could be generated from the original image. After the reduction of the whole image, two methods were used in combination to extract the object positions: Symmetry Analysis and Model Matching. The first method was used to detect the rear, front and side views of all the object types by the measurement of inherent vertical symmetric structure. The symmetry detection was formulated as an optimization problem, and solved by a neural network. Finally, model matching was used to backup the fusion and symmetry results. Although this method provided good performance according to the case study presented in the paper, it had the limitation that its performance was easily affected by the changes in illumination and slight differences in the right and left parts of the object. Therefore, the symmetry-based method is still a challenging undertaking under real conditions

As commonly known, color can be a useful feature in object detection. However, few existing detection and tracking applications use color for recognition, because color-based recognition is complicated, and existing machine vision techniques that use color have not been shown to be effective, especially in realistic outdoor images.

Buluswar et al. (1998) explored an effective solution of real-time color recognition in outdoor scenes. The authors presented a technique for achieving effective real-time color recognition in outdoor scenes. The technique used MDT (Multivariate Decision Tree) for piecewise linear non-parametric function approximation to learn the colors of target objects from training samples, and then detected targets by classifying pixels based on the approximated function. This method has been successfully tested in several domains, such as autonomous highway navigation, off-road navigation and target detection for unmanned military vehicles.

Exploring the fact that some objects have rectangular shapes with four corners, Bertozzi et al. (1997) proposed a corner-based method to hypothesize the vehicle locations. The system in this paper was composed of a pipeline of two different engines: PAPRICA, a massively parallel architecture for the efficient execution of low-level image processing tasks, improved by the integration of a specific feature for direct data input/output (I/O); and a traditional serial architecture running medium-level tasks aimed at the detection of the vehicle position in the sequence. Moreover, a preliminary version of the system was announced and was demonstrated on the MOB-LAB land vehicle.

The use of constellations of vertical and horizontal edges has shown to be a strong cue for hypothesizing objects. In an effort to find pronounced vertical structures in an image, Matthews et al. (1996) used edge detection to determine strong vertical edges. To localize the left and the right positions of a vehicle, they computed the vertical profile of the edge image followed by smoothing by using a triangular filter. By finding the local maximum peaks of the vertical profile, they were able to find the left and the right positions of a vehicle. Goerick et al. (1996) proposed a method called Local Orientation

Coding (LOC) to extract edge information. An image obtained with this method consisted of strings of binary codes representing the directional gray-level variation in the neighborhood of a pixel. These codes carry essential edge information.

HV (Hypothesis Detection) Method

After carrying out the HG step in the detection, information is sent to HV (Hypothesis Verification). During HV, tests are performed to verify the correctness of a hypothesis. Approaches of HV can be classified mainly into two groups: 1) template-based methods, and 2) appearance-based methods.

Template-based methods use a predefined pattern of the object class and perform correlation between the image and the template. Handmann et al. (2000) proposed a template based on the observation that the rear/frontal view of a vehicle has a "U" shape (i.e., one horizontal edge, two vertical edges, and two corners connecting the horizontal and vertical edges). During verification, they considered a vehicle to be present in the image if they could find the "U" shape. Ito et al. (1995) used a very loose template to recognize pronounced vertical/horizontal edges and symmetry. Due to the simplicity of the template, they did not expect very accurate results, which was the main reason for employing active sensors for HG.

Appearance-based methods learn the characteristics of object appearances from a set of training images which capture the variability in the object class. Wu et al. (2001) used standard Principal Components Analysis (PCA) for feature extraction, together with a nearest-neighbor classifier, reporting a level of accuracy of 89%. Bertozzi et al. (1998) used PCA for feature extraction and a Neural Network for classification. First, each sub-image containing object candidates was scaled to 20×20 followed by subdividing it into twenty five 4×4 sub-windows. PCA was applied on every sub-windows (i.e., "local PCA") and the output was provided to a Neural Network to verify the hypothesis.

1.4.4 Remote Monitoring

Remote monitoring is a somewhat mature area in the field of computer science. It has been applied in many fields of science and engineering. For example, mechanical engineers use this in large power equipment, where it is integrated with fault diagnosis techniques. Process engineers use it in process-condition monitoring of mass-production lines to detect defects in products and optimize the production quality. Scientists employ this technology to monitor air quality or creatures living in the deep ocean. There are several commercial products such as remote video monitoring employed in elevators at airports and other security applications.

The Internet has provided an intermediate transmission medium to support remote monitoring. A number of Internet-based remote monitoring systems such as DigiEye, TeleEye Pro, Digital Surveillance and Vision-based Traffic Surveillance System have been developed and marketed, which are outlined by Fucik et al. (2004). These systems can be connected to the Internet through a phone line or a Local Area Network (LAN), and provide multiple remote monitoring functions such as live video viewing, digital video recording and motion detection.

Web-based remote monitoring is a network-based remote monitoring approach, which is constructed under the B/S (Browser/Server) network architecture. The web-based remote monitoring has the following benefits: 1) Relatively small data storage space is required by the remote clients, 2) Software need not be installed on the remote client side. 3) Any computer that connects to the Internet can be used as a remote monitoring client.

A large number of applications have been implemented in industry. Fucik et al. (2004) presented a system called Unicam for traffic monitoring applications, which uses remote monitoring and machine vision technologies. The system provides a real time video capture and transmission over communication interfaces. For example,, the system can be used to detect red-light violations at road intersections, speed measurement, traffic data collection, and video recoding or surveillance. Hui et al. (2003) compared differences among the traditional monitoring, Internet-based monitoring, and web-based monitoring. Next the authors introduced a web-based remote monitoring scheme called iSecure with intelligent monitoring, and presented two applications.

Due to the advancements of the computer and Internet technologies, another concept and has been proposed and research and developed have been carried out, which devotes to visualizing a stimulated environment of the real operating environment through the Internet from anywhere and at anytime. This technology is called Virtual Reality (VR), including Virtual Environment (VE), Virtual Laboratory (VL), and so on.

Marini et al. (1997) demonstrated a methodology to provide a new insight into cultural heritage which is based on the techniques of virtual reality, web navigation tools, advanced image analysis and photorealistic image synthesis methods. A VRML (Virtual Reality Modeling Language) model of the ancient Roman Theatre of Aosta was described for the purpose of examining possible restoration and conservation choices. Java script was implemented in this project. Virtual manufacturing is another useful application. Rubio et al. (2005) reviewed some of the more recent developments and methodologies for manufacturing systems, which provided a low-cost, secure and fast analysis tool. With this technology, globally located collaborating institutes are able to share resources and expertise. It can be used both in industrial and academic or research fields. Rohrig (1999) developed a network of remotely accessible laboratories called Virtual Lab using a client-server (CS) architecture. This system makes the interaction between students and instructor more interactive and convenient.

Several products of commercial monitoring and video broadcasting software are available; e.g., Microsoft® Windows Media Services, RealNetworks® Helix Platform, and Apple® Darwin streaming server 4.1. They provide a powerful development kit for the software engineers to easily embed the video stream servers into webservers. However, they have several drawbacks in practical applications since they need a universal architecture of a stream server for broadcasting video data in a certain domain in the network. Therefore, the broadcasting strategy cannot guarantee real-time performance in industrial applications. Furthermore, the heavy flow of video data will considerably occupy network resources, which can affect the performance of other network-based software and devices.

1.5 Contribution and Organization of the Thesis

The main contributions of this thesis are listed below.

1. A neuro-fuzzy (NF) data fusion architecture is proposed and developed for fault detection and diagnosis. The neuro-fuzzy network fuses features of data from different sensors. This architecture can fuse different data from various sensors. After

adequate training, a mapping can be setup between the input variables and the outputs.

2. The NF approach is implemented in an industrial machine--a fish cutting machine. Three types of sensors (accelerometers, microphones and CCD cameras) are used to acquire vibration, sound and vision signals, respectively, from an operating machine. After data acquisition and data preprocessing, groups of sample signals are sent to the NF data fusion system for offline training. This NF data fusion system can be implemented as a fault detection and diagnosis system.
3. Software and hardware aspects of a low cost, universal web-server network infrastructure is presented and implemented for remote monitoring of the fish cutting processing.

The organization and contents of the individual chapters of the thesis are summarized below.

Chapter 2 illustrates the overall system architecture in both hardware and software. Sensor selection and mounting methods are discussed. The details of data acquisition of vibration, sound and vision signals are presented followed by detailed algorithms of the data preprocessing strategies.

Chapter 3 describes the constructive NF data fusion architecture, which is the core of the fault detecting and diagnosis system developed in the thesis. The detailed fusion approach of neural networks and fuzzy logic is discussed. The training strategy of the NF network is demonstrated.

Chapter 4 presents the framework of infrastructure development for web-server based monitoring of a remote system, which includes hardware commissioning, and the development of the network architecture and software in both video stream server and web-based clients. This system is demonstrated on an industrial machine.

Chapter 5 presents the experimental results in both fault detection and diagnosis, and remote monitoring. Benchmarks are set in this chapter to evaluate the performance of the approaches developed in the previous chapters. The test results are discussed with respect to the accuracy and reliability criteria of the fault detection and diagnosis system, and the real-time and web-based criteria of the remote monitoring system.

Chapter 6 concludes the thesis by summarizing the overall research that has been carried out, implemented and validated. Some limitations of the work are indicated and suggestions are made of possible directions for future work.

Chapter 2

System Architecture

The main objective of this chapter is to introduce a general architecture for a fault diagnosis system, which will then be implemented in an industrial fish processing machine called the Iron Butcher. The machine has been his described in Chapter 1. The structure of the hardware and the design and development of the software of the fault diagnosis system will be presented in the present chapter. In particular, the arrangement and specifications of the sensors will be introduced in section 2.1. Moreover, the software design environment, which will facilitate data processing and neuro-fuzzy inference, will be illustrated in this section.

This chapter will also present the data acquisition and pre-processing strategies for the three source signals: sound, vibration and vision. Insight into the data acquisition software for the vibration and sound signals will be provided in section 2.2. Fast Fourier Transform (FFT), an algorithm which efficiently transform time domain signals into the frequency domain, will be introduced in section 2.3. Section 2.4 focuses on the image processing methods as applied in object tracking. In order to adopt the neuro-fuzzy intelligent fault diagnosis system, which will be discussed in Chapter 3, a method of generating feature vectors will be presented in section 2.5.

2.1 System Overview

The overall experimental architecture and data flow of the fault diagnosis system are shown in Figure 2.1. Three types of sensors, indicated by red circles: accelerometers, microphones and digital CCD cameras, are mounted on the machine. The outputs of the sensors are connected to a signal processing centre: a Pentium® IV computer, which carries a data acquisition board, a sound blaster card and a frame grabber. This signal processing centre also serves as a web server and a video server, which is connected to a general office Ethernet-based local area network. The detailed infrastructure and software modules of the remote monitoring system will be explored in Chapter 4.

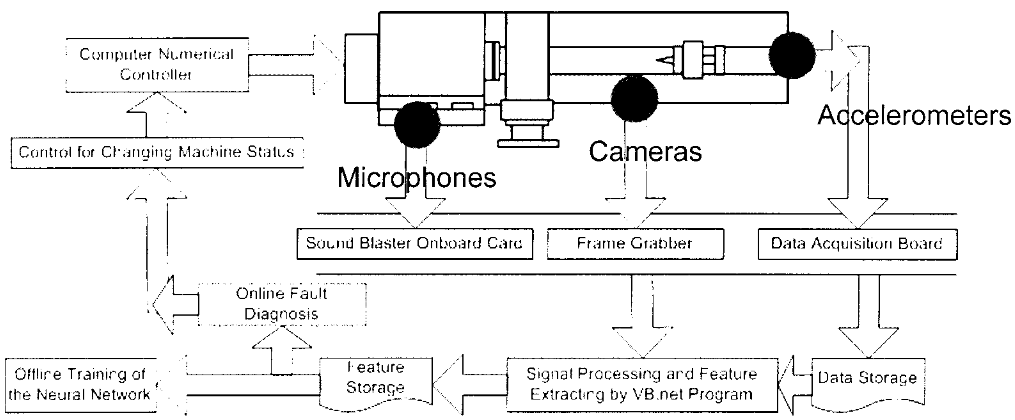


Figure 2.1: Architecture and data flow layout of the fault diagnosis system.

Customized software is developed and installed in the data processing centre to acquire the vibration, sound and vision signals, and store the real-time data while the machine is operating. This software has other functions as follows:

- Display the real-time signals of vibration and sound on the screen
- Display the real-time vision signals on the screen

- Transform the time domain signals of vibration and sound into the frequency domain with a FFT function and display the results on the screen
- Display results of the feature-based vision tracking program
- Generate feature vectors from vibration, sound and vision signals
- Generate neuro-fuzzy intelligent inference for fault diagnosis
- Web and video server programs for remote monitoring

2.1.1 System Hardware

In this section, detailed information of the sensors and relevant hardware devices which are incorporated in this project are introduced and their specifications are given. There are three groups of sensors for the purpose of acquiring three types of signals: vibration, sound and vision.

Transducers/sensors are devices that transducer (convert) physical variables such as temperature, pressure, and light intensity into electronic signals such as voltage and current. In this project, piezoelectric accelerometers are employed to measure vibration of the machine by converting the acceleration signals into voltage signals (de Silva, 2007a). Figure 2.2 shows one set of the hardware for vibration signal acquisition. It contains one Kistler® accelerometer, one Kistler® power supplier/conditioner, and a signal processing desktop computer with an NI® 16 bits data acquisition (DAQ) board.

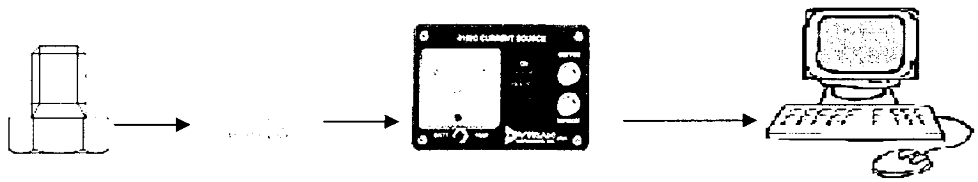


Figure 2.2: Diagram of the low impedance acceleration sensing system.

An accelerometer attached to an object can be modeled as a single degree-of-freedom vibrating system excited by a moving base. The principle of common piezoelectric accelerometers (de Silva, 2005a, 2007) is that the mass serves as a front-end element to convert the acceleration into an inertia force according to d'Alembert's principle and use that force to strain a piezoelectric element, which will generate a charge in proportion to the acceleration. The charge is sensed without leakage and simultaneously amplified using a charge amplifier. The specifications of the accelerometers used in the present work are listed in the Table 2.1.

Table 2.1: Specifications of the K-Shear® Accelerometer, 25g Sensor Type 8702B25.

Specification	**Units**	**Value**
Range	g	±25
Sensitivity	mV/g	±200
Frequency Range	Hz	1...8000
Resolution, Threshold	mg rms	2
Shock	g	2000
Transverse Sensitivity	%	1.5
Operating temperature range	°C	-55...100
Temp. coef. of sensitivity	%/°C	-0.054
Mass	g	8.7
Height	Mm	20.3

However, piezoelectric accelerometers are inherently high output impedance devices that generate small voltages (usually in mV level). For this reason, a charge amplifier (having capacitance feedback) with conditioner is required to trap the generated charge and magnify the output voltage signal of the sensor. In this project, a Kistler® 4-channel piezoSmart® power supply/ signal conditioner is used. It is a microprocessor controlled coupler which provides DC power and signal processing for 4 channels of low impedance, voltage mode piezoelectric pressure, force or acceleration sensors. Its specifications are given in Table 2.2.

Table 2.2: Specifications of the Kistler® Power Supplier/ signal Conditioner.

Specification	**Units**	**Value**
Sensor Excitation Current	mA	max.15
Sensor signal voltage	V	24
Frequency Range	Hz	0.1...68000
Output signal	V	max.10
Operating temperature range	°C	0...60
Width	Mm	93.5
Height	mm	141
Depth	mm	195.00
Mass	kg	1.75

In order to acquire voltage signals from the signal conditioners for further analysis in a desktop computer, a DAQ board, which is a versatile and powerful measurement device, is necessary. In this project, A National Instruments (NI®) PCI-6221 (37-Pin) DAQ board is employed in this project to carry out data exchange between the signal conditioner and the computer. It is a typical commercial DAQ card containing the ADC and DAC units that allow input and output of analog and digital signals in addition to digital input/output channels. The main specifications of this board are given in table 2.3.

Table 2.3: Analog channel specifications of the NI® PCI-6221 (37-Pin) DAQ board.

Specifications	**PCI-6221 (37-Pin)**
Number of Channels	2
Update Rate	833 kS/s
Resolution	16 bits
Maximum Voltage Range	-10 to 10 V
Range Accuracy	3230 μV
Range Sensitivity	3230 μV
Minimum Voltage Range	-10..10 V
Range Accuracy	3230 μV
Range Sensitivity	3230 μV
Current Drive (Channel/Total)	5 mA
I/O Connector	37-pin D-Sub

The sound signals are acquired by a microphone and a Creative® Sound Blaster board. A microphone is a device made to capture pressure waves in air, water (hydrophone) or a solid medium and translate them into electrical signals. The most common method is via a thin membrane producing a proportional electrical signal. Most microphones in use today for audio use electromagnetic induction (dynamic microphones), capacitance change (condenser microphones) or piezoelectric generation to convert the membrane motion into an electrical signal. A sound card (also known as an audio card) is a computer expansion card that facilitates the input and output of audio signals to/from a computer under the control of computer programs. Typical uses of sound cards include providing the audio signals for multimedia applications such as music composition, editing video or audio signals, presentation/education, and entertainment (games). Many computers have sound capabilities built in, while others require additional expansion cards to provide for audio capability. In the present research project, a microphone and a Creative Sound Blaster Card are employed.

