Monitoring and Fault Diagnosis for Chylla-Haase Polymerization Reactor

Abdelkarim M Saed Ertiame

In partial fulfilment of the requirement of Liverpool John Moores University for the degree of Doctor in Philosophy

August 2016

Abstract

The main objective of this research is to develop a fault detection and isolation (FDI) methodologies for Cylla-Haase polymerization reactor, and implement the developed methods to the nonlinear simulation model of the proposed reactor to evaluate the effectiveness of FDI methods. The first part of this research focus of this chapter is to understand the nonlinear dynamic behaviour of the Chylla-Haase polymerization reactor. In this part, the mathematical model of the proposed reactor is described. The Simulink model of the proposed reactor is set up using Simulink/MATLAB. The design of Simulink model is developed based on a set of ordinary differential equations that describe the dynamic behaviour of the proposed polymerization reactor.

An independent radial basis function neural networks (RBFNN) are developed and employed here for an on-line diagnosis of actuator and sensor faults. In this research, a robust fault detection and isolation (FDI) scheme is developed for open-loop exothermic semi-batch polymerization reactor described by Chylla-Haase. The independent (RBFNN) is employed here when the system is subjected to system uncertainties and disturbances. Two different techniques to employ RBF neural networks are investigated. Firstly, an independent neural network is used to model the reactor dynamics and generate residuals. Secondly, an additional RBF neural network is developed as a classifier to isolate faults from the generated residuals.

In the third part of this research, a robust fault detection and isolation (FDI) scheme is developed to monitor the Chylla-Haase polymerization reactor, when it is under the cascade PI control. This part is really challenging task as the controller output cannot be designed when the reactor is under closed-loop control, and the control action will correct small changes of the states caused by faults. The proposed FDI strategy employed a radial basis function neural network (RBFNN) in an independent mode to model the process dynamics, and using the weighted sum-squared prediction error as the residual. The Recursive Orthogonal Least Squares algorithm (ROLS) is employed to train the model to overcome the training difficulty of the independent mode of the network. Then, another RBFNN is used as a fault classifier to isolate faults from different features involved in the residual vector.

In this research, an independent MLP neural network is implemented here to generate residuals for detection task. And another RBF is applied for isolation task performing as

a classifier. The fault diagnosis scheme is developed for a Chylla-Haase reactor under open-loop and closed-loop control system.

The comparison between these two neural network architectures (MPL and RBF) are shown that RBF configuration trained by (RLS) algorithm have several advantages. The first one is greater efficiency in finding optimal weights for field strength prediction in complex dynamic systems. The RBF configuration is less complex network that results in faster convergence. The training algorithms (RLs and ROLS) that used for training RBFNN in chapter (4) and (5) have proven to be efficient, which results in significant faster computer time in comparison to back-propagation one.

Another fault diagnosis (FD) scheme is developed in this research for an exothermic semi-batch polymerization reactor. The scheme includes two parts: the first part is to generate residual using an extended Kalman filter (EKF), and the second part is the decision making to report fault using a standardized hypothesis of statistical tests. The FD simulation results are presented to demonstrate the effectiveness of the proposed method.

In the lase section of this research, a robust fault diagnosis scheme for abrupt and incipient faults in nonlinear dynamic system. A general framework is developed for model-based fault detection and diagnosis using on-line approximators and adaptation/learning schemes. In this framework, neural network models constitute an important class of on-line approximators. The changes in the system dynamics due to fault are modelled as nonlinear functions of the state, while the time profile of the fault is assumed to be exponentially developing. The changes in the system dynamics are monitored by an on-line approximation model, which is used for detecting the failures. A systematic procedure for constructing nonlinear estimation algorithm is developed, and a stable learning scheme is derived using Lyapunov theory. Simulation studies are used to illustrate the results and to show the effectiveness of the fault diagnosis methodology.

Finally, the success of the proposed fault diagnosis methods illustrates the potential of the application of an independent RBFNN, an independent MLP, an Extended kalman filter and an adaptive nonlinear observer based FD, to chemical reactors.

Acknowledgment

First, I would like to express my deep gratitude to my supervisor Professor Dingli.Yu, for his help and supervision on this project. This work would not be possible without his invaluable suggestions and guidance. I truly appreciate his encouragement and helped me to expand my knowledge through self-learning. I am also grateful to my second supervisor for his support during this work. Dr. Barry.Gomm.

Finally, I would like to thank my family for their love and support. I wish to express my gratitude to my parents, whose prayer, love and encouragement have been a great support throughout my education. I am deeply indebted to my brothers and sisters whose positioned themselves into my deep heart by their exceptional support and very special thanks to all my friends.

Table of Contents

List of Figures

List of Tables

Glossary of Symbols

Nomenclature

η_k Standardized innovation sequence

μ Batch viscosity $(kg\,m^{-1}\,s^{-1})$

θ_1, θ_2 Transport delay (s)

τ_p Heating/Cooling time constant (s)

ρ_M, ρ_P, ρ_W Densities $(kg\,m^{-3})$

$\phi_j(t)$ RBFNN activation function (Gaussian function)

σ_j Width of Gaussian function

$\varepsilon(t)$ Sum-squared errors

μ_{wall} Wall viscosity $(kg\,m^{-1}\,s^{-1})$

γ_k Innovation residual of EKF

ΔH Heath enthalpy $(kJ\,kg^{-1})$

A Jacket heat transfer area (m^2)

B_1 Reactor bottom area (m^2)

B_2 Jacket bottom area (m^2)

$c_j(t)$ Centres of Gaussian function

$C_{p,M}, C_{p,P}, C_{p,C}$ specific heat at constant pressure $(kJ\,kg^{-1}\,K^{-1})$

$e(t)$ Error

E Activation reaction $(kJ\,kmol^{-1})$

E Activation energy in CSTR $(kcal/kgmol)$

f Solid mass friction

$\hat{f}(*)$ approximation function

$f(*)$ Process nonlinear function

F_k, H_k Jacobean matrices

F Volume flow rate in CSTR (m^3/h)

h heat transfer coefficient $(kW\,m^{-2}\,K^{-1})$

h_f^{-1} Fouling factor $(m^2\,K/kW)$

h_i Output of each hidden neuron of MLPNN

h_o Output of hidden layer of MLPNN

$h(*)$ Observation nonlinear function

H Reaction heat in CSTR $(kcal/kgmol)$

HA Overall heat time tank area $(kcal/h*K)$

HD Heat capacity time density in CSTR $(kcal/m^3*K)$

i Impurity factor

k Reaction rate

k Non-thermal factor in CSTR $(1/h)$

$k_0, k_1, a_0, c_0, c_1, c_2, c_3, d_0, d_1$ constants

$K_P(c)$ Heating/Cooling function

K_k Optimal EKF gain

m_M Monomer mass (kg)

m_P Polymer mass (kg)

$\dot{m}_M^{in}$ Monomer feed rate (kg/s)

n_h Hidden nodes number

P Reactor perimeter (m)

$P_{k|k-1}$ Predict estimate covariance

Q_{rea} Reaction heat (kW)

Q_k Diagonal process noise covariance matrix

R_p Polymerization rate (kg/s)

R Ideal gas constant $(kJ\,kmol^{-1}\,K^{-1})$

R Boltizmann's gas constant $(kcal/(kgmol*K))$

R_k Diagonal measurements noise covariance matrix

S_k Innovation covariance

$T(t)$ Reactor temperature (K)

$T_{jin}(t)$ Jacket input temperature (K)

$T_{jout}(t)$ Jacket output temperature (K)

T_j Average cooling jacket temperature (K)

T_{wall} Wall temperature (K)

T_{steam} Steam temperature (K)

T_{inlet} Inlet temperature (K)

$(UA)_{loss}$ Heat loss coefficient (kW/K)

U Overall heat transfer coefficient $(kW\,m^{-2}\,K^{-1})$

V Volume in CSTR (m^3)

v_{ji} Weight connected input vector with hidden layer inputs of MLPNN

V_k Measurements noise vector of EKF

w_{ji} Weights connected output layer with hidden layer

w_{jk} Weight connected output layer with hidden layer of MLPNN

W_k Process noise vector of EKF

$x(t)$ RBFNN input vector

$\hat{X}_{k|k-1}$ State prediction

X_k State vector of EKF

Y_k Observation vector of EKF

Abbreviations

ARIMA	Auto-regressive Integrated Moving Average
BPE	Backpropagation Error
BP	Backpropagation Algorithm
CSTR	Continues Stirred Tank Reactor
DMPL	Dynamic Multilayer Network
EKF	Extended Kalman Filter
FDI	Fault detection and Isolation
FD	Fault Detection
MLP	Multilayer Perceptron
MLPNN	Multilayer Perceptron Neural Network
MAE	Mean Absolute Error
MIMO	Multi-Input Multi-Output
NNs	Neural Networks
NNEKF	Neural Network Based Extended Kalman Filter
NARX	Nonlinear Autoregressive with Exogenous Inputs
OLA	Online Approximation Approach
ODE	Ordinary Differential Equation
PI	Proportional/integral Control
PCA	Principal Component Analysis

PLS	Partial Least Squares
REKF	Robust Extended Kalman Filter
RAS	Random Amplitude Signal
RBF	Radial Basis function
RBFNN	Radial Basis Function Neural Network
RLS	Recursive Least Squares Algorithm
RNN	Recurrent Neural Network
ROLS	Recursive Orthogonal Least Squares
SNN	Spiking Neural Network

Chapter 1

Introduction

1.1. Importance of Process Monitoring

In the recent years the task of monitoring the complex nonlinear process plants, have been intensively studied in process industry to detect faults. The fault detection and isolation (FDI) techniques are getting a lot of interest, because of the increasing demands for good performance and higher standards of safety and reliability of technical plants for improving the supervision and monitoring as part of the overall control of processes (Isermann, 1984, Isermann, 1993, Isermann, 1997, Gertler, 1988).

The fault detection and isolation has become a critical issue in the operation of high-performance chemical plants, nuclear plant, airplanes, ships, submarines, and space vehicles (Gertler, 1988, Isermann, 1997). In the chemical industry, fault can occur due to sensor failure, equipment failure or changes in process parameters. The occurrence of a fault may cause a process performance degradation (e.g., lower product quality), or in the worst cases, disastrous accidents, such as temperature runaway, which may require plant shut down for maintenance or will lead to break down the plant and even human fatalities. However, Fault detection and isolation (FDI) can help avoid all these major consequences (Deibert and Isermann, 1992, Isermann, 1984, Pierri et al., 2008). Deibert and Isermann (1992) illustrated that fault models can be divided into external faults: changes of power supply, contamination, collision, external disturbance, actuator faults: electric power failure, pomp failure and valve failure, process faults: abrupt variation and deviations in the process coefficients as heat transfer coefficient, and sensor faults.

FD system must avoid two kinds of errors, false alarms and missed alarms. A false alarm occurs when a fault is declared but the system is operating in healthy conditions; typically, they are due to model uncertainties and disturbances. On the other hand, a missed alarm occurs when under faulty condition, the FD system does not detect any fault. Usually, minimization of false alarm and missed alarms are conflicting requirements. Primary methods to fault diagnosis were often based on the so-called physical redundancy. The physical redundant methods are very reliable, but they need extra equipment and extra maintenance costs. For this reason, lots of research works have been carried out on techniques not requiring extra equipment. These techniques can be classified into two general categories, model free data-driven approaches and model-based approaches.

During the last decades theoretically and experimentally research has shown ways to detect and diagnose faults. One distinguish fault detection to recognize that the fault happened, fault diagnosis to find the cause and the location of fault.

Isermann (1997) illustrated that, the advanced methods of supervision and fault diagnosis are needed, which satisfy the following requirements:

(i) Early detection of small faults with abrupt or incipient time behaviour.

(ii) Diagnosis of faults in the actuator, process components or sensors.

(iii) Detection of faults in closed loops.

(iv)Supervision of processes in transient states.

The goal for the early detection and diagnosis is to have enough time for counteractions such as other operations, reconfiguration, maintenance or repair. The earlier detection can be achieved by gathering more information, especially by using the relationship between the measurable quantities in the form of mathematical models (Isermann, 1997).

1.2. Aims and Objectives

The aim of this project is to develop new fault detection and isolation (FDI) methodologies for nonlinear processes, and implement the developed methods to the nonlinear simulation model of the Chylla-Haase reactor to evaluate the effectiveness of the developed methods.

In order to achieve these aims, the research is designed with the following objectives:

- Develop and evaluate FDI method for open-loop reactor using an independent radial basis function (RBF) neural network and evaluate it on the Simulink model.
- Develop and evaluate FDI method for closed-loop reactor using an independent radial basis function (RBF) neural network and evaluate it on the Simulink model.
- Develop and evaluate FDI method for open-loop and closed-loop reactor using an independent multilayer perceptron (MLP) neural network and evaluate it on the Simulink model.
- Develop and evaluate FD method for reactor using extended Kalman filter and evaluate it on the Simulink model.
- Develop and evaluate an adaptive nonlinear observer based fault detection using a learning methodology and evaluate it on Simulink/Matlab model.

1.3. Thesis outline

The thesis is organized into nine chapters. Chapter one is an introduction chapter which gives an overview of the conducted work. It explains the motivation behind this research and the importance of monitoring the complex high nonlinear process systems. It also states the aims and objectives of the research. Chapter two reviews the cumulative research works that have been carried out over the last two decades on monitoring of chemical reactors. Chapter three gives a wide description of the process modelling and simulation. The main aim of this chapter is to understand the nonlinear dynamic

behaviour of the Chylla-Haase reactor. In this chapter, the mathematical model of the proposed reactor is described. The Simulink model of the proposed reactor is set up using Simulink/MATLAB. The Simulink block diagrams and the performances of the reactor are presented and discussed in details.

Chapter four illustrates the design and development of FDI method for open-loop Chylla-Hasse system using an independent radial basis function (RBF) neural network. In this chapter the investigation of employing two different techniques of RBFNN is described. Firstly, an independent RBFNN is employed to model the reactor dynamics and generate residuals for the detection part. Secondly, an additional RBFNN is designed as a classifier to perform the isolation task. The simulation performances are presented to demonstrate the effectiveness of the proposed techniques.

Chapter five investigates the dynamic fault detection and isolation for Chylla-Haase reactor under closed-loop control. In this chapter a cascade PI controller is designed. An independent RBF network is employed to model the process dynamics and generate residuals. The Recursive Orthogonal Least Squares algorithm (ROLS) is used to train the independent mode of the network. An additional RBFFNN is developed to isolate faults. The simulation results are presented.. Chapter six describes the design scheme of FDI for the proposed reactor in open-loop and closed-loop mode using an independent multilayer perceptron (MLP) neural network. In this chapter an independent MLP is employed to perform detection task, and another RBFNN is employed as a classifier for isolate faults. The simulation results are presented and discussed.

Chapter seven describes a fault detection (FD) scheme for the proposed reactor. In this chapter an Extended Kalman Filter (EKF) is designed and developed for online state and parameter estimation. Also a hypothesis testing is employed for fault detection. The simulation results are discussed and presented. Chapter eight illustrates the development

of an adaptive nonlinear observer based fault detection in nonlinear multivariable system using a learning methodology. A general framework is developed for model-based fault detection and diagnosis using on-line approximators and adaptation/learning schemes. In this chapter the proposed method is applied for CSTR reactor. Simulation results are presented to illustrate the effectiveness of the fault diagnosis methodology. Chapter nine gives a summary of main contributions and achievements of the conducted work.

1.4. Research Novelty and Originality

This research work will be focused on studying and developing a monitoring system for Chylla-Haase reactor, which is used as a benchmark problem in chemical industry. The main contribution of this research lies behind the fact that, there are no existing investigations into the FDI for the reactor. In addition, there are few papers dealing with FDI based on closed –loop performance for chemical reactors. The novelty and unique contribution of this research to knowledge is divided into four sections. First section will be focus on developing a new FDI method for open-loop and closed-loop reactor using an independent RBFNN, which will be a new contribution to knowledge. The second section will be focus on developing a new FDI method for open-loop and closed-loop reactor using an independent MLPNN, which will be a new contribution to knowledge. The third section is to develop a new FD method for reactor using EKF to against disturbances, which also will be a major challenge and a new contribution to knowledge. Finally, developing and designing an adaptive nonlinear observer based fault detection for reactor using a learning methodology is a new contribution to knowledge. These proposed methods are robust against the disturbances and can also cope with high nonlinearities of the reactor. The application of all proposed fault detection and isolation strategies for monitoring reactor. Thus the originality of the proposed research stands.

Chapter 2

Literature review

2.1. Introduction

Over the last two decades fault detection and isolation techniques have been widely used in chemical process industry to detect faults in actuators and sensors. Deibert and Isermann (1992) described that, in the chemical industry, fault can take place in the system, as result of sensors failures, equipment failures or changes in process parameters. The existence of a fault may cause a process performance degradation (e.g., lower product quality), or in the worst cases, disastrous accidents, such as run-away. Fault models can be divided into external faults: changes of power supply, contamination, collision, external disturbance, actuator faults: electric power failure, pomp failure and valve failure, process faults: abrupt variation and deviations in the process coefficients as heat transfer coefficient, and sensor faults.

Frank (1996) and, Frank and Köppen-Seliger (1997) describe that, in the chemical process faults can be classified in process faults, sensor faults and actuators. A graphical diagram of Faults is shown in Figure 2.1.

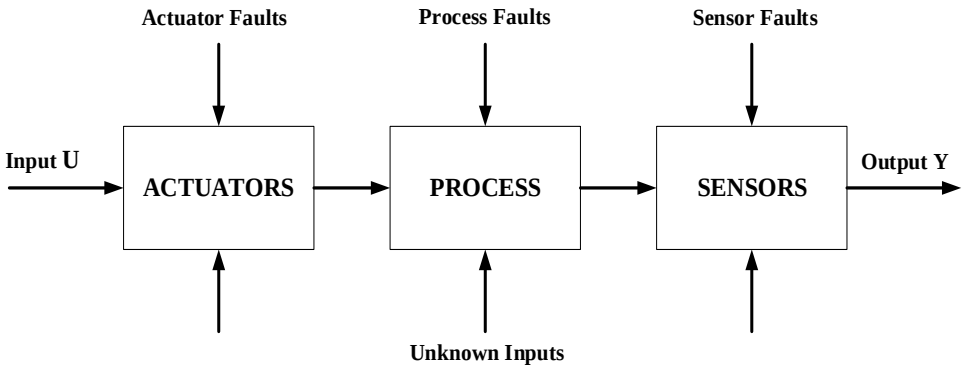


Figure 2.1 Fault classifications (Frank, et.al, 1997)

2.2. Fault Diagnosis Strategies

According to Fabrizio Caccavale (2011), Gertler (1988), Frank (1990), Isermann (1984) various types of failures may affect the safety, reliability and efficiency in chemical processes. The existence of faults may affect productivity of the process or, in the worst circumstances, may cause serious accidents. For that reason, fault detection and diagnosis has been widely studied in the recent years. The term fault is generally defined as a departure of an observed variable or a Parameter from an acceptable range. The causes of this abnormality, such as a failed coolant pump or a failed sensor, are called basic events or root events and are often referred as malfunctions or failures.

Fabrizio Caccavale (2011), Isermann (1997) and Venkatasubramanian et al. (2003d) explain that, fault diagnosis (FD) consists of three main tasks:

- Fault detection, i.e., the detection of the occurrence of a fault
- Fault isolation, i.e., the determination of the type and/or location of the fault; and
- Fault identification, i.e., the determination of the time evolution of the fault.

Fabrizio Caccavale (2011) illustrated that, FD system must avoid two kinds of errors, false alarms and missed alarms. A false alarm occurs when a fault is declared but the system is operating in healthy conditions; typically, they are due to model uncertainties and disturbances. On the other hand, a missed alarm occurs when, under faulty condition, the FD system does not detect any fault. Usually, minimization of false alarm and missed alarms are conflicting requirements. Early approaches to fault diagnosis were often based on the so-called physical redundancy, i.e., the duplication of sensors, actuators, computers, and software's to measure and/or control a variable. Typically, a voting scheme is applied to the redundant system to detect and isolate a fault. The physical redundant methods are very reliable, but they need extra equipment and extra maintenance costs. For this reason,

lots of research works have been carried out on techniques not requiring extra equipment. These techniques can be classified into two general categories, model free data-driven approaches and model-based approaches.

The classification of faults have been demonstrated by Isermann (1984), Patton and Chen (1992a), Venkatasubramanian et al. (2003b), Gertler (1988), Isermann (1997) as following:

2.2.1. Model free approaches

The classification of model free approaches is illustrated in figure 2.2. In contrast to the model-based approaches where a priori knowledge (either quantitative or qualitative) about the process is needed, in process free (history) based methods, only the availability of large amount of historical process data is needed. There are different ways in which this data can be transformed and presented as a priori knowledge to a diagnostic system. This is known as feature extraction. This extraction process can be either qualitative or quantitative in nature. Two of the major methods that extract qualitative history information are the expert systems and trend modelling methods. Methods that extract quantitative information can be broadly classified as non-statistical or statistical methods. Neural networks are an important class of non-statistical classifiers. Principal component analysis (PCA)/partial least squares (PLS) and statistical pattern classifiers form a major component of statistical feature extraction methods.(Gertler, 1988, Isermann, 1984, Isermann, 1997, Venkatasubramanian et al., 2003b, Patton and Chen, 1992b).

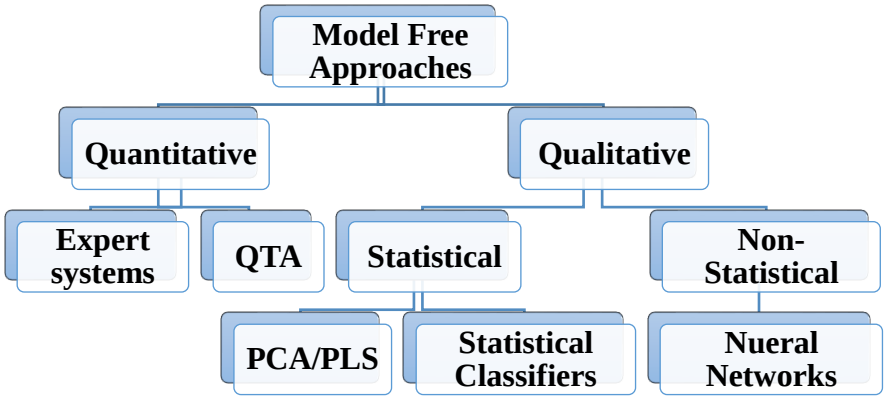


Figure 2.2 Classification of process history methods (Venkatasubramanian et al.,2003).

2.2.2. Model based approaches

Model-based approaches to fault diagnosis can be divided into qualitative methods and quantitative methods as shown in figure 2.3. In the recent years many research works have been carried out and focused mainly on quantitative model-based methods. Patton and Chen (1997), Venkatasubramanian et al. (2003b), Patton (1997) explain that in the following figure 2.3, model based methods based on the concept of analytical or functional redundancy, which use a mathematical model of the process to obtain the estimates of a set of variables characterizing the behaviour of the monitored system. The inconsistencies between estimated and measured variables provide a set of residuals, sensitive to the occurrence of faults. Later, the residuals are evaluated in order to identify and localize faults. Although there is a close relationship among the various quantitative model based techniques, observer-based approaches have become very important and diffused, especially within the automatic control community. Luenberger observers, unknown input observers, and Extended Kalman Filters have been mostly used in fault detection and identification for chemical processes and plants.

