The Adaptive Intelligent Model for Process Diagnosis, Prediction and Control

Dr.ing Thesis

By

Meng, Tang

Department of Production and Quality Engineering
Faculty of Engineering Science and Technology
Norwegian University of Science and Technology (NTNU)

Trondheim, 2004

Printed by Tapir

ISBN 82-471-6537-6 (electronic)
ISBN 82-471-6538-4 (printed)
ISSN 1503-8181
NTNU series 2004:154

Preface

The work presented in this dissertation was carried out at the Department of Production and Quality Engineering (IPK), Faculty of Engineering and Technology, the Norwegian University of Science and Technology (NTNU) within the period Jan 2001 to November 2004. Professor Wolfgang H. Koch from this department supervised the work.

With much pleasure, I would like to take the opportunity to thank all who stood beside me and always supported me through my studies.

First of all, I would like to express my gratitude to my supervisor, Prof Wolfgang H. Koch, for his constant support, invaluable guidance, advice and encouragement in the course of this research work.

My appreciation is also extended to Professor J. Vatn from IPK, thanks for his help, support and valuable advice during my research work.

Special thank is also for my colleagues and good friends, Dr. Aleksandar Milovic and Dr. Aleksandar Sekulic, who always gave me constant support and help during my doctoral studies. Their enthusiasm and friendship made my study and work in profession as well as personal development. They have made the period of my doctoral studies both enjoyable and unforgettable.

This work could not have been a reality without the help of Mr. Lars H.Vik in SINTEF Industrial Management Division. He provided important data and valuable discussions and suggestions in my work for me.

Finally, special thank for my wife, Wei Qin, it was impossible to fulfill my work without her love, encouragement and understanding, limitless tolerance, support and patience during the whole period of my graduate studies.

Trondheim, Norway
August 2004

Meng Tang

The Adaptive Intelligent Model for Process Diagnosis, Prediction and Control

Production and Quality Engineering (IPK),
Faulty of Technology and Science, NTNU
Meng Tang

Abstract

This research work focuses at first on the intelligent model development for process state (special for fault) detection, behavior prediction and process control for complex industrial processes. In the model architecture, Fuzzy Neural Networks (FNNs) are employed as process state classifiers for process state (fault) detection. Other (different) Neural Networks (NNs) models are applied for system identification of process characteristics in different process states. The model detects process states (faults) and predicts process behavior according to process input and historical output (behaviors), whose combination of influences generates the final results of process state (fault) detection and quantitative prediction. The whole model is constructed based on Fuzzy TS dynamic Nonlinear AutoRegressive with eXogenous input models (NARX models).

Secondly, two different model optimization schemes for different purposes are designed to minimize an objective function, which is defined as the difference between the desired response and the real process output. One is for optimal diagnosis and another is for optimal prediction. Different methods have been applied for model optimization. It works to deal with time varying processes.

Thirdly, a specific state space equation of discrete time varying system is being derived from the intelligent model. In the state space equation, the state transition matrix A is determined by the fuzzy degree of process state classification produced by process historical behavior in time t instant, and the input transition matrix B by the fuzzy degree of process state classification produced by process real input in time t instant. The state observer vector H is determined by optimization results generated by model optimal or even adaptive scheme. Based on this kind of state space equation, some process control strategies can be developed for predictive model control or optimal process control, which finally realize process control based on fault detection and prediction.

Lastly, to confirm the validity of the theoretical results from above, an application case has been studied for supply forecasting in supply chain management. The study and application results indicate that the model not only has good performance for fault detection, but also provides excellent quantitative prediction of process output. It can be applied in process state (fault) detection, diagnosis and prediction for process behavior, as well as optimal process control.

One paper related to the research work has been published in the international conference "IEEE 14th Intelligent Systems Design and Applications (ISDA 2004) ", August 26-28, 2004, Budapest, Hungary [Tang, M., Koch, W. H., 2004] and another has been accepted in "2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS)" Singapore, December 1-3, 2004.

Summary

The main objective of the work reported in the thesis has been to study and implement models and methodologies that can improve process diagnosis, behavior prediction and control in non-linear industry processes or systems. The emphasis is to model complex process with intelligent technologies and then provide good analysis and control capability. The model and modeling approaches are mainly based on *Computational Intelligence* (CI) technologies *System* and *Control* technologies, for instance, fuzzy inference systems, neural networks, Genetic Algorithms and system identification, optimization and control as well.

Information and Communication Technologies (ICT) have shown a strong life and perspective to penetrate to every application fields within the last decades, especially in industry and business areas. ICT play more and more important role in these areas and also effectively impulses development progress in these traditional areas. On the other hand, the industry and business processes also become more and more complex due to more operations and activities to be involved in. With the product, market and economy competition globalization, the production and business processes are being changed from facing to products into facing to customers. Therefore, industrial and business companies have to improve production efficiency, operation flexibility, response time and accuracy for decision-making to response the customer's requirements and market changes.

The ICT enables to collect data from industrial and business processes. Monitoring and control are used in time dependent industrial and business processes in order to ensure that these processes are effective. Monitoring is particularly important in aligning the process with other processes and ensuring the process operation according to the desired specifications. Many variables are measured during monitoring process, for example, in industrial processes, it may includes pressure, temperature, humidity, power, etc, and in business processes, it should include price, product, amount, sale, customers, supply, transport, etc. The amount of data collected from processes is always increasing with the time. Thus, a challenge is how to model the process with massive data and then get the valuable information and knowledge in order to guide decision making and process operations. This is the main research aim in this thesis.

The complication of the industrial and business processes makes it difficult to be modeled with traditional mathematical model and approaches based on inner laws of the real system or process. *Computational Intelligence* (CI), as one kind of effective tool, provides the good ability to model complex processes based on their historical data without priori condition. The *Fuzzy Logic* (FL) in CI is a tool for embedding existing structured human knowledge into mathematical models. It can not only be used to express some vague, uncertain, or even linguistic variables into mathematical models, but also simulate human being's reasoning and inference process. The *Neural Networks* (NNs) in CI recovers underlying dependencies between the given inputs and outputs by using training data sets, after training, NNs represent high dimensional nonlinear functions in multidimensional space, which can approximately express real relationship of system inputs-outputs. They are typical mathematical models obtained in an experimental way, namely, learning from historical data. All of the advantages of CI

provide strong tools and applications to model complex non-linear system or processes just based on historical data.

One important branch covering above problems is *Data Mining* (DM) *and Knowledge Discovery* (KD). It bridges the gap between data and information and focuses on extracting useful knowledge and information that hides behind a massive amount of data. In the research work, it is used to extract underlying knowledge for process diagnosis.

System optimization, identification and parameter estimation are applied for optimal model and model adaptability so as to cover nonlinear or dynamic characteristics of real processes in this research work.

As a conclusion of all, the thesis focuses on building model for fault diagnosis, behavior prediction as well as process control with application of computational intelligence, data mining and system optimization and control methods, for example, defining process state, identifying change of process state, predicting process behavior, especially in abnormal behavior, and mining the causes that trigger the aberrant behavior in time series during in the industrial and business processes. Some main aims as follows have been investigated:

- The process state identification, performance measurement and improvement
- The fault diagnosis and process output prediction
- Process control.

To realize these goals, first, a definition and approach based on Fuzzy logic for process state description is developed, the defined process state is a criterion for measurement and improvement of process performance and it is also a criterion for fault detection. A set of membership functions are generated for all input and output variables based on the state definition and approaches. All data can be fuzzfied with these membership functions of input-output variables. The second, Fuzzy Neural Networks (FNNs) are trained with these input-output fuzzy variables and also designed as a residual generator (or state classifier) for fault detection (or process state classification). After process states are identified by fuzzy neural networks, different NN models are applied in order to build different models for different process states. The third, whole model is constructed based on Fuzzy TS dynamic NARX model, which is used for fault detection and output prediction. In the Fuzzy TS dynamic NARX model, the antecedent part, which is generated by two fuzzy classifiers**,** constructs the residual generator for fault detection (or process state classification), the consequence part, which is generated by system identification based module Neural Networks and fuzzy partition information of process states, constructs the process prediction output. The map relationship between input-output of fuzzy state classifiers describes the relationship among system input (fault symptom) and process state (fault occurrence).

In the model, to state classification (also fault detection), the final fault residual signal is generated by output of two fuzzy classifiers, one fuzzy classifier is used for abnormal process state (fault) detection measured from real system inputs and another fuzzy classifier is used for abnormal process state (fault) detection measured from historical behaviors. The outputs of two fuzzy classifiers are calculated by fuzzy operation to produce final fault residual signal. If the residual signal is beyond a defined threshold, then an abnormal (fault) event could be detected by this system. To process prediction output, it is determined by the combination of two aspects, one is produced by real system input, and another is by historical behaviors of real process. The two kinds of process outputs are integrated by Fuzzy TS model so as to produce a final process output.

At the same time, data mining technology is used to extract relevant process diagnosis rules when process state changes from one to another. These rules can be applied in fault diagnosis, behavior control and prediction as well as process risk analysis and process performance measurement.

Two different model optimization schemes have been investigated for different purposes to cover time-varying process. One is for process diagnosis and another is for process prediction. Based on different model optimization schemes, different optimization methods are applied. In the thesis, nonlinear optimization and GA are used to optimize model key parameters, which are a threshold for process state classification and sensibility, so as to provide optimal process diagnosis. Least Square (LS) approaches are employed to describe and estimate optimization parameter so as to provide optimal process prediction. Hence, the integrated intelligent model possesses adaptive threshold for different process state classification as well as provide optimal process prediction.

State space equation is basic mathematic model and tool for system control. To the intelligent model, it can be derived as a discrete time-varying system model expressed by a specific state space equation, whose state variable is defined as process states in real processes. In the state space equation, the state transition matrix A is determined by the fuzzy degree of process state classification produced by process historical behavior in time t instant, and input transition matrix B by the fuzzy degree of process state classification produced by process real input in time t instant, and state observer vector H is determined by optimization result generated by model optimal or adaptive scheme. Based on this state space equation, process control strategies can be developed for predictive model control or optimal process control, which can realized process control based on state detection and predictive output.

Thus, the application model built with CI, data mining and system optimization and control provides a kind of research methodologies and ideas to analyze complicated non-linear process based on learning from historical data. A real case is developed based on this model for supply forecasting in supply chain management. It shows a good result can be reached based on this model and methodologies. The model also can be applied in difference process analysis and control, for example, dynamic analysis for transport system, node flow analysis and measurement in network system, process risk analysis and so on.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 INTRODCUTION

1.1 Industry and business processes

1.1.1 Introduction to Industry and Business Processes

An industrial process is a series of operations performed in manufacturing /production or some other industry activities. A business process can be regarded as a set of logically related tasks performed to achieve a defined business outcome and goal [E. Ikonen, K. Najim, 2002]. During the whole process, many factors are involved in and play different and important roles for whole process state, performance and output.

From system and control point of view, a complex industry and business process can be treated as close loop system as following Figure 1.1.

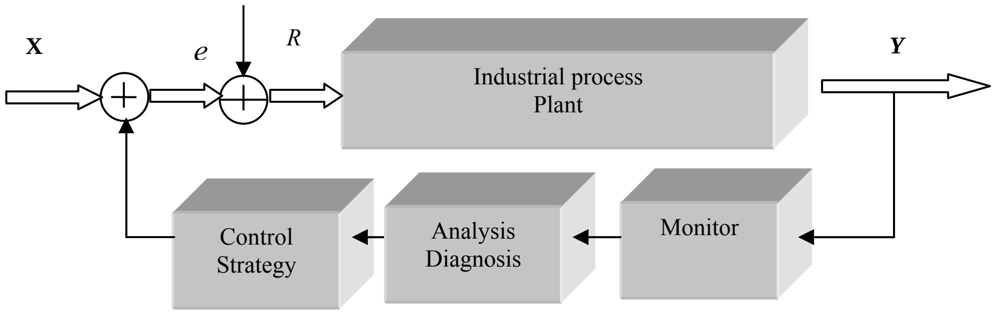


Figure 1.1: Basic elements in a closed loop industrial process

Where, X is reference input of system, Y is the system output and R is disturbance from outside of system.

Monitor: determining the feature attributes that are very important to represent the plant characteristics and collecting data from the process based on certain data acquisition scheme or system online.

Analysis and Diagnosis: processing data primarily, for example, reducing noise data, analyzing acquired data based on certain physical laws or business rules, and providing system fault diagnosis, performance measurement and behavior attributes and prediction with support of knowledge base or expert system, the diagnosis system will discover some potential problems about the plant.

Control: adjusting input variables, system parameters or control parameters/rules based on result of the system analysis and diagnosis in order to fit the goals of system process as much as possible. Many different control schemes have been developed based on different models and objects, for example, real time and multivariable feedback control for a dynamic system, process control for industrial processes, quality control for production processes and process and cost control for business processes and so on.

One aspect, enterprises devote to seek effective methods so as to improve performance of industrial and business processes, on the other aspects, further requirements need these methods not only implement correct process control and instructions, but also provide good process analysis and decision making support during whole process.

Hence, good model architecture and implementation methodologies are important and active to industrial or business processes and as well as research field.

In this thesis, the main research concentrates on researching a kind of methods and model for process state detection, process diagnosis and output prediction, as well as process control in industry and business processes, namely, modeling for complex process, diagnosing abnormal state (fault) for potential process problem, and revealing the process characteristics, discovering the knowledge inner the process and further predicting the process behaviors, implementing certain process control, etc.

1.1.2 Changes and Challenges to Industry and Business Processes

Today, with the science and technology progress, the industry and business processes become more and more complex due to more complex operation and activities to be involved in. With the product, market and economy competition globalization, the more and more production and business processes are being changed from facing to products into facing to customers, namely, change from products as center into customers as center. These industry and business companies have to improve production efficiency, operation flexibility and accurate ability for making decision so as to response the customer's requirement and market change quickly and well. These trends require the industry and business processes more and more flexible and effective. As the result of global competition, the enterprises have to face more and more complicated industry and business processes and activities due to:

- More and more drastic product global competition
- More and more factors are involved in industry and business processes
- More and more personalized needs from customers
- Rapid market response
- More production, organization type are being involved in, for example, virtual factory, virtual manufacture, etc
- More flexible cooperation ways are being adopted for more flexible operation, The relationship among enterprises become more flexible and dependent, for example, flexible supply chain system, etc

In order to adapt these situations and development tendency, enterprises have to fact more challenge and devote themselves to improve operation flexibility, production efficiency, rapid market response and process performance in order to match these changes from outside and inside. Naturally, the higher requirement to manipulate and control industry and business processes is desired by enterprises now, for example,

- Higher performance is needed to run, operate and measure
- Higher flexibility and rapid response
- Effective and easy control operation and analysis method
- Better making decision
- Better risk analysis and control, etc…

All these requirements and challenges today are direct drive power to improve process performance and specification. It is also the motivation of this research works in the paper.

1.2 Modeling for Process Analysis and Control

1.2.1 Process or System Models

From system theory point of view, model is a basic tool in analysis and control of modern process [E. Ikonen, K. Najim, 2002]. Explicit models are required by many of the system description and control methods, or models are required during control design and process analysis and operation. In the control of non-linear processes, the role of models is even more emphasized. In the model-based approaches, the controller can be seen as an algorithm operating on a model of process (subject to disturbances), and optimized in order to reach given control design objectives. Figure 1.2 gives a classification of model approaches [N. K. Sinha, M. M. Gupta, 1999].

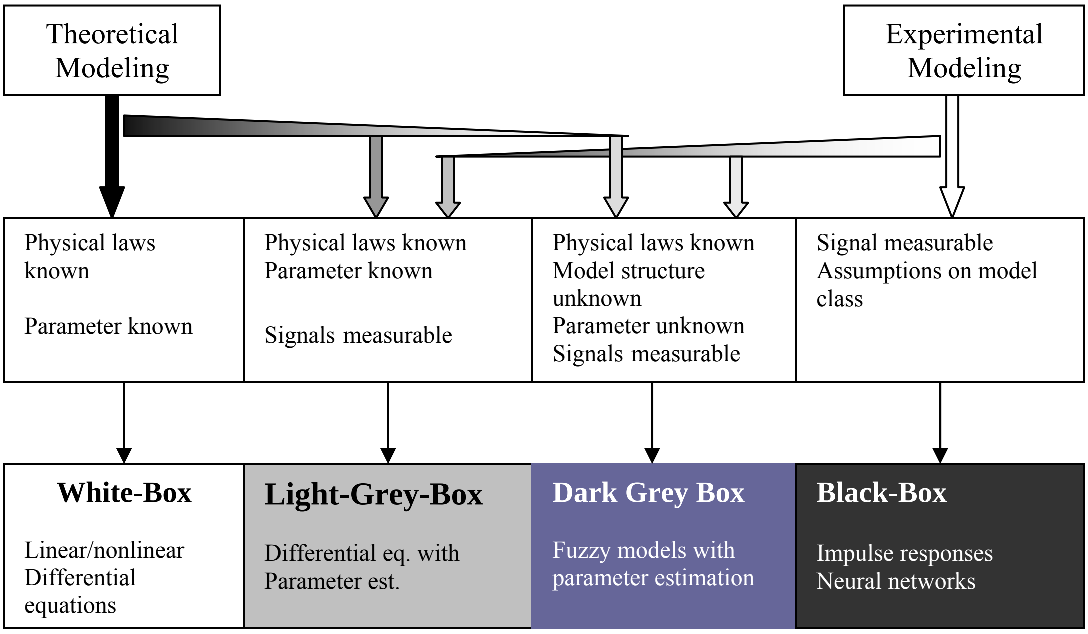


Figure 1.2: The classification of modeling approaches

In theoretical modeling, *first principles* are utilized to establish a system of algebraic and differential equations, the resulting process description is called *white box* equations, all parameters possess a physical meaning and are known prior. In contrast, in the case of experimental modeling, only a model class is assumed, both structure and parameters are determined from measurement data. Since the structure and the parameters cannot be interpreted physically, this is called as *black-box* modeling. The *grey-box* models represent a compromise between white-box and black-box models. They are based on information gathered from both first principles and measurement data [N. K. Sinha, M. M. Gupta, 1999].

Several approaches and techniques are available for deriving the desired process model. As usual, the standard modeling approaches including three main streams:

- The first-principle (white-box) approach and
- The identification of a parameterized black-box model.
- Human expert knowledge

As known, the available mechanistic knowledge of object and process data from empirical knowledge contribute the modeling together. Based on the different knowledge from real plant, different modeling method is applied. The relationship of them is shown as Figure 1.3 [J. Abonyi, 2002].

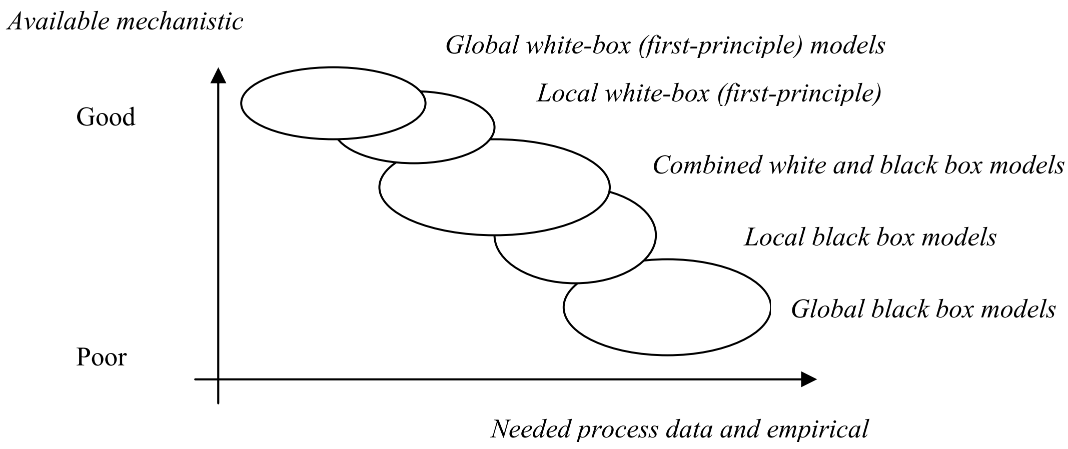


Figure 1.3: The relationship between mechanistic knowledge and data in modeling

The first-principle approach (white-box models) denotes models based on the physical laws and relationship that are supposed to govern the system behavior. In these models, the structure of model reflects all physical insight about the process, and all the variables and the parameters have direct physical interpretation. As we known, for complex non-linear process control, it is difficult to directly apply the first-principle modeling approaches due to the complexity of the process itself.

In such cases, variables characterizing the behavior of the considered system can be measured and used to construct a model. This procedure is usually called *system identification*. Identification governs many types of methods. The models used in identification are referred to as black-box models (or experimental models), since the parameters are obtained through identification from experimental data.

Between the two extremes of white-box and black box models lay the semi physical gray-box models. They utilize physical insight about the underlying process, but not to the extent that a formal first-principle model is constructed.

In modeling, the choice of both the model structure and the associated parameter estimation techniques are constrained by the *function approximation* and *interpolation capabilities* (e.g., linear approximations, smoothness of non-linearities, a priori information). Clearly, the choice of modeling method is of essential importance, and therefore a large part of this thesis.

In this thesis, an integrated intelligent model and its application architecture is designed and developed for process diagnosis, prediction and control based on the black-box models with technologies of Computation Intelligence (CI) and System.

Modeling always involves approximations since all real systems are, to some extent, non-linear, time varying, and distributed. Thus it is improbable that any set of models will contain the "true" system structure. All models just provide an acceptable level of approximation. The model in the thesis is developed based on higher characteristic extraction from real process. Some common, key features and attributes are extracted from real objects and some special feature in real processes are ignored and discarded in order to solve some common problems.

1.2.2 Model-based Fault Diagnosis and FDI

As many technological systems become more complex, widespread and integrated, the effects of system failure can simply devastating to the infrastructure of modern society. Feedback control is just one important component of total system supervision. Fault Detection and Isolation (FDI) is a second component with extensive commercial, industrial and societal implications if only we could work out how to do it in a reliable and inexpensive manner.

Most contributions in fault diagnosis reply on the analytical redundancy principle, Hardware and physical redundancy is the usual solution of the practical FDI problem. The basic idea consists of using an accurate model of the system to simulate the real process behavior, if a fault occurs, the *residual signal* (i,e. the difference between real system and model behavior) can be used to diagnose and isolate the malfunction [S. Simani, C. Fantuzzi, 2002]. Many traditional methods for FDI in real industrial processes or systems today are researched and developed based on physical model, hence, to this kinds methods, building dynamic model based on inner physical law is crucial and inevitable first step.

There is an increasing interest in theory and applications of model-based fault detection and diagnosis methods, because of economical and safety related matters. In particular, well-established theoretical development can be seen in many contributions published in the IFAC (International Federation of Automatic Control) Congresses and IFAC Symposium SAFEPROCESS (fault Detection, Supervision and Safety of Technical Processes).

In FDI system, the key core is to build system model in term of inner physical law, some analysis methods have been investigated for FDI modeling. To linear system, the good performance of the modeling has been reached by some control system model and methods, for example, state space model and robust observer design method. More investigations concentrate on modeling for complex and nonlinear plant, which is difficult to be described with traditional approaches of linear system. Under the circumstances, some modeling approaches based on Neural Network and Fuzzy Logic System have been more and more involved in and some system identification methods have also been investigated in term of black box principles.

1.3 Intelligence Technology and Computational intelligence (CI)

One kind of definitions for the intelligence system is given in [N. K. Sinha, 1999]. *The intelligence system can attempt to emulate important characteristic of human intelligence, these characteristics include adaptation and learning, inferring, planning under large uncertainty, and coping with large amount data, etc.*

During the past fifty years, intelligence technology has been developed and expanded with an amazing rate [Azvine, B., Azarmi, N. Nauck, D. D, 2000] [Zilouchian, Ali, Jamshidi Mo, 2001]. Its applications have pervaded into almost every aspect of scientific, industrial, business, military and many other areas. The main applications are divided into the following several categories:

- Modeling: including intelligent control, robot control, process control and identification, modeling nonlinear system and process, chemical structures, signal process and compression, dynamic system identification, dynamic and adaptive system control and so on.
- Pattern Recognition and Classification: including customer and market profile, loan risk/business state, product quality, signature recognition, image recognition, property valuation, medical categorization, population types and spectra analysis etc.
- Prediction: including market preference, economic indicators, future sales, production requirement, energy requirements, and chaotic time based variables and so on.
- Novelty detection: including performance or fault monitoring, fraud detection, identifying rare or different case and so on.
- Knowledge discovery and data process, representation and extraction from complex data and process, etc.

In essence, intelligence technology focuses on researching intelligence algorithm and units model that possess intelligent characteristics. Kernel algorithm focuses on improving the basic capability of basic intelligence element and units. Model dedicates to effectively combine the basic units so as to solve some practical problems.

Intelligence technologies includes many different technology branches, *Computational intelligence* (CI) provides the basic ability modeling complex processes based on the historical data. Fuzzy Logic (FL) in CI is a tool for embedding existing structured human knowledge into mathematical models. Neural Networks (NNs) in CI recovers underlying dependencies between the given inputs and outputs by using training data sets. After training, NNs represent high dimensional nonlinear functions, which can approximately express real relationship of system inputs-outputs. They are mathematical models obtained in an experimental way, namely, learning from historical data. Genetic Algorithms (GA) provides an optimization method based on random search way in optimization space. All of the advantages of CI provide strong tools and application to model complex system just based on historical data.

Because of various shortcomings of both neural networks and fuzzy logic models and the advantages of combining them with other technologies, integrated and modular solutions are becoming popular today. In addition, complex problems in real world also required more complex solutions but a signal network (or one-sided approach) can provide.

1.4 Why Intelligence Technology in Industry and business processes

Most of industry and business processes are complex processes. They are typical multivariable input-output (MIMO) system. First, the input variables take some effects to the system behavior. The second, there are also many parameters inside processes or systems to effect process output results. In order to implement control to industry and business processes, the first step is modeling so as to express and describe the real process and then based on the model, control objective and operations can be realized.

However, to some complicated process plants, it is not easy to obtain exact mathematical models to describe these real processes due to their complexity.

Most of industry and business processes are nonlinear processes. Nonlinear means two things here, first, the model class will not be restricted to linear input-output maps, and second, the dependence of the cost function that measures the goodness will be nonlinear with respect to the unknown parameters. There are many nonlinear units or elements in industry processes. The characteristics of nonlinear process make it very difficult to describe and control real process, it is also difficult to build model to measure and express real process with traditional mathematical methods. However, NNs and FL can describe nonlinear processes well due to their good nonlinear approximation, classification, interpolation and even inference abilities.

Most of industry and business processes are dynamic time-variant processes. It means system parameters of processes should be changed with time. This requires the model adaptive and self-learning as well as good robustness. It also need that the models can self-learn from input data or environment in order to dynamically adapt all changes from the outside of the model. System identification and optimization play two important and key roles to reach the goal. Another approach is from NNs and FL due to their strong robustness, noise-tolerance and good self-learning abilities.

Most of Industry and business processes are typical close loop system. It means the behavior or output of processes are not only affected by many factors inside system and real system inputs, but also by historical behaviors of real plant. The output in current time is often regarded as feedback data to adjust the process performance.

Most of industry and business processes are hybrid of symbolic and numerical processes. One aspect, In the process, the data collected from real time process can be presented numeric, for example, in industrial processes, the temperature signal, input energy, pressure signal, speed, flow, current, voltage, etc, in business processes, the price, change rate, sale amount, product market scale, etc. All of these data are numeric and their values are certain and clear. On the other hand, it is inevitable to invite some symbolic parameters in these processes, for example, the criteria of performance evaluation, risk control standard, quality evaluation, etc, for process analysis and control. This makes CI to play an important role due to their good abilities dealing with symbolic, vague or fuzzy variables.

All these are big challenge if process model is based on precious mathematical model and methods with white-box principles. By contrast with mathematical model approaches, intelligence technology expresses strong perspective to treat these kinds of problems based on black box principles. In fact, today, the Computational Intelligence (CI) has been widely applied in industry and business processes for modeling, system identification and optimization, as well as control. This is the background of technology and application that the research work is developed in the thesis.

1.5 Research Reports on FDI Application and Trend

There is an increasing interest in theory and applications of model-based fault detection and fault diagnosis methods. Many publications and increasing number of applications (IFAC Congress and IFAC Symposia SAFEPROCESS) between 1991-2002 show some interesting trends [S. Simani, C. Fantuzzi and R. J. Patton, 2002]. Table 1.1 gives all contributions taking into account the application and the type of faults considered are distinguished according to Table 1.2, among all contributions, the fault detection and fault classification methods are classified as in Table 1.3. The change detection and fault classification methods are indicated in Table 1.4 and the reasoning strategies are summarized in Table 1.5. The contributions are summarized in Table 1.6 [S. Simani, C. Fantuzzi and R. J. Patton, 2002].

Table 1.1: FDI application and number of contributions

Application	Number of contribution
Simulation of real processes	55
Large-scale pilot processes	44
Small-laboratory processes	18
Full-scale industrial processes	48

Table 1.2: Fault type and number of contribution

Fault Type	Number of contribution
Sensor faults	69
Actuator faults	51
Process Faults	83
Control loop or controller faults	8

Table 1.3: FDI method and number of contribution

FDI method	Number of contribution
Observer	53
Parity space	14
Parameter estimation	51
Frequency spectral analysis	7
Neural Networks	9

Table 1.4: Residual evaluation methods and number of contribution

Residual evaluation methods	Number of contribution
Neural Networks	19
Fuzzy Logic	5
Bayes Classification	4
Hypothesis testing	8

Table 1.5: Reasoning strategies and number of contribution

Reasoning strategies	Number of contribution
Rule based	10
Sign directed graph	3
Fault symptom tree	2
Fuzzy logic	6

Table 1.6: Application of model-based fault detection

FDI Application	Number of contribution
Milling and grinding processes	41
Power plants and thermal process	46
Fluid dynamic processes	17

Combustion engine and turbines	36
Automotive	8
Inverted pendulum	33
Miscellaneous	42
DC motors	61
Stirred tank reactor	27
Navigation system	25
Nuclear process	10

Table 1.6 shows that DC motor applications are mostly investigated among mechanical and electrical processes. Table 1.3 shows that parameters estimation and observer-based methods are used in nearly 70% of all application considered. Neural networks, parity space and combined methods are significantly less often applied.

Among all the described processes, linear models have been used much more than non-linear ones. On processes with non-linear models, observer-based methods are mostly applied, but parity equations and neural networks also play an important role.

The use of neural networks and combinations seems to be increasing. Concerning the fault diagnosis methods, in recent years, the field of classification approaches, especially with neural networks and fuzzy logic has steadily been growing. Also, rule-based reasoning methods are increasingly being based on fault diagnosis. Applications using neural networks for classification are increasing and the trends are analogous to the increasing number of non-linear process investigations. Nevertheless, the classification of generated residuals seems to remain the most important application area for neural networks.

1.6 Motivations and Objects of This Thesis

The initial objective of this study is to investigate how to develop a good model in order to express and analyze a complex processes with intelligent technology. The main goals of modeling for complex process in the paper are listed as follows:

- Modeling for process state detection (abnormal state or fault state) and fault diagnosis
- Process behavior prediction and
- Process control

From system and control point of view, when a fault occurs, the system behaviors in fault states should be different with that in normal states. The different behaviors should be exposed via all relevant data and related inner dependence. These characteristics of process behaviors or states are embedded in these process data if there is a fault occurrence. From intelligence technology point of view, these process data in abnormal (fault) states could be in different distribution space in multi-dimension space with normal process data. Hence, identifying the different states (fault state and normal state) in multi-dimension space based on their process data is the first step. If the fault states can be correctly identified, the characteristics in different states can then be investigated and analyzed by their corresponding state data. It is possible to make fault diagnosis, system identification and behavior prediction based on these data.

Based on above principles, the model in the thesis is not only used for fault detection and diagnosis, but also for process state identification, behavior prediction and process control. Some valuable knowledge and rules about the process state change (fault detection) can be extracted and analyzed in term of different process states and their corresponding data.

Until now, many researches and applications with intelligence technologies in engineering and business processes have been done. To intelligence technologies, there are three different research directions covering from theory to application as follows:

- Kernel and basic algorithm research: It offers basic algorithms and kernel to form intelligence unit and kernel
- Model unit and algorithm research, for example, different Neural network units and their learning algorithm, like BP, MLP, etc
- Application research: application of model and algorithm for practical science, engineering and business problems

The thesis focuses on application model for process state (fault) diagnosis, process output prediction and control with existing algorithms and unit models in computational intelligence, data mining technologies and system and control as well. It concentrates on solving some practical problems but develops new mathematical algorithms, kernel models and theories. Almost work in the thesis faces to practical application in production processes.

In this thesis, three important aspects are investigated and discussed as follows:

- Modeling for complex processes for process state (fault) diagnosis, prediction and control
  - The model is built according to some intelligence units and strategies
  - The model can identify the process state and detect fault, especially in aberrant state and behaviors
  - The model can extract the rules and knowledge about process state change for process analysis and diagnosis
  - The model can provide good behavior prediction for real process
  - The model possesses adaptability to cover time-varying processes
  - The model can be derived as a control model for process control
- Applying the model and methodologies to solve practical problem
  - The model is applied in supply chain process for product supply forecasting and process risk control.
- Analyzing and evaluating the performance of the models
  - Analyzing model performance and
  - Comparison performance with other model and methodologies

Some potential applications of this model in industry and business processes are listed as follows:

- Process performance definition, measurement and evaluation for complex Process, for example, Supply chain performance measurement, risk measurement and so on
- Process behaviors prediction and risk analysis
- Process state change detection, for example, aberrant state and aberrant behaviors detection

- The underlying reason and rule analysis for process diagnosis, for example, aberrant behavior and failure analysis

In addition, our knowledge about some industrial and business processes is very limited. The model is built based on some common characteristics and requirements extracted from industry and business processes. It provides a basic architecture for solving some common problems. It can not provide full solutions for all problems of them.

1.7 Outline of the Thesis

The remained parts of this thesis are classified into seven chapters. Chapter 2 first starts to introduce the some basic concepts and characteristics about monitoring and diagnosis in industrial processes. As mentioned in the section, the almost industrial processes are based on time series. Hence, the data distribution based on the time series is tightly linked with process characteristics. In fact, the data distribution with time series in industry and business processes is just the outside representation of system's inner law. From the data, process model can be built and process diagnosis and analysis can be implemented. In the later section, the model-based fault diagnosis technique is introduced from control and system point of view. Some basic problems based on these techniques are also given. This is why the integrated intelligent model is developed for fault detection and diagnosis problems. In the second section, some analysis and discussions of the fundamental elements for industrial processes is done. Some basic characteristics in industrial processes are extracted and abstracted as common characteristics so that the model built on these characteristics has common attributes at the scale. These basic characteristics can be regarded as premises for developing the integrated intelligent model in the thesis. Finally, some basic concepts and terms concerning process analysis are defined for model development.

In chapter 3, the fundamental intelligence technologies and methods, for example, Soft Computing, Neural Network and Fuzzy Logic Model are briefly introduced first. The data mining technologies, which are employed in the paper too, are also described here. These technologies played important roles in the intelligent model development for process state (fault) diagnosis, prediction and control. Based on these basic intelligent technologies and real process characteristics in real world, an outline of integrated intelligent model is built and depicted. The basic architecture of the integrated intelligent model and each unit functions are introduced as well. In essence, the integrated intelligent model for process state (fault) detection, diagnosis and prediction complies with Fuzzy NRAX TS dynamic model. In the NARX Fuzzy TS dynamic model, the *Antecedent part*, which is generated by two fuzzy classifiers**,** constructs the residual generator for fault detection and the *Consequence part*, which is generated by system identification based module Neural Networks structure and fuzzy partition information, constructs the process output prediction. These inference rules based on fuzzy TS model describe the relationship between fault occurrence and its quantitative value of process output. In essence, the whole model is consisted of two Fuzzy TS inference system, one is used to process state (fault) problems produced by real system input, another is produced by real historical behaviors. The combination of there two fuzzy TS system constructs a Fuzzy NARX TS dynamic inference system. In the last section, the implementation technologies and methods for the model are illustrated in detail step by step.

Applying the optimization and adaptive method for integrated intelligent model is main goal in the chapter 4. In the chapter, a model optimization scheme is first introduced. It is used to implement optimal model in order to make the model fit real process or plant as more as possible. In the next section, two different model optimization schemes have been investigated for different purposes to cover time-varying process. One is for process diagnosis and another is for process prediction. Based on different model optimization schemes, different optimization methods are applied. In the thesis, nonlinear optimization and GA are used to optimize model key parameters, which are a threshold for process state classification and sensibility, so as to provide optimal process diagnosis. Least Square (LS) approaches are employed to describe and estimate optimization parameter so as to provide optimal process prediction. Hence, the integrated intelligent model possesses adaptive threshold for different process state classification as well as provide optimal process prediction.

In chapter 5, a state space equation is derived from the integrated intelligent model. It can be regarded as a discrete time-varying system model expressed by a specific state space equation, whose state variable is designed by process states in real process. In the state space equation, the state transition matrix A is determined by the fuzzy degree of process state classification produced by process historical behavior in time t instant, and input transition matrix B by the fuzzy degree of process state classification produced by process real input in time t instant, and state observer vector H is determined by optimization result generated by optimal model or model adaptive scheme. Based on this kind of state space equation, optimal process control method is applied for process control.

A real case in supply chain process for Norsk Kjøttsamvirke AS Norway is implemented based on the integrated intelligent model in chapter 6. The prediction result proves the integrated intelligent models can provide good process diagnosis and prediction ability for this kind of problems in supply chain, especially in behaviors prediction of business process, process diagnosis and risk identification, etc. The first, a problem of supply chain process is introduced and then supply chain process and relevant data are analyzed. It indicates the problem of supply chain process can be regarded as process state detection and process behavior prediction problem in essence. The further work about implementation of intelligence model for the supply chain process is given and the detail implementation process and results are also done step by step based on the intelligence model architecture. The final goal of the supply chain problems is reached with high prediction accuracy and certain decision-making support. The business process could benefit from all of these results for goods supply forecasting, risk identification, and process analysis, etc. These results prove the integrated intelligent model has good capability for the business process improvement in real world.

Finally, Chapter 7 presents the concluding words and indications for further research and applications in the area.

CHAPTER 2 INDUSTRIAL PROCESS AND PROCESS DIAGNOSIS TECHNOLOGY

2.1 Industrial Processes

It is difficult to give a clear and standard definition for industrial processes. One definition is given as: An industrial process is a series of operations performed in manufacturing or production or some other industrial activities. It can be regarded as the complex system and a set of logically related activities, operations and equipments performed to achieve a defined industry outcome [K. Evanglos, W. Antoni, 2000]. In an industrial process, many factors are involved in and they can be described as:

Material flow: all materials in industrial processes and their flow conveyance, including raw material, operator, equipment and so on.

Power flow: all needed power and flow conveyance and form, including drive power, electrical power, and human resource and so on.

Data flow: all data generated during the industrial processes and their flow conveyance, including planning data, product data, production and process monitoring data and so on.

Information flow: all information related to industrial processes, for example, instruction, plan, knowledge and rules etc.

Value flow: all value generated during the industrial process and their flow conveyance, for example, product schedule, quality management, control and quality criteria, and process improvement and so on.

All of them in an industrial process can be shown as Figure 2.1.

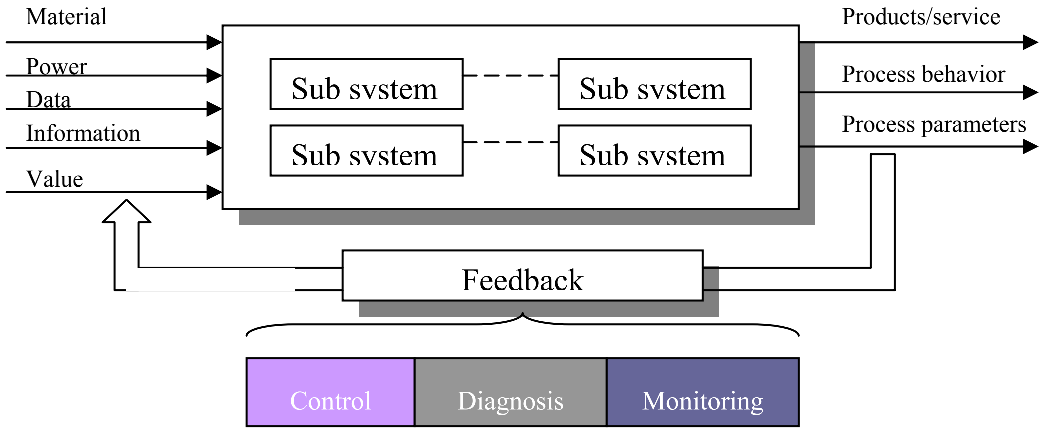


Figure 2.1: The basic elements and briefly description for industry processes

An industrial process consists of many different sub systems or processes, which provide different functions and play different roles respectively. Briefly speaking, an industrial production process covers from machine and equipment system, manufacturing system, and then production system, it should be extended to some management processes, for example, Total Quality Management (TQM), Logistic

System, Enterprise Resource Plan (ERP) and Supply Chain Management System (SCM), Customs Relation Management (CRM) and so on, the simple and rough relationship among these sub systems in industrial processes is shown as following Figure 2.2.

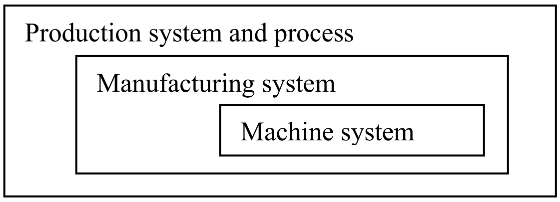


Figure 2.2: The relationship among sub systems related to industrial processes

2.1.1 *Monitoring and Diagnosis in Industrial Processes*

In an industrial process, it is not easy to evaluate if the industrial process is operating well or the operation state is well if one do not rely on any ways. The sole method, which is also popular method in industrial process today, is to monitor the whole industrial process, at least, monitor some key sections. Monitoring and diagnosis is a kind technique of sensing the process or equipment health and operating information and then analyzing this information to quantify (or condition) of industrial equipment or process, so that potential problems can be detected and/or diagnosed in their development, and corrected by suitable recovery measures, before they become severe enough to cause plant breakdown and other serious consequences [K. Evanglos, W. Antoni, 2000].

Monitoring plays an extremely important role in industrial processes. One aspect, the system operation or process state can be got or analyzed through data analysis from monitoring system, another aspect, the industrial process behavior can be adjusted correctly according to the correct process state and performance evaluation. Therefore, monitoring industrial processes can provide important criterion for evaluating performance of industrial processes, system or equipment and also provide correct control and operation strategies.

Monitoring is used in time dependent industrial processes in order to ensure that the process is effective. It is particularly important in aligning the process with other processes and ensuring that the process operates according to the desired specifications. At meantime, the performance of the industrial processes also can be evaluated through the result data from monitoring system. Therefore, a monitoring and diagnosis system generally involves the design and utilization of sensing arrangements on industrial plant, together with data acquisition and analysis system, plus predictive and diagnostic methods, with the objective of ensuring safety, risk, high efficiency and availability of the equipment as well as maintenance in a systematic way using various monitoring an diagnosis knowledge and techniques [K. Evanglos, W. Antoni, 2000]. Hence, the monitoring and diagnosis schemes are already an important role in modern production processes. Some researches and practical applications of monitoring have been done recently [Ramdén, T., 1995][Mathew, J. ,Alfredson, R. J., 1983][Koizumi, T. and Kisv, M., 1986] [Hogan, P. A., et al., 1991].

2.1.2 Model-based Fault Diagnosis Techniques

Because of economical and safety related matters, there is an increasing interest in theory and application of model-based fault detection and diagnosis methods, which is developed based on model of system physical laws and some system analytical methods.

The developments began in the early 1970s, the accomplishment of that time was the well-known “failure detection filter” approach for linear systems by Beard [Beard, 1971] and Jones.

Willsky [Willsky, A.S., 1976] gave a summary of this early development, the first book on model-based on methods for fault detection and diagnosis with specific application to chemical processes was published by Himmelblau [Himmelblau, D. M., 1978]. In further research work in the field, parameters estimation techniques for fault detection of system was proposed and demonstrated [Hohmann, H.,1977] [Bakiotis, C. al., 1979] [Geiger, G.,1982] etc,. The parameter and state estimation was then summarized by Isermann [Isermann, R.,1984] and [Isermann, R., 1997].

The further development fault detection and isolation methods to the present time is summarized in the books of Pau [Pau, L. F., 1981], then Patton [Patton R., J., al., 2000], Basseville and Nikiforov [Basseville and Nikiforov, 1993], Chen and Patton [Chen and Patton, 1999], Gertler [Gertler, J.,1998] etc.

These developed theory and methods have been widely applied in modern industrial system and processes, for example, in chemical production processes, Petrochemical processes, complex machine system, gas turbine machine, offshore platform and so on[R. Diversi, , S. Simani, , U. Soverini, 2002][R. J. Patton,S. Simani, S. Daley, 2001][R.J. Patton, C.T. Lopez-Toribio, S. Simani, 2001][Frank, P.M 1993]. Some researches in theory aspect also have been done [Patton, R. J., Frank, al, 1989, 2000][Gertler, J, 1998][Mitchell, J. S., 1981].

In the section, a brief introduction concerning model based fault detection and isolation techniques in modern industrial processes is given. Some related questions and technologies and trends are emerged so as to provide basic knowledge for model development with intelligence technologies in the thesis.

1, Model-based fault diagnosis in dynamic system using identification techniques

A traditional approach to fault diagnosis in the wider application context is based on hardware or physical redundancy methods, which use multiple sensors, actuators, component to measure and control particular variables. The diagnosis scheme can be built based on comparison and analysis of these collected process data and then evaluates if the process performance that measured by monitoring system satisfies process operation specification. Hence, developing some methods for fault detection based on system laws or process data is main research and application direction for process diagnosis in modern industrial processes.

Model-based fault detection method uses ***residuals signal***, which indicates changes between the real process and the process model. A general principle of the model-based fault detection is shown as following Figure 2.3 [S. Simani, C. Fantuzzi and R. J. Patton, 2002]. It also indicates the essential problem in model-based fault detection and isolation (FDI) is to generate a good residual model describing the behavior of the monitored system. Some different methods based on dynamic physical model to generate residual generator, for example, state space model and output observers, parity relations and parameters estimation methods, have been investigated in recent research.

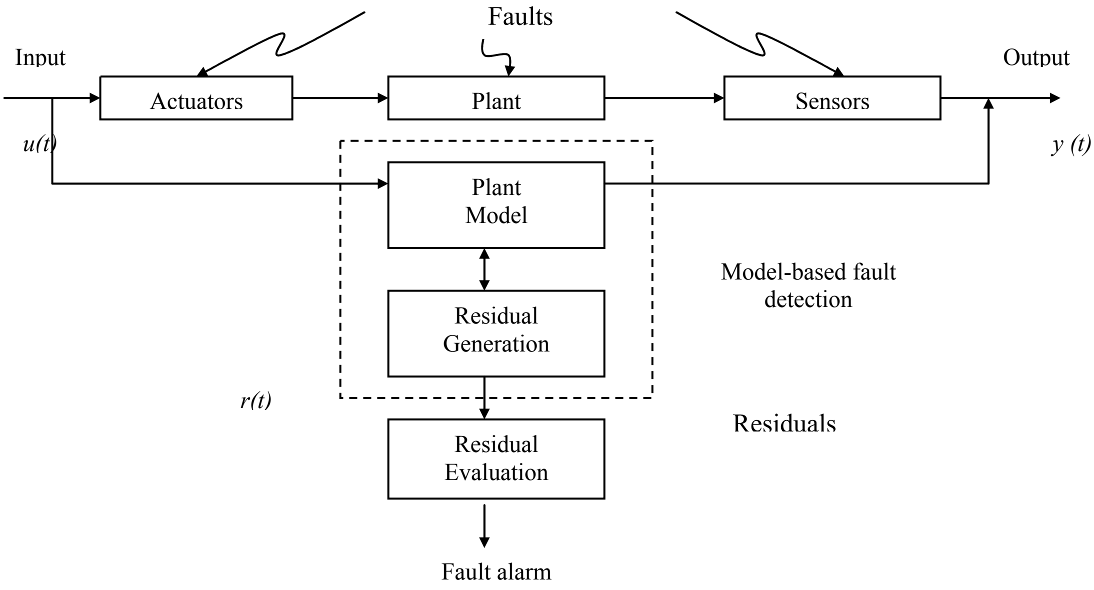


Figure 2.3: Scheme for the model-based fault detection

In the figure, *u(t)* is system input variables, *y(t)* is system output, *r(t)* is the residual signal, which is used to compare the difference between real process and system model. The model-based approach (analytical redundancy) is a widely accepted modern approach for fault detection and isolation (FDI). It is based on the idea that the measurements from dissimilar sensors are functionally related. Any violation of these relationships indicates the occurrence of faults [S. Simani, C. Fantuzzi and R J. Patton, 2002].

As the above shown, the model-based FDI consists of two stages, residual generation and decision-making, which are introduced as follows:

- **Residual generation**, in which the inputs and outputs of systems are monitored and manipulated to generate a signal or vector, the so-called "residual". The residual should indicate if a fault has occurred. It should normally be zero or close to zero under no fault condition, meantime, distinguishably different from zero when a fault occurs. It means that the residual is characteristically independent of process inputs and outputs in ideal condition.
- **Residual evaluation and decision-making**, for analysing the residual to examine the likelihood of faults. A decision is then made based on the knowledge about the process and the symptoms.

If only output signals *y(t)* can be measured, signal model-based methods can still be applied, e,g, vibration signal, which is related to rotating machinery or electrical circuits. Type signal model-based methods of fault detection are listed below:

- Bandpass filter
- Spectral analysis (FFT)

- Maximum-entropy estimations

The characteristic quantities or features from fault detection methods show stochastic behavior with mean value and variances. Deviations from the normal behavior must then be detected by methods of *change detection* (residual analysis) like:

- Mean and variance estimation
- Likelihood-ratio test, Bayes decision
- Run-sum test, etc

Model of faulty system

To Multi-Input and Multi Output (MIMO) dynamic system, the basic scheme of fault diagnosis in a closed-loop system is given in Figure 2.4 [S. Simani, C. Fantuzzi, Ron J. Patton, 2002].

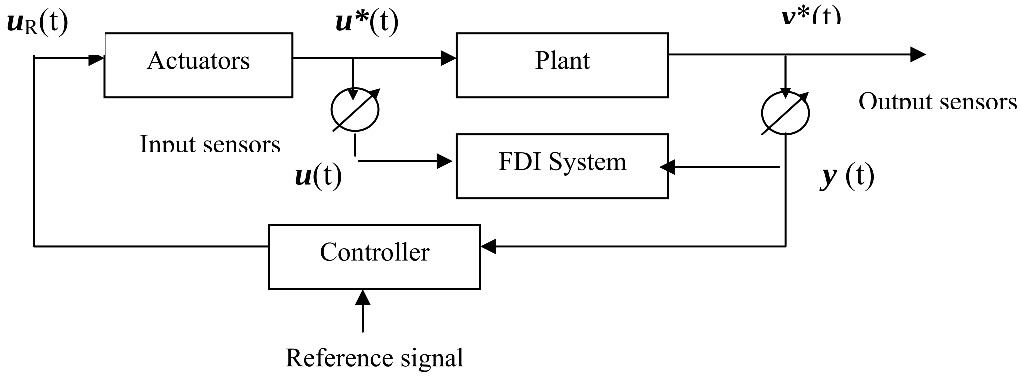


Figure 2.4: The basic scheme of fault diagnosis and isolation in control system

The main components are the *Plant* under investigation, the *Actuators* and *Sensors*, which can be further sub-divided as input and output sensors and finally the Controller.

The system working condition will be monitored by means of its input *u(t),* output *y(t)* measurements and signals from the controller $u_R(t)$, which are supposed completely available for FDI purposes. Once the actual process input and outputs *u*(t) and y*(t)* (usually not available) are measured by the input and output sensors, FDI theory can be treated as an observation problem of *u(t)* and output *y(t)*.

Mathematical expression

Based on general case, the system affected by all possible faults can be described by the state-space model [S. Simani, C. Fantuzzi, 2002]. They can be described in time domain equation and frequency domain equation as below respectively:

Time domain equation

$$\begin{cases} x(t+1) = Ax(t) + Bu^*(t) + L_1 f(t) \\ y(t) = Cx(t) + L_2 f(t) \\ u(t) = u^*(t) + L_3 f(t) \end{cases} \tag{2.1}$$

Where, *x(t):* state vector; *y(t)*: System output (measurement value); *u(t)* system input (measurement value); *u*(t);* plant input value, vector *f(t)=* $\left[f_a^T, f_U^T, f_C^T, f_y^T\right]$ correspond to specific faults.

Based on above state-space expression for the fault diagnosis system, the frequency domain equation can be drawn as following [Chen and Patton, 1999, Getler, 1998]:

$$y(z) = G_{yu*}(z)u^*(z) + G_{yf}(z)f(z) \tag{2.2}$$

The transfer matrices $G_{yu*}(z)$ and $G_{yf}(z)$ are defined as:

$$\begin{cases} G_{yu*}(z) = C(zI - A)^{-1}B \\ Gyf(z) = C(zI - A)^{-1}L_1 + L_2 \end{cases} \tag{2.3}$$

The state-space description provides general and mathematically rigorous tools for system modeling and robust residual generation. For both deterministic (noise free measurement) and the stochastic case (measurement affected by noises), the system matrix A, B and C, can be obtained by system identification [S. Simani, C. Fantuzzi, 2001].