Two CCD cameras and one frame grabber have been used in the present work to capture digital images of the moving parts of the machine. Digital camera is an electronic device used to capture and store photographs digitally, instead of using photographic film as in conventional cameras, or recording images in an analog format to a magnetic tape like in conventional video cameras. In this project, one charge-coupled device (CCD) digital camera is mounted on the top of the machine, which is used to track the motion of fish along the conveyor. Another camera is employed as a global camera which records and monitors the condition of operation of the fish cutting machine. The quality of the processed fish may be imaged as well. The frame grabber is a component of a computer vision system designed for capturing an image view (frame) from the camera in a digital form and storing in a data buffer. Reconstruction of images from digital protocols may be facilitated as well by the frame grabber. Typically, a frame grabber consists of:

- The physical interfaces to the camera such as Camera Link, GigE Vision, LVDS or RS-422.
- Image reconstruction engine to re-assemble the image.
- On-board memory for storing the acquired images (frame buffer).
- A bus interface through which the main processor can control the acquisition and access of the data.
- Some general purpose I/O for triggering of image acquisition or control of external equipment.

The core of the hardware is a data processing computer which hosts a data acquisition board, a sound blaster board and a frame grabber, and is connected to the Ethernet network through a 10BaseT network interface card (NIC). It consists of one Pentium IV 2.0GHz processor running Windows 2000 with in-house-developed software. It also acts as a web server and a video server, which will be discussed in Chapter 4.

2.1.2 System Software

Development, use and application of computer software constitute a significant part of the present project. In particular, the work employs four software subsystems for: data acquisition, data preprocessing, neuro-fuzzy inference and remote monitoring. Two types of computer programming language, Visual Basic and Java, have been used to develop software for accomplishing these functions.

Visual Studio is a Microsoft's flagship software development tool for computer programmers. It provides a friendly programming environment for Basic and C/C++ programmers. Moreover, it has strong compatibility with the Microsoft operating systems (Windows 2000, Windows XP, etc.) and is well supported by Windows API for hardware level programming. In the present project, Visual Studio is used to accomplish the tasks of data acquisition, signal processing and neuro-fuzzy inference in view of of its easily designed GUI (Graphic User Interface) for Windows applications.

Java language is also a powerful computer programming language originally designed by SUN Micro systems and released in 1995. Its basic features such as platform independence, object oriented programming, robust and automatic memory management, security, and built-in multi-thread networking capabilities, make it a very powerful and popular programming language. Moreover, because of its good integration with network technologies, Java programming language is particularly suitable for web-based software applications.

Several other computer technologies are applied in this thesis to enhance the system performance. For example, Windows API (Application Programming Interface), informally WinAPI, is one of them. It is the core set of application programming interfaces available in Microsoft Windows operating systems, and it provides low-level hardware access to a Windows system from third-party software (sound card access, etc.). ActiveX object is another Microsoft technology implemented in this project, which is used for developing reusable object oriented software components. ActiveX is mainly used here to build plug-ins for Internet Explorer. In particular, in the present project, an ActiveX component has been designed for the purpose of remote monitoring and control of the local camera from remote clients.

2.2 Data Acquisition of Vibration and Sound Signals

This section describes how vibration, sound and vision data are acquired and stored in a computer for further processing, in the present project. Data acquisition involves acquiring of data from the real world application through sensors, which can be sampled (discretized) and digitized (converted into digital form) for subsequent manipulation by computers. Data acquisition typically involves acquisition of signals and waveforms and processing them to obtain the desired information. The components of a data acquisition system include sensors that sense and convert the measurement variables into electrical signals, and data acquisition hardware which acquires and transmits the electrical signals into a computer. In addition, customized software is developed to control the data acquisition hardware.

2.2.1 Signal sampling

In signal processing, sampling is the operation that converts a continuous signal into a discrete signal (sequence of data points, typically at equal space—sampling period). It is a rather crucial part in a signal acquisition system. An important consideration is sampling rate or the speed at which a signal is sampled. Since the fastest fluctuations in the waveform correspond to the highest frequencies, it is clear that these frequencies will be lost if the sampling frequency is too slow. On the other hand, oversampling is not desirable as it will waste the hardware and software resources. Shannon's famous *Sampling Theorem* states that (de Silva, 2007a) an analog signal containing components up to some maximum frequency f_l Hz may be completely represented by regularly-spaced samples, provided the sampling rate is at least $2f_l$ samples per second. For example, an audio signal having f_l=4 kHz should be sampled at least 8000 times per second. Note that this corresponds to *two samples per period* of the highest frequency present, or the sampling interval T is clearly: $T=1/2f_l$.

In this section, the process of sampling and digitalizing the acquired vibration and sound signals is introduced. In the present project, the important frequencies of the vibration signals of the fish cutting machine range from 0 Hz to 300 Hz. According to the Nyquist-Shannon sampling theorem, the sampling frequency should be higher than 600 Hz. Therefore, the sampling frequency for vibration signals is selected to be 1000 Hz which is 400 Hz higher than the highest machine frequency.

In this project the sampling rate for sound signals is chosen as 44,100 Hz which covers the entire 18-20,000 Hz range of human hearing and meets the requirements of the Nyquist-Shannon theorem. In addition, the audio signals are recorded at a 16-bit resolution. Figure 2.3 shows some real sampling results of the vibration and sound signals of the fish-cutting machine.

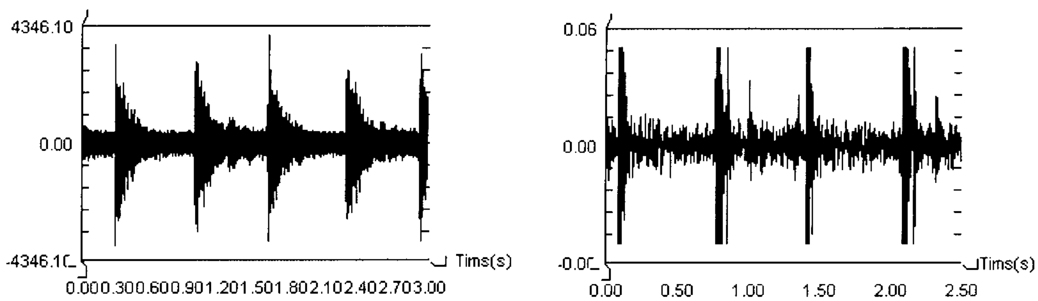


Figure 2.3: Sampling results of vibration and sound signals.

Vibration and sound signals are acquired for three seconds. After sampling, these digitalized vibration and sound data are stored in a computer and transformed into the frequency domain using FFT (Fast Fourier Transfer), for further processing.

2.2.2 Data storage

When data is acquired from the sensors, it is stored in the computer according to a customized format for further processing. Figure 2.4 shows the file format of vibration and sound data in this project.

```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 52 49 46 46 C8 4D 0E 00 57 41 56 45 66 6D 74 20 ; RIFF萜..WAVEfmt
00000010h: 10 00 00 00 01 00 02 00 44 AC 00 00 11 2B 00 00 ; ........D?..+..
00000020h: 04 00 10 00 64 61 74 61 A4 4D 0E 00 02 00 02 00 ; ....data  ......
00000030h: 03 00 03 00 03 00 03 00 03 00 03 00 02 00 02 00 ; ................
00000040h: 01 00 01 00 00 00 00 00 FF FF FF FF FF FF FF FF ; ........
00000050h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000060h: 00 00 00 00 01 00 01 00 02 00 02 00 03 00 03 00 ; ................
00000070h: 04 00 04 00 05 00 05 00 05 00 05 00 05 00 05 00 ; ................
```

Figure 2.4: The storage format of vibration and sound data.

It is a text file containing a file head in the front of the file which describes the data and sampling information of the vibration and sound signals. Table 2.3 gives the definition of the file head. According to the definition, Figure 2.3 is a sound file with 2 channels. The sampling frequency is 44100, and the length of the data is 19912.

Table 2.4: The definition of the file head.

File Address	Data Type	Length	Description
00H	char	4	RIFF (Resource Interchange File Format)
04H	long	4	Length of the file
08H	char	4	Data format in the file
0CH	char	4	Format mark
10H		4	Temp area
14H	int	2	Data format
16H	int	2	Number of channels
18H	int	2	Sampling rate
1CH	long	4	Speed of the transmission
20H	int	2	Size of the buffer
22H	int	2	Single channel or dual channel
24H	char	4	" data " mark
28H	long	4	The length of data

2.2.3 Data acquisition software

The main components of a vibration data acquisition software are given below:

```
DAQmxErrChk DAQmxCreateTask("", taskHandle)
DAQmxErrChk _
DAQmxCreateAIVoltageChan(taskHandle, lblPhysicalChannel.Caption, "", _
DAQmxInputTermCfg.DAQmx_Val_InputTermCfg_RSE, -10, 10, _
DAQmx_Val_VoltageUnits1_Volts, "")
DAQmxErrChk   DAQmxCfgSampClkTiming(taskHandle,   "OnboardClock",   _
    lblVibrationFrequency.Caption, DAQmx_Val_Rising, _
DAQmx_Val_AcquisitionType_FiniteSamps, CLng(lblVibrationSample.Caption))
DAQmxErrChk DAQmxGetTaskNumChans(taskHandle, numChannels)_
    arraySizeInSamps = numSampsPerChannel * numChannels
DAQmxErrChk DAQmxReadAnalogF64(taskHandle, numSampsPerChannel, 10#, _
    fillMode, data(0), arraySizeInSamps, sampsPerChanRead, ByVal 0&)
DAQmxErrChk DAQmxStopTask(taskHandle)
DAQmxErrChk DAQmxClearTask(taskHandle)
```

For sound signal acquisition in this project, windows API is applied, which is the core set of application programming interfaces (APIs) available in the Microsoft Windows operating systems. It allows a client program to interact with a Windows operating system, especially for the purpose of lower level accessing to hardware via device drivers. The main components of the sound acquisition API are given below:

```
mciSendString("close all", 0&, 0, 0)
mciSendString("open new type waveaudio alias capture", 0&, 0, 0)
mciSendString("set capture channels 2", 0&, 0, 0) ' 2 channels for stereo
mciSendString("seek capture to start", 0&, 0, 0) 'Always start at the beginning
mciSendString("set capture samplespersec 44100", 0&, 0, 0) 'Signal Quality
mciSendString("set capture bitspersample 16", 0&, 0, 0)  '16 bits for better sound
mciSendString("record capture", 0&, 0, 0)  'Start the recording
```

2.3 Data Processing of Signals (Fast Fourier Transform or FFT)

Signal processing is the task of analyzing, interpreting and manipulating the raw signals provided by the data acquisition system. It involves multiple signal operations such as filtering, compression, transformation and feature extraction. In the present research, the FFT (Lathi, 2002) algorithm (Fast Fourier Transform) which transforms time domain signals into the frequency domain at high speed is used to analyze the acquired sensory signals. The FFT algorithm is introduced next.

Fourier analysis has been applied to continuous-time (analog) phenomena for nearly two hundred years. It has had a profound influence on many branches of engineering and applied science, including electronics and signal processing. Today, the digital computer and special-purpose hardware are widely used for analyzing the frequency components of signals, and the frequency performance of systems. Discrete Fourier Transform (DFT) is an important method in the analysis of digital signals and systems, and is given by

$$F(r\omega_0) = \sum_{k=0}^{N_0-1} Tf(kT)e^{-jr\frac{2\pi}{N_0}k} \qquad (2.1)$$

where N_0 is the number of samples of the signal, T is the sampling interval, and $F(r\omega_0)$ and $f(kT)$ are periodic sequences with period N_0. In order to improve its computational speed, DFT is widely implemented using the so-called Fast Fourier Transform (FFT) algorithm. It factorizes the DFT and removes the redundant computations, thereby improving the computational speed in an exponential manner (de Silva, 2007b). Figure 2.5 shows the results of the sound and vibration frequency spectra as computed by the FFT approach.

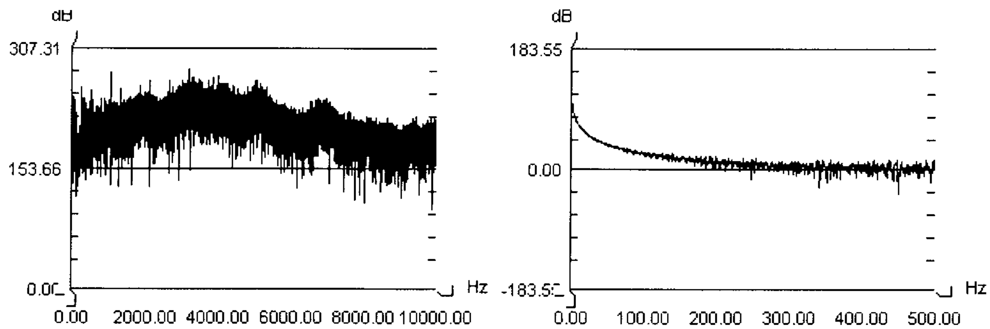


Figure 2.5: FFT Spectra of the sound and vibration signals.

2.4 Machine vision

Machine vision (MV) is often considered the application of computer vision to industrial and manufacturing systems. Whereas computer vision is mainly focused on machine-based image processing, machine vision most often requires digital input/output devices and computer networks to control other manufacturing equipment such as robotic arms. Machine Vision is a subfield of engineering that encompasses computer science, image processing, mechanical engineering, and industrial automation. A common application of Machine Vision is the inspection of manufactured goods such as semiconductor chips, automobiles, food and pharmaceuticals. Just as human inspectors working on assembly lines visually inspect parts to judge the quality of workmanship, machine vision systems use digital camerasand image processing software to perform similar inspections.

Machine vision systems are programmed to perform narrowly defined tasks such as counting the objects on a conveyor, reading serial numbers, and searching for surface defects. Manufacturers prefer machine vision systems for visual inspection that requires high-speed, high-magnification, 24-hour operation, and/or repeatability of measurements. Frequently these tasks extend the roles traditionally occupied by humans whose degree of failure is classically high through distraction, tiredness, illness and circumstance. However, humans may display finer perception over the short period and greater flexibility in classification and adaptation to new defects and quality assurance policies. In the present research, a feature based vision tracking approach known as the Scale Invariant Feature Transform (SIFT) (Low, 2004) is applied to the vision data in order to track the object of interest (fish) in a robust manner.

The vision based approach of object recognition and tracking has been widely used in industrial applications, particularly in the field of fault detection and inspection. Most of existing commercial systems primarily depend on the approaches of correlation-based template matching, which have been proved to be effective in a variety of applications. However, the scheme becomes infeasible when the object pose, scale and illumination are allowed to vary, especially when dealing with partial visibility of the tracking object. As a result, an alternative approach which is based on object features has been developed. In particular, Lowe proposed an object recognition and tracking algorithm called the Scale Invariant Feature Transform (SIFT), which uses a class of local image features (Low, 2004). In his algorithm, features are detected and image keys are created that allow for local gradients in multiple orientation planes and at multiple scales. The keys are used as input to a nearest-neighbor indexing method that identifies candidates for object matching. Verification of each match is achieved by finding a low-residual least-squares solution. In the present work, this tracking approach is applied to track a fish as it moves along the conveyer.

2.5 Generation of Feature Vectors

For vibration and sound signals, software which is introduced in the previous section is developed to acquire data in the time domain and transform it into the frequency domain. Then, each frequency spectrum is divided into four sub-regions as shown in Figure 2.6. For each region, a summation operation is carried out to calculate the sum of the amplitudes at each discrete frequency, using

$$V_i = \sum_{j=(N_0\times(i-1))/4}^{(N_0\times i)/4} M_j \quad for\ i = 1\ to\ 4 \tag{2.2}$$

Finally, a feature vector $V = (V_1, V_2, V_3, V_4)^T$ is generated, which represents the signature of the acquired signal. The feature vector is further normalized as

$$V_i = \frac{V_i}{\sqrt{V_1^2 + V_2^2 + V_3^2 + V_4^2}} \quad for\ i = 1\ to\ 4 \tag{2.3}$$

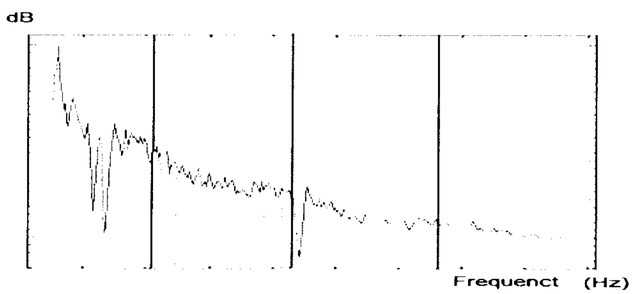


Figure 2.5: Feature vector of vibration and sound signals.

For the vision signal, a two dimensional (x-y) coordinate frame, which is shown in Figure 2.7, is attached on the plane of the conveyer table. At each time instant, the SIFT-based vision tracking software detects and tracks the positions and orientations of the fish. A vision vector is then generated for the fish, which contains the three elements: x, y, and orientation angle θ. This vision vector combined with the feature vectors of the vibration and sound signals will be provided as inputs to a neuro-fuzzy diagnosis system, as presented in the next chapter.