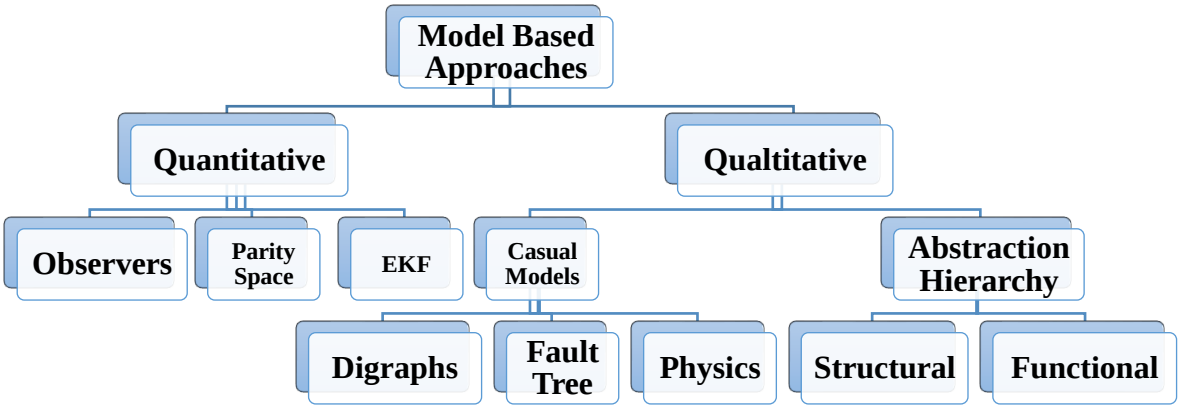


Figure 2.3 Classification of model based methods (Venkatasubramanian et al.,2003).

Deibert and Isermann (1992) explain that, using model based methods for fault detection in control loop have some advantages, first advantage is obtaining much deeper diagnosis as standard limit, and the second advantage is it can be use modelling once for controller design and fault diagnosis. Isermann (2005) illustrate that, Process model-based methods require the knowledge of a usually dynamic process model in form of a mathematical structure and parameters. For linear processes in continuous time the models can be impulse responses (weighting functions), differential equations of frequency responses. Corresponding models for discrete-time (after sampling) are impulse responses, difference equations or z-transfer functions. For fault-detection in general differential equations or difference equations are primarily suitable. In most practical cases the process parameters are partially not known or not known at all. Then, they can be determined with parameter estimation methods by measuring input and output signals if the basic model structure is known. Deibert and Isermann (1992) demonstrate that, most of the state space approaches yields the information about faults via so called residuals. The only difference is how to design the observer feedback matrix H and the weighting

matrix W. In the fault free case the residual equals zero, and if a fault occurs, the residual deviates from zero in a matter which is typical for the specific algorithm used. The figure below shows the common block diagram. State space approaches which are sensitive for sensor faults in the sense of having influence on the C-Matrix are not useful for control loops, because the C-Matrix of the sensor subsystem occurs in both A-Matrix and C-Matrix of the entire control loop representation. Isermann (2005) explain that, if the process parameters are known, either state observers or output observers can be applied. Frank (1996), Patton and Chen (1997), Frank (1990) and Patton (1997) explained that, the unknown input observer can be derived through the generalised Luenberger observer. The main goal of the unknown input observer is to force each of the state estimation error to become independent of the uncertainty. Once, the estimation error vector is de-coupled from the uncertainty, the residual will also be de-coupled from uncertainty.

Venkatasubramanian et al. (2003d) explain that, the plant disturbances are random fluctuations and oftentimes only their statistical parameters are known. One solution to the fault diagnosis problem in such systems entails monitoring the innovation process or the prediction errors. The objective is to design a state estimator with minimum estimation error. It involves the use of optimal state estimate, e.g. the Kalman filter, which is designed on the basis of the system model in its normal operating mode. It is well known that the Kalman filter is a recursive algorithm for state estimation and it has found wide applications in chemical as well as other industrial processes.

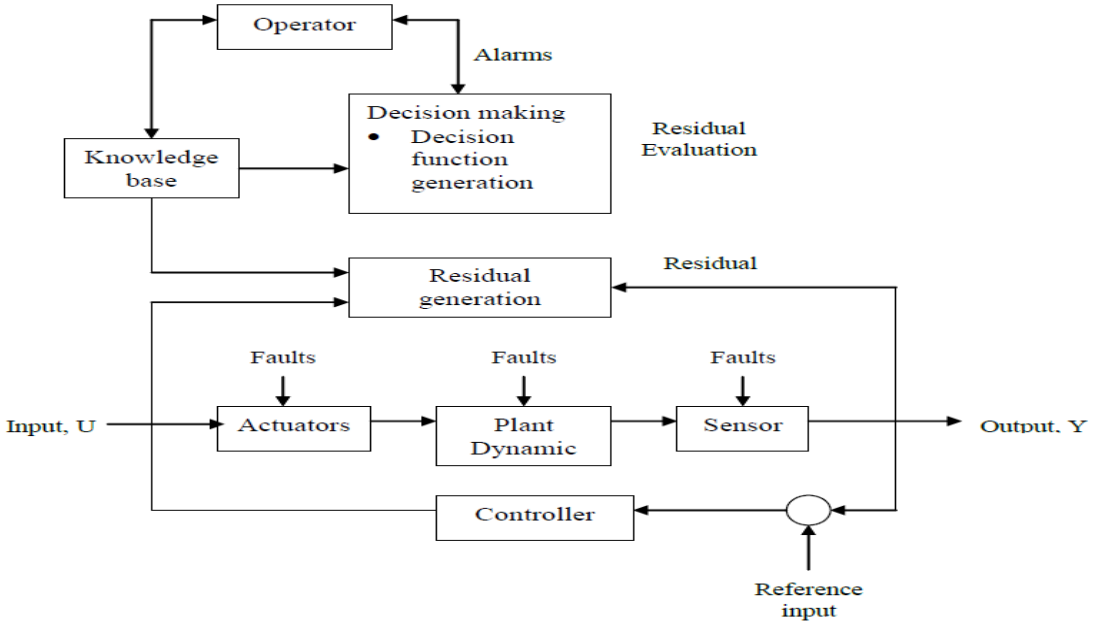


Figure 2.4 General Scheme of model based FD (Patton, 1997).

2.3. RBF Model based FDI

In recent years, the task of monitoring complex nonlinear processes has been intensively studied. Fault detection and isolation (FDI) techniques have attracted much interest due to the increasing demand for good performance and higher standards of safety and reliability of technical plants for improving the supervision and monitoring as part of the overall control of processes (Isermann, 1984). FDI has become a critical issue in the operation of high-performance chemical plants, nuclear plants, airplanes, ships, submarines, and space vehicles, etc. (Gertler, 1988). In the chemical industry, faults can occur due to sensor failures, equipment failures or changes in process parameters. Occurrence of a fault may cause process performance degradation, or in the worst cases, may cause disastrous accidents such as temperature runaway, which may require plant shut down for maintenance to prevent break down of the plant and perhaps even human fatalities. However, early detection of faults can help avoid all these major consequences (Deibert and Isermann, 1992, Pierri et al., 2008, Wang et al., 2006).

Fabrizio Caccavale et al. (2011) illustrated that, FD system must avoid two kinds of errors, false alarms and missed alarms. A false alarm occurs when a fault is declared but the system is operating in healthy conditions; typically, they are due to model uncertainties and disturbances. On the other hand, a missed alarm occurs when under faulty condition, the FD system does not detect any fault. Usually, minimization of false alarm and missed alarms are conflicting requirements. Primary methods to fault diagnosis were often based on the so-called physical redundancy. The physical redundant methods are very reliable, but they need extra equipment and extra maintenance costs. For this reason, lots of research works have been carried out on techniques not requiring extra equipment. These techniques can be classified into two general categories, model free data-driven approaches and model-based approaches (Patton and Chen, 1992b, Venkatasubramanian et al., 2003c).

Due to severe nonlinearity and time varying feature of the reactor dynamics, the observer methods, parity space methods, and other first-principle model-based methods cannot be successfully applied for FDI of the Chylla-Haase reactor. The application of neural networks (NN) for FDI has been intensively studied over the last two decades. Patton et al. (1994) proposed an approach for detecting and isolating faults in a non-linear dynamic process using neural networks. Firstly, a multi-layer perceptron (MLP) network was trained to predict the future system states, and then the residual was generated using the differences between the actual and predicted states. Secondly, another neural network was used as a classifier to isolate faults from these state prediction errors. However, this method used the neural network model in its so-called dependent mode.

Many research works have been carried out to study NNs for FDI. Yu et al. (1999) studied sensor fault diagnosis in chemical process via RBF neural networks; a semi-independent NN was used for sensor fault diagnosis. Moreover, the thins-plate-spline function was

used for the neural model and the Gaussian function was used for the neural classifier. Another study was conducted by Gomm and Yu (2000) that introduced the selection of radial basis function (RBF) network centres with recursive orthogonal least squares training. Frank and Köppen-Seliger (1997), Koppen-Seliger and Frank (1995) studied fuzzy logic and neural network applications for fault diagnosis. Their paper introduced fuzzy logic for residual evaluation, a dependent neural network for residual generation, and a neural network for residual evaluation by using another dependent neural network for generating residuals. All those authors used dependent and semi-dependent mode of NN for FDI. As the residual of these methods is affected by the plant output, the residual is made insensitive to the faults. Although a partial dependent mode is used to enhance the residual to fault sensitivity, the fault detect threshold is still high such that fault with small amplitude cannot be detected.

Ferrari et al. (2008), Xiaodong (2011), Xiaodong et al. (2002), Zhang et al. (2010) studied the design and analysis of a robust fault detection and isolation scheme for nonlinear uncertain dynamic systems, the proposed architecture consists of a bank of nonlinear adaptive estimator, one of the estimators is used for the detection and approximation of a fault, whereas the rest are used for online fault isolation decision scheme is based on adaptive threshold functions. In their method they used state space nonlinear model and then used a simple NN as an estimator for online learning the system output to be equal to the plant output, however this method needs to have plant nonlinear model and sometimes model need to be very accurate, this accurate model is difficult to produce. These methods may not be applicable to some industrial plants where accurate analytical models are difficult to derive and the physical parameters are not all available, for example, the chemical reactors. However, in their method they have used a dependent

mode of RBFNN which is performs as one-step ahead prediction and cannot run independently of the process.

In the first part of this research, a new robust FDI scheme is developed for open-loop Chylla-Haase polymerization reactor using an independent RBFNN. The independent RBFNN is employed here for on-line diagnosis of faults on the actuator and sensors when the system is subjected to system uncertainties and disturbances. The independent neural network mode is developed to generate enhanced residuals for diagnosing faults in the reactor. Then, a second neural network is developed as a classifier to isolate these faults. The basis Gaussian function is used for the neural network model, and for the neural network classifier. The K-means clustering algorithm is used to choose the centres of the RBF networks, and a p-nearest-neighbours algorithm is used to choose the widths. Moreover, a recursive least squares (RLS) algorithm is used to update the weights. Most of the recent investigations of fault diagnosis for chemical reactors using an independent RBF neural networks have been studied by (Ertiame et al., 2013).

Most of the previous research studied a dependent RBFNN based FDI for open-loop systems. In contrast to develop FDI methods for open loop system, the second part of this research will be focused on developing a new robust FDI scheme for the Chylla-Haase polymerization reactor that is under cascade PI control. An independent RBFNN is employed to predict the process output on-line and consequently to generate the residual. Then, a second neural network is used as a classifier to isolate these faults. The Gaussian function is used for the neural network model and the classifier as the nonlinear basis function. The K-means clustering algorithm is used to choose the centres for the RBF networks, and a P-nearest-neighbours algorithm is used to choose the widths. Moreover, a recursive orthogonal least squares (ROLS) algorithm is used to train the weights.

2.4. MLP model based FDI

In contrast to the model-based approaches where a priori knowledge about the model (either quantitative or qualitative) of the process is assumed, in process history based methods only the availability of large amount of historical process data is assumed There are different ways in which this data can be transformed and presented as a priori knowledge to a diagnostic system. This is known as the feature extraction process from the process history data, and is done to facilitate later diagnosis. This extraction process can mainly proceed as either quantitative or qualitative feature extraction. In quantitative feature extraction one can perform either a statistical or non-statistical feature extraction. (Venkatasubramanian et al., 2003b, Willsky, 1976, Frank, 1996, Isermann, 1997, Isermann, 1984).

The literature presents several classes of strategies to deal with fault detection and isolation. These strategies, in general, can be divided into two kind of approaches (i) qualitative and (ii) quantitative. In this section we focus mainly on diagnostic systems that are built on non-statistical feature extraction quantitative model known as Multilayer Perceptron Neural Networks (MLP NNs). The requirement of a mathematical model of the plant can lead to several difficulties in the implementation of these approaches, for instance due to factors such as system complexity, high dimensionality, nonlinearities and parametric uncertainties. Further, in the case the neural network plays a role as an observer, it falls into the class of quantitative approaches.

Neural networks have been proposed for classification and function approximation problems. In general, neural networks that have been used for fault diagnosis can be classified along two dimensions: (i) the architecture of the network such as sigmoidal, radial basis and so on; and (ii) the learning strategy such as supervised and unsupervised learning. Different network architectures have been used for the problem of fault

diagnosis (Venkatasubramanian et al., 2003b). In supervised learning strategies, by choosing a specific topology for the neural network, the network is parameterized in the sense that the problem at hand is reduced to the estimation of the connection weights. The connection weights are learned by explicitly utilizing the mismatch between the desired and actual values to guide the search. This makes supervised neural networks a good choice for fault classification as the networks are capable of generating, hence classifying, arbitrary regions in space (Venkatasubramanian et al., 2003b). On the other end of the spectrum are neural network architectures which utilize unsupervised estimation techniques. These networks are popularly known as self-organizing neural networks as the structure is adaptively determined based on the input to the network. The most popular supervised learning strategy in neural networks has been the back-propagation algorithm. During the past two decades there are many researchers have addressed the problem of fault detection and diagnosis using multilayer perceptron (MLP) neural networks. Mrugalski and Korbicz (2007) studied in their work the application of MLP neural networks to the robust fault detection. Another study has been conducted by Maki and Loparo (1997), in their study, the multilayer feedforward neural network that has one hidden layer was used, A two-stage neural network was proposed as the basic structure of the detection system. The first stage of the network detects the dynamic trend of each measurement, and the second stage of the network detects and diagnoses the faults. Akhoondzadeh (2013) investigated the Total Electron Content (TEC) time series by using a Multi-Layer Perceptron (MLP) neural network to detect seismo-ionospheric anomalous variations induced by the powerful Tohoku earthquake of March 11, 2011. The results show that the MLP presents anomalies better than referenced and conventional methods such as Auto-Regressive Integrated Moving Average (ARIMA) technique. Young-Moon et al. (1996) used feedforward neural networks are used to solve an optimal tracking

control problem for discrete-time nonlinear dynamic systems. Two multilayer neural networks were constructed as the feedforward and the feedback controllers. The feedback controller is trained by Backpropagation through algorithm to minimize a general quadratic cost function. The proposed methodology was useful as an off-line control method. Another study conducted by Parlos et al. (1994), A nonlinear dynamic model was developed for a process system, namely a heat exchanger, using the recurrent multilayer perceptron network. A dynamic gradient descent learning algorithm is used to train the recurrent multilayer perceptron, resulting in an order of magnitude improvement in convergence speed over a static learning algorithm used to train the same network. In developing the empirical process model the effects of actuator, process, and sensor noise on the training and testing sets are investigated. Johnson et al. (2009) studied the application of a spiking neural network (SNN) and a multi-layer perceptron (MLP) for online identification of generator dynamics in a multi-machine power system. Jung-Wook et al. (2002) studied the performances of a multilayer perceptron network (MLPN) and a radial basis function network (RBFN) were compared, for the on-line identification of the nonlinear dynamics of a synchronous generator.(Mahmud et al., 2014) investigated multi-layered perceptron (MLP) network using various types of training algorithms for fault classification in extra high voltage (EHV) transmission lines. The performance of the suitable training algorithm in MLP network resulted the highest accuracy for fault classification. Dash et al. (2010) studied the application of MLP NN techniques for the detection of stator inter-turn fault of an induction motor. Clark and Warwick (1995) considered a multilayer perceptron (MLP) neural network for detection of faults in a high speed packaging machine.Wen et al. (2000) proposed a stable learning law of the dynamic multilayer neural Networks (DMPL). A Lyapunov-like analysis is used to derive this stable learning procedure for the hidden layer as well as for the output layer. An algebraic

Riccati equation is considered to construct a bound for the identification error. The suggested learning algorithm is similar to the well-known backpropagation rule of the static multilayer perceptron. Gomm et al. (1996) described two methods for representing data in a multi-layer perceptron (MLP) neural network, and the resultant ability of networks, trained by the standard back-propagation algorithm, to identify the dynamics of non-linear systems was investigated. Souahlia et al. (2012) discussed MLP neural network-based decision for power transformers fault diagnosis using an improved combination of Rogers and Doernenburg ratios DGA. Another study conducted by Golovko et al. (2001) Modelling nonlinear dynamic using multilayer neural Networks, Proposed method provides the calculation of Lyapunov exponents using multilayer neural networks trained by modified backpropagation error (BPE) algorithm.(Pandey and Barai, 1995) presented an application of multilayer perceptron in the damage detection of steel bridge structures, the issues relating to the design of network and learning paradigm are addressed and network architectures have been developed with reference to trussed bridge structures. The training patterns are generated for multiple damaged zones in a structure and performance of the networks with one and two hidden layers were examined.

Most of the previous mentioned approaches studied a dependent MLPNN based FD for open-loop systems. Whereas, in this research work a new FDI approach is developed for open-loop and closed-loop systems using an independent mode of MLPNN.

The comparison between using RBFNN and MLPNN for FDI is discussed in more details in chapter 6.

2.5. Extended Kalman filter based FD

Batch and semi-batch reactors are widely used in chemical industry for the production of fine chemicals, pigment, polymers, and pharmaceuticals. The dynamics of these reactors are nonlinear in nature and this makes them are very difficult to control and monitoring

(Gertler, 1988, Isermann, 1984) . In chemical processes, different types of failure may cause safety and productivity problems. Deibert and Isermann (1992) Have illustrated that fault models can be divided into external faults: changes of power supply, contamination, collision, external disturbance; actuator faults: electric power failure, pump failure and valve failure; and process faults: abrupt variation and deviations in the process coefficients such as heat transfer coefficient and sensor faults.

In the recent years monitoring and fault detection for complex nonlinear processes have been intensively studied. The early approaches to fault diagnosis were based on so called physical redundancy. Despite the reliability of using the physical redundancy method, it requires extra equipment and extra maintenance costs. Isermann (1984) and Venkatasubramanian et al. (2003b) have classified the fault diagnosis techniques that don't required extra equipment into two general categories, model free data-driven approaches and model-based approaches(Ertiame et al., 2013, Ertiame et al., 2015, Ertiame, 2015).

In the first category, the methods require the availability of large amount of historical process data. There are different ways in which this data can be transformed and presented as a priori knowledge to a diagnostic system. This is known as feature extraction. This extraction process can be either qualitative or quantitative in nature. Two of the major methods that extract qualitative history information are the expert systems and trend modelling methods. Methods that extract quantitative information can be broadly classified as non-statistical or statistical methods. Neural networks are an important class of non-statistical classifiers. Principal component analysis (PCA)/partial least squares (PLS) and statistical pattern classifiers form a major component of statistical feature extraction methods (Patton and Chen, 1992a, Venkatasubramanian et al., 2003b, Isermann, 1984, Isermann, 1997). Many research works have been carried out to study

model free data-driven approaches for FDI (Gomm and Yu, 2000, Barton and Himmelblau, 1997, Patton et al., 1994, Frank and Köppen-Seliger, 1997, Yu et al., 1999, Zhou et al., 2003, Ertiame et al., 2013, Ertiame et al., 2015, Ertiame, 2015) .

The second category referred to model-based approaches. These approaches can be divided into qualitative and quantitative. Qualitative methods are divided into casual models and abstraction hierarchy. In addition, the quantitative methods are classified into observers, parity space, and extended Kalman filter (EKF) (Patton and Chen, 1992a, Venkatasubramanian et al., 2003b, Isermann, 1984).

In the recent years, the EKF has been intensively for state and parameter estimation. Many research works have been carried out to study EKF for fault detection for chemical processes. Menaa et al. (2003) studied the estimation of the rotor resistance in induction motor by application of the spiral vector theory associate to extended Kalman filter. Ouhrouche et al. (1998) presented the application of an extended Kalman filter to rotor speed and resistance estimation in induction motor vector control. Loron and Laliberte (1993) studied the application of the extended Kalman filter to parameters estimation of induction motors, in their paper the extended Kalman filter was used as a parameter estimator for the tuning of the indirect field-oriented controller. Another study carried out by Graichen et al. (2005a) presented an adaptive feedforward Control with Parameter Estimation for the Chylla-Haase Polymerization Reactor, an extended Kalman filter is designed to estimate the reaction heat and the heat transfer coefficient during polymerization. Wei and Yang (2011) studied the localization of a mobile robot based on neural network based extended Kalman filter (NNEKF) algorithm. Extended Kalman filter (EKF) is used to fuse the information acquired from both the robot optical encoders and ultrasonic sensors in order to estimate the current robot position and orientation. Then the error covariance of the EKF is tracked by the covariance matching technique.

Fu et al. (2015) employed an adaptive extended kalman filter for navigation system based on red shift for spacecraft mission in solar system. Another study was conducted by Kai et al. (2010) presented a novel robust extended kalman filter (REKF) for discrete-time nonlinear systems with stochastic uncertainties is proposed. The filter is derived to guarantee an optimized upper bound on the state estimation error covariance despite the model uncertainties as well as the linearization errors. The method was applied in an X-ray pulsar positioning system. Khanesar et al. (2012) presented a method of using extended Kalman filter for the optimization of the parameters of type-2 fuzzy logic systems. The extended Kalman filter was shown a better performance as compared to the gradient descent-based methods and particle swarm optimization method.Jassemi-Zargani and Necsulescu (2002) studied extended Kalman filter-based sensor fusion for operational space control of a robot arm. Senjyu et al. (2003) presented a high efficiency control of synchronous reluctance motors using extended Kalman filter. He et al. (2015) presented a model-based fault diagnosis scheme to detect and isolate the faults of the current and voltage sensors applied in the series Lithium-Ion battery pack based on an adaptive extended kalman filter. Hatami et al. (2014) designed of a fault tolerated intelligent control system for a nuclear reactor power control by using extended Kalman filter. Salahshoor and Mosallaei (2008) proposed a model-based process fault monitoring approach which utilizes a multi-sensor data fusion technique. The fusion algorithm is based on a discrete-time extended Kalman filter (EKF). The presented EKF was modified to incorporate the asynchronous sensor measurements. Liu (1999) presented an extended Kalman filter and neural network cascade fault diagnosis strategy for the glutamic acid fermentation process. Dalle Molle and Himmelblau (1987) Studied fault detection in a single-stage evaporator via parameter estimation using Kalman filter. Another study was conducted by Chetouani (2004) that introduced fault detection method based on statistical

information generated by EKF. De Vallie`re and Bonvin (1989) Studied the estimation of states and parameters of batch reactor using EKF. Benkouider et al. (2009) proposed an approach for fault detection in semi-batch and batch reactor based on statistical approach test and discrete extended Kalman filter with parameter estimation. Another study carried out by Benkouider et al. ((2009) introduced a hybrid approach for the detection and isolation of faults in semi-batch and batch reactors based on statistical test using extended Kalman filter and neural network for the diagnosis part. Li and Olson (1991) Developed fault detection method in a closed-loop nonlinear distillation process using EKF, where the EKF is applied inside the control loop. Walker and Huang (1995) Studied FDI using extended Kalman filter for parameter estimation of an industrial actuator benchmark. Mehra and Peschon (1971) Proposed a method for fault detection in dynamic systems using statistical test decision theory based Kalman filter.

The approaches as stated above are not fit for use in high nonlinear processes such a Chylla-Haase polymerization reactor because they cannot meet the requirements for (i) sensitivity to incipient failure and (ii) robustness to model uncertainties in maintaining low false rates. In this research the FDI scheme is developed for Chylla-Haase polymerization reactor using EKF. The idea of using the proposed approach is to estimate on-line the states. Then a standardized innovation sequence for the standardized hypothesis of statistical tests is used for fault detection. Therefore, two hypotheses are defined; the first one is the hypothesis H0 referred to the innovation statistics in the normal mode, the second one is the hypothesis H1 referred to an abnormal mode.