The technology for residual generation

As mentioned above, the most key task in FDI is to design a reliable residual generator based on above mathematic model. The general FDI system model can be illustrated as Figure 2.5 based on system frequency domain equation (2.2) [Patton, RJ and Chen, J, 1991].

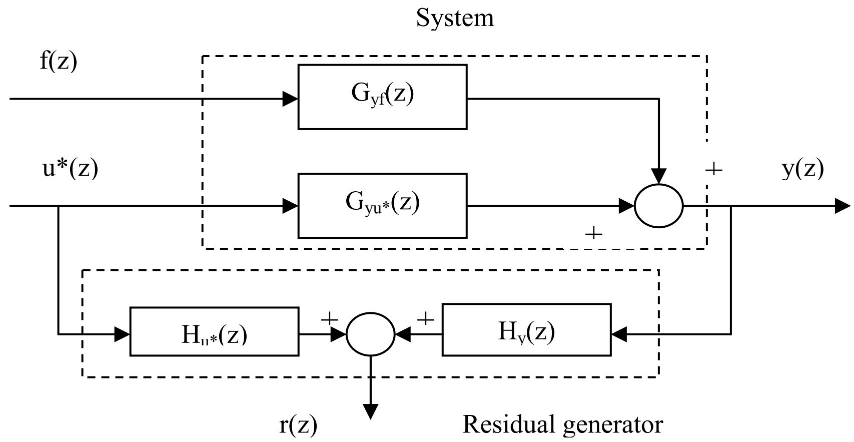


Figure 2.5: Residual generator structures

Hence, the residual generator structure is expressed mathematically by the generalized representation.

$$r(t) = \begin{bmatrix} H_{u*}(z) & H_y(z) \end{bmatrix} \begin{bmatrix} u^*(z) \\ y(z) \end{bmatrix} = H_{u*}(z)u*(z) + H_y(z)y(z) \tag{2.4}$$

where, *$H_{u*}(z)$ and $H_y(z)$* are discrete transfer matrices which can be designed using stable discrete-time linear systems. The functions *u*(z),y(z), r(z)* and *f(z)* are the Z-transform of the corresponding discrete-time signals.

According to the definition, the residual *r(t)* has to be designed to satisfy the zero for fault-free case and different form zero if the fault occurred, namely,

r(t)=0 if and only of *f(t)=0*

A possible output signal from residual generator is given in Figure 2.6, which is generated by the research of fault detection for gas turbine [S.Simani, C.Fantuzzi, 2001]. When the residual signal was not zero, it indicates a fault occurs.

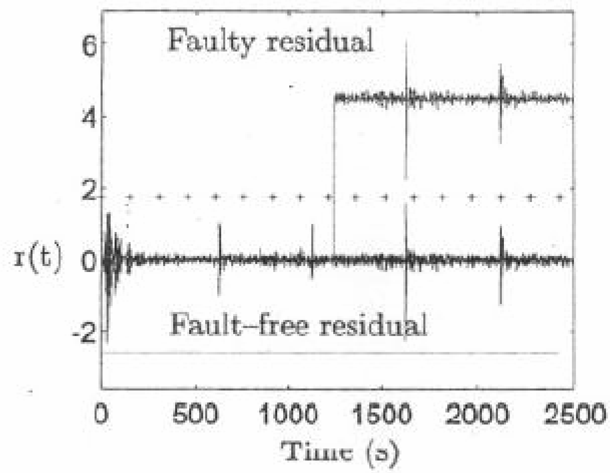


Figure 2.6: An example of residual signal in fault detection for gas turbine

Hence, in order to satisfy the above criteria, the design of the transfer matrices *Hu*(z)* and *Hy(z)* must satisfy to the constraint conditions.

$$H_{u*}(z) + H_y(z)G_{yu*} = 0 \qquad (2.5)$$

In essence, when above equation (2.5) cannot be satisfied when new signal input the system; it means there is a fault occurs. *It also indicates the symptom or residual signal depend on faults and are independent of system operation states*.

It is worth noting that the success of fault diagnosis depends on the quality of the residuals. Successful diagnosis requires good residual signals, which should be robust with respect to modeling uncertainty. Several approaches are involved in residual generation as following:

- Observer-based approaches [Frank P.M, 1993, Basseville, M, 1998]
- Parameter estimation [Isermann, R., 1984, Patton R.J, al.,2000][Fantuzzi,-C.; Simani,-S, 2001][P. Eykhoff, 1977][MS Rajbman, VM Chadeer 1980]
- Parity vector (relation) Methods [Chow and Willsky, 1984, Delmaire al., 1999] [Patton, RJ. and Chen, J,1991]
- Directional residual
- Statistical approach [Denison D.G.T, Holmes C.C, 2002]
- Qualitative approach

The method of fault identification

In FDI problems, if symptoms change differently for certain faults, a first way of determining them is to use *classification* methods, which indicate changes of symptom vectors.

Some classification methods are developed as follows:

- Geometrical distance and probabilistic methods, for Bayes classification [Denison D.G.T, Holmes C.C, 2002], etc
- Artificial Neural Network[Karpenko M, Sepehri N, 2002][Chen YM, Lee ML 2002][Calado JMF, Korbicz J,2001], etc
- Fuzzy and Fuzzy Clustering method [Applebaum, 2001]
- Support Vector Machine (SVM)…

When more information about the relations between symptoms and faults is available in the form of diagnostic models, reasoning method can be applied. The typical reasoning methods are given as:

- Probabilistic reasoning method
- Bayes inference system [Denison D.G.T, Holmes C.C,al, 2002]
- Fuzzy inference system [Takagi, H. and Hayashi, I., 1991]
- Possibilistic reasoning with Fuzzy logic
- Reasoning with artificial neural network…

It shows that many different methods have been developed during the last two decades, and it is also clear that many combinations of these methods are possible.

The residual generation problem

Although the analytical redundancy method for residual generation has recognized as an effective technique for detecting and isolating faults, it still exists the critical problems, which is unavoidable modeling uncertainty due to dynamic system, noise, big range of operation condition, etc.

The main problems regarding the reliability of FDI schemes is the modeling uncertainty, which is due to, for example, process noise, parameters variations and non-linearities, etc. the design of an effective and reliable FDI scheme for residual generation should take into account of the modeling uncertainty with respect to the sensitivity of the faults. Therefore, the task of the design of an FDI system is thus to generate residuals which are ***robust*** to modeling uncertainty, disturbance and process noise, etc. some new methods and technologies, for example, disturbance de-coupling approaches, Kalman filters, system identification approach, have been developed in detail.

The central task in model –based fault detection is the *residual generation*. Most residual generation techniques are based on linear models, for nonlinear systems, the traditional approach is to linearise the model around the system operating points. However, for systems with high non-linearity and a wide dynamic operating range, the linearised approach fails to give satisfactory results. It also indicates the model-based fault detection approach has good performance to linear and time-invariant system but poor performance to nonlinear system with wide operation range.

Based on the above problem and challenges, some approaches are investigated in order to improve the performance, for example,

- Adaptive fault detection thresholds

- Robust approaches to improve the sensitivity to fault but not noise or model uncertainty
- Statistical methods such as Likelihood Ratio Test(LRT), Markov Model or Bayesian-based probabilistic reasoning methods
- Soft computing approaches, including Fuzzy Logic and Neural Networks
- Combination of quantitative and qualitative methods together

At the same time, to nonlinear system with wide operation range, one solution is to use a large number of linearised models corresponding to a range of operation points. This means that a large number of FDI schemes corresponding to each operating point are needed. Hence, it is important to study residual module for treating nonlinear system. One kind of approaches is still investigated based on System model and another is to employ the neural network, fuzzy model and hybrid models.

2, Process (Fault) diagnosis in dynamic system using intelligent techniques

The main problem with model-based diagnosis technologies introduced above is that the precision of the process model affects the accuracy of the detection and isolation system as well as the diagnosis sensibility. At the meanwhile, the majority of real industrial processes is non-linear system and cannot be modeled by using a signal model for all operation condition, accurate modeling for a complex non-linear system is very difficult to achieve in practice. Sometimes, it is impossible to describe them by analytical equations.

In recent years, neural networks (NNs) have been used successfully in pattern recognition, system modeling as well as system identification, etc. They have been proposed as a possible technique for fault diagnosis and isolation, too. There are some reasons to employ NNs into fault diagnoses and detection field. Some applications have been done with intelligence techniques in different application fields [Karpenko M, Sepehri N, 2002] [Chen YM, Lee ML 2002] [Calado JMF, Korbicz J,2001] [Lee IS, Kim JT, 2003] [J.B.Gomm, 1998].

NNs do not need a deep insight into law of the processes. It is robust to noise data and has a good ability to generalize the relationship learnt to successfully diagnosis learning faults as well as new conditions.

NNs are a powerful tool of handling non-linear problems. One of the most important advantages is their ability to implement non-linear transformations for functional approximation model. It can be used to represent real system with any arbitrary degree of accuracy [K,Vojislav. 2001]. Hence, it is a good tool to tackle process (fault) diagnosis problems for the nonlinear system. NNs can handle nonlinear behaviors and partially known process they learn the diagnostic requirements by means of the information of the training data.

NNs are noise tolerant and their ability to generate the knowledge as well as known to approximate any non-linear even dynamic function by given suitable weight and architecture. Moreover, on-line training makes it possible to change the FDI system easily in case where changes are made in the physical process or control system [Hoskins, J.C and Himmelblau, D.M. 1988, Dietz, 1989].

Here, NNs have been successfully applied to fault diagnosis problems due to their capabilities to cope with *non-linearity*, *complexity*, and *uncertainty*, *noisy or corrupted* data [K,Vojislav. 2001]. It is a good modeling tool for highly non-linear processes. Generally, it is easier to develop a neural network based on model for a range of operating points than develop many linear models, each one for a particular operation point. At the same time, NNs are ideal tools for generating residuals due to their good nonlinear classification ability. The drawback of using neural networks for classification of faults is their lack of transparency.

Industrial plants often work at different operation points. However, as usually, the neural networks for fault detection and diagnosis consider only a single working point/condition or small changes of operation points. The standard scheme for the design neural network for fault diagnosis at all operating points may be impractical due to the unavailability of suitable training data for all working points and different conditions.

Because of these assumptions, the modeling method for real non-linear system with neural networks, fuzzy logic system seems to be a natural tool to handle complicated and uncertain condition, and also is applied more and more in real industrial process for process (fault) diagnosis and isolation.

3, Integrated model for fault detection and diagnosis

A typical integrated fault diagnosis system uses both analytical and heuristic knowledge of the monitored system. The knowledge can be processed in terms of residual generation (analytical knowledge) and feature extraction (heuristic knowledge). The process knowledge is then provided to inference mechanism, which can comprise residual evaluation, symptom observation and pattern recognition.

A typical scheme of integrated FDI model is: using the method-based system model for fault detection and using nonlinear classification method based intelligence technologies for fault identification. Some investigations have been done in the field recently, for example, A combination of method based physical model and based NNs is developed for fault diagnosis in power plant using neural networks [S. Simani, C. Fantuzzi 1998] due to advantages of each approaches. In this way, a Kalman filter was applied to generator residual generator for fault detection and NN models are applied for fault identification. These combinations of methods provide good solution for complex dynamic system in respect of fault detection and identification.

2.1.3 Problems and Challenge in Monitoring and Diagnosis in Industrial Processes

In the past years, many effective monitoring and diagnosis techniques and systems have been investigated and developed for the mechanical machinery monitoring and diagnosis as well as for complex processes, such as vibration monitoring, thermal monitoring and internal defect detection etc, these techniques mainly focuses on how to extract the pertinent signals or features from the equipment heath information. However, the related yet more important problem is the way of how to analyze and utilize this information.

Traditional methods tend to process and analyze these information by conventional computation methods, for example, system identification, signal process and analysis, probability and statistic methods, Bayesian classifier and inference method and model method, and so on. These conventional methods might work well under the circumstance of simple mechanical equipment or simple process with relatively simple working environment and operation point. But they will face unprecedented challenges coming from more and more complex and flexible equipments and processes today.

In modern industry, some new demands from system monitoring and diagnosis is needed, the traditional monitoring and diagnosis methods have to face to bigger and bigger challenge as follows:

- The most of traditional monitoring and diagnosis schemes today focuses on extracting process feature, data acquirement and measurement, simple data analysis or monitoring process state in real time on line, but, it lacks further data analysis and decision-making support for practical process diagnosis. It means more attention to monitoring, but less attention to diagnosis and prediction.
- The monitoring and diagnosis system is effective to simple mechanical equipments or processes with relatively simple or stationary working environment, but it is difficulty to apply in some dynamic, wide range operation and complex processes.
- In data analysis for system performance, most of traditional monitoring and diagnosis system just provides limited analysis capability, in fact, some most desired and valuable analysis is important to control and evaluate whole process and system, for example, process state detection, process behavior prediction, and abnormal behavior identification, decision-making support for fault detection and diagnosis, risk analysis and process performance measurement, etc.
- Most of monitoring systems collect numerical data from industry and business processes easily, it is also easy to deal with these numerical data to apply in diagnosis or data analysis today, but, some variables in real industrial processes, which is uncertain, vague and fuzzy, or not numerical data, are important factors to effect and evaluate the process performance and behaviors. It should be better and more accurate solution if these variables are involved in the process analysis and system diagnosis.

All problems mentioned above in industrial processes make big challenge to monitoring and diagnose system today. Based on these reasons, developing and designing new model and methods to solve these problems is naturally the hot topic in research field.

2.2 Common Characteristic Extraction and Abstraction for Modeling

According to practical industrial processes, some basic characteristics can be extracted and abstracted from system point of view.

2.2.1 Basic Characteristics of Industrial Processes

Generally, most of industrial processes are complicated processes in real world. Different processes have different process characteristics and their inner laws. In order to make the model available to solve some common problems in real industrial processes, it is necessary to further extract and abstract some common characteristics from real industrial processes in high-level sight for developing process/system model. These common characteristics are also the base and fundament for modeling for process (fault) diagnosis and prediction with intelligence techniques in this paper. These basic characteristics of real industrial processes are summarized as follows:

An amount of data and many activities are involved

In industrial and business processes, many activities and operations are involved in whole processes. Normally, it is difficult to build a mathematic model to describe the real processes from inner law point. At the same time, amount of data from processes can be got in order to monitor and evaluate process performance. Hence, process characteristics and performance can be investigated and analyzed by these sampled data on certain scale. It means that the data collected from processes is main way to reflect process properties.

Multivariable Systems

Industrial processes normally are Multi-Inputs or Multi-Output (MIMO) system. One aspect, these multivariable affect process performance and behaviors and these input variables have different influences on whole process behaviors, respectively. Another aspect, these input variables also affect each other and sometimes, it is difficult to determinate which input variable is the most important to the process performance. All of the characteristics make it difficult to analyze process performance through traditional control method and system theory.

Time-varying System

A major part of conventional system theory is based on the assumption that the systems have constant coefficients. This assumption of time-invariance is fundamental to conventional design procedures. In real life, however, most industrial processes exhibit time varying behaviors. The properties of the process and/or its signals change with time due to component wearing, changes in process instrumentation, updates in process equipment, failures, etc. When changes in the process are significant, the controller or system analysis model needs to be re-designed in order to maintain satisfactory behavior of the controlled process.

Complex Non-linear System

All industrial processes however, are inherently non-linear characteristics. Non-linearities may be due to constraints, saturations or hysteresis in the process variables. Non-linearities may also be presented during the normal operation of the process due to the non-linearity of the process phenomena, for example, transport phenomena, and so on. The complicacy of process or system makes the description of linear system limitation for these nonlinear systems.

2.2.2 Premise of Modeling

The basic common characteristics for industrial processes are described above from system point of view. Although there are some different characteristics in concrete industry and business processes, some similar and common characteristics can be extracted and abstracted from these different processes.

Based on above all characteristics, some further conditions and constraints for building process model in this paper can be drawn as follows:

- The variable data in the process must be dependent with time. All of these data are available by data acquisition system
- The input data and output data can be described for different symptoms and process state (fault) types using membership function of fuzzy set, for example, they can be divided into normal input data, abnormal input data (corresponding to certain symptoms), normal and abnormal output behaviors (corresponding to certain fault types), etc, and the process can be controlled based on the kinds of variables
- The process must be stable, dynamic and changeable process on the certain scale. Namely, the process works on one work point with certain range
- The input of the process or system could be multi-variables, which could have certain correlation to effect process output. The process plant is a typical nonlinear system
- The results of model output or controlled results for process can be allowed some error and deviation

Based on above preconditions of the process abstraction and model establishment, it is clear to see the preconditions and constraints are not too strict, most industrial processes in modern industry satisfy these conditions. Therefore, the model built on the thesis can be widely applied to some real industrial processes. In the thesis, the model is applied to diagnose and predict supply chain process so as to provide good supply prediction and risk analysis ability.

2.2.3 Basic Concepts and Terminologies

Based on the extraction of industry process characteristics and understanding of diagnosis in real world, some basic concepts and terminologies related to modeling for process monitoring, diagnosis and prediction have been given in order to further describe diagnosis model and process simply and well.

States and signals

Fault

An unpredicted deviation of at least one characteristic property or parameter of the system from the acceptable, usual or standard condition

Failure

A permanent interruption of a system's ability to perform a required function under specified operating conditions

Malfunction

An intermittent irregularity in the fulfillment of a system's desired function

Error

A deviation between a measured or computed value of an output variable and its theoretically correct one

Residual

A fault indicator, based on a deviation between measurements and model-equation-based computations

Symptom

A change of an observable quantity from normal behavior

Process state

When an industry process is on way, the process always functions on certain state or changes from one state to another state due to different input or inner parameter change in process, and then the different outcome is got according to different input and inner attributes. The operation or running state of real process can be treated as different process state at certain time.

Normal process state

A normal process state can be regarded as a kind of process states in which the process possesses desired or good process behavior. Under the state, the process has a good situation and health, and good process output can be acquired without fault occurrence. To common industrial processes, the real processes most stay in this state. The process state can be measured by some important process output data or be defined with a new indictor.

Abnormal process state

An abnormal process state can be treated as a kind of process states in which the process has bad output or abnormal process behaviors. Under the state, the process has a bad situation and health with fault occurrence, and deviation of regulated process goal and behaviors. The state is defined according concrete process goal and condition. In a common industry or business process, the abnormal state is undesired and need to avoid.

Normal input data

To a set of input feature vector, if the input data keeping in certain range make the process respond with a normal process state, then, these input data in the range can be defined as normal input data in the certain operation point.

Abnormal input data

To a set of input variables, if the input data are outside of normal data range and they also could affect the process response with an abnormal process state, then, these inputs features data can be defined as abnormal input data. Normally, abnormal input data corresponds with certain symptoms in fault detection problems.

Normal behaviors

According to some important process output, if these process output or new defined indictor can be used to express process performance and process response capability on large degree, these parameters can be used to measure process behaviors on line, When these output data stay under normal state, it indicates the process make a normal behavior responds in the certain operation point.

Abnormal behaviors

When these output data defined to represent process performance and process response capability stay under abnormal state, it indicates the process make an abnormal behavior responds. Normally, abnormal behaviors correspond with certain fault state in fault detection problems in the certain operation point.

Abnormal event

Aberrant events are all triggering factors that can trigger aberrant behavior occurrence of the process and make the process migrate to abnormal state. An aberrant event should be produced by an aberrant input data, parameter change from inner system or outer environments, and fault occurrence, and so on. When an abnormal event occurs, the abnormal process output and behaviors are triggered and then the process migrate abnormal state.

Functions

Fault detection

Determination of faults present in a system and the time of detection

Fault isolation

Determination of the kind, location and time of detection of a fault, Follows fault detection

Fault identification

Determination of the size and time-variant behavior of a fault, Follows fault isolation

Fault diagnosis

Determination of kind, size location and time of detection of a fault, Follows fault detection. Includes fault detection and identification

Monitoring

A continuous real-time task of determining the conditions of a physical system, by recording information, recognizing and indication anomalies in the behavior

2.3 Chapter Summary

In this chapter, first, a model–based fault diagnosis for dynamic system is introduced. It is mainly used to deal with time-invariant linear system. Some problems for the diagnosis of nonlinear time varying system are emerged in, and then, an overview of challenges with monitoring and diagnosis in industrial processes is given. Some main problems, challenges and new needs from modern industrial processes are presented here. It is these new needs that impulse the research work for developing intelligent model to deal with fault diagnosis in this paper. The second, in order to ensure the new process model to fit and solve some practical problems, the basic and common characteristics of complex industry plants are abstracted and extracted from system point of view at certain level, based on these basic characteristics, some preconditions or premises for building the intelligent model are fully defined.

Finally, to accurately understand the model and process concerning process analysis and fault diagnosis, some important and basic concepts and terminologies for practical process are given and defined in the last section.

CHAPTER 3 INTEGRATED INTELLIGENT MODEL FOR FDI TECHNOLOGY

In this chapter, some basic knowledge about intelligence technologies is first introduced and then, an integrated intelligent model based on intelligence technologies and data mining technologies for fault diagnosis and prediction is developed and implemented.

3.1 The Introduction of Computational Intelligence

3.1.1 Introduction

Intelligence means a kind of ability to learn from environment, history data, experience, and guild some activities in future based on the results of study. The implication of intelligence in higher-level means it has ability to sense the environments and make responses based on the environment change. These activities can be described in mathematic language, which mainly includes two kinds of activities, *classification* and *prediction* [N. K. Sinha, M. M. Gupta, 1999].

> *Classification*: It is a kind of ability that some difference among many objects can be identified and classified according to some certain different features of these objects
>
> *Prediction*: It is a kind ability that the possible behavior in future can be known according to current and history feature of these objects

These combinations of the two kinds of above abilities can form basic intelligence activities, for example, judgment, inference, induction and deduction and so on.

Intelligence has its typical feature, for example, learning ability, reasoning ability, induction and deduction ability. So far, these abilities can be reached a certain degree with some mathematical models and algorithms.

Some main applications with intelligence technologies focus on following fields in industrial and business fields:

- Modeling for complex process or system
- Intelligence system for control and monitoring
- Intelligent information process and analysis, for example, data mining and knowledge discovery, intelligence diagnose, intelligence information retrieve, pattern recognition, decision-making support and risk control, etc.

3.1.2 Soft Computing and Computation Intelligence

The world around us is imprecise, uncertain, and randomly changing, however, we can cope with such environment. The desire to mimic such coping leads to the basic premises and the guiding principles of soft computing, according to [L.AZadeh, 1994], the basic premises of soft computing are:

- The real world is pervasively imprecise and uncertain
- Precision and certainty carry a cost

And the guiding principle of soft computing, which follows from these premises, is

- Exploit tolerance for imprecision, uncertainty, and partial truth to achieve
- Tractability, robustness, and low solution costs

Both the premises and the guiding principle differ strongly from those in classical *hard computing*, which requires precision, certainty, and rigor. However, since precision and certainty carry a cost, the soft computing approach to computation, reasoning, and decision-making should exploit the tolerance for imprecision (inherent in human reasoning) when necessary. Soft computing draws from these ideas.

The term *Soft Computing* (SC) describes a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost. *SC* is not a closed and clearly defined discipline at present, it includes an emerging and more or less established family of problem-stating and problem-solving methods that attempt to mimic the intelligence found in nature. Learning from experimental data (statistical learning and neural computation) and fuzzy logic methods are two of the most important components of soft computing. In addition, there are, for example, genetic or evolutionary algorithms, probabilistic reasoning, fractals and chaos theories, machine learning, and belief networks and so on. Fuzzy logic is mainly concerned with imprecision, vagueness and approximate reasoning, neural computation with (sub-symbolic) learning and probabilistic reasoning with uncertainty.

Soft computing is not a mixture of NNs (Neural Networks), SVMs (Support Vector Machine), FLMs (Fuzzy Logic Models) but a discipline in which each of these constituents contributes a distinct methodology for addressing problems in its own domain, in a complementary rather than a competitive way. The principal members of the SC consortium are: fuzzy logic (FL); neuroucomputing (NC); evolutionary computing (EC); probabilistic computing (PC); and parts of machine learning theory (ML) [Tom M. Mitchell, 1998]. The common elements of these three models are *generalization, through nonlinear approximation and interpolation*, *in (usually) high-dimension spaces* [K. Vojislav, 2001]. For more theory and application with intelligence system and soft computing, please refer to [Azvine, B., Azarmi, N./Nauck, D. D, 2000] [Zilouchian, Ali, Jamshidi Mo, 2001].

Computation Intelligence (CI)

As we known, the classical regression and Bayesian classification statistical techniques are based on the very strict assumption that probability distribution models or probability-density functions are known. Unfortunately, in many practical situations, there are not enough information about the underlying distribution laws, and distribution-free regression or classification is needed that dose not require knowledge of probability distributions. This is very serious restriction but very common in real-world applications. Based on the premises and the fact, the *Computation Intelligence* (CI) can overcome the limitation and plays more and more important role in modeling and learning from data.

So for, there is no clear definition of the computation intelligence concept apart from the simple fact that it represents a category of techniques in *Artificial Intelligence* (AI) that can be used for analyzing, designing and developing intelligent systems. According to the normal understanding, the group consists of Artificial Neural Networks (ANNs), Fuzzy Logic Systems (FLS) and Genetic Algorithms (GA).

Artificial Neural Networks (ANNs)

According to one of the definition of ANNs [Kosko B, 1992 and Hecht-Nielsen, 1990]

The technologies discipline concerned with information processing systems that autonomously develop operational capabilities in adaptive response to an information environment.

Artificial Neural Networks (ANNs) is a network in topology, which is consisted of many neurons and is formed by connection of these neurons according to some rules. Normally, it has different layer structures. Neurons are similar to creature neurons, and the artificial neural network simulated the structure and function of human brain at low level. Its essential principle is to first study some learning samples, which must be able to perfectly describe the wished performance of system. During the learning phase of ANNs, sample data input ANNs and ANNs can adjust the weight of network connection based on some algorithms, these weights can converge to some certain value in order to make ANNs have a certain output value, hence, every input sample has a corresponding output sample. When the ANNs has a new data as input, the ANN trained well by samples can correctly response it based on the relationship mapped between sample input and output in multidimensional space. ANNs is built based on strong mathematic theories. Much research has indicated that ANNs has many traits.

1. Arbitrarily press on towards nonlinear
2. Studying and adapting unknown and uncertain system
3. Parallel structure
4. Robust and tolerant capability…

All features indicate ANNs could be widely applied in data process, modeling and process control, especially in complex and uncertain nonlinear system. In monitoring and diagnosis field, one of possible applications is to use ANNs as a classifier for detecting critical units' abnormal situations. ANNs is a black-box approach to describe real system.

There are many different topology structures of neural network and learning algorithms applied in neural network, for example, Multiplayer Perceptrons (MLP), RBF (Radial Basis Function), Self-Organizing Map (SOM) with difficult topology structures and learning algorithms, respectively. Neural networks establishes system model through training its network structures and parameters. Many different training algorithms and strategies are employed to train NN model, more knowledge about NNs, please refer to [Kosko,B., 1992][N.K.Sinha, M.M.Gupta, 1999][Pedrycz, W., 1993][K.Vojislav 2001].

Fuzzy Logic (FL)

Fuzzy Logic (FL) is a tool for embedding structured human knowledge into workable algorithms, According to the definition of fuzzy logic, in a narrow sense, fuzzy logic is considered an approximate rather than exact. In a wider sense, fuzzy logic is treaded as a fuzzy set theory of classes with unsharp or fuzzy boundaries [K. Vojislav 2001]. Fuzzy logic method can be used to design intelligent system on the basis of knowledge expressed in a common language. The main reason for such versatility is that this method can process both of symbolic and numerical information.

The contribution of fuzzy logic is that it allows building and dealing with system by our word in a very natural fashion. It is a white box approach in the sense that it is assumed that there is already human knowledge about the solution. Therefore, the modeled system is known. On the application level, FL can be considered as an efficient tool for embedding structure human knowledge into useful algorithms. As in human reasoning and inference, the truth of any statement, measurement, or observation is a matter of degree. The degree is expressed through the membership functions that quantify (measure) a degree of belonging of some (crisp) input to given fuzzy subsets.

Fuzzy sets

To describe linguistic statements such as: x is large, Zadeh introduced the concept of a fuzzy set[L. A Zadeh, 1965]. A fuzzy set A is a collection of elements defined in a universe of discourse labeled X. It generalizes the concept of a classical set by allowing its elements to have partial membership (usually $\in[0,1]$), and the degree to which the generic element X belongs to A is characterized by a membership function $\mu_A(x)$, which associates with each element $x \in X$, a number $\mu_A(x)$ representing the grade of membership of x in A, and is designated as:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

Associated with a classical binary or crisp set is a characteristic function, which returns 1 if the element is a member of that set and 0 if not. The fuzzy membership function generalizes this concept by allowing elements to be partial members of a set, reflecting degrees of uncertainty about the information. In fact, almost variables can be regarded as fuzzy variables.

Membership function

Each linguistic term, such as cool, medium or hot, is represented by a membership function and the set of all these terms determines how an input variable is represented within the fuzzy input as follows Figure 3.1.

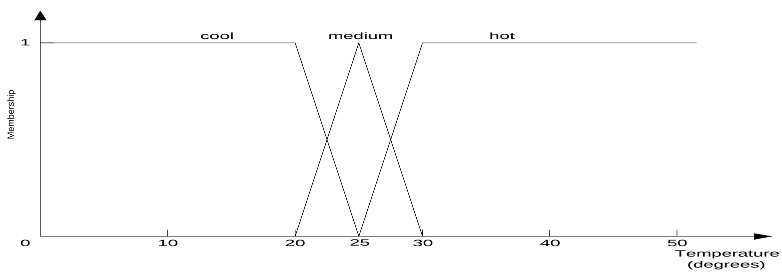


Figure 3.1: Membership functions for temperature

The support a fuzzy set A is the set of inputs that have a non-zero membership function value, i.e. the support for fuzzy set "cool" is $\{temperature\colon \mu_{cool}(temperatue) > 0\}$. For example, a temperature x with 22.5 degrees can be regarded as belonging to fuzzy set "cool" with a membership grade of 0.5 and at the same time it can also be regarded as belonging to fuzzy set "medium" with a membership grade of 0.5.

Besides the triangle and trapezoid functions used in Figure 3.1, other commonly used membership functions include Gaussian functions and B-spline functions of different orders, and so on.

Fuzzy operations

Using fuzzy sets, the behavior about the object can be represented as the form of fuzzy relations. These relations are composed of fuzzy expressions that are connected by fuzzy logical operators. Four logical operators are commonly applied in a fuzzy relation: intersection (AND), union (OR), complement (NOT) and implication (IF … THEN …).

Defuzzification

Defuzzification is the process of representing a fuzzy set with a crisp number. Internal representations of data in a fuzzy system are usually fuzzy sets but the output frequently needs to be a crisp number that can be used to perform a function, such as commanding a valve to a desired position in a control application, or indicate a problem risk index. The most commonly used defuzzification method is the center of area method, also commonly referred to as the centroid method. This method determines the center of area of fuzzy set and returns the corresponding crisp value.

Fuzzy Inference Systems (FLS)

Fuzzy modeling stems from advances in logic and cybernetics. Originally, fuzzy systems were developed in the 1960s as an outcome of fuzzy set theory [L. A. Zadeh, 1978]. The fuzzy sets are a mathematic means to represent vague information. In application to process modeling and control, the uncertainty handling aspects have, however, received less interest. Instead, the focus has been in extending the interpolation capabilities of rule-based expert system.

In rule based inference system, the universe is partitioned using concepts, models via sets. Reasoning is then based on expressions of logical relationships between the concepts: if –then rules. Some popular fuzzy inference system, for example, Fuzzy manmdani system, Fuzzy TS system and Adaptive Neural Fuzzy System (ANFS), have been widely applied in real world. More knowledge about Fuzzy system, please refer to [Kandel, A., 1996] [L. A. Zadeh, 1978].

Basically, *fuzzy variables*, *fuzzy relationships*, *fuzzy operations* and *fuzzy inference* make up major parts of a fuzzy system.

Fuzzy System in Process Modeling

Neural networks have been shown to be very efficient in their function approximation capabilities, which mimic the observed input-output behavior. Unfortunately, neural network appear as black-box models to the developer and end user, it is difficult give any additional explanation on what the mapping rule inside is or what relation between input and output is. Obviously, some transparency will help to evaluate the validity of the model and to locate unsatisfactory behavior when further model development is need. The need for transparency has motivated the use of fuzzy system.

From the process modeling point of view, the applications of fuzzy system have shown two main contributions due to use of fuzzy systems:

- A reduction of the complexity of system, based on the use of fuzzy sets; and
- A transparent form of reasoning (similar to the conscious reasoning by humans)

In process modeling, fuzzy logic is an efficient way to quickly build a model or a controller for a process when only rough information is available. Also non-linear system can be considered without extra effort. In a standard learning approach:

1. Fuzzy sets and rule are stated by the experts
2. The system structure is established, and
3. The membership function and /or output constants are fine-tuned using data

This allows us to build a system model based on experimental human knowledge.

Genetic algorithm (GA)

Genetic Algorithm (GA) was first introduced by John H. Holland in the 1960s and was developed by Holland and his students in the 1960s and 1970s [J. H. Holland, 1962, 1973, 1994]. GA is a global searching and optimization strategy. GA use global information, perform parallel search and do not require local gradient information, which enables it to find globally optimal or near globally optima solutions. GA is similar with the natural selection and evolution of creature. The basic idea behind GA is to evolve a group (called generation in GA) of possible candidate solutions (also called chromosomes) to a problem at hand, using several operators (such as crossover, mutation or inversion).

A GA is an iterative procedure that maintains a constant population size and works as following. An initial population is generated at random or heuristically. During each iteration step (generation), the individuals in the current population are evaluated a given a fitness value based on the fitness function, which represents optimization objective function. To form a new population, individuals are selected with a probability proportional to their relative fitness. This ensures that the expected number of times and chosen individual is approximately proportional to its relative performance in the population, so that good individuals have more changes of being reproduced. The selection procedures alone cannot generate any new point in the search space. Two genetic operators, namely *crossover* and *mutation* are used for generating new individuals. Crossover is the most important recombination operator, which takes two individuals called parents and produces two new generations called the offspring by swapping parts of the parents. Through crossover the search is biased towards promising regions of the search space. The mutation operator is essentially background noise that is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space.

GA is stochastic iterative algorithms without converge guarantee. Termination may be triggered by reaching a maximum number of generations or by funding an acceptable solution.

Some basic issues between ANN and FL

Summarily, a multiplayer perceptron, and RBF network, as powerful nonlinear tools, have the same structure from mathematic point of view. Thus, their representational capabilities are the same or very similar after successful training. All models learn from a set of training data and try to be as good as possible in modeling typically sparse and noisy data pairs in high-dimensional space. The output from these models is a hyper-surface in multidimensional space. In trying to find the best models, one should be able to measure the accuracy or performance of the models. To do that, some measures of goodness, performance or quality typically are applied.

The basic difference between these models is, each uses a different norm (error, risk, and cost function) that measures the goodness of the models, and optimization of the different measures results in different models. The application of different norms also leads to different learning (optimization) procedures. The basic norms (risk, cost function, error) applied in developing the two basic networks is shown Table 3.1 [K,Vojislav 2001].

Table 3.1: Basic models and their (Risk) functions

Models	***Risk Function***
Multiplayer Perceptron	$E=\sum_{i}^{P}(d_i - f(X_i,W))^2$
Radial Basic Function	$E=\sum_{i}^{P}(d_i - f(X_i,W))^2 + \lambda \parallel Pf \parallel^2$

A multilayer perceptron is a representative of nonlinear basis function expansion (approximation):

$$o = f(X,W,V) = \sum_{i=0}^{N} w_i \varphi_i (X_i, V_i), \tag{3.1}$$

An RBF Network is a representative of a linear basis function expansion:

$$o = f(X,W) = \sum_{i=0}^{N} w_i \varphi_i (X_i), \tag{3.2}$$

A fuzzy logic model, like an RBF network, can be a representative of a linear or nonlinear basis function expansion:

$$o = y = f(X,C,r) = \frac{\sum_{i=0}^{N} G(X,C_i) r_i ,}{\sum_{i=1}^{N} G(X,C_i)} \tag{3.3}$$

Both NNs and FL models are universal approximators in the sense that they can approximate any function to any degree of accuracy provided that there are enough hidden layer neurons in NNs or rules in FL. This is called universal approximation theory.

3.2 Data Mining and Knowledge Discovery

3.2.1 What is Data Mining and Knowledge Discovery

Simply speaking, Data Mining (DM) refers to extracting or mining knowledge from large amount of data. The basic functions for data mining mainly include: *generalization, classification, cluster, association analysis and prediction.*

The major reason that data mining has attracted a great deal of attention in the information technology in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, information analysis and intelligent technologies, production control, market analysis, engineering design and science exploration.

Data mining is a result of natural evolution of information technology and data process and analysis. The evolutionary path is from data collection and data base creation, data management (for example, database for data storage and retrieval, database transaction processing), and data analysis and understanding (involving data warehousing and data mining). Data mining performs data analysis and may uncover important data patterns, contributing greatly to business strategies, knowledge bases, and scientific research. It bridges the widening gap between data and information.

Data mining focuses on extracting useful knowledge and information that hides behind a massive amount of data. Normally, it is very difficult to find these knowledge and information directly by your sense. At the same time, these knowledge and information only can be addressed from massive data. It means that knowledge and information are embedded into an amount of data and also flooded by these data.

Data mining technology is first developed and improved by business requirements, and now, it has been very successful to be widely applied in different fields, especially in business field. Some application of data mining is listed as follows:

- Data analysis and decision support, for example, market process and state analysis, market strategy support and so on
- Market analysis and management, for example, increasing sale and profitability, segmentation, customer retention, state and loyalty, etc
- Risk analysis and management, for example, business process risk, financial risk, supply chain risk, production sale risk and so on
- Fraud detection, for example, Credit analysis, fraud behavior detection and so on
- Customer analysis and identify, for example, customer segments, potential customer detection. Customer loyalty analysis, etc

3.2.2 What is Data Mining Function

From a data analysis point of view, data mining can be classified into two categories, *descriptive data mining* and *predictive data mining*. *Descriptive data mining* describes the data set in a concise and summarative manner and presents interesting general properties of the data set. *Predicative data mining* analyzes the data in order to construct one or a set of models, and attempts to predict the behavior of new data sets. They are comprised of follows functions.

Concept Description

The simplest kind of descriptive data mining is concept description. It generates descriptions for characterization and comparison of the data. There are two types of concept description. Characterization provides a concise and succinct summarization of the given collection of data; comparison (discrimination) provides descriptions comparing two or more collections of data.

Association Analysis

Association rule mining is used for finding interesting association or correlation relationships among a large set of data items. The discovery of interesting association relationships among huge amount of process data can help decision making and risk analysis.

For example, in business filed, a typical example of association rule mining is market basket analysis. This process analyzes customer-buying habits by finding associations between different items that customer place in their “shopping baskets”.

Association rules associate a particular conclusion (for example, the purchase of a particular product) with a set of conditions (the purchase of several other products), the rule is defined as follows:

X->Y [support, confidence]

Support(X->Y) = No of tuples containing both X and Y/Total number of tupies

Confidence (X->Y) = No of tuples containing both X and Y / No of tuples containing X

For example, the rule

Beer<=cannedveg & frozenmeal (173,17.0%,0.84)

States that beer often occurs when canedveg and frozemeal together. The rule is 84% reliable and applies to 17% of the data, or 173 records. Association rule algorithms automatically find the associations that hide behind of amount of data.

The advantage of association rule algorithms over the more standard decision tree algorithms (C5.0 and C&R Trees) is that associations can exist between any of the attributes. Decision tree algorithms build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

Classification and Predictions

Classification and prediction are two forms of data analysis that can be used for extract models describing important data class or predict future data trends. *Classification predicts categorical labels and prediction models continuous-valued functions*. The classification is a process of supervised learning and includes two-step process, the first is to determine predefined class by one of the attributes called the class label attribute and then train and build the classifier using the training data and training class label, the second is to classify real test data using the classifier.

Some technologies, for example, decision tree induction, Bayesian classification, Bayesian belief networks, and neural networks, have been developed for classification. The classification and prediction have numerous applications including credit approval, medical diagnosis, performance prediction and selective marketing etc. for prediction, including linear, nonlinear, and generalized linear regression models are applied in prediction model.

Clustering

When class label of each object is not known, clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other objects. In the clustering, the distance measures in multi-dimension are used for assessing the attribute values describing the objects.

Some technologies and methodologies, for example, partitioning methods, hierarchical methods, density-based methods and models-based methods, have been developed for clustering.

There are some typical applications of clustering, for example, in business, clustering can help markers discover distinct group in their customer bases and characterize customer groups based on purchasing patterns. More application has done in biology, information process, pattern recognition, etc.

Many algorithms are employed for clustering, for example, similarity and dissimilarity measurement between objects, Partition algorithms, K-means clustering method, density-based clustering, hierarchical methods, grid-based methods, model-based clustering method, etc.

3.3 General System Model, Structures and Identification

Information for the modeling and identification of processes or system can be obtained from different sources:

- Mechanistic knowledge obtained from first principles (physics and chemistry)
- Empirical or expert knowledge expressed as linguistic rules
- Measurement data, obtained during normal operation or from an experimental process

According to the type of available information, three basic levels of model synthesis can be defined.

- **White-Box or first principle model**. A complete mechanistic model is constructed from prior knowledge and physical laws. Here, the dynamic models are derived based on mass, energy and momentum balances of the process.
- **Black-box model or empirical model**. No physical (prior) knowledge is used to construct the empirical model.
- **Fuzzy logic model**. A linguistically interpretable rule-based model is formed based on the available expert knowledge [M.A Henson, 1998].

This means, if the mechanistic knowledge about the process is known, a White-Box model described by analytical or deferential equations can be adopted; if only information or experience available, the model based Fuzzy system can be used; if the most valuable information comes from input-output data colleted from process, the black-box models is needed. The black-box models are very valuable when an accurate model of the process dynamics is desired. Therefore, nonlinear black-box modeling is a challenging and promising research field [E.Hernandez and Y.Arkun, 1993][J.Sjoberg, Q.Zhang,1995].

3.3.1 Process Dynamical Model

As the nomenclature of nonlinear dynamical model is based on the terminology used to categorize linear input-output models, in the following the linear empirical model structures that can be summarized by the general family [L.Ljung, 1987][J Sjoberg, Q, zhang, 1995].

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k) \qquad (3.4)$$

Where, q denotes the shift operator. For instance, *A(q)* is a polynomial in $q^{-1.}$ This model can be given in a "pseudo-linear" regression form

$$\hat{y} = Q^T X(k) \qquad (3.5)$$

where the regressors, *i.e* the components of *X(k)* can be given by:

- *u(k-i),* i=1,…n_b, control signals (associated with the B polynomial)
- *y(k-i),* i=1,…n_a, measured process outputs (associated with the A polynomial)
- $\hat{y}(k-i)$, simulated outputs from past u(k) (associated with the F polynomial)
- $e(k-i) = y(k-i) - \hat{y}(k-i)$, prediction errors (associated with the C polynomial)
- $e_u = y(k-i) - y_u(\hat{k}-i)$, simulation errors (associated with the D polynomial)

Based on these regressors, different types of model structures can be constructed. For instance, the simplest linear dynamic model is the *finite impulse response* (*FIR*) model, which is expressed as following:

$$\hat{y}(k) = B(q)u(k) + e(k) = b_1 u(k-1) + ... + b_n(k-n_b) + e(k) \qquad (3.6)$$

In this equation, the corresponding predictor $\hat{y} = B(q)u(k)$ is based on the $x(k) = [u(k-1),...u(k-n_b)]$ regression vector.

Some special cases of (3.4) as the Box-Jenkins (BJ) model (A=1), the ARMAX model (F=D=1), the output-error (OE) model (A=C=D=1) and the ARX model (F=C=D=1).

Following this nomenclature of linear models, it is natural to construct similar nonlinear model as:

- *NFIR, Nonlinear Finite Impulse Response Models,* in which case, the regressor nonlinear model structures is comprised as:
$$X(k) = [u(k-1),..., u(k-n_b)] \qquad (3.7)$$
- *NARX, Nonlinear AutoRegressive with eXogenous input models,* which used regressors as:
$$X(k) = [y(k-1),..., y(k-n_a), u(k-1),..., u(k-n_b)] \qquad (3.8)$$
- *NOE, Nonlinear Output Error Models,* which used regressors as
$$X(k) = \left[\hat{y}(k-1),..., \hat{y}(k-n_a), u(k-1),..., u(k-n_b)\right] \qquad (3.9)$$

- *NARMAX, Nonlinear AutoRegressive Moving Average with eXogenous input models,* where
$$X(k)=\left[y(k-1),...,y(k-n_a),u(k-1),...,u(k-n_b),\varepsilon_u(k-1),...,\varepsilon_u(k-n_e)\right] \tag{3.10}$$
- *NBI, Nonlinear Box-Jenkins models,* where the regressors are past inputs, past estimated outputs, estimation errors using past outputs and the estimation errors using pasting estimated outputs.
$$X(k)=\left[\hat{y}(k-1),...,\hat{y}(k-n_a),u(k-1),...,u(k-n_b),\varepsilon_u(k-1),...,\varepsilon_u(k-n_e),\varepsilon(k-1),..\varepsilon(k-n_e)\right] \tag{3.11}$$

On the soft computing and system identification of nonlinear system, the NARX model is called a series parallel model, while the NOE is referred to as a parallel model [K.S. Narendra and K.Parthasarathy, 1990]. The NARMAX, NOE and NBJ models are recurrent models, because they use the estimated output that constitutes a feedback. This makes the identification of these models difficult. Because the NARX model structure is non-recursive, its parameters are easy to be estimated. To present a dynamic system, after making some moderate assumptions, it has been proved that any nonlinear, discrete, time-invariant system can always be represented by a NARX model [K. Vojislav, 2001]. Therefore, the NARX model is frequently used for system description and identification.

3.3.2 Fuzzy Inference Model

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all of the elements in fuzzy system: *fuzzy variables, membership functions, fuzzy logic operators and if-then* fuzzy inference rules. Typically, there are two main types of fuzzy inference system that can be implemented in the fuzzy logic: *Mamdani–type* and *Sugeno type*. There two types of inference systems vary somewhat in the way outputs are determined [Jang, J.-S. R 1997, Mamdani, E.H 1975, Sugeno, M., 1985].

Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert system, system modeling and identification and so on.

Mamdani's fuzzy inference System

Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. It was among the first control system built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani [Mamdani, E.H 1975]. The whole fuzzy inference process can be described in Figure 3.2 as follows with a simple example.

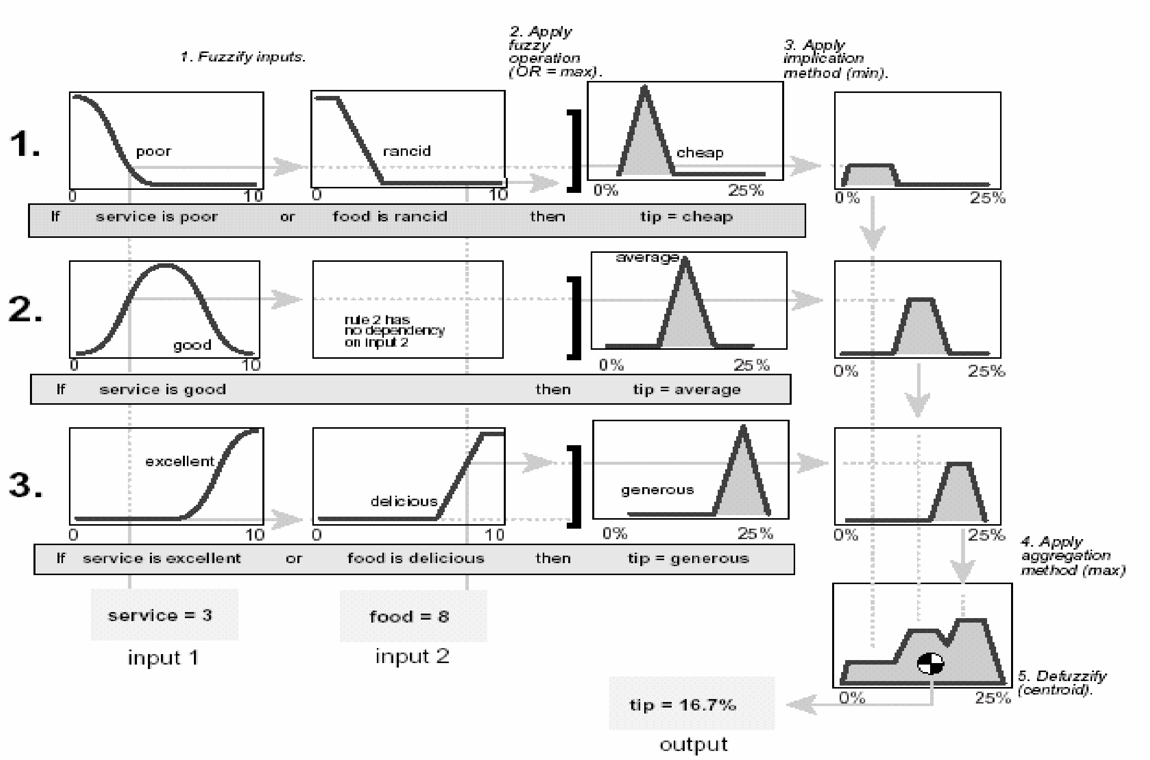


Figure 3.2: The fuzzy inference process based on Mandani model

As shown above figure, there are five parts in whole fuzzy inference process based on Mandani type. They correspond to sequence number shown in above Figure 3.2.

1) *Fuzzyfication of the input variables*. Take the input and determine the degree to which they belong to each of appropriate fuzzy sets via membership functions. It realizes transition from crisp numerical input value into fuzzy degree value. The output is a fuzzy degree of membership in the qualifying linguistic set.
2) *Application of fuzzy operator (AND or OR) in the antecedent part*. Once the input has been fuzzfied, we know the degree to which each part of the antecedent has been satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator (normally, includes AND or OR) is applied to obtain one number that represents the result of the antecedent for that rule. The number will then be applied to the output function. The input fuzzy operator is two or more membership values from funzified input variables. The output is a single truth-value.
3) *Implication from the antecedent to the consequent part*. The input for the implication process is a single number given by the antecedent part, and the output is a fuzzy set in consequent side. Implication is implemented for each rule.
4) *Aggregation of the consequents across the rules*. It is a process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variables, just priori to fifth and final step. Defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each

rule. The output of the aggregation process is one fuzzy set for each output variables.

5) *Defuzzification*. The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single crisp number. The most popular defuzzification is called the center-of gravity or centroid calculation, which returns the center of area under the curve. It is described by the equation below (3.12). There are numerous other types of defuzzifies such as centre-of sums, first-of-maxima, and middle-of-maximal [D.Driankov, Hellendoorm, 1993].

$$y = \frac{\sum_{j=1}^{N_r} \beta_j f_j(X)}{\sum_{j=1}^{N_r} \beta_j} \qquad (3.12)$$

Some of Advantages of the Mamdani Method can be listed as follows:

- It is intuitive
- It has widespread acceptance
- It's well-suited to human input

Takagi-Sugeno-Kang, inference model

Another type of fuzzy inference model is so called Sugeno, or Takagi-Sugeno-Kang (TSK or TS) method, which is introduced in 1985 [Sug 1985], it is similar to the Mandani method in many respects. The first two parts of the fuzzy inference process, fuzzfying of the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

A typical rule in a Sugeno fuzzy model has the form:

If Input 1=x, and Input 2=y then output is Z=ax+by+c (3.13)

For a zero order Sugeno fuzzy model, the output level z is a constant (a=b=0). The Sugeno rule operates as shown in the following diagram as the examples.

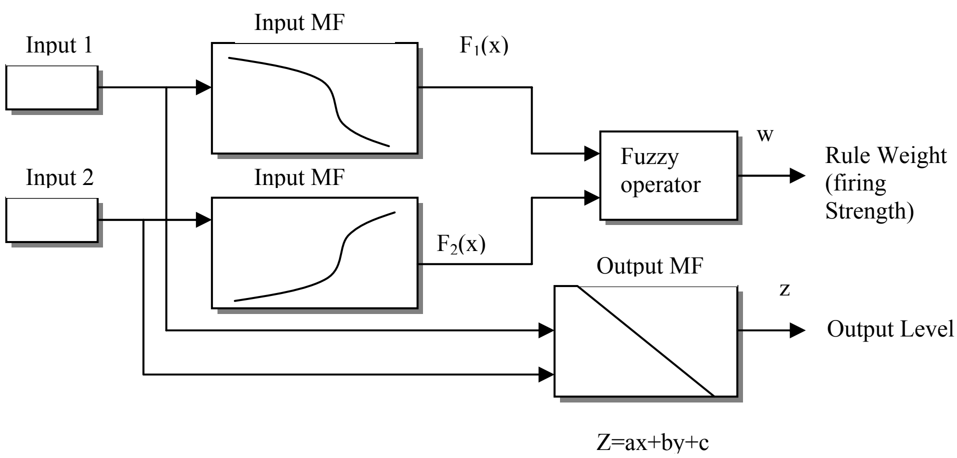


Figure 3.3: The operation process based on TS fuzzy model

The final output of the system is the weighted average if all rule outputs, computed as equation (3.14).

$$Y = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{3.14}$$

The whole Takagi-Sgueno fuzzy model can be illustrated as following Figure 3.4 as an example.