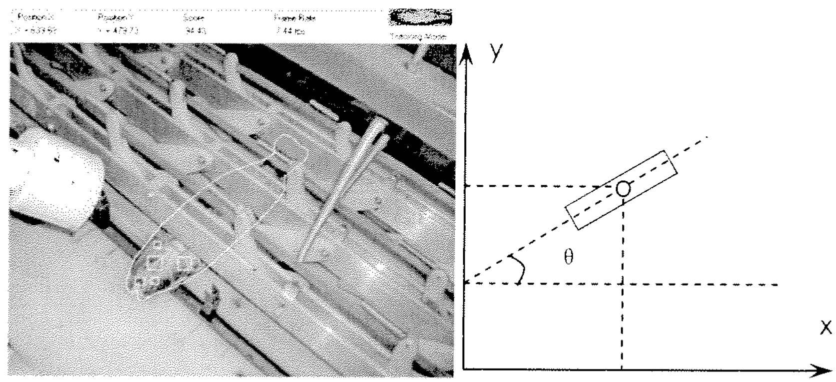


Figure 2.7: Fish tracking and generation of the feature vector.

2.6 Summary

The overall architecture of the developed fault diagnosis and remote monitoring system was presented in this chapter. First, the system hardware and software were introduced. The detailed specification of the hardware was listed and the programming environment used to develop the software in this project was described. Next, the data acquisition subsystem for vibration, sound and vision signals was presented. Utilizing the FFT algorithm, the time domain vibration and sound singles were transformed into the frequency domain. Furthermore, a SIFT method was applied for tracking of fish. Finally, feature vectors of vibration, sound and vision signals were generated for the neuro-fuzzy based fault diagnosis system, which will be introduced in the next chapter.

Chapter 3

Fault Diagnosis System with Sensor Fusion

The main objective of the present research is to develop a fault diagnosis system for an industrial fish cutting machine (the Iron Butcher) using multi-sensor data fusion. As reviewed in Chapter 1, neural networks and fuzzy logic are two successful methods in the applications of fault diagnosis systems. Therefore, as a main contribution of this thesis, a neuro-fuzzy based fault diagnosis method is developed, as presented in this chapter. In particular, this approach is applied to fuse vibration, sound and vision signals and make predictions of real time states of a fish cutting machine during operation.

In Section 3.1, the detailed infrastructure of the neuro-fuzzy inference network is introduced, which contains a fuzzy layer and a neural network layer. Then, in sections 3.2 and 3.3, the methodologies of these two layers are introduced. By utilizing the generated feature vectors as discussed in Chapter 2, training samples are obtained for offline training of the diagnosis network. The training method and strategy are discussed in section 3.4. After training, the weights of the network are updated and the neuro-fuzzy fault diagnosis network is implemented in the physical fish-cutting machine for online real time fault diagnosis. Finally, in section 3.5, the software technologies of the neuro-fuzzy system are demonstrated.

3.1 Neuro-fuzzy Fault Diagnosis

The Neuro-fuzzy approach (Tsoukalas, 1997 and Carray, 2004), which belongs to the field of artificial intelligence, hybridizes the two soft computing technologies of artificial neural networks and fuzzy logic. Fuzzy logic is derived from fuzzy set theory to deal with approximate and qualitative reasoning rather than to make precise deductions using the classical predicate logic (Karray and de Silva, 2004). It can be thought of as the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem (Klir, 1997). A neural network (NN) is an interconnected group of artificial neurons, which represents a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an an Artificial Neural Network (ANN) is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms neural networks are tools of non-linear statistical data modeling or decision making. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

Neural network and fuzzy systems represent two distinct methodologies that deal with uncertainty which arises with the increase of system complexity. Each of these two approaches has its own advantages and disadvantages. Neural network can represent complex nonlinear relationships, and they are very good at classification of phenomena into pre-selected categories. On the other hand, the precision of the outputs is sometimes limited and the time required for proper training of a neural network can be long. Fuzzy logic systems address the qualitative or human-perception nature of the input and output variables directly by defining them with fuzzy numbers that can be expressed in a linguistic manner. Hence, fuzzy logic systems are easier to formulate and modify, and thereby more tractable. Therefore, although neural networks and fuzzy logic are quite different, their unique capabilities can be combined in a synergistic way, therby gaining particular advantages in the present project.

Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Neuro-fuzzy hybridization is widely termed as Fuzzy Neural Network (FNN) or Neuro-Fuzzy System (NFS) in the literature. Neuro-fuzzy system (the more popular term is used henceforth) incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of neuro-fuzzy systems is that they are universal approximators with the ability to solicit interpretable IF-THEN rules.

As discussed before, the main purpose of the neuro-fuzzy network is to approximate a nonlinear mapping between sensor signals and potential machine faults. In this thesis, a neuro-fuzzy network, which approximately maps input sensor signals to potential faults of the machine, is developed to diagnose and predict the current and potential faults. The overall architecture of this fault diagnosis module is shown in the Figure 3.1.

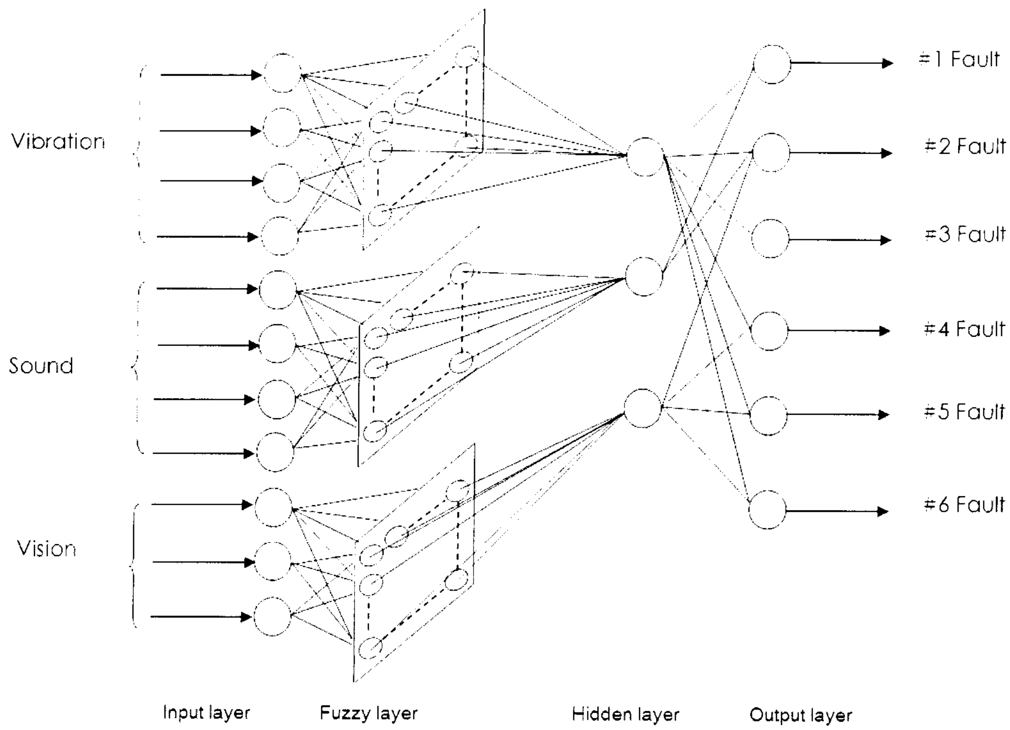


Figure 3.1: The overall architecture of the neuro-fuzzy network for fault diagnosis.

In Figure 3.1, the input layer will accept three feature vectors which are the vibration vector (x_1, x_2, x_3, x_4), the sound vector (x_5, x_6, x_7, x_8), and the vision vector (x_9, x_{10}, x_{11}) signals, as defined in Chapter 2. First, the three vectors are sent to the fuzzy layer for inference. It uses fuzzy decision making to match the input vectors to a known system status. The outputs of the fuzzy layer are then provided as inputs into the hidden layer of the neural network for identifying potential faults. This four-layer neuro-fuzzy network has a three-node hidden layer and a six-node output layer. Each output of the network represents one of six typical faults of the fish cutting machine.

3.2 Fuzzy Layer

As seen from Figure 3.1, the neuron-fuzzy network contains two main parts: the fuzzy layer and the neural network layer. In order to elaborate the function of the fuzzy layer, assume a two-input (i.e. x_1 and x_2) and one-output (i.e. y) situation which is shown in Figure 3.2. Each input variable is represented by a Gaussian type membership function as shown in Figure 3.3, which includes five values labeled as Negative Large (NL), Negative Small (NS), Zero (ZO), Positive Small (PS), and Positive Large (PL). Each function has two parameters: c, the centroid of the Gaussian function; and d, the width of the Gaussian function. After characterizing the Gaussian fuzzy membership function with the two parameters, the inference formula can be written as

$$y = \max(\exp(-\left\| \frac{C^i - X}{D^i} \right\|^2)) \; for \; i = 1 \; to \; T \qquad (3.1)$$

where T is the total number of fuzzy rules, which is 25 according to Figure 3.3, $X = (x_1, x_2)^T$ is the input vector, $C^i = (c_1^i, c_2^i)^T$ and $D^i = (d_1^i, d_2^i)$ denote the center and the width vectors of the two input memberships, and $\|\bullet\|$ denotes the norm operator which gives the Euclidean distance. After the normalizing operation, the final output of the fuzzy inference will range from 0 to 1.

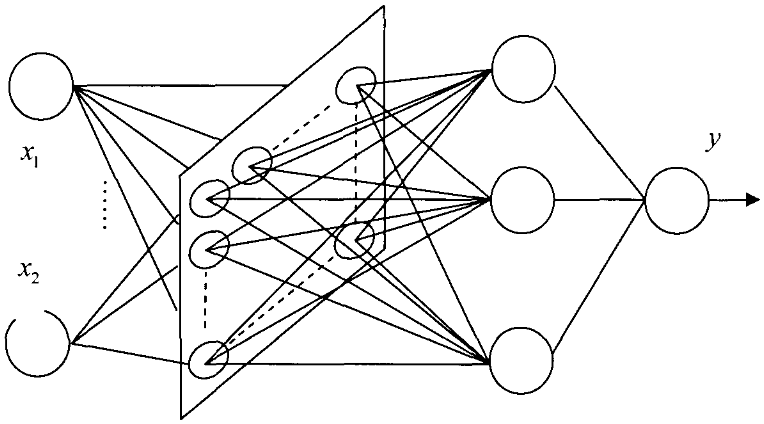


Figure 3.2: Simplified two dimensional neuro-fuzzy architecture.

In order to deal with the feature vectors of vibration, sound and vision signals simultaneously, three individual fuzzy inference modules are applied in this thesis, which constitute the fuzzy inference layer. For each element in the vector, a Gaussian type membership function including NL, NS, ZO, PS and PL is applied with two parameters c and d, as shown in Figure 3.4. Therefore, the dimensions of the fuzzy layers for vibration, sound and vision signals are 4, 4 and 3, respectively. By computing Equation 3.1, the fuzzy layer outputs are generated, which range from 0 to 1. They form the inputs to the neural network layer in Figure 3.1. It is commonly known that a fuzzy logic inference involves three steps: fuzzification, inferencing and defuzzification. However, the benefit of the present approach is to simplify the fuzzy inference by eliminating the procedures of fuzzification and defuzzification.

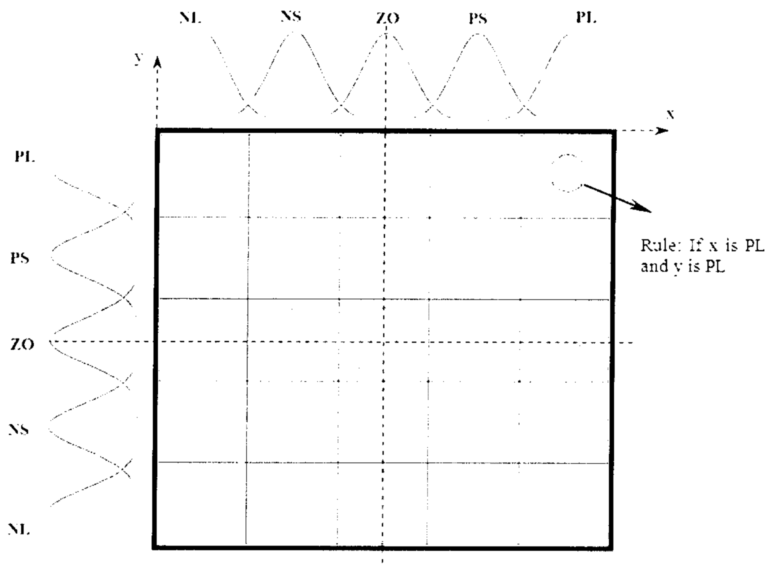


Figure 3.3: Representation of the fuzzy layer.

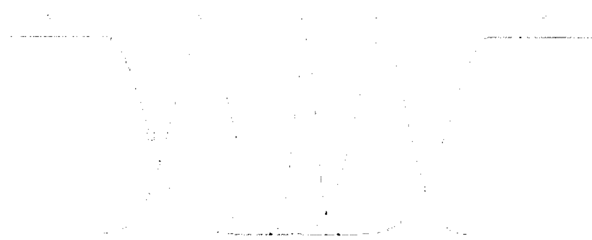


Figure 3.4: Membership functions of the fuzzy quantities.

3.3 Neural Network Layer

The second component of the fault diagnosis network is the neural network layer whose architecture has been presented in Figure 3.1. Now, training of the neural network is introduced and the training algorithm is explained, giving the overall strategy and the scheme for implementing the backpropagation (BP) method (Alpaydin, 2004) in the neural network, the training method, network structure, and convergence conditions.

Some general problems related to backpropagation are discussed in this section. The performance of the overall system will be addressed as well.

The training error of the neural network is closely related to the number of hidden nodes and epochs in the network. This relationship is shown in Figure 3.5 and Figure 3.6.

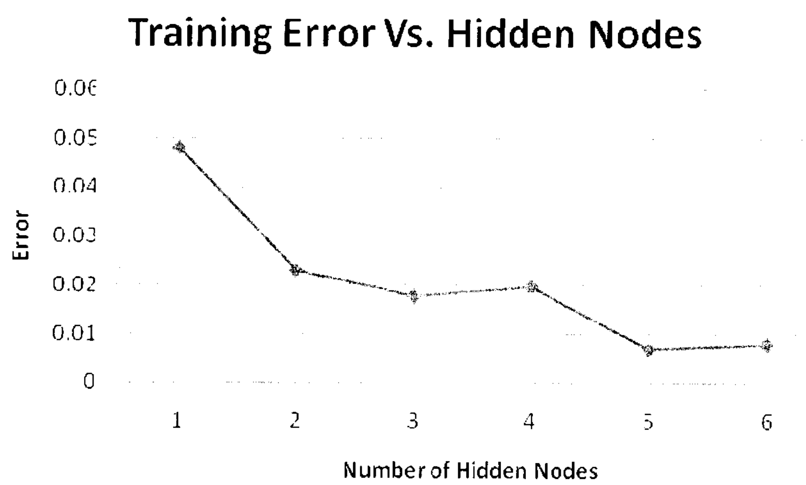


Figure 3.5: Training error versus The number of hidden nodes.

As shown in Figure 3.1, the neural network contains three layers which are the input layer, the hidden layer and the output layer. Also, there are 3 input nodes (fuzzy layer outputs) and 6 output nodes in the neural network. The number of the hidden nodes is chosen to be 3 because it gives a relatively small error and fast convergence speed according to Figure 3.5 Figure 3.6. Another advantage of this type of structure is that it is easy to code. Furthermore, too many hidden nodes in the network can lead to problems. In particular, an overly complex model memorizes the noise in the training set and does not generalize to the validation set.

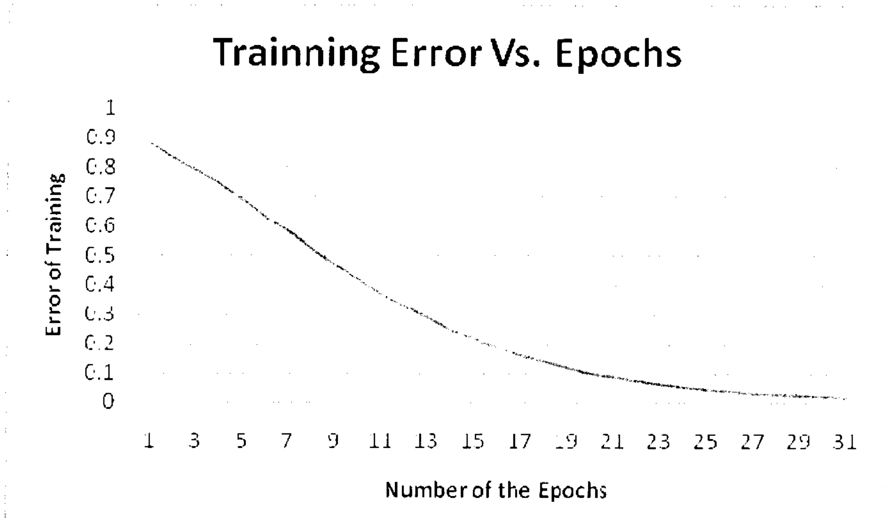


Figure 3.6: Training error versus epochs.

The backpropagation training algorithm is used in this project to adjust weights in the neural network. The detailed procedure of the algorithm is shown in Figure 3.7. In the algorithm, $sigmoid(w_h^T x^T)$ is given by

$$y = sigmoid(w^T x) = \frac{1}{1+\exp[-W^T x]} \qquad (3.2)$$

This training algorithm can be applied in both on-line and off-line training. In this thesis, off-line training is carried out before the neural network is implemented in the diagnosis system of the fish processing machine. The criterion used to stop training in the present application is: when the mean squared error is less than 2%, the training process automatically stops. It is known that a phenomenon called over-fitting will appear when training is continued for too long. In particular, as more training epochs are made, the error on the training set decreases, but the error on the validation set starts to increase beyond a certain point. Thus, training should be stopped before it is too late to alleviate the problem of over-fitting. There are two factors which will indicate that the algorithm has learned correct mapping. The first indicator is whether the error is generally decreasing with the training, and the other is whether the error stops changing (i.e., converges). Under optimized conditions, as given by Figure 3.6, the neural network will properly learn the intended strategy. It takes about 16 epochs to converge, which corresponds to a fast convergence speed.