2.6. Nonlinear observer based FD

With associate increasing demand for higher performance moreover as for a lot of safety and reliability of dynamic systems, fault diagnosis has received a lot of attention. The matter of on-line fault detection and isolation has become a serious issue in chemical

engineering. Several fault diagnosis (FD) approaches have been proposed for processes operating mainly in steady-state conditions e.g., continuous reactors. Due to the high nonlinear dynamics and unsteady operating conditions of batch chemical systems, the application of these techniques are very challenging task to implement. Moreover, the full state measurements and an accurate knowledge of parameters of batch reactors are hardly available. Existing fault diagnosis approaches for chemical processes can be roughly classified in model-free approaches i.e., approaches based on statistical analysis, neural networks or expert systems and model-based approaches e.g., observer-based techniques. Model-free approaches do not require a model of the system but only a database of historical data collected in normal operating conditions (Caccavale et al., 2009).

The traditional engineering approach to achieving fault in dynamical systems is through the use of *hardware redundancy.* This approach corresponds to constructing redundant physical subsystems. However, often times the additional cost, space and/or complexity of incorporating redundant hardware makes this approach unattractive. Most of the current research in FDA is based on the use of *analytical redundancy.*

In the last two decades numerous approaches to FDA have been intensively studied using analytical redundancy. Some of these approaches can be categorized as following: the detection filter; the innovation test, the parity space approach; and the parameter estimation technique. The derivation of an accurate mathematical model of the physical system is believed to be one of major issues in applying analytical redundancy approaches to FDA (Demetriou and Polycarpou, 1998, Polycarpou and Helmicki, 1995, Trunov and Polycarpou, 2000, Vemuri and Polycarpou, 1997, Xiaodong et al., 2002).

The most commonly used quantitative model-based FDI methods are: analytical redundancy, diagnostic observers, parity relations, Kalman filters and parameter estimation. One of the major advantages of using the quantitative model-based approach

is that we will have some control over the behaviour of the residuals. However, several factors such as system complexity, high dimensionality, process nonlinearity And/or lack of good data often render it very difficult even impractical, to develop an accurate mathematical model for the system. This, of course, limits the usefulness of this approach in real industrial processes. The evaluation of residuals usually involves threshold testing. Statistical tests have been utilized for residuals generated from parity relation as well as observer-based designs (Venkatasubramanian et al., 2003d).

Various approaches to FDA using analytical redundancy have been studied during last two decades. Most of these results can be categorized based on the use of a few basic concepts, such as: linear observers (Corradini et al., 2012, de Lira et al., 2011, Pierri and Paviglianiti, 2007, Pierri et al., 2008, Wang et al., 2015, Zhang et al., 2016); the detection filter (Iftikhar et al., 2015, Wang and Shang, 2015, Chen et al., 2007, Zhuang et al., 2014); the parity space approach (Zhong et al., 2015, Odendaal and Jones, 2014, Zhang et al., 2006, Naik et al., 2009, Medvedev, 1995, Kabbaj et al., 2009); and parameter estimation technique(Gertler, 1997). For more details on the general FDA problem we refer to comprehensive survey articles by (Gertler, 1988, Patton and Chen, 1992b, Frank and Ding, 1997, Willsky, 1976, Venkatasubramanian et al., 2003a).

It is very interesting to notice that in practice, instead of residuals, output signals of the process under consideration are often directly evaluated and compared with a given threshold. In the analytical observer-based approach, the generation of residuals reflecting the faults is done by estimating outputs of the process and using the estimation errors as the residuals. For the fault detection task, a single observer or Kalman filter is sufficient whereas, for the localization of the faults, properly structured sets of residuals are required. The latter can be generated by using banks of the observers, so-called dedicated and generalized observer schemes (DOS and GOS). Depending on the circumstances, one

may use linear or nonlinear, full or reduced-order, or fixed or adaptive observers(Frank and Ding, 1997).

Robustness is a vital task in control and monitoring of dynamic systems, which can be easily accomplished using the observer based fault detection technique. Nevertheless, the faults with slow time constants might not be detected, since the improvement of robustness is related with the decrease of the sensitivity of the observer to faults with slow time constants. Therefore, an adaptive observer is proposed to use to overcome this difficulty. An adaptive observer is a dynamical system that estimates states and (slowly varying) unknown parameters of the observed system. One may expect that a residual generator based on an adaptive observer does not only maintain the important property of early detection of abrupt changes, but also delivers estimates of faults with slow time constants. Another motivation is that by applying on-line identification the process model can continuously be updated and the robustness of the residual with respect to model uncertainties can thus be enhanced (Frank and Ding, 1997).

In Ballesteros-Moncada et al. (2015), FD method was developed for CSTR using Luenberger fuzzy observer and Walcott-Zak observer. However, this method cannot be worked in more complex chemical reactors such as Chylla-Haase reactor. In Zhu and Cen (2010), FDI for a class of uncertain nonlinear systems based on observers is designed. Firstly, by using the sliding model control and adaptive observer design techniques, we develop a robust and adaptive full-order observer design method. The full-order observer is considered as a detection observer directly since it is robust to the disturbances of the system but sensitive to the actuator faults. Secondly, by choosing a special gain matrix, a reduced-order observer is constructed and it can eliminate the influence of the disturbances and faults directly. However, this method is sensitive for disturbances when applied to complex processes.

Another study conducted by Zarei and Shokri (2014), FD method was proposed for CSTR using a Nonlinear Unknown Input Observer NUIO) for robust sensor fault detection. The proposed method is based on cubature rule. NUIO decouples disturbances and uncertainties from estimated states in nonlinear systems.

The above studies deal almost with linear systems subject to simple additive failures. In this work we present a nonlinear observer based fault detection. The application of observer based fault detection (FD) has been intensively studied over the last two decades. In Polycarpou and Helmicki (1995), detecting faults in nonlinear dynamic systems using observer model based approach was proposed. Another method was studied in Polycarpou and Vemuri (1995)used a learning methodology for failure detection and accommodation. The main idea behind this approach is to monitor the physical system for any off-nominal behaviour in its dynamics using nonlinear modelling techniques. In Demetriou and Polycarpou (1998) authors studied the design and analysis of a general framework for model-based fault detection and diagnosis of a class of incipient faults. An automated fault diagnosis architecture using nonlinear online approximators with an adaptation scheme is designed and analysed. In Trunov and Polycarpou (2000) researchers presented in their paper a robust fault diagnosis scheme for detecting and approximating state and output faults occurring in a class of nonlinear multiinput–multioutput dynamical systems. The robust fault diagnosis scheme utilizes on-line approximators and adaptive nonlinear filtering techniques to obtain estimates of the fault functions. In Keliris et al. (2015)authors developed a nonlinear observer-based approach for distributed fault detection of a class of interconnected input–output nonlinear systems, which is robust to modelling uncertainty and measurement noise. First, a nonlinear observer design is used to generate the residual signals required for fault detection. Then, a distributed fault detection scheme and the corresponding adaptive thresholds are

designed based on the observer characteristics. In Xiaodong et al. (2002) researchers studied a robust fault diagnosis scheme for abrupt and incipient faults in nonlinear uncertain dynamic systems. A detection and approximation estimator is used for online health monitoring. Once a fault is detected, a bank of isolation estimators is activated for the purpose of fault isolation.

In this work a fault diagnosis methodology for incipient and abrupt faults is developed. We consider nonlinear dynamical systems whose dynamics change at some unknown time due to a failure. This change is modelled as an unknown nonlinear function of the state and input variables with a time-varying failure profile. In order to capture the nonlinear characteristics of faults, we design a nonlinear estimator using the online approximation (OLA) approach with an adaptive scheme for the adjustable parameters or weights. The stability and performance properties of the fault diagnosis scheme are rigorously established under the assumption of full state measurement. These results are obtained in the presence of *approximation errors,* that is, errors arising as a result of imperfect modelling of the system deviations due to faults by the online approximator. From an adaptive theory viewpoint, the objective of this section is to develop a learning methodology for incipient failure detection. In this framework, online approximators such as neural networks are used to monitor the system for any deviations due to faults. By using the adaptively capabilities of online approximators, they can be used not only to detect the occurrence of System failures, but also to provide an online estimate of the fault characteristics (diagnosis). The main limitations in the use of learning methods for fault diagnosis are the need for significant computational capabilities and the requirement to obtain rigorous analytical results on the performance properties of the fault diagnosis scheme. The derivation of analytical results on the performance properties of the fault diagnosis scheme is difficult due to the nonlinear nature of the problem and the inherent

coupling between estimation and adaptation. The fault diagnosis scheme is developed for MIMO nonlinear systems with both state and sensor faults, which may occur simultaneously or independently. The fault in each of the states/outputs is allowed to evolve at a different rate, covering both incipient and abrupt faults.

Chapter 3

Chylla-Haase Benchmark Process Modelling

In the last two decades, batch and semi-batch processes have been widely used in the fine chemicals industry. Many chemical manufactures such as polymer and pharmaceutical products are manufactured in batch and semi-batch operations. From a process system point of view, the semi-batch operations are described as a reactant may be added with no product removal. Whereas, in batch operations, all the reactants are added and charged in a reactor at the start with no material added or removed (Bonvin, 1998, Srinivasan et al., 2003).

In this research, a semi-batch polymerization reactor benchmark is considered which is described by Chylla and Haase (1993) and used as a benchmark for process control applications, Due to its semi-batch nature, the process shows time varying behaviour and high nonlinear. In addition, changes in the viscosity of the polymer solution over the course of the reaction is resulted changing in heat transfer characteristics. Due to the increasing of the fouling of the reactor walls, the behaviour of the process change from batch to batch. Also the behaviour often changes due to changes in the environmental conditions such as cooling water temperatures and external temperatures. All those semi-batch nature make the reactor very complex high nonlinear and difficult to control (Clarke-Pringle and MacGregor, 1997).

3.1. Description of the Process

The schematic diagram of the semi-batch polymerization reactor is shown in Figure 3.1 (Chylla and Haase, 1993). It consists of a stirred tank reactor with cooling jacket and a coolant recirculation. The reactor temperature is controlled by manipulating the temperature of the coolant, which is recirculated through the cooling jacket of the reactor. The heat released through the reaction must be removed by circulating cold water through the jacket, where both hot and cold jacket streams are available. When the jacket temperature controller output is between 0 and 50%, the valve is opened and cold water is inserted, and when the jacket controller output is between 50 and 100%, the valve is opened and steam is inserted (Beyer et al., 2008, Graichen et al., 2005b).

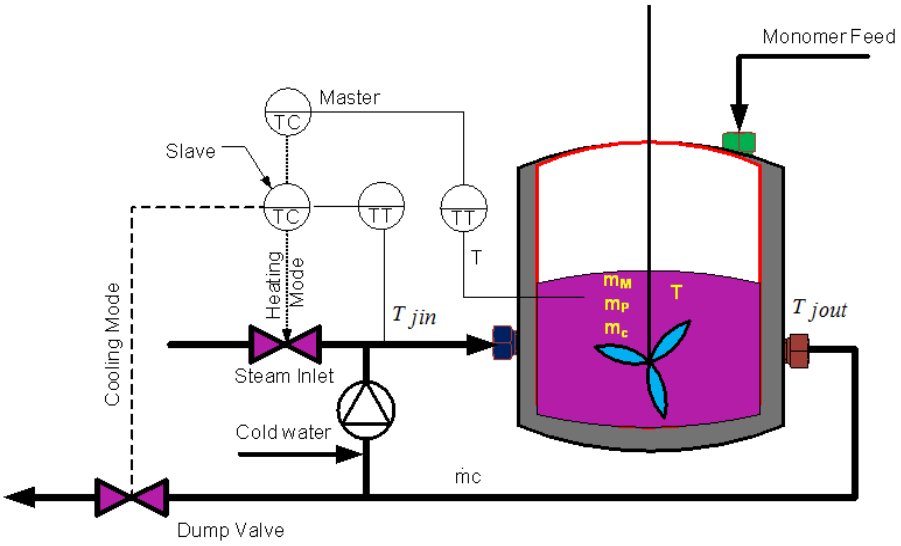


Figure 3.1 Chylla-Haase Reactor Schematic (Chylla and Haase,1993).

3.1.1. Polymerization Reactor Dynamic Model

The mathematical model of the Chylla-Haase reactor is described by a set of five ordinary differential equations (ODE) which come from material and heat balances inside the reactor. The reactor simulation model used here in this research work is developed using

MATLAB/SIMULINK. The material balances for monomer mass m_M and polymer mass m_P are described by equations (3.1) and (3.2) respectively, as follows:

$$\frac{dm_M}{dt} = \dot{m}_M^{in}(t) + \frac{Q_{rea}}{\Delta H} \tag{3.1}$$

$$\frac{dm_P}{dt} = -\frac{Q_{rea}}{\Delta H} \tag{3.2}$$

Where $\dot{m}_M^{in}$ is the monomer feed rate, ΔH is the heat enthalpy and Q_{rea} is the reaction heat. The reaction heat here is defined as:

$$Q_{rea} = -\Delta H * R_p \tag{3.3}$$

Where R_p is the rate of polymerization. The reaction rate is usually defined as a function of temperature and represented as:

$$R_p = i * k * m_M \tag{3.4}$$

$$k = k_0 \exp(-E/RT)(k_1\mu)^{k_2} \tag{3.5}$$

Where i is the impurity factor, k is the reaction rate, E the activation energy, R the ideal gas constant, μ the batch viscosity which is described as:

$$\mu = c_0 \exp(c_1 f) * 10^{c_2(a_0/T - c_3)} \tag{3.6}$$

$$f = m_P / (m_P + m_M + m_C) \tag{3.7}$$

Where f is the solid mass fraction.

The reactor model includes the material balances (3.1) and (3.2) for the monomer mass $m_M(t)$ and the polymer mass $m_P(t)$, the energy balance (3.8) with the reactor temperature $T(t)$, plus the energy balances (3.9) and (3.10) of the cooling jacket and the recirculation loop with the outlet and inlet temperatures $T_{jin}(t)$ and $T_{jout}(t)$ of the coolant. The

available measurements of the process are the temperature of the reactor and the cooling circuitry (Graichen et al., 2006):

$$\frac{dT}{dt} = \frac{1}{\sum_i m_i C_{P,i}} [\dot{m}_M^{in}(t) C_{P,M} (T_{amb} - T\ ) - UA(T - T_j) - (UA)_{loss} (T - T_{amb}) + Q_{rea}\ ] \tag{3.8}$$

$$\frac{dT_{jout}}{dt} = \frac{1}{m_C C_{p,C}} [\dot{m}_C C_{P,C} (T_{jin}(t - \theta_1) - T_{jout}) + UA(T - T_j)] \tag{3.9}$$

$$\frac{dT_{jin}}{dt} = \frac{dT_{jout}(t - \theta_2)}{dt} + \frac{T_{jout}(t - \theta_2) - T_{jin}}{\tau_P} + \frac{K_P(c)}{\tau_P} \tag{3.10}$$

The overall heat transfer coefficient U is calculated as following:

$$U = 1 \Big/ (h^{-1} + h_f^{-1}) \tag{3.11}$$

$$h = d_0 \exp(d_1 \mu_{wall}) \tag{3.12}$$

Where h is the heat transfer coefficient , and h_f^{-1} is the fouling factor and should change as illustrated in Table 3.1.

Table 3.1 Fouling factor values

Batch	1	2	3	4	5
h_f^{-1}	0.0	0.176	0.352	0.528	0.704

The heating/cooling function $K_P(c)$ is influenced by an equal-percentage valve with valve position $c(t)$ as shown in equation (6):

$$K_P(c) = \begin{cases} 0.8 \times 30^{-c/50} (T_{inlet} - T_{jin}(t)), & c < 50\,\% \\ 0, & c = 50\,\% \\ 0.15 \times 30^{(c/50 - 2)} (T_{steam} - T_{jin}(t)), & c > 50\,\% \end{cases} \tag{3.13}$$

For $c < 50\%$, cold water with inlet temperature T_{inlet} is injected in the cooling jacket, whereas a valve position $c > 50\%$ leads to a heating of the coolant by injecting steam with temperature T_{steam} into the recirculating water steam.(Graichen et al., 2006).

3.1.2. Uncertainties and Disturbances in the Process

In order to model the following practical issues of the control of polymerization reactors, various disturbances and uncertainties are identified:

- The impurity factor $i \in [0.8:1.2]$ in the polymerization rate R_p is random but constant during one batch, which tries to simulate fluctuations in monomer kinetics caused by batch to batch variations in reactive impurity.
- The fouling factor $1/h_f$ in the overall heat transfer coefficient U increases with each batch and accounts for the fact that during successive batches a polymer film builds up on the wall resulting in a decrease of U.
- The delay times θ_1 and θ_2 of the cooling jacket and the recirculation loop may vary by $\pm 25\%$ compared to nominal values.
- The ambient temperature T_{amb} is different during summer and winter. This affects the temperature of the monomer feed $\dot{m}_M^{in}$, as well as the initial conditions $T(0)$, $T_{jin}(0)$ and $T_{jout}(0)$ given by T_{amb} (Graichen et al., 2006).

Table 3.3 describes the empirical relations for the polymerization rate, the jacket heat transfer area, and the overall heat transfer coefficient (Graichen et al., 2006).

Table 3.2 Parameters value of the reactor

Symbol	Unit	Value of Polymer A	Value of Polymer B
$m_{M,0}$	kg	0	0
$m_{P,0}$	kg	11.227	11.010
m_W	kg	42.750	42.010
ρ_m	kgm^{-3}	900	900
ρ_p	kgm^{-3}	1040	1040
ρ_w	kgm^{-3}	1000	1000
$c_{p,M}$	$kj\,kg^{-1}\,K^{-1}$	1.675	1.675
$c_{p,P}$	$kj\,kg^{-1}\,K^{-1}$	3.140	3.140
$c_{p,W}$	$kj\,kg^{-1}\,K^{-1}$	4.187	4.187
m_C	kg	21.455	21.455
$\dot{m}_C$	$kg\,s^{-1}$	0.9412	0.9412
$c_{p,c}$	$kj\,kg^{-1}\,K^{-1}$	4.187	4.187
k_0	s^{-1}	55	20
k_1	$m\,s\,kg^{-1}$	1000	1000
k_2	$m\,s\,kg^{-1}$	0.4	0.4
E	$kj\,kmol^{-1}$	29560.89	29560.89
c_0	$kg\,m^{-1}\,s^{-1}$	$5.2*10^{-5}$	$3.2*10^{-5}$
c_1	$kg\,m^{-1}\,s^{-1}$	16.4	19.1
c_2	$kg\,m^{-1}\,s^{-1}$	2.3	2.3
c_3	$kg\,m^{-1}\,s^{-1}$	1.563	1.563
a_0	K	555.556	555.556
$-\Delta H_p$	$kj\,kmol^{-1}$	−70,152.16	− 765,593.2
d_0	$kW\,m^{-2}\,K^{-1}$	0.814	0.814
d_1	$m\,s\,kg^{-1}$	− 5.13	− 5.13
$\dot{m}_M^{in,\max}$	$kg\,s^{-1}$	0.007560	0.006048
$[t_M^{in},0,t_M^{in},1]$	min	[30,100]	[30,90]
$[t_M^{in},2,t_M^{in},3]$	min		[120,160]
T_{set}	K	355.382	353.160
P	m	1.594	
B_1	m^2	0.193	
B_2	m^2	0.167	
R	$kj\,kmol^{-1}\,K^{-1}$	8.314	

$(UA)_{loss}$	kWK^{-1}	0.00567567	
τp	s	40.2	
θ_1	s	22.8	
θ_2	s	15	
T_{amb}	K	280.38(W),305.38(S)	
T_{inlet}	K	278.71(W),294.26(S)	
T_{steam}	K	449.82	
i	–	[0.8:1.2]	

3.2. Matlab Simulink Model Development

Simulink is a part of MATLAB software that provides a graphical environment and solvers for modelling, simulating and analysing of dynamic systems. Here in this section, the Simulink model for the proposed reactor is developed by material and energy balances equations described in (3.1) -(3.13). Figure 3.2 describes the main Simulink model for Chylla-Haase polymerization reactor. The proposed model consists of five main sub-system blocks as shown in Figure 3.3. Each sub-system block represents a mathematical model for material and energy balances as described in previous section.

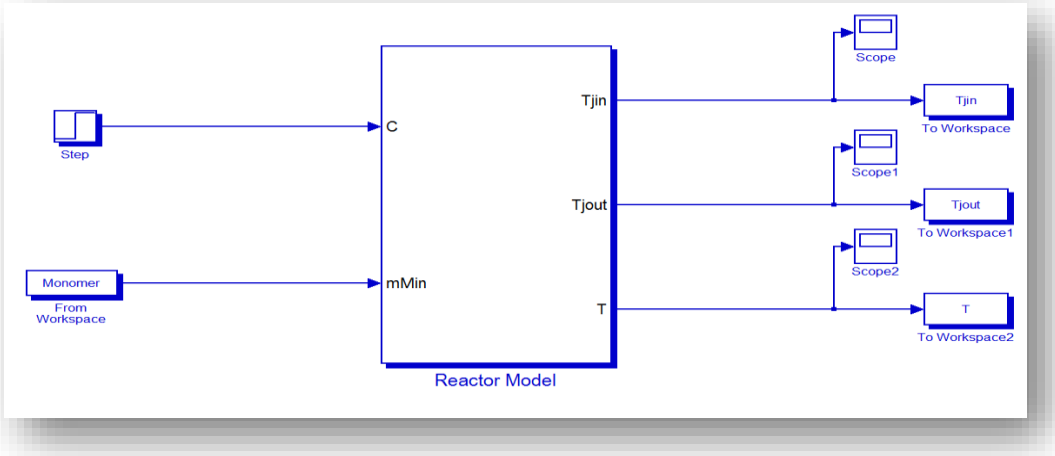


Figure 3.2 Chylla-Haase simulink model

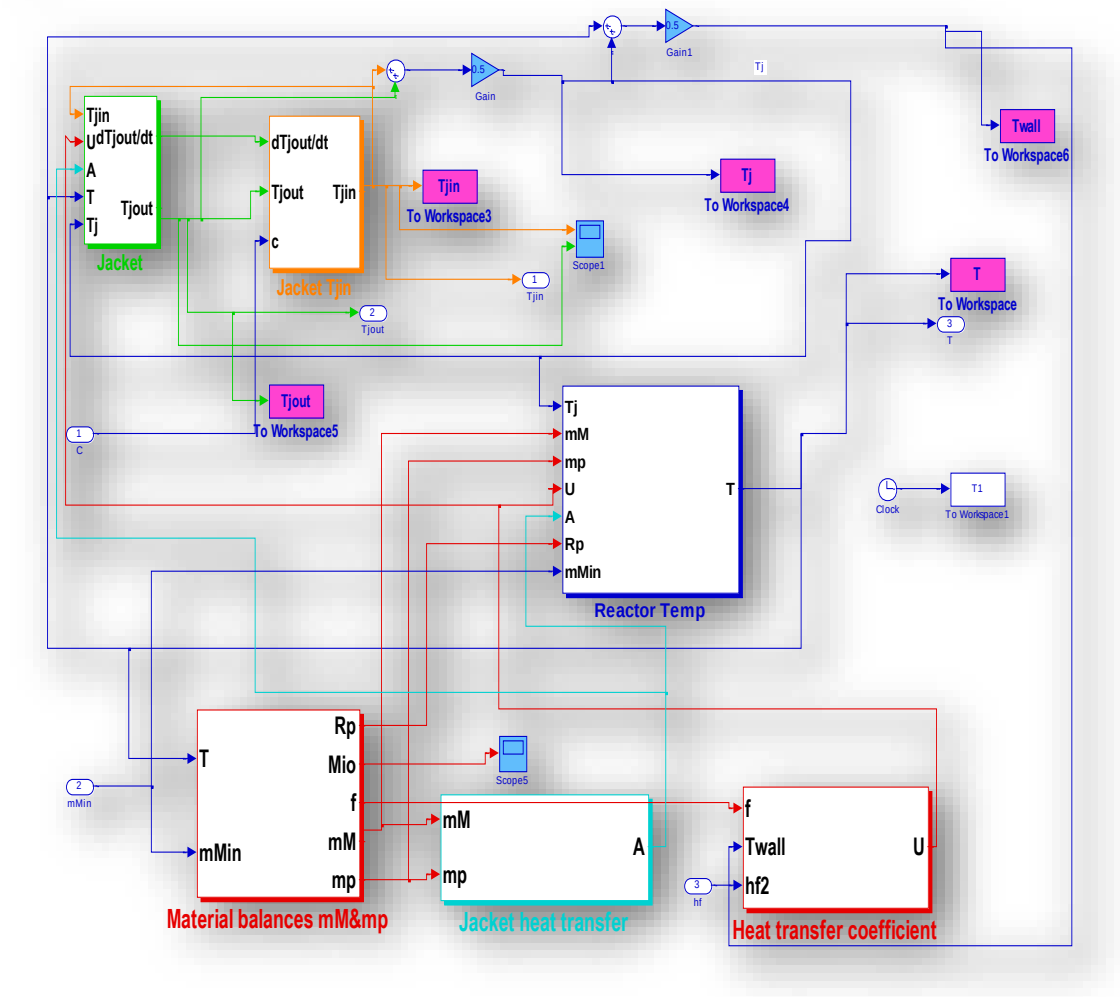


Figure 3.3 Subsystem block of reactor simulink

Figure 3.4 represents the mathematical model for material balances as described in equations (3.1)-(3.7). It can be seen that, the three embedded MATLAB function have been created to solve the empirical relations for the rate of polymerization as described in Table 3.3.