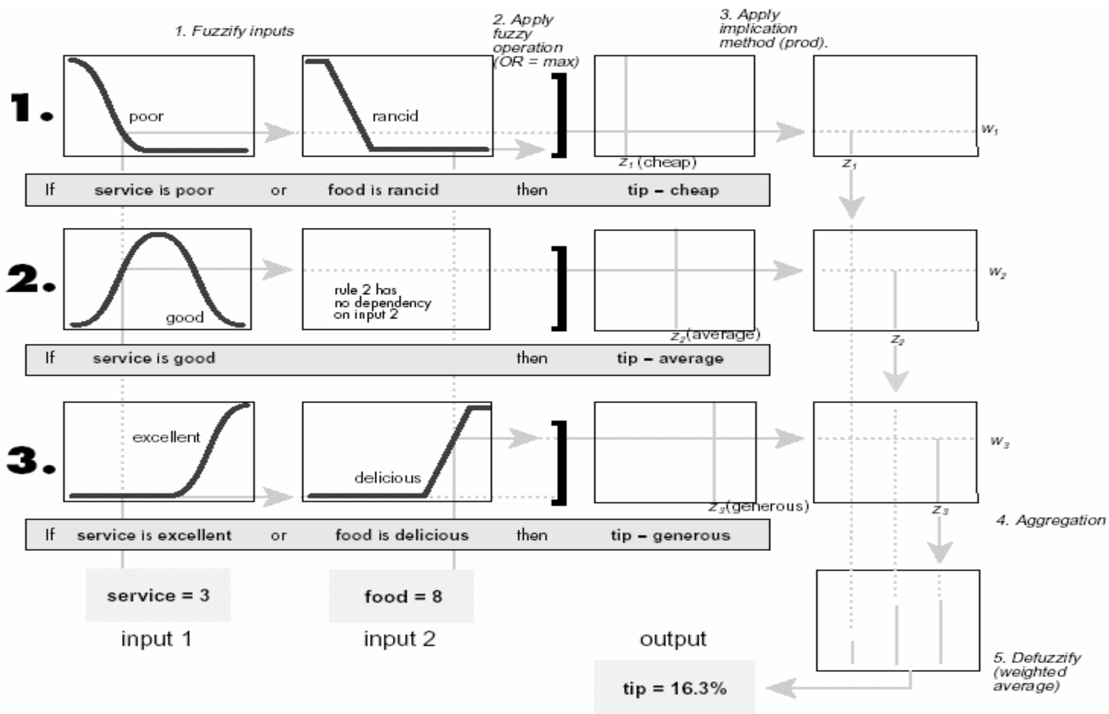


Figure 3.4: The fuzzy inference process based on TS model

Because it is a more compact and computationally efficient representation than a Mamdani system, the Sugeno system lends itself to the use of adaptive techniques for constructing fuzzy models. *These adaptive techniques can be used to customize the membership functions so that the fuzzy system best models the data*. As we known, the Adaptive Neural Fuzzy Inference System (ANFIS), which is used for fuzzy modeling procedure to learn information about a data set, so as to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output, is built based on Sugno Fuzzy model. Some advantages of the Sugeno Model are listed as follows:

- It's computationally efficient.
- It works well with linear techniques
- It works well with optimization and adaptive techniques
- It has guaranteed continuity of the output surface
- It's well-suited to mathematical analysis.

3.3.3 Fuzzy Models of Dynamic Systems

Fuzzy models of dynamic system are usually based on Nonlinear AutoRegressive with eXogenous input (NARX) model structure. The model established a nonlinear relation between the past inputs and outputs and the predicted output, the system prediction output is combination of system output produced by real inputs and system historical behaviors. It can be expressed as:

$$\hat{y}(k) = f(y(k-1),...,y(k-n_a),u(k-n_d),...,u(k-n_b-n_d)) \qquad (3.15)$$

Here, n_a and n_b are the maximum lag considered for the output, and input terms, respectively, n_d is the discrete dead time, and f represents the mapping of fuzzy model.

The structure of the NARX TS fuzzy model

The NARX Takagi-Sugeno fuzzy model interpolates between local linear, time-invariant (LTI) ARX models as follows:

Rj: *if $z_1(k)$ is $A_{1,j}$ And …And $z_n(k)$ is $A_{n,j}$*
Then

$$\hat{y}(k) = \sum_{i=1}^{n_a} a_i^j y(k-i) + \sum_{i=1}^{n_b} b_i^j u(k-i-n_d) + c^j \qquad (3.16)$$

where the element of **z**(k) "scheduling vector" are usually a subset of the **x**(k) regressors that contains the variables relevant to the nonlinear behaviors of the system,

$$Z(k) \in \{y(k-1),...y(k-n_a),u(k-n_d),...u(k-n_b-n_d)\}$$

while the $f_j(q(k))$consequent function contains all the regressors q(k)=[X(k) 1],

$$f_j(q(k)) = \sum_{i=1}^{n_a} a_i^j y(k-i) + \sum_{i=1}^{n_b} b_i^j u(k-i-n_d) + c^j \qquad (3.17)$$

The NARX type zero-order TS fuzzy model (singleton or sugeno fuzzy model) is formulated by simple rules consequents as:

Rj: if $Z_1(k)$ is $A_{1,j}$ And…and Zn(k) is $A_{n,j}$
Then

$$\hat{y}(k) = c^j$$

While, the z(k) contains all inputs of the NARX model:

$$Z(k) = X(k) = [y(k-1),...,y(k-n_a),u(k-n_d),...,u(k-n_b-n_d)] \qquad (3.18)$$

The diffidence between NARX TS fuzzy method and Fuzzy TS model is:

The output from fuzzy ST model is linear and constant, and the output from fuzzy NARX fuzzy TS model is NARX function. But they have same inference structure.

3.3.4 System Identification

Both TS Fuzzy model and NARX model give good process description if their model parameters are determined well. But a practical problem is how to determine these parameters so as to make fuzzy model match the real process perfectly. Indeed, one aspect, some prior knowledge about the real process is helpful to build these expressions due to fully understanding these processes underlying rule. On the other hand, system identification is also applied in the filed to build the real system model through experiment ways.

One kind of definition of system identification is given as: *The determination of a mathematical model of a process or system is known as system identification* [E. Ikonen, K. Najim 2002]. In control system, a mathematical model of a process or system is in most cases necessary for the design of the controller. There are basically two ways of determining a mathematical model of a system: *by implementing known laws of nature or through experimentation on the process*, a popular approach to obtaining a model is to combine both ways.

Mathematical models may be distinguished as parametric and nonparametric models. Parametric model involves parameters, for example, the coefficients of differential or difference equation, of state equation, and of transfer function, etc. nonparametric models describes real system without explicit model parameters.

The system identification problems

The basic dynamic model is the linear difference equation in system description. An example of such an equation can be NARX model (in equation 3.8) in nonlinear system. The output at time t is thus computed as a linear combination of past outputs y and past inputs u. The output at time t depends on the input signal at much previous time instant. This is what the word dynamic refers to. The system identification problems are then to use measurements of u and y to figure out:

- The coefficients in the equation
- How many delayed outputs to use in the description
- The time delay in the system is
- How many delayed inputs and outputs are usually referred to as the model order(s)

System identification using dynamic mathematic model

There are many ways to implement system identification due to different model structures, for example, to mathematical model of linear, time-invariant, single-input-output (SISO) discrete time system, described by different equation, the proposed method for the identification (estimation) of the coefficients (parameters) of a difference equation is experimental and may be briefly described as follows. First, a set of N linear algebraic equations are formulated, where N is the number of measurements. From these equations, a *canonical equation* whose solution yields the parameter estimate $\Theta(N)$, where Θ is vector parameter under identification. If an estimate of the initial conditions of the dynamic equation is also required, the $N+n$ measurements are taken and hence $N+n$ equations are produced, where n is the order of the difference equation, for example, it can be implemented as follows:

The n order difference equation has the form

$$y(k) + a_1 y(k-1) + ... + a_n y(k-n) = b_1 u(k-1) + ... + b_n (k-n) \quad (3.19)$$

with initial condition *y(-1),y(-2),...y(-n)*. The unknowns are the parameters $a_1, a_2, \ldots a_n$, $b_1, b_2, \ldots b_n$. As before take *N+n* measurements, we can replace these measurements in the above form and get the N equations. This N equation can be expressed as:

$$y = \phi\theta \tag{3.20}$$

Where, $\theta^T = \left[a_1\, a_2 \quad a_n\, b_1 \quad b_{n-1}\, b_n\right]$

$$y^T = \left[y(n)\, y(n+1) \quad y(n+N-1) \quad \right]$$

Based on above relation, the canonical equations for 3.20 take the form

$$\phi^T \phi\theta = \phi^T y$$

and therefore

$$\theta = (\phi^T \phi)^{-1} \phi^T y \tag{3.21}$$

under the assumption that the matrix $\phi^T \phi$ is invertible.

The above system identification is called *off-line* parameters estimation due to the parameter identification off line way.

In many practical cases, it is necessary that the parameter estimation takes place concurrently with the system operation. Which is called *on line identification* and its methodology usually leads to a recursive procedure for every new measurement (or data entry). For this reason, it is also called *recursive identification*. On-line identification is based on the idea: Assume that we have available an estimate of the parameter vector *Θ* based on N pairs of input-output data entries. Let this estimate be denoted by *Θ(N)***.** Assume that *Θ(N)* is not accurate enough and we wish to improve the accuracy using the new (next) N+1 data entry. Clearly using N+1 data entries, we will obtain a new estimate for *Θ***,** denoted as *Θ(N+1)***.** Which is expected to be an improved estimate compared with the previous estimate *Θ(N)* **.**

$$\theta(N+1) = \theta(N) + \Delta\theta = \theta(N) + \gamma(N)\left[y_{N+1} - \varphi^T(N+1)\theta(N)\right] \tag{3.22}$$

where, $\gamma(N)$ and $\varphi(N+1)$ are known vector quantities and y_{N+1} is N+1 measurement of the output y of the system. This formula shows that for the determination of *Θ(N+1)* one can use the previous estimate *Θ(N)* plus a corrective term $\Delta\theta$, which is due to the new measurement, instead of starting the estimation procedure right from the beginning. Based on the equation (3.22), some system optimization methods have been developed so as to minimize the error function, which is define as the difference between real system output and system prediction output in (3.22), for example, SPR-Lyapunov Design approach, Gradient Method, Least-Squares method, etc. As the result of system optimization, the optimal parameter values can be regarded as the value of system parameter estimation. The further methodology can be referred many literatures [CI Byrnes, A Lidquist1,1986].

System identification using NNs

In reality, the nonlinear function *f* from (3.15) is very complex and generally unknown. The whole idea in the application of NNs is to try to approximate *f* by using some known and simple functions. It is significant to involve system identification method to determine the nonlinear function *f* in equation (3.15). The NNs provide a good choice.

The identification phase of the mathematical model (3.15) can be given a graphical representation as Figure 3.5 with NNs model. Two different schemes can be applied. They are: *Series-parallel* and *parallel* as follows:

$$\hat{y}(k+1) = f\{y(k),...,y(k-n);u(k),...,u(k-n)\} \quad \text{Series-parallel} \quad (3.23)$$

$$\hat{y}(k+1) = f\left\{\hat{y}(k),...,\hat{y}(k-n);u(k),...,u(k-n)\right\} \quad \text{Parallel} \quad (3.24)$$

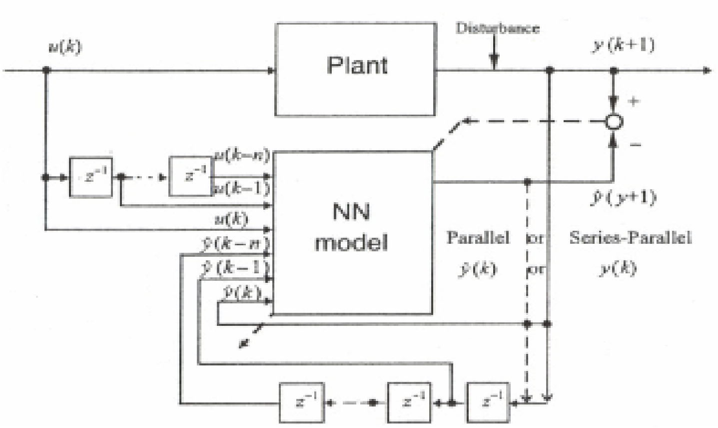


Figure 3.5: The basic scheme of the system identification using NNs

Narendra and Annaswamy (1989) shown (from linear system) the series-parallel method to be globally stable, but similar results are not available for the parallel mode yet. The parallel model has an advantage of avoiding noise existing in real-plant output signals; on the other hand, the series parallel scheme uses the real plants outputs, and this generally enforced identification.

According to above system identification scheme using NNs series parallel scheme Figure 3.5, the expression of whole system of Series-parallel is:

$$\hat{y}(k+1) = f(u(k-n)...u(k), y(k-n)...y(k)) \quad (3.25)$$

In fact, the equation (3.25) also indicates the similar structure with NARX dynamic model, which is expressed in equation (3.8). It means the series parallel scheme of NNs model in Figure 3.5 can be applied in system identification to express a NARX dynamic model due to the similar mathematic expression and role.

With time varying processes, the model parameters also need to be updated on-line. Online identification may be performed continuously, so that the model is updated at each sample instant. Alternatively, the model may be updated at certain times 'when necessary'. An adaptive system is able to adjust to the current environment: to gain information about the current environment, and to use it. An adaptive system is *memoryless* in the sense that it is not able to store this information for later use. All new information replaces the old one and only the current information is available. A learning system, instead, is able to recognize the current environment and to recall previously learned associated information from it memory [E. Ikonen, K. Najim 2002]. An adaptive system merely adjusts to its current environment. But a learning system is an adaptive system with memory. Learning means that system adapts to its current environment and stores this information to be recalled later. Thus, one may expect that a learning system improve its behavior with time.

3.4 The Integrated Intelligent Model

In this section, the integrated intelligent model and architecture for process diagnosis in industrial processes is presented and also the detail implementation approach is given.

3.4.1 Model-based via Intelligence-Based Diagnostic Methods

In process diagnosis problem in real industry, many methods have been developed based on different technology fields, such as *System and control theory*, *signal processing* and *artificial intelligence* (AI).

In control theory field, fault detection and isolation (FDI) model based method is developed for online diagnosis. Some methods, for example, diagnosis observers, Kalman filter, parity space equations, process identification [S.Simani, C. Fantuzzi and R. J. Patton 2002], have been investigated for fault detection and identification. All of these methods are based on the analytical redundancy inherent in dynamic process models. It means the controlled object must be a completely known and expressed as linear (nonlinear) model, for instance, with state space equation, transfer function equation and so on. But it should be not available in many cases, especially in nonlinear dynamic process with large operation range and many operation points.

Fault detection and diagnosis using neural networks is achieved by exploiting their no-linear pattern classification properties. The diagnosis ability of a neural net depends upon the discrimination of decision regions corresponding to various fault classes in the measurement space, a fault space is defined as the region spanned by the measurement data from the various sensors in a process.

The FDI based model and FDI based neural network are two fundamentally different methods. To two systems, FDI based physical model requires reasonably accurate physical model equations as prior knowledge and the FDI based neural network just needs significant amount of process history data.

FDI based on physical model is built based on first principles knowledge of the process in the form of model equations. For the faults that are already modeled, the FDI model generally dose quite well. However, its diagnosis may not always result in the unique identification of the underlying fault. Some demerits should be occurred due to the fact, namely, the availability of first-principles models may be limited for many processes or the development of such models may be quite difficult. The other drawback is to deal with the noise data. Another disadvantage is that FDI based physical model can only identify the faults that have been modeled already and might misclassify a novel abnormal situation.

As the FDI model based neural network, the advantage of it is that it dose not need first-principles models for its development and can work with historical process data. However, it is still limited by the availability of such data and extent and richness of process behavior that is exhibited in the data. With respect to the noise tolerance issue, it is more robust than FDI based physical model approach due to Neural Network's strong robustness performance. It also performs better on the novel fault identification problem.

As a summary conclusion, the performance comparison of two different kind methods has been investigated and given in Table 3.2. [R.Rengaswamy, D. Mylaraswamy, 2001].

Table 3.2: comparison of method of FDI based model and method of Neural Network

Criterion	FDI based model	FDI based Neural Network
Online computational	Low	Low
On-line computational efforts	Need accurate model	No accurate physical model
Novel fault identification	Poor	Fair
Robustness to noise	Fair	Good
Tuning	Bad	Good
Completeness	Good	Fair
Resolution	Fair	Good
Adaptability	Good	Fair
Range of application	Good	Bad

Both the FDI based physical model and FDI based neural network have their strengths and weaknesses, they also complement each other in these aspects. Hence, it is possible to combine these methods to make a good performance. Some applications have been done in the kind of methods, for example, using FDI based physical model detects fault occurrence and using neural network identifies the fault. Some investigations have been made recently in fault diagnosis in power plant using neural networks.

3.4.2 Why Intelligence Technology in FDI

Neural Networks (NNs) provide another way to implement the FDI model due to following factors:

Strong nonlinear classification ability

Neural networks (NNs) in fault diagnosis have been unusually exploited to classify measurement patterns according to the operation of the process. The classification method is typically an off-line procedure, where the fault model is first defined and data is collected. In this situation, certain measurement patterns correspond to normal operation and other patterns correspond to fault operations. The classifier should be capable of providing one of the following decisions:

- Normal: from a definition of normal process operation, through the specification of training patterns corresponding to the normal process behavior, the Neural Network should be able to determine that no fault occurred. This is essential to avoid unnecessary alarms due to process and measurement noise.
- Abnormal and known fault: if the observation pattern falls in the "proximity" of the training patterns of a particular faults class, then that fault should be announced.
- Abnormal and unknown fault: when the observation pattern falls "far" from the training patterns, the network should be able to say that abnormal behavior has occurred but it cannot determine that class to which the pattern belongs.

Approximation modeling ability

Fault detection and diagnosis using neural networks is achieved exploiting their nonlinear pattern classification properties. Neural networks actually perform an approximation, and properly a classification of static patterns, at the same time, the neural networks is also a good approximator, which can approximate function of real process with certain degree of accuracy. These features can be used for modeling real process under normal state at certain degree. The residual generator can be obtained by error comparison between output of real process and output of system model.

Strong robust ability to noise disturbance

A big advantage of Neural Networks in FDI problem is its good robustness to noise. In FDI based physical model approaches, noises factors have to be considered so as to make correct fault detection. However, the neural networks can deal with this noise due to its global nonlinear classification ability and universal approximation.

Fuzzy information processing and inference ability

The FDI intelligent model with fuzzy information process ability can provide better fault detection and isolation solution due to amount of fuzzy information existence in real life. At the same time, fuzzy Inference system can also be employed to analyze and diagnose the fault. It is also useful to extract and represent high-level knowledge of fault diagnosis by fuzzy information and inference rules.

Adaptive and Learn ability

Neural networks can adjust the model due to its adaptive and learn ability. These kinds of abilities can make neural networks model complex system with different environments change. In FDI problems, this ability can make the NNs model adapt change of real object and still provide a good classification (fault detection) result.

3.4.3 Recent Research in FDI Model Based Intelligence Technology

Recently, a number of researches have studied the application of neural networks to process diagnosis problems, some applications have been studied with a multiplayer feed forward network based machine state identification method, they represented certain fuzzy relationship between the fault symptoms and causes with the high nonlinearity between the input and output of the network; Ogaji SOT and Singh R [Ogaji SOT, Singh R, 2002] studied the application of ANNs for gas turbine fault diagnosis. The benefits derived from the application of ANNs was addressed in the research; Alexandru M involved neural fuzzy structure to learn the exact input-output relation of the fault detection process for induction motor, the first neural-fuzzy architecture maps the residuals into two classes: one of fixed direction residuals and another of faults belonging to velocity sensor. The second adaptive neural-fuzzy network will provide updated membership function of the set of fixed oriented residuals that better describe the fault diagnosis map.

A big challenge of application of neural networks is to treat the fault diagnosis of industrial plants at different working points. It was introduced and studied by [S. Simani, R. J. Patton, 2002]. A developed case shown that one neural network trained with data from a primary operating point can be successfully used to diagnose fault as the secondary operating points.

A combination of methods based physical model and based NNs was developed for fault diagnosis in power plant using neural networks [S. Simani, C. Fantuzzi, 2000] due to advantages of each approaches, in the way, a Kalman filter was applied to generate residual signal for fault detection and a NN model was applied for fault identification. These combinations of methods provide a good solution for complex dynamic system in respect of fault detection and identification.

Many different neural networks have been applied in these fields, for example. MLP, RBF and also Fuzzy logic system, at the same time, some hybrid systems of NNs, FLS and GA still have advantage to solve some special problems.

Theses above methods indicate that these methods based intelligent technologies can provide good solution in some aspects, especially under the condition of no clear and uncertain physical model of complex system. But it still has some problem in these fields, for example:

- Still difficult to treat different operation point and big change range
- Lack of quantitative analysis for fault identification
- Most of them were applied as residual generator with a crisp classification. (Either fault or not)
- Lack of the adaptive ability to improve model for fault detection and identification

There has been much effort recently in making a fusion of fuzzy logic and neural networks for better performance in decision-making systems. The uncertainties involved in the input description and output decision are taken care of by the concept of fuzzy sets, which the neural net theory helps in generating the required decision region. There is a large and growing body of literature pertaining to the "joning" of neural nets and fuzzy systems. Many of these papers discuss:

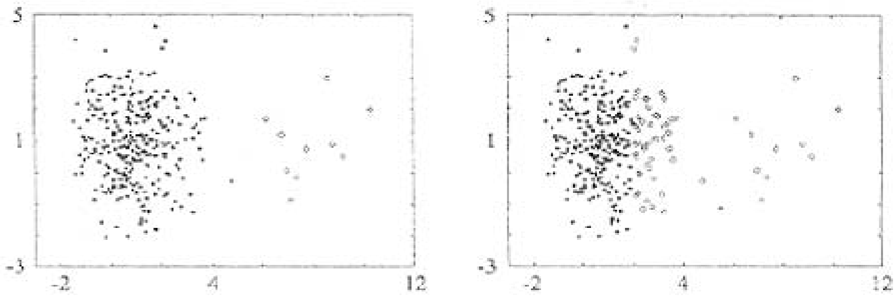
- Using a neural net to "turn" the fuzzy system [Gupta, M. M. and Rao, D. H., 1994][Brown, M. and Harris, C., 1994]
- Combing neural network and fuzzy system, into so-called hybrid systems
- Using a neural net to approximate a fuzzy system

According to above literatures in research and application fields, three main schemes that the Neural Network and FLS are applied in FDI modeling for fault detection are:

- ***Fault classifier***: In this scheme, neural networks play a fault classifier so as to detect fault occurrence according to classification of the input feature vector in decision space. It could reach good classification ability and performance if the neural networks is trained well, and the classified sets (fault set and normal set) has explicit and clear border in multidimensional decision space. Namely, if fault set (category) has explicit border with normal sets (category), the neural networks can easily classify them due to its good nonlinear classification ability. Under this circumstance, fault classifier based on neural networks can be applied to detect the fault occurrence well. Some applications based on the schemes were introduced in [Karpenko M, Sepehri N, 2002][Chen YM, Lee ML 2002][Calado JMF, Korbicz J,2001][Lee IS, Kim JT, 2003][Rodriguez, C, Rementeria, S, 1996][J.B.Gomm, 1998] [Rasmussen, M. et al., 1993].

- ***Approximate models***: In this scheme, the neural networks play an approximate function to map the input-output relationship of the real object. In approximation theory, the neural networks are universal approximators in the sense that they can approximate any function to any degree of accuracy provided that there are enough hidden layer neurons. Hence, if the trained data are perfect to represent the real process characteristics and the Neural Networks is trained enough well, it should be a good modeling tool to represent the real plant. Once the neural networks plays approximate model at perfect degree of accuracy trained by process data in normal state, it can be employed into FDI problem for fault detection. If the error between output of real plant and output of neural network model is so bigger than a threshold, a fault can be confirmed.
- ***Fuzzy system and expert system***: the process analysis and diagnosis can be done based on the fully understanding of process behaviors, namely, if we have good understanding and experience to the real process, it is very useful to build a sets of diagnosis rule (If-then) for process diagnosis and analysis. Under the circumstance, fuzzy expert system is good strategies for FDI objects. At the meantime, some intelligent models, for example, Fuzzy inference system, Adaptive Neural Fuzzy System (ANFS), Knowledge Mining algorithms etc, can extract diagnosis rule automatically from learning from data. These rules build the relationship between process symptoms and failure occurrence. Some applications were introduced in [Javadpour R, Knapp GM, 2003] [Balle, Peter, 1998] [Dietz, W.E., Kietch, E.L., and Ali, M 1989] [P, Thomas; A, Mihiar, 1997] [Zhang, Jie; Morris, 1994] [Caminhas, W.M.;Tavares, H.; Gomide, F 1996].

However, both schemes of fault classifiers and approximate models based neural network function work well under the certain conditions and premises. To the scheme of fault classifier based neural networks, if the fault sets in multidimensional space have certain and clear border with normal set or other sets, it is easy to classify these faults with neural networks, for instance in two dimension space, see Figure 3.6(a), if these borders among faults sets and normal sets are uncertain or vague, for instance in two dimension space, see Figure 3.6 (b), the fault detection in FDI problem cannot be treated as a simple classification problem, the classification degree of belonging to different sets should be considered. To the scheme of approximate models based neural network, if the approximate neural network model possesses a certain error to represent the real object due to without good neural network structure or lack of enough and good feature trained data, (this situation is common in real application), the fault detection based this scheme should probably make some error. Hence, neural network has some disadvantages in fault classification and fault identification when it is applied to solve the complex problems. To the fuzzy inference System, it is also up to the correction of expert's knowledge and decision rule. Some researches and applications concerning fuzzy classification for fault identification have been done as [Applebaum, 2001][Q, Shen, C, Alexios, 2000] [Chen, Yubao, 1995][Huallpa, B. N, N, Euripedes; al., 1998].



(a) (b)

Figure 3.6: Two different forms of fault and normal sets in feature space (2D)

(a) Classified sets has clear or explicit border between them
(b) Classified sets has uncertain or vague border between them

In fact, in real process, some big challenges here are listed as follows:

- Real system or processes have more nonlinear characteristics, it is difficult to obtain their exact mathematic models
- Almost, it is difficult to explicitly define a fault, for example, seal ring problem, what degree of leakage is fault state? It means the fault state is no clear and crisp border with normal state. Hence, it is difficult to solve these kinds of problems with simple classification way.
- Most FDI methods based intelligent technologies focus on the fault detection; it is lack of fault identification and quantitative analysis.

Hence, it is significant to further research in this field so as to solve above problems well. It is also main goal in this thesis from technology point of view, namely,

- Fuzzy classification for fault detection
- Fuzzy information and inference system for fault quantitative identification, and
- Fuzzy decision-making

3.4.4 The Design Objectives of the Integrated Intelligent Models

The main task of the model is developed to detect the process state (fault state) and quantitatively identify it. According to FDI function division, the main function of integrated intelligent model consists of three parts as follows:

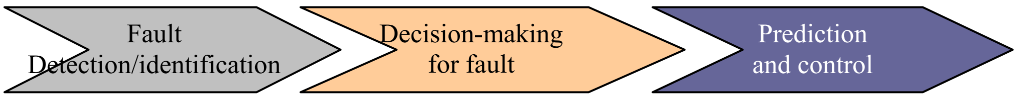


Figure 3.7: The main function of integrated intelligent model

- *Process state (Fault) detection and identification*: detect which process state the process is on or if there fault occurred and quantitatively identify process state (fault), namely, the fault amplitudes
- *Decision-making:* discover the decision rule of fault occurrence and accompanying phenomenon (relation between symptoms and fault) for process diagnosis

- *Process prediction and control:* predict process behavior based on process state detection and identification, process control based on process prediction result

In this paper, the research work focuses on modeling with intelligent technology in order to realize above main functions:

1. **Residual generator based on fuzzy classification**. A classifier based on fuzzy classification is needed to realize fault detection due to no explicit discrimination border between normal and fault state in real life. When a system or process works on certain operation point, the performance of the system or process fluctuates on certain range. Once a fault or abnormal state occurs, the one kind of performance of process should make the corresponding response. Hence, the process state fluctuates beyond initial range and these process states can be divided into at least two different states: *abnormal state* and *normal state*. It is illustrated as Figure 3.8 by simple description. Fault detection is then implemented by classifying these normal or abnormal states, however, state definition cannot be done by explicit way, so, fuzzy information is employed and a classifier based fuzzy information process is also needed as the residual generator. When abnormal state is detected and beyond a threshold, it indicates a fault is occurred.

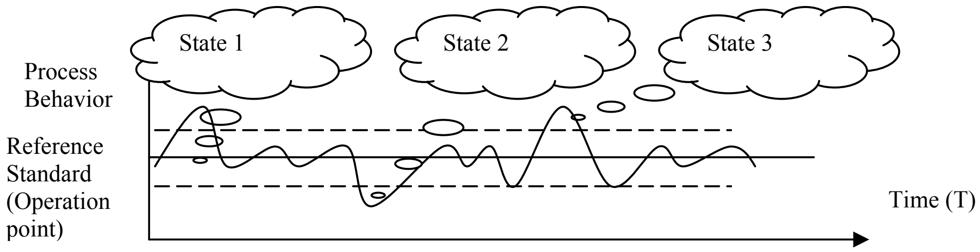


Figure 3.8: The possible process state in industrial process.

2. **Fault identification based on fuzzy information.** A fuzzy classifier is implemented for state classification, the result of fault classification still includes fuzzy degree of fault occurrence, and hence, it is natural to involve this fuzzy information to describe fault identification so as to improve the accuracy of fault identification.

3. **Knowledge discovery and rule extraction for making decision**. The model is also used to seek and analyze the essential underlying causes and relation between process state change and input symptom. For example, the relationship between symptoms and fault occurrence. The output result cannot only reveal characteristic of complex system but also can present these characteristics with friendly knowledge way so as to provide diagnosis, analysis and making decision.

4. **Process prediction and control**. Based on process state definition and description with fuzzy information, as well as certain system adaptive scheme, model prediction and process control are developed and implemented so as to improve model prediction and control accuracy.

The model based on above goals can be widely applied in industry fields, for example,

- Process Performance Measurement, especially in dynamic characteristics
- Process state identification

- Aberrant behaviors detection and process risk monitoring;
- Prediction for process behavior and performance
- Intelligence analysis, risk analysis, intelligence diagnose and making decision
- System identification and process control with intelligence approaches…

3.4.5 The Design Ideas

The integrated intelligent model for FDI problems here is developed based on some following ideas:

From dynamic system model point of view

As introduced before, to a dynamic system, it has been proved that any nonlinear, discrete, time-invariant system can always be expressed by a NARX model [V. Kecman, 2001] as equation (3.8~10) after making some moderate assumptions.

The NARX described: the output of system is determined by the combination of the process response output produced by current system inputs and historical behaviors. It means the real system output is determined by both the system response yielded by current system inputs and historical behaviors. Hence, in order to get accurate system output, for fault detection problem, it is natural to consider above two kinds of factors, namely, system input and system historical behavior. Both of them have difficult influence on the real system output. The real system output is determined by their combination with different weight average.

From process and system theory point of view

Industry processes can be regarded as complicated controlled objects or systems. These processes always operate on certain state or range. The process state could change from one state to another state due to different system input, inner parameters and operation conditions, for example, operation point change, fault occurrence, etc, More information about process characteristics emerge out during the process state change. It is significant to research the process characteristics when the system's state is being changed or system in change process. Hence, defining process state, which is related to the fault state, and identifying state change for the controlled object is the one important and key idea in the model.

Obviously, there are no certain and explicit ways for state definition due to different processes and control objects. But, it still can be defined by further abstraction and extraction of characteristics of object behaviors. For example, in fault detection problem, the process state can be defined as normal state, fault state (fault occurrence), or from risk analysis problems, it can be defined as high risk state, medial risk state and low risk state, etc.

To all input variables, they also can be defined as the corresponding state to express different symptoms, for example, in fault problems, they can be defined as normal input data, high abnormal input data or low abnormal input data for certain variables, etc.

To simplify system model, we can give an assumption or premise in order to reveal (map) the relationship between inputs symptoms and fault occurrence. To a process on certain operation conditions: *If the process state is on abnormal state (fault state), it could be due to one and more of input variables with the abnormal input data (some symptoms) or influence from process historical behaviors.*

It means fault occurrence always accompanies the change of input variables from one state to another state, or certain influence from process historical behaviors, namely, the fault occurrence always accompanies some certain symptoms.

After the model has abilities to define or discriminate different states based on the input-output data of system, it means the fault state can be detected with judgment of process state, the different function units are developed for further dealing with fault identification and quantitative analysis based on different process states. For example, when the system is in aberrant state, the corresponding unit can function for the aberrant state process. The model can switch to different units for different system state. The whole system output is sum of outputs of all different state processing units based on fuzzy inference model architecture as following Figure 3.9.

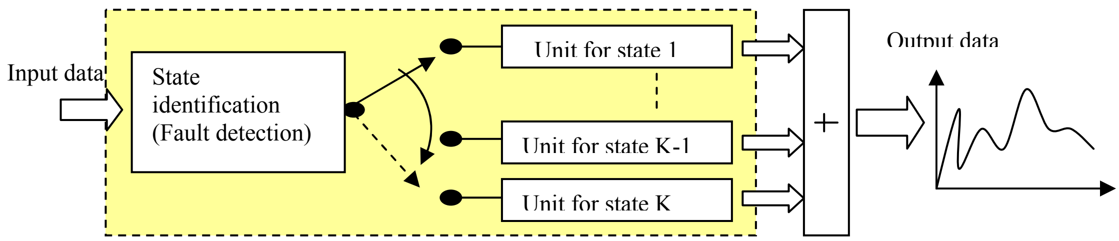


Figure 3.9: The partition of process states identification and process

The main goal to divide whole operating process into different process states is based on following reasons:

1. Process behavior can be treated according to the different process states respectively. Different sub models can be applied in the different process states in order to ensure more precise response.
2. To Fault Detection and Isolation (FDI) problems, in essential, the change of process states from normal to abnormal state means one kind of fault occurrence on certain degree. It provides a good way to fault detection and identification with these state identification and fuzzy information.
3. The underlying rule between symptoms and fault occurrence can be captured and analyzed through these data in diffident process state. They can be used for building the relationship between input data (symptoms) and fault occurrence (fault) and also discovering inner rule embedded in them. These knowledge and rules are very useful to fault diagnosis, process performance analysis and making decision.

From intelligence technologies point of view

In essential, to a process, many factors can trigger a state change (fault occurrence), for example, input signal, system historical behavior, inner elements of system, fault of process, environment change, noise signal, etc, they are shown in Figure 3.10. It is very difficult to exactly know these relationships among theses factors and system response due to the system's complexity and nonlinear characteristics. It is also difficult to get the exact mathematic expression to describe these relationships. Hence, a black-box based strategy is applied for modeling this kind of system so as to reach certain purposes.

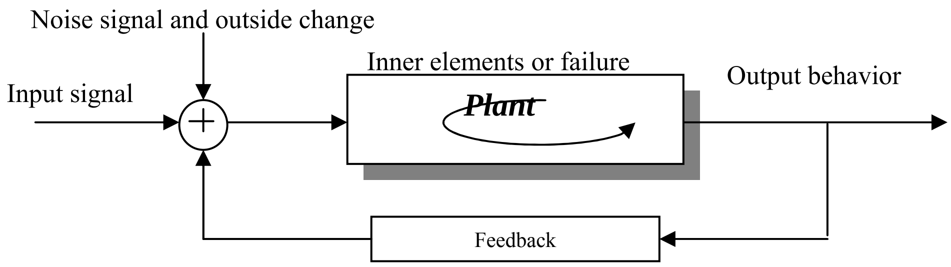


Figure 3.10: All factors to effect process behavior in complex processes

To complex processes or systems, some intelligent methods are involved in order to establish the mapping relationship between inputs and output (in fault problem, it is symptoms and fault occurrence) through learning from history event and sample input data. These mapping relationships are embedded in model in order to identify and judge what kinds of response occur when new input variables arrive.

Neural Networks, regarded as good tools, is used to build model and map the relationship between input-output of object. It is a black box approaches to describe real object through training input-output pairs, after training, the real object is presented with network structure and weight values embedded on neural structure. It is difficult to use these weights to explain real plant because they do not directly represent inner laws of real plant, but it can express the characteristics of real object with different form in high dimension space. It is easy to build model but difficult to understand the model. FL is a white box approach to model complex object, whose inner control rules have been obtained in advance. Hence, it provides a high–level description for the object. It is easy to express and understand FL model but difficult to build model for complex system due to no much prior knowledge or fully understanding inner laws in advance. Data mining plays an important role to discovery knowledge from massive data. It builds a bridge to link NN and FL model. Their relationship can be summarized in Figure 3.11. The NNs can be used to model real process by learn from data, Data mining is used to discover inner rules and FL is used to express the object rules or control real object with certain rules.

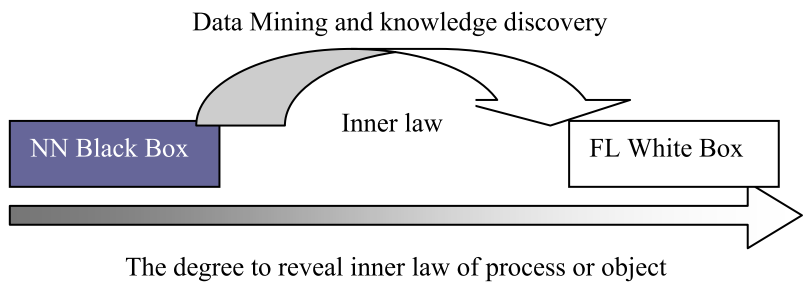


Figure 3.11: The relationship among NN, Data mining and FL

On the other hand, it is also impossible to make a quantified description and clear definition of the process state on a large degree. Hence, fuzzy sets and fuzzy inference methods are involved to treat the uncertain problem. Based on fuzzy information, process state can be defined and described with fuzzy sets, and further quantities process is treated so as to get accurate output. The whole architecture of intelligent model in this paper is built on these above basics and ideas.

From Neural Networks point of view

In essence, modeling a plant with intelligent technologies is a process to find universal approximators in the sense that they can approximate any function to any degree of accuracy in high dimension space, for example, to NNs an FL models, if there are enough hidden layer or rules [V. Kecman, 2001].

Neural Networks is robust and noise tolerance. It represents the real process with network structure and weight value embedded on neural structure. From NNs theory point of view, NNs model can approximate any function to any degree of accuracy in high dimension space. However, in real application, the NNs only reach approximate function due to limited layers, neurons and trained sample data. Hence, the weight values, which are determined by sample data, cannot represent all training data characteristics at certain degree. On the other hand, not all train data have same influence on these weight values. It can be illustrated with simple example in Figure 3.12 through analysis of NNs training process.

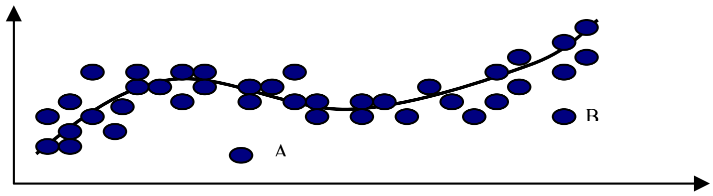


Figure 3.12: The approximate map function relationship in NNs

In Figure 3.12, the approximate function presented by a curve can be established according to the training pairs in two-dimension space. It is clear to know that not all points have same contribution to build the function. Some points, far away the curve, for example, point A or pint B, or some noise data, just has weaker influence on building the function than other points closer the curve do. It can be deduced by the NN's training phase [Hung T. Nguyen Nadipuram R.et 2002]. This is why Neural Networks has strong robustness of noise tolerance. Based on this feature of NNs model, NNs should ignore or fade some training data, which are far away from approximate function in multidimensional space, for example, noise data, aberrant data in industry process, etc. (Because most data are normal data, just less data are aberrant data in a normal process). It means the NNs model can give a good approximation to global characteristics represented by whole training data set, but is weak to express local characteristics represented by part training data set. This is also the main reason and difference between adaptability and learning ability.

As we known, only is training data enough well to represent the real process or plant, the trained NN models can have good characteristics or performance to approximate real plant. Hence, if the whole training data is classified as different data sets, which have different process characteristics as training data to train corresponding NNs models respectively, the results of the integrated NNs model should be better than that one NNs model trained with whole data. It means using several NN models for real plant should be better to express real process that one NN model dose.

Based on above analysis about the Neural Networks, some basic conclusions can be drawn as follows:

1. Most processes include many input-output variables. These processes can be described with approximate function relationship between input-output variables in multidimensional space. Hence, finding the approximate relationship between input-output spaces is main task to build process model.
2. In theory, it is possible to find an approximate function relationship to depict a process through learning from data if these data can fully represent real process. Some algorithms and models have also been developed for these goals and the approximate relationship can be gotten with certain cost of error and risk. Many models, for example, NNs and FL models have strong robust to deal with the kind of problem. These algorithms and models have good characteristics to express global relationship but poor to deal with abrupt change relationship in high dimension space.
3. Under some circumstances in industry field, the information about abnormal or abrupt section is more interesting in analysis of process, especially in fault detection problems.

3.4.6 Outline of Integrated Intelligent Model

Based on the above design ideas and objectives of the intelligent model, the integrated intelligent model consists of five units. Each part has different functions and sub-models. The outline of the model is shown in Figure 3.13 and the work process and the functions of five units are introduced as follows:

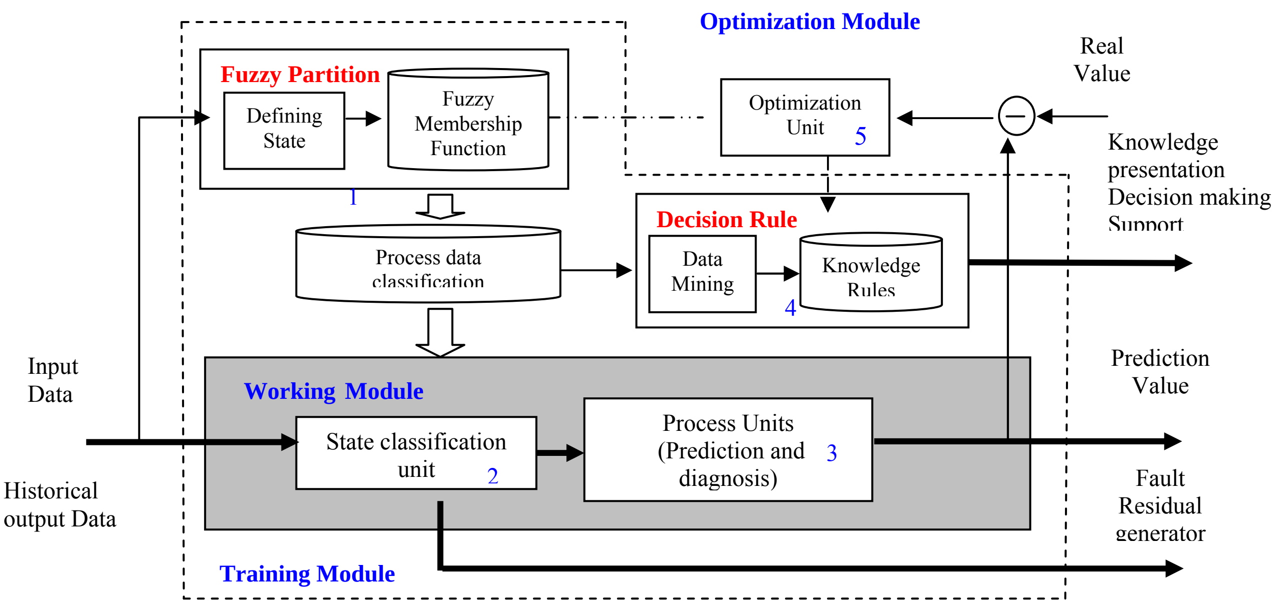


Figure 3.13: The Architecture of Integrated Intelligent Model

The model is built based on black-box principles. It meant it is built through learning and from history data. It consists of three modules, 1, *Training module*, which is used for build model by learning from historical data. 2, *Working module*, which functions for process diagnosis and prediction. 3, *Optimization module*, which is responsible for optimal model and model adaptability. Whole model work consists of two phases, training and work phases. In training phase, 1) the input-output data are fuzzfied and different process states are defined by some statistic principles via from historical data; 2) a state classifier is trained as fault residual generator by these fuzzfied input-output data; 3) Based on the different process states, fault diagnosis or prediction model could also be established; 4) Some underlying rules or laws between fault symptoms and process state(as well as fault state) are mined from input data in different process states and corresponding output behaviors, these rules are essential resource for intelligent diagnosis for process behavior. In work phase, when a new input data is coming, state classifier first identified which process state occurred (if there a fault occurred) and then corresponding sub units are called to quantitatively identify process behaviors. The basic functions of each unit are introduced as following Table 3.3.

Table 3.3: The basic units and their functions in integrated intelligent model

Number	Sub Model	Function
1	State definition unit	Define the membership function for input/output variables (symptom and process states)
2	State classification unit (Fault residual generator)	It is trained as fault residual generator for detecting fault occurrence, as well as process state.
3	Fault identification (Diagnosis/prediction)	Quantitatively diagnosing and identifying process behavior.
4	Diagnosis Rule generation	Data Mining underlying rule between symptoms and different process state (fault).
5	Model optimization	Optimizing model and adaptively

3.4.7 Mathematical Expression of Work Module

Based on the design ideas and outline, the process diagnosis and prediction model developed in this chapter can be explained as a fuzzy NARX TS dynamic model, which can be used to detect fault and also give a quantitative identification for process behavior. The mathematical expression of work module is given in Figure 3.14. In figure, the input 1 represents all relevant process input variables and the input 2 for relevant history output.

Process state (fault) detection

The output ***w*** of model can be used to detect if a fault or abnormal process state occurs or not. It plays the same function as residual generator in FDI based model and identification (presented in chapter 2). The bigger the *w* value is, the more possibility the fault occurs.

Process behavior diagnosis and prediction

The output ***z*** of model can be used to quantitatively identify process output behaviors. The value can also be used to evaluate and predict process output.

The model detects fault and predicts process behaviors in terms of two important aspects: *process input variables* and *historical output (behaviors)*. The final detection result for process state (or fault) is generated by the combination of influences of real system input and historical process behaviors. The process behavior is also influenced by the two factors. Whole model not only provides a residual generator for process state (fault) detection, but also gives a quantitative identification of process behaviors. It can be described as a NARX Takagi-Sugeno fuzzy dynamic model as equation (3.26) and Figure 3.14.

Rj: if $z_1(k)$ is $A_{1,j}$ And $z_2(k)$ is $A_{2,j}$

Then

$$\hat{y}(k) = \sum_{i=1}^{n_a} a_i^j y(k-i) + \sum_{i=1}^{n_b} b_i^j u(k-i-n_d) + c^j \tag{3.26}$$

Here,

- $z_1(k)$ is process state (fault) fuzzy variable produced by real system input, $A_{1,j}$ is fuzzy sets defined by the membership function (for example in fault problem, fault state, normal state, etc…)
- $z_2(k)$ is process state (fault) fuzzy variable produced by historical output, $A_{2,j}$ is fuzzy sets defined by the membership function (for example, in fault problem, fault state, normal state, etc…).

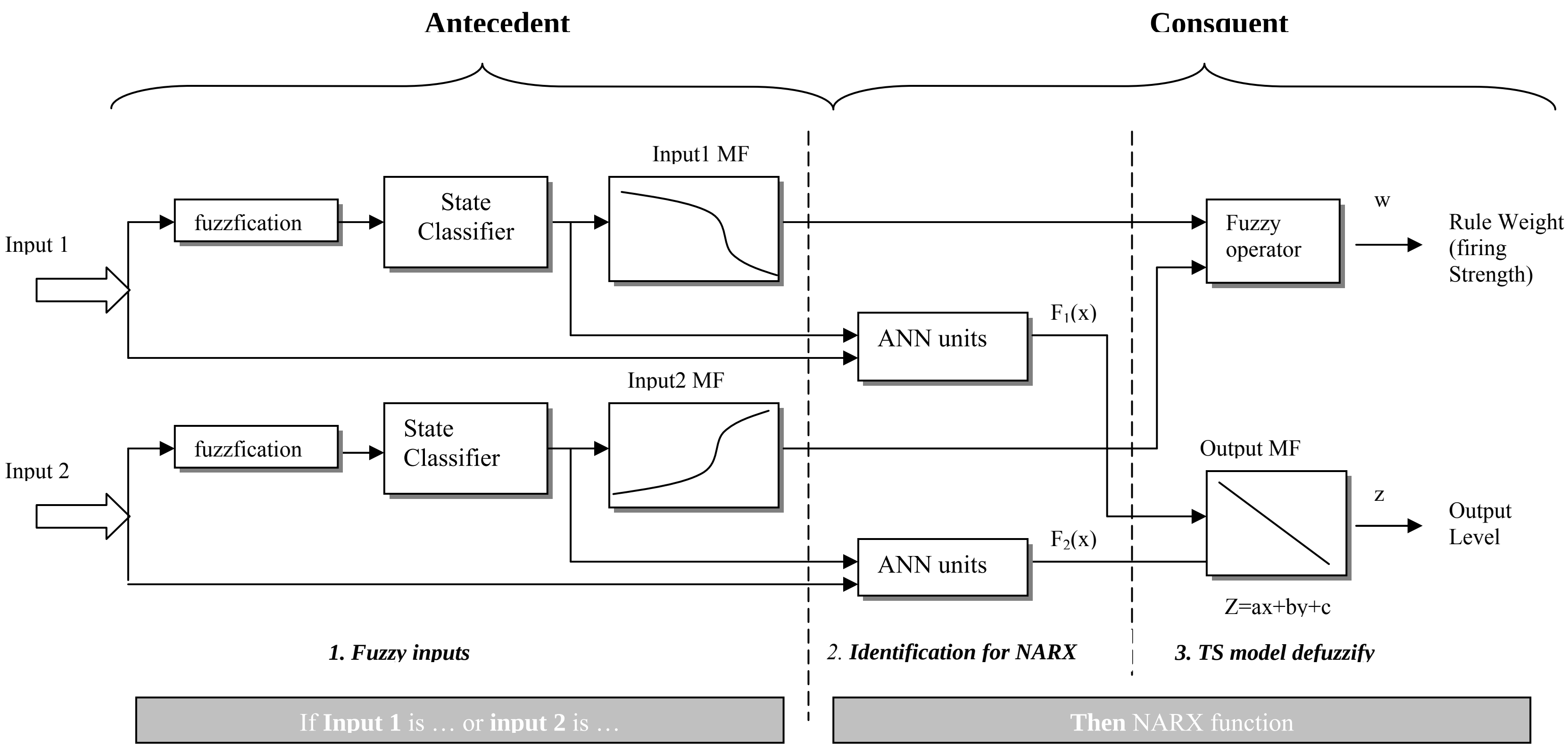


Figure 3.14: The system model of work unit based on mathematic view

1, input 1: real system input; input 2: process historical output variables (process historical behaviors).
2, input 1 MF: membership functions for process state variables produced by input 1, and Input 2 MF: membership function for process state variables produced by input 2.
3, $F_1(x)$, and $F_2(x)$: the model output value based on certain process state and different input condition.

4, R_j: if $z_1(k)$ is $A_{1,j}$ And …And $z_n(k)$ is $A_{n,j}$ Then $\hat{y}(k) = \sum_{i=1}^{n_a} a_i^j y(k-i) + \sum_{i=1}^{n_b} b_i^j u(k-i-n_d) + c^j$ (inference Rule).

here, the whole process can be described as a fuzzy NARX TS inference system with two input variables, $z_1(k)$ *and* $z_2(k)$, under the circumstance, the fuzzy rule can be deduced from high-level description of process as follows:

To input 1(only consider real system input variable):

The whole model is a fuzzy TS model described as equation (3.14) and Figure 3.15.

Rule 1: If input 1 produces abnormal (or fault) state, then whole process is in fault state, and the output is Z_{11}.

Rule 2: If input 1 produces normal state, then whole process is in normal state, and the output is Z_{12}.

To input 2(only consider process historical behavior):

The whole model is a fuzzy TS model described as equation (3.14).

Rule 1: If input 2 produces abnormal (fault) state, then whole process is in fault state, and the output is Z_{21}.

Rule 2: If input 2 produces normal state, then whole process is in normal state, and the output is Z_{22}.