Initialize all v_{ih} and w_{hj} to rand(-0.01, 0.01)

Repeat
For all $(x^t, r^t) \in X$ in random order
For h=1, ..., H
$z_h \leftarrow sigmoid(w_h^T x^T)$
For i=1, ..., K
$y_i = v_i^T z$
For i=1, ..., K
$v_i = \eta(\sum_i (r_i^t - y_i^t) v_{ih} z_h (1 - z_h) x^t)$
For i=1, ...K
$v_i \leftarrow v_i + \Delta v_i$
For h=i, ..., H
$w_i \leftarrow w_i + \Delta w_i$
Until convergence

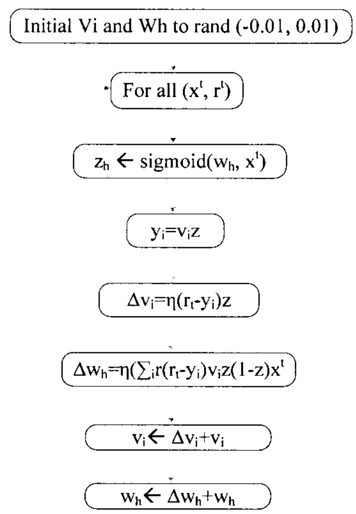


Figure 3.7: Flow chart of the backpropagation algorithm for training.

3.4 Neuro-fuzzy Sensor Fusion

Sensor fusion, which is also known as multi-sensor data fusion, is generally defined as the use of techniques that combine data from multiple sources and gather that information in order to achieve inferences, which will be more efficient than if they are achieved by means of a single source. It is shown that sensory data fusion from disparate sources can possibly provide a better solution than when these sources are used individually.

In this thesis, vibration, sound and vision are the sensory data sources and the neuro-fuzzy network is the means of fusing them for fault diagnosis. In the present system, offline training is applied by acquiring specimens under six typical working conditions given by:

- Blocked (jammed) fish
- Failure of the hydraulic cylinder system
- Failure of the conveyor system
- Failure of the hydraulic pump
- Failure of the hydraulic servo valves
- Failure of the pneumatic controlled cutter
- Normal operation

For each working condition of the fish-cutting machine, the vibration, sound and vision signals are acquired. After acquiring signals from the sensors, the software extracts the three feature vectors and sends them to the neural network for fault diagnosis. Through checking the real preset fault, one output of the neural network is set to "1" and other outputs of the network are set to "0." A training sample is generated in this manner. For each potential fault and the normal (non-faulty) operating condition, hundreds of samples are generated in this manner. When the mean squared error becomes less than 2% for each condition, the training process is stopped. After training, the neuro-fuzzy fault diagnosis network is capable of properly identifying the faults in the fish cutting machine.

3.5 Neuro-fuuzy Software Development

In this section, some important source codes of the neuro-fuzzy software are introduced, which include the fuzzy inference and backpropagation neural network. The variables which are defined in this software are given below:

```
Option Explicit
Private mW1() As Double              'weights of the hidden layer
Private mW2() As Double              'weights of the output layer
Private mB1() As Double              'bias of the hidden layer
Private mB2() As Double              'bias of the output layer
Private mErr() As Double             'mean squared error
Private mMinMax() As Double          'up limit of input
Private mS1 As Long                  'number of the node of the hidden layer
Private mS2 As Long                  'number of the node of the output layer
Private mR As Long                   'number of the node of the input layer
Private mGoal As Double              'goal rate of convergence
Private mLr As Double                'learning rate
Private mMaxEpochs As Long           'max epochs
Private mIteration As Long           'actual epochs

Private HiddenOutput() As Double     'outputs of the hidden layer
Private OutOutput() As Double        'outputs of the input layer
Private HiddenErr() As Double        'error of the each node of the hidden
layer
Private OutPutErr() As Double        'error of the each node of the output
layer
Private Pdealing() As Double         'current inputs
Private Tdealing() As Double         'current outputs
Private OldW1() As Double            'old weight vector of the hidden layer
Private OldW2() As Double            'old weight vector of the output layer
Private OldB1() As Double            'weight error vector of the hidden layer
Private OldB2() As Double            'weight error vector of the output layer
Private Ts As Long                   'the number of the input vector
Private Initialized As Boolean       'is it initialized?
```

The source code of the backpropagation training is given below:

```
Public Sub Train(P() As Double, T() As Double)
    Dim i As Long, j As Long, Index As Long
    Dim NmP() As Double
```

```
        mR = UBound(P, 1)                     'number of elements in each input
vector
        mS2 = UBound(T, 1)                    'number of outputs
        Ts = UBound(P, 2)                     'number of elements in the input vector
        NmP = CopyArray(P)
        IniParameters NmP                     'initialize the parameters and arrays
        mIteration = 0
        For i = 1 To mMaxEpochs
            mIteration = mIteration + 1
            Index = Input(i)
            For j = 1 To mR
                Pdealing(j) = NmP(j, Index) 'process input vectors
            Next
            For j = 1 To mS2
                Tdealing(j) = T(j, Index)     'process output vectors
            Next
            HiddenLayer                       'calculate the output of each hidden
node
            OutputLayer                       'calculate the output of each output
node
            OutError                          'calculate the error of each output node
            HiddenError                       'calculate the error of each hidden node
            Update_W2B2                       'update weights between output nodes
                                            and 'hidden nodes
          Update_W1B1                         'update weights between input nodes
                                              and 'hidden nodes
            If iteration Mod 1000 = 0 Then RaiseEvent Update(mIteration)
            If mErr(mIteration) < mGoal Then Exit Sub    'stop learning if it reaches
                                                         the 'criteria
        Next
    End Sub
```

Some sub-functions in the training function, including the computations of the hidden layer and output layer, error calculation, and weight updating are given blow:

```
********************Operation of the hidden layer********************
    Private Sub HiddenLayer()
        Dim i As Long, j As Long
        Dim Sum As Double
        For i = 1 To mS1
            Sum = 0
            For j = 1 To mR
                Sum = Sum + mW1(i, j) * Pdealing(j)
```

```
            Next
            HiddenOutput(i) = 1 / (1 + Exp(-(Sum + mB1(i))))
        Next
    End Sub

********************Operation of the output layer********************
    Private Sub OutputLayer()
        Dim i As Long, j As Long
        Dim Sum As Double
        For i = 1 To mS2
            Sum = 0
            For j = 1 To mS1
                Sum = Sum + mW2(i, j) * HiddenOutput(j)
            Next
            OutOutput(i) = Sum + mB2(i)
        Next
    End Sub

********************Error Calculation ********************
    Private Sub OutError()
        Dim i As Long, j As Long, Mse As Double
        For i = 1 To mS2
            OutPutErr(i) = Tdealing(i) - OutOutput(i)
            Mse = Mse + OutPutErr(i) * OutPutErr(i)
        Next
        mErr(mIteration) = Sqr(Mse / mS2)
    End Sub

********************Weights updating ********************
    Private Sub Update_W2B2()
        Dim i As Long, j As Long
        Dim Temp As Double
        For i = 1 To mS2
            For j = 1 To mS1
                Temp = mLr * OutPutErr(i) * HiddenOutput(j) + mGama * OldW2(i,
j)
                mW2(i, j) = mW2(i, j) + Temp
                OldW2(i, j) = Temp
            Next
            Temp = mLr * OutPutErr(i) + mGama * OldB2(i)
            mB2(i) = mB2(i) + Temp
            OldB2(i) = Temp
        Next
    End Sub
```

3.6 Summary

In this chapter, a customized neuro-fuzzy architecture for fault diagnosis, including the input layer, the fuzzy layer, the neural network layer and the output layer, was presented. In order to implement this system in a practical industrial application--a fish cutting machine--the backpropagation training method was employed. The detailed training algorithm was introduced. The software development for the neuro-fuzzy system was also presented, and some important source codes were given. In the next chapter, experiments will be carried out to validate the neuro-fuzzy approach developed in this chapter.

Chapter 4

Web-based Remote Monitoring

Web-based remote monitoring is an evolution of the traditional monitoring systems. It is easily configurable, flexible, powerful and manageable because it is based on modern communication network technologies. A remote monitoring system includes a plant to be monitored, data transmission mechanism and some remote clients. The system may be used through a computer connected to the communication network at any location. In this manner, a remote monitoring system enables engineers and researchers to proactively manage various equipment and systems in remote locations over the web (Internet), providing a cost effective and more productive work environment. In this chapter, the basic framework of a web-based remote monitoring architecture is presented and important design issues are addressed.

Section 4.1 gives an overview of the developed web-based remote monitoring architecture. The interconnection hardware is discussed in Section 4.2. Section 4.3 focuses on the architecture of the software and some issues and challenges in the aspect of software design. Sections 4.4, 4.5 and 4.6 introduce the web, video and camera control servers and their detailed configurations, as used in the present thesis. In Section 4.7, the Graphic User Interface (GUI) of the web-based remote monitoring system is demonstrated. This is followed by a practical application, presented in Section 4.8. Section 4.9 gives a summary of the remote monitoring system developed and implemented in this thesis.

4.1 System Overview

The specific application of the remote monitoring system in this thesis focuses on developing a universal network architecture which includes both hardware and software. The overall system architecture is shown in Figure 4.1.

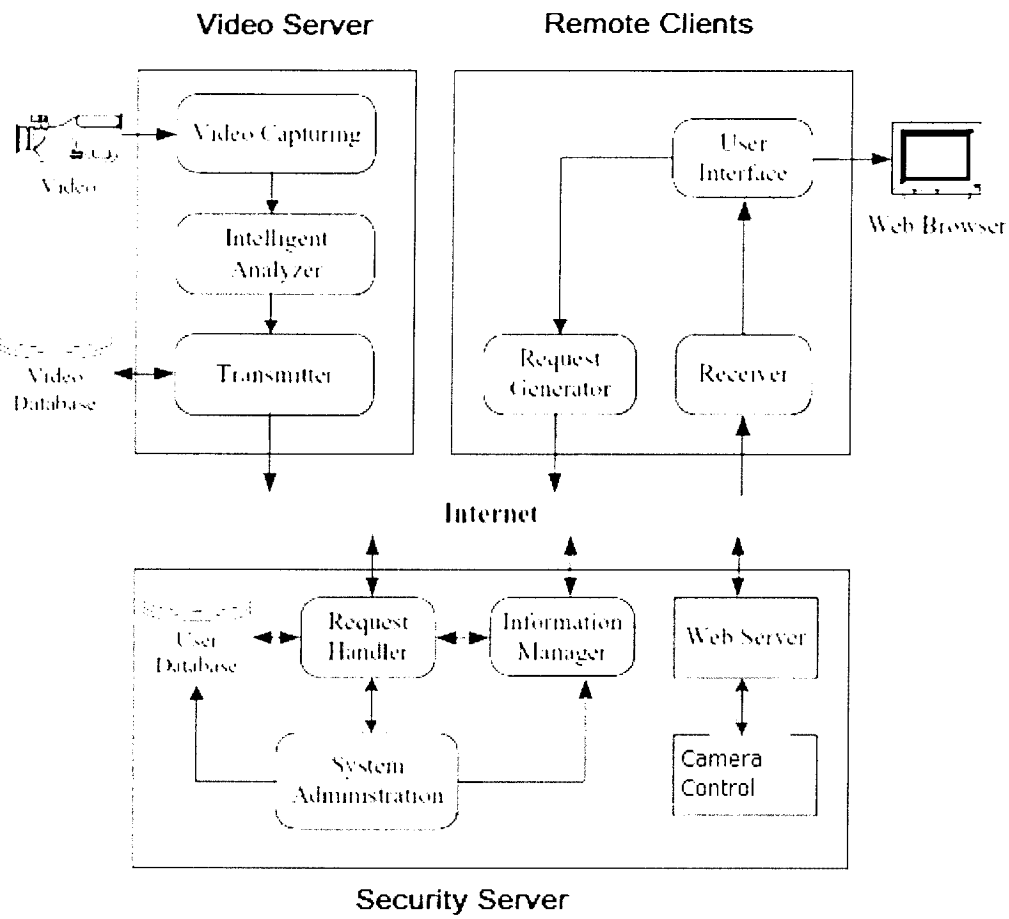


Figure 4.1: The System architecture of the remote monitoring system.

There are three crucial parts in this architecture. They are the three video servers, the security server and the remote clients. These three servers are installed on a Pentium IV personal computer, providing the services of video stream transmission, camera control, data security control and execution of several pre-programmed tests. This design allows authorized researchers and engineers to access the laboratory website and acquire real time machine health conditions at any of locations via Internet connection. The detailed functions of these three parts will be described in sections 4.4, 4.5 and 4.6.

4.2 System Hardware

Objectives of this thesis include development of an architecture for remote monitoring of machinery for fault diagnosis and to implement the system using existing network technologies which only need a Fast Ethernet and a low cost Network Interface Card (NIC). The interconnection structure of the hardware is shown in Figure 4.2. A Panasonic KXDP702 color camera is placed at a strategic location to capture live video of the operating machine and its environment. It has three degrees of freedom with built-in pan, tilt and 21x zoom, which are controlled by a server computer using an RS232 port. For capturing and encoding the video signals from the camera, a frame grabber is installed in this computer, capturing the video signals with a resolution of 640×480 pixels and a speed of 30 fps (frames per second). In addition, multiple cameras can work together since this frame grabber simultaneously supports 16 channels.

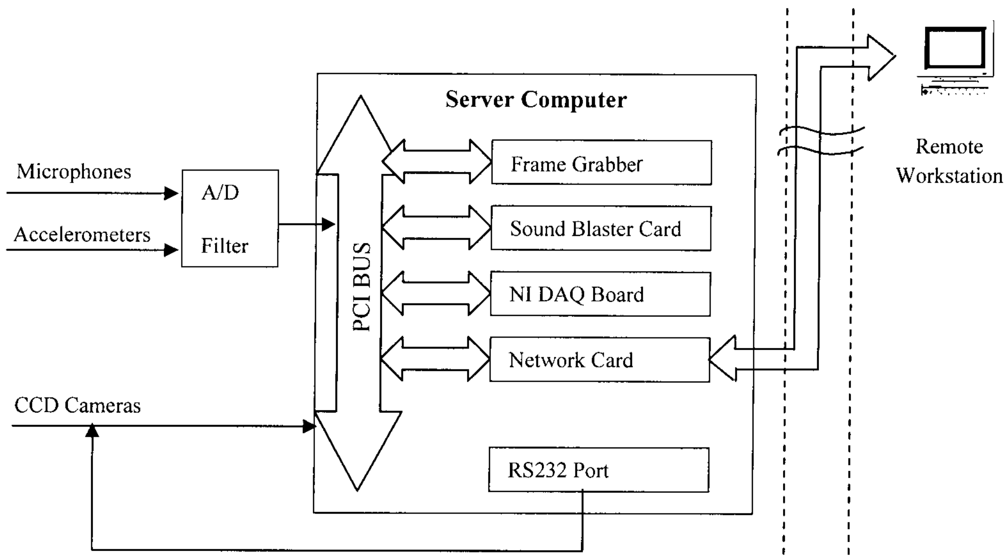


Figure 4.2: The hardware structure of the remote monitoring system.

4.3 Software Architecture

Figure 4.3 shows the system components and information flow in the software system. The server uses Microsoft Windows 2000 operating system. It is a browser-server (BS) application architecture including a web-server, a security server and a camera control server.

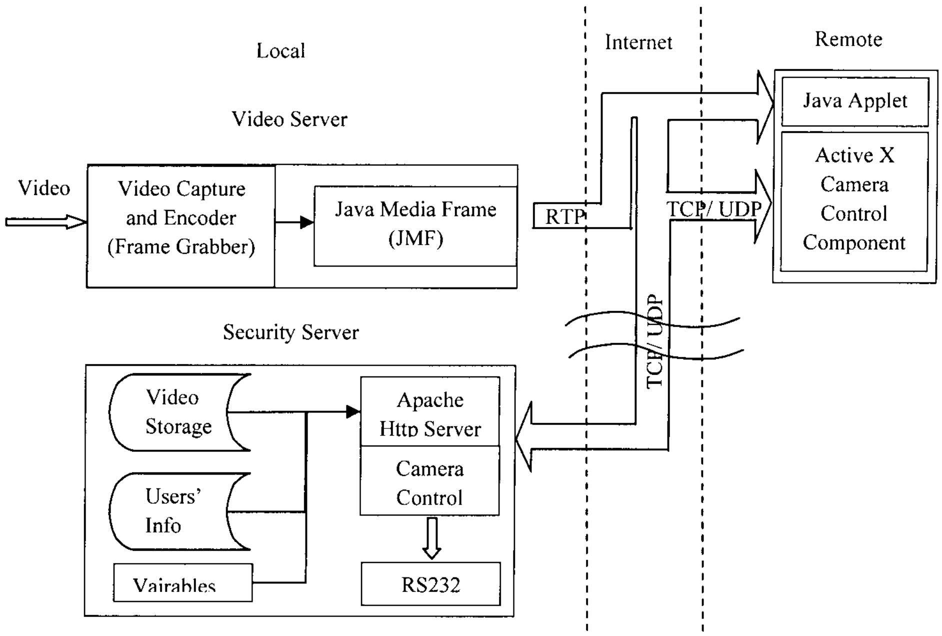
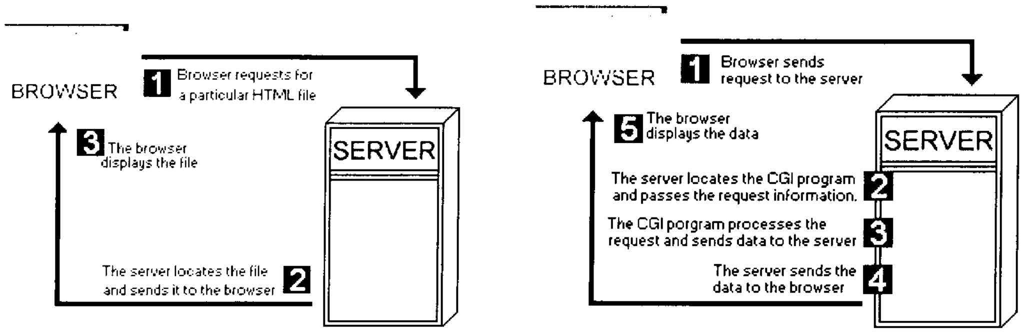


Figure 4.3: System components and information flow.