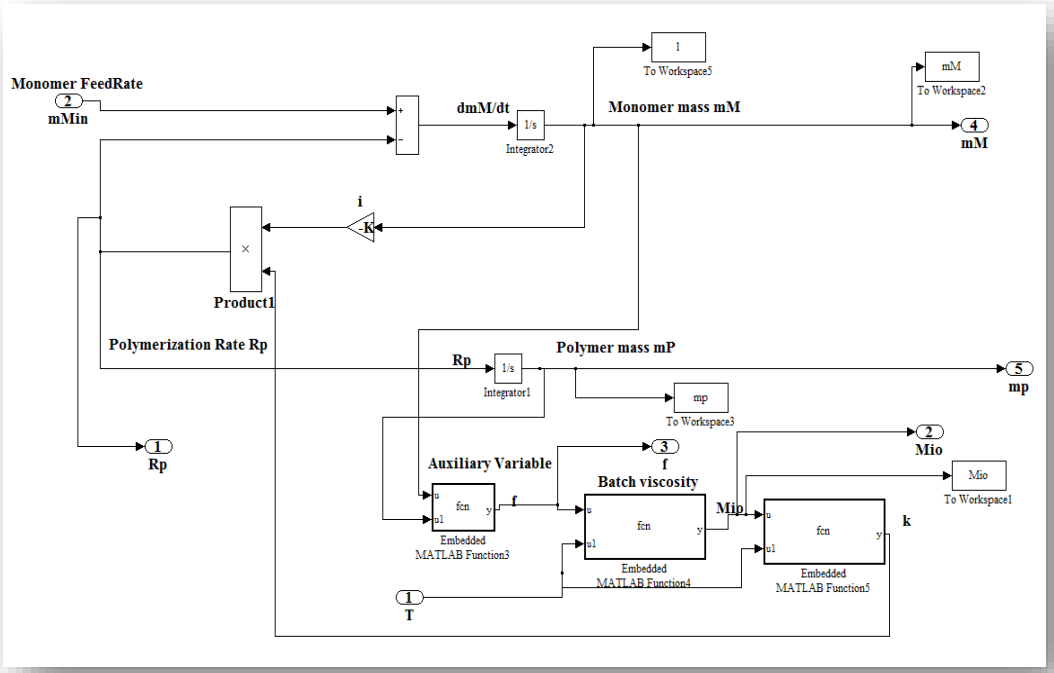


Figure 3.4 Simulink block diagram for material balances

Figure 3.5 shows the Simulink model diagram of the mathematical model for overall heat transfer coefficient which is described by equations (3.11) and (3.12) and illustrated in Table 3.3.

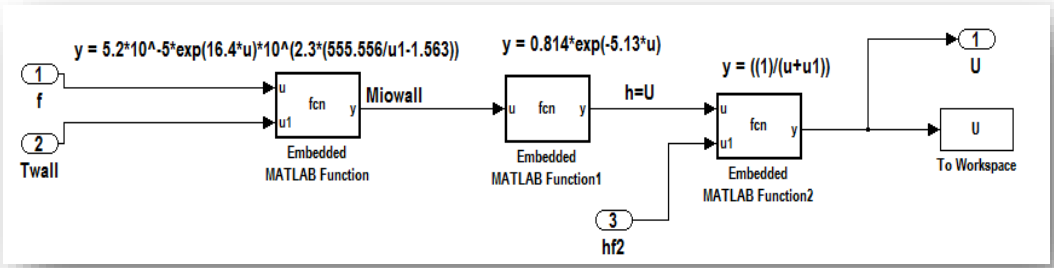


Figure 3.5 Simulink block diagram for overall heat transfer

The Simulink models of the recirculation loop, jacket temperature, and reactor temperature are designed and developed according to the mathematical equations (3.9) - (3.10) as shown in Figures 3.6, 3.7 and 3.8 below.

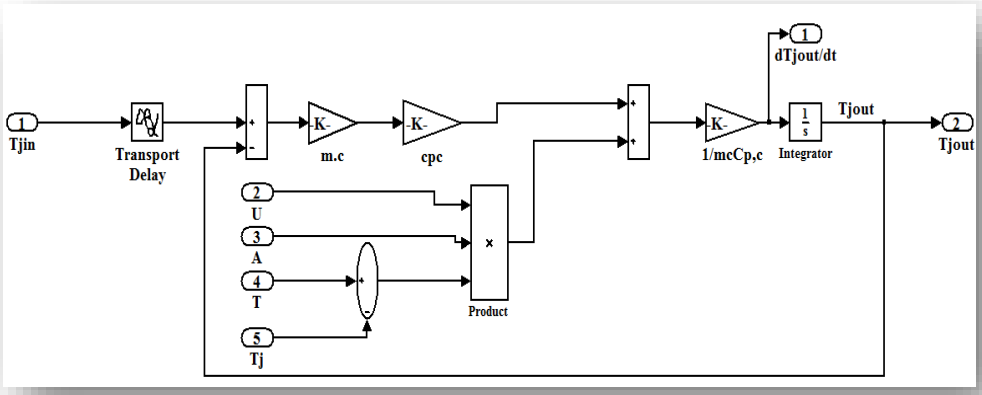


Figure 3.6 Simulink block diagram for jacket

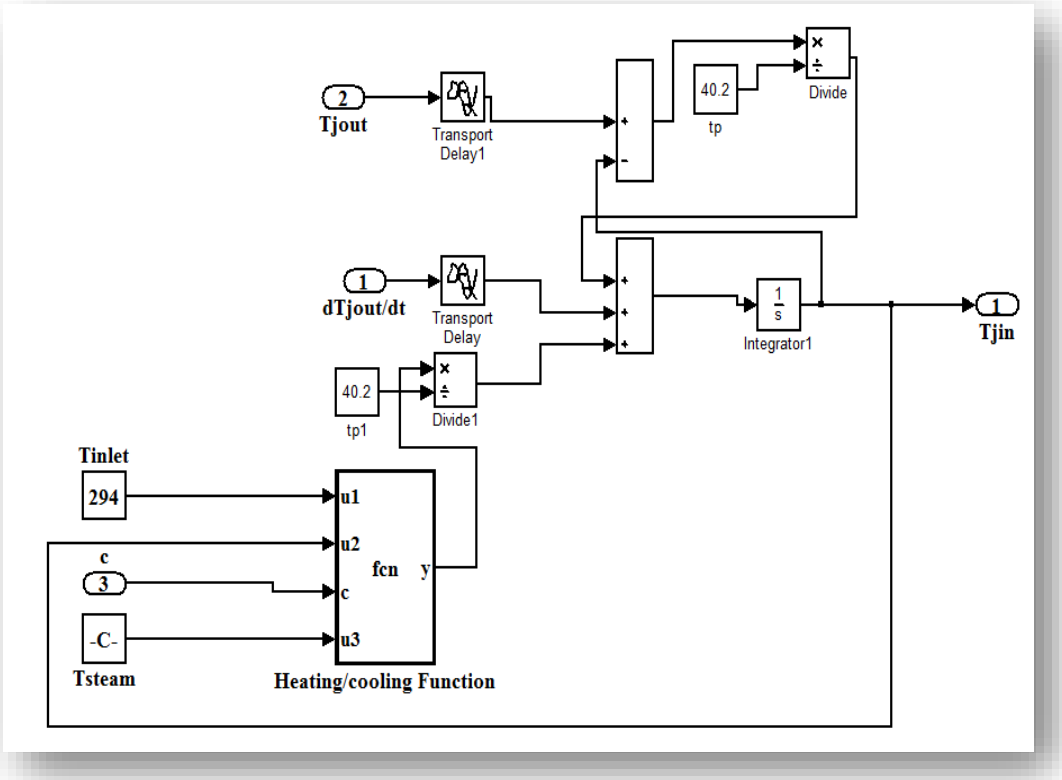


Figure 3.7 Simulink block diagram for recirculation loop

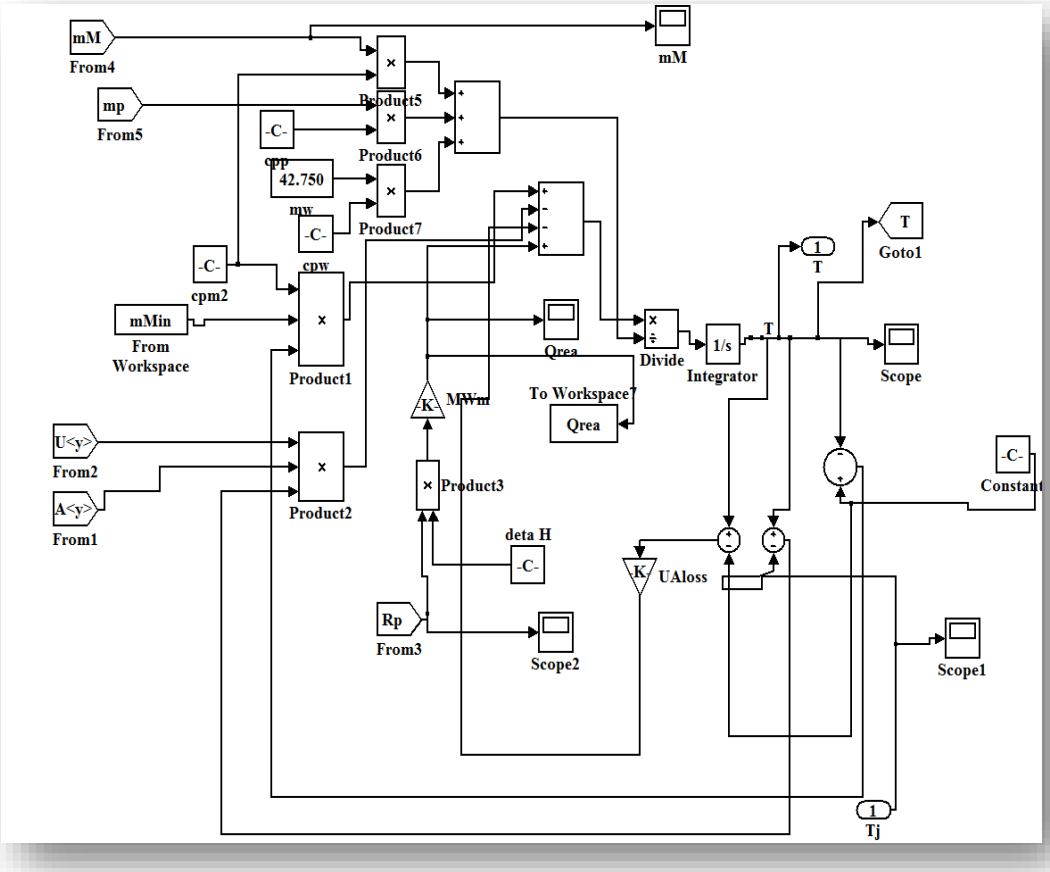


Figure 3.8 Simulink block diagram for reactor temperature

3.3. Performances and Discussion

After building up the Simulink model for the reactor, the polymerization process is simulated using the parameters values as described in Table 3.4. Firstly, we run the reactor using the parameter values for polymer A. the initial parameters of polymer, monomer, and water are set into reactor at ambient temperature. Before feeding in monomer into the reactor, the valve is set up to fully open mode in order to heat up the reactor and full steam inserted. After 1800s the monomer is fed into reactor at 0.0075 kg/s until 6000s as shown in Figure 3.9.and the reactor temperature reached 450K as shown in Figure 3.10. After the feed of monomer has stopped the reactor temperature is decreased at held at its set point value as shown in Figure 3.10.

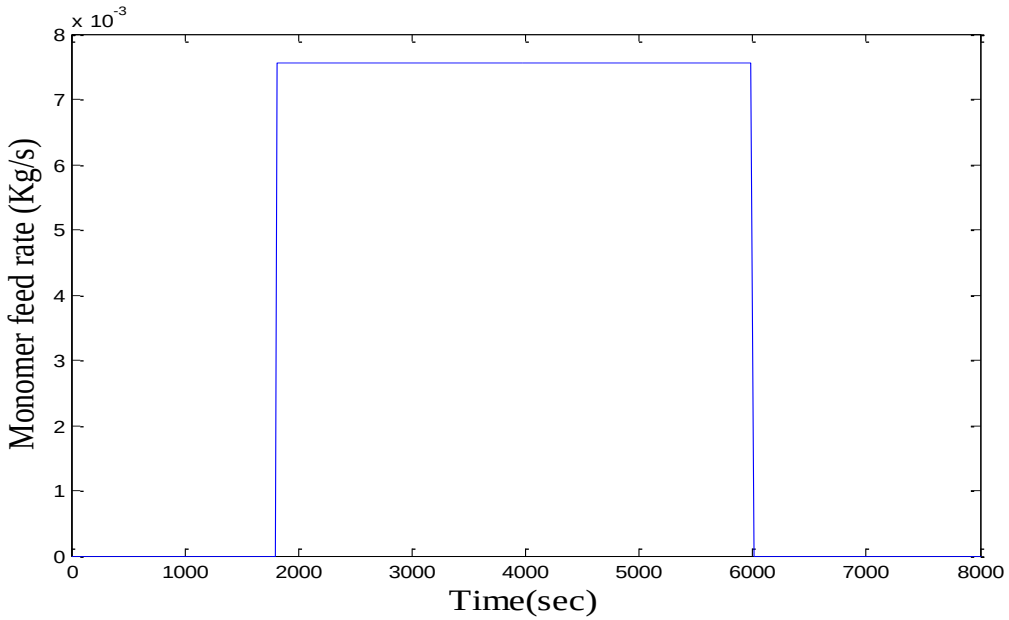


Figure 3.9 Monomer feed rate (Polymer A)

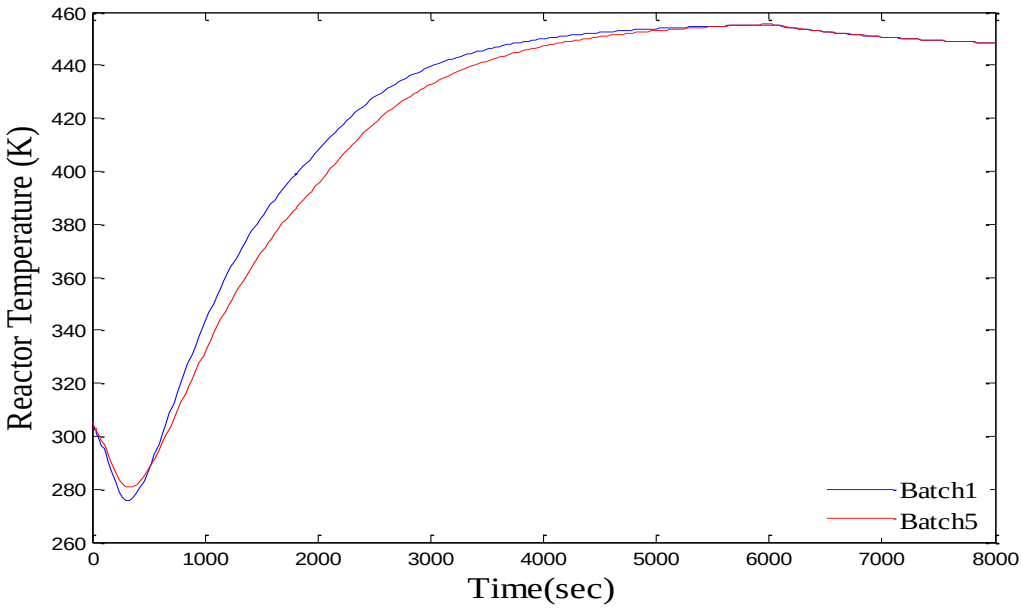


Figure 3.10 Reactor temperature (Polymer A)

Figure 3.11 and 3.12 show the response of the jacket input temperature and jacket output temperature respectively.

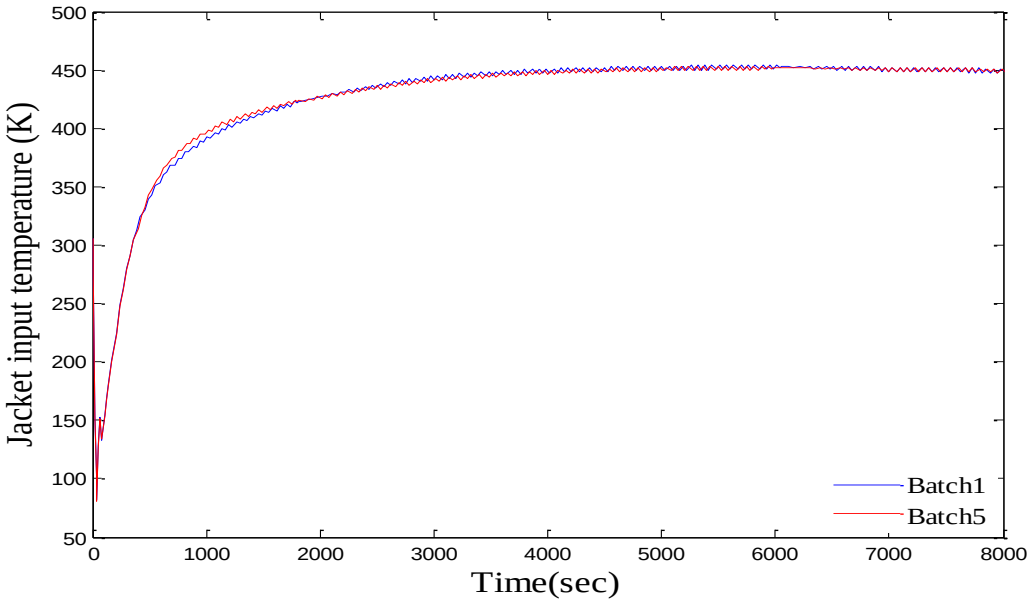


Figure 3.11 Jacket input temperature (Polymer A)

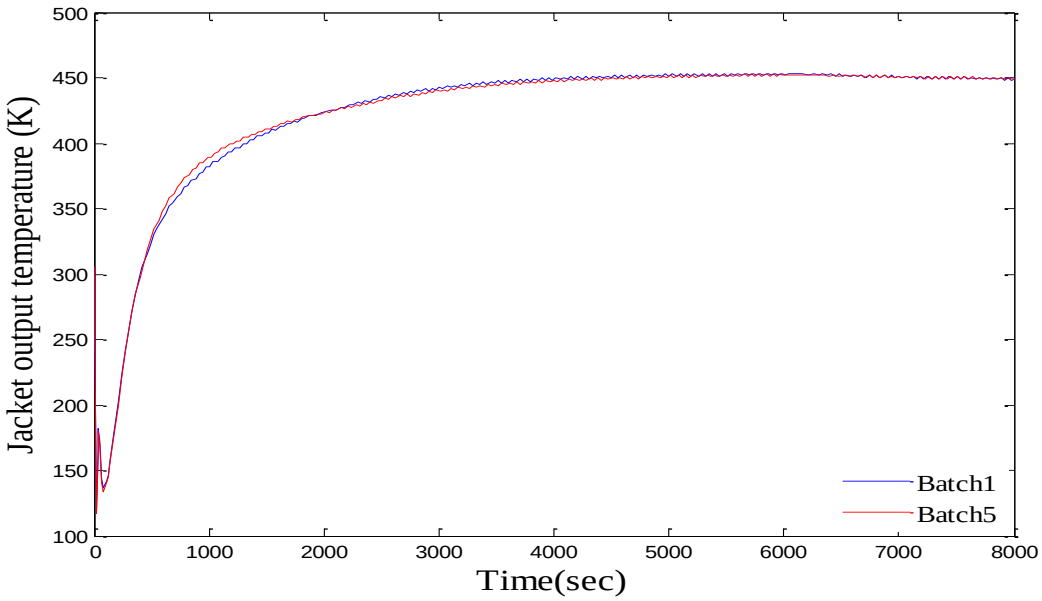


Figure 3.12 Jacket output temperature

Due to nonlinearity of reaction kinetics, the heat transfer coefficient sharply decreases during a batch because of viscosity increasing, as shown in Figure 3.13. It can be clearly noticed that, the difference between the responses from batch one when fouling factor is equal to zero to batch five when fouling factor is 0.704, due to the increase of the fouling

factor and viscosity. So this conclude that, the heat transfer coefficient is related inversely with the fouling factor and the viscosity.

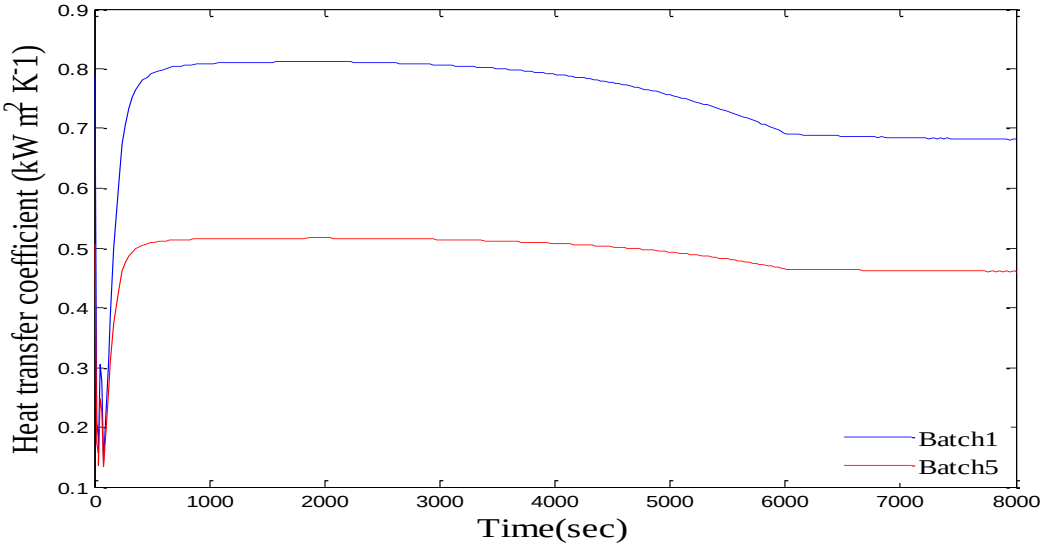


Figure 3.13 Overall heat transfer coefficient (Polymer A)

Figure 3.14 shows the response of the reaction heat. It can be clearly seen that, the direct correlation between the reaction heat and the rate of polymerization as described previously in equations (3.3)-(3.7). When the monomer fed into the reactor at 1800s the reaction heat of the reactor is increased and stayed steady until the monomer is stopped feeding at 6000s then the reaction heat is rapidly decreased to zero at 6000s.