The fuzzy TS model for input 1 can be illustrated in Figure 3.15 as follows:

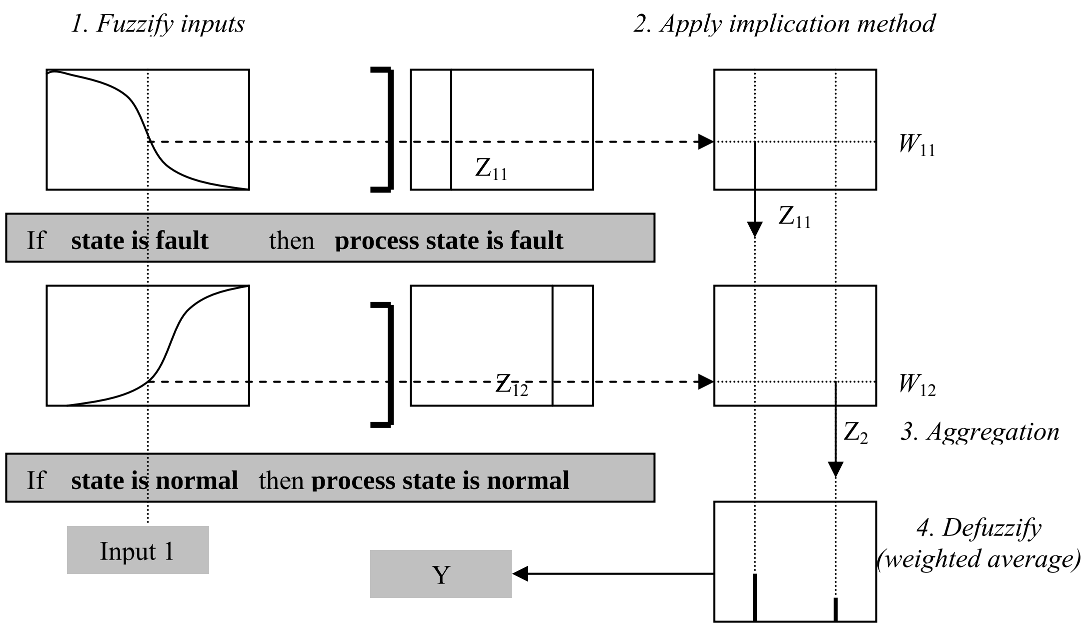


Figure 3.15: The fuzzy TS inference model and model output for variables input 1.

Here, the Z_{11} indicates the process output value when the process state is in abnormal (fault) state, and Z_{12} is the process output value when process state is in normal state. The system final output (only input 1 is considered) is given as equation (3.14) as follows.

$$Y = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i}$$

Here, w_i is fuzzy degree [0, 1] for different state, which are produced by state classifiers based fuzzy NN model. In this case, $\sum_{i=1}^{N} w_i = 1$, the final output produced by variable input 1 is given below:

$$Y_{input1} = \frac{\sum_{i=1}^{N} w_{1i} z_{1i}}{\sum_{i=1}^{N} w_{1i}} = \sum_{i=1}^{N} w_{1i} z_{1i} = w_{11} \times Z_{11} + w_{12} \times Z_{12} \tag{3.27}$$

The same principle as the variables input 2. It output value is given in equation (3.28) when only input 2 is considered:

$$Y_{input2} = \frac{\sum_{i=1}^{N} w_{2i} z_{2i}}{\sum_{i=1}^{N} w_{2i}} = \sum_{i=1}^{N} w_{2i} z_{2i} = w_{21} \times Z_{21} + w_{22} \times Z_{22} \tag{3.28}$$

Then whole integrated model combining variable input 1 and variable input 2 is depicted as a fuzzy TS NARX dynamic model. Its inference rule, system model and system output are illustrated as Figure 3.16.

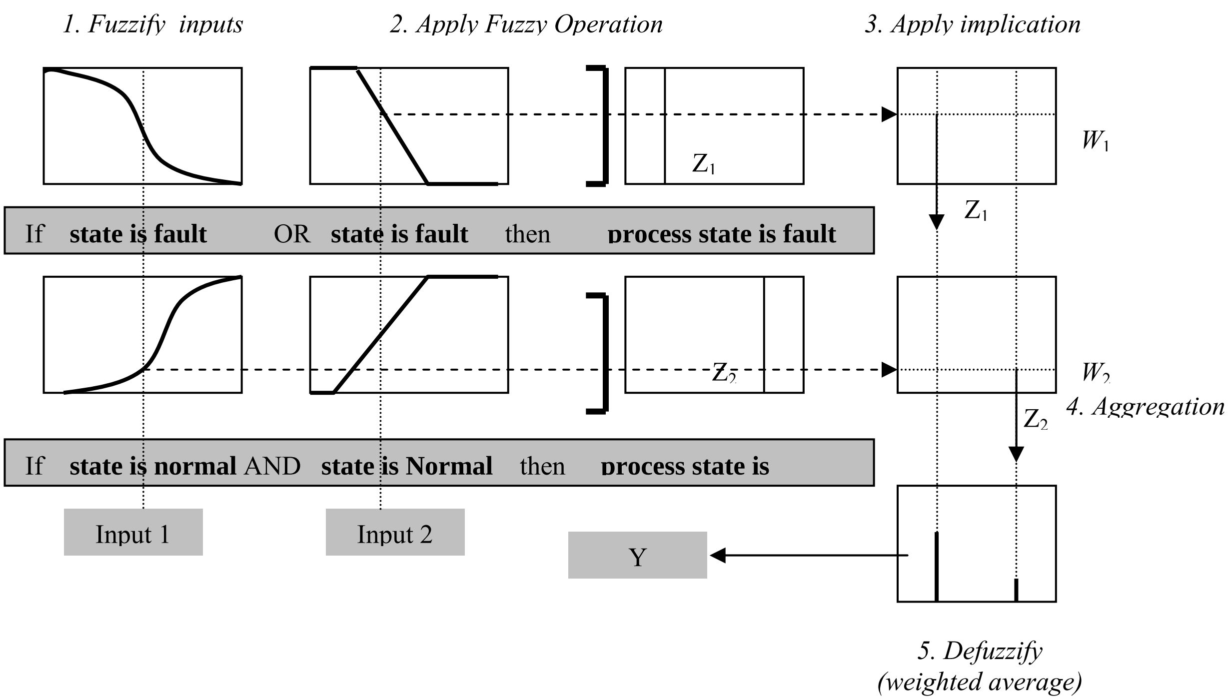


Figure 3.16: The fuzzy TS NARX inference model and its output.

The function Z_{11}, Z_{12}, Z_{21}, Z_{22}, are determined by system identification method with corresponding NNs model according to different process states and input variables, they are defined as follows Table 3.4:

Table 3.4: The relationships among function unit, input variables and process state

Input variables	Process State	Function
Input 1	Abnormal (Fault)	Z_{11}
Input 1	Normal	Z_{12}
Input 2	Abnormal (Fault)	Z_{21}
Input 2	Normal	Z_{22}

In term of the Fuzzy TS model Figure 3.14, the model output produced by input 1 and input 2 is determined according to the fuzzy TS model as equation (3.29).

$$Y_{total} = a \times Y_{input1} + b \times Y_{input2} + c = a \times (w_{11}Z_{11} + w_{12}Z_{12}) + b(w_{21}Z_{21} + w_{22}Z_{22}) + c \qquad (3.29)$$

where, a and b can be decided in term of the affection degree of input 1 and input 2 to whole process output, if input 1 (real system input) and input 2 (system historical behaviors) play the same role and degree to affect real process, then a=b=0.5.

According to the system structure, the fault residual generator is generated by combination (Fuzzy logic "Or" operation) of the fuzzy degree value of $A_{1,j}$ and $A_{2,j;}$ the quantitative identification value of process output is determined by the output value of NARX Takagi-Sugeno Fuzzy model of dynamic model (equation 3.26), namely *y(k)* is determined by the way of system identification with NNs model approaches. Hence, the whole integrated intelligent model for process state (fault) detection and process output prediction can be descried as follows:

Fault Residual Generator: the final fault residual signal is generated by output of two Fuzzy classifiers, one Fuzzy classifier is used for fault detection measured from real system inputs and another fuzzy classifier is used for fault detection measured from historical behaviors. The outputs of two fuzzy classifiers are combined by fuzzy operation to produce final fault residual signal. If the residual signal is beyond a defined threshold value, then a fault event can be detected by this system.

Fault Identification: the final process output is determined by the combination of two aspects, one is from process behavior output resulted from real system input, and another is from process behavior output resulted from historical behaviors of real process. These two kinds of process outputs are integrated by Fuzzy TS model so as to produce a final proper output. The process characteristic in different process state is determined by system identification with NNs model way.

Summary, in this Fuzzy TS inference model, the *Antecedent part* of inference rule, which is generated by two fuzzy classifiers**,** constructs the residual generator for process state (fault) detection. The map relationship between input-output of fuzzy state classifiers describes the relationship among system input (fault symptom) and process state (fault occurrence). And the *Consequence part* of inference rule, which is generated by system identification based module Neural Networks structure and fuzzy partition information, constructs the process prediction output. These inference rules based on fuzzy TS model describe the relationship between fault occurrence and its quantitative value.

In essence, the whole model consists of two Fuzzy TS inference system, one is used to deal with fault problems produced by real system input, another is used to fault problems measured by real historical behaviors. The combination of there two fuzzy TS system constructs a Fuzzy NARX TS dynamic inference model for fault detection and identification (Equation 3.26 and Figure 3.14).

3.5 Implementations for Intelligence Models

The implementation approaches of all above steps are addressed in this part in details.

3.5.1 Fuzzyfication for Input/Output Variables

The process behaviors are affected by many factors, for example, the parameters of elements inside process, environment change, operation change, noise input and input signal etc, hence, the first step is to select feature input vector and then determine which variables has more influence on process behaviors and which of them will be involved in the process diagnosis and analysis. According the model structure, the main two kinds of variables should be considered: one is the system real input variables and another is process historical output variables (process historical behaviors).

Usually, it is not easy to give an accurate definition for real process states for real plant, the boundary of process states normally have no exactly discrimination. They are uncertain and imprecise in value. The main work here is how to determine process state (in fault problem, it is fault state) by fuzzy sets and membership functions of input-output variables so as to describe real process states.

The fuzzy sets and membership functions of input/outputs variables are used to describe process performances, for instance, the membership function of input variables are usually used for definition or description of process symptoms in fault problems, and the membership function of output variables are used for definition or description of fault state in fault problem, as well as risk analysis parameters, etc.

Determining for membership functions of input-output variables

There are many methods to determine the fuzzy sets and membership functions for all input-output variables under the different circumstances and operation condition. For example, probability-distributed function, mean values/expectation value in statistics disciplines and fuzzy cluster algorithm, etc. Here, mean and expectation value in statistic discipline are introduced to make the membership functions for describing different process attributes and states. These membership functions are decided just by observation data collected during whole process operation.

As proposed in [K. Evanglos, 2000], the observation data of each attribute is viewed as a time series, the duration of the training period is M time granules and it is called training session. During the training session ever signal value of each attribute is observed and registered in a time sequence. Figure 3.17 shows the time series observation by observing the attribute a_i during a training session of M time granules. For each attribute, three values are maintained at the end of the training session. Theses are the minimum value (a_i^{min}), the maximum value ($a_i^{\max}$) and the mean value (a_i^{mean}) of the attribute a_i, which are defined as follows:

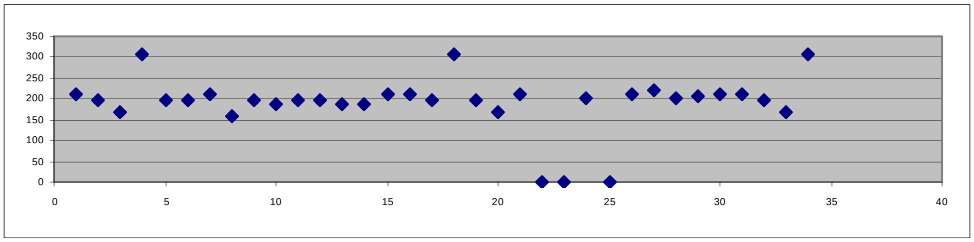


Figure 3.17: The time series data during T duration

$$\alpha_i^{\min} = Min_{j=1}^{M}\{a_i(j\tau)\} \tag{3.30}$$

$$\alpha_i^{\max} = Max_{j=1}^{M}\{a_i(j\tau)\} \tag{3.31}$$

$$\alpha_i^{mean} = \sum_{j=1}^{m} \frac{a_i(j\tau)}{M} \tag{3.32}$$

The aim of training is to extract fuzzy sets from observation data. Each fuzzy set describes the different input data or process output state. The idea of fuzzy definition based on historical data for these process states is introduced as follows:

The value of a_i, which are within the interval $[a_i^{\min}, a_i^{\max}]$, are considered perfectly normal, which other values of a_i are considered abnormal. The level of abnormality is expressed by a fuzzy membership function. Which assigns a number between [0, 1] to such the abnormal value of a_i. The more distant the value of a_i, is from the interval $[a_i^{\max}, a_i^{\max}]$, the more abnormal the value of a_i is.

The membership function is defined with parameters, $a_i^{\min}, \vartheta, a_i^{\max}$ value and a threshold $0 \le \vartheta \le 1$, the ϑ determines how the intensity that characteristics abnormal values changes with respect to a_i, the fuzzy set A_i, which describes the regularity of the values if the a_i, Even though the membership function of the fuzzy set A_i for different state may be any function, the triangular membership functions are employed in the paper for the sake of clarity. The triangle A_i is formed by considering the following points $(a_i^{\min}, \vartheta)$, $(a_i^{mean}, 1)$ and $(a_i^{\max}, \vartheta)$ in following Figure 3.18.

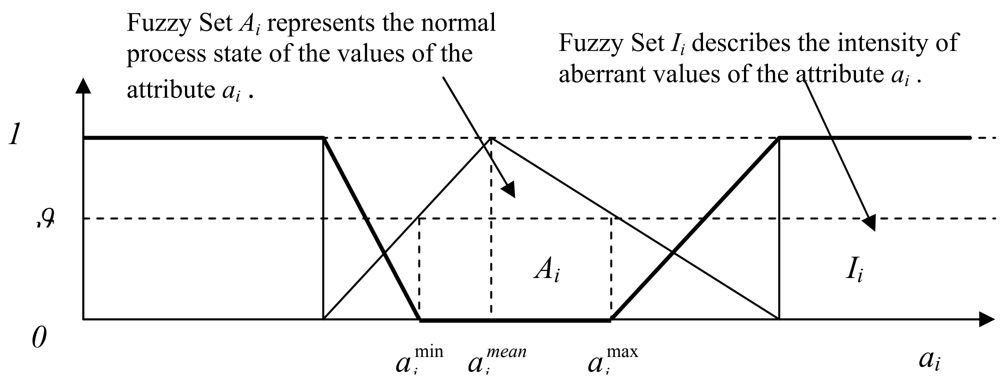


Figure 3.18: Fuzzy set A_i and I_i

When 0< ϑ <1. A_i is used to identify normal values of the attribute a_i and I_i for abnormal values of the attribute a_i, the majority of regular values of a_i are between [a_i^{max} , a_i^{max}], the more distant a value of a_i is from the interval [a_i^{min} , a_i^{max}], the more intense the aberrant value is.

The fuzzy set I_i that describes the intensity of an abnormal value of a_i is defined by using the following membership function:

$$I_i(a_t(t)) = \begin{cases} 0 & \text{if } \mathrm{A_i}(a_i(t)) \geq \vartheta \\ 1 - \dfrac{A_i(a_i(t))}{\vartheta} & \text{if } \mathrm{A_i}(a_i(t)) < \vartheta \end{cases} \quad 0 \leq \vartheta \leq 1 \qquad (3\text{-}33)$$

$$I_i(a_t(t)) = \begin{cases} 0 & \text{if } \mathrm{A_i}(a_i(t)) > \vartheta \\ 1 & \text{if } \mathrm{A_i}(a_i(t)) = \vartheta \end{cases} \quad \vartheta = 0 \qquad (3\text{-}34)$$

where a_i (t) is the value of the attribute a_i at time t, $I_i(a_i(t))$ is the membership value that expresses the level of abnormality of a_i (t) and $A_i(a_i(t))$ is the membership value that expresses the level of regularity of a_i.

When ϑ =0, the membership functions of A_i and I_i is shown as Figure 3.19, in this particular case, the I_i fuzzy set become a crisp set, it means all values of a_i are outside of the interval [a_i^{min} , a_i^{max}] are identified as fully abnormal (Figure 3.19.a), the system is fully sensitive, which means that if a value appears to be outside the interval [a_i^{min} , a_i^{max}], no matter how far away, it is considered abnormal with maximum intensity. If ϑ=1, then, for all t, $A_i(a_i(t))$ =1, and , $I_i(a_i(t))$=0, it means that none of the values of a_i as abnormal values (Figure 3.19 .b) , the system is fully indifferent. None of the values is identified as abnormal, no matter how far away from normal level the value is.

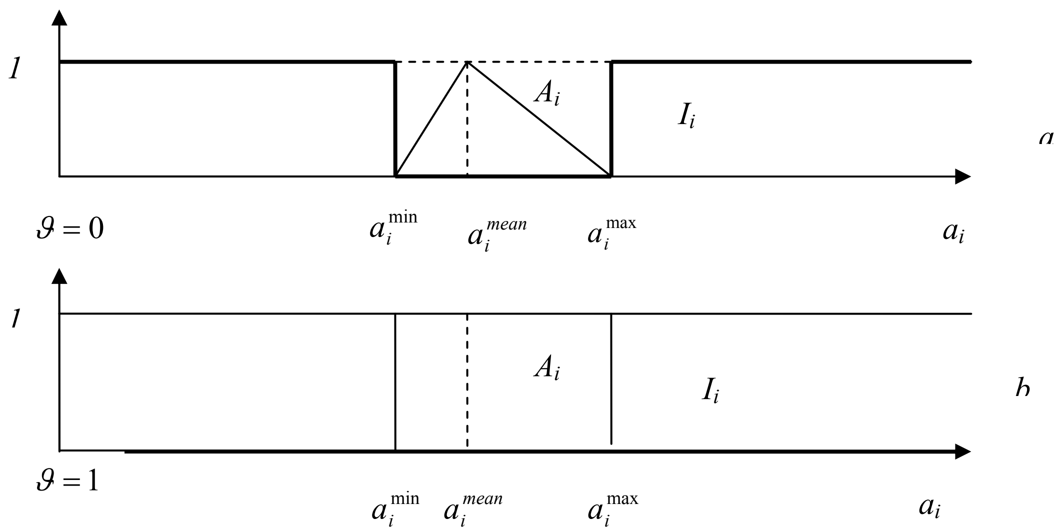


Figure 3.19: Fuzzy Set A_i and I_i, when (a) $\vartheta = 0$ and (b) $\vartheta = 1$.

Hence, the parameter ϑ specifies the sensitivity of the system in identifying abnormal values. The threshold ϑ is used to define the membership functions of the fuzzy sets A_i, and I_i, the fuzzy set I_i is then used to identify the intensity of abnormal values. It can be defined first and optimized according to the specific objective function. In the integrated intelligent model architecture, it is the free model parameter that is optimized so that the whole system can keep the adaptive ability for complex system and environment change. More approaches for defining membership functions and key parameters have been studied in [H.X. Li, 2001].

3.5.2 Identifying and Mapping Process State Change

In the content of abnormal event identification, there has been recently an effort in Knowledge Discovery and Data mining research areas. To event detection from time series data [V. Guralnik, 1999], the authors investigated the potential to identify the *time points* at which the behavior change (*change-point detection problems or event detection*) [Katharina M, 2001] [Tsien CL, 2000][Hunter J, McIntosh N, 1999] [Basseville, M.,1998]. They suggested that there was a need to determine the number of change points and then to find the functions that match the behavior between two successive change-points. More information and methods about change point determination in time series data is given in [Guralnik. V., Srivastava, J 1999].

According the approaches for determining the change point in [K. Evanglos, W.Antoni ,2000], a simple method for change point determination are defined automatically by using the minimum, maximum and statistical mean values of attributes, which are obtained during a training session. Hence, the significance of membership function in Figure 3.18 is that it not only defines fuzzy relationship between fuzzy sets and crisp value in real process, but also defines a change point, which is used to define and identify process state change.

A process state change from normal state to abnormal state does not mean a real fault occurrence. It just means there is a big probability of fault occurrence. Hence, state classifier based Fuzzy information is employed as residual generator for fault detection. If the output value [0.1] of residual generator based FNN is beyond a threshold and stay above the level, it can be announced and confirmed there a fault yields.

After the change point detection, all data (includes input data and process output data in time Window) can be roughly classified into corresponding data sets of process state. However, as we can see, this kind of change point detection for process state classification is very rough. This is why the final process behavior output is needed by compensation with fuzzy partition information with Fuzzy TS inference system in this model.

3.5.3 Training Fault Detention Unit

A process state classifier, which is based on Fuzzy Neural Networks (FNNs), can be trained with these classified data and their fuzzy partition information produced by membership functions. Fuzzy neural networks and fuzzy neurons were first introduced in the early 1970s. In the section, a classifier based on Fuzzy NNs is developed as a residual generator for process state (fault) detection.

The FNNs are neural networks are able to process fuzzy information. A straightforward way of allowing such a process model consists of fuzzy arithmetic operations to handle fuzzy quantities, as well as fuzzy sets. The classification consists of the separate treatment of the *fuzzy neural networks operating with fuzzy sets describing linguistic terms, and the fuzzy neural network operating with fuzzy sets in the form of fuzzy numbers.* Each of them is suitable for different tasks and need the use of different mathematical tools. Two different groups of FNNs are defined as follows:

- Fuzzy Arithmetic Neural Networks (FANNs) operating on fuzzy numbers using fuzzy arithmetic; and
- Fuzzy Logic Neural Networks (FLNNs) operating on fuzzy sets of linguistic using fuzzy logic

The difference between them is their neurons adopt different operator, different input vectors and different output representation.

The basic units of fuzzy neural networks-the fuzzy neurons

The basic building units of FANNs are fuzzy arithmetic neurons (FANs), they use fuzzy arithmetic operations to process an input vector of fuzzy quantities, $x \in F_n$, and yield a scalar fuzzy output $y \in F_1$. Thus, the input and output of a FAN are fuzzy numbers representing uncertain (vague) numerical values.

The generalized model of FANN can be derived from the generalized model of the crisp neuron by assuming that the input vector $Xa \in Fn$, the weigh vector $Wa \in Fn_{+1}$, and the output signal $y \in F_1$ are fuzzy quantities.

The basic building units of FLNNs are neurons with fuzzy logic. Termed fuzzy logic neurons, they use fuzzy logic operations to process the fuzzy inputs x from the unit hypercube [0, 1] n to obtain an output y ∈[0.1]1, the input and the output of such a fuzzy logic neuron express the degree of membership of certain quantities to the given polarization sets (e,g concepts, or linguistic hedges) rather than the fuzzy (uncertain value) as in the case of the FANs.

The generalized model of FLN can be derived from the generalized model of crisp neurom by assuming the input vector x∈[0,1]n , the weight vector w∈[0,1]n , and the output signal y ∈[0.1]1 are expressed in terms of their membership functions. Consequently, the operations, fuzzy logic synaptic operation, fuzzy logic aggregation and fuzzy logic output function are applied in neurons.

Based on the FLNs, a product-based fuzzy logic neurons use the confluence operation realized by T and S fuzzy logic operations. Namely, T-norms are used to realize the fuzzy logic synaptic operator, which is described the relation.

$$z_i = w_i x_i = \min(w_i, x_i)$$

And the fuzzy logic-based aggregation S operation (T-conorms) is defined as

$$u = \bigoplus_{i=1}^{n} z_1 = \max_{i=1}^{n} z_1$$

In the FLNs, the output function perform the mapping

$$y = u^r$$

The Fuzzy Neural Network

A brief produce –based FLN classifier can be expressed as follows Figure 3.20 (here, two input vectors) [N. K. Sinha, M. M. Gupta 1999].

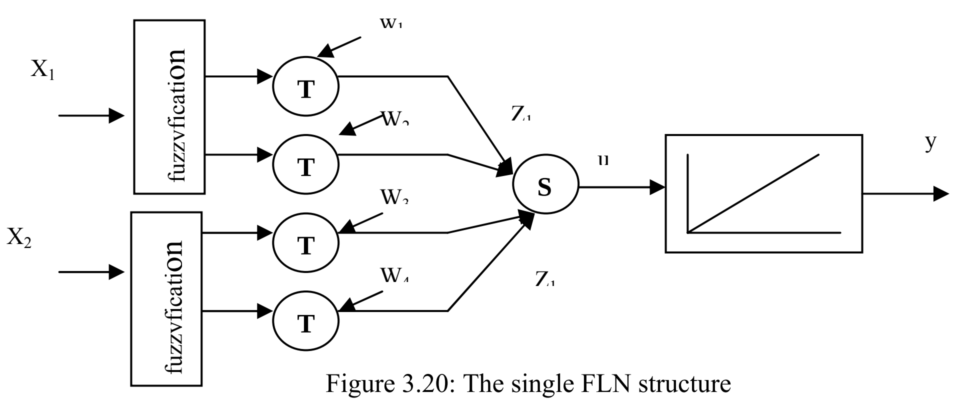


Figure 3.20: The single FLN structure

FNN can be trained using the gradient-based technique. The detail training algorithm, please refer to [N. K. Sinha, M. M. Gupta 1999] [Gupta, M. M., 1994].

Another way for fuzzy information process with neural networks was proposed by [Pan Dan, Zheng, Qilun, al, 1999], in the neural networks using BP algorithm, the input vector consists of membership values of fault symptoms while the output vector consists of membership values of fault symptoms in terms of each fault reason. It indicates this approach can still reach good fuzzy classification results.

3.5.4 Knowledge Discovery for Making Decision Rules

After different process states are identified and discriminated, the major process analysis and diagnosis are based on data in different process state and their state change. The main task is to extract the underlying causes and rules of process state change or fault occurrence. The concrete desired rules from process diagnosis depend on the concrete application, but some basic requirements can be implemented here for further analysis of process state change. These rules are listed as follows:

- *Decision tree rule* for process state classification (fault occurrence) and process state change.
- *Association rule* for frequency pattern discovery between all input (symptom) and process behaviors (fault occurrence).

As mentioned before, the classifier based on FNNs can be trained, but it is difficult to give explicit classification rule for process state change due to the black-box principles of NNs models. These rules are very important to process diagnosis and performance analysis for complex process or system. Based on certain requirements and premises, a decision tree algorithm is used here to extract the classification rule. The basic algorithm for decision tree is a *greedy algorithm* that constructs decision trees in a top-down recursive divide-conquer manner. The basic decision algorithm is presented as following:

Algorithm: Generate_decision_tree, generate a decision tree from the given training data.

Input: the training samples, *samples*, represented by discrete valued attributes; the set of candidate attributes, *attribute-list*.
Output: a decision tree.
Method:
(1) Create a node N;
(2) If *samples* are all of the same class, C then
(3) Return N as a left node labeled with the class C:
(4) If *attribute-list* is empty then
(5) Return N as a left node labeled with the most common class in samples; //majority voting
(6) Select *test-attribute*, the attribute among *attribute-list* with the highest information gain;
(7) Label node N with *test-attribute;*
(8) For each known value *ai* of *test-attribute* //partition the samples
(9) Grow a branch from node N for *samples* for which *test-attribute=ai;*

(10) Let *ai* be the set of samples in *samples* for which *test-attribute=ai;* //a partition
(11) If *ai* is empty then
(12) Attach a leaf labeled with the most common class in samples;
(13) else attach node returned by Generate_decision_tree (*ai*, *attribute-list-test-attribute*)

So far, many commercial versions, for example, C4.5, C5 algorithm for decision tree has been developed and applied in real world. In the intelligent model, the decision tree is applied in extracting the rule for classification of input symptoms and fault states.

To Association rule and algorithms, as proposed by Agrawal & Srikant 1994, *Apriori* algorithm, which is basic association rule-mining algorithm for single-dimensional, single-level, Boolean association rules, is described here for discovering frequent items sets in process state change. Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. It employs an iterative approach known as a level wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets found. This set is denoted L_1, L_1 is used to found L_2 , the set of frequent 2-itemsets, which is used find L_3 ,and so on, until no more frequent k-itemsets can be found[Han jawei, K. Micheline, 2001].

In Apriori algorithm, the Major idea is:

- A subset of a frequent itemset must be frequent
    - E.g., if **{beer, diaper, nuts}** is frequent, **{beer, diaper}** must be. Anyone is infrequent, its superset cannot be!
- A powerful, scalable candidate set pruning technique:
    - It reduces candidate k-itemsets dramatically (for k > 2)

The Apriori Algorithm

Join Step

Ck is generated by joining Lk-1with itself

Prune Step

Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset, hence should be removed.
(*Ck*: Candidate itemset of size k)
(Lk : frequent itemset of size k)

Apriori—Pseudocode

Ck: Candidate itemset of size k
Lk : frequent itemset of size k
L1 = {frequent items};

for (*k* = 1; *Lk* !=∅; *k*++) **do begin**
Ck+1 = candidates generated from *Lk*;
for each transaction *t* in database do
increment the count of all candidates in *Ck+1* that are contained in *t*

Lk+1 = candidates in *Ck+1* with min_support
end
return $\cup k$ *Lk*;

In The Apriori Algorithm, two key steps are implemented as follows:

1) How to Generate Candidates?

Suppose the items in *Lk-1* are listed in an order
Step 1: self-joining *Lk-1*
insert into ***Ck***
select ***p.item1, p.item2, …, p.itemk-1, q.itemk-1***
from ***Lk-1 p, Lk-1 q***
where ***p.item1=q.item1, …, p.itemk-2=q.itemk-2, p.itemk-1 < q.itemk-1***
Step 2: pruning
forall ***itemsets c in Ck*** do
forall ***(k-1)-subsets s of c*** do
if *(s is not in Lk-1)* **then delete** *c* **from** *Ck*

2) How to Count Supports of Candidates?

Why counting supports of candidates a problem?

- The total number of candidates can be very huge
- One transaction may contain many candidates

Method:

- Candidate itemsets are stored in a *hash-tree*
- *Leaf* node of hash-tree contains a list of itemsets and counts
- *Interior* node contains a hash table
- *Subset function*: finds all the candidates contained in a transaction

3.5.5 Process State (Fault) Identification and Diagnosis

Fault detection and diagnosis in the model is based on system input data and process historical behaviors (see Figure 3.14). Process state (Fault) diagnosis focuses on classification problems for discrete labels and prediction for continuous value in multidimensional space. There are many candidates to provide good solution for prediction and diagnosis due to the different statistic feature proprieties and sample feature data. If the data's probability distributed function is known, the Bayesian classification or regression can be applied to solve the kinds of problems. If the models do not know anything else except amounts of history data, then, some algorithm and model, for example, Support Vector Machine (SVM), Neural Networks (NNs), Fuzzy logic (FL) can be applied to solve the kinds of problems. In this case, we assume no more additional prior information is known except sample historical data, and hence the NNs are employed to make the fault diagnosis and prediction.

Fault Identification Model

Fault identification is a process that fault quantitatively diagnose. In this integrated intelligent model, fault identification, which is based on *module network architecture*, is implemented by different sub units in term of different process states. The final output is leveraged by fuzzy information of state classification. If a set of input vector is classified to belong to two different states, the fuzzy degree for these two different state is $\boldsymbol{W}$ (w_1, w_2), at the same time, the output from two sub modules is $\boldsymbol{Y}$(y_1,y_2), respectively, then, the output from whole system is calculated as: $Z = Y \times W = (y_1 \times w_1 + y_2 \times w_2)/(w_1 + w_2)$, see equation (3.14). It means the fuzzy degree value for different state is applied to adjust the final output value. The reason that involves fuzzy degree adjustment for final output is: The sub units trained by the data, which is classified using rough change point detection, can not precisely describe real process state characteristics. Hence, fuzzy information to leverage final output is necessary. The model principle is shown in Figure 3.21. Based on different process states, the prediction result of whole process can be obtained by combination of different of sub units with fuzzy weigh average.

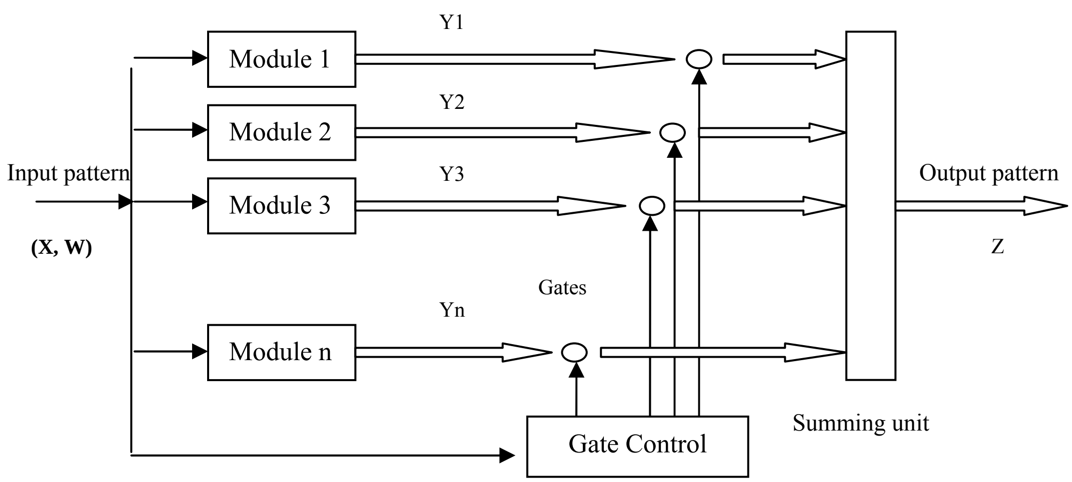


Figure 3.21: The architecture of diagnosis/ prediction model based on module

The real output value from the model can be used to evaluate the real fault diagnosis, it plays fault identification role. When the output error between the model and real object is so big that exceed a certain threshold, it still indicates there are some fault occurrence even there is no fault detection in residual generator.

Hence, in this model, there are two approaches for the process state (fault) detection, one is residual generator based the results of FNN classifier, and another is a Fuzzy TS inference model based different sub models. The difference between them is the residual generator is generated by fault *classification method* and another is generated by *approximate model way*. Their relationship in the integrated intelligent model is: *the fault fuzzy classifier provides valuable information for approximate model so as to make the model to fit the real process better.*

3.6 Chapter Summary

In the chapter, some relevant introduction of intelligence technologies, data mining and process model and system identification technologies are presented briefly first. These technologies and background knowledge contributes to fundament of the integrated intelligent model for process state (fault) detection, diagnosis and prediction in the paper.

The architecture of integrated intelligent model is developed and illustrated for application of intelligent technologies for fault detection and diagnosis/process output prediction in next section. Some key questions, such as “why is intelligence technology needed in fault problem”; “the design idea”, “what the integrated intelligent model is” and “how to implement the integrated intelligent model” are answered. The research background in the field is also given. The integrated intelligent model for modeling process is based on some premises or precondition, namely, the process can be divided into different process state and also corresponding to fault occurrence, which can be expressed as fuzzy membership functions. Based on the process state division, the model is trained and built for process state classification (fault detection), the knowledge discovery for fault diagnosis and process output prediction.

Some key techniques for integrated intelligent model have been implemented in detail in this chapter too. First, fuzzy information is obtained by defining all input-output feature variables and generating corresponding membership functions according to data distribution characteristics. These fuzzy sets and membership functions provide a description in high level for process plant. They are then regarded all input-output variables to train NNs so as to roughly describe process characteristics in different process state. The second, these fuzzy data and their process states are used to train a process state classifier base FNN (fault classification). Data mining is also employed to discover decision rule for fault detection and process state change, for example, decision tree algorithm is applied to extract classification rule for process state change with respect to symptoms condition; association rule is applied to discover frequent patterns for accompanying phenomena of fault occurrence. Finally, the same scheme as Fuzzy TS inference model is implemented for processing fault identification for process diagnosis or prediction. These diagnosis models can provide better diagnosis accuracy and capability than single NN model.

In essence, the integrated intelligent model is gotten from Fuzzy TS model of dynamic system, which is used to describe dynamic model with fuzzy TS inference system and NARX dynamic system structure. The integrated intelligent model for process state (fault) detection and process output prediction is explained and realized by Fuzzy NARX TS dynamic model from mathematic view. In the model, the Antecedent part, which is generated by two fuzzy classifiers**,** constructs the residual generator for fault detection, The map relationship between input-output of fuzzy state classifiers describes the relationship among system input (fault symptom) and process state (fault occurrence); and then, the consequence part, which is generated by system identification based module Neural Network structure and fuzzy partition information, constructed the fault identification and process output prediction.

In the model, the final fault residual signal is generated by output of two Fuzzy classifiers: one Fuzzy classifier is used for fault detection measured from real system inputs and another fuzzy classifier from historical behaviors. The outputs of two fuzzy classifiers are combined by fuzzy operation to produce final fault residual signal. If the residual signal is beyond a defined threshold and stay on the level, then a fault event can be detected by this system. The final process output is determined by the combination of two aspects, one is from process behavior output resulted from real system input, and another is from process behavior output resulted from historical behaviors of real process. These two kinds of process outputs are integrated by Fuzzy TS model so as to produce a final proper output. The process output of mathematic expression is determined by system identification with NNs models.

CHAPTER 4 OPTIMAL MODEL AND ADAPTIBILITY

4.1 Introduction

In the past years, many effective optimization methods were employed in engineering and science fields. Optimization theory and techniques are used to find the values of a set of parameters that minimize or maximize some error of cost function of interest. The error or cost function stands for a measure of how good the solution is for a given problem.

There are many ways to be applied for the problem of finding an optimal (best) or suboptimal (close to best) set of parameters. From the mathematic point of view, the problems of finding the minimum or maximum for linear optimization is relatively easy to be analyzed, but nonlinear optimization methods are still under investigation [V. Kecman, 2001][Koch, W. H,1993, 1994]. In real applications, the error function E is normally a hypersurface representing the mapping of a high-dimensional weights matrix into the measure of goodness. The E is typically a nonlinear hypersurface that cannot be visualized if its dimension is more than three.

Generally speaking, using Local Quadratic (LQ) approximation to a nonlinear error function is a good solution due to two reasons as follows: the first, quadratic approximation results in relatively simple theorems concerning the general properties of various optimization methods. Second, in the neighborhood of local minima (maxima), quadratic approximations behave like the original nonlinear functions.

Optimization is very important for modeling real plants. It searches suitable model parameters so as to make the model track the real plants or processes as much as possible. Optimization is also important for learning system and intelligence model, In NN, SVM and FLMs model, the problem of learning some underlying function, for example, mapping, dependency, degree of relatedness between input and output variables, is realized to the nonlinear search (estimation or identification) of the weights of NNs, SVM and FLMs [V, Kecman. 2001]. In this chapter, two different optimization schemes and different optimization methods are developed for optimal model and adaptive abilities.

4.2 Optimization Goals and Schemes

To complex plants, generally speaking, it can be abstracted on certain degree first in order to build a model to describe it. Some unimportant factors for modeling real process for specific problems are often ignored. It is a fact that the model always has some error and difference with the real plant whether you employed methods. Some changes and factors make the model deviate the real process as follows:

- Different condition and external environment occurs
- Inner parameters change of real plant
- Noise input and disturbance and so on

When models are used to represent real processes, the main model error is listed as follows:

- The error to extract the model from the real processes
- The method error to model the real plant
- Noise signal or measurement error under building the model
- The error due to the model simplification and calculation

This all makes it necessary that process model must be optimized or have adaptive ability adjusting itself so as to match real plants as much as possible. Hence, model optimization is regarded as a key way to search and adaptively adjust the model's key parameters.

In the integrated intelligent model developed in last chapter, it has been developed for process state identification, fault detection and process output prediction for complex plants so as to reveal the system's behavior characteristics [Tang, M. Koch, W. H., 2004]. The comparison result (error function) between process output and model output within certain time interval is an important criterion to evaluate the model's precisions. Therefore, the error function between real plant output and model output can be regarded as the main measure criterion for model optimization. It is shown in Figure 4.1.

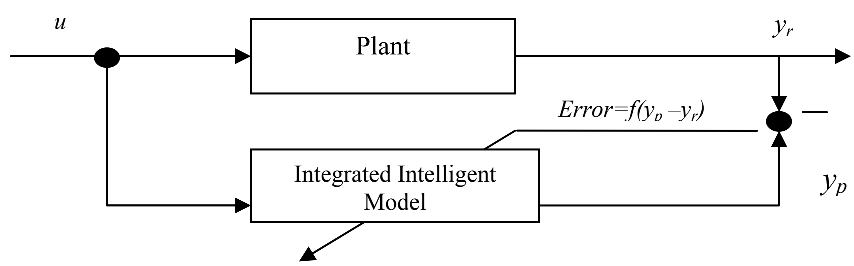


Figure 4.1: The model optimization architecture

Here, y_p is the output of integrated intelligent model and y_r is the output of real plant. Variable u is the input of the real plant. The function $f(y_p\text{-}y_r)$ is the error function, In the figure, the integrated intelligent model can posses optimal or adaptive ability to trace real process based on certain error criterion if the error value can be controlled under certain range by turning some key model parameters. The smaller the error $f(y_p\text{-}y_r)$ is, the more precise the model is reflecting the real plant. The error is regarded as a key criterion to adjust the inner parameters of the integrated intelligent model by some optimization approaches.

Based on the principle in Figure 4.1, the objective function is defined as the error function between the real plant output $f(x_i)$ and model output $f_a(x_i)$ within certain time interval. From the vector space point of view, the error can be regarded as the approximation comparison of two different vector functions in vector space. Their error can be measured with Euclidean norm due to two reasons: The first, the assumption about the Gaussian character of the noise in a control systems environment is an acceptable and reasonable one. The second, the L_2 is mathematically simple and tractable. It is given in (4.1) below.

$$L_2 : \|f - f_a\|_2 = (\sum_i^P |f(x_i) - f_a(x_i)|^2)^{\frac{1}{2}} \quad \text{(Euclidean norm)} \qquad (4.1)$$

Here, f, f_a are different vectors in the same vector space. The L_2 norm (Euclidean norm) is almost universally used in many fields, particularly in signal processing, system identification and control. Hence, the objective function for optimizing intelligent model is designed based on L_2 norm as follows:

$$F(x) = \sqrt{\sum_{i=1}^{T} (f(x_i) - f_a(x_i))^2} \qquad (4.2)$$

In equation (4.2), $f(x_i)$ represents the function of real plant, $f_a(x_i)$ represents the approximation function (model function) expressed by integrated intelligent model, T is a certain time interval.

The steps involved in formulating an optimization scheme for integrated intelligent model include:

1. Determining the variables for the optimization
2. Formulating the objective function(s)
3. Defining all constraints necessary to consider under model optimization

In order to formulate the optimization scheme for integrated intelligent model, the objective functions, optimization variables and constraints must be determined first. After formulation, suitable optimization algorithms and methods have to be chosen to implement optimization goals.

From the architecture of the integrated intelligent model in Figure 3.13, the model has two functions:

- Process state identification (as well as fault detection) for process diagnosis and
- Process output prediction

Some key parameters in integrated intelligent model are going to be optimized based on the model output error function between real plant output and model prediction output for above two purposes. In the following section, the two different model optimization schemes and implementation methods are given in detailed.

4.2.1 Optimization Schemes for Process Diagnosis

Under the circumstance, the model is mainly used for process state (or fault) detection, diagnosis and process prediction based on different process state. The main goal of the model is correctly to identify process state changes and also classify different process state so as to diagnose process states.

Variables for model optimization

Although the function $f_a(x_i)$ can be described with mathematical expression in objective function *F(x)* in (4.2), it is too complex to be applied well in (4.2). From the architecture of the integrated intelligent model, the model is characterized as following some parameters (here, to model description, they are called parameters, but to objective function *F(x)*, they are independent variables):

1. The relevant parameters for defining all membership functions, for example, if these membership functions are defined as triangle shape based on statistic parameters a^{min}, a^{mean} and a^{max} (see chapter 3.), then, a^{min}, a^{mean} and a^{max}, ϑ of the whole input-output variables are one kind of the model parameters.
2. The relevant weights ω_{FNN} for process state classifiers based on FNNs. These weights ω_{FNN} are used to characterize the model.
3. The relevant weights ω_{NN} and nodes of sub NNs models.

From the above parameters for characterizing the whole integrated intelligent model, the optimized variables of objective functions can be determined due to following reasons:

1. The model parameter ϑ of output variable is designed as a key quantity for process state identification or fault detection, as well as model output prediction. It plays a threshold role of the residual generator. When residual signal is greater than the threshold value and stays on this level, it indicates that the process is in abnormal state or a fault has occurred. Hence, the model parameter ϑ is an important indicator to detect the process state changes or/and fault occurrences. It is also used to define the sensitivity of process state classification (see Figure 3.18 and Figure 3.19). At the same time, the value of ϑ is also used to determine the output of the fuzzy TS dynamic model. When the process or system is in normal state, namely, the output of fuzzy NN classifiers is less than ϑ**,** the model output of the fuzzy TS dynamic model is only calculated by different NN models output associating with normal state. If the output of the fuzzy NN classifier is greater than ϑ**,** it means system could be in fault state or abnormal state, the system output of the fuzzy TS dynamic model is determined by NN models output corresponding to both normal and abnormal state. The final output is determined by the combination of both outputs of these NN modes with Fuzzy degree compensation (See Figure 3.16 and equation 3.28).
2. The model parameters $\boldsymbol{a}^{min}$, $\boldsymbol{a}^{mean}$ and $\boldsymbol{a}^{max}$ are defined based on historical data for the membership function of all input-output variables. They can be automatically and online generated with historical data. It means the membership functions for all variables are determined by the historical data, which can represent real process characteristics. If necessary, they can be renewed and updated by statistic methods. Hence, it is not necessary to involve them as optimization variables for whole model optimization.
3. The model parameters ω_{FNN} and ω_{NN} are obtained by training with historical data. They are used to represent the characteristics of sub NN models and also be optimized by training algorithms. Hence, it is not necessary to regard them as optimized variables for model optimization.

Therefore, there is only one model parameter ϑ, which is the threshold of process states (or fault state) identification, is considered as optimization variable of objective function in (4.2) for process state identification due to the fact that it is used to:

- Classify different process states in the model.
- Measure the sensitivity for process state detection.
- Calculate the prediction output of the model

Then, the objective function can then be expressed as:

$$F(x,\vartheta) = \sqrt{\sum_{i=1}^{T}(f(x_i) - f_a(x_i,\vartheta))^2} \qquad (4.3)$$

Constraints

Based on integrated intelligent model and the objective function, the constraints can be defined as follows:

Model parameter ϑ is defined for *sensitivity* of process state classification and *threshold* for residual generator. It is the threshold value to represent fuzzy degree belonging to different process states. When ϑ=1, it means the model deals with all process states as abnormal process state (or fault state), no normal state can be defined and identified. when ϑ=0, it means the model treats with all process states as normal process state, no abnormal state (fault state) are defined and identified too. Hence, the constraint is defined as:

$$0 \le \vartheta \le 1 \qquad (4.4)$$

Hence, the objective function here of model optimization is defined as in equation (4.5).

$$F(x,\vartheta) = \sqrt{\sum_{i=1}^{T}(f(x_i) - f_a(x_i,\vartheta))^2} \quad 0 \le \vartheta \le 1 \qquad (4.5)$$

Here, $f(x_i)$ represents the function of real plant, $f_a(x_i,\vartheta)$ represents the approximation function (model function) of integrated intelligent model, T is certain time interval. The parameter ϑ Є [0, 1] is regarded as a threshold of residual signal for measurement of process state change (fault) as Figure 4.2.

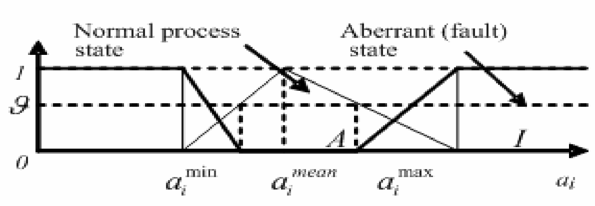


Figure 4.2: The membership functions of process state and parameter ϑ

When fuzzy degree of normal process state $w_{normal} > \vartheta$, it indicates the process completely is in normal state, hence, the model output is just determined by normal state output. When fuzzy degree of abnormal (or fault) state $w_{abnormal} > \vartheta$, the process is regarded in abnormal (or fault) state and model output is determined by output of both normal state and abnormal state. It can be calculated in the fuzzy TS NARX model, as equation (3.28) and (3.29). Hence, different model output and process state (fault) detection can be obtained due to different ϑ value. The value of ϑ directly affects the result of process state (fault) detection as well as the result of prediction output.

To the fuzzy model, the prediction value can be expressed in equation (4.6) (here, only three process states are given, for example, normal state (state 2) and abnormal state (state 1 and state 3)).

$$\begin{aligned} X(k+1) = {} & a\left(\frac{X_{i1}(k)*w_{i1}(k)+X_{i2}(k)*w_{i2}(k)+X_{i3}(k)*w_{i3}(k)}{(w_{i1}(k)+w_{i2}(k)+w_{i3}(k))}\right) \\ & +b\left(\frac{X_{u1}(k)*w_{u1}(k)+X_{u2}(k)*w_{u2}(k)+X_{u3}(k)*w_{u3}(k)}{(w_{u1}(k)+w_{u2}(k)+w_{u3}(k))}\right) \end{aligned} \quad (4.6)$$

where, $X(k+1)$ is prediction value of model in time (k+1). $X_{i1}(k)$, $X_{i2}(k)$ and $X_{i3}(k)$ are output value in abnormal state (process state 1), normal state (process state 2) and abnormal state (process state 3) in time (k) produced by historical behavior output, respectively. $X_{u1}(k)$, $X_{u2}(k)$ and $X_{u3}(k)$ are output value in abnormal state (process state 1), normal state (process state 2) and abnormal state (process state 3) in time (k) produced by real system input, respectively.

a is the weight value for historical behavior output and *b* is for real system input in the equation (4.6).

$w_{i1}(k)$, $w_{i2}(k)$ and $w_{i3}(k)$ are fuzzy degree (weights) of the real process state belonging to abnormal state (process state 1), normal state (process state 2) and abnormal state (process state 3) produced by system input, respectively. $w_{u1}(k)$, $w_{u2}(k)$ and $w_{u3}(k)$ are fuzzy degree (weights) of the real process state belonging to abnormal state (process state 1), normal state (process state 2) and abnormal state (process state 3) produced by historical behavior output, respectively.

Because of the characteristics of $w_{i1}(k)$, $w_{i2}(k)$, $w_{13}(k)$ and $w_{u1}(k), w_{u2}(k), w_{u3}(k)$ in the model, there are equations:

$$w_{i1}(k)+w_{i2}(k)+w_{i3}(k)=1 \quad (4.7)$$

$$w_{u1}(k)+w_{u2}(k)+w_{u3}(k)=1 \quad (4.8)$$

Hence, the equation (4.6) can be rewritten in form of matrix and vector as follows:

$$X(k+1)=a\begin{bmatrix} w_{i1}(k) & 0 & 0 \\ 0 & w_{i2}(k) & 0 \\ 0 & 0 & w_{i3}(k) \end{bmatrix}\begin{bmatrix} X_{i1}(k) \\ X_{i2}(k) \\ X_{13}(k) \end{bmatrix}+b\begin{bmatrix} w_{u1}(k) & 0 & 0 \\ 0 & w_{u2}(k) & 0 \\ 0 & 0 & w_{u3}(k) \end{bmatrix}\begin{bmatrix} X_{u1}(k) \\ X_{u2}(k) \\ X_{u3}(k) \end{bmatrix} \quad (4.9)$$

According to the scheme of model prediction value based on threshold parameter ϑ adjustment, the model prediction output with ϑ adjustment is given as:

$$Y(k+1)=\begin{cases} aX_{i2}(k)+bX_{u2}(k) & \text{if } w_2>\vartheta_2 \\ a(X_{i1}(k)*w_{i1}+X_{i2}(k)*w_{i2})+b(X_{u1}(k)*w_{u1}+X_{u2}(k)*w_{u2}) & \text{if } w_1>\vartheta_1 \\ a(X_{i3}(k)*w_{i3}+X_{i2}(k)*w_{i2})+b(X_{u3}(k)*w_{u3}+X_{u2}(k)*w_{u2}) & \text{if } w_3>\vartheta_3 \end{cases} \quad (4.10)$$

where, $Y(k+1)$ is optimal value of prediction output produced by model optimization algorithms. ϑ_2 is the threshold of real process state belonging to the normal process state (process state 2); ϑ_1 is the threshold for abnormal process state 1 (process state 1) and ϑ_3 is the threshold for abnormal state 3 (process state *3*), The equation (4.10) can be explained as:

- When real process works in normal state, it means the fuzzy weight (w_2) belonging normal state is greater than the threshold ϑ_2, then the prediction value from this model is calculated only by model output of normal state.
- When real process works on abnormal state, it means the fuzzy weight w_i, (i=1…n, i≠2) belonging abnormal state is greater than a threshold ϑ_i, (i=1…n, i≠2), the prediction value from this model is calculated by weight average among output value in normal states and abnormal states.