4.3.1 Browser-Server Architecture

The browser-server (BS) architecture is a special kind of client- server (CS) architecture. Here, the web employs a connection-less protocol, which means that after every client-server interaction the connection between the client and the server is lost. In particular, the client (browser) requests an HTML file stored in the remote machine (server) through the server software. The server locates this file and passes it to the client. The client then displays this file on the machine. In this case, the HTML page is static, as shown in Figure 4.4(a). In this thesis, an advanced CGI browser-server application is implemented instead of the traditional BS architecture, which is shown in Figure 4.4(b). In this architecture, the server has to perform more work to generate dynamic HTML pages for different clients and requests. Therefore, it is more flexible and has a higher level of security. Remote users access the web site of the monitoring system by using their authorized user names and passwords. Different users can monitor different information of the machine, as controlled by CGI scripting.



(a) Traditional BS architecture (b) Advanced BS architecture--CGI Scripts

Figure 4.4: The browser-server architecture.

4.3.2 Web Server

In this thesis, the web-server runs the open-source Apache HTTP server which supports Perl/CGI (Common Gateway Interface) and Java™ Applets. The Apache is a web server notable for playing a key role in the initial growth of the World Wide Web. It supports a wide variety of operating systems including Microsoft Windows, Novell NetWare and Unix and Unix-like operating systems such as FreeBSD, Linux Solaris and Mac OS. Moreover, it is primarily used to serve both static content and dynamic Web Pages on the World Wide Web. Many web applications are designed using the environment and features that Apache provides. When compared with the Microsoft Internet Information Service (IIS) and .NET platform, it has significant advantages with features of free usage and open source code. Therefore, Apache is installed in the server computer in this project providing web service for remote clients.

Two network communication protocols are applied in this project: Transmission Control Protocol (TCP), and User Datagram Protocol (UDP), which are protocols in the transport layer. TCP is a connection-oriented protocol and provides reliable point-to-point data transfer. Data delivery is assured using flow control, sequence numbering, retransmission, and a timer. In addition, TCP also provides congestion control by regulating the rate at which data is injected into the network. Therefore, it is a network transmission protocol which can guarantee the transmission of data packages form one point of the network to the other. UDP does not guarantee reliability or ordering in the way that TCP does. Datagrams may arrive out of order and appear duplicated, or go missing without notice. However, it has its own benefits compared to TCP. In particular, it has faster transmission speed, and it is more efficient. In this project, TCP is applied to authority and control information transmission, since it can ensure the reliability and accuracy in data transmission, while UDP is used for streaming media transmission which is time-sensitive.

4.3.3 Camera Control Server

The camera control server, which is also called the camera control console, handles requests from the remote clients for state changes of the local monitoring camera. The camera is connected with the server by a serial cable. Consequently, in order to manipulate the camera, including pan, tile, and zoom, the camera control console, which is developed by Visual Basic (VB) has capabilities to receive commands from remote clients and send control signals to the video camera through the RS232 communication link (serial port). The communication mechanism between the control console and the camera is shown in Figure 4.5. Commands are passed to the camera in a serial sequence. For each command, the camera returns a response to the control console. If the command from the console is illegal, the camera sends a "NAK" feedback to the console. Otherwise, it performs the action as requested by the command and sends an "ACK" feedback to the console followed by the current state of the camera. This communication procedure is shown in Figure 4.5, A summary of the camera motion control commands is given in Table 4.1 where each command is represented by hex data.

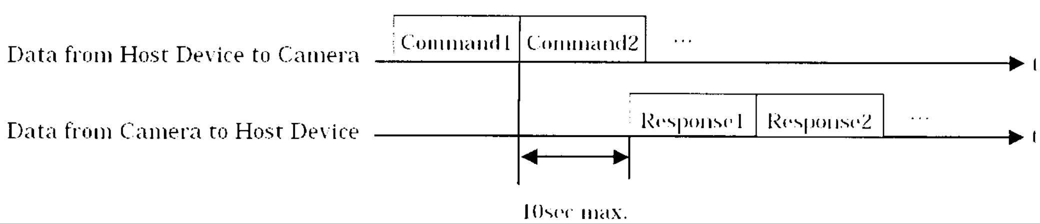


Figure 4.5: Communication between the camera control software and the camera.

Table 4.1: The predefined camera control commands.

Command	**Description**
0FH	Parameters Initialization

Table 4.1: The predefined camera control commands.

12H	Read Preset Position Information
20H	Pan (left): Move to Pan Direction
21H	Pan (right): Move to Pan Direction
22H	Tile (up): Move to Tilt Direction
23H	Tilt (down): Move to Tilt Direction
24H	Zoom Wide: Move Zooming lens to "Wide"
25H	Zoom Tele: Move zooming lens to "Tele"
26H	Focus Far: Move focusing lens to "Far"
27H	Focus Near: Move focusing lens to "Near"
38H	Home Position: Move to home position

4.3.4 Video Server

The video server is developed using the Java Media Frame (JMF) package which was originally developed by Sun Microsystem, Silicon Graphics, Intel and IBM. It enables audio, video and other time-based media to be added to Java applications and applets. Moreover, it supports capture, playback, stream and transcode of multiple media formats, and extends to cross-platform multimedia applications. A JMF based software is developed and implemented in this thesis to capture the video stream from the CCD camera and store it in a video database. Meanwhile, the video stream is distributed over the Internet. A remote user can view the video stream on a local Internet Browser where a JMF applet is embedded to receive the data streams from the video sever. Real-time Transport Protocol (RTP) is applied in this project, which is a standardized packet format for delivering audio and video over the Internet. A main advantage of this protocol is that it is possible to receive only one part of the transmission, commonly audio data, which decreases the total bandwidth. In addition, applications using RTP are less sensitive to packet loss.

4.3.5 Network Performance

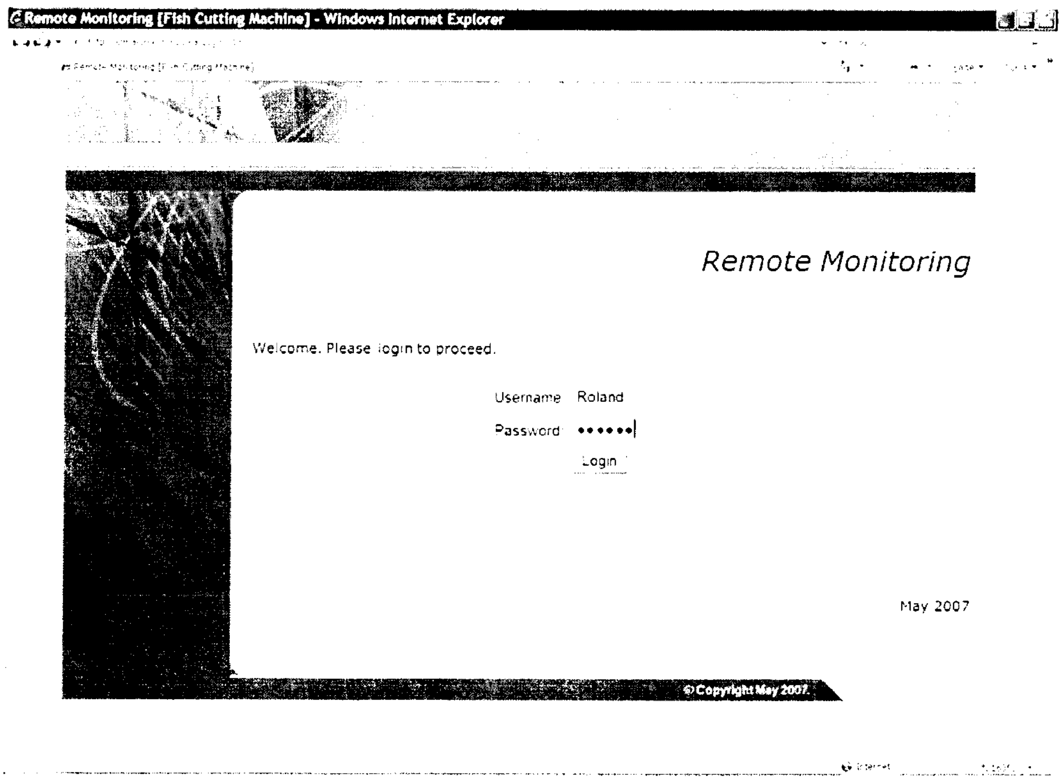
The quality of network transmission significantly relies on the performance of the network connection. Therefore, as a video stream transmission application, the network condition must be considered since streaming video can be unreliable under poor network conditions. When a video stream is prepared for transport from servers to clients, data is broken into component parts (often known as frames, packets or segments) which are transported in sequence. Several factors such as bandwidth, package loss, latency, and network traffic jam (conjestion) can affect the delivery of data. In some cases, the network may not work at all; in others, it may be slow or unusable. If video stream applications run over these networks, their performance will suffer. Therefore, it is better to know the network condition before implementing data transmission. In the present project, a network evaluator is developed to analyze the connecting performance of the network between the server and the clients by measuring its Round Trip Time (RTT). The RTT, which includes the node delays and the media transit time, is calculated through handshake by measuring the time between segment transmission and ACK receipt. It follows that the result of the RTT is significantly related to the performance of the network. In the present project, through the measurement of RTT, the performance of the network is divided into three levels: poor connection, good connection and ideal connection. If the network connection is poor, only data of vibration and sound signals and their frequency spectra are transported and displayed on the web page of remote clients. If the connection state is good, the video stream is transported at low quality (160×120). Under ideal conditions, the video stream is transported at high quality (640×480).

4.4 The Graphic User Interface

Figure 4.6 shows the Graphic User Interface (GUI) which is displayed on the remote clients when the web server is requested by the Internet Explorer. An authorized user name and password are required by the system in order to access the remote monitoring website in Figure 4.6(a). Once the security server accepts the input user name and password, the connecting performance will be evaluated, and the page will automatically jump to the remote monitoring web page, as shown in Figure 4.6(b). The remote monitoring page contains two parts: a camera control ActiveX component and a Java Applet.

ActiveX is a Microsoft technology which is used for developing reusable object oriented software components. In the present thesis, an ActiveX component is developed by Visual Basic to build a plug-in for Internet Explorer. The camera control ActiveX component is automatically downloaded from the server when the page is loaded for the first time. It will connect to the camera control console which runs in the local server. Button ressing actions will result in sending camera control commands from the client to the server through the TCP/IP communication protocol. This component can control the motion (tilt and pan), the focus and the backlight of the camera. There are 10 pre-set poses of the camera which can efficiently set the focus of the camera. The communication commands for the local camera have been introduced in Section 4.3.3.

The Java applet also contains two parts. One displays the video stream which is located on the top of the camera control ActiveX component. The other is in the right region of the web page displaying received sound and vibration signals and their spectra.



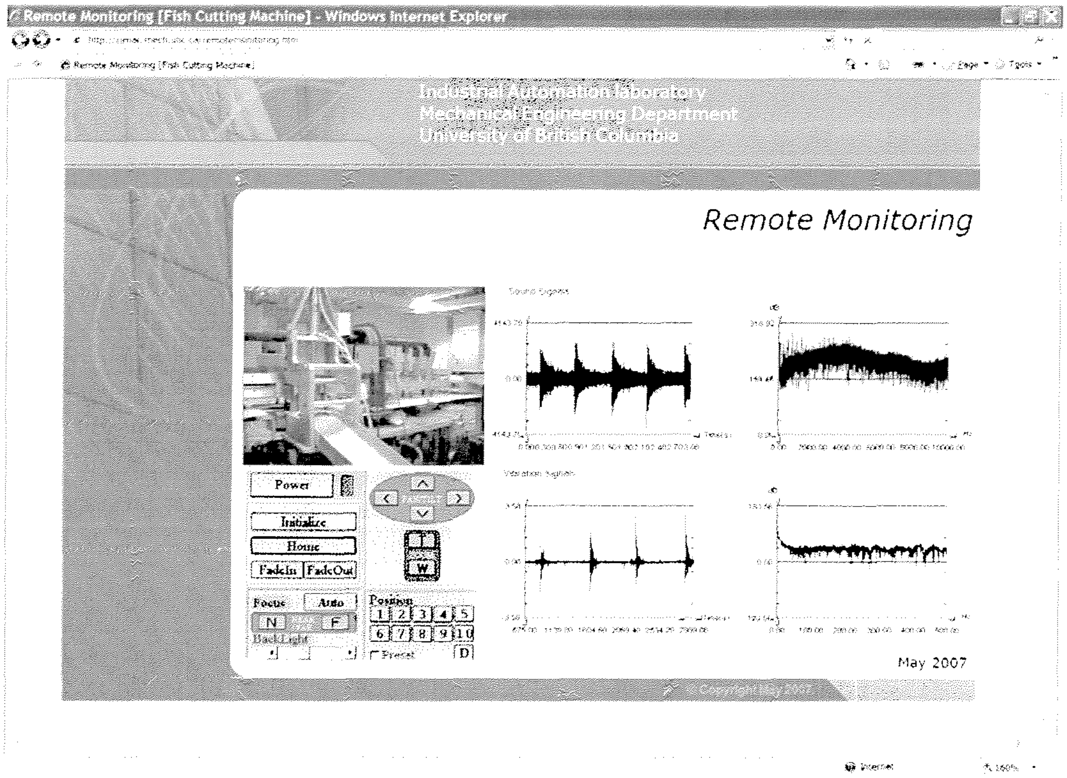


Figure 4.6: The graphic user interface of the remote monitoring system.

4.5 Summary

In this chapter, a practical architecture for the web-based remote monitoring and camera control system was developed and presented. The interconnection of hardware and software of the web server, the video server and the security server was established in both the physical connection level and the information flow level. The fish cutting machine described in Chapter 1 serves as the monitoring plant. The developed architecture is useful in industrial facilities, research laboratories, and academic environments for remote monitoring and system diagnosis.

Chapter 5

Performance Evaluation

This chapter evaluates the performance of the neuro-fuzzy fault diagnosis system, which has been proposed and developed in the present thesis. In Section 5.1, the test bed of the project--the industrial fish cutting machine--is described. Three crucial subsystems, the conveyor system, the pneumatic system and the hydraulic system are introduced, and the main components of these three subsystems are described. Section 5.2 focuses on the simulation of different faults of these three crucial subsystems, where the detailed simulated faults are described. In Section 5.3, the developed signal processing and neuro-fuzzy faulty diagnosis software, as presented in previous chapters, are implemented in the test system, results are obtained, and the system performance is discussed. A comparison is made between results from single-sensor diagnosis and multiple-sensor-fusion diagnosis, and the performance is discussed in Section 5.4. Finally, the highlights of the chapter are summarized in Section 5.5.

5.1 The Test Bed

There are three crucial subsystems in the industrial fish cutting machine: the conveyer system, the pneumatic system and the hydraulic system. These are described next.

5.1.1 Mechanical System (Conveyer)

Figure 5.1 shows the mechanical system which is responsible for conveying the fish from the feeding zone to the cutting zone. The conveying system is powered by an AC induction motor, where the rotary motion is transformed into a linear push-pull stroke of a three copper sliding bars through a mechanical linkage.

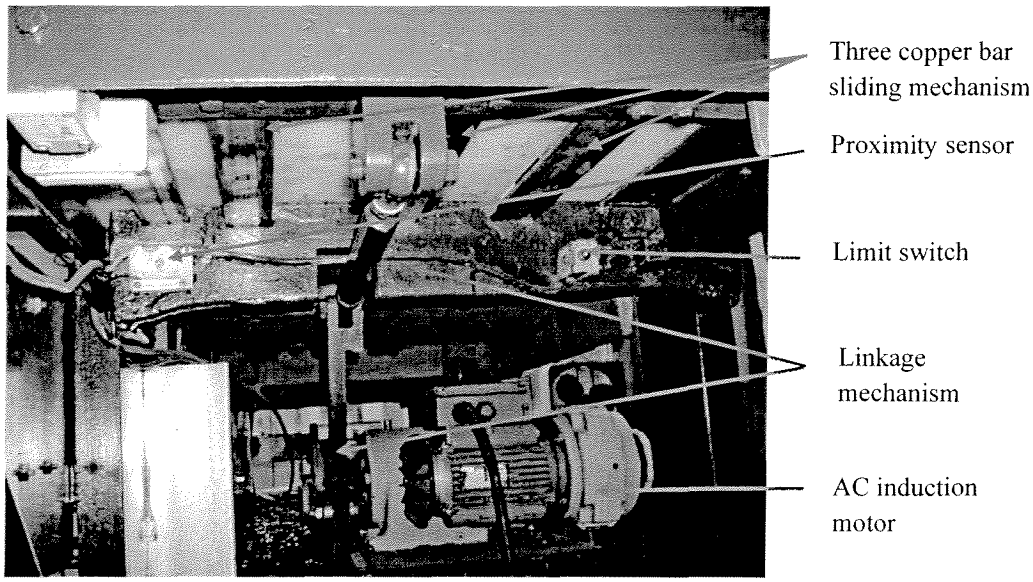


Figure 5.1: The fish conveying mechanism.