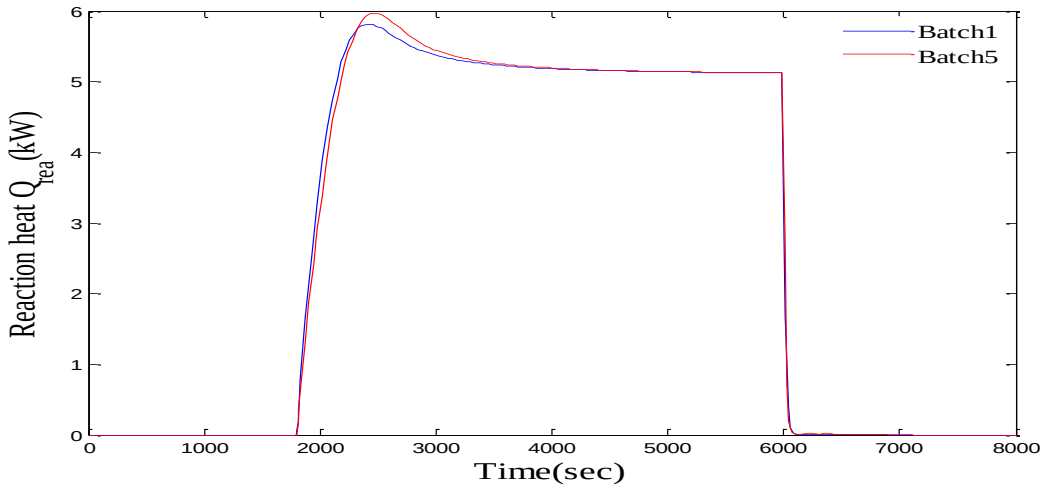


Figure 3.14 The reaction heat (Polymer A)

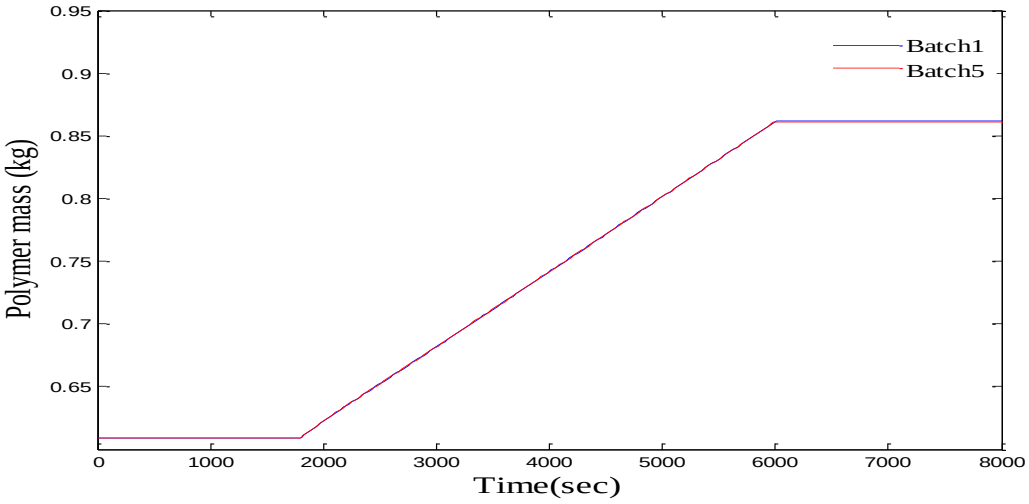


Figure 3.15 Polymer mass (Polymer A)

Figure 3.15 and 3.16 show the response of the polymer mass and monomer mass respectively. It can be clearly noticed from both figures the relation between monomer mass, polymer mass, rate of polymerization and reaction heat as illustrated in equations (3.1) -(3.7).

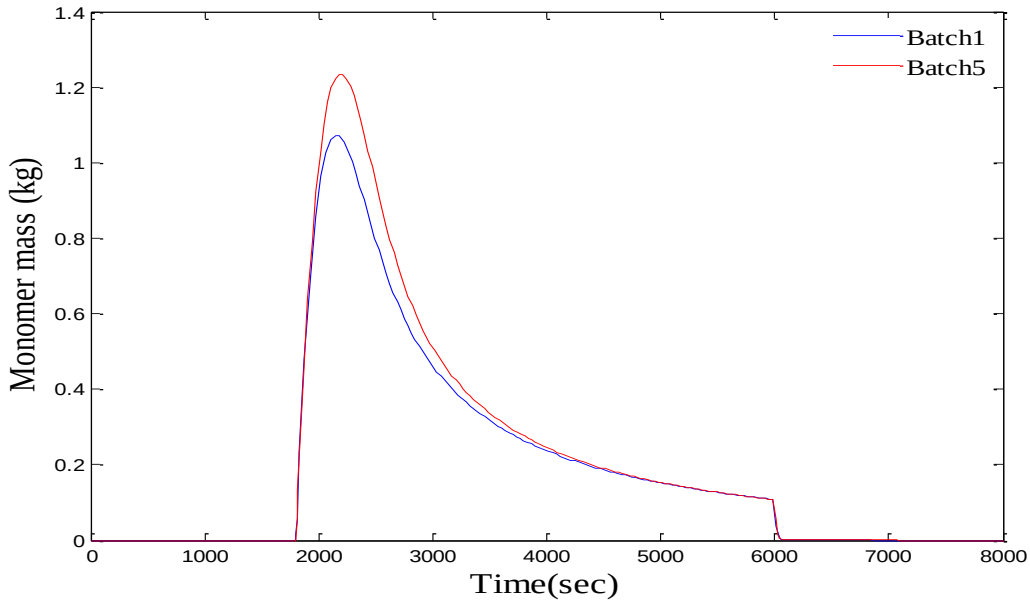


Figure 3.16 Monomer mass (Polymer A)

In this section, the polymerization reactor is simulated using the parameter values for polymer B as described in Table 3.4. Two recipe of polymer b product is feed into the reactor. The monomer of first recipe of polymer B product is fed into reactor at 0.006048 kg/s for 3600s, starting form 1800s stopped at 5400s. Then the second recipe is fed at 0.006048 kg/s starting from 7200s and stopped at 9600s as shown in figure 3.17. Figure 3.18 shows the open-loop response of the reactor temperature for different batches. It can be clearly seen that the reactor temperature is increased when the monomer is fed into reactor, and decreased when the monomer is stopped feeding. Moreover, it can be noticed that the difference of the reactor temperature from first batch to fifth batch, that's due to the increase of fouling factor from batch to batch.

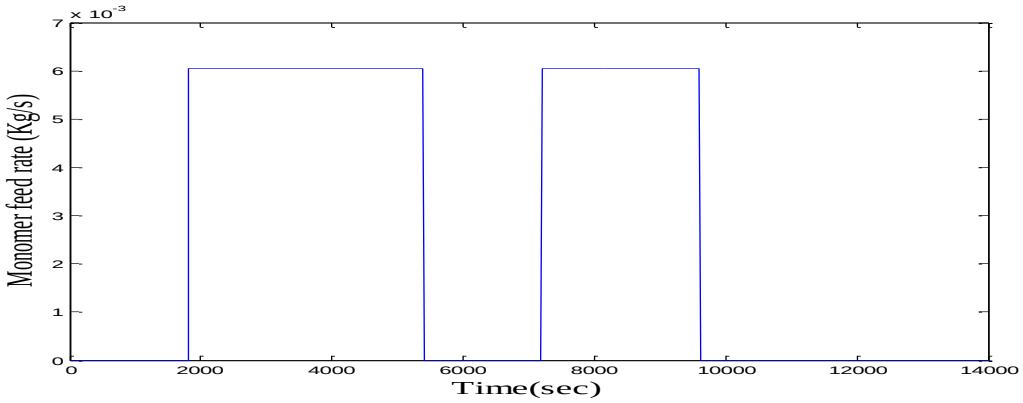


Figure 3.17 Monomer feed rate (Polymer B)

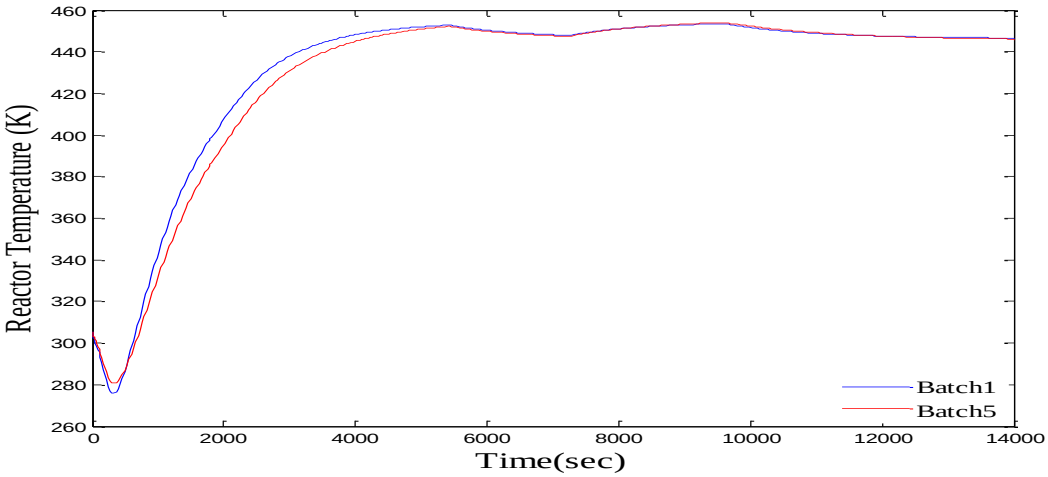


Figure 3.18 Reactor temperature (Polymer B)

The effect of increasing the fouling factor from batch to batch on the overall heat transfer coefficient can be clearly seen in Figure 3.19. This proved the invers relation between the fouling factor and overall heat transfer coefficient as described in equations (3.11) and (3.12).

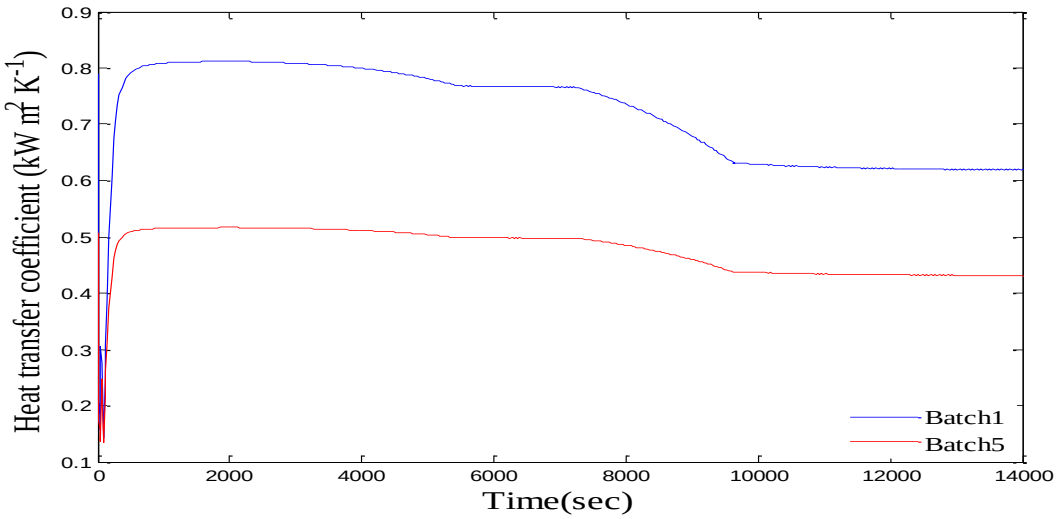


Figure 3.19 Overall heat transfer coefficient (Polymer B)

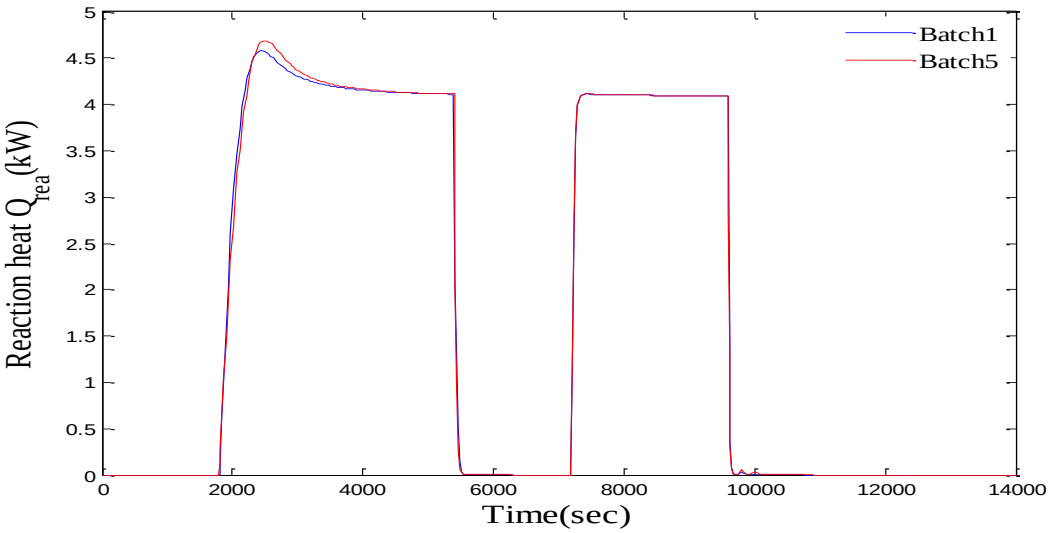


Figure 3.20 Reaction heat (Polymer B)

Figure 3.21 and 3.22 describe the response of polymer mass and monomer mass, respectively.it can be clearly noticed that, the polymer mass is linearly increased with the feeding of the monomer, then stayed steady when the monomer stopped feeding into reactor.

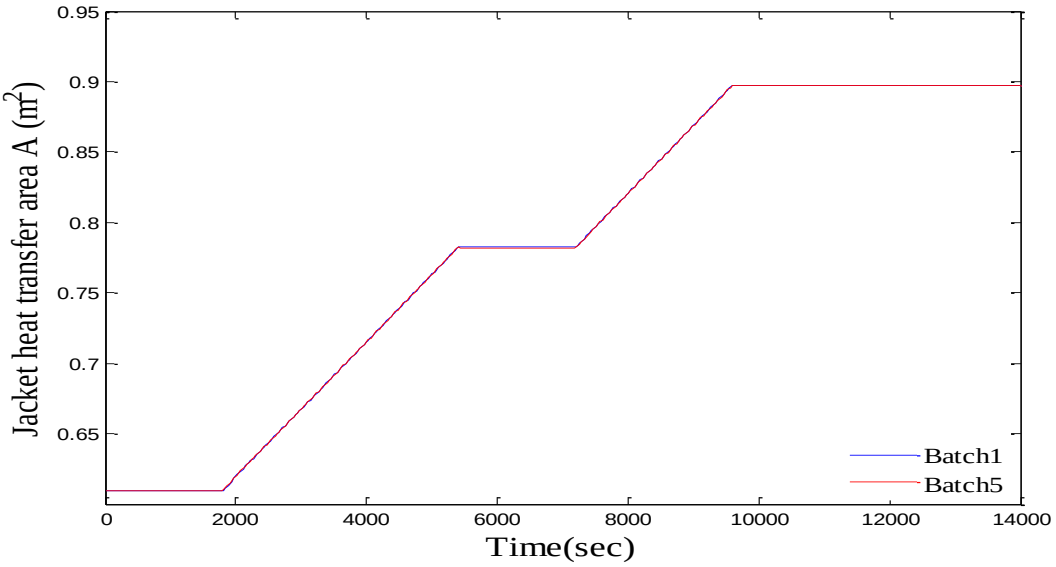


Figure 3.21 Polymer mass (Polymer B)

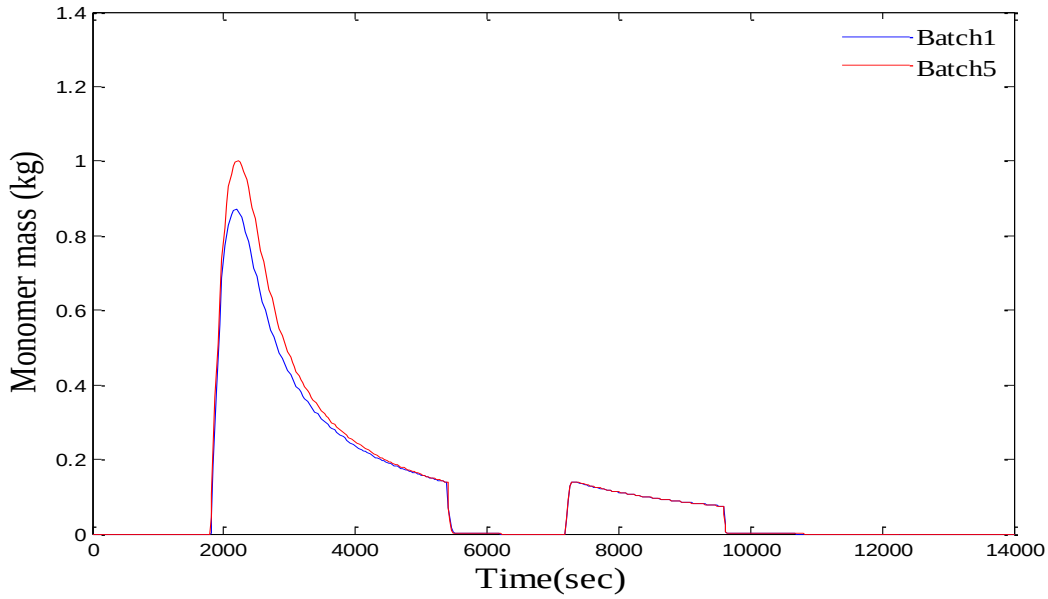


Figure 3.22 Monomer mass (Polymer B)

3.4. Summary

The main aim of this chapter is to understand the nonlinear dynamic behaviour of the Chylla-Haase polymerization reactor. In this chapter, the mathematical model of the proposed reactor is described. The material and energy balances of the reactor are illustrated in more details. All the uncertainties and disturbances in the process is

discussed. Moreover, all parameter values for polymer A and B and all the empirical relations for the polymerization rate, the jacket heat transfer area, and the overall heat transfer coefficient are represented. The Simulink model of the proposed reactor is set up using Simulink/MATLAB. The design of Simulink model is developed based on a set of ordinary differential equations (3.1)-(3.12) that describe the dynamic behaviour of the proposed polymerization process. The Simulink block diagram of the proposed reactor is presented and discussed in more details. The simulation results of open-loop Polymerization process for both polymer A and B are presented and discussed.

Chapter 4

RBF NN Model Based FDI for Open-loop System

An independent radial basis function (RBF) neural networks (RBFNN) are developed and employed here for an on-line diagnosis of actuator and sensor faults. In this research, a robust fault detection and isolation (FDI) scheme is developed for open-loop exothermic semi-batch polymerization reactor described by Chylla-Haase. The independent (RBFNN) is employed here for on-line diagnosis of faults when the system is subjected to system uncertainties and disturbances. Two different techniques to employ RBF neural networks are investigated. Firstly, an independent neural network is used to model the reactor dynamics and generate residuals. Secondly, an additional RBF neural network is developed as a classifier to isolate faults from the generated residuals. Three sensor faults and one actuator fault are simulated on the reactor. Moreover, many practical disturbances and system uncertainties, such as monomer feed rate, fouling factor, impurity factor, ambient temperature and measurement noise are modelled. The simulation results are presented to illustrate the effectiveness and robustness of the proposed method.

4.1. Radial Basis Function Neural Networks (RBFNN)

The RBF network performs nonlinear mapping for modelling nonlinear dynamic systems. Figure 4.1 illustrates the structure of the RBFNN, which consists of three layers: input layer, hidden layer and output layer. The hidden layer contains a number of RBF neurons,

and each of them represents a single radial basis function, with associated centre and width. The transfer function of the hidden layer neurons is radial basis function.

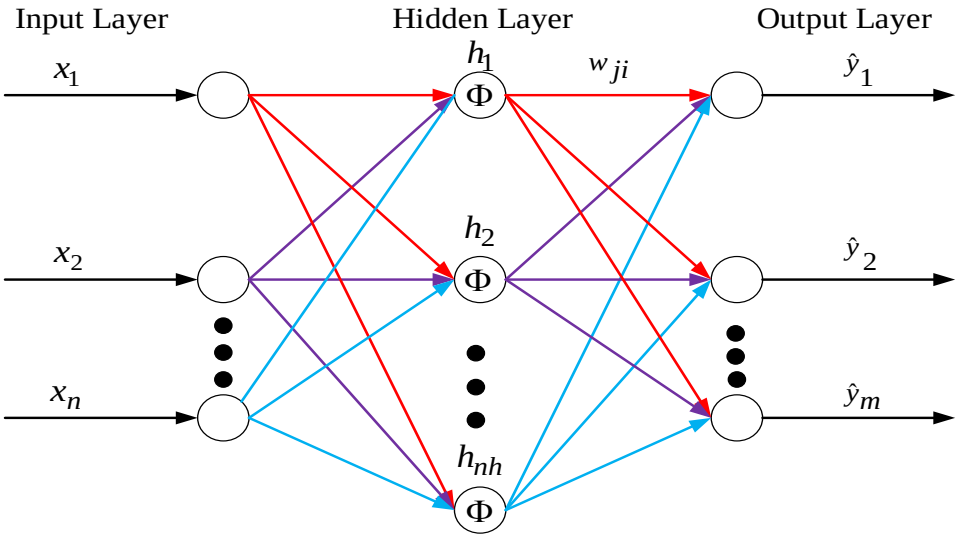


Figure 4.1 RBF NN Structure

4.2. RBF Neural Network Modelling of Cylla-Haase Reactor

4.2.1. Training Algorithm

The output of the hidden layer nodes in RBFNN is produced by so called a nonlinear activation function $\phi_j(t)$. In this work the Gaussian basis function is chosen as the nonlinear activation function.

$$\phi_j(t) = \exp\left(-\frac{\left\| x(t)-c_j(t)\right\|^2}{\sigma_j^{\,2}}\right), \quad j=1,\cdots,n_h \tag{4.1}$$

Where $c_j(t)$ is j^{th} centre, here in this research K-means clustering algorithm is used to choose the centres of the RBF to minimize the sum squared distance from each input data to its closest centre so that the data is adequately covered by the activation function.

$x(t)$ is the neural network input vector which is given as:

$$x(t)= f\left[y(t-1),\ldots,y(t-n_y),u(t-1-d),\ldots,u(t-n_u-d)\right] \tag{4.2}$$

Where σ_j is a positive scalar called a width and n_h is the number of centres. Here in this work the p-nearest algorithm is used to choose the widths, and the recursive training algorithm is employed to update and calculate the weights.

The network outputs are then computed as a linear weighted sum of the hidden node outputs and bias as shown below in equation (4.3):

$$\hat{y}_i(t)=\sum_{j}^{n_h}\phi_j(t)^T w_{ji} \quad ,i=1,\ldots,q \tag{4.3}$$

Where w_{ji} is the output layers weight connecting the j^{th} centre output and i^{th} network output, and q is the number of outputs.

4.2.2. Independent and Dependent Modes of RBF Modelling

Using RBFNN for modelling, a non-linear dynamic system can be modelled in two modes: a dependent mode and an independent mode as shown in figure 4.2 and 4.3. The first model referred to is a dependent mode, since the past system output is used as network input. Thus, the model is dependent on the system output and cannot operate independently from the system. In the independent mode, the past model output is used as network input. Therefore, the model is not dependent on the system output and can operate independently from the system. The independent model has an advantage in that the model can be used to simulate the system to obtain long-range prediction. In contrast, the dependent model performs as one-step-ahead predication.

The RBF model of the dependent form uses both input and output of the process to be modelled. Then, when the process has a fault, the fault will affect the process output, and consequently affect the RBF model output. When the model output is compared with the process output to generate the residual, the residual will not be sensitive to the fault. On the contrary, the RBF model of the independent form use process input and the model output rather than the process output. In this way the occurring fault will not affect the

model output because the process is not fed into the RBF model. Thus, the residual generated by comparing the model output with the process output will be sensitive to the fault.