According to equation (4.6), the output of fuzzy TS NARX model can be expressed as:

$$Y(k+1)=H(k)^T X(k+1)=\begin{bmatrix}1 & 1 & 1\end{bmatrix}\begin{bmatrix} aw_{i,1}(k)X_{i,1}(k)+bw_{u,1}(k)X_{u,1}(k) \\ aw_{i,2}(k)X_{i,2}(k)+bw_{u,2}(k)X_{u,2}(k) \\ aw_{i,3}(k)X_{i,3}(k)+bw_{u,3}(k)X_{u,3}(k) \end{bmatrix} \quad (4.11)$$

Here, *H(k)* is 3X1 matrix, which is used to represent process state identification.

When the process is in normal state, the model prediction output is given in (4.12), namely, only normal state output is considered under the circumstance. The prediction output is given as:

$$Y(k+1)=H(k)^T X(k+1)=\begin{bmatrix}0 & 1 & 0\end{bmatrix}\begin{bmatrix} aw_{i,1}(k)X_{i,1}(k)+bw_{u,1}(k)X_{u,1}(k) \\ aw_{i,2}(k)X_{i,2}(k)+bw_{u,2}(k)X_{u,2}(k) \\ aw_{i,3}(k)X_{i,3}(k)+bw_{u,3}(k)X_{u,3}(k) \end{bmatrix} \quad (4.12)$$

When the process is in abnormal state 1 or 2, the corresponding $H(k)$ can be gotten. Thus, the vector $H(k)$, which is determined by the result of optimized threshold value ϑ_i, can be expressed as:

$$H(k,\vartheta)^T=\begin{cases} \begin{bmatrix}1 & 1 & 0\end{bmatrix} & \text{if } w_1>\vartheta_1 \\ \begin{bmatrix}0 & 1 & 0\end{bmatrix} & \text{if } w_2>\vartheta_2 \\ \begin{bmatrix}0 & 1 & 1\end{bmatrix} & \text{if } w_3>\vartheta_3 \end{cases} \quad (4.13)$$

Thus, from the equation (4.13), the final model prediction output with ϑ adjustment can be written in vector form as:

$$Y(k+1) = H(k,\vartheta)^T * X(k+1) \qquad (4.14)$$

4.2.2 Optimization Scheme for Process Prediction

Under the circumstance, the model optimal prediction output is not determined by the threshold variable ϑ in equation (4.10). The optimization goal is to search the optimal model prediction.

According to the model architecture in Figure 3.13 and the principle of work module in Figure 3.14, the final desired prediction output of the model is not the output from fuzzy TS NARX model. It is generated by an optimal fuzzy TS NARX model based on the optimization scheme. The final desired prediction value is determined by two factors: one is output *X(k+1)* from fuzzy TS NARX model and another is optimization unit $H(k+1)$, it can be illustrated in Figure 4.3.

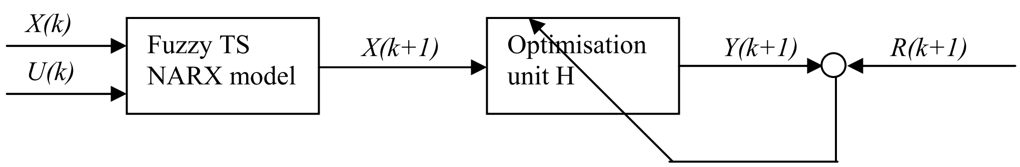


Figure 4.3 The optimization unit in integrated intelligent model

Here, *X(K)* is 3×1 matrix corresponding to different state output at k moment produced by historical behaviors and *U(K)* is 3×1 matrix corresponding to different state output at k moment produced by real system input. Namely,

$$X(k) = \begin{bmatrix} X_{i1}(k) \\ X_{i2}(k) \\ X_{i3}(k) \end{bmatrix} \text{ and } U(k) = \begin{bmatrix} X_{u1}(k) \\ X_{u2}(k) \\ X_{u3}(k) \end{bmatrix}$$

$X(k+1)$ is a 3×1 matrix, which is the comprehensive output corresponding to different process state in fuzzy TS NARX inference model. It can be expressed as:

$$X(k+1) = \begin{bmatrix} X_1(k+1) \\ X_2(k+1) \\ X_3(k+1) \end{bmatrix} = aX(k) + bU(k) = \begin{bmatrix} aw_{i,1}(k)X_{i,1}(k) + bw_{u,1}(k)X_{u,1}(k) \\ aw_{i,2}(k)X_{i,2}(k) + bw_{u,2}(k)X_{u,2}(k) \\ aw_{i,3}(k)X_{i,3}(k) + bw_{u,3}(k)X_{u,3}(k) \end{bmatrix}$$

$Y(k+1)$ is the final desired prediction output from the model and $R(k+1)$ is real process output.

Hence, the final prediction value from the system model can be calculated by follows:

$$Y(k+1) = H(k)^T X(k+1) \qquad (4.15)$$

In this sense, the model optimization aim is to search the optimal $H(k)$ so as to minimize the objective function (4.16).

$$F(x,H)=\sqrt{\sum_{i=1}^{T}(f(x_i)-f_a(x_i,H))^2} \tag{4.16}$$

The difference between two model optimization schemes is that the *H(k)* is optimized with different methods and rules. In the first case, *H(k)* is determined by optimal threshold ϑ^*, then, only three forms of *H(k)* is gotten in (4.13). In the second case, *H(k)* is determined by LS approach (see 4.4.1) without any consideration of optimal threshold ϑ^*. It can reach better prediction output than the first case does, but it can not provide any information about process state changes.

4.3 Model Optimization Methods for Process Diagnosis

4.3.1 Deterministic Approaches

Under the circumstance, the optimization aim here is to search optimal ϑ^* so as to minimize the objective function (4.5). A nonlinear optimization algorithm, which is called Modified Levitin/Polyak algorithm [E. S Levitin,, B. T.Polyak, 1966] derived from Newton-Type method, is used for the optimization problems. After the optimal solution ϑ^* is obtained, optimal prediction value of model can be calculated from equation (4.10 or 4.13). It is demonstrated in the application case for products supply forecasting in the paper.

Based on the characteristics of objective function and constrains in the model optimization, the optimization problems can be classed as linearly constrained nonlinear minimization problems. In general, it can be expressed as [Koch, 1983]:

for a general nonlinear objective function f: $R^n -> R, (i.e.,)$ $f(x) \neq C^T x + C_0$ and $f(x) \neq \frac{1}{2} x^T H x + h^T x + h_0$ (H positive definite), the problem is given

$$f(x) -> Min!$$

$$\text{s.t} \quad x \in D := \{x \in R^n : -b + Bx \leq 0\} \subseteq R^n, \tag{4.17}$$

If $x^* \in D$ is called a local minimum point of problem (4.17), then the necessary optimality conditions can be given as:

$$\nabla f(x^*)^T (y - x^*) \geq 0, \forall y \in D \tag{4.18}$$

is satisfied.

For the optimality condition (4.18), a modified Levitin/Polyak algorithm has been selected here because of successive quadratic approximation [Koch. W, 1983].

Modified Levitin/Polyak algorithm is a Newton-Type method, a convex quadratic approximation of f(x) in below form is adopted successfully.

$$f(y) \approx q_k(y) = df(x^k, s_k)^T (y - x^k) + 0.5(y - x^k)^T M(x^k, s_k)(y - x^k) \quad (4.19)$$

The gradient, the first order derivative of f, $\nabla f(x^k)$, it is being approximately calculated by:

$$\left[df(x^k, s_k)\right]_{j=1,\dots n} = \left[f(x^k + s_k e^j) - f(x^n - s_k e^j)\right] / 2s_k$$

The Hessian matrix $\nabla^2 f(x^k)$ of the second order derivatives of f is approximately calculated by:

$$M(x^k, s_k) = \begin{cases} H(d^2 f(x^k, s_k))\text{...discreate approximation of the HESSE matrix} \\ diag(d^2 f(x^k, s_k) > 0 : j = 1, \dots n)\text{... diagonalised HESSE matrix} \\ I \qquad \text{unit matrix} \end{cases} \quad (4.20)$$

Based on the optimization and approximate principles, the algorithm is given as below.

Algorithm

Step 1: Choose $x^0 \in R^n, C \in (0, \tfrac{1}{2}), \varepsilon > 0$, and a sequence $\{s_k\}$ of discretisation parameters with $s_k > 0$, $\lim_{k \to \infty} s_k = 0$

Step 2: Compute a solution points y^k of the quadratic programming problems

$$q_k(y) \to \min!$$

s.t.

$$y \in D$$

Set $p^k := y^k - x^k$

Step 3: If $\left\|df(x^k, s_k)\right\| \le \varepsilon$ or $\left\|p^k\right\| \le \varepsilon$ or $\left|df(x^k, s_k)^T p^k\right| \le \varepsilon$ stop.

Step 4: Compute a step length $a_k \in (0,1]$ with the ARMIJO algorithm and

$$f(x^k + a_k p^k) \le f(x^k) + a_k C df(x^k, s_k)^T P^k$$

Step 5: Set $x^{k+1} := x^k + a_k p^k$, $k := k + 1$, go to step 2.

The algorithm is practicable and implementable. The sketch in Figure 4.4 is illustrated the successive generation and solution of quadratic minimization problems combined with ARMIJO's step length determination [L. ARMIJO, 66].

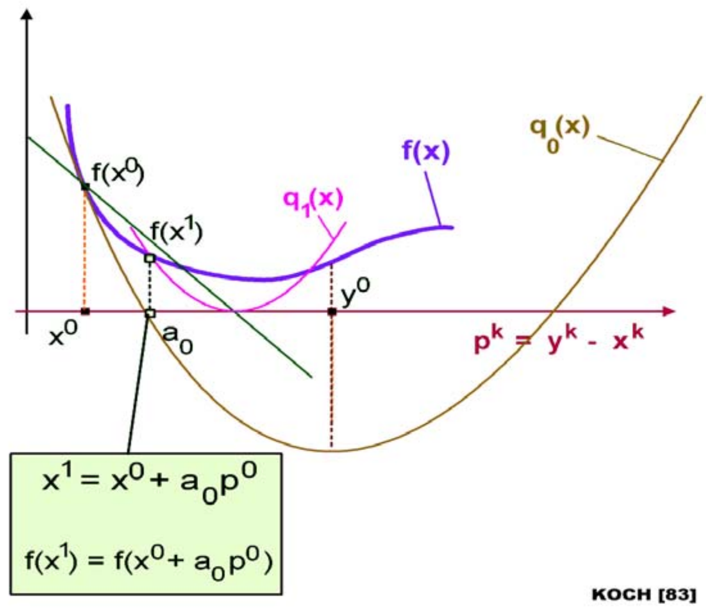


Figure 4.4: The optimization principles and search process

Under the usual assumptions for this kind of algorithms, it can be proved both global convergence and locally quadratic convergence [Koch. W, 1983]. In chapter 6, the deterministic optimization algorithm in *NOSYS® (Nonlinear Optimization System Software)* tools is applied for calculating the adaptive threshold for process state identification and process output prediction.

4.3.2 Genetic Algorithm

"Genetic algorithms are based on a biological metaphor: They view learning as a competition among a population of evolving candidate problem solutions. A 'fitness' function evaluates each solution to decide whether it will contribute to the next generation of solutions. Then, through operations analogous to gene transfer in sexual reproduction, the algorithm creates a new population of candidate solutions."

-Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand- by John H. Holland

- From *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*, Fourth Edition, Luger, George F. 2002. Harlow, England: Addison-Wesley.

The principle of GA passes through the loop of three operations and shown in Figure 4.5 [J. H. Holland, 1962, 1973].

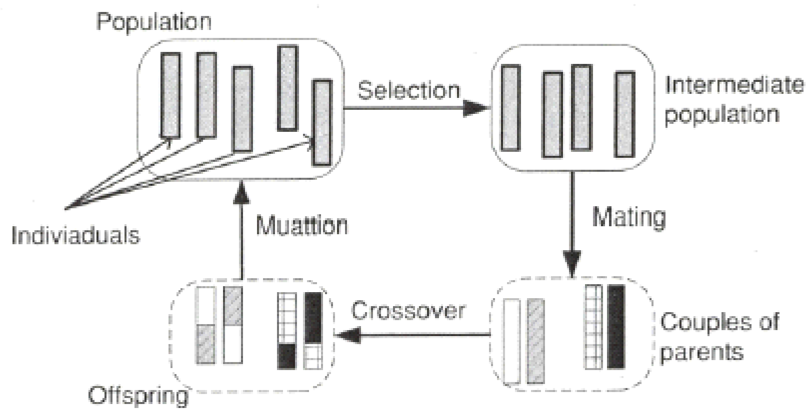


Figure 4.5: The basic procedure with GA optimization

1. Selection of the best gene strings (by, for example, using a so called roulette wheel, selection operation)
2. Genetic operation (crossover or resemblance, mutation)
3. Replacement of bad gene strings of the old population with new gene strings (children)

Before the optimization loop begins, the variables to be optimized have to be transformed in a suitable form by encoding. The encoding transforms problem-specific information into binary numbers, which is so called gene code in GA, and then realizes optimization operation based on them.

The GA works with aggregation of gene strings, called population, initially, a population is generated randomly, and then a selection operation is applied so as to get a good population.

Selection is used to evaluate all gene strings by calculating error functions for each set of parameters (gene strings). Some of the gene strings with highest fitness values are selected from the population to generate the children. A stochastic method, called roulette wheel method, is applied for selection. It provides a mechanism to represent each string in the population on the wheel in proportion to its fitness value.

Crossover combines subparts of two parents gene strings, which are chosen by the selection, to produce children with some parts of both parent's genetic material. It can be shown in Figure 4.6.

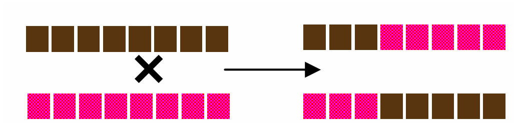


Figure 4.6: The crossover operation

Mutation is an operator that introduces variation into the gene string, the operation occurs occasionally, usually with a small probability P_m, each bit in gene string will be tested and if necessary, inverted. Mutation is applied to each child after crossover. It randomly alters each gene with a small probability, typically 0.001.

Mutation provides a small amount of random search, and helps to ensure that no point in the search space has a zero probability of being examined.

After the selection and the genetic operations by which the new children are produced, they will replace a bad part of the parent's generation and become a component of the following generation. Each sequence produces the new set of gene strings, and one must check whether the new gene strings is better than last generation based on fitness function value. Hence, the best gene string will be keep until reach the optimal solution.

GA has following advantages

- They work with binary code version of variables. It allows simultaneous optimization of all.
- Their search is from a population of points simultaneously, not for one signal point, it can help to avoid local hills.
- They use only objective function information, no derivatives are needed.
- They use stochastic reproduction instead of deterministic rules.

The drawback of *GA* is that the calculation of *GA* takes much time, especially when the optimization search is around the solution points.

GeneHunter® [GeneHunter® 2000] is a tool that solve problems based on the genetic theory of evolution, it mainly works in Microsoft Excel environment as an add-in element, after formulating the problem with the fitness functions, the chromosomes and constraints appropriately defined in the excel spreadsheet, *GeneHunter*® is then activated to work on the evolution process. The application of GA for optimizing parameter ϑ for optimal model is done in chapter 6.

4.4 Model Optimization Methods for Process Prediction

4.4.1 Method of Least Squares (LS)

The method of least squares is essential in system and control engineering. It provides simple tools for estimating or optimizing the parameters of a linear system. In the method in the model optimization, the key model parameter $H(k)$ is estimated by the least squares approach. Consider the model expression (4.21)

$$x(k+1) = H^T x(k) \tag{4.21}$$

It can be expressed as

$$y(k) = H^T x(k) + e(k)$$

where $H(k)$ is a column vector of parameters to be estimated from observations *y(k), x(k)*, *k=1,2,…k,* and where *x(k)* is independent of $H(k)$, *k* is number of observations.

The goal is to obtain an estimate of the parameters of the model. The commonly accepted method for parameter estimate is LS. The estimation criterion is given as follows:

$$J(H) = \frac{1}{K}\sum_{k=1}^{K} a_k \left[y(k) - H^T x(k)\right]^2 \qquad (4.22)$$

This quadratic function to be minimized with respect to H expresses the average of the *weighted* squared errors between the K observed outputs, *y(k)*, and the predictions *x(k)* provided by the model. The scalar coefficients a_k allow the weighting of different observations.

The important benefit of having a quadratic function is that it can be minimized analytically. Because a quadratic function has the shape of parabola, and thus possesses a single optimal solution (optimum point). The optimum (minimum or maximum) can be solved analytically by setting the derivative to zero and the examination of the second derivative shows whether a minimum or maximum is in question.

According to the derivation of LS approaches, the parameter vector minimizing the cost function J is given by (if the inverse of the matrix exists)[E. Ikonen, K. Najim, 2002,][S. Haykin, 1996]:

$$\hat{H} = \left[\sum_{k=1}^{K} a_k x(k) x^T(k)\right]^{-1} \sum_{k=1}^{K} a_k x(k) y(k) \qquad (4.23)$$

For the optimum to be a minimum, the matrix of second derivative is positive definite (Matirx A is positive definite if $x^T Ax \geq 0\, for$ x≠0)

The optimum is a minimum if the second derivative is positive, *i.e*. the matrix of second derivative is positive definite. The algorithm is given below:

Algorithm

Let a system be given by

$$y(k) = H^T x(k) + \xi(k) \qquad (4.24)$$

Where *y(k)* is the scalar output of the system, H is the true parameters vector of the system of size Ix1; x(k) is the prediction vector of size Ix1, and $\xi(k)$ is system noise. The least squares parameter estimate $\hat{H}$ of H is:

$$\hat{H} = \left[\sum_{k=1}^{K} a_k x(k) x^T(k)\right]^{-1} \sum_{k=1}^{K} a_k x(k) y(k) \qquad (4.25)$$

If $\sum_{k=1}^{K} a_k x(k) x^T(k)$ is invertible, then there is a unique solution. The inverse exists if the matrix is positive definite.

The calculation of the least squares estimation can be convert compact matrix form due to more convenient calculation.

Collect the observations at the input of the model to a $K \times I$ matrix

$$\Phi = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \vdots \\ x^T(K) \end{bmatrix} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_I(1) \\ x_1(2) & x_2(2) & \cdots & x_I(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(K) & x_2(K) & \cdots & x_I(K) \end{bmatrix} \tag{4.26}$$

And observation at the output to a $K \times 1$ vector

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(K) \end{bmatrix}$$

The K equations can be represented by a matrix equation

$$Y = \Phi H + E \tag{4.27}$$

Where ***E*** is a Kx1 column vector of modeling errors, now the least squares algorithm that minimizes

$$J = \frac{1}{K}(y - \Phi H)^T (y - \Phi H) \tag{4.28}$$

Can be represented in a compact form by

$$\hat{H} = \arg\min_{\vartheta} J = \left[\Phi^T \Phi\right]^{-1} \Phi^T Y \tag{4.29}$$

Where

$\frac{\partial^2 J}{\partial \vartheta^2} = \Phi^T \Phi$ must be positive definite

4.4.2 Recursive Least Squares (RLS) Method

The least squares method provides an estimate for the model parameters based on a set of observations. Because the observation pairs are obtained one by one from the process, and the model parameters would be updated whenever new information becomes available, this can be done by adding the new observation to the previous set of observations and recomputing. It is necessary to derive a recursive formulation instead of recomputing the estimates with all available data. The new model parameters are updated with the new data samples, namely, the recursive algorithms is written as follows [E. Ikonen, K. Najim, 2002][S. Haykin, 1996]:

$$H(k+1) = H(k) + a_k (Y(k+1) - X(k+1)) \tag{4.30}$$

Where,

$H(k+1)$: New estimate; $H(k)$: old estimate; a_k : Correction factor; $Y(k+1)$: New observation; $X(k+1)$: Prediction with old estimate.

Derivation

The least squares estimate at sample instant k-1 is given by

$$\hat{H}(k-1) = \left[\sum_{i=1}^{k-1} a_i x(i) x^T(i)\right]^{-1} \sum_{i=1}^{k-1} a_i x(i) y(i) \tag{4.31}$$

At sample k, new information is obtained and the least squares estimate is given by

$$\hat{H}(k) = \left[\sum_{i=1}^{k-1} a_i x(i) x^T(i) + a_k x(k) x^T(k)\right]^{-1} \times \left(\sum_{i=1}^{k-1} a_i x(i) y(i) + a_k x(k) y(k)\right)$$

Define

$$R(k) = \sum_{i=1}^{k} a_i x(k) x^T(i)$$

Then, the recursive formula for *R(k)* can be led as following

$$R(k) = R(k-1) + a_k x(k) x^T(k) \tag{4.32}$$

The least squares estimate can be rewritten as

$$\hat{H}(k) = R^{-1}(k)\left[\sum_{i=1}^{k-1} a_i x(i) x^T(i) + a_k x(k) x^T(k)\right]$$

The estimate at iteration k-1 can be rewritten as follows

$$\hat{H}(k-1) = R^{-1}(k-1) \sum_{i=1}^{k-1} a_i x(i) x^T(i) \tag{4.33}$$

Which gives

$$\sum_{i=1}^{k-1} a_i x(i) x^T(i) = R(k-1)\hat{H}(k-1)$$

Substituting this equation into (4.33), it can be gotten following recursive formation

$$\hat{H}(k) = R^{-1}(k)\left[R(k-1)\hat{H}(k-1) + a_k x(k) y(k)\right] \tag{4.34}$$

The final recursive formula can be reorganized as

$$\hat{H}(k) = \hat{H}(k-1) + R^{-1}(k) a_k x(k)\left[y(k) - x^T(k)\hat{H}(k-1)\right] \tag{4.35}$$

Which together with (4.32) is recursive formula for the least squares estimate.

In recursive equation (4.35), the matrix *R(k)* needs to be inverted at each time step. In order to avoid this, introduce

$$P(k) = R^{-1}(k)$$

The recursive o*f R(k)* (4.32) becomes

$$P^{-1}(k) = P^{-1}(k-1) + a_k x(k) x^T(k) \tag{4.36}$$

The it can be able to update *P(k)* directly without needing to do matrix inversion.

Algorithm

The recursive least squares algorithm can now be given following

$$L(k) = \frac{P(k-1)x(k)}{\frac{1}{a_k} + x^T(k)P(k-1)x(k)}$$

$$\hat{H}(k) = \hat{H}(k-1) + L(k)\left[y(k) - \hat{H}^T(k-1)x(k) \right] \tag{4.37}$$

$$P(k) = P(k-1) - L(k)x^T(k)P(k-1)$$

Where $k = k_0 + 1, k_0 + 2, k_0 + 3...$, The initial values are obtained by using the LS on the first $k_0 > \dim \vartheta$ samples

$$P(k_0) = \left[\sum_{i=0}^{k_0} a_i x(i) x^T(i) \right]^{-1}$$

$$\hat{H}(k_0) = P(k_0) \sum_{i=1}^{k_0} a_i x(i) y(i) \tag{4.38}$$

4.5 Chapter Summary

Optimization method is applied into process model in order to make the process model fit real plant as much as possible. In the integrated intelligent model, it is a key way to make it adaptive so as to adapt change from real plants or outside environments. In the chapter, two optimization schemes based on different model goals are addressed so as to obtain optimal diagnosis and prediction results.

A principle of model optimization is defined according to the architecture of integrated intelligent model and optimization goals first. It includes: defining objective function, optimized variables and constraints, and then selecting optimization algorithms, etc. In this research work, the integrated intelligent model is developed for two different purposes: 1) intelligence process diagnosis and 2) process prediction. Based on the two different optimization goals, two different model optimization schemes are investigated.

To two optimization schemes, the objective functions of them are defined as an error function, which is difference between the output value of real plant and output value of intelligent model within a certain time interval. The variables (optimized parameters in model), which are main free parameters to represent and influence the performance of intelligent model, are also defined according to model structure for two different schemes. To the first case of model optimization for process diagnosis, the free parameter in the model for process diagnosis is defined by parameter of threshold ϑ. The parameter ϑ represents the sensitivity for identifying or classifying different process state as well as threshold of residual signal in fault detection. Afterward, the constraints for optimization are defined too. Finally, certain nonlinear optimization algorithms and GA are applied in model optimization for optimal process state identification, classification and diagnosis. To the second case of model optimization for process prediction without any consideration of process state changes, a matrix $H(K)$ are regarded as optimization parameters of model, Least Square (LS) approaches are involved in the kind of problems so as to estimate the optimal model parameters. An RLS algorithm is also employed for model adaptability due to the property of dynamic time-varying process. Based on the model optimization, the integrated intelligent model has adaptability for fault residual signal or process state classification as well as the optimal process prediction.

The difference between two model optimization schemes is also given in the chapter. In essence, the first model optimization can be regarded as one case of the second case. The model optimization problems of two cases have same mathematics expression, just different constraints in the optimization problems. The first model optimization scheme can provide optimal process state classification and identification with compromise in prediction ability. The second case can provide optimal process prediction without any consideration of process state classification.

CHAPTER 5 Process Control based on Model

5.1 Why Process Control is Necessary

In pervious chapters, the model based on Fuzzy TS NARX model for process state identification (as well as fault detection) and process output prediction has been developed. The aims of process state identification and process behaviors prediction are to better understand process or plant characteristics so as to make corresponding or desirable response when the process plants works on different process state. It means some special control strategies are needed for these processes based on these information if these processes are controllable.

Control is an important approach applied in dynamic processes or complex system to assure real process operation or performance to satisfy process specification, for example, feedback control, process optimal control and model predictive control even quality control and so on. Almost control strategies are developed based on system or process or system model. To modern control system, mainly work concentrated on three aspects in

Figure 5.1.

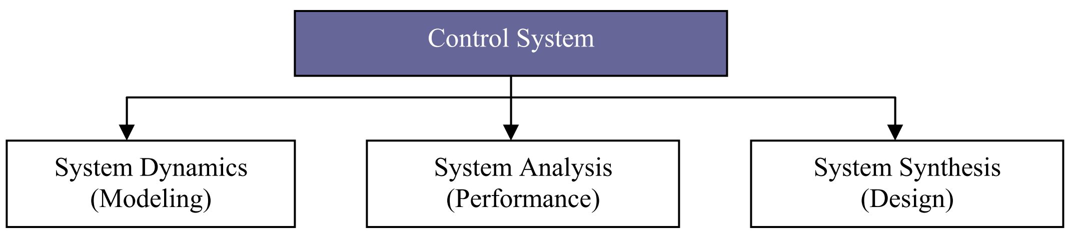


Figure 5.1: Components of a Modern Control System

Figure 5.1 shows components of a modern control system, it includes three components of modern control field. The first stage of any control system theory is to obtain or formulate the dynamics or modeling in term of dynamic equation such as *differential* or *difference* equations. Next, the system is ***analyzed*** for its performance to find out mainly stability of the system and sensitivity analysis and so on, for example, the contribution of Lyapunov stability theory. If the system performance is not according to our specification, controller design is used to improve the system performance. In this chapter, the first, a specific state space model is deduced and built from previous Fuzzy TS NARX model in order to easily implement specific control strategies and algorithms. After state space model, an important control strategy, which is call Linear Optimal Tracking Control, is introduced for implementation of process control based on the predictor of the Fuzzy TS NARX model.

5.2 Control System Model

5.2.1 Process Model

Modern control theory concerned with Multiple Inputs and Multiple Outputs (MIMO) is based on state variable representation in terms of a set of first order differential (or difference) equations. Hence, the system (plant) is characterized by state variables, say, in Linear, Time Invariant (LTI) form as:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t) \qquad (5.1)$$

Where, dot denotes differentiation with respect to (*w.r.t*) t, *x(t), u(t)* and *y(t)* are *n,r,*and *m* dimension state, control, and output vectors respectively, and A is *nxn* state matrix , B is *nxr* input matrix, C is *mxn* output matrix and D is *mxr* transfer matrix. Similarly, a nonlinear system is characterized by:

$$\dot{x}(t) = f(x(t), u(t), t)$$
$$y(t) = g(x(t), u(t), t) \qquad (5.2)$$

The modern theory dictates that all state variables should be fed back after suitable weighting, It can be illustrated as Figure 5.2 that in modern control configuration [D. S Naidu. 2002].

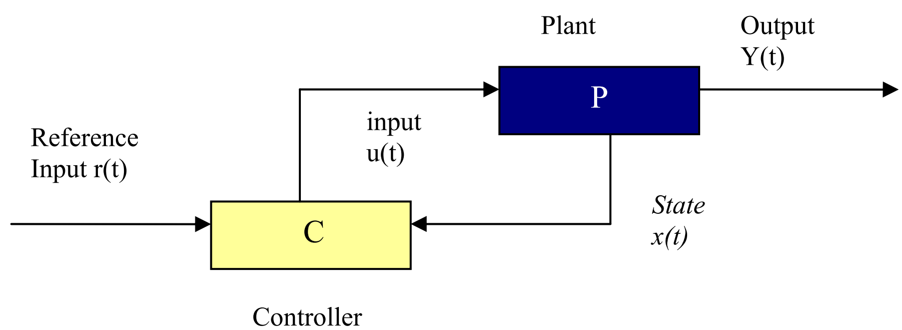


Figure 5.2: Modern Control Configuration

In this figure, the input *u(t)* is determined by the controller (consisting of error detector and compensator) driven by system states *x(t)* and reference signal *r(t)*. All or most of the state variables are available for control.

Normally, two kinds of main methods are used in modeling for plant or system as:

- The first principles (white-box) approach, for example
- The identification of a parameterized black-box model

The first-principle approach (white-box models) denotes models based on the physical laws and relationships (mass and energy balances, *etc,*) that are supposed to govern the system's behavior. In these models, the structure reflects all physical insight about the process (plant or system), and all parameters and variables have direct physical interpretations [E. Ikonen,

K. Najim, 2002]. But identification is the experimental approach to process modeling. In general, these types of methods do not require detailed knowledge of the underlying process. The chemical and physical phenomena need not be fully understood. Instead, good measurements of the plant behaviors need to be available. From the point of view, the system model developed in the thesis is base on system identification method.

To the process modeling and control, two main approaches have been developed. The first is based on the parametric model of controlled process. The parametric model could be described in the form a transfer function, or in the state-space domain. An important disadvantage of using the parametric model is that it represents a linearized model of the process. The control of strong nonlinear processes could be unsatisfactory. The second approach proposed is based on the nonparametric model. The advantage of this approach is that the model coefficient can be obtained directly from samples of the input and output response without assuming the model structure. In the chapter, the process control based on fuzzy TS NARX model represents a combination of the nonparametric and parametric approaches. Model structure is determined by state space model (parametric model way) and the model coefficients are determined by system identification method (nonparametric model way).

Hence, from control theory points of view, developing a control system model is a key and first step for process control. In next section, the control system model based state space equation will be built and deduced from previous Fuzzy TS NARX model, which is developed for process state identification and process output prediction with intelligence technologies.

5.2.2 State Space Equation

An intelligence model based on dynamic Fuzzy TS NARX model has been established for process state identification (as well as fault detection) and process output prediction in chapter 3. In this section, further analysis from this Fuzzy TS dynamic NARX model indicates that the model can be represented with state space description. It means some control strategies and methodologies can be applied based on this model so as to make process to reach our specification.

State space expression from Fuzzy TS NARX model

To a discrete process or system, in Figure 3.14, prediction value of the Fuzzy TS NARX model of the process can be calculated by equation (4.6). (here, for simplification, only three process states are defined). Based on the characteristics of $w_{i1}(k)$, $w_{i2}(k)$, $w_{13}(k)$ and $w_{u1}(k)$, $w_{u2}(k)$, $w_{u3}(k)$ in the integrated intelligent model in equation (4.7)(4.8), it can be denoted with matrix and vector form as equation (4.9) as well as following equation (5.3).

$$X(k+1) = A(k)X(k) + B(k)X(k) \tag{5.3}$$

Here,

$$A(k) = a\begin{bmatrix} w_{i1}(k) & 0 & 0 \\ 0 & w_{i2}(k) & 0 \\ 0 & 0 & w_{i3}(k) \end{bmatrix} \quad B(k) = b\begin{bmatrix} w_{u1}(k) & 0 & 0 \\ 0 & w_{u2}(k) & 0 \\ 0 & 0 & w_{u3}(k) \end{bmatrix} \tag{5.4}$$

Furthermore, Acceding to the model optimization schemes in chapter 4, two different model schemes are given according to different model optimization goals.

- Process state identification (as well as fault detection) for process diagnosis and
- Process output prediction

To the first schemes, the model optimization is based on process state identification so as to provide good process monitoring and diagnosis performance. The result of the model optimization is given in equation (4.14). The final model prediction output with ϑ adjustment can be written in vector form as:

$$Y(k+1) = H(k,\vartheta)^T * X(k+1) \tag{5.5}$$

Here, $H(k,\vartheta)$ can be regarded as transform matrix and *X(k+1)* is 3X1 vector, which is given:

$$X(k+1) = \begin{bmatrix} X_1(k+1) \\ X_2(k+1) \\ X_3(k+1) \end{bmatrix} = aX(k) + bU(k) = \begin{bmatrix} aw_{i,1}(k)X_{i,1}(k) + bw_{u,1}(k)X_{u,1}(k) \\ aw_{i,2}(k)X_{i,2}(k) + bw_{u,2}(k)X_{u,2}(k) \\ aw_{i,3}(k)X_{i,3}(k) + bw_{u,3}(k)X_{u,3}(k) \end{bmatrix}$$

Based on the model optimization, the corresponding $H(k,\vartheta)$ can be gotten. Thus, the vector $H(k)$, which is determined by the result of optimized threshold value ϑ_i, can be expressed as:

$$H(k,\vartheta)^T = \begin{cases} [1 \quad 1 \quad 0] & \text{if } w_1 > \vartheta_1 \\ [0 \quad 1 \quad 0] & \text{if } w_2 > \vartheta_2 \\ [0 \quad 1 \quad 1] & \text{if } w_3 > \vartheta_3 \end{cases} \tag{5.6}$$

To the second schemes, the model optimization is to achieve better process prediction. The result of the model optimization is addressed in equation (4.15) in chapter 4. The final model output without any consideration of ϑ adjustment can be written in vector form as:

$$Y(k+1) = H(k+1)X(k+1) \tag{5.7}$$

In this case, the corresponding $H(k)$ can be gotten by LS approach, which is presented in chapter 4.

Combining the equations (5.4), (5.5) and (5.7), they can be expressed using matrix and vector as following form (5.8):

$$\begin{aligned} X(k+1) &= A(k)X(k) + B(k)U(k) \\ Y(k+1) &= H(k+1)X(k+1) \end{aligned} \tag{5.8}$$

where, $Y(k+1)$ is the optimal model output. It means that the final prediction value of the integrated intelligent model is generated by combination of fuzzy TS NARX model and certain optimization schemes. Namely, The final prediction is determined by two factors: the output value *X(k+1)* of fuzzy TS NARX model and optimization unit *H(k)*.

In equation (5.8), control model coefficients *A(k)* and *B(k)* are obtained from Fuzzy TS NARX model and *H(k)* is identified by optimization unit in the integrated intelligent model.

From equation (5.8), *A(k), B(k) and H(k)* can be regarded as state transition matrix ($n \times n$), input transition matrix ($n \times n$) and state observer vector ($n \times 1$), respectively. They are time variant with time (k). Hence, from control system point of view, the system (5.8) can be regarded as Discrete Linear Time-Variant System (DLTV) model. That means, in the thesis, a Fuzzy TS NARX model is first developed for process state identification (as well as fault detection and diagnosis) and process behaviors prediction, it also can be deduced and expressed as a Discrete Linear Time-Variant system (DLTV) model to implement process control.

The equation (5.8) also indicates that the intelligence model, which is expressed with Fuzzy TS NARX model for process state identification (as well as Fault detection) and process behavior prediction, can be fully described by the specific discrete state space equation.

Based on the control system model expressed with state space equation (5.8), some control strategies can be applied so as to control real process behaviors. That means the whole model developed in the thesis can be fully applied in s*ystem process diagnosis (state identification or fault diagnosis), process behavior prediction and process control*.

Hence, from technical point of view, the whole model can be explained as:

The process model is developed for process control by description of a Discrete Linear Time Variant system (DLTV) with state space equation, which is deduced from previous Fuzzy TS NARX dynamic model and certain model optimization schemes. The Fuzzy TS NARX dynamic model is designed for process state identification (as well as fault detection) or process behavior prediction, its model parameters are identified by system identification way with Neural Networks, parameter estimation and system optimization algorithms.

Hence, the whole model can be illustrated in Figure 5.3 below.

In the figure, the state space equation (5.8) of control model is identified by the Fuzzy TS NAXR model and optimization units in the integrated intelligent model.

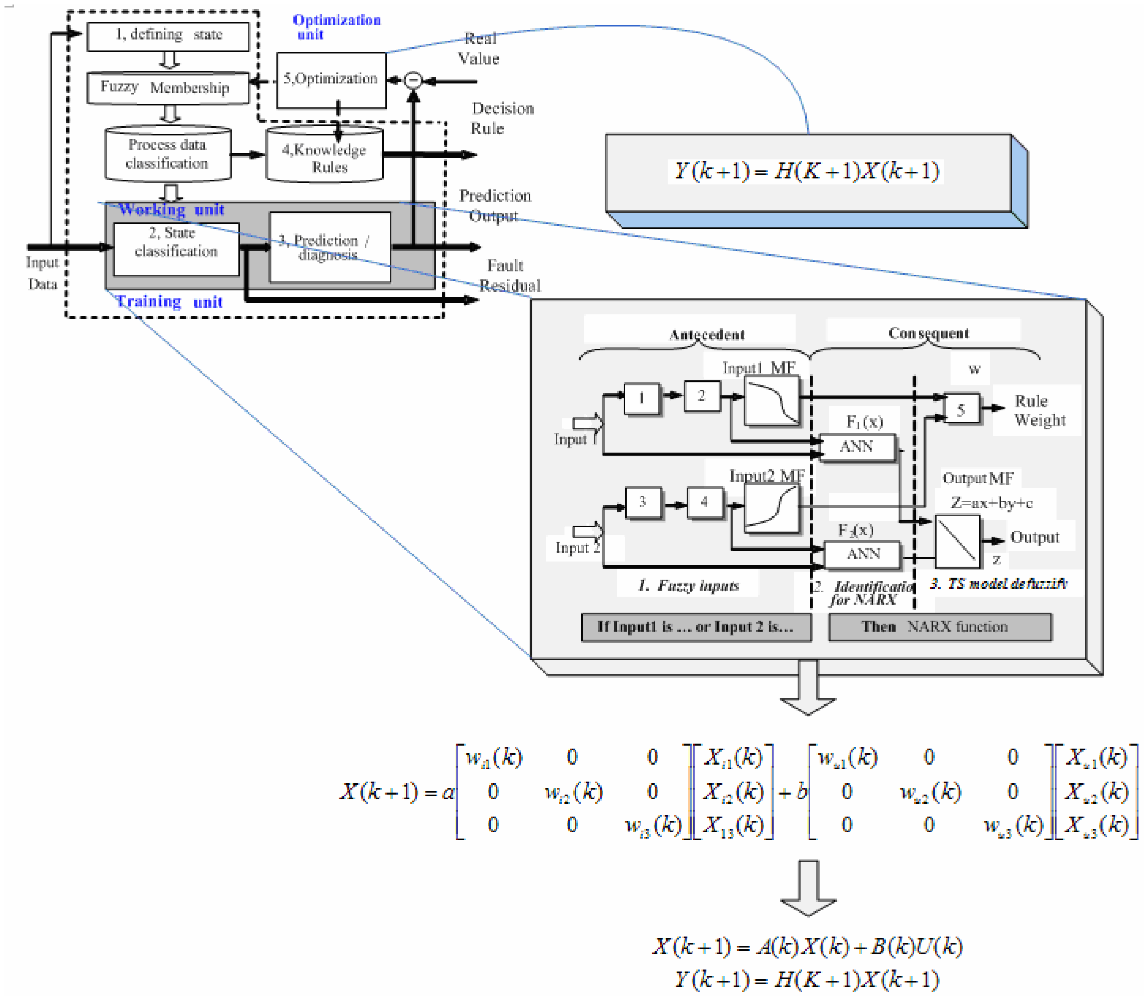


Figure 5.3: Inner structure and relationship in integrated intelligent model

After control model is obtained or identified, certain process control methods can be applied for process control purposes, for example, Model Predictive Control (MPC) method for process control, optimal control for process optimal control, etc[D. S Naidu, 2002] [E. Ikonen, 2002]. In the next section, optimal process control is suggested and introduced to implement process control based on the integrated intelligent model.

5.3 Optimal Process Control

5.3.1 Optimal Control

In optimal control theory, the design usually with respect to a performance index, the main objective of optimal control is to determine control signals that will cause a process (plant) to satisfy some physical constraints and at the same time reach (maximize or minimize) a chosen performance criterion (performance index or cost function). According to the figure 5.2, the optimal control task is to find the optimal control $u^*(t)$ that plant P from initial state to find state with some constraints on controls and states and at the same time extrmizing the given performance index J. the formulation of optimal control problems need:

1. A mathematical description (or model) of the process
2. A specification of the performance index and
3. A statement of boundary conditions and the physical constraints on the states and /or controls.

The closed loop optimal control of linear plants or systems with quadratic performance index or measure is Linear Quadratic Regulator (LQR) system dealing with state regulation, output regulation, and process or system tracking.

5.3.2 Problem Formulation

A Linear Time-Varying (LTV) system can be denoted as:

$$\begin{aligned} \dot{X}(t) &= A(t)X(t) + B(t)u(t) \\ y(t) &= C(t)X(t) \end{aligned} \tag{5.9}$$

With a Cost Function (CF) or performance index (PI) [D. S Naidu. 2002]:

$$J(u(t)) = J(x(t_0), u(t), t_0)$$
$$= \frac{1}{2}\left[z(t_f) - y(t_f)\right]' F(t_f)\left[z(t_f) - y(t_f)\right] + \frac{1}{2}\int_{t_0}^{t_f} \left(\left[z(t) - y(t)\right]' Q(t)\left[z(t) - y(t)\right] + u'(t)R(t)u(t)\right)dt \tag{5.10}$$

Here; *x(t)*: state vector; *y(t)*: output vector; *z(t):* reference vector; *u(t)*: control vector; $e(t) = z(t) - x(t)$: error vector; ***A(t)***: *nxn* state matrix; ***B(t)***: *nxr* control matrix, ***C(t)***; *mxn* output matrix.

We can find that the optimal control is closed-loop in nature, that is the optimal control ***u(t)*** is a function of the state ***x(t)*** or the output ***y(t).***

Three system categories based on different cost function and requirements:

1. *State Regulator*
   Keep the state *x(t)* near zero (i.e., z (t)=0 and C=I): the objective is to obtain a control *u(t)* which takes the plant from a nonzero state to zero state, for example, a plant is subjected to unwanted disturbances that perturb the state.
2. *Output Regulator*
   Keep the output *y(t)* near zero.
3. *Tracking System*
   Keep the state and output near a desired state or output.

Analysis for various Matrixes in cost function (5.10).

1. The *Error Weighted Matrix* ***Q(t)***

Must be positive semidefinite in order to keep the error *e(t)* small.

2. the *Control Weighted Matrix* ***R(t)***

   The quadratic nature of the control cost expression $\frac{1}{2}u^{'}(t)R(t)u(t)$ indicates that one has to pay higher cost for larger control effort, since the cost the control has to be a positive quantity, the matrix ***R(t)*** should be positive definite.

3. Control signal ***u(t)***

   Assumption for no constrains for control signal.

4. The *Terminal Cost Weighted Matrix* $\boldsymbol{F(t_f)}$:

   The main purpose of the term is to ensure the error small at final time t_f, is as small as possible, to guarantee this, the corresponding matrix should be positive semidefinite.

5. *Infinite Final Times* $F(t_f)$

   $F(t_f)$ must be zero due to no any realistic sense.

In a real process, a reference output is desired, hence, in this part, Linear Quadratic tracking (LQT) system is discussed based on this system model, which is deduced in chapter 5.2.

In tracking (trajectory following) system, the control goal is to obtaining a closed-loop control scheme that enables a given system track (or follow) a desired trajectory over the given interval of time. Based on the state space model, optimal algorithms and control objects, optimal control system can be designed according to following process [D. S Naidu. 2002].

5.3.3 Design for Optimal Process Control

The formulation

A Discrete Linear Time Varying System is described by the following state equation

$$X(k+1) = A(k)X(k) + B(k)u(k)$$
$$Y(k) = C(k)X(k)$$

The performance index to be minimized is

$$J = \frac{1}{2}\left[C(k_f)X(k_f) - Z(k_f)\right]^{'} F(t_f)\left[C(k_f)X(k_f) - Z(k_f)\right] + \frac{1}{2}\sum_{k=k_0}^{k_f-1}\left\{\left[C(k)X(k) - Z(k)\right]^{'} Q(k)\left[C(k)X(k) - Z(k)\right] + u^{'}(k)R(k)u(k)\right\} \tag{5.11}$$

Where, *x(k), u(k),* and *y(k)* are *n, r,* and *n* order state, control and output vectors, respectively. Here, *F(k)* and *Q(k)* are each *nxn* dimensional positive semidefinite systematic matrices, and *R(k)* is an *rxr* positive definite symmetric matrix. The initial condition is given as $x(k_0)$ and the final condition $x(k_f)$ is free with k_f fixed. The control goal is the error $e(k) = y(k) - z(k)$ as small as possible with minimum control effort, where $z(k_f)$ is n dimensional reference vector. The methodology to obtain the solution for the optimal tracking system is carried out using the following steps [D. S Naidu. 2002].

.

1. Hamiltoonian
2. state and costate system
3. open-loop optimal control
4. Riccati and vector equations
5. Closed-loop optimal control

The details derivations are given in [D. S Naidu. 2002]. The implementation procedure for the linear quadratic tracking system is summarized in table 5.1.

Table 5.1 The implementation procedure of linear quadratic tracking system

A: statement of the Problem	
Plant $X(k+1) = A(k)X(k) + B(k)u(k)$ The output relations as $y(k) = C(k)X(k)$ Performance index: $J(k_0) = \frac{1}{2}\left[C(k_f)X(k_f) - z(k_f)\right]' F(t_f)\left[C(k_f)X(k_f) - z(k_f)\right]$ $+\frac{1}{2}\sum_{k=k_0}^{k_f-1}\left\{\left[C(k)X(k) - z(k)\right]' Q(k)\left[C(k)X(k) - z(k)\right] + u'(k)R(k)u(k)\right\}$ Boundary conditions $x(t_0)=x_0$, t_f is free, and k is fixed Find the optimal control and state	
B: Solution of the problem	
Step 1	Solve the matrix Differential Riccati Equation Problem (DRE) $P(k) = A'(k)P(k+1)\left[I + EP(k+1)\right]^{-1} A(k) - V(k)$ With $P(k_f) = C'(k_f)F(k_f)C(k_f)$, where $V(k) = C'(k)Q(k)C(k)$ and $E(k) = B(k)R^{-1}(k)B^{-1}(k)$
Step 2	Solve the vector difference equation $g(k) = A'(k)\left\{I - \left[P^{-1}(k+1) + E\right]^{-1} E\right\}g(k+1) + W(k)z(k)$ With $g(k_f) = C(k)F(k)z(k_f), where, W(k) = C(k)Q(k)$
Sttep 3	Solve the optimal state x*(t) from $\dot{x}^*(k+1) = \left[A(k) - B(k)L(k)\right]x^*(k) + B(k)L_g(k)g(k+1)$ Where, $L(k) = \left[R + B'(k)P(k+1)B(k)\right]^{-1} B'(k)P(k+1)A(k)$ and $L_g(k) = \left[R + B'(k)P(k+1)B(k)\right]^{-1} B'(k)$
Step4	Obtain the optimal control *u*(t)* as

	$u^*(k) = -L(k)x^*(k) + L_g(k)g(k+1)$.

The implementation of the discrete-time optimal tracker is shown in Figure 5.4. [D.S. Naidu, 2002].

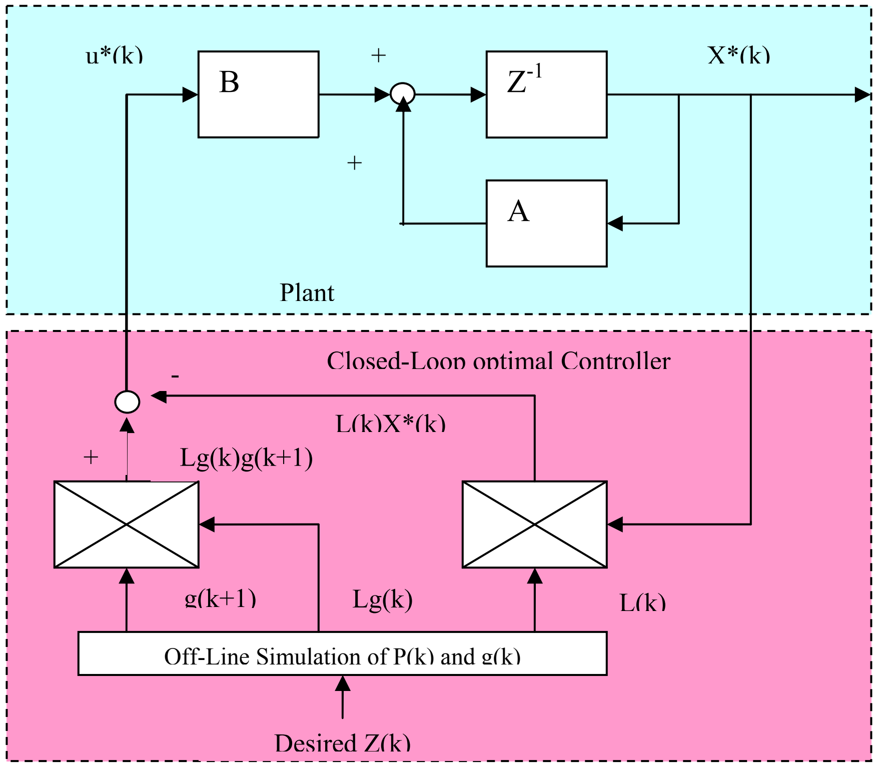


Figure 5.4: The implementation of discrete time optimal tracker

5.4 Summary

In this chapter, the first, the control model based on state space equation is deduced from pervious Fuzzy TS NARX model and optimization unit, which is developed for process state identification, diagnosis and process output prediction. It means that the integrated intelligent model can be fully described by a specific state space of discrete system. In the state space equation, the state transition matrix ***A*** is determined by the fuzzy degree of process state classification produced by process historical behavior at time t instant. The input transition matrix ***B*** is determined by the fuzzy degree of process state classification, which is produced by real process inputs at time t instant. The state observer vector ***H*** is determined by the optimization unit results. Based on the control model, optimal process control methods are investigated so as to provide good control ability for whole process. The control result can make real processes to follow real process specification.

Summary of all model developed in the thesis, from technology point of view,1)The process model, which is described by a Linear Time Variant system (LTV) with state space equation and deduced from previous Fuzzy TS NARX dynamic model, can be applied in process analysis and control. 2) The Fuzzy TS NARX dynamic model is developed for process state identification (as well as fault detection), process diagnosis and process behavior prediction based on certain model architecture design, its model parameters are identified by system identification ways with Neural Networks and system optimization algorithms. 3) Based on the model, process performance and control can be investigated and implemented.

CHAPTER 6 CASE STUDY-PREDICTION FOR SUPPLY CHAIN PROCESS

6.1 Business Processes and Business Intelligence

6.1.1 Business Processes

One basic definition of a process is:

"…a logical series of related transactions that converts input to results or output."

Davenport & Short (1990) defined business process as "*a set of logically related tasks performed to achieve a defined business outcome*." A process is "*a structured, measured set of activities designed to produce a specified output for a particular customer or market, it implies a strong emphasis on how work is done within an organization*" (Davenport 1993). In their view processes have two important characteristics: (i) They have customers (internal or external), (ii) They cross-organizational boundaries, i.e., they occur across or between organizational subunits.

Business process comprises all things we do to provide someone who cares with what they expert to receive. It also contains all the actions we take when we fail to meet those expectations.

Within any true processes, inputs of all types-such as raw materials, information, knowledge, commitments, and state are transformed into outputs and results. This transformation occurs according to process guidance, such as policies, standards, procedures, rules, and individual knowledge. Reusable resources are employed to enable the change to happen. These resources include facilities, equipment, technologies, and people, etc.

Therefore, monitoring, diagnosing process activities and predicting process output in business processes are important methods to ensure business activities correct and effective toward the business goals.

6.1.2 Problems and Challenges in Business Processes

Business process is more complex due to more human being's factors involved, for example, business policies, business goals and business rules and standards, etc. The performance of business processes can not only fully rely on some business data, also rely on human being factors. At the same time, the evaluation standards for business processes and performances are also different in different companies. Some of criteria are difficult to quantify as guideline or indictor. Some key variables (or important factors) in business processes are impossible to get the quantified values. Based on these factors, it is difficult to monitor, diagnose and evaluate, even improve business processes. Some main problems in business processes are depicted as follows:

- More factors are involved in the business processes, especially in human being's factors.
- Most of business operations, control rules and business process guidelines are based on linguistic instruction, these variables are difficultly treated. Some process variables (input variables, output variable), process goals are also difficultly to be quantified.
- No comprehensive and effective standards and methods to evaluate performance of business processes
- In complex business processes, it is difficulty to find some potential key factors affecting the business process performance
- It is also difficult to build an effective business models and evaluation systems

Some researches about business process management and improvement, especially in business process model, have been done for recent few decades [T. Burlton, Roger 2001]. However, most of these researches focus on management and business operations. In this thesis, the integrated intelligent model concentrates on modeling for business process with data analysis and processing in order to quantitatively analyze some business processes, diagnose some business problems and discover some process rules. The main goals to build process model as follows:

- State measurement for business processes, such as abnormal state identification, abnormal behavior prediction, performance evaluation for business processes etc
- Behavior and output prediction for business processes
- Business rule discovery, especially for state change and risk analysis

In this Chapter, a practical business case with the application of integrated intelligent model developed in this paper is presented so as to solve some practical business problems.