5.1.2 Pneumatic System

Figure 5.2 shows the pneumatic system of the fish cutting machine. There are two sub-systems: 1) Three groups of spring loaded holding system which are used to stabilize or hold the fish in a stationary pose while the matrix of retaining pins move backwards and forwards to transport the fish to the cutter. 2) The pneumatic cutter controlled by a pneumatic cylinder which operates the vertical movement of the cutter blade.

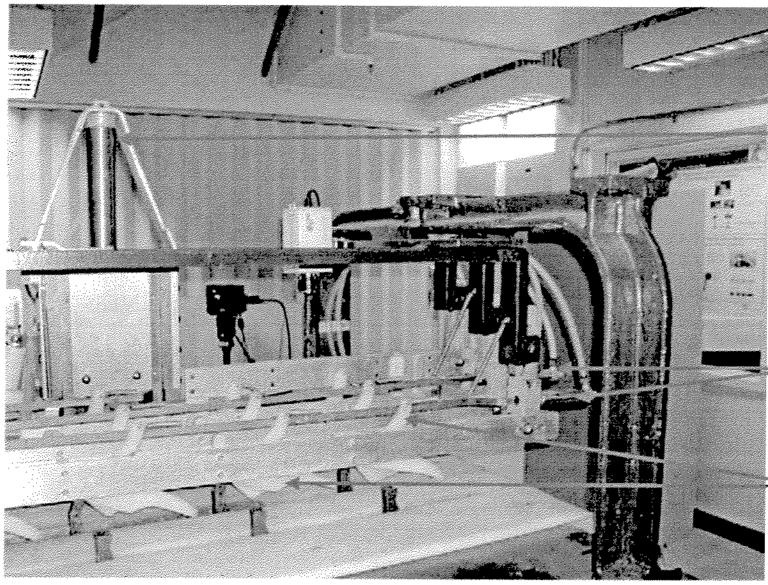


Figure 5.2: The pneumatic system.

5.1.3 Hydraulic System

The hydraulic system is a very important and complex system in the fish cutting machine. The main task of this system is to position the carriage of the cutter table to a specified location (as determined by the vision unit) for execution of the cutting action with least meat waste. This is in fact a two-axis hydraulic manipulator system.

The manipulator system, which consists of a Cartesian electro-hydraulic manipulator, can be divided into three sub-systems, as shown in Figure 5.3. The hydraulic power subsystem provides sufficient energy to the overall system which contains a placement pump, heat exchanger, an oil reservoir and other accessories (de Silva, 2005a). The servo valve subsystem contains two flow control electro-hydraulic servo-valves which controls the hydraulic fluid flow into the cylinder and hence the cylinder motion. It also acts as an interface between the power subsystem and the hydraulic cylinders (actuators) of the load subsystem. The load subsystem contains two double acting single rod power cylinders which serve as the actuators to position the cutter table along X and Y directions. The physical servo-valve subsystem and the load subsystem are shown in Figure 5.4.

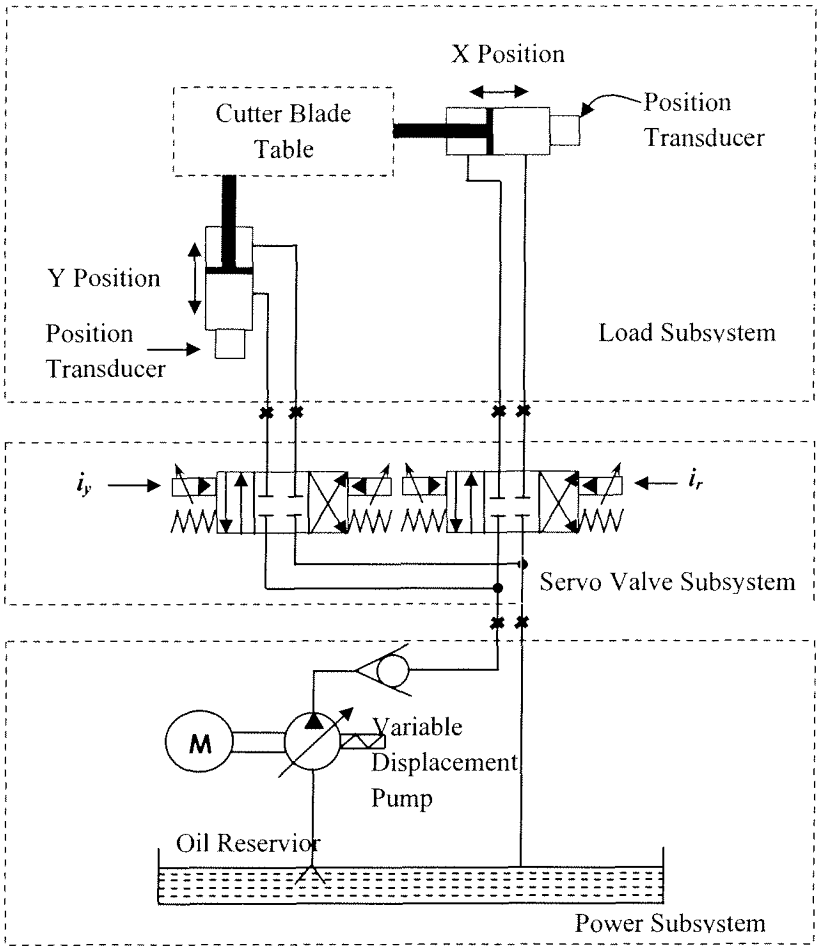


Figure 5.3: Schematic diagram of the electro-hydraulic manipulator system.

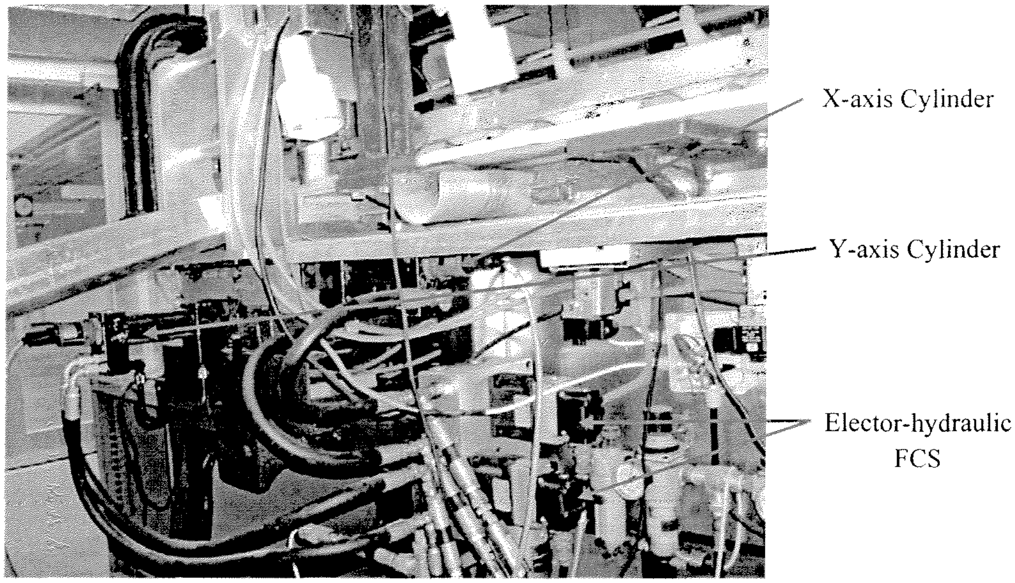


Figure 5.4: The servo-valve subsystem and the load subsystem.

5.1.4 Sensor Arrangement

The physical arrangement of the sensors of the test machine is shown in Figure 5.5. In this project, five types of sensors are used on the fish cutting machine to detect the potential faults. They are 4 accelerometers, 2 position sensors, 4 pressure sensors, 2 microphones and 2 CCD cameras. Figure 4 and Table 1 show the positions of these sensors, as mounted on the machine. Four accelerometers are mounted on the machine mainframe, the conveyer, the vertical cutter, and the cutting table, respectively. Between the accelerometers and the data acquisition board, there is a set of power amplifier. A JVC camera is mounted on the left side of the cutter, to sense the movement of the conveyor system. Another JVC CCD camera is mounted on the right side of the pneumatic cutter to track the position and orientation of the fish. Moreover, there is a global three degrees of freedom Panosonic CCD camera providing the video stream of working conditions of the machine for the purpose of remote monitoring.

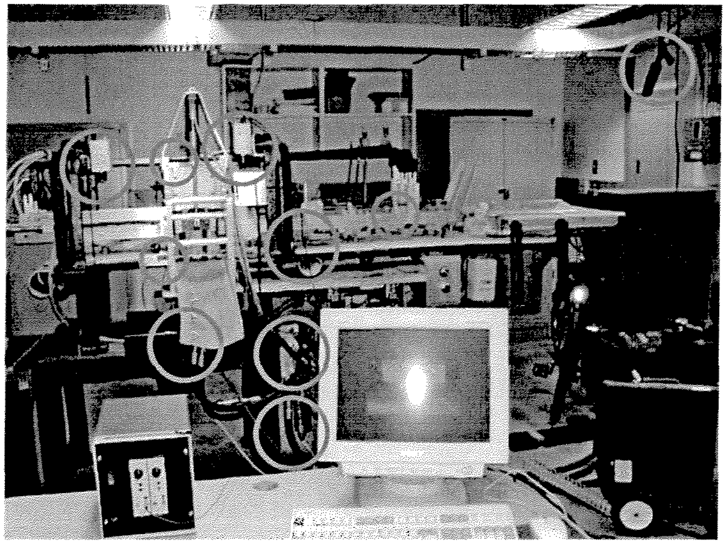


Figure 5.5: Arrangement of the physical sensors on the test machine.

5.2 Simulated Faults

As we know, each subsystem of the fish cutting machine consists of several important components such as valves, actuators, motors, and so on. Failure of any of these components can cause abnormal working conditions in the machine. In the following section, most potential faults will be identified, which are related to the three main subsystems. Symptoms of failures will be described, and the causes of failures will be examined as well.

State #1: The hydraulic system is running normally

State #2: Failure of the hydraulic pump

Symptoms:

- The cutting table is not able to move.
- There is no change in the outputs of the position sensors (both x and y cylinders).
- The output of the pressure sensor is 0 (Electro-hydraulic FCS).

- The operating noise of the pump is not present

Causes: The hydraulic pump is totally shut down.

State #3 Failure of electro-hydraulic FCS

Symptoms:

- Cutting table cannot move along one/ two directions.
- The reading of one/ two position sensors remains constant.
- The output of the pressure sensor is 0.

Causes: Failure of electro-hydraulic FCS

State #4 Failure of the *x*-axis or *y*-axis cylinders

- Cutting table cannot move along one/ two directions.
- The reading of one/ two position sensors remains constant.

Causes: Failure of x or y cylinders.

State #5 Cutting table is blocked

- The cutting table cannot move along one/ two directions.
- The reading of one/two position sensors remains constant.
- The output of the pressure sensor is normal.

Causes: Cutting table is blocked by fish or some other object

State #6: The pneumatic system is operating normally

State #7: Failure of the air compressor

Symptoms:

- The air pressure is not established.
- The vertical cutter does not move.
- The holding mechanism does not work.

Cause: There is no air pressure since the air compressor is totally shut down.

State #8: Failure of the vertical cutter blade cylinder

Symptoms:

- The vertical cutter does not work.
- There are no vibration signals from the accelerometer mounted on the cutter.

Cause: The cylinder fails or the vertical cutter is blocked.

State #9: Failure of the single action cylinders

Symptoms:

- No vibration signals from the holding mechanism.
- Holding mechanism does not work

Cause: Air pressure is not high enough or the holding mechanism is blocked.

State #10: Air leakage in the pneumatic system

Symptoms:

- The air pressure is decreasing.
- Actions of the vertical cutter are becoming sluggish.
- Actions of the holding mechanism are weak.
- Some peculiar noise due to air leakage.

Cause: The air leakage is taking place.

State #11: The conveyer system is working normally

State #12: The conveyer is blocked

Symptoms:

- The motion of the conveyer becomes sluggish.
- Some peculiar vibrations are generated by the conveyer.

Cause: The conveyer mechanism is blocked by fish or some other object.

State #13: Wrong position and orientation of the fish

Symptoms:

- Extra vibration and noise generated by the blocked fish
- The fish may not move along the conveyer.

Cause: A fish is blocked.

Table 5.1: List of simulated faults and sensor usage.

No.	Fault #	Use of sensors	Signals for diagnosis
1	#2 Failure of the hydraulic pump	#1 Microphone	Sound and pressure
2	#3 Failure of the electro-hydraulic FCS	4 Pressure Sensors	Pressure
3	#4 Failure of the x-axis and y-axis cylinders	#3 Accelerometer Position sensors Pressure Sensors	Pressure, position and vibration
4	#5 The cutting table is blocked	#4 Accelerometer Position sensors Pressure Sensors	Position, pressure and vibration
5	#7 Failure of the air compressor	#1 Microphone #1 Accelerometer	Sound and vibration
6	#8 Failure of the vertical cutter blade cylinder	#3 Accelerometer #2 Microphone	Sound and vibration
7	#9 Failure of the single action cylinders	#2 Accelerometer #1 Microphone	Sound and vibration
8	#10 Air leakage in the pneumatic system	#2 Microphone	Sound

Table 5.1: List of simulated faults and sensor usage.

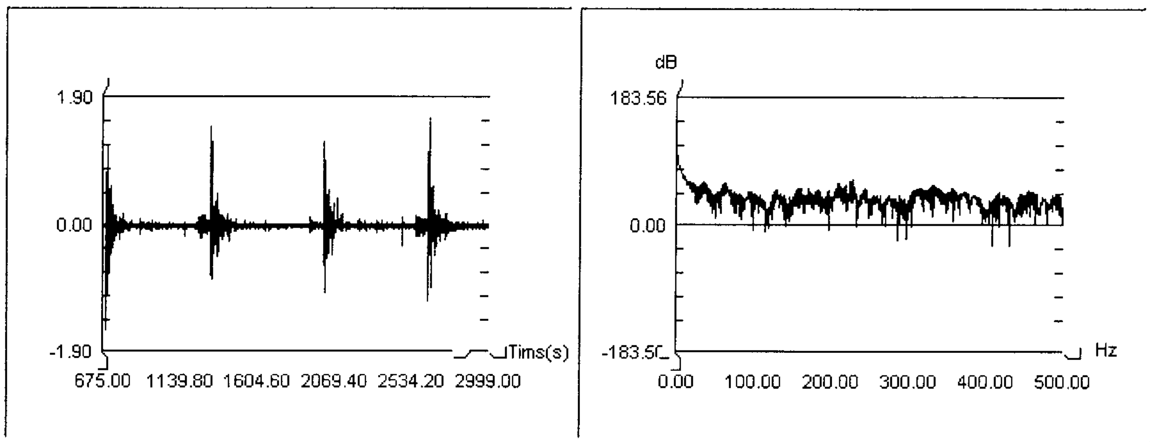
9	#12 The conveyer is blocked	#1 Accelerometer #2 Camera #1 Microphone	Sound, vibration and Vision
10	#13 Wrong position and orientation of the fish	#1 Camera	Sound, vibration and vision

5.3 Experimental Results

In this section, the neuro-fuzzy fault diagnosis system which was introduced in Chapter 3 is applied to the fish cutting machine for experimentation and performance evaluation. This diagnosis system will mainly focus on the six potential faults which were introduced in the previous chapters. The rest of the faults which were introduced in the previous section of this chapter are directly detected and diagnosed by single sensors.

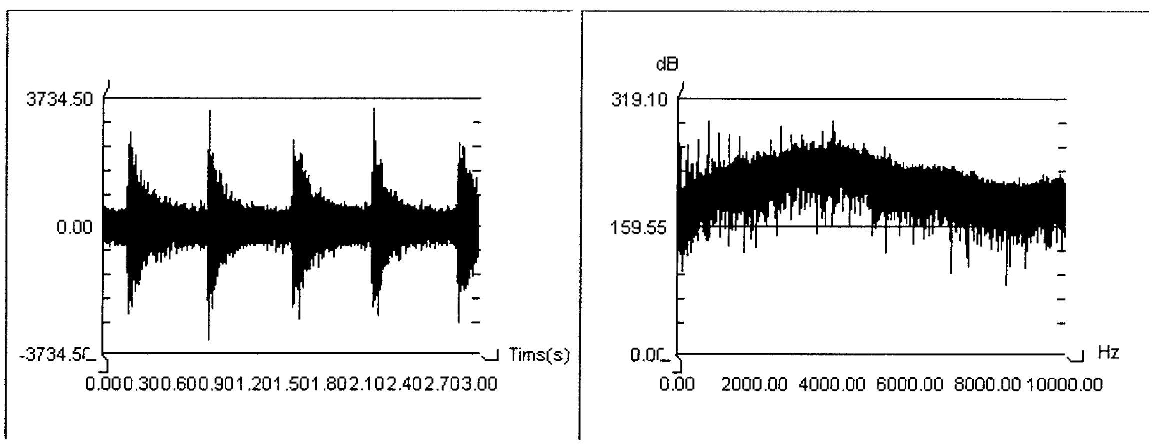
In the system, the outputs of the neural network represent the decision of the diagnosis system. When the output value is equal to or greater than 0.75, the relevant output node on the GUI software turns red which implies that the corresponding fault may be taking place at a very high probability. Similarly, it turns yellow when the output ranges from 0.25 to 0.75, which will imply a moderate possibility of the corresponding fault. It will be green if the output value is less than 0.25, which indicates that the machine is operating normally 9i.e., fault free). The diagnosis time is three seconds, which means that in every three seconds, new vectors of vibration, sound and vision signals are generated and sent to the diagnosis system for decision making.