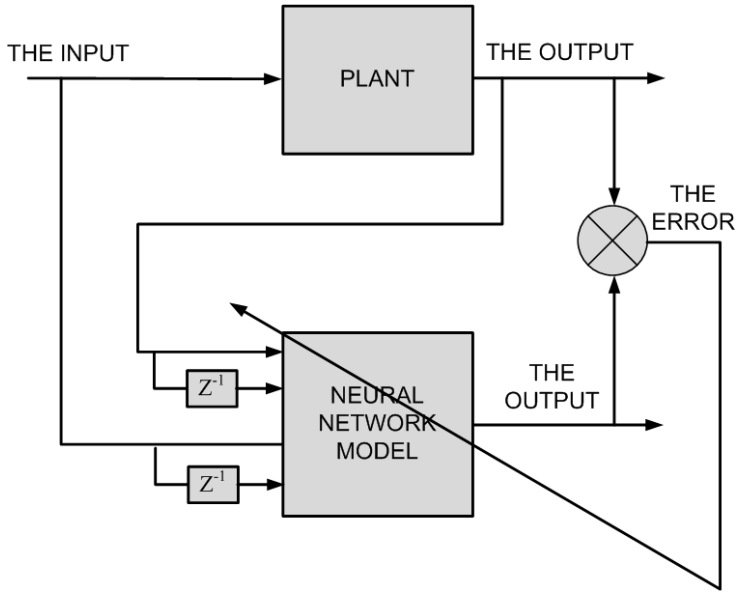


Figure 4.2 Dependent mode

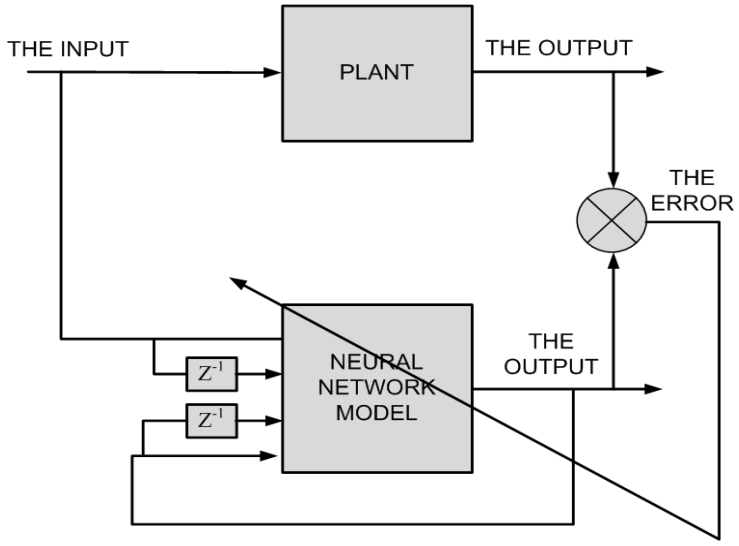


Figure 4.3 An independent mode

The nonlinear dynamic plant to be modelled is presented by the non-linear autoregressive with exogenous inputs (NARX) model as shown in equation (4.4) below:

$$y(t) = f[y(t-1),\ldots,y(t-n_y),u(t-1-d),\ldots,u(t-n_u-d)] + e(t) \qquad (4.4)$$

Where $u \in \Re^m$ and $y \in \Re^p$ are plant input and output respectively. $e \in \Re^p$ is random noise, m and p are the number of plant inputs and outputs respectively, n_y and n_u are the maximum lags in the model output and input, respectively, d is the time delay in inputs, and $f(*)$ is a vector valued non-linear function.

The dependent mode of the network model can be represented by equation (4.5), which is referred to dependent mode as the prediction uses the process output and therefore, the model cannot run independent of the process.

$$\hat{y}(t) = \hat{f}[y(t-1),\ldots,y(t-n_y),u(t-1-d),\ldots,u(t-n_u-d)] \qquad (4.5)$$

Where $\hat{f}(*)$ is a function approximation of $f(*)$. If the past process outputs in the network input are replaced by the network outputs as in equation (4.6) below, then the model is referred to an independent model

$$\hat{y}(t) = \hat{f}[\hat{y}(t-1),\ldots,\hat{y}(t-n_y),u(t-1-d),\ldots,u(t-n_u-d)] \qquad (4.6)$$

4.2.3. Input-Output Determination of RBF Model

The first step towards developing a neural network model of the process is to obtain training data. Training data is obtained by designing a set of random amplitude signals (RAS) for the five inputs to the reactor: monomer feed rate, fouling factor, ambient temperature, impurity factor, and valve position, as shown in figure 4.3. These five inputs are the system inputs (monomer feed rate, manipulated variable) included the uncertainties and disturbances in the process. The second step towards developing a neural network model of the process is to determine the network input variables and the input vector and output vector. The network input vector consists of the past values of the five system inputs and the past values of the three system outputs. The determination of the inputs and outputs of the system is based on the equations (1) to (5). A total data set

of 2000 samples is collected from the system Simulink model, and 4s are used as the sampling time. The first 1500 samples are used for training the network model, and the remaining 500 samples are used for testing the network model. Before training and testing, the raw data is scaled linearly into the range of [0 1] using the following formulae:

$$u = \begin{bmatrix} m_M \\ 1/h_f \\ T_{amb} \\ i \\ c \end{bmatrix}, \quad y = \begin{bmatrix} T_{jin} \\ T_{jout} \\ T \end{bmatrix} \tag{4.7}$$

$$u_{scaled}(k) = \frac{u(k) - u_{\min}}{u_{\max} - u_{\min}} \qquad y_{scaled}(k) = \frac{y(k) - y_{\min}}{y_{\max} - y_{\min}} \tag{4.8}$$

4.2.4. Data Acquisition and Pre-processing

In this research, an independent RBF network is used to represent the NARX model in equation (4.6). Thus, in order to get a good training result with minimum modelling error, several numbers of maximum lags in the outputs and inputs, and several numbers of the maximum time delay in the inputs are tried. The maximum lags in the output were selected as 3, the maximum lags in the input is selected as 3, and the maximum time delay in the inputs is selected as 2, as described in equation (4.9). Thus, the RBF model is designed to have 24 inputs and 3 outputs, as shown in figure 4.8. The hidden layer nodes are selected as 21. The centres are chosen using a K-means clustering algorithm as 21. Moreover, a p-nearest-neighbours algorithm is used to choose the widths. In the training of the network model, the recursive least squares (RLS) algorithm is used to update the weight matrix since the weights are linearly related to the output, and the parameters of the RLS algorithm are selected as follows: $\mu = 0.999$, $w(0) = 10^{-6} * U(n_h, 3)$ and

$p(0) = 10^6 * I(n_h)$ where μ is the forgetting factor, I is an identity matrix, U is the element unity matrix, and n_h is the number of hidden layer nodes.

$$x(t) = [\hat{y}(t-1)\ \hat{y}(t-2)\ \hat{y}(t-3)\ u(t-k-1)\ u(t-k-2)\ u(t-k-3)]^T \quad (4.9)$$

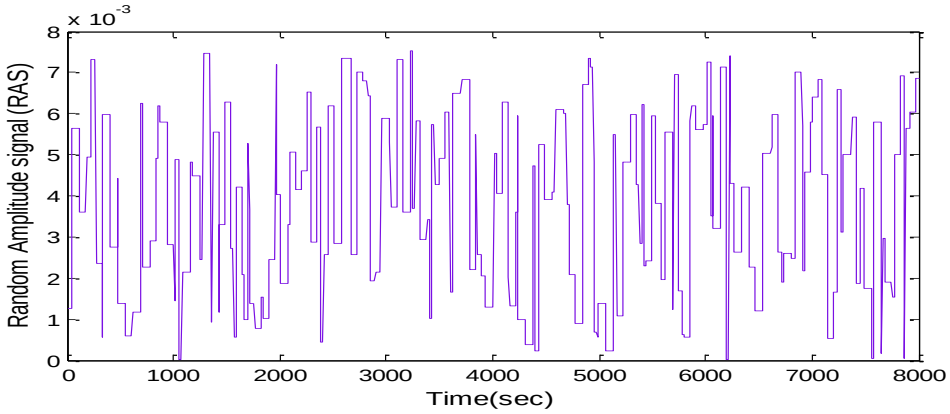


Figure 4.4 RAS signal

Based on equations (4.7) -(4.9), figure 4.5 demonstrates the fault detection approach. An independent model is implemented in parallel with the system to generate the residuals for detecting the sensor and actuator faults in the reactor.

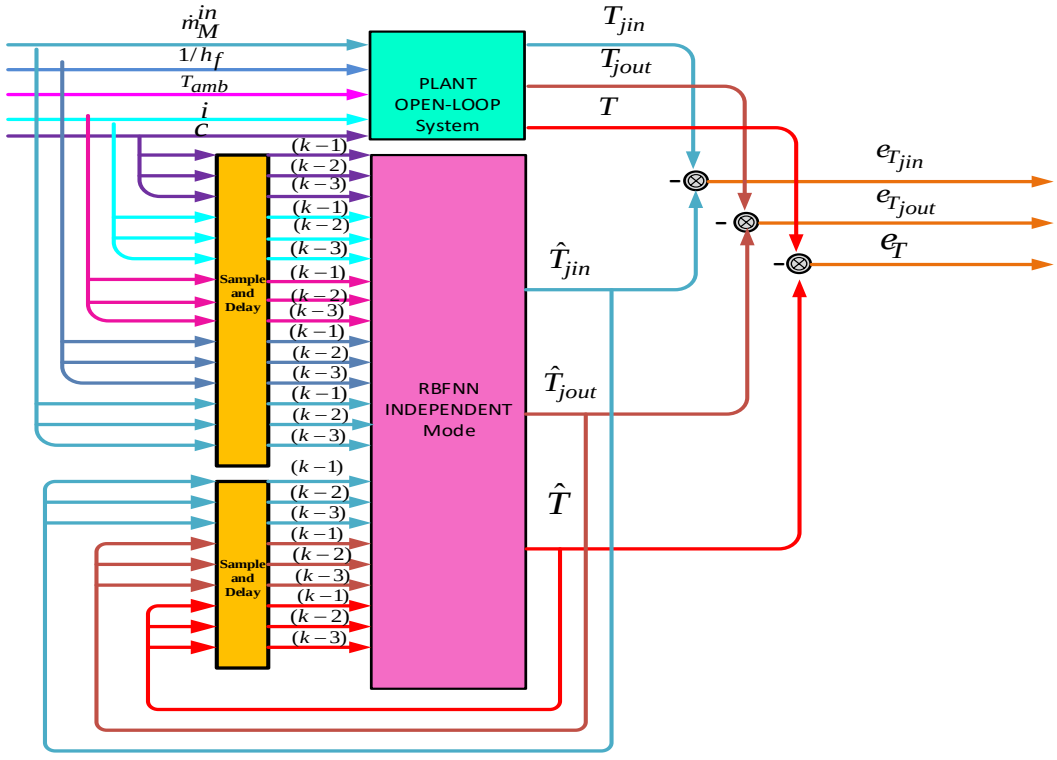


Figure 4.5 The structure of FD using an independent RBFNN

Figure.4.6, 4.7, and 4.8 shows the last 200 sample intervals in the training data set and the first 200 sample intervals in the testing data set. It can be clearly seen that the model outputs track the system output with a small modelling error. The mean absolute error (MAE) for the jacket input temperature, jacket output temperature and reactor temperature are 0.004, 0.0054 and 0.0072, respectively.

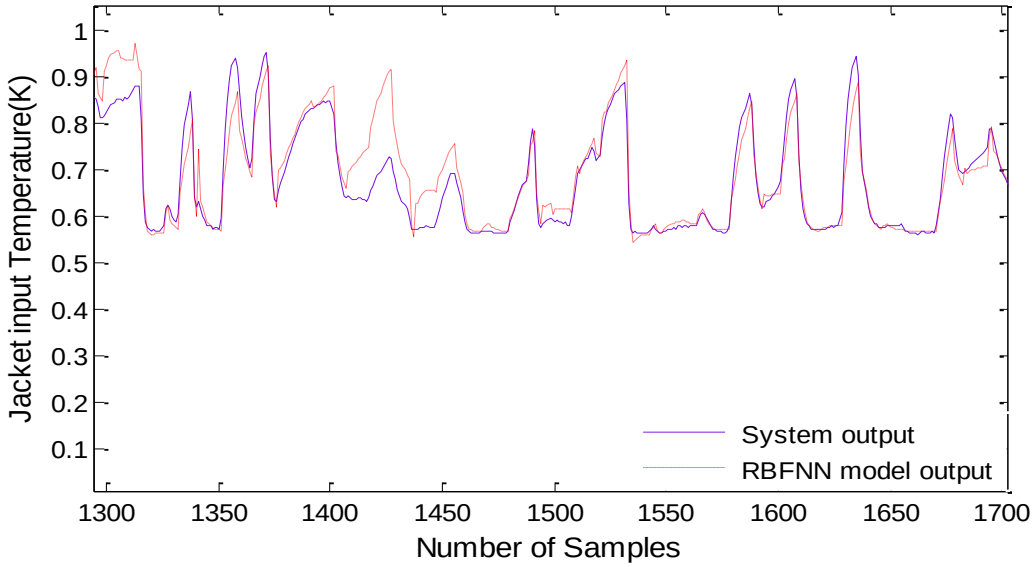


Figure 4.6 Jacket input temperature and RBF model

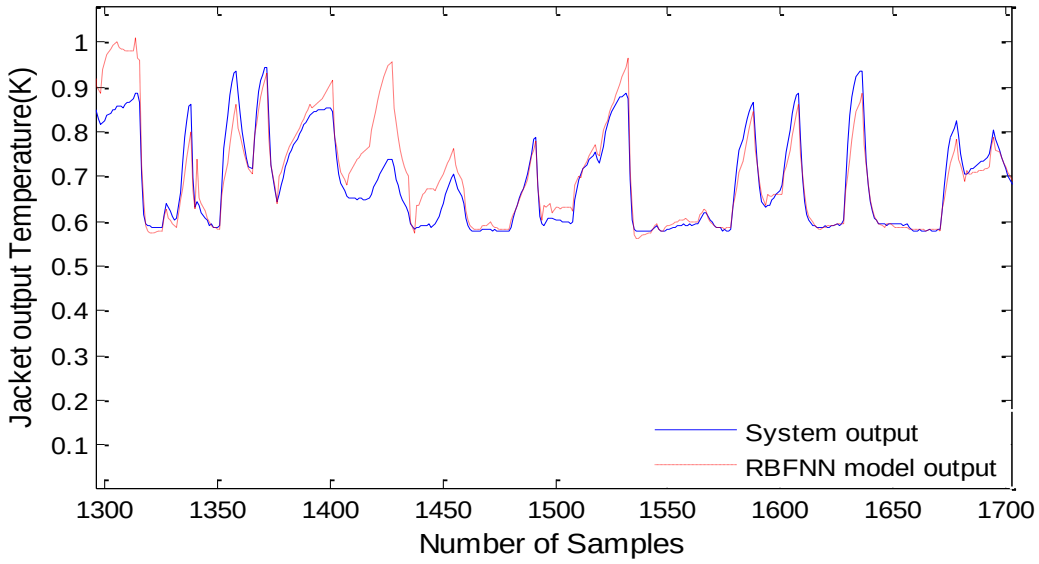


Figure 4.7 Jacket output temperature and RBF model

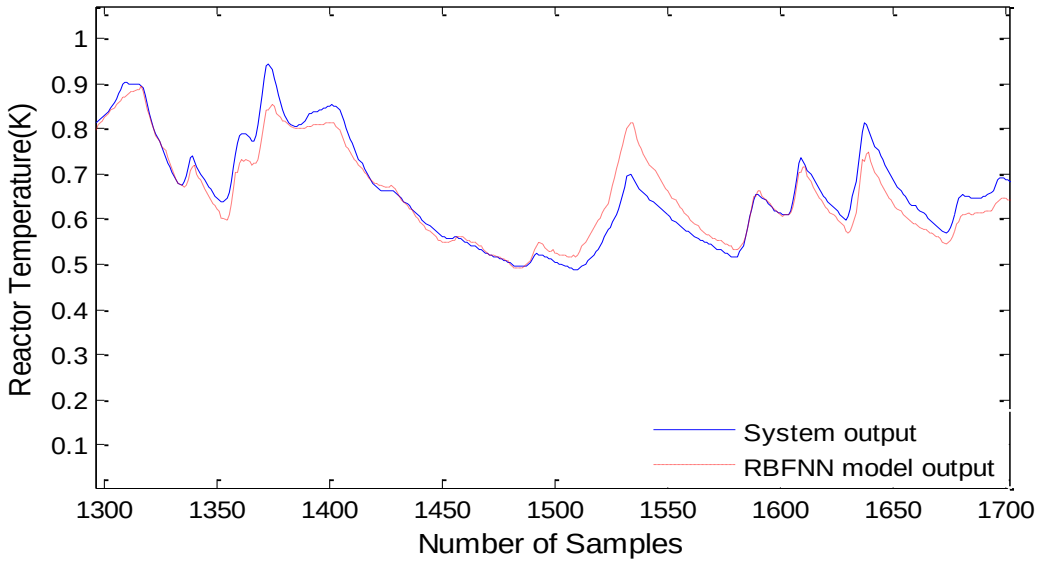


Figure 4.8 Reactor temperature and RBF model

4.3. Fault Detection

4.3.1. Simulating Faults

In this study, after training the independent RBF network model with healthy data, the model will be tested with faulty data. The faulty data is obtained by simulating different faults in the proposed reactor. These faults are classified as three sensor faults and one actuator fault. The sensor faults are jacket input temperature sensor fault, jacket output temperature sensor fault, and reactor temperature sensor fault, and the actuator fault is the inlet temperature. These faults are simulated as following:

- **Simulating Sensor Faults**

The jacket input temperature sensor fault is superimposed with 10% change of the measured jacket input temperature, and simulated from the sample number 400 to 500, as shown in Figure 4.9 and 4.10. Additionally, the jacket output temperature sensor fault is superimposed with 10% change of the measured jacket output temperature, and simulated from the sample number 600 to 700, as shown in figure.4.9 and 4.10. Furthermore, the sensor fault of the reactor temperature is superimposed with 10% change of the measured

temperature, and simulated from the sample number 800 to 900, as shown in figure.4.9 and 4.10.

- **Simulating Actuator Fault**

The heating-cooling function is influenced by an equal-percentage valve with valve position. When the valve position $c<50\%$, cooling water with inlet temperature (278.71 k) is inserted into the cooling jacket. When the valve position $c>50\%$, steam with temperature (449.82 k) is injected into the recirculating water stream, which will lead to heating up of the coolant. Consequently, it is assumed here that a failure in the pump position of cooling mode has occurred, which leads to increase in the temperature by 10% change of the measured inlet temperature. This inlet temperature fault is simulated from the sample number 1000 to 1100, as shown in figure.4.9 and 4.10.

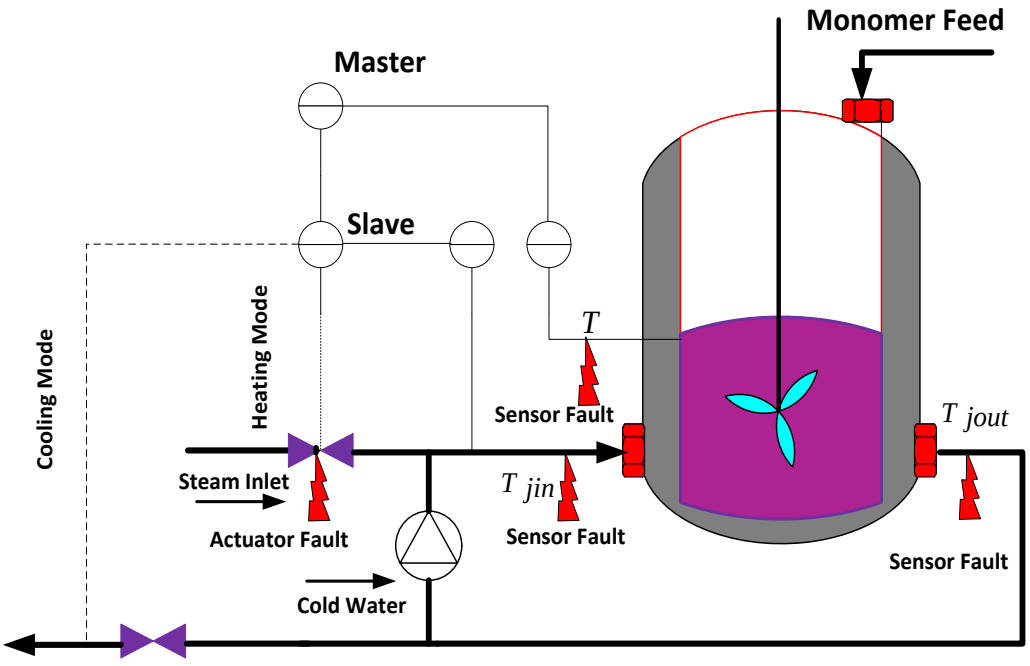


Figure 4.9 The schematic of Chylla-Haase reactor with four faults

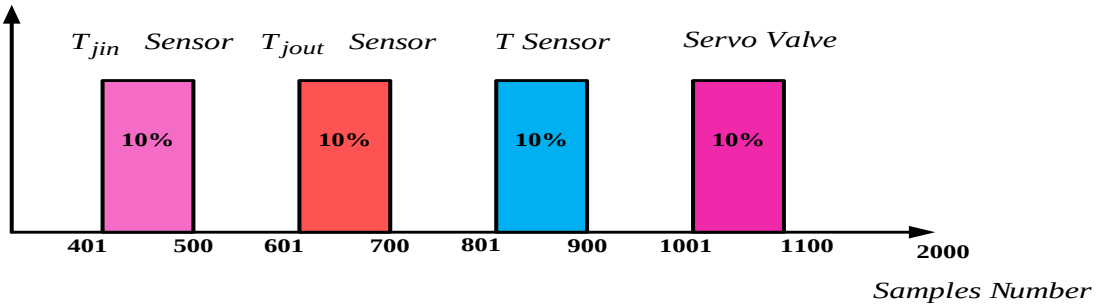


Figure 4.10 Fault structure with respect to number of samples

4.3.2. Residual Generation

After training the network model with healthy random data, as described in the previous section, all four faults were simulated to the reactor model. Then, with another set of 2000 samples, faulty square data is collected. These faulty data are collected by designing a set of square waves for all inputs.

These five inputs are the system inputs (monomer feed rate, manipulated variable) included the uncertainties and disturbances in the process. The second step towards developing a neural network model of the process is to determine the network input variables and the input vector and output vector. The network input vector consists of the past values of the five system inputs and the past values of the three system outputs. Where the $m_M(t), 1/h_f, T_{amb}$, i, and $c(t)$ are the inputs of the system; and jacket input temperature $T_{jin}(t)$, jacket output temperature $T_{jout}(t)$ and reactor temperature $T(t)$ are the outputs of the system. Moreover, the collected data is scaled linearly. After determining and scaling the input and output vectors of the system, the multivariable NARX is used to represent the non-linear dynamics of the reactor, the maximum lags in the output were selected as 3, the maximum lags in the input is selected as 2, and the maximum time delay in the inputs is selected as 2, as described in equation (4.9). Here again the neural network is realised by a RBF network with Gaussian basis functions. Moreover, the centres are chosen again using a K-means clustering algorithm and the widths are chosen using p-

nearest-neighbours. Different numbers of hidden nodes, such as 21, 31, and 51, are used in order to get good results. The recursive least squares algorithm is used to update the weight matrix. The parameters of the recursive least algorithm are selected as follows: $\mu = 0.999$, $w(0) = 10^{-5} * U(n_h, 5)$ and $p(0) = 10^5 * I(n_h)$ where μ is the forgetting factor, I is an identity matrix, U is the element unity matrix, and n_h is the number of hidden layer nodes. The RBF network model is tested with these faulty square data to generate fault–detection residuals. The filtered model prediction errors are shown in figure 4.10, 4.11, and 4.12. In this study, the residual ε is generated as the sum-squared filtered modelling error as follows:

$$e(t) = [y(t) - \hat{y}(t)]$$

$$\varepsilon(t) = \sqrt{(e_{T_{jin}})^2 + (e_{T_{jout}})^2 + (e_T)^2}$$

The residuals of testing the neural model are slightly bigger than the residuals of training the neural model. The mean absolute error (MAE) index is used to evaluate the modelling effects. The MAE for the jacket input temperature, jacket output temperature and reactor temperature are 0.004, 0.0054and 0.0072, respectively. Figure 4.11, 4.12, and 4.13 demonstrate the residuals after using a low pass filter. The first model prediction error of jacket input temperature is shown in figure 4.11 and that for jacket output temperature and reactor temperature are shown in figure 4.12 and figure 4.13, respectively. It can be observed that the independent network model output is not influenced by any type of fault, because an independent model does not use past faulty measurements as inputs. Thus, it can be clearly noticed that all faults have been clearly detected since all signals are over the threshold setting, the detection threshold is chosen as 0.2 for jacket input sensor fault ,0.3 for jacket output sensor fault and 0.7 reactor temperature fault. Moreover, no false alarms are thereby produced, so this verifies that the proposed scheme has shown excellent diagnostic performance.