6.1.3 Business Intelligence

What is Business Intelligence?

In simplest *terms*, business intelligence is any effort to capture and analyze business data in order to understand clearly, improve business processes, and compete more effectively.

By providing the right information to the right people at the right time, business intelligence enables companies to make better decisions faster than ever before. It can perform to transform data into knowledge, knowledge into action, and action into success.

The measure of any business intelligence solution is its ability to derive knowledge from data, the capability to process volumes of information and identify patterns, tends, rules, and relationships that are too large to be handled through simple human analysis.

What does business intelligence analyze?

The efficiency of enterprise solutions ERP(Enterprise Recourse Management), CRM(Client Relationship Management), SCM (Supply Chain Management) and much more has made it easier than ever to collect business information. If your organization is typical, every department gathers data: behavioral data about enterprise's customers; performance data about operational effectiveness, manufacturing efficiency and inventory state; marketing data on enterprise's competitors and demographic data about changing markets; Supply Data about suppliers and partners; plus web click-stream data and internet commerce data.

Take together, all of that data could provide a detailed and accurate perspective of every aspect of enterprise business. For most large organizations, the information infrastructure consists of mix of proprietary and legacy systems. Intelligence and compatibility issues ensure that data tends to remain trapped in the hands of the department that collected it.

Business intelligence opens that data up to everyone in your organization by providing the means to extract it from operational applications, convert it to a standard format, and then store it in a central location optimized for both rapid delivery of summarized information and more detailed query and analysis. The result is an immediate, personalized, relevant view of all available information that goes far beyond the standard reporting abilities of transaction-based systems, enabling faster, better-informed decision-making at every level of your organizations.

Business intelligence provides the means for giving each group an appropriate view of corporate data it needs to be successful:

- **Sales and marketing**: Business intelligence offers powerful new tools for understanding customer needs and responding to new market opportunities. With comprehensive business intelligence system in place, marketing analysis can gauge the effect of pricing and promotions, target customer segments more accurately and develop true real-time one-to-one marking.
- **Operation:** From quality control to inventory management to production planning, business intelligence is a mechanism for analyzing the performance of any operational process.
- **Finance**: by providing financial planners with immediate access to real-time data. Business intelligence builds new value into all financial operations, including budgeting and forecasting.
- **Customer service**: Business intelligence tools let enterprise accurately assess the value of market segments and individual customers, and help you retain the customers that deliver the most profit to your company.
- **Supplier relation**: business intelligence takes advantage of online supplier and partner integration to provide new levels of supplier performance analysis, new design collaboration opportunities, and more.

Business intelligence can be implemented based on intelligent technology and real business process model and rules. In business intelligence, intelligence technology and data mining play two important roles in modeling business process and discovering knowledge for specific business goals. Therefore, sufficiently understanding business process, rules and model, and then combining specific intelligence and data mining techniques, are critical means to implement the business intelligence.

6.2 The Description of Supply Chain Processes

A supply chain process can be defined in a number of different ways. A **supply chain** is a network of facilities and distribution options that performs the functions of procurement of materials; transformation of these materials into intermediate and finished products; and distribution of these finished products to customers [Ganeshan. R, Harrison, T.P., 1995].

A supply chain essentially has three main parts, the supply, manufacturing and distribution:

- The **supply** side concentrates on how, where from and when raw materials are procured and supplied to manufacturing.
- **Manufacturing** converts these raw materials to finished products and
- **Distribution** ensures that these finished products reach the final customers through a network of distributors, warehouses and retailers.

The chain can be said to start with the suppliers of your suppliers and ends with the customers of your customer. A chain of logical connected, repetitive activities that

- Utilizes the enterprise's resources to
- Refine an object (physical or mental)
- For the purpose of achieving specified and measurable results /products for
- Internal or external customers.

The process is depicted as Figure 6.1.

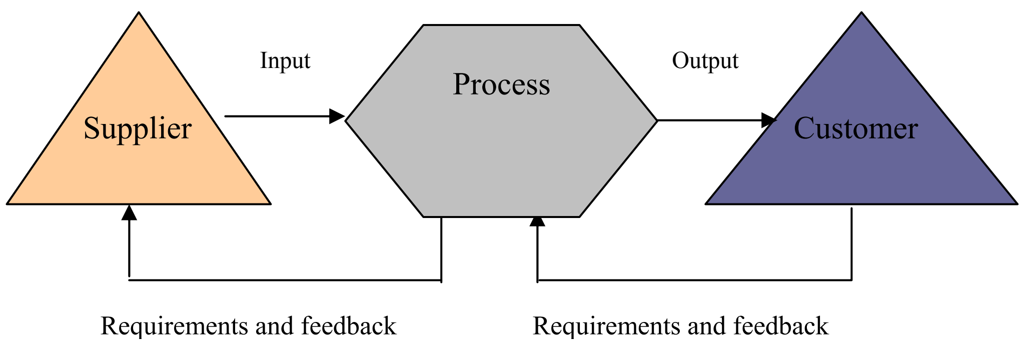


Figure 6.1: The process with supplier and customer

6.2.1 *Problems Definition*

One of the major challenges facing any business is how much to charge for its products and services with market change. In production and supply chain processes, the change of market directly influences on a few of input factors during whole process. Norsk Kjøttsamvirke AS is a major large enterprise in the Norwegian distribution market. Its main business is to collect different kinds of products from many different factories and then sales, provides and distributes them for different store markets in Norway. The company wishes to increase prediction ability and flexibility to organize and plan their production, sales, store and transport. This will further help them to decide the price and amount for the market. The aim of the project is to develop a *supply-forecasting model* and *business* *analysis model* for fast making decision and response to market. The further requirements to analyze these rules and causes for market change are also desired. All relevant data of product supply and requirement from customers have been received from SINTEF Industrial Management in Norway. The data set contains a list of the customers and their goods order every day every year.

To the large enterprise of products supply and distribution in Norway, a lot of kinds of products and an amount of money are involved in the supply chain process every day. The product amount, type and price for supply and distribution to different store market in Norway mainly depend on the consumer market and production cost. To the distribution enterprise, it can benefit from a good prediction result in future for product plan, transport, store and supply, etc. Hence, it is very significant to accurately predict supply price and product amount for the enterprise every day. Concretely speaking, the main aims in the supply chain process are described as follows:

- *Modeling for business process:* Modeling real supply chain process based on the inner business laws, user requirement and business goals so as to approximately describe process behaviors in the supply chain process.
- *Forecasting market change of products:* Predicting product demands from customer and market change correctly on certain degree. Prediction result with high accuracy is desired with the models.
- *Business process analysis, evaluation and risk identification:* Diagnosing and analyzing the supply chain process and discovering business rule for process performance and risk identification. All information is important to control and improve business process performance and behaviors.

6.2.2 *The Analysis for Supply Chain Process and Business Data*

6.2.2.1 *Supply Process Analysis*

To Norsk Kjøttsamvirke AS, the main business and production process is a typical supply chain process. Many factors, for example, the money, products, providers and customers, market, transport, store, plan have been involved in the supply chain process. The whole business process can be described as a node of whole supply chain system as Figure 6.2.

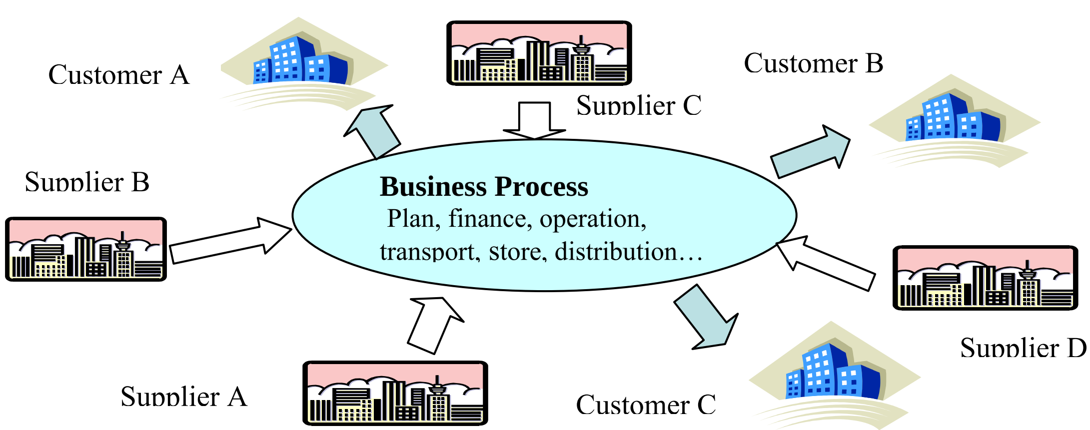


Figure 6.2: The business process descriptions

In the supply chain process, Norsk Kjøttsamvirke AS first collects a variety of goods from different manufacture enterprises based on market plan and customer demand, and then supply, sale, distribute these goods to different end-market. It plays a key role and node to link goods production and market sale. The main business problem to this company is how to plan and use their money, product and store and transport for correct market requirements with high efficiency and low cost.

Normally, a supply chain process in a life cycle of production process can be described as Figure 6.3 [T. Burlton, Roger, 2001].

Customer demand: It is a key factor and final purpose for supply chain process. The following activities are decided according to the customer demand and enterprise development. One aspect, these companies do their best to satisfy current customer demand, on the other aspect, potential customer is also desired to further help, organize and improve supply chain process.
Production plan: According to market or customer demand, some overall plans are needed to organize and arrange real activities in whole production process.
Procurement: It includes all activities to buy relevant products or materials according to the production plan.
Operation: Some necessary operation for whole production process and supply chain process, etc. For example, design, manufacturing, buy, supply, transport, store, etc.
Delivery: production delivery process ensures final production can reach the customer and market on time at high efficiency and flexibility.

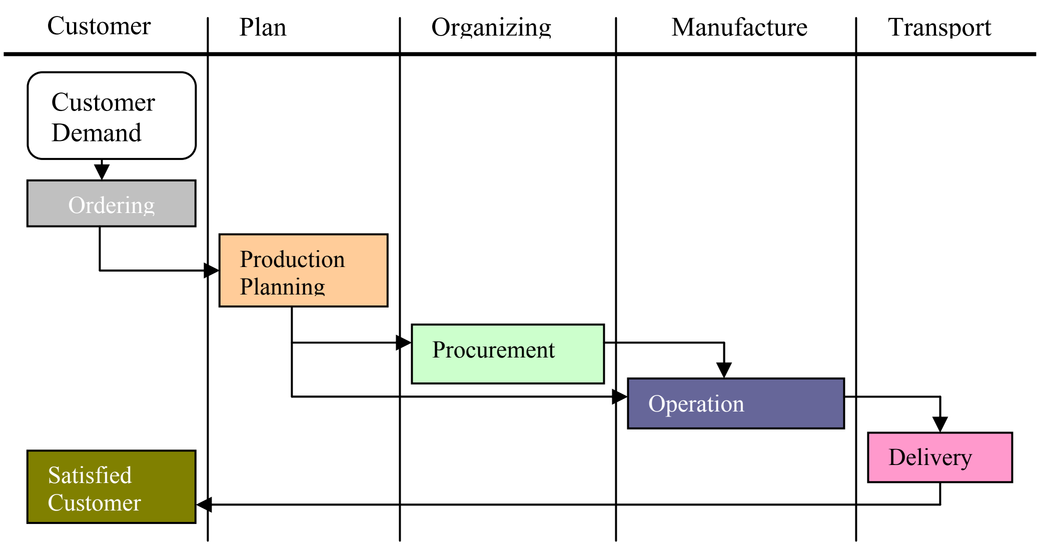


Figure 6.3: A basic process in production process in a life cycle

In this process, the plan plays the key factor to effectively implement whole business process, production process and supply chain process. The plan is made according on the fully understanding of current process state and certain prediction ability for future. It is tightly linked with market and customer.

6.2.2.2 Detail Business Goals

Based on above analysis of whole supply chain process, the detail process goals are given as below:

1. Accurately predicting product amount for market. It not only predict the tendency of product amount change with the time from global view, but also can capture the abrupt change factors and give the good prediction result in future under the circumstance of the abrupt state.
2. Further discovering the underlying rules that result from occurrence of abrupt behaviors in the supply chain process in order to provide risk analysis and control. It is significant to predict business behavior under this circumstance.
3. Measuring performance and process states of the supply chain in order to provide flexibility and decision-making support for supply chain process.

The traditional way is to analyze process history data using some statistic analysis method for further making decision. These statistic methods can provide certain analysis and prediction ability but also has some disadvantage when higher requirement is need or more complex process is involved. Based on the reason, the intelligence technologies and data mining and knowledge discovery are involved in the field and provide better solution.

6.2.2.3 Business Data Analysis

Business data type

The part of the data items provided by Norsk Kjøttsamvirke AS is listed in Appendix A. The main data is shown in Table 6.1 in detail.

Table 6.1: The Basic data in product supply in Norsk Kjøttsamvirke AS (Goods number: 472674)

Year	Week	Tekst	kjede	Name	Mon	Tues	Wends	Thru	Fri.
2000	36	Karb Kaker 0,8kg Pk	04153	ISP NKL-kjede Prix	256.8	121.6	183.2	240.8	258.4
2000	36	Karb Kaker 0,8kg Pk	04157	ISP NKL-kjede S-Marked	56.8	53.6	27.2	123.2	84.8
2000	36	Karb Kaker 0,8kg Pk	04160	ISP NKL-kjede Mega	236.8	112	84.8	154.4	100.8
2000	36	Karb Kaker 0,8kg Pk	04400	ISP Hakon-kjede Ica	20	20	16	16	84
2000	36	Karb Kaker 0,8kg Pk	04550	ISP NKL-kjede Obs	40	0	0	40	120
2000	36	Karb Kaker 0,8kg Pk	10000	ISP Storkjøkken	0	0	0	0	0

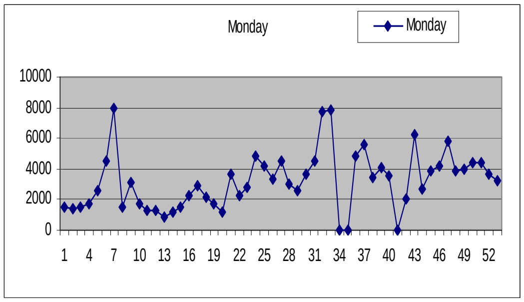
…

From the data table, main data types include supply date, product types and name, customer name and code, sale price, product amount from Monday to Friday, supply amount from Monday to Friday in real supply process.

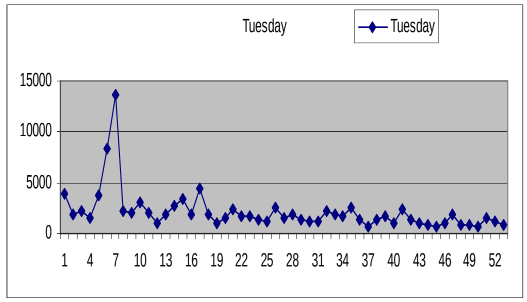
Data distribution and tendency

Figure 6.4 shows data distribution and tendency of product supply (product number 472674) with time series. From the figure, it is clear to see that most data distribution and change are in certain range in which the data distribution and change is easy to be predicted. Then we call these data in the range as “normal data”. The process with these normal data can be called as “normal state”. Only a little data outside of normal range has drastic fluctuation and abrupt change, and then these data can be called as “abnormal data” and the corresponding supply process state is called “abnormal state”.

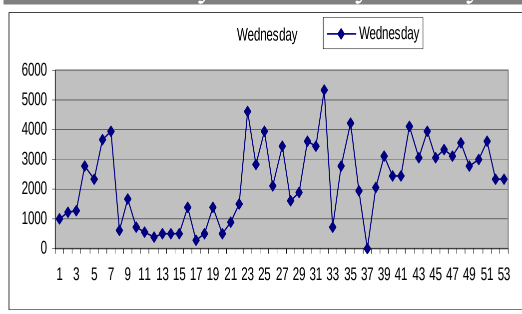
Data from every Monday in one year



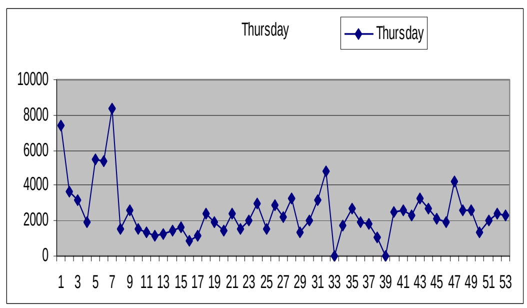
Data from every Tuesday in one year



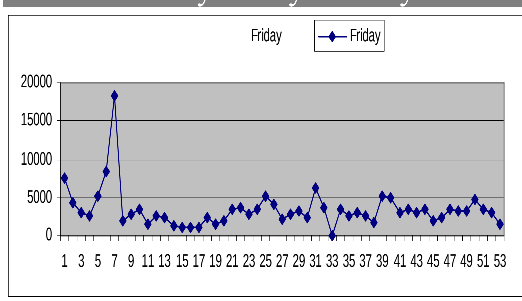
Data from every Wednesday in one year



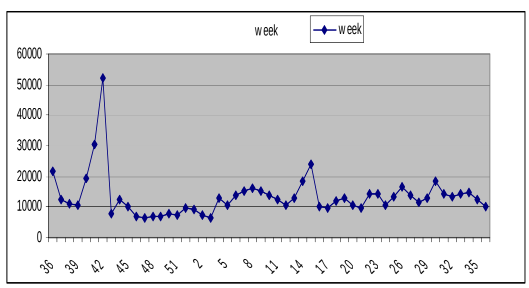
Data from every Thursday in one year



Data from every Friday in one year



Data from week by week in one year



Data from day by day in one year

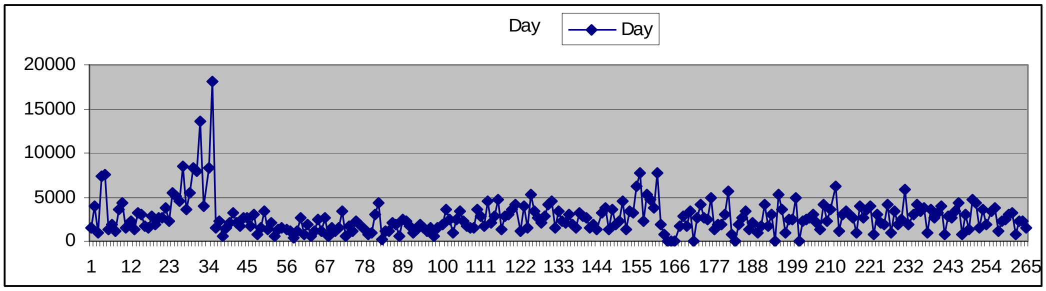


Figure 6.4: The data distribution and tendency with time in one year

(From 36th week 1999 to 36th week, 2000)

As we known, it is easy to predict the normal process behavior in this supply chain process due to the similar data attribute, tendency and certain fluctuation range, but it is difficult to deal with the abnormal behavior and data in this supply chain process. So, the big challenges under the circumstance are:

- Identify the process state (normal state or abnormal state) based on system inputs
- Predict the value of these data in abnormal state and their development tendency
- Discover the causes or relationship between process state change and process inputs

The integrated intelligent model based on NARX Fuzzy TS model in this thesis will answer and solve the kind of problems.

6.2.2.4 Some Solutions for Supply Chain Prediction

The prediction model based on Single ANNs

Based on the data provided by Norsk Kjøttsamvirke AS, a simple analysis can be done by some traditional way. Here, an approach employed ANN models for product amount prediction is proposed and the principle of the ANN model structure for forecasting is shown in Figure 6.5.

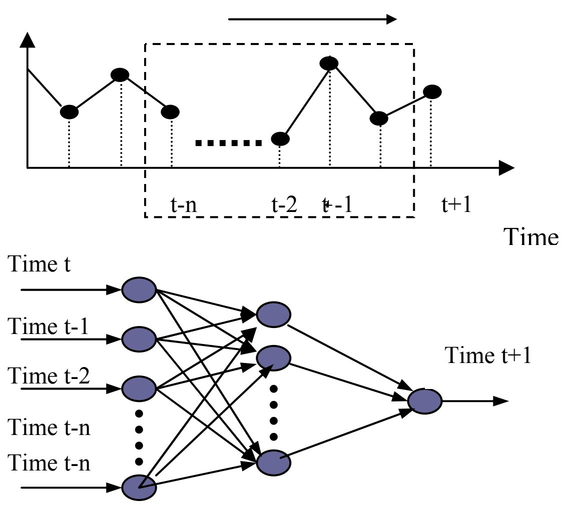


Figure 6.5: The basic principles of ANN based prediction model

In the figure, the fluctuating variables are sampled at an appropriate rate (every day, every week), and then the samples are introduced into the input layer. But, not all previous sample values are regarded as the input value. At every time increment, a new sample value is introduced into the topmost input neuron and a sample value in the bottommost input neuron is discarded. It indicates that the value next day (Time t+1) is only relevant to samples before several days.

In the case, we list the one-month data and give their graph as Figure 6.4. It can be seen that the time series model of the case is typical seasonality and the data in each week has similar characteristics. They present some similarity and periodic disciplinarian. So it is a best choice to decide the period N=5. Namely, only previous 5 days samples are regards as the input data so as to predict the value next day. At the same time, the product amount data of last week is also important to predict the value of next day because product amount data per week comprises season factor which is a critically important factor to affect consumer market.

The prediction result is got according to following training data sets and test sets in Table 6.2, during training process, the training error is set up as 0.03 and the training error is shown as Figure 6.6.

Table 6.2: The data sets for training NNs model for supply prediction of product 472674

Data Set	Corresponding Date	Record number	Percentage
Total data	From 36th week 1999 to 36th week 2000, one year	265	
Training data	From 36th week 1999 to 30th week 2000, 47 week	234	88.3%
Test data	From 30th week 2000 to 36th week 2000, 6 week	31	11.7%

The result of product supply prediction (From 30th week 2000 to 36th week 2000) is shown in Table 6.3 [Wang, K.S, Tang, M, 2002]

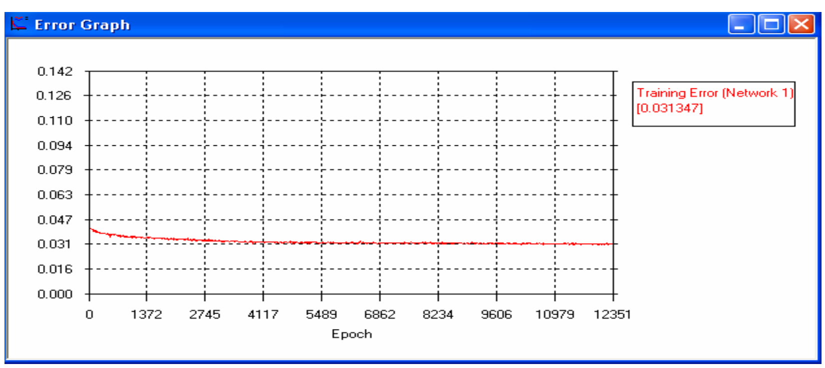


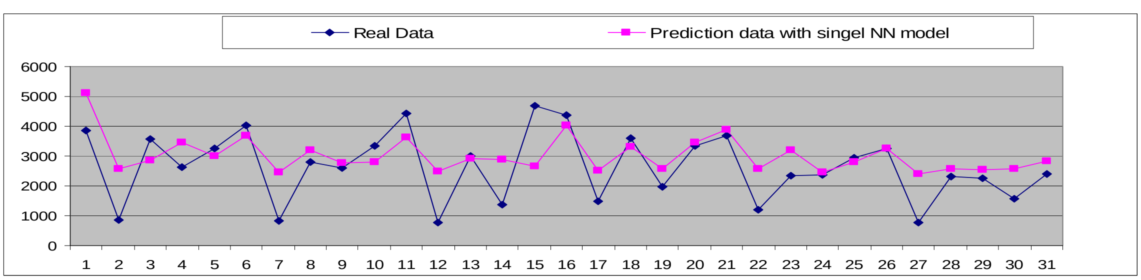
Figure 6.6: Training error of Single ANN Model

Table 6.3: The prediction result for product amount (472674) with single NNs model

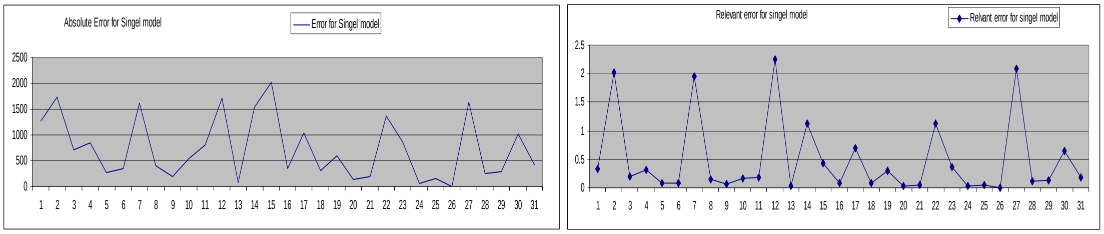
(From 30th week 2000 to 36th week 2000)

Date	Real Data	Prediction Data with single NN Model	Error(S)A	Error(S)R
1	3845.6	5120.780827	1275.180827	0.331594765
2	853.6	2575.974504	1722.374504	2.017777066
3	3571.2	2865.307306	705.8926944	0.197662605
4	2621.6	3464.473216	842.8732156	0.321510992
5	3258.4	2990.501984	267.8980165	0.082217658
6	4014.4	3671.945602	342.4543978	0.085306496
7	828	2449.070001	1621.070001	1.957814011
8	2791.2	3198.600436	407.4004361	0.145958884
9	2595.2	2784.577265	189.3772655	0.072972128
10	3332	2799.337182	532.6628177	0.15986279
11	4426.4	3620.185184	806.2148159	0.182137813
12	764	2480.110223	1716.110223	2.246217569
13	3001.6	2923.893446	77.70655363	0.025888377
14	1364	2899.424467	1535.424467	1.125677762
15	4681.6	2659.092454	2022.507546	0.432012036
16	4365.6	4014.633145	350.9668552	0.080393727
17	1477.6	2507.947744	1030.347744	0.697311684
18	3612.8	3308.057286	304.742714	0.08435084
19	1976	2572.217484	596.2174843	0.301729496
20	3333.6	3459.298813	125.6988131	0.037706627
21	3695.2	3894.871271	199.6712707	0.054035308
22	1202.4	2565.493657	1363.093657	1.133644093
23	2339.2	3208.396194	869.1961942	0.3715784
24	2385.6	2445.869816	60.26981611	0.025264007
25	2948	2801.951942	146.0480578	0.049541404
26	3248	3252.576196	4.576196283	0.001408927
27	782.4	2409.519486	1627.119486	2.079651695
28	2315.2	2564.99703	249.7970295	0.107894363
29	2260.8	2549.086827	288.286827	0.127515405
30	1560.8	2578.334442	1017.534442	0.651931344
31	2389.5	2816.880271	2034.480271	2.600307095
		Average Error Value	733.0998271	0.495723416

In table, Error(S) A represents absolute error and Error(S)R represents relative error. The comparison between real value and prediction value for product amount value is shown in Figure 6.7 a, prediction error is also shown in Figure 6.7 b and c here.



(a)



(b) (c)

Figure 6.7: The prediction results with single NN model

(a) comparison between real value and prediction value for product amount
(b) absolute error (c) relative error

Some conclusions can be drawn through the analysis of results as follows:

1. A basic prediction can be obtained with above single ANNs model, but accuracy is not enough good.
2. Data change tendency can be predicted for global view but abrupt change tendency and data value cannot be precisely predicted by the single NNs model. This information of abrupt change and its tendency are often important in a supply chain process.
3. From the model and result, it is difficult to reveal useful inner laws of supply chain process for process diagnosis and prediction for state abrupt change.

Some relevant research about forecasting time series with intelligence technology have been done, such as [N, Khiripet, 2001] [S , Leonard; M, Francesco, 1997].

6.2.3 Why the Integrated Intelligent Model Can Solve the Problems

Based on above basic analysis of the supply chain process and data distribution, it is clear to see the supply chain process has similar common features and precondition that is extracted and abstracted as a basic feature for building integrated intelligent model in chapter 2. These similar common characteristics are listed bellows:

- All data collected in the supply chain process is dependency with time, they are typical time series data. The process can be regarded as discrete time system.
- The supply chain process is complex and multivariable input-output process. It is also nonlinear process and difficult to express and extract the relationship between input and output variables in physical insight.
- The supply chain process is relatively stable, but it also dynamic and time-vary process on the scale.
- The input data and output data can be described using membership function of fuzzy set, for example, they can be described as normal input data, abnormal input data, normal and abnormal output behaviors, etc. The supply chain process can be measured and controlled based on the kinds of variables.
- The control results or prediction output for the supply chain process allows some error and deviation.

- The business goal of supply chain process is to predict the amount of goods supply in future, analyze the cause to affect goods supply amount and price, in essence, this problem belongs to process diagnosis and prediction problem.

Hence, the business process basically satisfies the premise and precondition of integrated intelligent model developed in this thesis.

In the supply chain process, when input data fluctuates beyond the normal range, it can be regarded as aberrant data (or symptoms). When output data of process is beyond the normal state, it can be regarded as aberrant behavior (or a fault). Hence, the prediction of product supply can be implemented with different prediction models under corresponding state, respectively. It means different prediction models are needed when supply chain process is in aberrant state or normal state. The underlying rules to trigger process fluctuation can be regarded as the causes, which result in process state change from one to another. Hence, the supply chain problem in the project can be extracted as abnormal state detection (*state detection*) and prediction (*diagnosis identification of the quantitative analysis*) of dynamic process behaviors.

According to the characteristics of the supply chain process and business goals, the integrated intelligent model can be involved in order to reach following aims in details.

1. An aberrant state (*abnormal state*) in the supply chain process can be detected. This is very useful to identify business risk.
2. Good prediction ability and change tendency of a product supply amount can be reached when the supply chain process is under different state.
3. Some underlying rules that trigger the state change of supply chain process can be extracted and analyzed for business risk control, business behaviors analysis and making decision.

Hence, from technical point of view, the integrated intelligent model is applied into the business problems due to the below principles,

- The *function of fault detection* in integrated intelligent model is employed to find or detect state of supply chain process, especially aberrant state.
- The *function of prediction* in integrated intelligent model is involved into the supply chain process to analyze quantitatively output prediction.

The application of integrated intelligent model for this supply chain process can be explained with following technical term:

Applying integrated intelligent model detects the process states, which is aberrant change in the supply chain process, and provides quantitative prediction based on the certain state.

6.3 Implementation of Integrated Intelligent Models for This Process

6.3.1 Intelligent Model Architecture of Supply Chain Process

The integrated intelligent model depicted in Figure 3.14 in chapter 3 can reach the main business goals in the supply chain process. At first, the different state in supply chain process can be defined and identified by learning from historical sample data. After the stages, the different states in this supply chain process can be roughly distinguished and then these relevant data corresponding to these states can also be classified, respectively. These data and process states will provide a key data source to train sub NN models, analyze underlying reason for process state change.

In training process, main task is modeling the supply chain process, it includes:

- *Definition of process state*: Fuzzy sets and membership functions can be generated to define different aberrant data for input variables and process states for important output variables respectively.
- *Training process state classifier:* A process state classifier based on fuzzy neural network is trained with it historical data in different states and fuzzy sets.
- *Training ANNs sub units for different process states:* Different sub units are trained with different data corresponding to different process states so as to roughly describe process characteristics in different process states.

In work process, when a set of new input variables is coming, the model works for:

- *Judge process state (process state detection):* the output of state classifier (as well as the residual generator) indicates which process states the real process will be or if there occurs abnormal state or fault.
- *Prediction.* Based on the different process state, the model predicts the process output according to fuzzy information of process state and fuzzy inference model.
- *Adaptive or optimal model*: Important model parameters are adaptively adjusted based on adaptive model architectures or are optimized by optimization schemes. A key parameter of model is designed as a threshold for residual signal or for definition of different process state.
- *Mining the process rules and discovering knowledge*. Based on different process state and corresponding input variables, a decision tree rule is applied to build the classification rules between crisp input data and process states; association rule is employed to discover frequent pattern among data states of input variables and process states.

The above functions and workflow concerning real the supply chain process are implemented step by step as following section in detail.

6.3.2 Feature Vector Selection

According to the analysis of supply chain process before, the time series model of the case is typical seasonality and the data in each week has similar characteristics. They appear some similarity and periodic features. More knowledge about time series, please refer to [Peter J. R, Richard A. D, 2002] [Mark L, Yaron K, and Abranham K, 2001]. Hence, it is nature to use the previous samples before 5 days as the input data to predict the value of next day. At the same time, the supply chain process is strong conjunctive to season, which is an important factor to affect consumer market in supper market. Therefore, the season factor should be involved into the model. In the case, the season influence of product supply can be represented by the product amount of every week or every month in supply market. The product amount data of every week is involved in the model as a key factor to express season factors. The basic feature input vector is listed in Table 6.4.

Table 6.4: The feature vector for input and output

Input feature data or vector	Output
• Data on Last 5th day • Data on Last 4th day • Data on Last 3th day • Data on Last 2th day • Data on Last 1th day • Total amount of a week • Total amount of a month	• Data of next day (Prediction value)

According NARX dynamic system model in chapter 3, the system output is determined by both historical behaviors (historical output) and system output affected by system input. In this case, only system historical behaviors are considered because the main aim is prediction with historical data, namely, the system output is only considered to relate to its historical behavior (output in past). Hence, the model output can be expressed as:

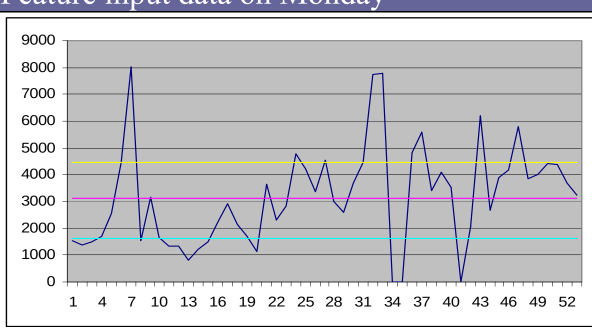
$$\hat{y}(k+1) = f\{y(k),\ldots,\ y(k-n)\} \tag{6-1}$$

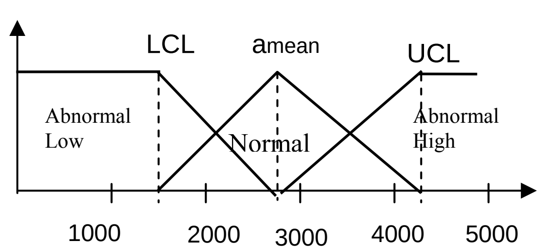
6.3.3 The Definition of Process States

Based on the implementation process of integrated intelligent model, the first step is to define all crisp variables of inputs and outputs as fuzzy variables. The definition will result in a series of fuzzy sets and membership functions for input and output variables.

The all input variables for the supply chain process are listed below and the fuzzfication of input data and output data (behavior) based on different process states are shown bellow Figure 6.8. In the figure, a^{mean} represents mean value during T date cycle, UCL denotes “UP Control Line” and LCL expresses “Low Control Line”. In this case, the membership function is design as triangle shape due to simplification of problem.

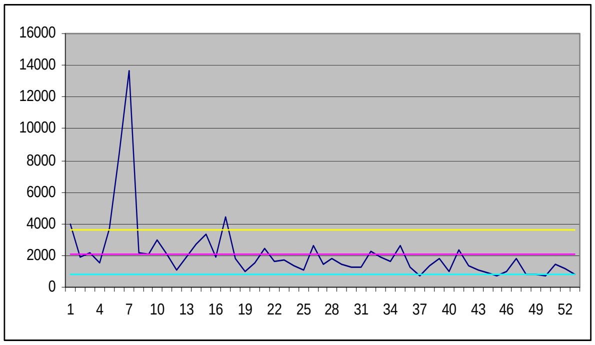
Feature input data on Monday

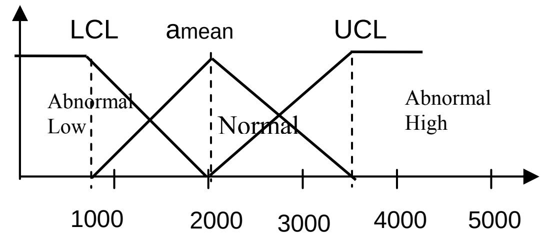




a^{mean} =2681.381, UCL =1568.638, LCL =4269.931

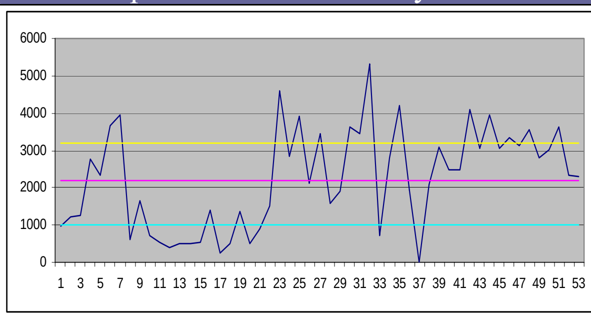
Feature input data on Tuesday

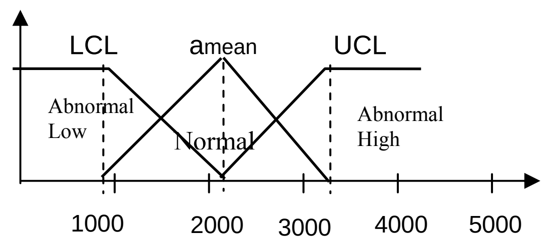




a^{mean} =2077.067 UCL =3645.12 LCL =807.4

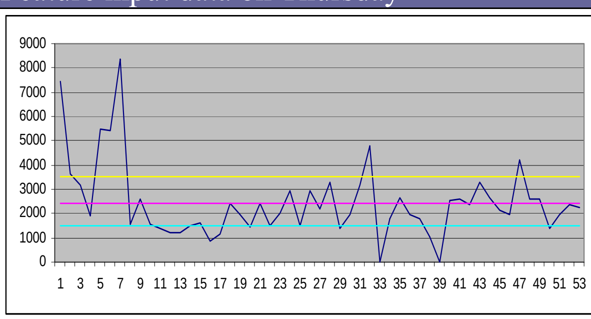
Feature input data on Wednesday

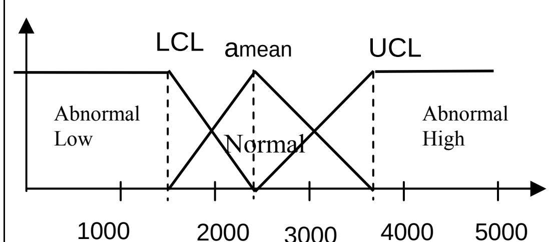




a^{mean} = 2201.185,UCL = 3212.469, LCL =988.553

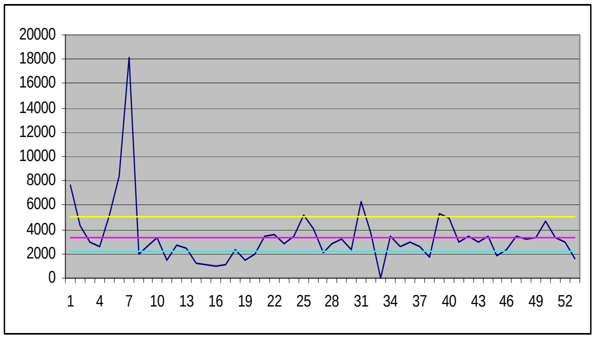
Feature input data on Thursday

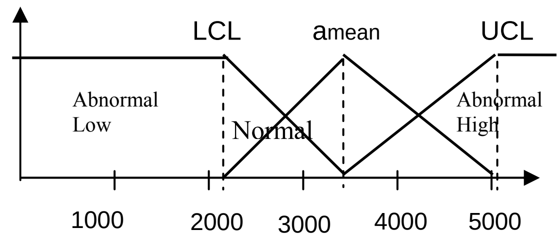




a^{mean} = 2409.593, UCL = 3536.645, LCL =1524.225

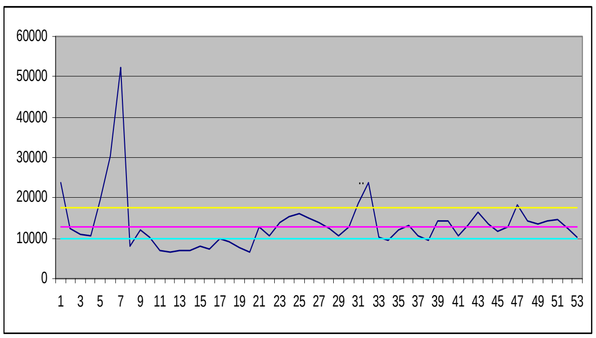
Feature input data on Friday

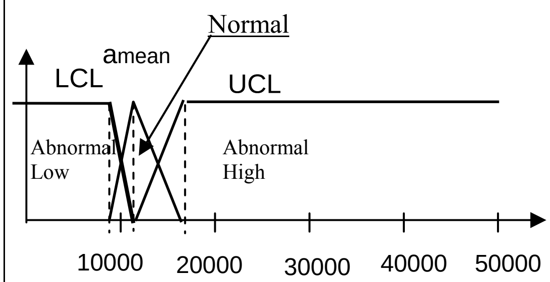




a^{mean} = 3344.03,UCL = 5029.145, LCL =2204.87

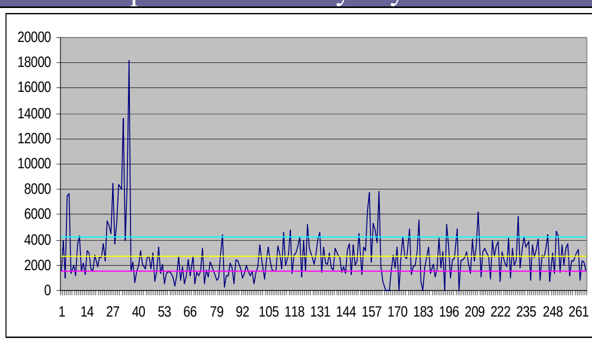
Feature input data on every week

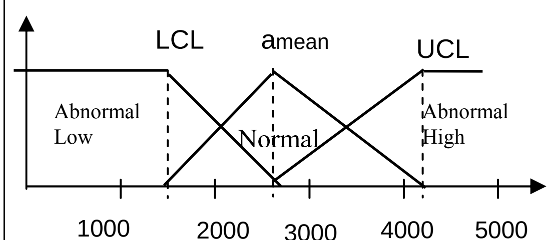




a^{mean} = 12956.37, UCL = 17493.35, LCL =9695.421

Feature input data on every day





a^{mean} = 2681.381, UCL = 4269.931, LCL =1568.638

Figure 6.8: The fuzzfication process and their membership functions

In this Figure, all variables are divided as three symbolic variables, namely, Abnormal Low, Normal and Abnormal High. To input variables, they corresponds three symptoms are "Abnormal Low Data", "Normal Data" and "Abnormal high Data". To output variable, it corresponds three symbolic variables as "Abnormal Low state", "Normal state" and "abnormal high state". The membership functions are generated base on certain data statistic methods.

6.3.4 Identification of Process States

After all fuzzy sets and membership functions for all input-output variables are defined, the relationship between states of input variables (Abnormal Low data, Normal data and Abnormal High data) and process states (Abnormal Low state, Normal state and Abnormal High state) can be established. Different process states can be roughly identified and classified through these membership functions, and then corresponding input data are organized into different process states.

Table 6.5: The different process states classification by membership functions

Date	Abnormal High	Date	Abnormal Low	Date	Normal
4	7444	1	1552	2	3984
5	7620.6	3	982	7	1938
10	4359	6	1381	9	3648.4
24	5461.4	8	1208	12	2201.2
25	5155.6	11	1492.2	14	3193
26	4465.6	13	1270	15	2971
27	8433.6	17	1506	16	1715
29	5427.2	36	1520.8	18	2776
30	8339.2	38	604	19	1876
31	8009.6	39	1556	20	2649
32	13614.4	48	725.6	21	2573.8
34	8376	49	1558.4	22	3695.6
35	18200	51	1329.6	23	2336.5
82	4413.6	53	536	28	3649.6
113	4612	54	1359.2	33	3936
116	4794.4	55	1529.6	37	2213.6
125	5220	56	1321.6	40	2001.6

••••

As we can see, the classification result implemented by membership functions is crisp and rough classification. The data sets for different process state are shown in Figure 6.9. (From 30th week 2000 to 36th week 2000).

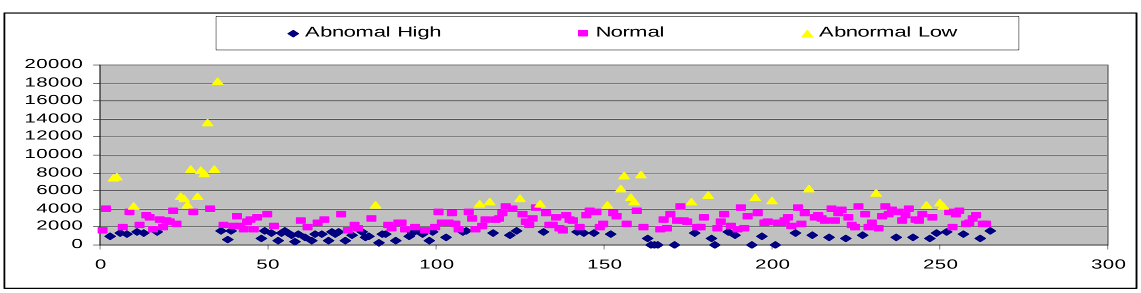


Figure 6.9: Data group in different states

(From 30th week 2000 to 36th week 2000)

The data in different process states is used to train different sub ANNs models, which is used to express process characteristics for corresponding process states.

6.3.5 Training Process State Classifier and Prediction Model

A state classifier is trained by different data in different process states with their corresponding fuzzy information. Three different ANN models are also trained by data in corresponding process state, respectively. Namely, 1) the characteristics of abnormal high process state is expressed by ANN1 model, which can be denoted as $f_1(x)$; 2) the characteristics of normal process state is expressed by ANN2 model, which is denoted as $f_2(x)$ and 3) the characteristics of abnormal low process state is expressed by ANN3 model, which can be denoted as $f_3(x)$. They are illustrated in Figure 6.10 .After the training phase, process characteristics in different process states can be expressed by its corresponding ANN subunits and approximate functions. The approximate expression could have some error, but it should better than using one ANN unit to express whole process characteristics.

6.3.6 The Business Rule Discovery for Making Decision

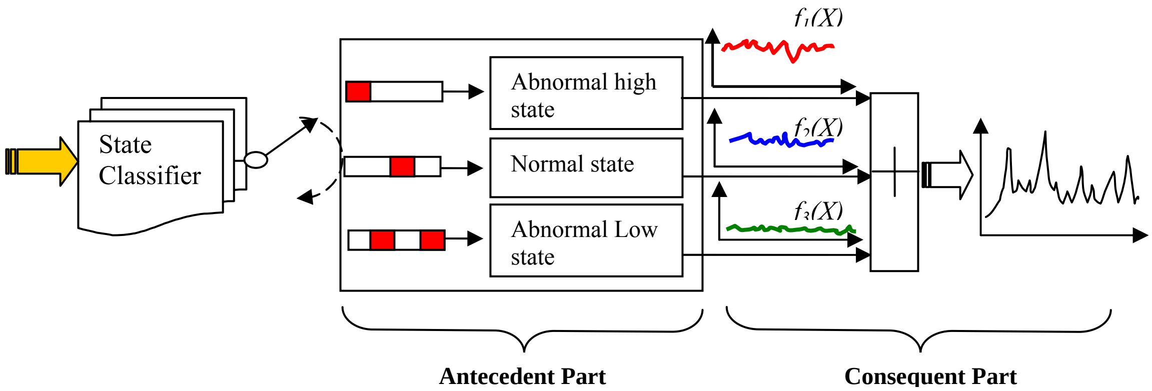
In the case, the whole process state has been divided into three different states. All data in different process states can be collected and organized into corresponding state, respectively. Analyzing these data in these states is more effective to reveal the inner law than analyzing all data in whole process.

Decision tree algorithm C5.0 is employed to extract the classification rule. Decision Tree rule establishes the relationship between crisp input variables and symbolic process state. These rules are extracted by Clementine ® 6.0 software.

Association Rule establishes the frequency pattern relationship between input data state (for example, input symptoms) and output process state (for example, fault state). It is very useful to analyze whole process behaviors and characteristics with these rules.

6.3.7 Prediction in Supply Chain Process

In integrated intelligent model, the data in different process states has been distinguished and stored respectively, and different ANN models and their approximate functions for characteristics of different process state have been trained. The whole prediction output of model can be obtained by sum of prediction values calculated by approximate functions of different process states with certain fuzzy operation. In this model, the final prediction output is determined by Fuzzy TS inference model. The basic principle is illustrated in Figure 6.10.



If state is **Abnormal high**	Then	**Output is $f_1(X)$**
If state is **Normal**	Then	**Output is $f_2(X)$**
If state is **Abnormal Low**	Then	**Output is $f_3(X)$**

Figure 6.10: Prediction output based on Fuzzy TS model

In the figure, $f_1(X)$, $f_2(X)$ and $f_3(X)$ are the approximate functions corresponding to three process states. They are determined by system identification method based NN model and used to express the process characteristics in different process states, respectively. The final prediction output is calculated with weight average based on Fuzzy TS model below.

$$Y_{output} = \frac{\sum_{i=1}^{3} w_i f_i(X)}{\sum_{i=1}^{3} w_i} = \sum_{i=1}^{3} w_i f_i(X) = w_1 f_1(X) + w_2 f_2(X) + w_3 f_3(X) \qquad (6\text{-}2)$$

where, w_i is fuzzy degree corresponding to different process states. It is gotten by state classifier. Here, $\sum_{i}^{3} w_i = 1$.

According to figure 6.10, the first step is to classify process states using state classifier.

Classify process state based on state classifier

State classifier is employed to identify or classify process state when new input feature vector is coming. The classification result is list in Table 6.6 (Input data from 30th week 2000 to 36th week 2000).

Table 6.6: Classification result of process state using state classifier

Mon	Tues	Wed	Thru	Friday	Week data	Real data	Fuzzy degree of low abnormal State (w_1)	Fuzzy degree of normal State (w_2)	Fuzzy degree of high abnormal State (w_3)
5811.2	1831.2	3108.8	4224.8	3429.6	18405.6	3845.6	0	0.58	0.42
1831.2	3108.8	4224.8	3429.6	3845.6	18405.6	853.6	0	0.99	0.01
3108.8	4224.8	3429.6	3845.6	853.6	18405.6	3571.2	0.12	0.88	0
4224.8	3429.6	3845.6	853.6	3571.2	18405.6	2621.6	0	0.69	0.31
3429.6	3845.6	853.6	3571.2	2621.6	18405.6	3258.4	0	0.992	0.008
3845.6	853.6	3571.2	2621.6	3258.4	14150.4	4014.4	0	0.78	0.22
853.6	3571.2	2621.6	3258.4	4014.4	14150.4	828	0.59	0.41	0
3571.2	2621.6	3258.4	4014.4	828	14150.4	2791.2	0.09	0.91	0
2621.6	3258.4	4014.4	828	2791.2	14150.4	2595.2	0.12	0.88	0
3258.4	4014.4	828	2791.2	2595.2	14150.4	3332	0.05	0.95	0
4014.4	828	2791.2	2595.2	3332	13560.8	4426.4	0	0.83	0.17
828	2791.2	2595.2	3332	4426.4	13560.8	764	0.57	0.43	0
2791.2	2595.2	3332	4426.4	764	13560.8	3001.6	0.12	0.88	0
2595.2	3332	4426.4	764	3001.6	13560.8	1364	0.08	0.92	0
3332	4426.4	764	3001.6	1364	13560.8	4681.6	0	0.12	0.88
4426.4	764	3001.6	1364	4681.6	14237.6	4365.6	0	0.66	0.34
764	3001.6	1364	4681.6	4365.6	14237.6	1477.6	0.63	0.37	0
3001.6	1364	4681.6	4365.6	1477.6	14237.6	3612.8	0.01	0.99	0
1364	4681.6	4365.6	1477.6	3612.8	14237.6	1976	0.24	0.76	0
4681.6	4365.6	1477.6	3612.8	1976	14237.6	3333.6	0	0.88	0.12
4365.6	1477.6	3612.8	1976	3333.6	14765.6	3695.2	0	0.77	0.23
1477.6	3612.8	1976	3333.6	3695.2	14765.6	1202.4	0.54	0.46	0
3612.8	1976	3333.6	3695.2	1202.4	14765.6	2339.2	0.05	0.95	0
1976	3333.6	3695.2	1202.4	2339.2	14765.6	2385.6	0.26	0.74	0
3333.6	3695.2	1202.4	2339.2	2385.6	14765.6	2948	0.05	0.95	0
3695.2	1202.4	2339.2	2385.6	2948	12570.4	3248	0	0.94	0.06
1202.4	2339.2	2385.6	2948	3248	12570.4	782.4	0.59	0.41	0
2339.2	2385.6	2948	3248	782.4	12570.4	2315.2	0.21	0.79	0
2385.6	2948	3248	782.4	2315.2	12570.4	2260.8	0.26	0.74	0
2948	3248	782.4	2315.2	2260.8	12570.4	1560.8	0.15	0.85	0
3248	782.4	2315.2	2260.8	1560.8	10167	2389.5	0.13	0.87	0

The fuzzy information for process state classification is given in above table. This fuzzy information not only indicates the degree of process state (in fault problem it indicate the fault degree), but also provides calculation compensation for final prediction result. According to the model structure, the final prediction output also depends on the ϑ value, which is used to indicate the threshold for process states change. The different ϑ value can result in different model prediction output. Hence, ϑ value can be used to adjust the process state as well as model prediction output.