Figure 5.6 shows the vibration signals in the time domain and the frequency domain under normal working conditions. It is a screen shot form the GUI of the running software. Because the duration of a data processing interval is three seconds, the time graph shows a period containing 3072 discrete points (sampling rate: 1024Hz). The frequency range displayed on the GUI is from 0 to 500Hz according to the sampling theorem. Figure 5.7 shows the sound signals in the time domain and the frequency domain under normal operating conditions. The data updating interval is also three seconds, and its sampling rate is 44,100 Hz.



(a) The vibration signal (b) The vibration spectrum

Figure 5.6: The vibration signal (Normal operating conditions).



(a) The sound signal (b) The sound spectrum

Figure 5.7: The sound signal (Normal operating conditions).

Table 5.2 shows the feature vectors of the vibration and sound signals containing four elements. The generation of the feature vectors was discussed in Chapter 2. They form the inputs to the neuro-fuzzy fault diagnosis network. In this normal operating condition, the six outputs of the network are 0.112, 0.153, 0.201, 0.187, 0.215, 0.098, which indicate that all six outputs are less than 0.25. Therefore, the output nodes in the UGI appear green, indicating a normal operating condition of the fish cutting machine.

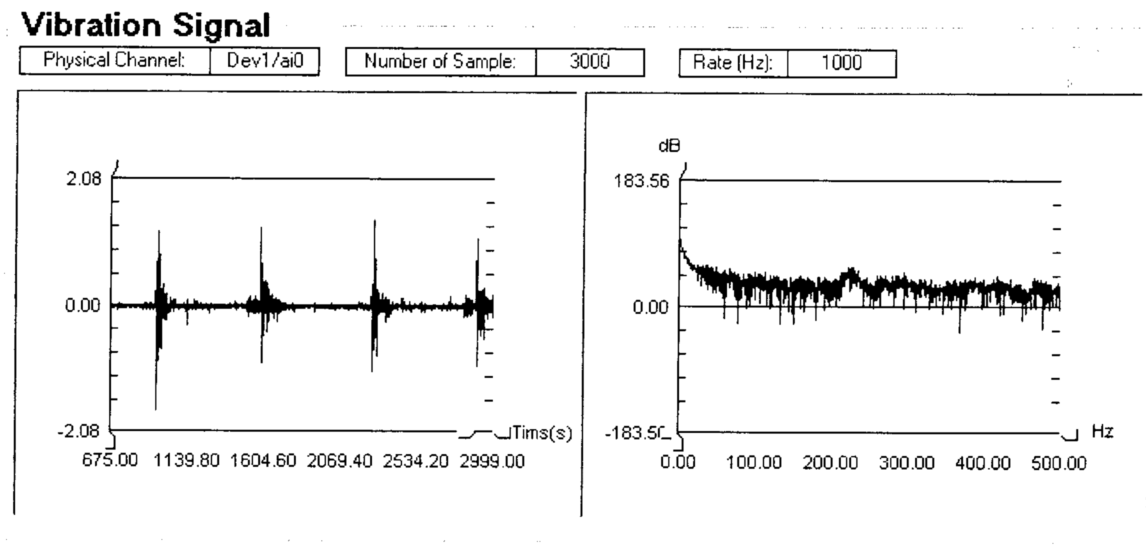
Table 5.2: Feature vectors of vibration and sound signals in a normal operating condition.

Vibration feature vector	0.49156	0.55061	0.50059	0.45234
Sound feature vector	0.69448	0.48401	0.41385	0.33491

Failure of the Hydraulic Pump

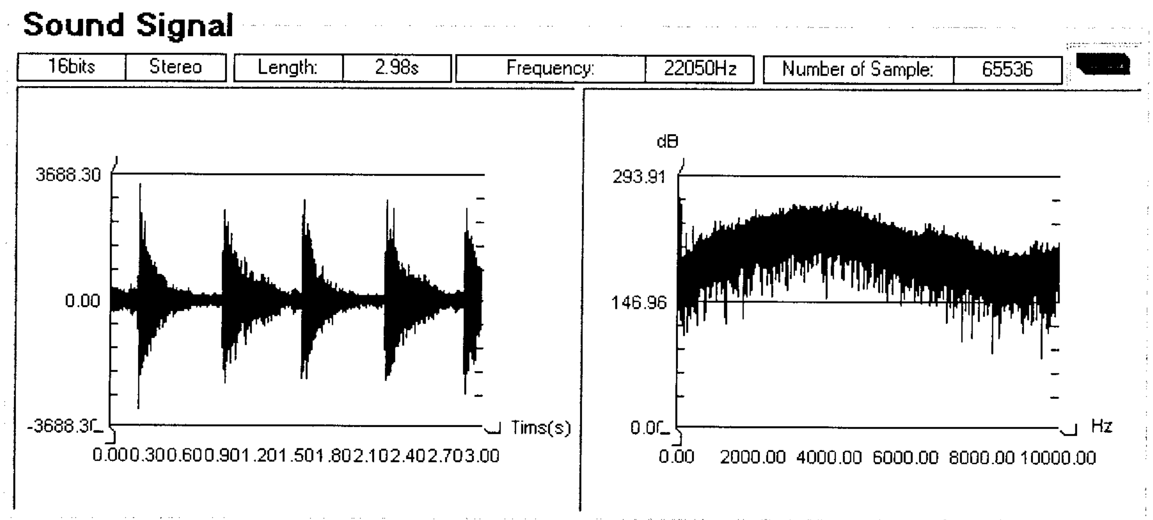
Figure 5.8 shows the vibration signals and their frequency spectra when the hydraulic pump fails. Figure 5.9 shows the sound signals and their frequency spectra under the same condition. Subsequently, the vibration and sound feature vectors are generated based on the vibration and sound spectra, as given in Table 5.3. In this operating condition, the value of the 4th output of the neuro-fuzzy network is 0.92 which has a high probability that the hydraulic pumping is not operating properly. The other five output values are less than 0.25 which indicate that the other components of the machine are working properly. Consequently, it is observed that the 4th output of the diagnosis network on the GUI of the software turns red, which will generate an alarm while the other five output nodes remain green.

Figure 5.10 shows the outputs of the nuero-fuzzy diagnosis network when the hydraulic pump fails. The failure time is about 900 seconds. A diagnosis result is generated in each three second period. As seen in Figure 5.10, the value of the fourth output remains greater than 0.75 which indicates that the hydraulic pump is failing at a high possibility.



(a) The vibration signal (b) The vibration spectrum

Figure 5.8: The vibration signal (hydraulic system fails).



(a) The sound signal (b) The sound spectrum

Figure 5.9: The sound signal (hydraulic system fails).

Table 5.3: Feature vectors of vibration and sound signals when the hydraulic pump fails.

Vibration feature vector	0.6828	0.47092	0.42323	0.36454
Sound feature vector	0.49292	0.55339	0.50143	0.4465

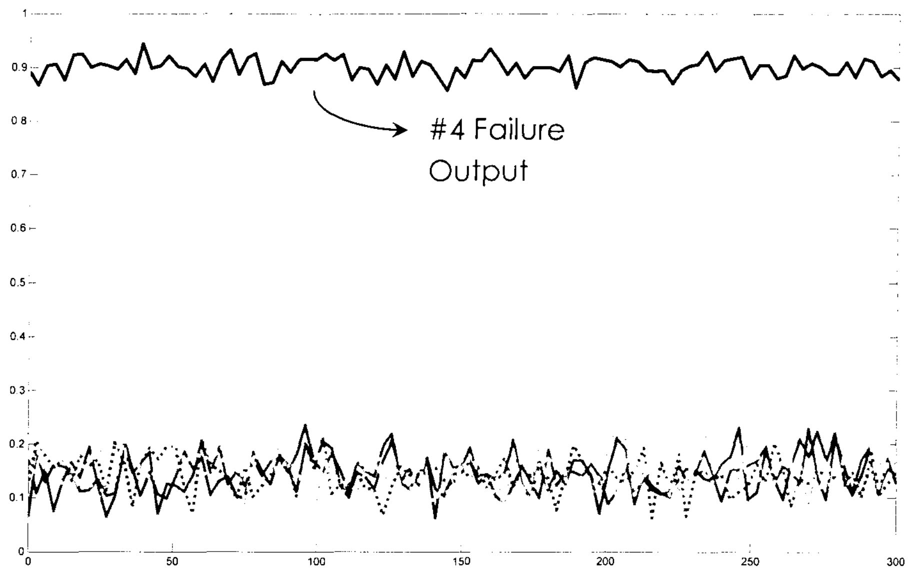
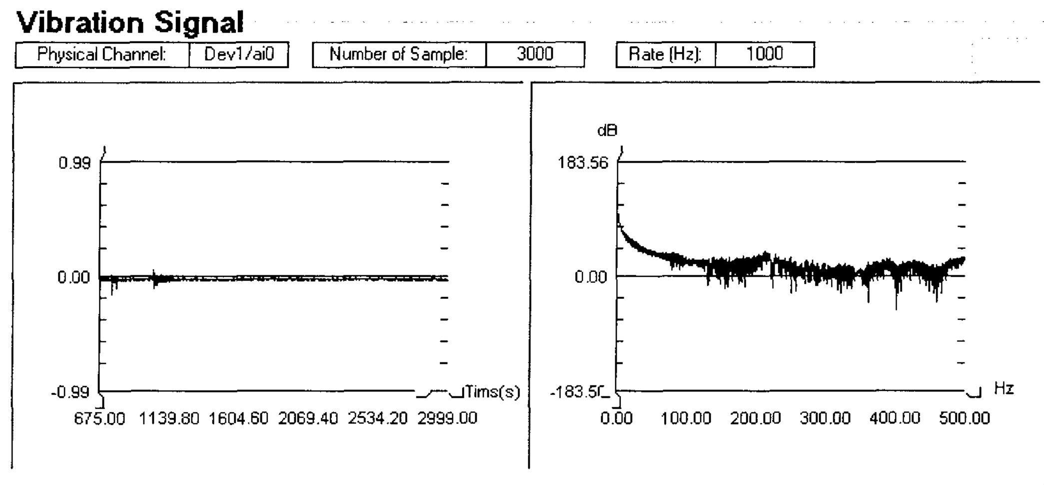


Figure 5.10: The outputs of the diagnosis system.

Failure of Conveyor System

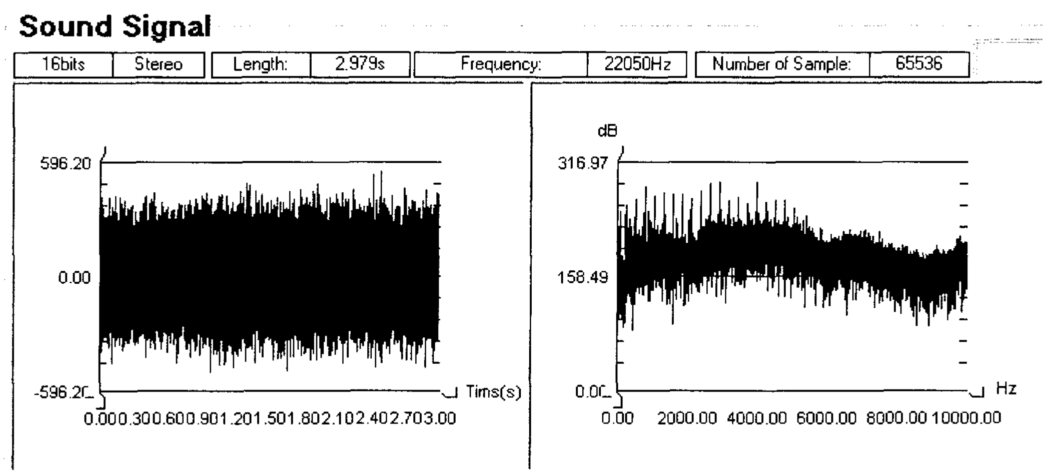
Figure 5.11 shows the vibration signal and its frequency spectrum when the conveyor system fails, while Figure 5.12 shows the sound signal and its frequency spectrum. It is observed from figures 5.11 and 5.12 that the pulses of vibration and sound signals in the time domain, seen to be in the normal condition, disappear when the conveyor system is down. Table 5.4 shows the vibration and sound feature vectors for the neuro-fuzzy diagnosis system under this operating condition. The value of the 3rd output of the neuro-fuzzy system is 0.89 which indicates a high probability for the conveyor system to malfunction. The output values of the other five nodes are less than 0.25 which means that the other components of this machine operate properly. Similar to the case of hydraulic pump failure, the 3rd output of the fault diagnosis network turns red and the rest of the nodes remain green.

Figure 5.13 shows the outputs of the neuron-fuzzy network when the conveyor system fails. The duration of the failure is 900 seconds. A diagnosis result is generated in every three seconds. As seen from the figure, the third output remains greater than 0.75 which indicates that the conveyor system is failed at a high probability.



(a) The vibration signal (b) The vibration spectrum

Figure 5.11: The vibration signal (hydraulic system fails).



(a) The sound signal (b) The sound spectrum

Figure 5.12: Sound signals (hydraulic system fails).

Table 5.4: Feature vectors of the vibration and sound signals when conveyor system is failed.

Vibration feature vector	0.86514	0.39742	0.14608	0.26881
Sound feature vector	0.57642	0.0651	0.60424	0.54625

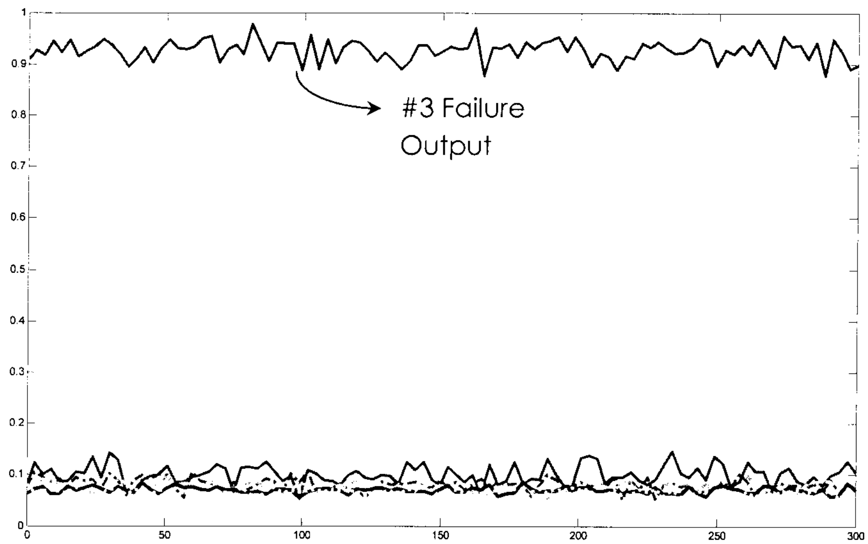


Figure 5.13: The outputs of the diagnosis system.

5.4 Advantages of the Multi-sensor Fusion Approach

In this section, a comparison between single sensor detection and multi-sensor fusion detection is given and the performance is discussed when faults occur in the hydraulic system. Through this comparison, the benefits of the multi-sensor fusion approach are highlighted.

As discussed in Section 5.2, the hydraulic system failure can be caused by several factors such as failure of the hydraulic cylinder, failure of the hydraulic pump, and the presence of blocked/tangled fish. When these faults occur, it is very difficult for a single sensor (sound, vibration or vision sensors) to determine the fault source because the sound signal can only be used to detect the failure of the hydraulic pump; the vision signal can only be used to judge the failure of blocked fish; and the vibration signal can only be applied to detect the motion of the cutting table. If the cutting table does not work, failure of any of these components (the hydraulic pump, cylinder, etc.) may cause this problem. In this case, multi-sensor fusion technology becomes necessary because the signal sensor approach is not adequate to detect and diagnose all fault sources. The following experiment will demonstrate this case and validate the multi-sensor fusion approach in dete3cting and diagnosing faults in a complex system.

Figure 5.14 shows vibration and sound signals when the hydraulic cylinder fails. Non-moving cutting table is the symptom of this failure. Although a single vibration sensor can detect the motion of the cutter, it cannot tell which component of the hydraulic system has failed. In this situation, by combining the sound and vision signals, the neuro-fuzzy sensor fusion system is able to determine that the hydraulic pump operates normally, and no fish is blocked.

Figure 5.15 shows how the diagnosis network has identified the three types of failure in the hydraulic system. In this experiment, the machine was first run under normal conditions for 30 seconds. Then a fault of blocked fish was simulated on the machine from the 30th to 40th second. It is observed from Figure 5.15 (a) that the value of the first output of the diagnosis network has increased significantly while the other outputs still remain at low values. Because the first output of the network corresponds to the fault of "failure of blocked fish," Figure 5.15 (a) demonstrates that the network has correctly identified the pump failure in that period.