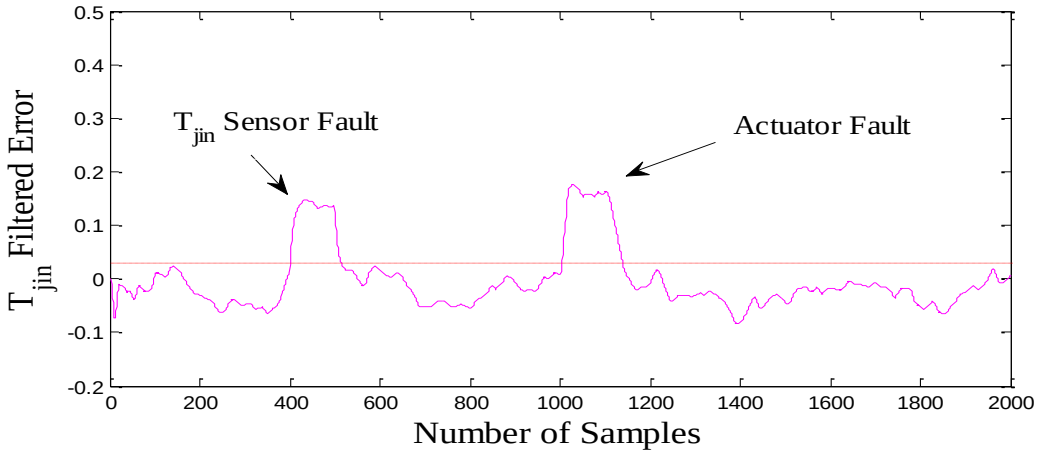


Figure 4.11 Residual filtered model prediction error of Tjin

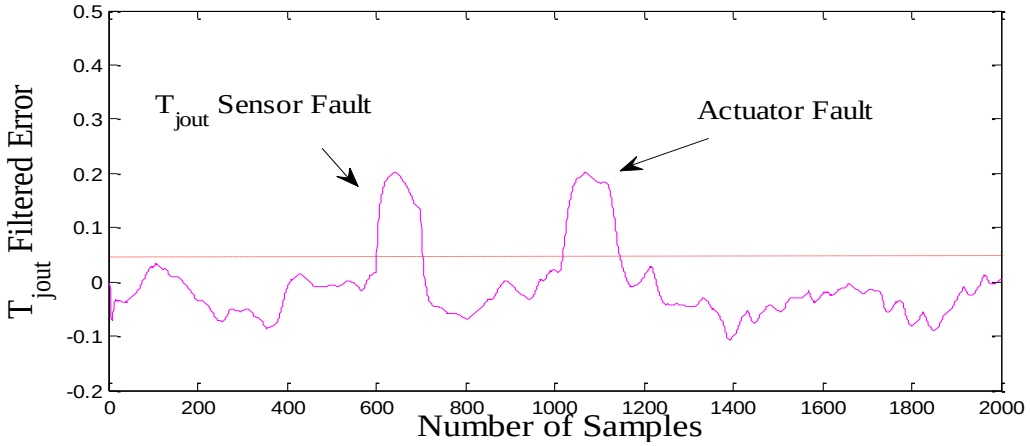


Figure 4.12 Residual filtered model prediction error of Tjout

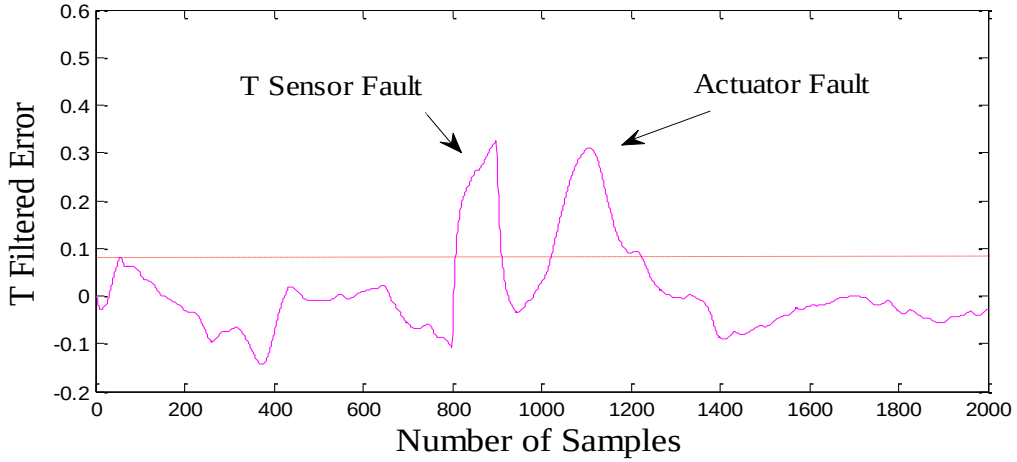


Figure 4.13 Residual filtered model prediction error of T

4.4. Fault Isolation

Figure 4.14 illustrates the fault isolation strategy; an additional neural network is applied as a classifier for fault isolation. The application of NNs for fault isolation has been used by many researchers, such as Patton et al. (1994) and Yu et al. (1999) used an RBF network, Yu et al.(1996a) using an MLP network, and Patton and Benkhedda (1996) used a B-spline network. In the fault detection, a residual is generated to report a fault occurring. However, it is difficult to identify which fault has occurred among all pre-specified possible faults using the residual, due to the fact that the residual is a scalar and carries little information about fault types. In this work, it is proposed to isolate faults according to model prediction errors. The model prediction errors are multi-dimensional, three-dimension in this case, and different faults will have different impacts on these vectors in three-dimension vector space. Classification of these features of different faults on the model prediction error vectors will lead to classification of different faults. Therefore, the faults that have occurred can be isolated. In this work, the neural classifier is developed by an RBF network with Gaussian basis functions. The residuals that shown in Figure 4.11, 4.12, and 4.13. which are the difference between the real system output and the tested neural output were used as inputs for RBF network classifier.

Moreover, the neural classifier was developed with five outputs, with four outputs associated to the four faults, and one output for (no-fault) case. The centres are chosen again using a K-means clustering algorithm and the widths are chosen using p-nearest-neighbours. Different numbers of hidden nodes, such as 51, 151, and251, are used in order to get good results. Finally, 51 hidden layer nodes are selected and the centres are chosen as 51. The parameters of the recursive least algorithm are selected as follows: $\mu = 0.9999$, $w(0) = 10^{-6} * U(n_h, 5)$ and $p(0) = 10^{6} * I(n_h)$ The samples arranged for fault occurrence are illustrated in Table 3. Moreover, the target is set such that all four outputs are set as

zero for the healthy condition data, and one output is set as 1 for a specific fault, with the others remaining at zero. Thus, once the first output is 1 and the other outputs are zero, this means that the jacket input temperature sensor fault with 10% change has occurred.

In the same way, the jacket output temperature sensor fault with 10% is believed to have occurred when the second output is 1, while the others remain at zero. Similarly, the reactor temperature sensor fault and the inlet temperature actuator fault with 10% changes will have occurred when the third and the forth outputs are 1. After training, the RBF network classifier is tested with another set of faulty data with the same arrangement of training data. The samples arranged for fault occurrence can be different from those of the training data. Table 4.1 shows the classification of faults with respect to the number of samples. The four outputs of the neural classifier after use of a filter are displayed in figure 4.15-4.18. It can be clearly noticed that all faults have been clearly detected and isolated. The isolation thresholds are chosen as 0.4 for all cases as shown in Figures 4.15-4.18.

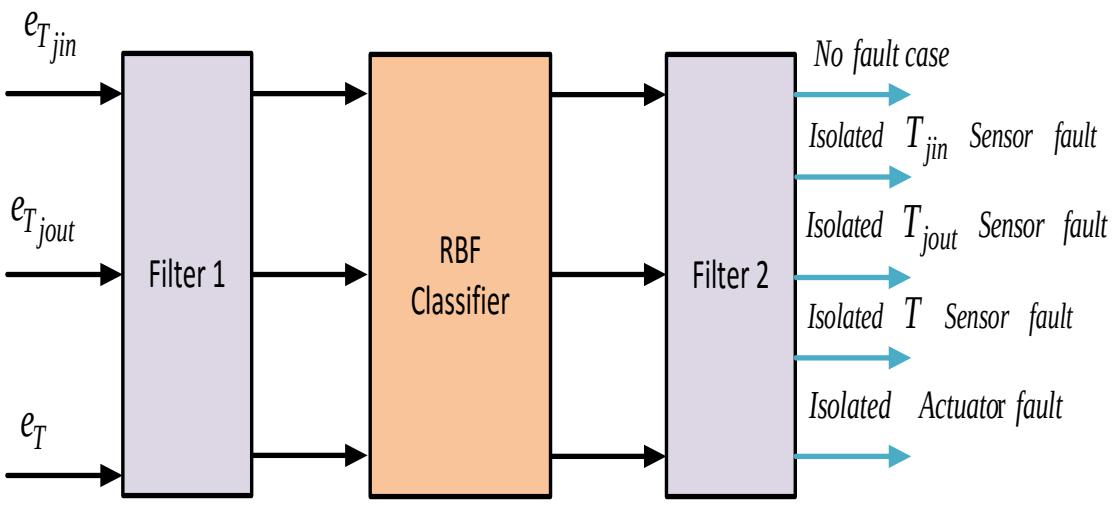


Figure 4.14 Block diagram for fault isolation

Table 4.1 Classification of faults with respect to number of samples

Faults	Number of samples
No fault	0 ~ 400
T_{jin} sensor fault	401 ~ 400
No fault	501 ~ 600
T_{jout} sensor fault	601 ~ 700
No fault	701 ~ 800
Reactor temperature sensor fault	801 ~ 900
No fault	901 ~ 1000
Inlet temperature actuator fault	1001 ~ 1100
No fault	1101 ~ 2000

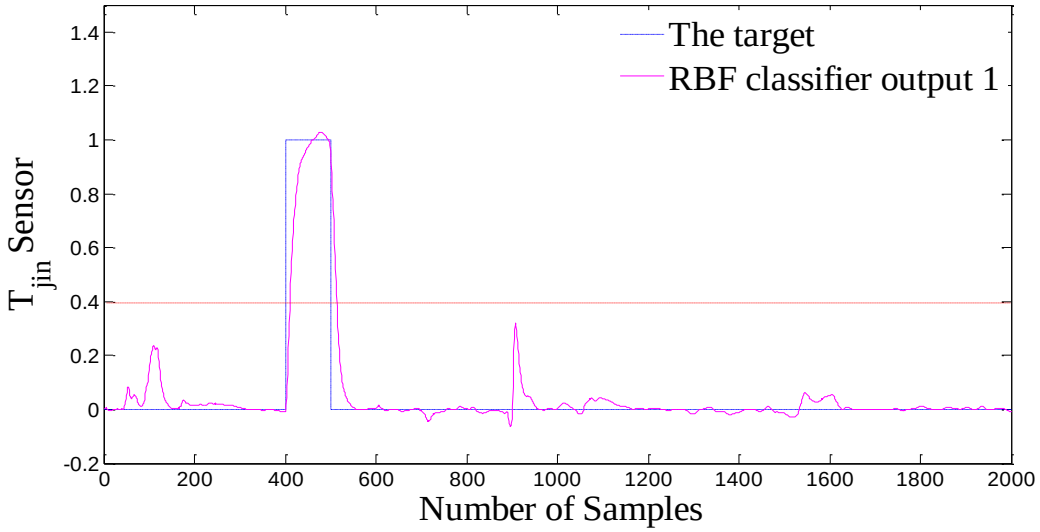


Figure 4.15 Classifier output 1

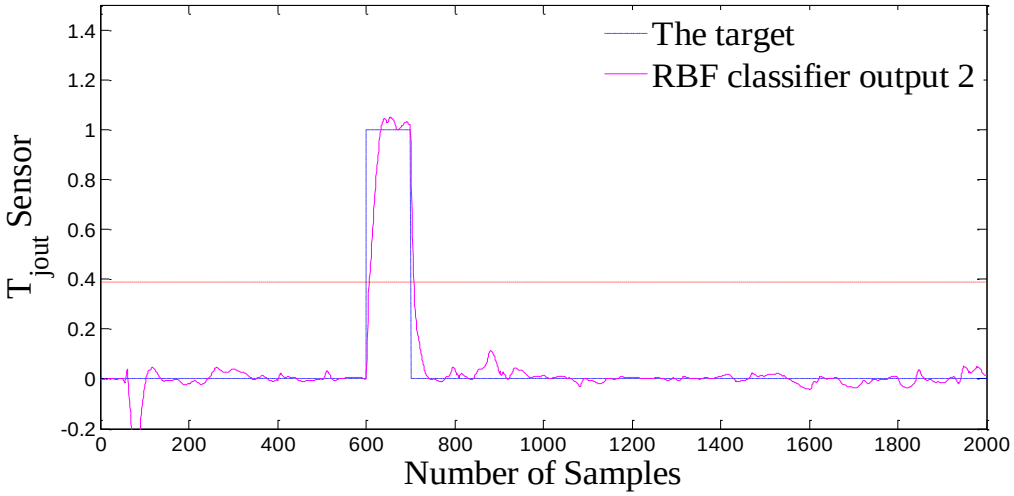


Figure 4.16 Classifier output 2

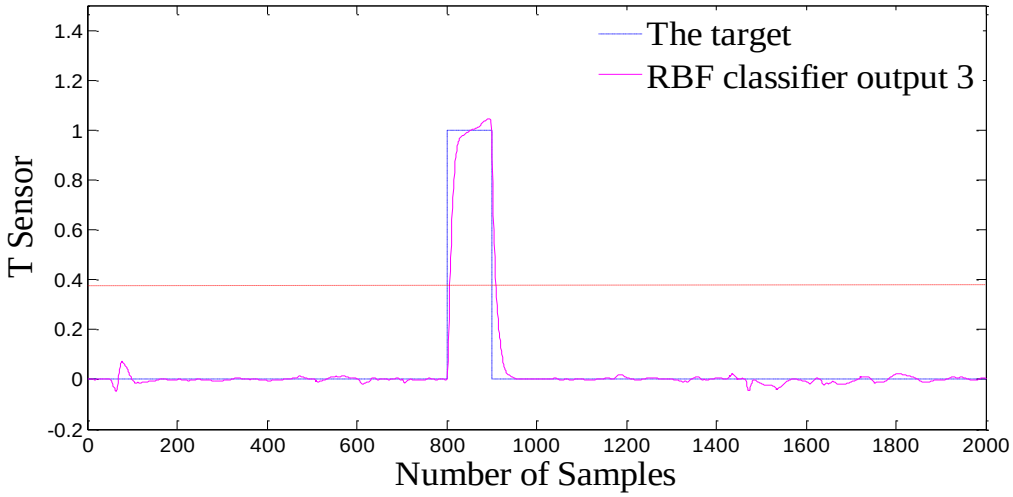


Figure 4.17 Classifier output 3

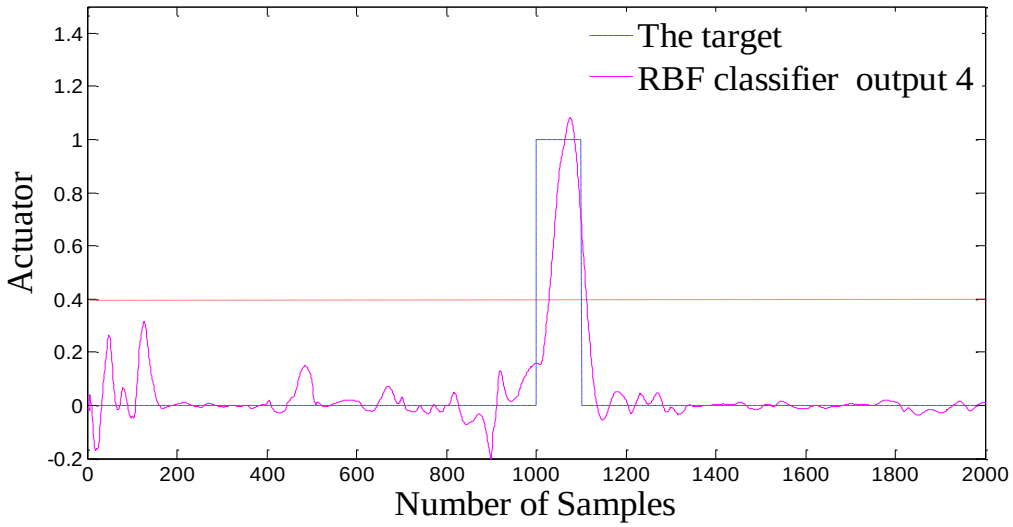


Figure 4.18 Classifier output 4

4.5. Discussion

Robust is that the fault detection always sees residual sensitive to the fault but insensitive to the disturbances. When the disturbances come in it will not affect the report of the fault, and will not increase false alarm reading. False alarm reading is that where there is no fault but fault is reported and when there is a fault but is not reported. False alarm reading should be zero percentage when all faults are reported, if there is no fault but report affected by disturbances this should be zero, but if not zero then should be reduced as small as possible.in this research work, when collecting training data all disturbances are simulated, because of this the model is trained considering the disturbances. When disturbances happened will not affect the residual that because the disturbances in this system is not big enough to make the residual high, in this process the disturbances just change the nonlinear function of the system and that is big enough from the control point of view. Its observed from simulation results that all faults have been clearly detected and isolated, and no false alarm were thereby produced, so this verifies that the proposed scheme has shown an excellent performance. Note that the outputs are not zero when no faults occur, as a result of the effects of the disturbances.

4.6. Summary

A new robust fault diagnosis scheme has been developed for open-loop Chylla-Haase reactor using an independent RBFNN. Three sensor faults and one actuator fault have been simulated on the reactor. All the simulated faults are superimposed with 10% changes of the measured temperatures, and simulated for different numbers of samples. Moreover, the uncertainties and disturbances in the process have been simulated. Two different techniques to employ RBF neural networks for fault diagnosis have been investigated. The first technique is implementing an independent RBNN for residual generation. Moreover, the generated residuals were used for detecting actuator and sensor

faults. The second technique is applying an additional RBFNN as a classifier to perform the classification task for residual evaluation and therefore to diagnose and isolate the actuator and sensor faults from the generated residuals. The simulation results show that all faults were clearly detected and isolated. Moreover, no false alarms are thereby produced, so this verifies that the proposed scheme has shown excellent diagnosis performance. The main contribution of this work is to show how to apply an independent RBFNN to open-loop Chylla-Haase reactor fault diagnosis.so this proposed method can contribute to the safety of chemical reactors.

Chapter 5

RBF Model Based FDI for Closed-Loop System

In this chapter, a new robust fault detection and isolation (FDI) scheme is developed to monitor a multivariable nonlinear chemical process called the Chylla-Haase polymerization reactor, when it is under the cascade PI control. The scheme employs a radial basis function neural network (RBFNN) in an independent mode to model the process dynamics, and using the weighted sum-squared prediction error as the residual. The recursive orthogonal Least Squares algorithm (ROLS) is employed to train the model to overcome the training difficulty of the independent mode of the network. Then, another RBFNN is used as a fault classifier to isolate faults from different features involved in the residual vector. Several actuator and sensor faults are simulated in a nonlinear simulation of the reactor in Simulink. The scheme is used to detect and isolate the faults on-line. The simulation results show the effectiveness of the scheme even the process is subjected to disturbances and uncertainties including significant changes in the monomer feed rate, fouling factor, impurity factor, ambient temperature and measurement noise. The simulation results are presented to illustrate the effectiveness and robustness of the proposed method.

5.1. Closed-loop control system design and performances

In order to produce polymer of desired quality a very tight temperature control is essential for the reactor. The controller should be able to keep the reactor temperature T within

an interval of $\pm 0.6K$ around the desired set-point under all operating conditions and disturbances. Commonly used for a chemical reactor is a PI cascade control structure. The block diagram of the cascade PI control is shown in Figure 5.1. The master control regulates the reactor temperature T by manipulating the set point T_j^{set} of the mean cooling jacket temperature T_j.The slave controller adjusts the valve position c in order to control the mean jacket temperature T_j set by the master controller.