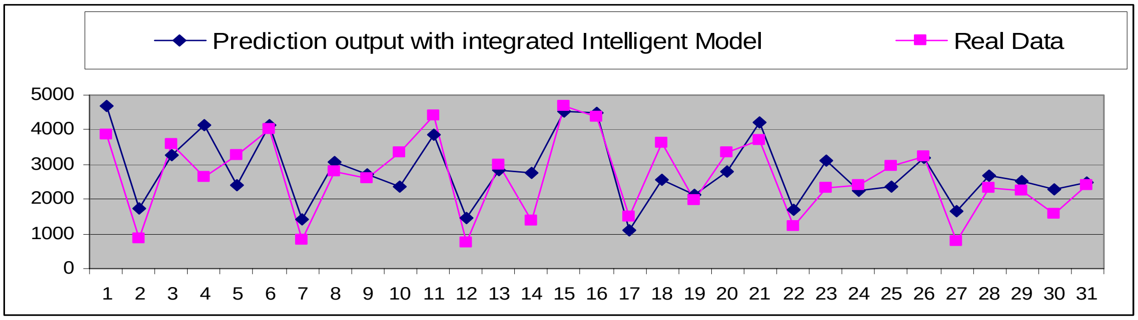
The prediction result based on Fuzzy TS dynamic model

Based on model structure, the Fuzzy TS dynamic model output is generated in Table 6.7. Its prediction curve and error curve are shown in Figure 6.11 below. All test data are from 30th week 2000 to 36th week 2000. Please note, prediction data with Fuzzy TS model in the table is only output data from Fuzzy TS model, they are not the final output from integrated intelligent model. The final prediction output of integrated intelligent model is generated by combination between Fuzzy TS model output and model optimization scheme.

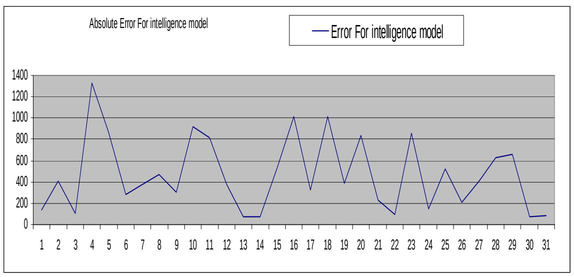
Table 6.7: The real data, prediction result and prediction error

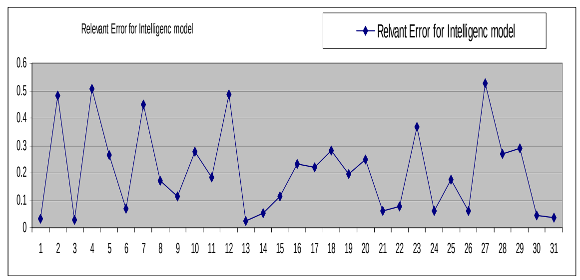
Date	Real Data	Prediction data with Fuzzy TS model	Error(I)A	Error(I)R
1	3845.6	4688.149888	132.9089082	0.03456129
2	853.6	1743.186708	409.5550217	0.47979735
3	3571.2	3258.603508	102.020308	0.02856751
4	2621.6	4151.78606	1326.110762	0.50584024
5	3258.4	2412.544745	863.772069	0.26509086
6	4014.4	4123.40763	285.1417825	0.07102974
7	828	1430.52078	371.4547143	0.4486168
8	2791.2	3082.812728	473.9190804	0.16979044
9	2595.2	2718.519115	300.9346937	0.11595819
10	3332	2359.757945	922.3344104	0.27681105
11	4426.4	3850.627118	812.25	0.18350127
12	764	1437.333637	371.9056879	0.48678755
13	3001.6	2846.138176	70.6331349	0.02353183
14	1364	2741.364935	72.22605588	0.05295165
15	4681.6	4512.066688	528.43	0.1128738
16	4365.6	4475.830551	1016.16	0.23276526
17	1477.6	1108.537687	325.7732022	0.22047455
18	3612.8	2578.660774	1014.254575	0.2807392
19	1976	2140.185426	388.3093532	0.19651283
20	3333.6	2778.532026	830.9894581	0.2492769
21	3695.2	4218.975393	226.4607215	0.06128511
22	1202.4	1676.899977	92.8695372	0.07723681
23	2339.2	3101.747802	858.3807949	0.36695485
24	2385.6	2238.541252	145.2794887	0.06089851
25	2948	2379.443255	522.4935717	0.17723663
26	3248	3173.765029	203.816995	0.06275154
27	782.4	1663.493775	411.9502465	0.52652128
28	2315.2	2675.339683	626.9251342	0.2707866
29	2260.8	2521.648687	654.6806093	0.28957918

30	1560.8	2278.766595	70.95821019	0.04546272
31	2389.5	2472.707698	335.2935829	0.42854497
	Average Error value		534.1148059	0.30136672



(a)





(b) (c)

Figure 6.11: The prediction results with Fuzzy TS model

(a) Comparison between real value and prediction value for product amount
(b) Absolute error. (c) Relative error

6.4 Model Optimization and Adaptability

As the model architecture, the final model prediction output is generated by combination between Fuzzy TS model output and optimization scheme for parameter ϑ. It is illustrated in Figure 4.3. Hence, based on different process characteristics, there are two different schemes for model optimization, one is to search optimal ϑ as so to get optimal model for process state identification and diagnosis, and another is optimal model for model prediction without any consideration of ϑ. They are introduced and implemented below.

6.4.1 Prediction Based on Process State Identification

This optimization scheme is the results that one optimized parameter ϑ is obtained for all realizations of the operation environment, assumed to be wide-sense stationary. When process is a stationary process, the model optimization based on process state identification can be used in the integrated intelligent model to produce final prediction output. In the problem of model optimization, the optimal parameter ϑ can be gotten below.

Model optimization with GA

1, only one parameter ϑ is regarded as threshold for definition of process states

Under the circumstance, only one model parameter ϑ is used to define three different process states. When fuzzy degree w_{2n} belonging to normal state of process state classification is bigger than a threshold value ϑ, then the process state is normal state in the moment. The model prediction output is calculated only by process output from process normal state. Otherwise, the model prediction output is calculated by fuzzy average between process outputs from normal process state and process outputs from abnormal process state. It can be illustrated in Figure 6.12 as well as in Chapter 3.

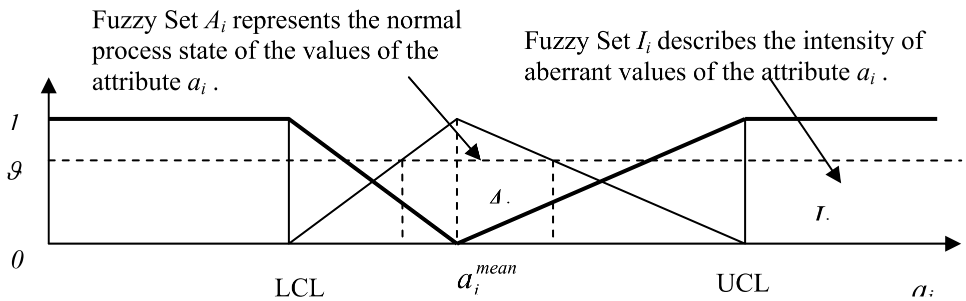


Figure 6.12: The one optimized parameter ϑ for process state classification

Hence, the parameter ϑ , which is regarded as a threshold value for process state classification, has strong influence on the model prediction output. It can be optimized based on error performance surface which is defined as the difference between process real response and actual model output. *The minimum point of this surface represents the optimal model solution.*

Here, the optimal model solution is obtained with Genetic Algorithm (GA) in software Genehunter®. The option concerning GA is set up and best fitness function curve is also shown in Figure 6.13. The final optimal model solution is given in Table 6.8.

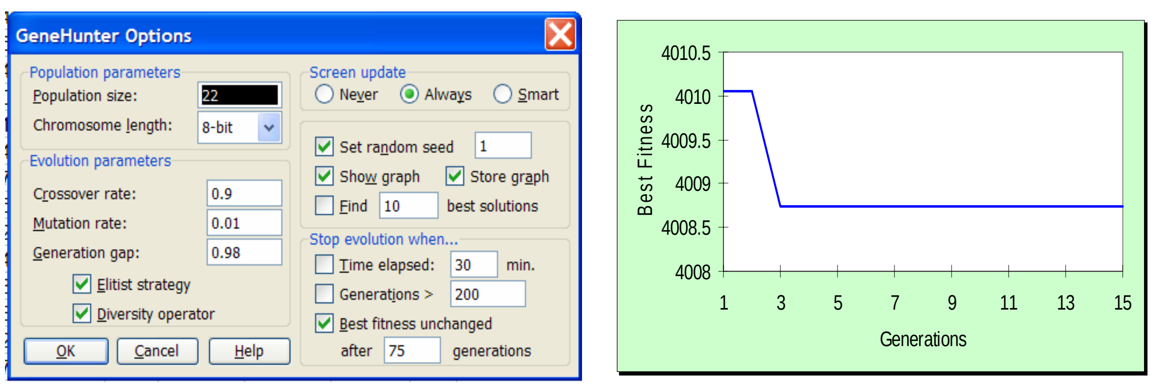


Figure 6.13: GA parameter option in GeneHunter® software

Table 6.8: The optimal model solution with respect to ϑ , T=31 days

Parameter ϑ	Fitness function
9.5	4008.744317

2, two parameters ϑ_1 and ϑ_2 are regarded as thresholds for definition of process states.

Under the circumstance, ϑ_1 and ϑ_2 are used to define three different process states. It means two parameters ϑ_1 and ϑ_2 are defined as threshold value for definition of different process states. When fuzzy degree w_{1l} belonging to abnormal low state of process state classification is less than a threshold value ϑ_1 and fuzzy degree w_{3h} belonging to abnormal high state of process state classification is less than a threshold value ϑ_2, then the process state is normal state in the moment. The model prediction output is calculated only by process output from process normal state. Otherwise, the model prediction output is calculated by fuzzy average between process outputs from normal process state and process outputs from abnormal process state. It can be illustrated in Figure 6.14.

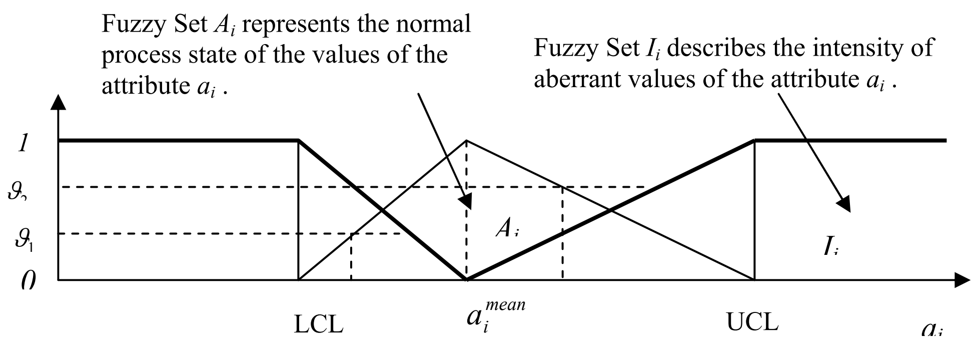


Figure 6.14: The two optimized parameters for process state classification

The optimal model solution is obtained in Table 6.9.

Table 6.9: The optimal model solution with respect to ϑ_1 and ϑ_2, T=31 days

Parameter ϑ_1	Parameter ϑ_2	Fitness function
0.97	0.96	4008.744317

The final prediction output from this optimal model is given in Table 6.10.

Table 6.10: The final prediction output from this optimal model (T=31 days)

Fuzzy w1l	Fuzzy w2n	Fuzzy w3h	Fuzzy TS output	Real value	Prediction output based on one parameter	Prediction Output based on two parameters
0.00	0.58	0.42	4688.149888	3845.6	4688.149902	4688.149902
0.00	0.99	0.01	1743.186708	853.6	1743.186768	1743.186768
0.12	0.88	0.00	3258.603508	3571.2	3258.603516	3258.603516
0.00	0.69	0.31	4151.78606	2621.6	4151.786133	4151.786133
0.00	0.99	0.01	2412.544745	3258.4	2412.544678	2412.544678
0.00	0.78	0.22	4123.40763	4014.4	4123.407715	4123.407715
0.59	0.41	0.00	1430.52078	828	1430.520752	1430.520752
0.09	0.91	0.00	3082.812728	2791.2	3082.812744	3082.812744
0.12	0.88	0.00	2718.519115	2595.2	2718.519043	2718.519043
0.05	0.95	0.00	2359.757945	3332	2359.757813	2359.757813
0.00	0.83	0.17	3850.627118	4426.4	3850.626953	3850.626953
0.57	0.43	0.00	1437.333637	764	1437.333618	1437.333618
0.12	0.88	0.00	2846.138176	3001.6	2846.138184	2846.138184

0.08	0.92	0.00	2741.364935	1364	2741.36499	2741.36499
0.00	0.12	0.88	4512.066688	4681.6	4512.066895	4512.066895
0.00	0.66	0.34	4475.830551	4365.6	4475.830566	4475.830566
0.63	0.37	0.00	1108.537687	1477.6	1108.53772	1108.53772
0.01	0.99	0.00	2578.660774	3612.8	2578.660889	2578.660889
0.24	0.76	0.00	2140.185426	1976	2140.185303	2140.185303
0.00	0.88	0.12	2778.532026	3333.6	2778.531982	2778.531982
0.00	0.77	0.23	4218.975393	3695.2	4218.975098	4218.975098
0.54	0.46	0.00	1676.899977	1202.4	1676.900024	1676.900024
0.05	0.95	0.00	3101.747802	2339.2	3101.747803	3101.747803
0.26	0.74	0.00	2238.541252	2385.6	2238.54126	2238.54126
0.05	0.95	0.00	2379.443255	2948	2379.443115	2379.443115
0.00	0.94	0.06	3173.765029	3248	3173.765137	3173.765137
0.59	0.41	0.00	1663.493775	782.4	1663.493774	1663.493774
0.21	0.79	0.00	2675.339683	2315.2	2675.339844	2675.339844
0.26	0.74	0.00	2521.648687	2260.8	2521.648682	2521.648682
0.15	0.85	0.00	2278.766595	1560.8	2278.766602	2278.766602
0.13	0.87	0.00	2472.707698	2389.5	2472.707764	2472.707764

We can see, the prediction output based on optimal model of one parameter ϑ_1 is same as the prediction output based on optimal model of two parameters ϑ_1 and ϑ_2 here. Because the model optimal solution when only one parameter ϑ_1 =0.95 are the same as the model optimal solution with two parameters ϑ_1 = 0.97 and ϑ_2 =0.96. At the same time, the prediction output from optimal model is also same as Fuzzy TS model output, this is because the fuzzy TS model is optimal model when ϑ_1 =0.95. It means the process states in the process are completely considered as abnormal state.

The optimal model solution can also be obtained with different optimization algorithms and tools, which is addressed in chapter 4. The optimization result of model parameter is given below. Summary, The optimal model solutions are given in Table 6.11 with respect to one parameter ϑ_1 and in Table 6.12 with respect to two parameters ϑ_1 and ϑ_2, respectively.

Table 6.11: The final prediction output from different optimization algorithm and Tools (Only one optimized parameter ϑ_1, T=31 days)

Optimization method /Tool	Genehunter/ GA	Matlab Optimization Toolbox	NOSYS/ Without Stochastic research	NOSYS/ With Stochastic research but EPST=0.1
Parameter ϑ_1	0.95	0.954733	1.00	0.9716
Object function	4008.744317	4008.744309	0.40167914D+04	0.40087443D+04

Table 6.12: The final prediction output from different optimization algorithm and Tools (Two optimized parameters ϑ_1, ϑ_2, T=31 days)

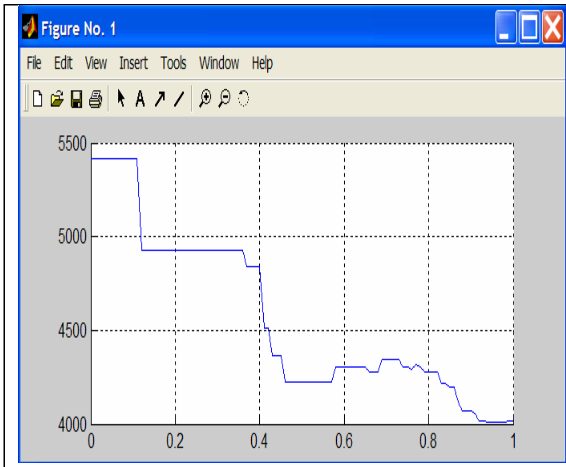
Optimization Tool/Method	Genehunter/ GA	Matlab/ Optimization Toolbox	NOSYS/ Without Stochastic research	NOSYS/ With Stochastic research but EPST=0.1
Parameter ϑ_1	0.97	0.954733	1	0.969

Parameter ϑ_2	0.96	0.982	0.95	0.9623
Object function	4008.744317	4008.744309	0.40167914D+04	0.40087443D+04

NOSYS® is a software tool for nonlinear optimization [Koch, W. H., 2002].

Genhunter® is a software tool based on genetic algorithms [GeneHunter, 2000].

The error performance curve or surface, here also objective function, is shown in Figure 6.15 with respect to one optimized parameter ϑ_1 and in Figure 6.16 with respect to two parameters ϑ_1 and ϑ_2, respectively (T=31 days).



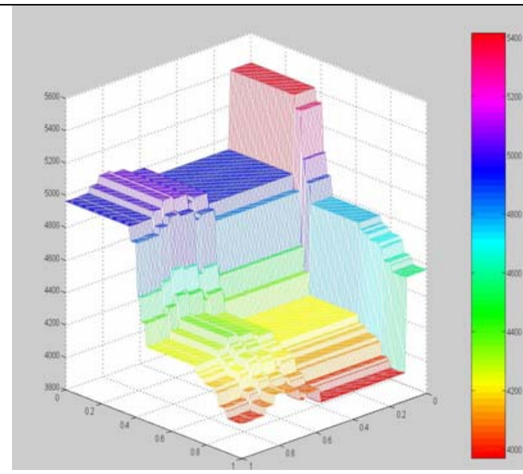
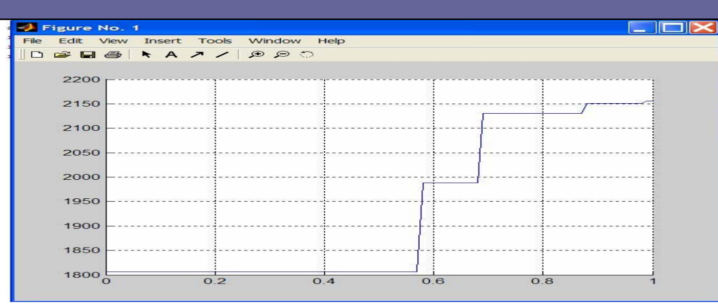


Figure 6.15: Error performance curve with respect to one parameter ϑ_1

Figure 6.16: Error performance curve with respect to two parameters ϑ_1 and ϑ_2

The optimal model parameter ϑ_1 (only consider one free parameter in the case) or ϑ_1 and ϑ_2 (two free parameters are considered for process state classification in the case) are obtained based on model optimization algorithms. All above optimal prediction results are calculated based time interval T=31 days, namely, from 30th week 2000 to 36th week 2000. It means the integrated intelligent model are optimized based on the model error produced during T=31 days. When different time interval T is given, the different model parameter values can be obtained, which is shown in Table 6.13 and Table 6.14.

Table 6.13: The optimal model parameter based on different time interval with respect to one parameter ϑ_1

Time interval	Optimal parameter	Error Curve
T=5	ϑ_1=0.58	

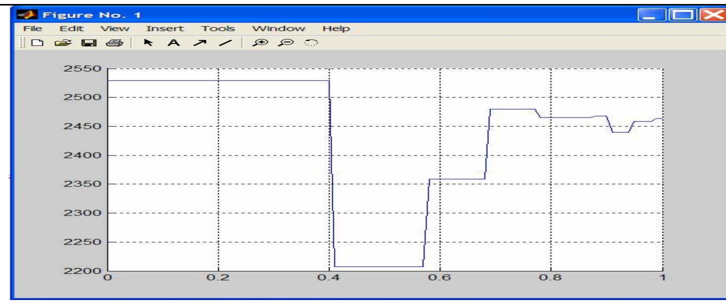
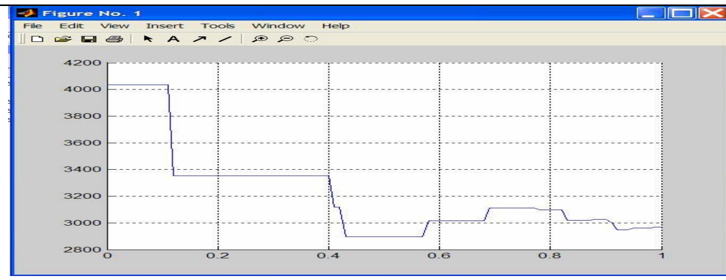
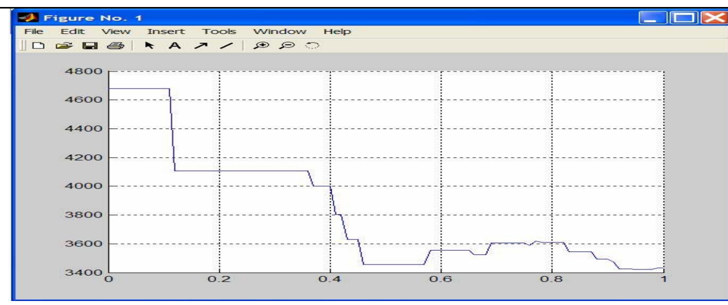
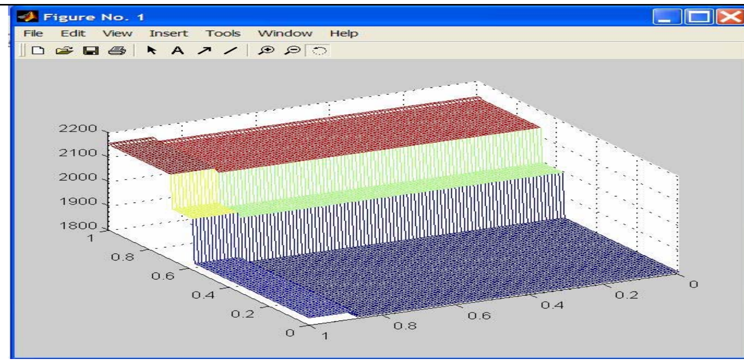
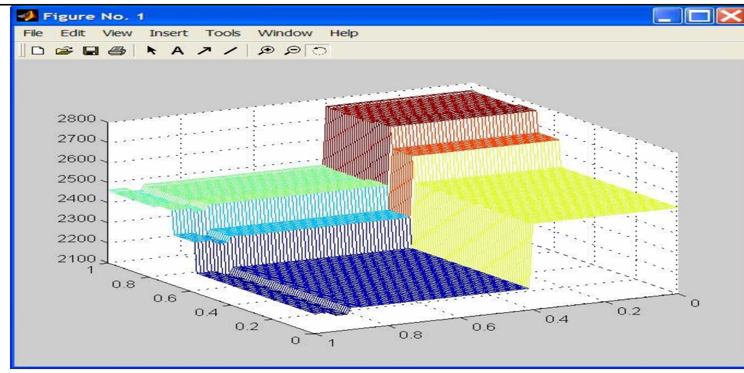
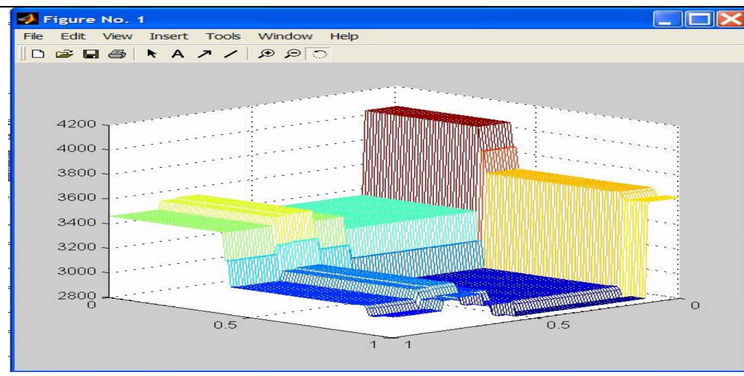
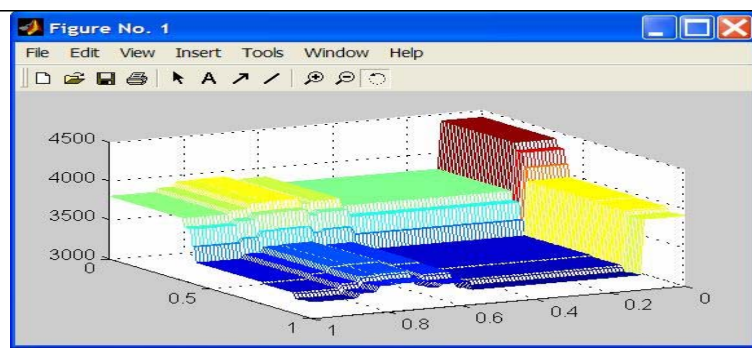
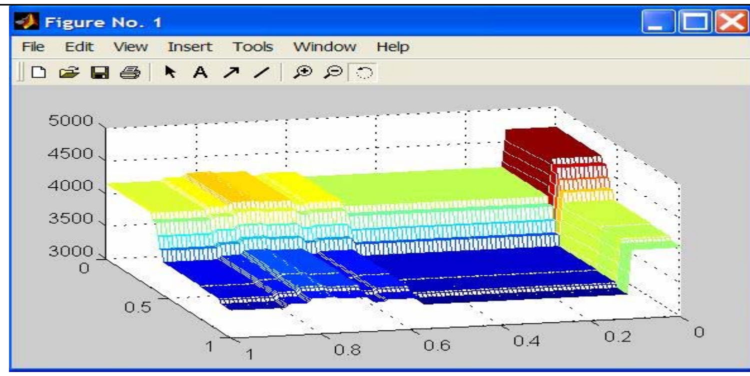
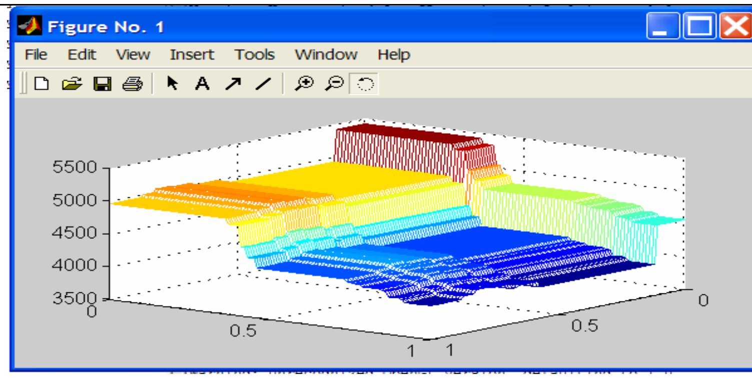
T=10	ϑ_1 =0.58	
T=15	ϑ_1 =0.58	
T=20	ϑ_1 =0.92	
T=25	ϑ_1 =0.95	

Table 6.14: The optimal model parameters based on different time interval
(with respect to two parameter ϑ_1 and ϑ_2)

Time interval	Optimal parameter	Error Curve

<table>
<tr><td>T=5</td><td>$\vartheta_1=0.58$
$\vartheta_2=0.87$</td><td>
</td></tr>
<tr><td>T=10</td><td>$\vartheta_1=0.58$
$\vartheta_2=0.90$</td><td>
</td></tr>
<tr><td>T=15</td><td>$\vartheta_1=0.92$
$\vartheta_2=0.58$</td><td>
</td></tr>
<tr><td>T=20</td><td>$\vartheta_1=0.92$
$\vartheta_2=0.58$</td><td>
</td></tr>
</table>

T=25	ϑ_1=0.95 ϑ_2=0.58	
T=31	ϑ_1=0.97 ϑ_2=0.96	

Here, the model prediction output could be different if different time intervals are adopted. It indicates the supply chain process that is described with integrated intelligent model can be regarded as a complicated discrete time-varying process. The optimal model parameters can be determined by model optimization scheme or updated by model adaptive scheme.

The prediction value and their comparison are given when the optimization parameters ϑ_1 from 0.1 to 1 in Table 6.15 and Figure 6.17. It is illustrated the different prediction value of the integrated intelligent model and their prediction performance based on different optimization parameter value of the model.

Table 6.15: the prediction value when optimization parameters ϑ_1 from 0.1 to 1

Real Data	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
3845.6	3978.509	3978.509	3978.509	3978.509	3978.5089	4688.15	4688.1499	4688.149888	4688.149888	4688.149888
853.6	1707.605	1707.605	1707.605	1707.605	1707.6054	1707.605	1707.6054	1707.605445	1707.605445	1743.186708
3571.2	3673.22	3673.22	3673.22	3673.22	3673.2203	3673.22	3673.2203	3673.220308	3258.603508	3258.603508
2621.6	3947.711	3947.711	3947.711	3947.711	3947.7108	3947.711	4151.7861	4151.78606	4151.78606	4151.78606
3258.4	2394.628	2394.628	2394.628	2394.628	2394.6279	2394.628	2394.6279	2394.627931	2394.627931	2417.023949
4014.4	3729.258	3729.258	3729.258	3729.258	3729.2582	3729.258	3729.2582	4123.407631	4123.407631	4123.407631
828	2201.737	2201.737	2201.737	2201.737	1430.5208	1430.521	1430.5208	1430.52078	1430.52078	1430.52078
2791.2	3265.119	3265.119	3265.119	3265.119	3265.1191	3265.119	3265.1191	3265.11908	3265.11908	3082.812728
2595.2	2896.135	2896.135	2896.135	2896.135	2896.1347	2896.135	2896.1347	2896.134694	2718.519115	2718.519115
3332	2409.666	2409.666	2409.666	2409.666	2409.6656	2409.666	2409.6656	2409.66559	2409.66559	2359.757945
4426.4	3525.511	3525.511	3525.511	3525.511	3525.5113	3525.511	3525.5113	3525.51134	3850.627119	3850.627119
764	2093.17	2093.17	2093.17	2093.17	1437.3336	1437.334	1437.3336	1437.333637	1437.333637	1437.333637
3001.6	3072.233	3072.233	3072.233	3072.233	3072.2331	3072.233	3072.2331	3072.233135	2846.138176	2846.138176
1364	2857.873	2857.873	2857.873	2857.873	2857.8729	2857.873	2857.8729	2857.872858	2857.872858	2741.364935
4681.6	2431.087	4512.067	4512.067	4512.067	4512.0667	4512.067	4512.0667	4512.066688	4512.066688	4512.066688
4365.6	3880.056	3880.056	3880.056	3880.056	3880.0564	3880.056	4475.8306	4475.830551	4475.830551	4475.830551
1477.6	2479.002	2479.002	2479.002	1108.538	1108.5377	1108.538	1108.5377	1108.537687	1108.537687	1108.537687
3612.8	2598.545	2598.545	2598.545	2598.545	2598.5454	2598.545	2598.5454	2598.545425	2598.545425	2578.660774
1976	2364.309	2364.309	2364.309	2364.309	2364.3094	2364.309	2364.3094	2140.185425	2140.185425	2140.185425
3333.6	2502.611	2502.611	2502.611	2502.611	2502.6105	2502.611	2502.6105	2502.610542	2778.532026	2778.532026
3695.2	3921.661	3921.661	3921.661	3921.661	3921.6607	3921.661	3921.6607	4218.975393	4218.975393	4218.975393
1202.4	2401.664	2401.664	2401.664	2401.664	1676.9	1676.9	1676.9	1676.899977	1676.899977	1676.899977
2339.2	3197.581	3197.581	3197.581	3197.581	3197.5808	3197.581	3197.5808	3197.580795	3197.580795	3101.747802
2385.6	2530.879	2530.879	2530.879	2530.879	2530.8795	2530.879	2530.8795	2238.541252	2238.541252	2238.541252
2948	2425.506	2425.506	2425.506	2425.506	2425.5064	2425.506	2425.5064	2425.506428	2425.506428	2379.443255
3248	3044.183	3044.183	3044.183	3044.183	3044.183	3044.183	3044.183	3044.183005	3044.183005	3173.765029
782.4	2303.04	2303.04	2303.04	2303.04	1663.4938	1663.494	1663.4938	1663.493775	1663.493775	1663.493775
2315.2	2942.125	2942.125	2942.125	2942.125	2942.1251	2942.125	2942.1251	2675.339683	2675.339683	2675.339683
2260.8	2915.481	2915.481	2915.481	2915.481	2915.4806	2915.481	2915.4806	2521.648687	2521.648687	2521.648687
1560.8	2401.579	2401.579	2401.579	2401.579	2401.5787	2401.579	2401.5787	2401.578703	2278.766595	2278.766595
2389.5	2661.056	2661.056	2661.056	2661.056	2661.0562	2661.056	2661.0562	2661.056193	2472.707698	2472.707698

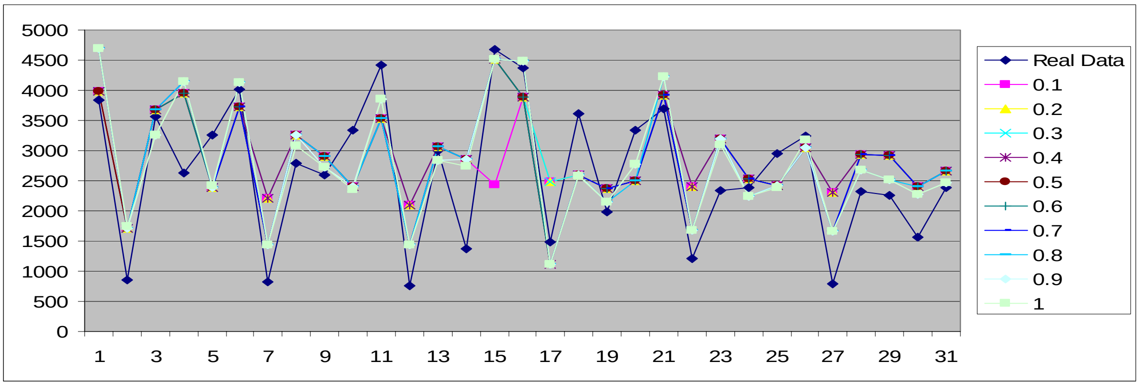


Figure 6.17: Prediction results when parameters ϑ_1 from 0.1 to 0.9

6.4.2 Prediction Based on Optimal Prediction Scheme

A model optimization scheme for optimal prediction output purpose is introduced in the chapter 4.2.2. The model optimization scheme is designed for the best prediction output without any consideration of process state factor. LS approach is used to reach the optimization goals.

According to the form of the least squares estimate from the compact matrix, the least squares algorithm that minimizes

$$J = \frac{1}{K}(Y - HX)^T (Y - HX)$$

can be represented in a more compact form by

$$\hat{H} = \left[X^T X\right]^{-1} X^T Y$$

where

$$\hat{H}\,\frac{\partial^2 J}{\partial H^2} = X^T X \text{ must be positive definite.}$$

In above formula, X is K× N input matrix collected from observations at the input of the model. K is sample number, N is input variable number.

$$X = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(K) \end{bmatrix} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_N(1) \\ x_1(2) & x_2(2) & & x_N(2) \\ \vdots & & \ddots & \vdots \\ x_1(K) & x_2(K) & \cdots & x_N(K) \end{bmatrix}$$

Y is K× 1 output vector from the observation

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(K) \end{bmatrix}$$

Based on LS algorithm and the model optimization scheme, the prediction output from the intelligent model is given in Table 6.16, and the absolute error and relative error based are also given in Table 6.17.

Table 6.16 , The prediction output based on optimal prediction model with LS (T=31)

Real Data	Prediction output based process state identification	Prediction output based LS
3845.6	4688.149902	4429.204684
853.6	1743.186768	1718.947456
3571.2	3258.603516	3195.152652
2621.6	4151.786133	3981.344906
3258.4	2412.544678	2385.722832
4014.4	4123.407715	3971.743433
828	1430.520752	876.715925
2791.2	3082.812744	2934.369528
2595.2	2718.519043	2514.734393
3332	2359.757813	2261.221823
4426.4	3850.626953	3727.150153
764	1437.333618	873.8137051
3001.6	2846.138184	2668.770115
1364	2741.36499	2596.198922
4681.6	4512.066895	4095.905903
4365.6	4475.830566	4259.576082
1477.6	1108.53772	901.146273
3612.8	2578.660889	2543.330334
1976	2140.185303	1766.307144
3333.6	2778.531982	2697.29386
3695.2	4218.975098	4067.555245
1202.4	1676.900024	1075.160896
2339.2	3101.747803	3001.485185
2385.6	2238.54126	1840.75491
2948	2379.443115	2275.961876
3248	3173.765137	3110.910142
782.4	1663.493774	912.0494066
2315.2	2675.339844	2287.523872
2260.8	2521.648682	2122.191079
1560.8	2278.766602	2011.184041
2389.5	2472.707764	2284.260221

Table 6.17 The Average absolute error and relative error (T=31)

Average absolute error	463.4951939
Average relative error	0.210928608

When different T is applied, the different prediction output and vector *H* can be obtained. Vector H is an important parameter to the model even model output. The different vectors *H* are listed in Table 6.18 when T is in different values.

Table 6.18 Different H when different T

T	H^T
T=5	[18.30811792914771 0.95658049712244 0.50946879161552]
T=10	[-0.29884532463873 1.08617629006863 0.40551379006156]
T=15	[-0.13025446848028 0.97837750132013 0.90253914201690]
T=20	[-0.10859960644081 1.03483335057790 0.88121746539757]
T=25	[0.16742604949242 1.01311142956099 0.88120757663277]
T=31	[-0.02994175188257 0.98870892470330 0.90217205308269]

The program with Matlab® for calculation H and model prediction output based on LS is given Appendix D.

6.5 Result Analysis and Performance Evaluation

In this section, some result analysis and evaluations for this supply chain process are presented here. It focuses on prediction ability for supply chain process and decision making ability for this case.

6.5.1 Product Supply Forecasting

The prediction result with single NN models and integrated intelligent model have been obtained, respectively. A comparison between them is given in Figure 6.18. The error of the prediction result is also given in Table 6.19.

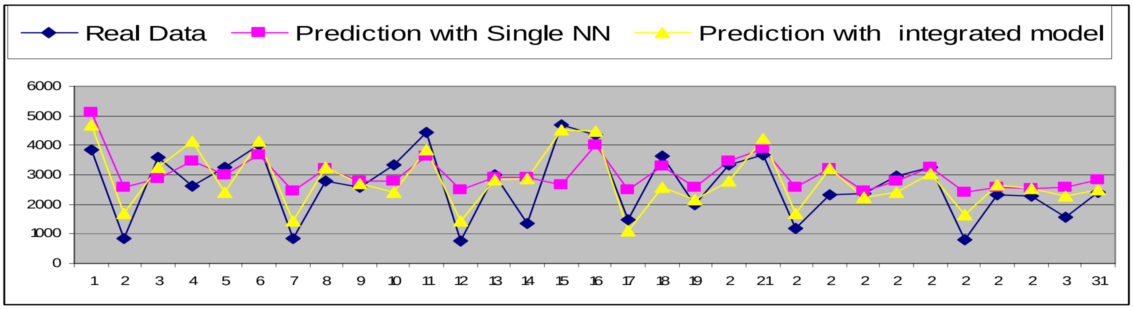


Figure 6.18: The Predication result comparison

(Among Prediction value of single NN model, integrated intelligent model and real value)

Table 6.19: The prediction error comparison between with single NN model and with integrated intelligent model

Average Absolute Error (AAE)		Average Relative Error (ARE)	
Integrated intelligent model	Single NN model	Integrated intelligent model	Single NN model
534.1148059	733.099	0.30136672	0.495

The error distribution curve can be shown as following Figure 6.19.

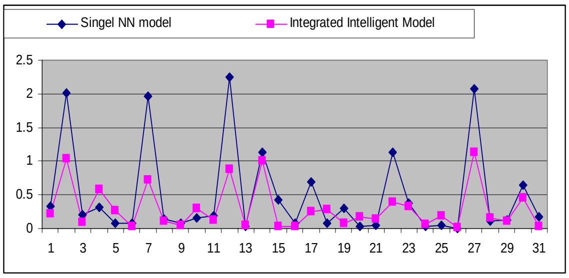


Figure 6.19: Error distribution curves comparison

(Single NN model and integrated intelligent model for supply prediction error)

(From 30th week 2000 to 36th week 2000)

According theses results, some conclusions can be drawn as follows:

1. The Integrated intelligent model for supply chain product forecasting not only provide better prediction ability than single NN model does, but also has a good ability to detect process state, correctly respond abrupt fluctuation state. This is very important for predicting aberrant process behaviors, evaluating process state and controlling process risk.
2. In integrated intelligent model, the process state classifier can provide classification ability for process state with fuzzy degree information. It is significant for process measurement, process state assessment and monitoring.

A comparison of prediction result between two different model optimization schemes is given in Figure 6.20 and prediction error is given in Table 6.20.

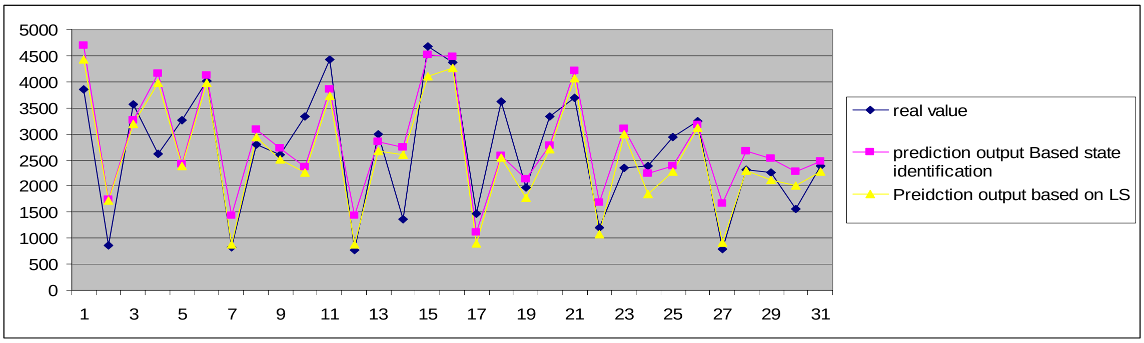


Figure 6.20 The comparison of prediction output

(Between two different model optimization schemes)

Table 6.20 Average absolute error and relative error between them

Average Absolute Error (AAE)		Average Relative Error (ARE)	
Optimal model based on state identification	Optimal model based on prediction	Optimal model based on state identification	Optimal model based on prediction
534.1148059	463.4951939	0.30136672	0.210928608

From Table 6.20, it is clear to see that the prediction output based on optimal prediction model can reach better prediction performance than optimal model based on state identification does. But it loses some important process state information, which is very important for process diagnosis and analysis.

Anyway, the integrated intelligent model either provides a supply prediction value for market demand in future or forecasts change tendency of supply chain process in future. The result proves that more precise prediction ability can be obtained with integrated intelligent model than with single NN model and traditional methods.

6.5.2 Business Process Analysis

In this case, two kinds of analysis for making decision are provided in integrated intelligent model: the Decision Tree Rule and Association Rules.

Decision Tree Rule Analysis

Here, Decision tree rule provides classification rules to build the relationship between input variable value and output labels. In the case, the output labels could be process state (fault state, normal state or abnormal state). The relationship is established between data in last 5 days and output states (Abnormal low state, normal state and abnormal high state). It is very valuable to analyze these rules and then make decision for the supply chain process. These decision tree rules are given in Table 6.21.

Table 6.21: Decision tree rule for process state classification

Rules for abnormal high process state	Rules for abnormal low process state	Rules for normal process state
Rule #1 for abnormal high: if Lastday5 >= 2488.1 and Lastday5 < 4325.6and Lastday1 >= 3609.8 and Lastday2 >= 1880.8 and Lastday4 >= 2226.25 then -> abnormal high Rule #2 for abnormal high: if Lastday5 >= 4325.6and Lastday1 >= 3609.8 then -> abnormal high	Rule #1 for abnormal low : if Lastday5 >= 129.6 and Lastday5 < 902.8 and Lastday2 < 4790.8 and Lastday3 < 2150.4 then -> abnormal low Rule #2 for abnormal low: if Lastday5 >= 1168.4 and Lastday5 < 1361.6 and Lastday2 < 4790.8 and Lastday3 < 2150.4 then -> abnormal low Rule #3 for abnormal low : if Lastday5 >= 129.6 and Lastday5 < 1239 and Lastday2 < 4790.8 and Lastday3 >= 2150.4 then -> abnormal low Rule #4 for abnormal low : if Lastday5 >= 1239 and Lastday5 < 1361.6 and Lastday2 < 4790.8 and Lastday3 >= 2150.4 then -> abnormal low Rule #5 for abnormal low : if Lastday5 >= 1361.6 and Lastday5 < 1655.2 and Lastday1 >= 3670.4 then -> abnormal low	Rule #1 for normal: if Lastday5 < 129.6 then -> normal Rule #2 for normal: if Lastday5 >= 902.8 and Lastday5 < 1168.4 and Lastday2 < 4790.8 and Lastday3 < 2150.4 then -> normal Rule #3 for normal: if Lastday5 >= 129.6 and Lastday5 < 1361.6 and Lastday2 >= 4790.8 then -> normal Rule #4 for normal: if Lastday5 >= 1361.6 and Lastday5 < 1655.2 and Lastday1 < 1854.8 then -> normal Rule #5 for normal: if Lastday5 >= 1361.6 and Lastday5 < 1655.2 and Lastday1 >= 1854.8 and Lastday1 < 3670.4 then -> normal Rule #6 for normal: if Lastday5 >= 1852.4 and Lastday1 < 744.8

	Rule #6 for abnormal low: if Lastday5 >= 1655.2 and Lastday5 < 1852.4 then -> abnormal low Rule #7 for abnormal low: if Lastday5 >= 1852.4 and Lastday1 < 744.8 and Lastday3 < 3239.6 then -> abnormal low Rule #8 for abnormal low: if Lastday5 >= 2488.1 and Lastday5 < 4325.6 and Lastday1 >= 3609.8 and Lastday2 >= 1880.8 and Lastday4 < 2226.25 then -> abnormal low	and Lastday3 >= 3239.6 then -> normal Rule #7 for normal: if Lastday5 >= 1852.4 and Lastday1 >= 744.8 and Lastday1 < 3609.8 then -> normal Rule #8 for normal: if Lastday5 >= 1852.4 and Lastday5 < 2488.1 and Lastday1 >= 3609.8 then -> normal Rule #9 for normal: if Lastday5 >= 2488.1 and Lastday5 < 4325.6 and Lastday1 >= 3609.8 and Lastday2 < 1880.8 then -> normal

There rules can be evaluated with further data in order to testify if these rules are correct , reasonable and valuable.

Association Rule Analysis

Association rule built the frequent pattern relationship between symbolic input variable (Abnormal Low data, Normal data and Abnormal high data) and symbolic process state output variable (Abnormal Low state, Normal state and Abnormal high state). These rules can be described according to different associate strength, for example, strong association, medium association and week association link. These all association rules is shown in Table 6.22 and association link is illustrated as following Figure 6.21 to Figure 6.23.

To Abnormal High State

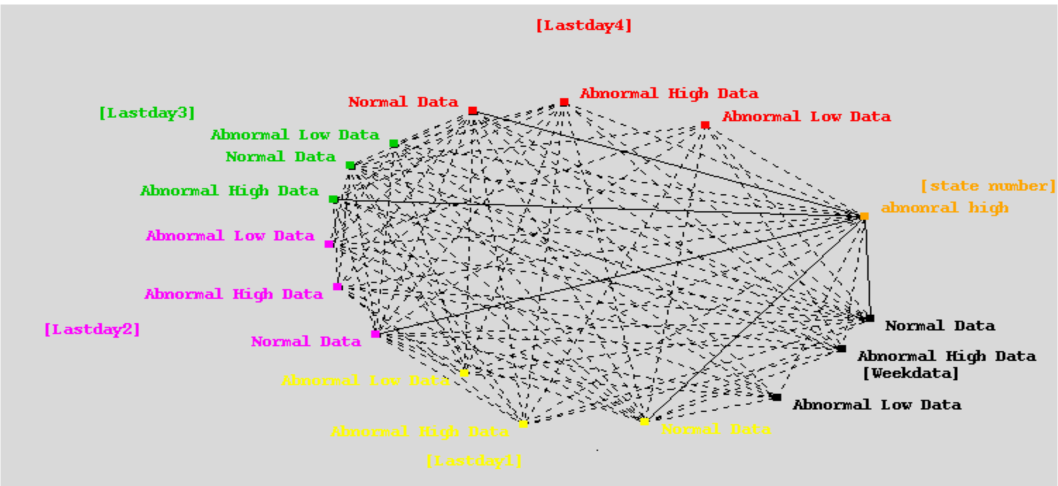


Figure 6.21: The association link for abnormal high state

To Abnormal Low State

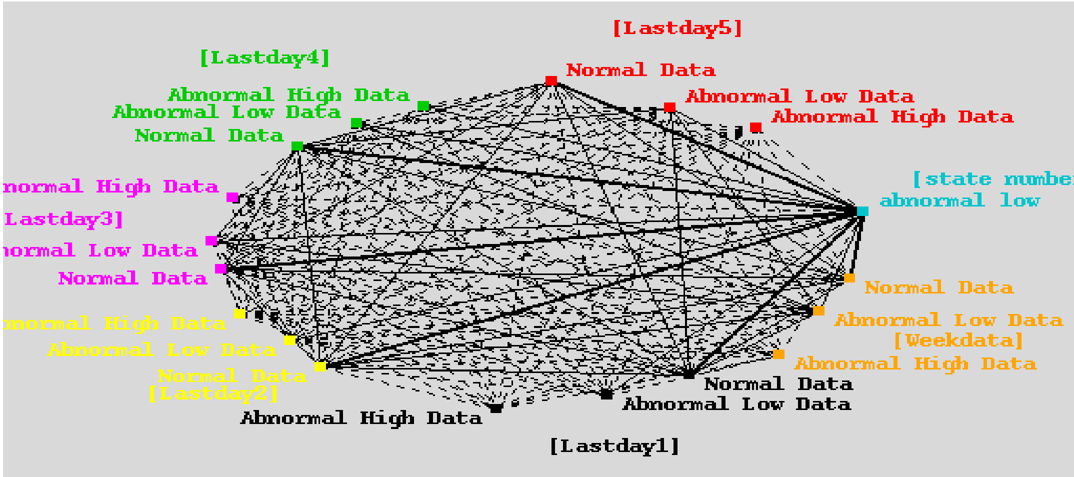


Figure 6.22: The association link for abnormal low state

To Normal State

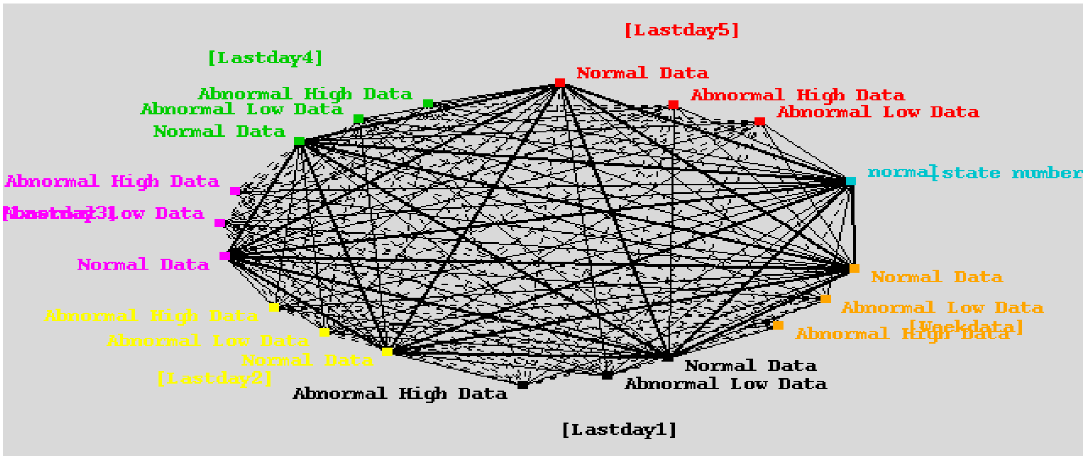


Figure 6.23: The association link for normal state

Association analysis algorithm (Aprior) can mine frequent pattern based on the multi-input variables, and then discover the relationship among these variables, these rules are listed in Figure 6.22. (by Clementine ® 6.0 software)

Table 6.22: Association rule for frequent pattern among process state and input data state

Rules for abnormal high:	**Rules for abnormal low :**	**Rules for normal:**
Rule #1 for abnormal high: If Lastday4==Normal Data then -> abnormal high Rule #2 for abnormal high: If Lastday3 == Abnormal High Data then -> abnormal high Rule #3 for abnormal high: If Lastday2==Normal Data	Rule #1 for abnormal low : if Lastday2 == Normal Data then -> abnormal low Rule #2 for abnormal low : if Lastday1 == Normal Data then -> abnormal low Rule #3 for abnormal low : if Lastday3 == Normal Data then -> abnormal low	Rule #1 for normal: if Weekdata == Normal Data then -> normal Rule #2 for normal: if Lastday3 == Normal Data then -> normal Rule #3 for normal: if Lastday5 == Normal Data then -> normal

then -> abnormal high Rule #4 for abnormal high: if Lastday1 == Normal Data then -> abnormal high Rule #5 for abnormal high: if Weekdata ==Normal Data then -> abnormal high	Rule #4 for abnormal low : if Lastday4 == Normal Data then -> abnormal low Rule #5 for abnormal low : if Lastday5 == Normal Data then -> abnormal low Rule #6 for abnormal low : if Weekdata == Normal Data then -> abnormal low	Rule #4 for normal: if Lastday1 == Normal Data then -> normal Rule #5 for normal: if Lastday4 == Normal Data then -> normal Rule #6 for normal: if Lastday2 == Normal Data then -> normal Rule #7 for normal: if Weekdata == Normal Data and Lastday3 == Normal Data then -> normal Rule #8 for normal: if Weekdata == Normal Data and Lastday1 == Normal Data then -> normal Rule #9 for normal: if Weekdata == Normal Data and Lastday4 == Normal Data then -> normal

6.5.3 Risk Identification and Analysis

Risk analysis and risk management is a discipline for dealing with the possibility that some future event will cause harm to personnel, equipment and company in future. It provides strategies and techniques to recognize and confront any threat faced by a company. There are three basic questions in risk management as following:

- What can go wrong?
- What will we do to prevent the harm from occurring and after an incident?
- If something happens, how will pay for it.