Next, another failure--the hydraulic cylinder failure--was simulated at approximately 69^{th} second. Again, this fault has been correctly captured by the neuro-fuzzy network, as shown in Figure 5.15(b). Finally, at the 153^{rd} second, the hydraulic pump was suddenly shut down to simulate the fault "hydraulic pump failure." Figure 5.15 (d) shows that the value of the 4^{th} output of the network has increased suddenly at that time, which indicates that there exists pump failure.

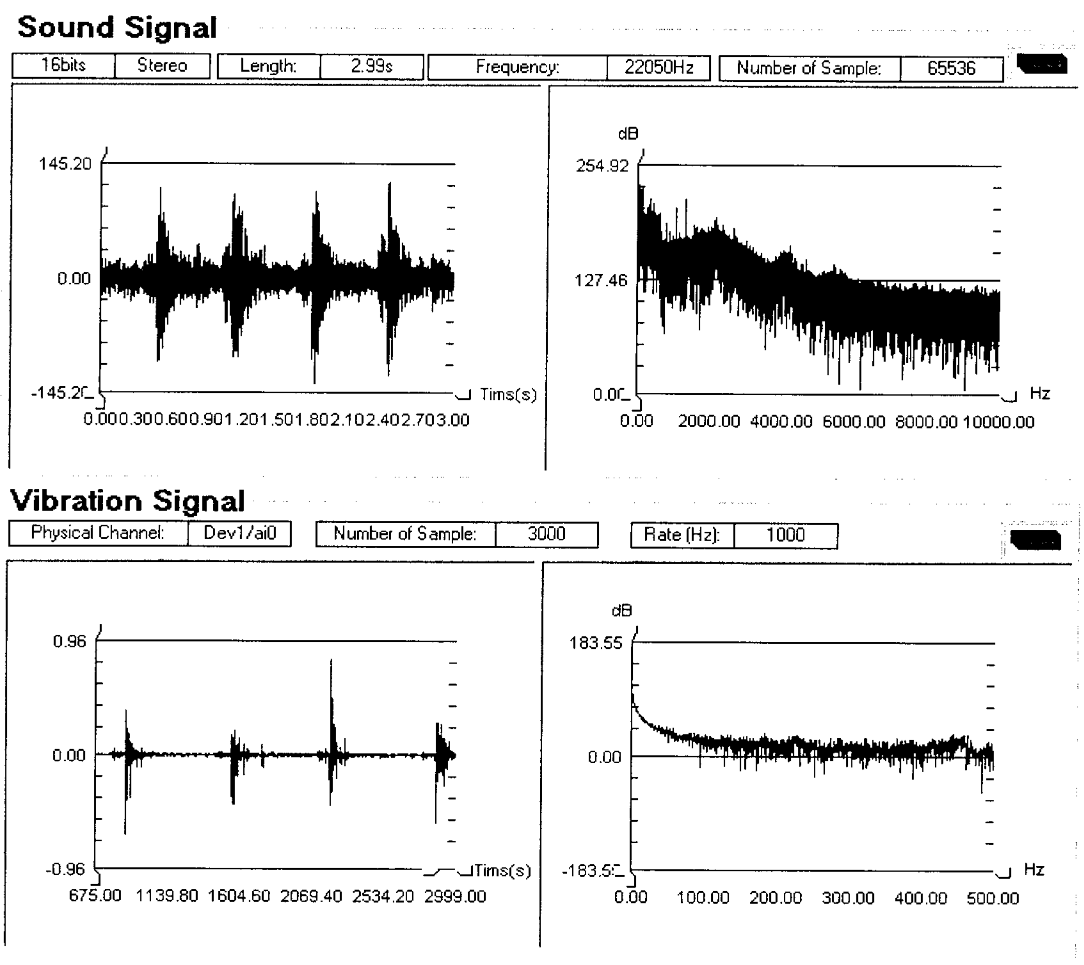
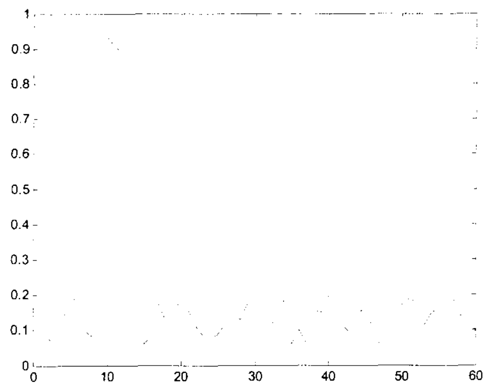
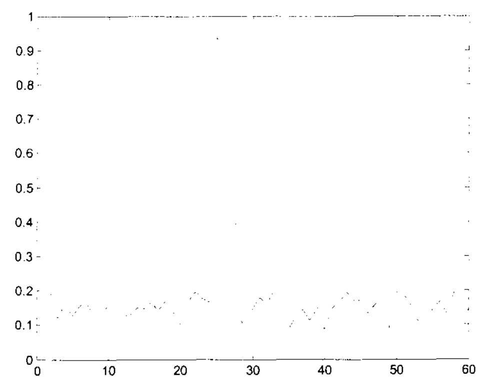


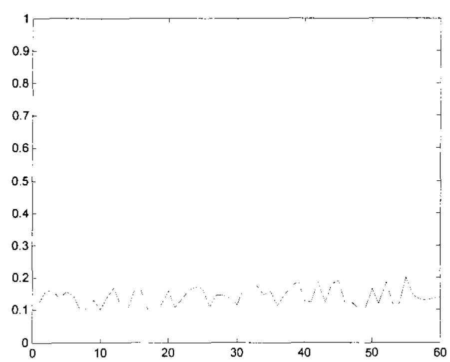
Figure 5.14: The signals under failure of the hydraulic cylinder.



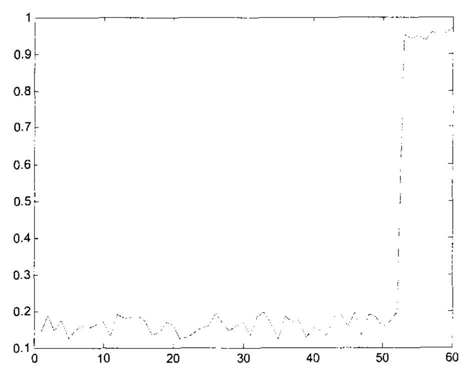
(a) #1 Output



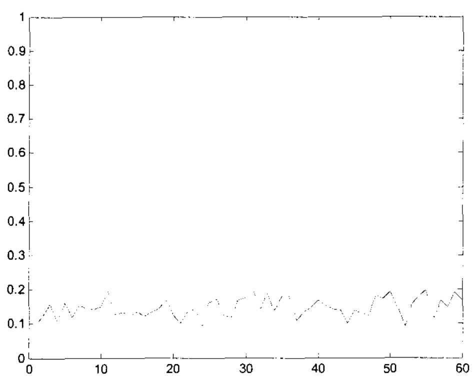
(b) #2 Output



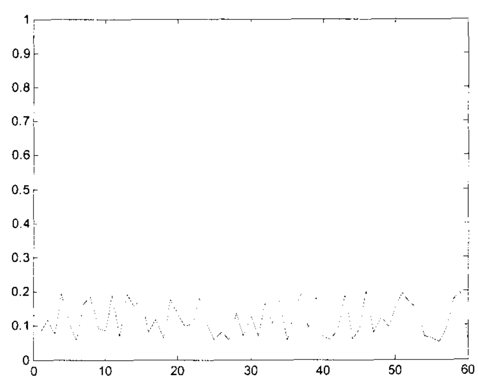
(c) #3 Output



(d) #4 Output



(e) #5 Output



(f) #6 Output

Figure 5.15: The outputs of a sensor fusion diagnosis experiment.

5.5 Summary

In this chapter, multiple experiments were carried out to validate the neuro-fuzzy multi-sensor fusion approach developed in the present thesis. First, the test bed, which is the industrial fish cutting machine, was described, and the potential faults in the fish cutting machine and their possible solutions were presented. The proposed and developed neuro-fuzzy fault diagnosis system was implemented on the fish cutting machine for real time fault diagnosis experimentation. Two fault working conditions, the failure of the hydraulic pump and the failure of the conveyor system, were simulated, and the diagnosis results were presented. Furthermore, a comparison of single sensor fault diagnosis and multi-sensor-fusion fault diagnosis approaches was presented. The experimental results in this chapter showed that the developed multi-sensor fusion approach was quite effective and robust in detecting and diagnosing potential faults in a complex industrial machine.

Chapter 6

Conclusion

6.1 Main Contributions

In this thesis, a fault diagnosis system based on a neuro-fuzzy sensor fusion approach was developed for a real engineering system—an industrial fish cutting machine. Vibration, sound and vision signals were detected by three types of sensors: accelerometer, microphone and CCD camera, which were used as inputs to the diagnosis system. A feature vector was defined and extracted from the FFT frequency spectra of the acquired signals. Also, a feature based vision tracking approach known as the Scale Invariant Feature Transform (SIFT) was applied to the vision data to track the object of interest (fish) in a robust manner. A four-layer neural network including a fuzzy layer was developed to analyze and diagnose a range of primary faults. By training the neural network with sample data for typical faults, six crucial faults in the fish cutting machine were accurately detected. Alarms to warn about impending faults may be generated as well during the machine operation.

In addition, a network based remote monitoring architecture was developed in the thesis, which could facilitate engineers to monitor the machine condition in a more flexible manner from a remote site. Developed approaches were implemented in a realistic experimental environment and validated using computer simulations and physical experimentation using the industrial fish processing machine.

6.2 Future Research

Although the neuro-fuzzy approach developed and evaluated in the present thesis has demonstrated good performance, there exist several issues and aspects that need to be further investigated and explored. For example, the neuro-fuzzy fault diagnosis system presented in this thesis is applicable in the detection of only six crucial faults, since it consists of six outputs corresponding to the six failures. However, it will be beneficial to extend the scheme to some intermediate fault types. More detailed classification of faults can be made and more rigorous experimentation and simulation may be carried out using the industrial fish cutting machine.

It is expected that if more sensors are used and their output information is fused using the neuro-fuzzy diagnosis architecture developed in this thesis, the performance of the fault diagnosis system would be further improved. Consequently, more faults of the machine could be detected and diagnosed.

A more effective strategy of feature vector generation may be employed so that the signatures of the signals can be extracted more accurately. Some advanced methods, such as statistical methods, Bayes modeling, wavelet techniques are believed to be useful in this context.

Bibliography

Alpaydin, E., *Introduction to Machine Learning,* MIT Press, Cambridge, MA, 2004.

Abouelela, A., Abbas, M. H., Eldeeb, H., Wahdan, A., and Nassar, M. S., "Automated vision system for localizing structural defects in textile fabrics," *Pattern recognition letters*, Vol. 26, No. 10, pp. 1435-1443, 2005.

Benko, U., Petrovcic, J., Juricic, D., Tavcar, J., Rejec, J., and A. Stefanovska, "Fault diagnosis of a vacuum cleaner motor by means of sound analysis," *Journal of sound and vibration*, Vol. 276, No. 3-5, pp. 781-806, 2004.

Bertozzi, M., Broggi, A., and Castelluccio, S., "A real-time oriented system for vehicle detection," *Journal of Systems Architecture*, Vol. 43, No. 1, pp. 317-325, 1997.

Betta, G., Liguori, C., Paolillo, A., and Pietrosanto, A., "A DSP-based FFT-analyzer for the fault diagnosis of rotating machine based on vibration analysis," *IEEE Transactions on Instrumentation and Measurement*, Vol. 51, No. 6, pp. 1316-1322, 2000.

Betta, G., D'Apuzzo, M., Liguori, C., and Pietrosanto, A., "An Intelligent FFT-Analyzer," *IEEE Transactions on Instrumentation and Measurement*, Vol. 51, No. 6, pp. 1173-1179, 1998.

Buluswar, D. S. and Draper, A.B., "Color machine vision for autonomous vehicles," *Engineering Applications of Artificial Intelligence*, Vol. 11, No. 2 , pp. 245-256, 1998.

de Silva, C.W., "Research Laboratory for Fish Processing Automation," *International Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 9, No. 1, pp. 49 - 60, 1992.

de Silva, C. W., *Mechatronics—An Integrated Approach,* Taylor&Francis/CRC Press, Boca Raton, FL, 2005a.

de Silva, C. W., Editor, *Vibration and Shock Handbook,* Taylor&Francis/CRC Press, Boca Raton, FL, 2005b.

de Silva, C. W., *Sensors and Actuators—Control System Instrumentation,* Taylor&Francis/CRC Press, Boca Raton, FL, 2007a.

de Silva, C. W., *Vibration—Fundamentals and practice,* 2nd edition, Taylor&Francis/CRC Press, Boca Raton, FL, 2007b.

Fučík O., Zemčík P., Tupec P., Crha L., and Herout A., "The Networked Photo-Enforcement and Traffic Monitoring System, " *Proceedings of Engineering of Computer-based Systems*, pp. 423-428, Los Alamitos, US, 2004.

Ghosh, N., Ravi, Y. B., Patra, A., Mukhopadhyay, S., Paul, S., Mohanty, A. R., and Chattopadhyay, A. B., "Estimation of tool wear during CNC milling using neural network-based sensor fusion ," *Mechanical Systems and Signal Processing*, Vol. 21, No. 1, pp. 466-479, 2007.

Goerick, C., Detlev, N., and Werner, M., "Artificial Neural Networks in real-time car detection and tracking applications," *Pattern Recognition Letters,* Vol. 17, No.4, pp. 335-343, 1996.

Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., and Seelen, W. V., "An image processing system for driver assistance," *Image and Vision Computing*, Vol. 18, No. 5, pp. 367-376, 2000.

Hong, J. S. and May, S. G., "Neural-network-based sensor fusion of optical emission and mass spectroscopy data for real-time fault detection in reactive ion etching," *IEEE Transactions on Industrial Electronics*, Vol. 52, No. 4, pp. 1063-1072, 2005.

Hui, S.C. and Wang, F., "Remote video monitoring over the WWW," *Multimedia Tools and Applications*, Volume 21, No. 2, pp. 173-195, 2003.

Ito, T., Yamada, K., and Nishioka, K., "Understanding driving situation using a network model," *Intelligent Vehicles '95 Symposium., pp 48-53, 1995.*

Karray, F. and de Silva, C.W., *Soft Computing and Intelligent Systems Design*, Addison Wesley, New Yotk, 2004.

Lathi, P. B., *Linear System and Signals*, Oxford University Press, Oxford, UK, 2002.

Lowe, G. D., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004.

Marini, D. , Rossi, M., Moltedo, L., and Salvetti, O., "Virtual reality and Web tools to convey the visual information of ancient monuments," *Computer networks and ISDN systems*, Volume 29, No. 14, pp. 1655-1660,October 1997.

Matthews, N., An, P., Charnley, D., and Harris, C., "Vehicle detection and recognition in greyscale imagery," *Control Eng. Practice,* Vol. 4, No. 4, pp. 473-479, 1996.

Palluat, N., Racoceanu, D., and Zerhouni, N., "A neuro-fuzzy monitoring system Application to flexible production systems," *Computers in Industry*, Vol. 57, No. 6, pp. 528-538, 2006.

Park, B. S., Lee, W. J., and Kim, K. S., "Content-based image classification using a neural network," *Pattern Recognition Letters*, Vol. 25, No. 3, pp. 287-300, 2004.

Rohrig, C. and Jochheim, A., "The Virtual Lab for controlling real experiments via Internet," *Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on*, pp. 279-284, Kohala Coast, HI, USA, 1999.

Rubio, E., Sanz, A., and Sebastián, M., "Virtual reality applications for the next-generation manufacturing," *International Journal of Computer Integrated Manufacturing*, Vol. 18, N0. 7, pp. 601-609, 2005.

Seelen, V. W., Curio, C., Gayko, J., Handmann, U., and Kalinke, T., "Scene analysis organization of behavior in driver assistance system," *IEEE Int'l Conf. Image Processing*, Vancouver, BC, Canada, Vol.3, pp. 524-527, 2000.

Sun, Z., Bebis, G., and Miller, R., "On-road vehicle detection: a review," *IEEE Transactions on Patten Analysis and Machine Intelligence*, Vol. 28, No. 5, pp. 585- 590, 2006.

Tse, P. W., Yang, W., and Tam, H. Y., "Machine fault diagnosis through an effective exact wavelet analysis," *Journal of Sound and Vibration*, Vol. 277, No. 4-5, pp. 1005-1024, 2004.

Tsoukalas, H. L. and Uhrig, E. R., *Fuzzy and Neural Approaches in Engineering*, A Wiley-Interscience Publication, John Wiley & Sons, Inc, New York, 1997.

Shibata, K., Takahashi, A., and Shirai, T., "Fault diagnosis of rotating machinery through visualization of sound signals", *Mechanical Systems and Signal Processing*, Vol. 14, No. 2, pp. 229-241, 2000.

Verma, V. and Simmons, R., "Scalable robot fault detection and identification,", *Robotics and Autonomous Systems*, Vol. 54, No. 2, pp.184-191, 2006.

Wang, W. Q., Golnaraghi, F. M., and Ismail, F., "Prognosis of machine health condition using neuro-fuzzy systems," *Mechanical Systems and Signal Processing*, Volume 18, No. 4, pp.813-831, 2004.

Wu, J. and Zhang, X., "A PCA classifier and its application in vehicle detection," *Proc. IEEE Int'l Conf. Neural Networks*, Washington, DC, Vol. 1, No. 1, pp. 600-604, 2001.