In this supply chain process case, the business risk is mainly from

- Market and customer demand change
- product supply risk
- Inner operation and decision-making risk

To decrease these risks, especially market change risk and risk of decision marking, the best way is to correctly predict market change and trend. At the same time, valuable supply chain knowledge, process information and process operation rules are very important to mark decision for supply chain process operation. In this supply chain case, the integrated intelligent model provides good function to risk identification and analysis.

In this supply chain process, for example, the whole process can be defined as normal, low risk and high risk state. In essence, the supply chain change is process state change, and the underlying process characteristics can be discovered with some description based rules, for example, decision tree rule, association rule and fuzzy rules, etc. Hence, integrated intelligent model can be applied for analyzing process risks due to following functions.

- Successfully predict market changes and customer demand changes on a certain scale, a good prediction value can be gotten, and process state also can be identified, especially for abrupt change in market and customer demand.
- Discover some valuable business rules for these business processes, decision tree rule describes the classification for market change, and association rule builds the relationship to express the frequent pattern. The two kinds of rules discover some supply chain process characteristics. They are helpful to decision making.

6.6 Chapter Summary

In this chapter, an application of the integrated intelligent model has been implemented for a practical supply chain process. It has demonstrated that the integrated intelligent model can be successfully applied in prediction and process analysis in supply chain process, especially for process state identification and performance measurement in supply chain process, process output prediction and risk analysis.

To a large enterprise of goods distribution for store market in Norway, a lot of kinds of products, an amount of money and operations are involved in the supply chain process every day. It is very important to correctly arrange these money and resource so as to adapt the changes from market and customers, and also decrease cost and improve efficiency. Hence, accurately predicting product amount and analyzing the supply chain process is an important business task and aim.

Firstly, a basic description about the supply chain process is given. The problems of supply chain process in Norsk Kjøttsamvirke AS is addressed and also their business goals are described.

Secondly, the supply chain process and its relevant data from Norsk Kjøttsamvirke AS are analyzed. Some basic features of the business process are also extracted. These results of the analysis show that these features approximately satisfy the precondition of integrated intelligent model developed in the paper. In essence, the case for product supply prediction in the supply chain process can be regarded as process output prediction and process state identification. It can be dealt with a process state detection and identification problems. This is why the integrated intelligent model can be applied in this business process analysis.

The model is implemented step by step in the section. Some important results from practical application are obtained and analyzed. It is obviously illustrated that the integrated intelligent model not only provides a good performance and also can solve some real practical problems. In this chapter, three main goals for supply chain process have been reached on the certain scale.

CHAPTER 7 CONCLUSIONS AND FURTHER WORK

7.1 Conclusions

The studies of the integrated intelligent model for process diagnosis, behavior prediction and control have shown with respect to both theoretical foundations and real application. In summary, the following work has been completed by this research.

1. Analyze the problems and challenges in process monitoring, diagnosis and control in modern processes
2. Extract some characteristics from industrial or business processes for process fault diagnosis problems
3. Present the basic techniques and methods for process fault detection, isolation and diagnosis
4. Answer some questions about integrated intelligent model for fault detection and quantitative analysis. Why is the integrated intelligent model needed? What are the design ideas, basic principles and design objectives? How dose it work?
5. Design and develop the integrated intelligent model for abnormal state (fault) identification, quantitative analysis and process prediction as well as process control
6. Discover knowledge and rules of process diagnosis and analysis with data mining technologies
7. Develop model optimization schemes and methods for model adaptive ability to cover the time-varying processes
8. Build a state space equation of discrete time varying system based on the integrated intelligent model for process control
9. Apply successfully the integrated intelligent model in a practical supply chain process for products forecasting and process diagnosis

Basis Features of the Integrated Intelligent Model

1. Modeling complex processes based on black-box principle. The model is established by training process data. No more prior knowledge of processes is needed. It can be regarded as an independent model on the concrete processes
2. Open architectures and structures. The model can be improved and elaborated using new technology and methodologies
3. Fuzzy information description for process symptom and process state (or fault state). Fuzzy sets and membership functions are involved to describe input data (as symptom) and process output data (as process fault)
4. Fuzzy information processing for process state (or fault) classification. The state classifiers based FNNs in the model play a residual generator for abnormal state (fault) detection as well as process state identification
5. Fuzzy inference model for process prediction based on process state identification. It is designed based on fuzzy TS dynamic NARX structure
6. Diagnosis knowledge and rules extraction with data mining technologies. Decision tree algorithms are applied for classification rules of process state change and association rules algorithms for frequent pattern

7. Optimal model and adaptability. Model optimization schemes are designed to cover time varying processes
8. Wide application fields and good solutions for complex process analysis, diagnosis and control

Application Areas

The integrated intelligent model can be applied in industrial and business processes as follows:

- Modeling complex processes for process monitoring, diagnosis and prediction and control
- Process performance measurement, especially in dynamic characteristics of processes
- Aberrant behaviors detection and fault detection and isolation (FDI)
- Prediction for process behavior and performance
- Process control based on different process state
- Risk analysis, control and process (fault) diagnose
- Business Intelligence (BI)

7.2 Major contributions

The contributions of this research work are summarized as follows:

1) Basic features extraction from real complex industrial and business processes.
2) Fuzzy description and definition for different process states. Fuzzy sets and membership functions are defined for input/output variables and they are generated by certain statistic methods from history data. These process state divisions are important for process analysis, fault diagnosis and knowledge discovery.
3) Fuzzy definition and quantitative description for process state definitions and classification.
4) Fuzzy information processing for detection of process state change as well as fault occurrence. Process state classifier based on FNNs is applied in process state classification (or fault detection in FDI problem) as well as process state identification. A threshold parameter in the model is defined for identification of process state change. When fuzzy degree of normal process state is greater than the threshold value, it indicates the process is in normal state, otherwise, it means the process is in abnormal state or a fault state occurs if the process always stays in abnormal states.
5) Fuzzy Inference model based on TS NARX dynamic structure for process state identification and process output prediction. In the model:
   - **Fault Residual Generator**: The process state (fault) residual signal w is generated by output of two fuzzy classifiers: one is used for detecting process state produced by real system inputs and another by historical behaviors. The outputs (w_1 and w_2) of two fuzzy classifiers are combined by fuzzy operation to produce final fault residual signal $w = \max(w_1, w_2)$.
   - **Process prediction**: The final model prediction output Y is determined by the combination of two aspects: one is from process response

produced by real system input, and another by process historical behaviors. It is calculated by Fuzzy TS NAXR model.

- **Inference Rule:** The Antecedent part, which is generated by two fuzzy classifiers, gives the residual generator for process state (or fault) detection and the Consequence part, which is generated by system identification based on module NNs structure and fuzzy degree information of process state, provides the process prediction output. These inference rules describe the relationship between process state and quantitative value of process output on high level description.

6) Two model optimization schemes for process diagnosis and prediction. One model optimization scheme is based on searching optimal or adaptive threshold values. Another is based on optimal prediction output from the model without any consideration process state classification. The first scheme is mainly used to process diagnosis. It can obtain better process or fault diagnosis ability but weaker prediction ability than second one. The second scheme is manly used to process prediction. It can provide better prediction ability but without any information concerning process state classification for process diagnosis. These optimization schemes can make model posses dynamic adaptive ability for change from outside environments.
7) Dynamic diagnosis knowledge and rules. Adaptive process state classification due to adaptive threshold value makes data sets in different process state dynamically change. Hence, diagnosis rules could be adaptively changed due to date sets dynamic changes.
8) Deduction for process state space model from integrated intelligent model and model optimization. In the state space equation, process state variables in the integrated intelligent model are defined as state variables. The state transition matrix $\boldsymbol{A}$ is determined by the fuzzy degree of process state classification produced by process historical behavior at time t instant, and input transition matrix $\boldsymbol{B}$ by the fuzzy degree of process state classification produced by real process inputs at time t instant, and state observer vector $\boldsymbol{H}$ is determined by optimization unit results. Based on the state space equation, optimal process Control can be applied for process control. It also can be regarded as a system identification way for system modeling.
9) A solution for products forecasting and process diagnosis in a real supply chain process for a Norwegian Company.

7.3 Further Work

An integrated intelligent model for process state deification, process diagnosis and prediction as well as control has been developed in the paper. It shows better performance to represent complex processes than some traditional methodologies do. However, the research work in this thesis just addressed some initial main problems encountered in the majority of applications and problems. Additional and some further envisioned work and several recommendations for further work are given as follows:

1) *Further improve the process diagnosis ability*. Some new algorithms based on fuzzy information processing could be involved in this model so as to improve model performance and output accuracy. For example, fuzzy decision tree

algorithm and fuzzy association rule algorithm for rule extraction and discovery. As the model shown, the whole model is built on fuzzy process state classification. The fuzzy information of process state classification not only plays the role of fault detection (as a residual generator), but also affect the result of quantitative process analysis. Hence, the algorithm of decision rule and association rule concerning fuzzy information should be considered for the case.

2) *Deal with different work points in processes*. Industrial processes often work at various operation points, however, demonstrated application for fault diagnosis usually only a signal (primary) operating point. Developing a standard scheme for process (fault) diagnosis at all operation points may be impractical due to the unavailability of suitable training data for less frequently used (secondary) operation points. Hence, How to involve the integrated intelligent model for further fault diagnosis in different operation points should be considered in further work.
3) *Establish the inner relationship between two different model optimization schemes*. In the model, two different model optimization schemes are developed so as to cover dynamic time-varying processes. One model optimization is for optimal or adaptive threshold values for process diagnosis and another is for optimal prediction output. Hence, it is more significant to establish their inner relationship to combine their advantages. A recursive equation for adaptive threshold value is also needed.
4) *Analyze control model derived from the integrated intelligent model and their performances*. Based on two different optimization schemes, two different state space equations can be obtained. The work about model comparison and performance analysis between them could provide some valuable conclusions. A real case based on their state space equations for process control is recommended and validated in further work. It is significant to evaluate the performance of state space model because the state space model is established with special system identification methods.
5) *Elaborate and decide the valuable and suitable rules from all rules*, which are gotten from knowledge discovery and data mining. Hence, developing effective standard for rule selection should be a reasonable research topic.

References

1. Alexandru M (2003), Analysis of Induction Motor Fault Diagnosis with Fuzzy Neural Network, APPLIED ARTIFICIAL INTELLIGENCE 17 (2): 105-133 FEB.
2. Azvine, B., Azarmi, N. Nauck, D. D, (2000), Intelligent Systems and Soft Computing. Prospects, Tools and Applications. Springer, New York, London, Hong Kong, Paris, 2000.
3. Applebaum (2001), Fuzzy Classification for Fault Isolation in Gas Turbine Engines.
4. Basseville, M (1998). Detecting Changes in Signals and Systems: A Survey. Automatic, 24 (3) :309-326.
5. Basseville, M and Nikiforov, I.V., (1993), Detection of Abrupt Changes in Signals and Dynamical System. Lecture Notes in Control and Information Sciences, Volume 77, Springer-Verlag, London.
6. Bakiotis, C., Raymond, J., and Rault, A., (1979). Parameter Identification and Discriminant Analysis for Jet Engine Mechanical State Diagnosis. IEEE Conference on Decision and Control, fort Lauderdale.
7. Brown, M. and Harris, C., (1994) Neurofuzzy Adaptive Modeling and Control, Prentice Hall Inc., London.
8. Balle, Peter (1998). Fuzzy Model-based Symptom Generation and Fault Diagnosis for Nonlinear Processes. IEEE International Conference on Fuzzy Systems, v 2, IEEE World Congress on Computational Intelligence, 1998, p 945-950.
9. Chen, J and Patton, R.J (1999). Robust Model–Based Fault for Dynamic System. Kluwer Academic.
10. Chow, E.Y. and Willsky, A.S (1984). Analytical Redundancy and The Design of Robust Detection Systems. IEEE Trans. Automatic Control, 29:603-614.
11. CI Byrnes, A Lidquist (1986). Modeling, Identification and Robust Control. Amsterdam:North Holland 1986.
12. Chen, Yubao (1995), Fuzzy Decision System for Fault Classification under High Levels of Uncertainty, Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, v 117, n 1, Mar, 1995, p 108-115.
13. Caminhas, W.M.;Tavares, H.; Gomide, F (1996). Neurofuzzy Approach for Fault Diagnosis in Dynamic Systems, IEEE International Conference on Fuzzy Systems, v 3, 1996, p 2032-2037.
14. Chen YM, Lee ML (2002) Neural Networks-based Scheme for System Failure Detection and Diagnosis, MATHEMATICS AND COMPUTERS IN SIMULATION, 58 (2): 101-109 JAN 7 2002.

15. Calado JMF, Korbicz J, Patan K, Patton RJ, da Costa JMGS (2001), Soft Computing Approaches to Fault Diagnosis for Dynamic Systems EUROPEAN JOURNAL OF CONTROL, 7 (2-3): 248-286 2001.

16. Delmaire, G., Gassar, P., Staroswiecki, M., and Christophe, C. (1999), Comparison of Multivariable Identification and Parity Space Techniques for FDI Purpose in M.I.M.O systems. In ECC'99. Karlsruhe, Germany.

17. D. Driankov, Hellendoorm, and M. Reifrank (1993). An Introduction to Fuzzy Control, Spring –Verlag, Heidelberg, Germany.

18. D. S Naidu (2003), Optimal Control System, CRC Press

19. Denison D.G.T, Holmes C.C, Mallick B.K, Smith A.F.M (2002), Bayesian Methods for Nonlinear Classification and Regression. WILEY, London. NY.

20. Dietz, W.E., Kietch, E.L., and Ali, M (1989). Jet and Rocket Engine Fault Diagnosis using Fuzzy Matching. IEEE Trans. On Sys. Man and Cyber. Part A: System, 27(5):673-682.

21. Davenport, T. H. & J. Short, (1990), The New Industrial Engineering: Information Technology and Business Process Redesign', Sloan Management *Review* 31(4), 11-27.

22. Davenport, T. H. (1993), Process Innovation. Reengineering Work through Information Technology, Harvard Business School Press, Boston. Tradotto in: Innovazione dei Processi, Franco Angeli, Milano.

23. E.Hernandez and Y.Arkun (1993). Control of Nonlinear Systems using Polynomial ARMA Models, AICHE Journal, 39 (3): 446-460.

24. E. Ikonen, K. Najim (2002), Advance Process Identification and Control, Marcel Dekker, Inc. NY.Basel, USA.

25. E. S Levitin, B. T.Polyak (1966), Metody Minimizacii Pri naličii Ograničenii. Ž.Vyčisl. Mat. i Mat. Fiz.6 (1966), 787 – 823.

26. Frank, P.M (1993) Advances in Observer-based Fault Diagnosis. Proc. TOOLDIAG'93 Conference. CERT, Toulose (F).

27. Fantuzzi,-C.; Simani,-S.; Beghelli,-S.,(2001),Robust Fault Diagnosis of Dynamic Processes using Parametric Identification with Eigenstructure Assignment Approach, Proceedings of the 40th IEEE Conference on Decision and Control, 2001: 155-60 vol.1.

28. Fuzzy Logic ToolBox for use with Matlab. User's Guide version 2.copyright 1995-2002 by The MathWorks, Inc.1997.

29. Gertler, J. (1998), Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, Now York.

30. Gupta, M. M., Rao, D. H., (1994) Neuro-Control Systems Theory and Applications, IEEE Press, New York.

31. Ganeshan. R, Harrison, T.P., (1995)- Introduction to Supply Chain Management, 1995,

32. Geiger, G.,(1982), Monitoring of an Electrical Driven Pump Using Continuous-time Parameter Estimation Models, In Pergamon Press editor, 6th IFAC Symposium on Identification and Parameter Estimation, Washington.

33. GeneHunter® Release 2.4, Copyright 1994-2000, Ward System Group, Inc.

34. Gertler, J. (1998).Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, New York.

35. Gupta, M. M., (1994) Fuzzy Neural Networks: Theory and Applications, Proceedings of SPIE, Vol. 2353, the International Society for Optical Engineering, pp. 303-325.

36. Guralnik, V., Srivastava, J.,(1999), Event Ddetection from Time Series Data. Proc. of KDD (1999) 33—42.

37. Hoskins, J.C, Himmelblau, D.M (1988), Artificial Neural Network Models of Knowledge Representation in Chemical Engineering. Comp. Chem.. Engng, 12:881-890.

38. Han jawei, Kamber Micheline (2001), Data Mining: Concepts and Techniques. MORGAN KAUFMANN OUBLISHERS.

39. Himmelblau, D, M., (1978), Fault Diagnosis in Chemical and Petrochemical Processes. Elsevier, Amsterdam.

40. Hohmann, H., (1977), Automatic Monitoring and Fault Diagnosis for Machine Tools. Dissertation, T. H. Darmstadt, Germany.

41. Hecht-Nielsen, R. (1990), Neurocomputing, Reading, MA: Addison-Wesley.

42. Huallpa, B. N, N, Euripedes; al., (1998), Fault Detection in Dynamic Systems based on Fuzzy Diagnosis. IEEE International Conference on Fuzzy Systems, v 2, IEEE World Congress on Computational Intelligence, 1998, p 1482-1487.

43. Hunter J, McIntosh N, (1999), Knowledge-based Event Detection in Complex Time Series Data, Artificial Intelligence in Medicine Lecture Notes in Artificial Intelligence, 1620: 271-280 1999.

44. Hogan, P. A., et al., (1991) A System for Diagnostic Faults in Hydraulic Circuits Based on Qualitative Models of Component Behavior, SAFEP ROCESS '91, IFAC-IMACS Symposium on Fault Detection, Supervision, and Safety for Technique Process, pp. 39-44.

45. Hung T. Nguyen Nadipuram R.et (2002),A First Course in Fuzzy and Neural Control. CRC Pr I Llc.

46. H.X Li and Guan S, (2001) "Hybrid Intelligent Control Strategy", IEEE Control Systems Magazine, full paper, Vol.21, No.3, Jun 2001, pp 36-48.

47. Isermann, R (1984). Process Fault Detection based on Modeling and Estimation Methods: A Survey. Automatic 20(4):387-404.

48. Isermann, R., (1997), Supervision, Fault Detection and Fault Diagnosis Methods: An Introduction. Control Engineering Practice, 5 (5): 639-652.

49. J, Abonyi (2002) Fuzzy Model Identification for Control. Birkhauser, boston, Basel, Berlin.
50. J.Sjoberg, Q.Zhang, L.ljung (1995). Nonlinear Black-box Modeling in System Identification; a Unified Overview. Automatic, 3:1691-1724.
51. J. H. Holland (1994), Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, The MIT Press.
52. J. H. Holland (1973), Genetic Algorithms and the Optimal Allocation of Trials. SIAM J. Comput. 2(2): 88-105 (1973).
53. John H. Holland (1962), Outline for a Logical Theory of Adaptive Systems. J. ACM 9(3): 297-314.
54. Jang, J.-S. R. and C.-T. Sun, (1997) Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall.
55. J.B.Gomm, (1998), Adaptive Neural Network Approach to On-line Learning for Process Fault Diagnosis, Transactions of the Institute of Measurement and Control, v 20, n 3, 1998, p 144-152, Inst of Measurement & Control.
56. Javadpour R, Knapp GM (2003), A Fuzzy Neural Network Approach to Machine Condition Monitoring, COMPUTERS & INDUSTRIAL ENGINEERING, 45 (2): 323-330 AUG 2003.
57. Jean-Marc Adamo, (2000), Data Mining for Association rules and sequential patterns, sequential and parallel algorithms, Springer, NY, Berlin, Hong Kong, London, Paris, Tokyo.
58. K,Vojislav (2001), Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. The MIT Press and Cambridge, London.
59. K. Evanglos, W.Antoni (2000), MAPS: A Method for Identifying and Predicting Aberrant Behavior in Time Series, Engineering of Intelligent Systems, 14th International Conference on Industrial an Engineering Applications of Artificial Intelligence and Expert System, IEA/AIE 2001.
60. Kosko, B., (1992) Neural Networks and Fuzzy Systems, Prentice-Hall, NJ.
61. K.S .Narendra and K.Parthasarathy (1990). Identification and Control of Dynamic Systems. IEEE Transactions on Neural Networks, 1:4-27.
62. Koch, W. H. (2002), Nutzerdokumentation zum Softwaresystem NOSYS für die dialoggeführte numerische, 2002.
63. Koch, W. H. (1993), NOSYS Nonlinear Optimization System - Oppgavesamling. Undervisningsmaterial 93-005-PT, Narvik Institute of Technology, Narvik, 1993.
64. Koch, W. H. (1994), NOSYS Nonlinear Optimization System - En arbeidsveiledning. Narvik Institute of Technology, Narvik, 1994.

65. Koch, W. H. (1996), PC-Bearbeitung stetiger Optimierungsaufgaben. Narvik, 1993, revidiert 1996.

66. Koch, W.H (1983), Adaptiv genaue LEVITIN/POLJAK-Typ-Verfahren zur Behandlung linear restringierter: Optimierungsaufgaben ohne Verwendung von Ableitungen. Dissertation, Fakultät Mathematik und Naturwissenschaften der TU Dresden.

67. Koizumi, T. and Kisv, M., (1986) Preventive Maintenance for Roller and Journal Bearings of Induction Motor Based on the Diagnostic Signature Analysis, Trans. ASME Journal of Vibration, Acoustics, Stress and Reliability in Design, Vol. 108, 26-31.

68. Karpenko M, Sepehri N (2002), Neural Network Classifiers Applied to Condition Monitoring of a Pneumatic Process Valve Actuator ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE 15 (3-4): 273-283 JUN-AUG 2002.

69. Kandel, A., (1996) Fuzzy Expert System Tools, John Wiley & Sons Ltd., London.

70. Kosko, B. (1992), Neural Networks and Fuzzy Systems, Englewood Cliffs, N.J.: Prentice-Hall.

71. Katharina Morik, (2001), Summarizing Time Series and the Detection of Event Sequences, Lecture Note, Unvi. Dortmund.

72. L. A. Zadeh (1965), Fuzzy Sets, Information and Control, Vol.8, 338-353.

73. L. A. Zadeh (1994), Fuzzy Logic, Neural Networks and Soft Computing, Communications of the ACM, 37(3):77-84

74. L. A. Zadeh (1978), Fuzzy Sets as a Basic for Theory of Possibility, Fuzzy Sets and Systems, Vol. 1, 3-28.

75. L. Ljung (1987), System Identification, Theory for the User. Prentice-Hall, New Jersey.

76. L. Armijo (1966), Minimization of Functions Having Lipschitz-Continuous first Partial Derivatives. Pacific J. Math. 16, 1 – 3.

77. Lee IS, Kim JT, Lee JW, Lee DY, Kim KY, (2003), Model-based Fault Detection and Isolation Method using ART2 Neural Network, INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS 18 (10): 1087-1100 OCT 2003

78. M.A Henson (1998), Nonlinear Model Predictive Control: Current Status and Future Directions. Computers and Chemical Engineering.

79. Mathew, J. and Alfredson, R. J., (1983) The Condition Monitoring of Rolling Element Bearings Using Vibration Analysis, ASME WAM.

80. MS Rajbman, VM Chadeer (1980). Identification of Industrial Processes. Amsterdam: North Holland, 1980.

81. Mitchell, J. S., (1981) An Introduction to Machinery Analysis and Monitoring, PennWellBook, Okla., USA
82. Mamdani, E.H. and S. Assilian, (1975), An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975
83. Mark L, Yaron K, and Abranham K, (2001), Knowledge Discovery in Time Series Database. IEEE Transaction on System, Man, and Cybernetics-Part B: Cybernetics. VOL. 31, No.1, FEBRUARY 2001.
84. N. K. Sinha, M. M. Gupta (1999), Soft Computing & Intelligent Systems. Theory & Applications, ACADEMIC Press, NY, London, Tokyo
85. N, Khiripet, (2001), An Architecture for Intelligence Time Series Prediction With Causal Information, Ph.D. Dissertation, university of Atlanta, Georgia USA, May 2001.
86. Pan Dan, Zheng Qilun, Huang (1999), Fuzzy Classification of Fault Diagnosis using ANNs. Huanan Ligong Daxue Xuebao/Journal of South China University of Technology (Natural Science), v 27, n 4, Apr, 1999, p 92-97.
87. Patton, RJ. And Chen, J (1991). A Review of Parity Space Approaches to Fault Diagnosis. In IFAC Symposium SAFEPROCESS '91 Baden-Baden.
88. Pedrycz, W., (1993) Fuzzy Control and Fuzzy Systems, John Wiley & Sons, New York.
89. Patton, R. J., Frank, P.M., and Clark, R.N editors (1989). Fault Diagnosis in Dynamic System, Theory and Application. Control Engineering Series. Prentice Hall, London.
90. P Eykhoff (1977). System Identification, Parameter and State Estimation. New York: John Wiley & Sons 1977.
91. Pau, L. F., (1981), Failure Diagnosis and Performance Control,Marcel Dekker, New York.
92. Patton R.J, Frank, P. M., and Clark, R.N., editors (2000), Issues of Fault Diagnosis for Dynamic Systems, Springer-Verlag, London Limited.
93. P, Thomas; A, Mihiar (1997), Application of a Hybrid Neuro-Fuzzy System to the Fault Diagnosis of An Automotive Electromechanical Actuator Fuzzy Sets and Systems, v 89, n 3, Aug 1, 1997, p 351-360.
94. Peter J. Rockwell, Richard A. D (2002), Introduction to Time Series and Forecasting, Second Edition, Springer.
95. Ogaji SOT, Singh R (2002), Neural network Technique for Gas Turbine Fault Diagnosis, ENGINEERING INTELLIGENT SYSTEMS FOR ELECTRICAL NGINEERING AND COMMUNICATIONS 10 (4): 209-214.
96. Q, Shen, C, Alexios (2000), A Modular Approach to Generating Fuzzy Rule with Reduced Attributes for the Monitoring of Complex Systems. Engineering Application of Artificial Intelligence 13 (2000) 263-278.

97. R. J. Patton,S. Simani, S. Daley, (2001), Fault Diagnosis of a Simulated Model of an Industrial Gas Turbine Prototype using Identification Techniques, 5th National Conference on Diagnostics of Industrial Processes. 2001: 445-51.

98. R.J. Patton, C.T. Lopez-Toribio, S. Simani, (2001), Robust Fault Diagnosis in a Chemical Process using Multiple-Model Approach, Proceedings-of-the-40th-IEEE-Conference-on-Decision-and-Control-Cat.-No.01CH37228. 2001: 149-54 vol.1.

99. Rengaswamy R, Mylaraswamy D (2001) A Comparison of Model-based and Neural Network-based Diagnostic Methods, ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE, 14 (6): 805-818 DEC.

100. R. Diversi, , S. Simani, , U. Soverini, (2002), Robust Residual Generation for Dynamic Processes using De-coupling Technique, Proceedings of the 2002 IEEE International Conference on Control Applications, 2002: 1270-5 vol.2.

101. Ramdén, T., (1995) On Condition Monitoring of Fluid Power Pumps and Systems, Unitryck, Linköping University, Sweden.

102. Rodriguez, C, Rementeria, S (1996), Modular Neural Network Approach to Fault Diagnosis, IEEE Transactions on Neural Networks, v 7, n 2, Mar, 1996, p 326-340, IEEE.

103. Rasmussen, M. et al., (1993) Neural Networks in Condition Monitoring and Diagnosis on Rotating Machinery, Fourth Ship Control System Symposium, Ottawa, Canada.

104. Ryszard S, Ivan Bratko (1998), Machine Learning and Data Mining. Methods and applications. John Wiley & Sons Ltd. NY, Singapore, Weinheim

105. S. Simani, C Fantuzzi (2002), Neural Networks for Fault Diagnosis and Identification of Industrial Processes 10th-European-Symposium-on-Artificial-Neural-Networks.-ESANN'2002.-Proceedings. 2002: 489-94.

106. S. Simani, C Fantuzzi (2000), Fault Diagnosis in Power Plant using Neural Networks, International Journal of Information Sciences, 127(3-4):125-136. Special Issue: Application to Intelligence Manufacturing and Fault Diagnosis: Part 1-Fault Diagnosis.

107. S. Simani, C. Fantuzzi and R.J. Patton (2002), Model-based Fault Diagnosis in Dynamic System Using Identification Techniques. The Springer press, London, Berlin, New York, Hong Kong, Paris and Tokyo.

108. S. Simani, R. J. Patton, (2002), Neural-Networks for Fault Diagnosis of Industrial Plants at Different Working Points. ESANN'02, P495-500, Bruges, Belgium, Proc. of the 10th European Symposium on Artificial Neural Networks.

109. SPSS inc, (2001), Clementine 6.5 User's Guild

110. S. Haykin (1996), Adaptive Filter Theory, Third edition, PRENTICE HALL.

111. Sugeno, M.,(1985), Industrial Applications of Fuzzy Control, Elsevier Science Pub. Co., 1985.

112. S, Leonard; M, Francesco, (1997), Building a Neuro-Fuzzy System to Efficiently Forecast Chaotic Time Series, Proceedings of the 1996 5th International Workshop on Software Engineering, Neural Nets, Genetic Algorithms, Expert Systems, Symbolic Algebra and Automatic Calculations in Physics Research, AIHENP'96, v 389, n 1-2, Apr 11.

113. Tang, M., Koch, W. H., (2004), An Integrated Intelligent Model based Fuzzy TS NARX for Process State Detection and Prediction, IEEE 4th International Conference on Intelligent Systems Design and Applications, ISDA 2004.

114. T. Burlton, Roger (2001), Business Process Management, Profiting from process.SAMS. USA.

115. Takagi, H. and Hayashi, I., (1991) NN-Driven Fuzzy Reasoning, International Journal Approximate Reasoning, Vol. 5, No. 3, pp.191-212.

116. Tsien CL, (2000), Event Discovery in Medical Time-Series Data, JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION, 858-862 Suppl. S 2000.

117. Tom M. Mitchell (1998), Machine Learning, Carnegie Mellon University, the McGraw-Hill Companies, Inc.

118. Wang, K.S, Tang, M (2002), Business Sales Forecasting for products of Norwegian Meat Centre Based on Artificial Neural Networks. The Conference of ICFDM, Dalian, China, July.

119. Willsky, A.S., (1976), A Survey of Design Methods for Failure Detection in Dynamic Systems, Automatica, 12(6): 601-611.

120. Willi. K, Jan M. Z, (2002), Handbook of Data Mining and Knowledge Discovery, Oxford University Press, Oxford, New York, Shanghai, Singapore, Taipei, Tokyo, Toronto, Berlin

121. Zhang, Jie; Morris (1994), A. Julian Process Dault Diagnosis using Fuzzy Neural Networks Proceedings of the American Control Conference, v 1, 1994, p 971-975.

122. Zilouchian, Ali and Jamshidi, Mo (2001) Intelligent Control Systems Using Soft Computing Methodologies. The CRC Press, London and NY.

Appendix A: Process Data from supply Chain Process (Part)

Aar	uke	salgsvare	tekst	kjede	navn
2000	36	472674	Karb Kaker 0,8kg Pk	04153	ISP NKL-kjede Prix
2000	36	472674	Karb Kaker 0,8kg Pk	04157	ISP NKL-kjede S-Marked
2000	36	472674	Karb Kaker 0,8kg Pk	04160	ISP NKL-kjede Mega
2000	36	472674	Karb Kaker 0,8kg Pk	04400	ISP Hakon-kjede Ica
2000	36	472674	Karb Kaker 0,8kg Pk	04550	ISP NKL-kjede Obs
2000	36	472674	Karb Kaker 0,8kg Pk	10000	ISP Storkjøkken
2000	36	472674	Karb Kaker 0,8kg Pk	04600	ISP Rema
2000	36	472674	Karb Kaker 0,8kg Pk	04403	ISP Hakon-kjede Rimi
2000	36	472674	Karb Kaker 0,8kg Pk	04200	ISP Norgesgruppen Bunnpris
2000	36	472674	Karb Kaker 0,8kg Pk	04159	ISP NKL-kjede S-lag
2000	36	472674	Karb Kaker 0,8kg Pk	88047	Nord Møre Romsdal ISp Kjede
2000	36	472674	Karb Kaker 0,8kg Pk	88054	Nord Norges Salgslag ISP Kjede
2000	36	472674	Karb Kaker 0,8kg Pk	04113	ISP Hakon-kjede Sparmat
2000	36	472674	Karb Kaker 0,8kg Pk	04115	ISP Hakon-kjede Servicemat
2000	36	472674	Karb Kaker 0,8kg Pk	04102	ISP Norgesgruppen Joker
2000	36	472674	Karb Kaker 0,8kg Pk	04103	ISP Norgesgruppen Nærmat
2000	36	472674	Karb Kaker 0,8kg Pk	04106	ISP Norgesgruppen Spar
2000	36	472674	Karb Kaker 0,8kg Pk	04105	ISP Kiwi Østerdal
2000	36	472674	Karb Kaker 0,8kg Pk	04107	ISP HK-kjøpmenn Service
2000	36	472674	Karb Kaker 0,8kg Pk	04111	Joker Reg.Innland
2000	36	472674	Karb Kaker 0,8kg Pk	04112	Nærmat Reg.Innland
2000	36	472674	Karb Kaker 0,8kg Pk	04109	ISP HK-kjøpmenn Hedemark
2000	35	472674	Karb Kaker 0,8kg Pk	04102	ISP Norgesgruppen Joker
2000	35	472674	Karb Kaker 0,8kg Pk	04105	ISP Kiwi Østerdal
2000	35	472674	Karb Kaker 0,8kg Pk	04103	ISP Norgesgruppen Nærmat
2000	35	472674	Karb Kaker 0,8kg Pk	04106	ISP Norgesgruppen Spar
2000	35	472674	Karb Kaker 0,8kg Pk	04112	Nærmat Reg.Innland
2000	35	472674	Karb Kaker 0,8kg Pk	04109	ISP HK-kjøpmenn Hedemark
2000	35	472674	Karb Kaker 0,8kg Pk	04107	ISP HK-kjøpmenn Service
2000	35	472674	Karb Kaker 0,8kg Pk	04113	ISP Hakon-kjede Sparmat
2000	35	472674	Karb Kaker 0,8kg Pk	04115	ISP Hakon-kjede Servicemat
2000	35	472674	Karb Kaker 0,8kg Pk	88047	Nord Møre Romsdal ISp
2000	35	472674	Karb Kaker 0,8kg Pk	88054	Nord Norges Salgslag ISP
2000	35	472674	Karb Kaker 0,8kg Pk	04153	ISP NKL-kjede Prix
2000	35	472674	Karb Kaker 0,8kg Pk	88035	Vestlandske Salgslag ISP
2000	35	472674	Karb Kaker 0,8kg Pk	10000	ISP Storkjøkken
2000	35	472674	Karb Kaker 0,8kg Pk	04600	ISP Rema
2000	35	472674	Karb Kaker 0,8kg Pk	04550	ISP NKL-kjede Obs
2000	35	472674	Karb Kaker 0,8kg Pk	04403	ISP Hakon-kjede Rimi
2000	35	472674	Karb Kaker 0,8kg Pk	04400	ISP Hakon-kjede Ica
2000	35	472674	Karb Kaker 0,8kg Pk	04200	ISP Norgesgruppen Bunnpris
2000	35	472674	Karb Kaker 0,8kg Pk	04160	ISP NKL-kjede Mega

salgsavd	kgmandag	kgtirsdag	kgonsdag	kgtorsdag	kgfredag	kgloerdag	utsalgsprisloerdag
049	256.8	121.6	183.2	240.8	258.4	0	0
049	56.8	53.6	27.2	123.2	84.8	0	0
049	236.8	112	84.8	154.4	100.8	0	0
049	20	20	16	16	84	0	0
049	40	0	0	40	120	0	0
049	0	0	0	0	0	0	0
049	152.8	72.8	168.8	106.4	224	0	0
049	68.8	108	48.8	184	70.4	0	0
049	102.4	139.2	48	88	201.6	0	0
049	0	6.4	0	2.4	0	0	0
049	640	0	0	1200	0	0	0
049	1651.2	0	1651.2	0	243.2	0	0
049	12.8	54.4	34.4	36.8	40.8	0	0
049	1.6	9.6	17.6	15.2	16	0	0
049	8	12	18.4	6.4	29.6	0	0
049	0	20.8	12	12.8	30.4	0	0
049	0	24	4.8	6.4	50.4	0	0
049	0	8	0	8	0	0	0
049	0	1.6	0	1.6	0	0	0
049	0	2.4	0	0	0	0	0
049	0	0	0	4.8	0	0	0
049	0	16	0	13.6	6.4	0	0
049	72.8	296.8	44.8	159.2	120.8	0	0
049	0	6.4	0	4.8	0	0	0
049	4	11.2	8	30.4	19.2	0	0
049	8	8	12	44	20	0	0
049	0	0	0	8	0	0	0
049	0	12.8	0	12	4	0	0
049	0	1.6	0	0	0	0	0
049	14.4	41.6	32.8	30.4	50.4	0	0
049	11.2	3.2	16	14.4	13.6	0	0
049	800	0	0	640	0	0	0
049	1651.2	0	1804.8	0	1216	0	0
049	223.2	164.8	90.4	253.6	272.8	0	0
049	0	0	0	0	0.8	0	0
049	0	0	0	12	0	0	0
049	195.2	59.2	110.4	124	262.4	2.4	82.5
049	240	240	0	364.8	524.8	0	0
049	52	42.4	39.2	134.4	123.2	0	0
049	12.8	11.2	17.6	20	12.8	0	0
049	125.6	132.8	45.6	95.2	153.6	0	0
049	173.6	96.8	89.6	236.8	66.4	0	0

...

bestiltkgmandag	bestiltkgtirsdag	bestiltkgonsdag	bestiltkgtorsdag	bestiltkgfredag	bestiltkgloerdag
258.4	122.4	183.2	240.8	258.4	0
56.8	53.6	28	123.2	84.8	0
236.8	112	84.8	154.4	100.8	0
20	20	16	16	84	0
40	0	0	40	120	0
0	0	0	0	0	0
156	72.8	168.8	107.2	224	0
68.8	108	49.6	184	70.4	0
103.2	139.2	48	88	201.6	0
0	6.4	0	2.4	0	0
640	0	0	1200	0	0
1651.2	0	1651.2	0	240	0
12.8	54.4	34.4	36.8	40.8	0
1.6	9.6	17.6	15.2	16	0
8	12	18.4	6.4	29.6	0
0	20.8	12	12.8	30.4	0
0	24	4.8	6.4	50.4	0
0	8	0	8	0	0
0	1.6	0	1.6	0	0
0	2.4	0	0	0	0
0	0	0	4.8	0	0
0	16	0	13.6	6.4	0
72.8	296.8	44.8	160	120	0
0	6.4	0	4.8	0	0
4	11.2	8	30.4	19.2	0
8	8	12	44	20	0
0	0	0	8	0	0
0	12.8	0	12	4	0
0	1.6	0	0	0	0
14.4	41.6	32.8	30.4	50.4	0
11.2	3.2	16	14.4	13.6	0
800	0	0	640	0	0
1651.2	0	1804.8	0	1216	0
224.8	164	90.4	253.6	275.2	0
0	0	0	0	0.8	0
0	0	0	12	0	0
195.2	59.2	110.4	124	267.2	2.4
240	240	0	364.8	524.8	0
52	42.4	39.2	134.4	123.2	0
12.8	11.2	17.6	20	12.8	0
125.6	134.4	45.6	95.2	153.6	0
173.6	96.8	89.6	238.4	66.4	0

...

Prognosekg mandag	Prognosekg tirsdag	Prognosekg onsdag	Prognosekg torsdag	Prognosekg fredag	Prognosekg loerdag
304	162	186	301	314	0
78	68	39	132	70	0
220	146	106	276	146	0
24	19	21	32	28	0
130	88	102	85	318	0
1	0	1	4	1	0
199	97	157	122	241	0
5	5	5	7	6	0
134	174	91	127	216	0
0	12	1	7	13	0
381	528	48	852	20	0
1780	477	1591	205	1726	0
17	44	42	42	66	0
8	12	25	15	25	0
11	64	25	52	57	0
8	18	23	25	29	0
11	28	12	30	29	0
11	64	25	52	57	0
8	18	23	25	29	0
11	28	12	30	29	0
17	44	42	42	66	0
8	12	25	15	25	0
381	528	48	852	20	0
1780	477	1591	205	1726	0
304	162	186	301	314	0
1	0	1	4	1	0
199	97	157	122	241	0
192	192	192	192	192	0
5	5	5	7	6	0
24	19	21	32	28	0
134	174	91	127	216	0
220	146	106	276	146	0

…

Appendix B: Program for Data Processing

```
/_______________________________________________________________________________
_
/                         Data prepare for neural network training
/ Under the Excell Work Environment, Visual Baisc application (VBA) programming
/_______________________________________________________________________________
_

Sub daydata()
Dim g(5) As String

r = 1
g(0) = "C"
g(1) = "D"
g(2) = "E"
g(3) = "F"
g(4) = "G"

ActiveWorkbook.Sheets(2).Select

For i = 2 To 54
For j = 0 To 4
ActiveWorkbook.Sheets(2).Select

strCell = g(j) & i

Range(strCell).Select
Selection.Copy
Sheets(3).Select
Range("b" & r).Select
ActiveSheet.Paste
r = r + 1

Next j
Next i
End Sub

Sub NeuData()

Dim g(6) As String

g(0) = "B"
g(1) = "C"
g(2) = "D"
g(3) = "E"
```

```
g(4) = "F"
g(5) = "G"

ActiveWorkbook.Sheets(3).Select
k = 0
For j = 0 To 265
   For i = 1 To 6

   ActiveWorkbook.Sheets(3).Select
   Range("b" & (i + j)).Select
   Selection.Copy
   ActiveWorkbook.Sheets(4).Select
   strCell = g(i - 1) & (j + 1)
   Range(strCell).Select
   ActiveSheet.Paste

Next i
Next j

End Sub
```

Appendix C: Model Optimization based on Process State Identification

/____
_
/   Model Optimization by **Matlab** ®
/ The program is developed based on Matlab , only one parameter is considered as a
/optimized parameter

____
_

Object function

```
function S = objectfunction(Data,a)
S=0;
for i=1:31

   w1=Data(i, 4);
   w2=Data(i, 5);
   w3=Data(i, 6);

   % low state
   if w3==0
      if w2> a
         PredValue(i,1)=Data(i, 1);
      else
         PredValue(i,1)=Data(i, 1)*w2+Data(i, 2)*w1;
      end
   end

   % high State
   if w1==0
      if w2> a
         PredValue(i,1)=Data(i, 1);
      else
         PredValue(i,1)=Data(i, 1)*w2+Data(i, 3)*w3;
      end
   end


   E=(PredValue(i,1)-Data(i, 8))^2;
   S=S+E;
end
 S=sqrt(S);
```

/----Optimization program with Matlab

Optimization main function

```
format long
load StateData.txt

StateData

%-----object function call for model optimization---------

for i=0:100
   x=i/100;
   y(i+1,1)=feval(@objectfunction,StateData,x);
end

grid on
hold on
x=0:0.01:1;
plot(x,y);

%optimization algorithm for this case
         x=fminbnd(@objectfunction,0,1.0)


/_____________________________________________________________________________
_
/                              Model Optimization by NOSYS®
/ The program is developed based on NOSYS System, only one parameter is considered
/  as a optimized parameter
/
/_____________________________________________________________________________

    ■  SUBROUTINE F(I,X,IAX,Y,IE)
C
C     Threshold optimization for fault detection... (Tang, M. & Koch, W. H. (2004))
C     by Gaussian least squares minimization of the tradeoff between transformed model
data and real plant values.
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     DIMENSION X(1)
     COMMON /STP1/IPRI,ITMA,KA,IWA
     COMMON /STP4/IT,TEST/IOP/LR,LW
     DIMENSION outpnorm(31),outplow(31),outphigh(31),w1l(31),w2n(31),w3h(31),
*realval(31),predval(31)
         IF (IWA.NE.0.OR.TEST.NE.1.0D0) GOTO 50
         OPEN(FILE='PROCDAT.DAT',STATUS='OLD',UNIT=8)
         DO 1 L=1,31
      READ(LR,5)outpnorm(L),outplow(L),outphigh(L),w1l(L),w2n(L),
w3h(L),realval(L)
1        CONTINUE
 5     FORMAT(7D15.7)
```

```
       WRITE(LW,10)
10     FORMAT(/2X,'Modell data for fault detection problem (Ex.286,TANG/KOCH
[2004])):',
    *  /4X,'Normal:        Low:         High:'/)
        DO 15 L=1,31
      WRITE(LW,20)outpnorm(L),outplow(L),outphigh(L)
15     CONTINUE
      WRITE(LW,16)
16     FORMAT(/4X,'w1l:       w2n:          w3h:       Real value:'/)
        DO 18 L=1,31
      WRITE(LW,21)w1l(L),w2n(L),w3h(L),realval(L)
18     CONTINUE
20     FORMAT(1X,3F15.7)
21     FORMAT(1X,4F15.7)
        CLOSE (UNIT=8)
 5      DO 80 L =1,31
        if (w2n(L).GT.alambda) then
        predval(L) = outpnorm(L)   C     Normal state
        Else
        If(w3h(L).EQ.0.D0) then
              predval(L) = outpnorm(L)*w2n(L) + w1l(L)*outplow(L)
        If(w1l(l).EQ.0.D0) Then
               predval(L) = outpnorm(L)*w2n(L) + w3h(L)*outphigh(L)
        End If
    80  CONTINUE
      Y = 0.0
      DO 100 L=1,31
      Y1 = predval(L) - realval(L)
      Y= Y + Y1*Y1
100    CONTINUE
        Y= DSQRT(Y)
        IWA=IWA+1
        IE=-1
     END
```

Parameters Set

1	Number of variables X1,...,XN
0	Number of nonlinear equality constraints
0	Number of linear equality constraints
0	Number of nonlinear inequality constraints
0	Number of linear inequality constraints
-3	Problem type (if negative: STOCH as starting phase)
1	Type of bounds
1	Upper bound for X1 * value of N
.5	= Coordinates of
1	Output format parameter

```
100             Maximum iteration number
.1              Discretisation parameter for derivatives
.000001         Epsilon bound for the stopping rule
-100            Method selection parameter
END
```

/____

/ Model Optimization with GegnHunter ®

/ The program is developed based on Visual Basic Application under Microsoft Excel /2002. Only one parameter is considered as an optimized parameter

---

```
Public Function Fitness(a)

    Dim w1 As Single
    Dim w2 As Single
    Dim w3 As Single
    Dim s As Long
    Dim O As Single

    ActiveWorkbook.Sheets("Optimization").Select
     s = 0
    For i = 1 To 31

        w1 = Worksheets("Optimization").Range("E" & (1 + i)).Value
        w2 = Worksheets("Optimization").Range("F" & (1 + i)).Value
        w3 = Worksheets("Optimization").Range("G" & (1 + i)).Value

    'low state
    If w2 > a Then
            O = Worksheets("Optimization").Range("A" & (1 + i)).Value
          Else
            If w3 = 0 Then 'low state
            O = w2 * Worksheets("Optimization").Range("A" & (1 + i)).Value + w1 *
Worksheets("Optimization").Range("B" & (1 + i))
            End If

            If w1 = 0 Then
            O = w2 * Worksheets("Optimization").Range("A" & (1 + i)).Value + w3 *
Worksheets("Optimization").Range("C" & (1 + i))
            End If
          End If
    'Prediction value

    ' Calculation the sum
    T = (O - Worksheets("Optimization").Range("J" & (1 + i)).Value) ^ 2
    s = s + T
```

```
    Next i
        Fitness = Sqr(s)

End Function
```

/____
/ Model Optimization with GA algorithm, Software Tool: GegnHunter ®
/the program is developed based on Visual Basic Application under Microsoft Excel
/2002 Two parameters are considered as optimized parameters
/____

```
Public Function Fitness(a1, a2)

    Dim w1 As Single
    Dim w2 As Single
    Dim w3 As Single
    Dim s As Long
    Dim O As Single

    ActiveWorkbook.Sheets("Optimization").Select
     s = 0
    For i = 1 To 31

        w1 = Worksheets("Optimization").Range("E" & (1 + i)).Value
        w2 = Worksheets("Optimization").Range("F" & (1 + i)).Value
        w3 = Worksheets("Optimization").Range("G" & (1 + i)).Value

    'low state
    If w2 > a1 Then
            O = Worksheets("Optimization").Range("A" & (1 + i)).Value
          Else
            If w3 = 0 Then 'low state
            O = w2 * Worksheets("Optimization").Range("A" & (1 + i)).Value + w1 *
Worksheets("Optimization").Range("B" & (1 + i))
            End If
    End If
     If w2 > a2 Then
            O = Worksheets("Optimization").Range("A" & (1 + i)).Value
        Else
            If w1 = 0 Then
            O = w2 * Worksheets("Optimization").Range("A" & (1 + i)).Value + w3 *
Worksheets("Optimization").Range("C" & (1 + i))
            End If
          End If
    '---predicqtion value
```

```
' caculation the sum
T = (O - Worksheets("Optimization").Range("J" & (1 + i)).Value) ^ 2
s = s + T

Next i

 Fitness = Sqr(s)

End Function
```

Appendix D: Model Optimization based on Optimal Prediction

```
%---------------------------------------------------------------------------------------------------
%
%       this code is used for iimplementing optimization process for model optimization
%                       with LS algorithm for optimal prediciton output
%                                      written by Meng Tang
%                                           2004.6.10
% --------------------------------------------------------------------------------------------------

% load model output state data, weights for different state and data real process data
from file "StateData.txt"
format long
load StateData.txt
K=31;

StateData

for i=1:K

   Pn(i)=StateData(i, 1)*StateData(i,5);
   Pl(i)=StateData(i, 2)*StateData(i,4);
   Ph(i)=StateData(i, 3)*StateData(i,6);
   Pr(i)=StateData(i,8);

% construct the input matrix of the from observations
   if i==1
      B=[Pl(1) Pn(1) Ph(1)];
      A=B;
   end
   if i>=2
      B=[A; Pl(i) Pn(i) Ph(i)];
      A=B;
   end

% construct the output matrix of the model
  if i==1
     T=[Pr(1)];
     Y=T;
  end

  if i>=2
      T=[Y; Pr(i)];
      Y=T;

   end
```

```
end

A
Y
%---Parameter estimation of model with LS algorithms

H=(inv(A'*A))*A'*Y

%---the prediction output from the model------------

for i=1:K
   Pn(i)=StateData(i, 1)*StateData(i,5);
   Pl(i)=StateData(i, 2)*StateData(i,4);
   Ph(i)=StateData(i, 3)*StateData(i,6);

   if i==1
     Pm=H'*[Pl(1);Pn(1);Ph(1)];
  end
   if i>=2
      T=[Pm; H'*[Pl(i);Pn(i);Ph(i)]];
      Pm=T;

   end

end

Pm
```