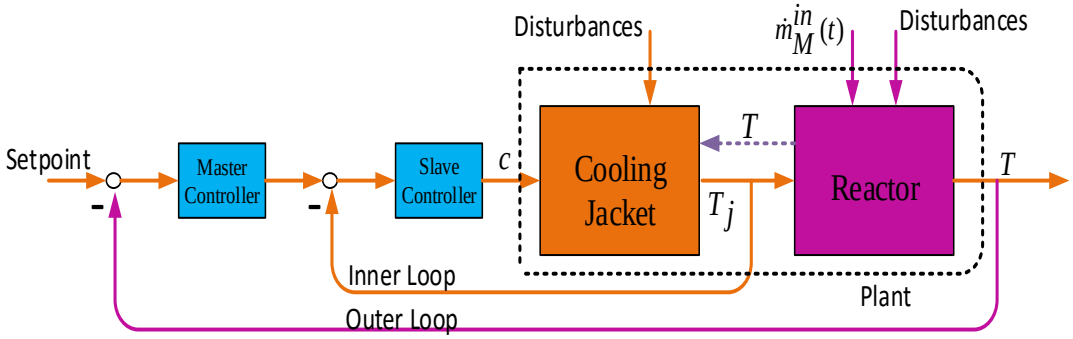
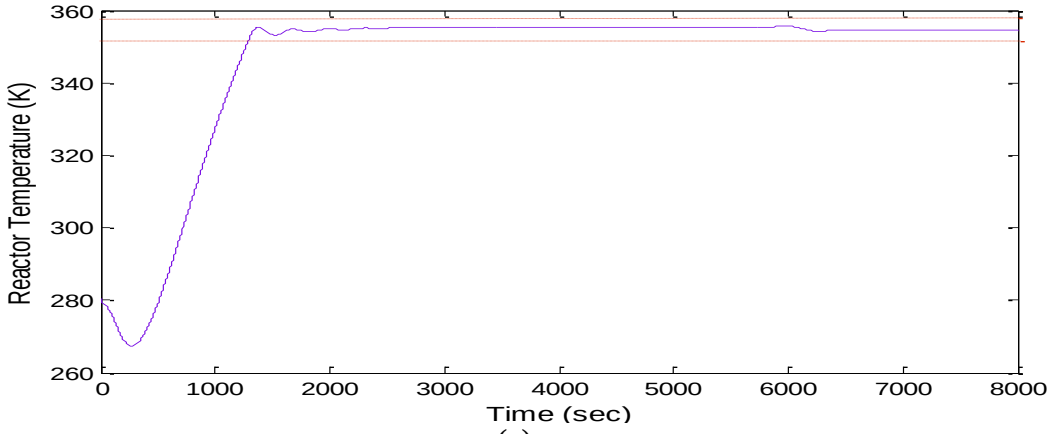


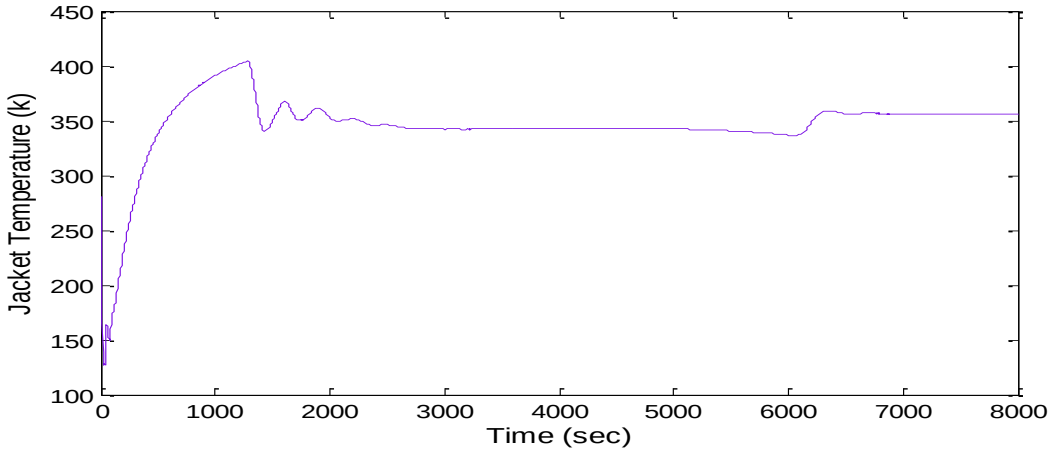
Figure 5.1 Block diagram of CASCADE control scheme

The parameters of the conventional cascade PI controllers have been tuned in simulation studies as $K_P = 21$, $K_I = 0.08$ for the master controller, and $K_P = 2.3$, $K_I = 0.09$ for the slave controller. The sampling times for both the slave and master controllers are set to $4s$. Figure 5.2 illustrates the reactor temperature response of the designed cascade PI control for the fifth batch, where the monomer was added at t = 1200 sec and withdrawn at t = 6000 sec. As the reaction release heat energy, the control variable was reduced when the monomer was added and increased when the monomer was withdrawn. It can be observed that the control scheme is effective to maintain the reactor temperature within the tolerance interval limit $\pm 0.6K$ around the set-point under major disturbance. The PI controller tuning is not optimal (see the oscillatory response when the monomer was added), this will not affect the FDI system design and evaluation. Note that all the

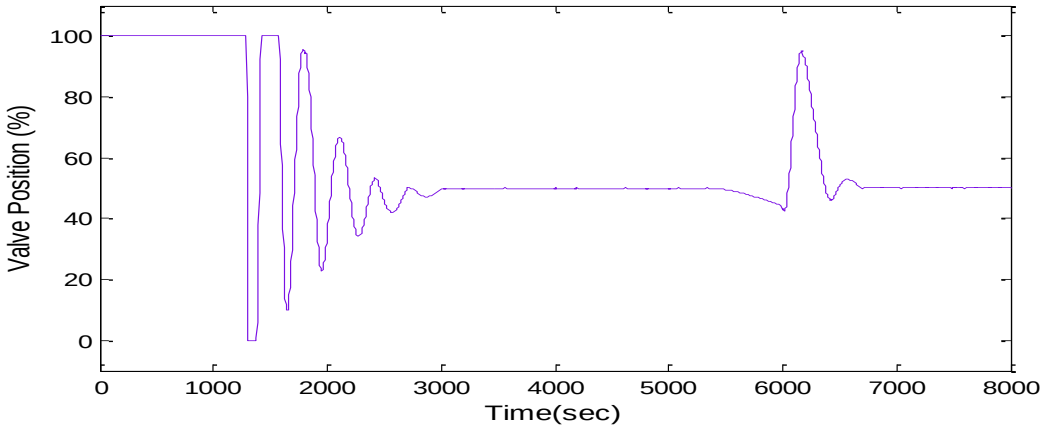
uncertainties and disturbances in the process, such as fouling factor, impurity factor, and measurement noise, have been simulated and taken into consideration.



(a)



(b)



(c)

Figure 5.2 Cascade PI control: (a) Reactor temperature, (b) Jacket temperature, (c) Valve position

5.2. Training algorithm

In this chapter, a radial basis function neural network (RBFNN) in an independent mode is employed to model the process dynamics. In this work the centres of the RBFNN are set by the K-means clustering method (Chen et al., 1990),whose objective is to minimise the sum squared distances from each input data to its closest centre so that the data is adequately covered by the activation functions $\phi(t)$. Moreover, the widths are computed by the p-nearest neighbours method(Chen et al., 1990). The excitation of each node should overlap with other nodes (usually closest) so that a smooth interpolation surface between nodes is obtained. In this method, the widths for each hidden node are set as the average distance from the centre to the p nearest centres as given by:

In this work, the weights were trained using the ROLS algorithm. Because the independent mode of RBF model requests much higher accuracy compared with dependent mode, also due to that the ROLS is a numerically robust algorithm. Training of the RBF network weights with the ROLS algorithm is as follows. Considering the network output as described in previous chapter in equation (4.3) at sample interval k for a set of N samples of input-output training data from $k-N+1$ to k ,in other words a window going back in time N samples, we have

$$Y(K)=\hat{Y}(K)+E(K)=\Phi(K)W(K)+E(K) \tag{5.1}$$

where $Y\in\Re^{N\times P}$ is the desired output matrix, $\hat{Y}\in\Re^{N\times P}$ is the neural network output matrix, $\Phi\in\Re^{N\times n_h}$ is the hidden layer output matrix, $E\in\Re^{N\times P}$ is the error matrix and equation (5.1) can be solved for $W(k)$ using the recursive MIMO Least Squares algorithm to minimize the following time-varying cost function,

$$J(k)=\left\|\begin{bmatrix}\sqrt{\lambda}Y(k-1)\\----\\y^T(k)\end{bmatrix}-\begin{bmatrix}\sqrt{\lambda}\Phi(k-1)\\----\\\phi^T(k)\end{bmatrix}W(k)\right\|_F \tag{5.2}$$

Where the *F*-norm of a matrix is defined as $\|A\|_F^2 = trace(A^T A)$ and $\lambda<1$ is used to introduce exponential forgetting to the past data. It has been shown Gomm and Yu (2000) that minimizing (5.2) is equivalent to minimizing the following cost function,

$$J(k)=\left\|\begin{bmatrix}\sqrt{\lambda}\hat{Y}(k-1)\\----\\y^T(k)\end{bmatrix}-\begin{bmatrix}\sqrt{\lambda}R(k-1)\\----\\\phi^T(k)\end{bmatrix}W(k)\right\|_F \tag{5.3}$$

Where R is an $n_h \times n_h$ upper triangular matrix, and $\hat{Y}$ is computed by an orthogonal decomposition as follows,

$$\begin{bmatrix}\sqrt{\lambda}R(k-1)\\----\\\phi^T(k)\end{bmatrix}=Q(k)\begin{bmatrix}R(k)\\---\\0\end{bmatrix},\begin{bmatrix}\hat{Y}(k)\\---\\\eta^T(k)\end{bmatrix}=Q^T(k)\begin{bmatrix}\sqrt{\lambda}\hat{Y}(k-1)\\----\\y^T(k)\end{bmatrix} \tag{5.4}$$

Where *Q* is an orthogonal matrix. Combining (5.3) and (5.4) and considering that the *F*-norm is preserved by orthogonal transformation, the following equivalent cost function is obtained,

$$J(k)=\left\|\begin{bmatrix}\hat{Y}(k)-R(k)W(k)\\---------\\\eta^T(k)\end{bmatrix}\right\|_F \tag{5.5}$$

This allows the optimal solution of $W(k)$ to be solved straightforwardly from

$$R(k)W(k)=\hat{Y}(k) \tag{5.6}$$

And leaves the residual at sample interval *k* as $\|\eta^T\|_F$. Since $R(k)$ is an upper triangular matrix, $W(k)$ can be easily solved from (5.6) by backward substitution.

The decomposition in (5.4) can be achieved efficiently by applying Givens rotations to an augmented matrix to obtain the following transformation by Gomm and Yu (2000):

$$\begin{bmatrix} \sqrt{\lambda}R(k-1) & \sqrt{\lambda}\hat{Y}(k-1) \\ \phi^T(k) & y^T(k) \end{bmatrix} \rightarrow \begin{bmatrix} R(k) & \hat{Y}(k) \\ 0 & \eta^T(k) \end{bmatrix} \tag{5.7}$$

The procedure of the ROLS algorithm is therefore the following: for on-line training, calculate $\phi(k)$ at each sampling period to update the augmented matrix and compute the Givens rotations to realize the transformation in (5.7). Then solve $W(k)$ in (5.6) with $R(k)$ and $\hat{Y}(k)$ obtained in (5.7). In this case, $W(k)$ is needed at each sample instant for prediction. Also, $\lambda<1$ is needed to follow time-varying dynamics at the current time. For use in off-line mode, the Givens rotations can be computed to realize the transformation in (5.7) continuously to the end of training, and then W is solved finally from (5.6). In this case, λ is set to 1. Initial values for $R(k)$ and $\hat{Y}(k)$ in both cases can be assigned as $R(0)=\mu I$ and $\hat{Y}(0)=0$, where μ is a small positive number, and I is a unity matrix with appropriate dimension.

5.3. RBF model development

The first step is to obtain training data. When acquiring training data, the excitation signal should be designed such that the training data has the persistently exciting property and should span over the entire network input space in every dimension, which can provide a good network model interpolation property and good generalization. A set of modified random amplitude signals (RAS) were designed for monomer feed rate, fouling factor, ambient temperature and impurity factor as shown in figure 5.3. the fifth input is the valve position which is the controller output and it cannot be designed. The second step in developing the RBF model of the process is to determine the network input variables. The network input variables consist of the input vector and output vector. According to the

reactor dynamics, the input vector was determined to include the five process inputs: monomer feed rate, fouling factor, ambient temperature, impurity factor and the fifth input is the controller output. The controller output cannot be designed when the reactor is under closed-loop control; this is one of the problems in closed-loop identification. In practice most systems work under closed-loop control. Most chemical processes operate as a part of a control configuration, and the control action will correct small changes of the states caused by faults. FDI system design for a plant itself or for the plant under closed-loop control would be quite different. The major difference lies in that the operating point for the closed-loop control system is in a small range while for an open-loop plant is the whole operating space. The FDI has been investigated in this paper for the chemical reactor under cascade control. The output vector was determined to include the three system outputs, jacket input temperature, jacket output temperature and reactor temperature. Therefore, the input and vector and output vector that used to determine the RBFNN input variables are shown in (5.8).

$$u = \begin{bmatrix} m_M \\ 1/h_f \\ T_{amb} \\ i \\ C \end{bmatrix}, \quad y = \begin{bmatrix} T_{jin} \\ T_{jout} \\ T \end{bmatrix} \tag{5.8}$$

Before training and testing, the input vector and output vector were scaled linearly into the range of [0 1] using the formulae (5.9). Then, in order to implement the proposed network in an independent mode, the network input vector *x* used the past value of the system output as mentioned in previous chapter in equation (4.6). Different lags and time delays have been tried, and one giving minimal model prediction error was used in the model development. The maximum lag in the output and the input are selected as 3 and 2 respectively. The time delay in the inputs is selected as 2, as described in equation (5.10).

Thus, the RBF model is designed to have 19 inputs and 3 outputs, as shown in Fig.8. The RBF model is implemented using Matlab. Different numbers of hidden layers' nodes, such as 21, 31, and 51, were used in order to get good results. Finally, 21 hidden layer nodes were selected with the centres being chosen using the *K*-means clustering algorithm. Moreover, the *P*-nearest-neighbours algorithm was used to choose the widths, and the ROLS algorithm was used to update the weight matrix. A data set of total 2000 samples was collected from the Simulink model of the closed-loop system, and 4 secs was used as the sampling time. The first 1500 samples were used for training the network model, and the remaining 500 samples were used for the model test.

$$u_{scaled}(k) = \frac{u(k) - u_{\min}}{u_{\max} - u_{\min}} \quad y_{scaled}(k) = \frac{y(k) - y_{\min}}{y_{\max} - y_{\min}} \tag{5.9}$$

$$x(t) = [\hat{y}(t-1)\ \hat{y}(t-2)\ \hat{y}(t-3)\ u(t-k-1)\ u(t-k-2)]^T \tag{5.10}$$

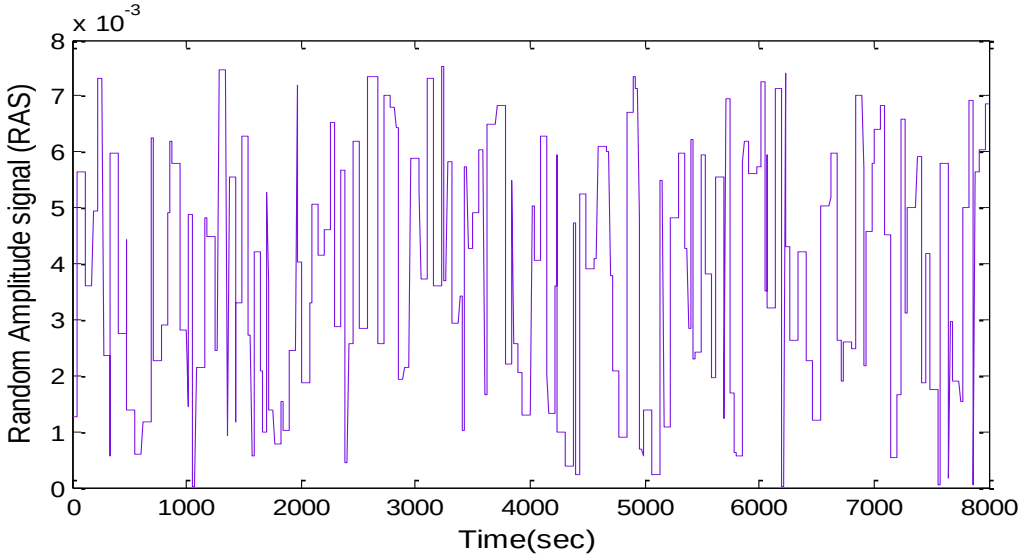


Figure 5.3 RAS signal

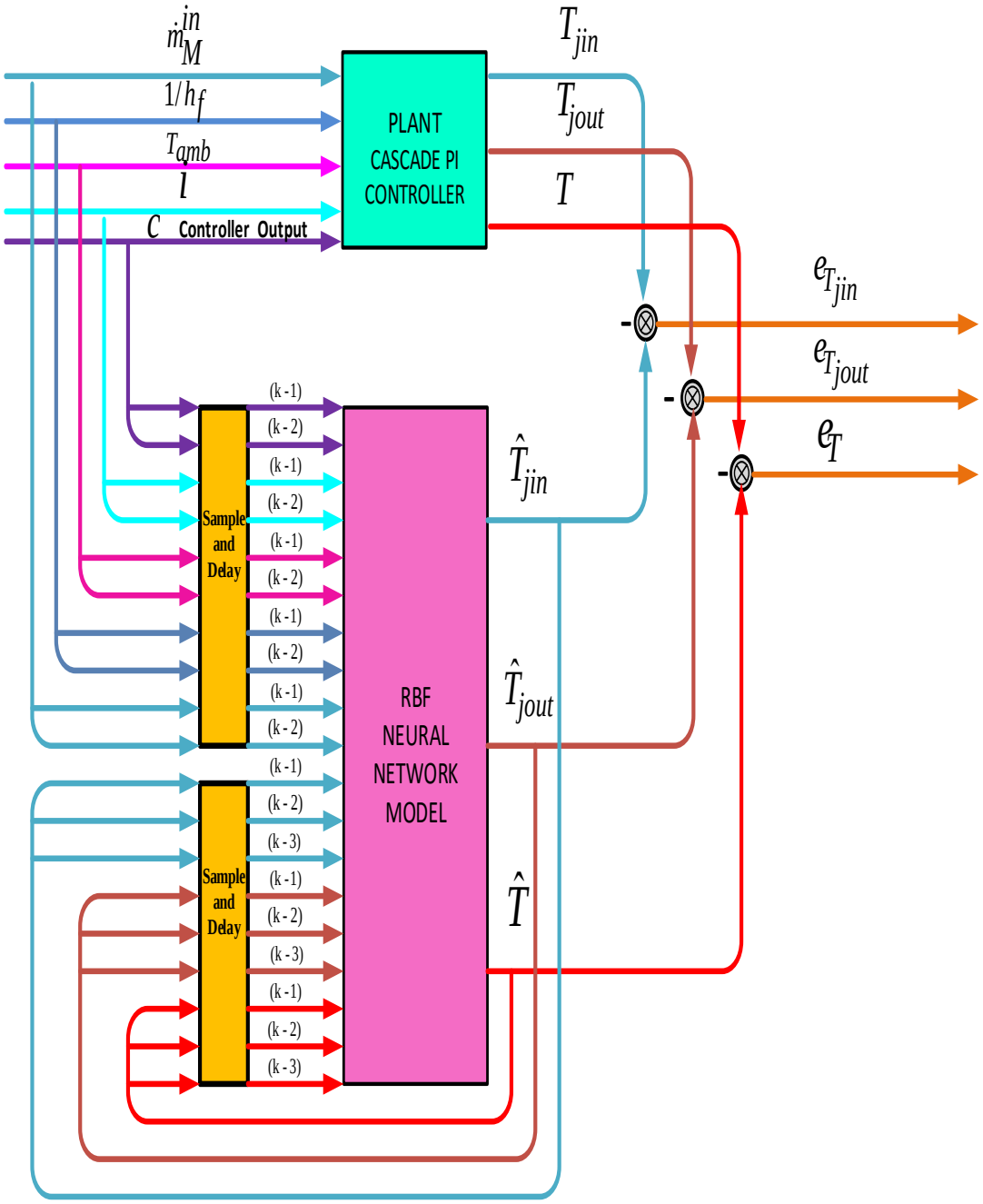


Figure 5.4 Structure of FD using an independent RBFNN

5.4. Residual generation

In this chapter, after training the independent RBF network model with healthy data, the model will be used to detect faults that occurred in the system, i.e. generate residual when the system is subjected to any fault. The faulty data is obtained by simulating different faults in the proposed reactor. The type and classification of faults that used in this work are similar to that used in previous chapter. Figure 5.4 demonstrates the fault detection approach. An independent model is implemented in parallel with the system to generate the residuals for detecting the sensor and actuator faults in the reactor. After training the network model with healthy random data, as described in the previous section, all four faults were simulated to the reactor model. Then, the fault detection is conducted with the network model using another set of 2000 samples faulty square data. These faulty data were collected when the system is given a set of designed square waves for monomer feed rate, fouling factor, ambient temperature and impurity factor as shown in figure 5.5-5.8 to simulate the realistic situation in the practical applications. The fifth input is the controller output which cannot be designed and with smaller amplitude is added to the controller output to excite the dynamics in different frequencies. Then, the input vector x of the independent RBFNN designed as shown in (5.10). Testing the proposed model was done many times with different sets of faulty square data, to ensure the efficiency performance of the proposed network model. Different numbers of hidden nodes, such as 21, 31, and 51, were used in order to get good results.

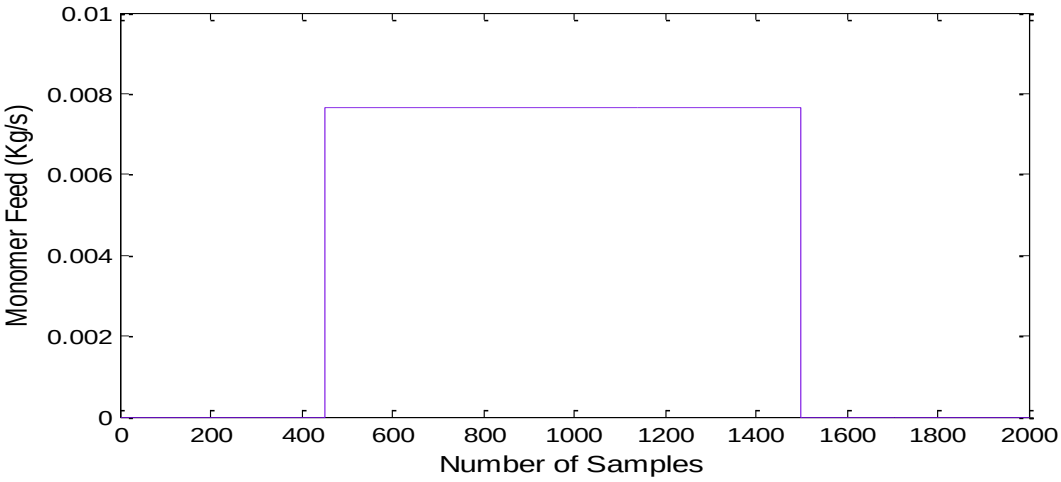


Figure 5.5 Monomer feed rate

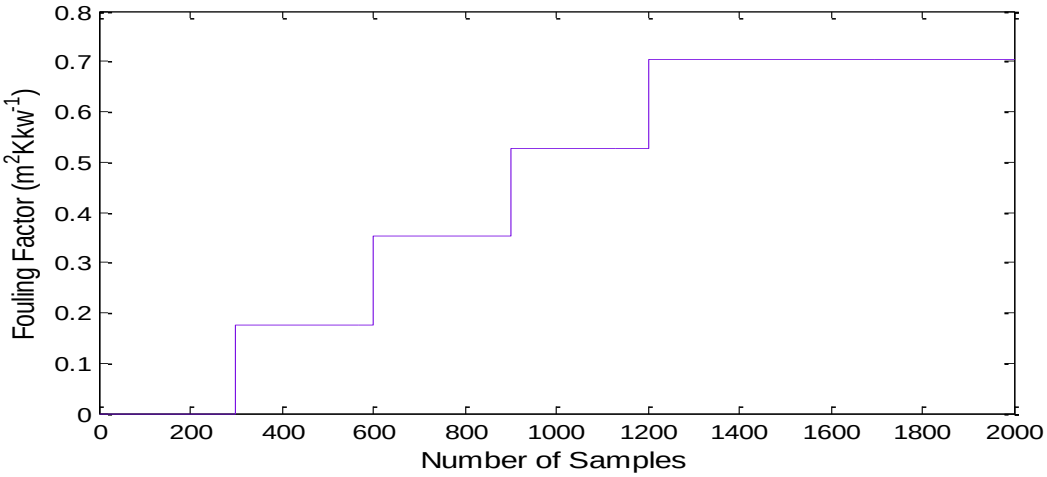


Figure 0.6 Fouling factor

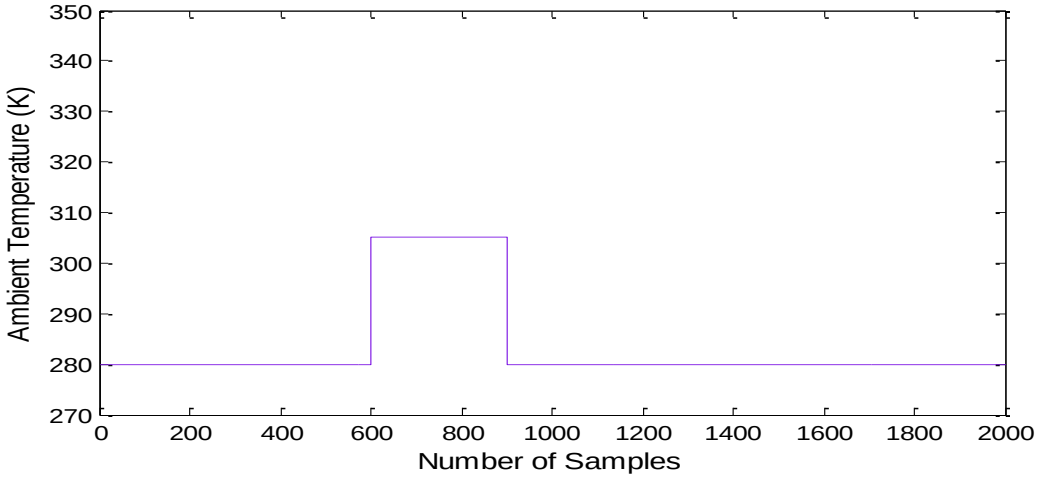


Figure 5.7 Ambient temperature

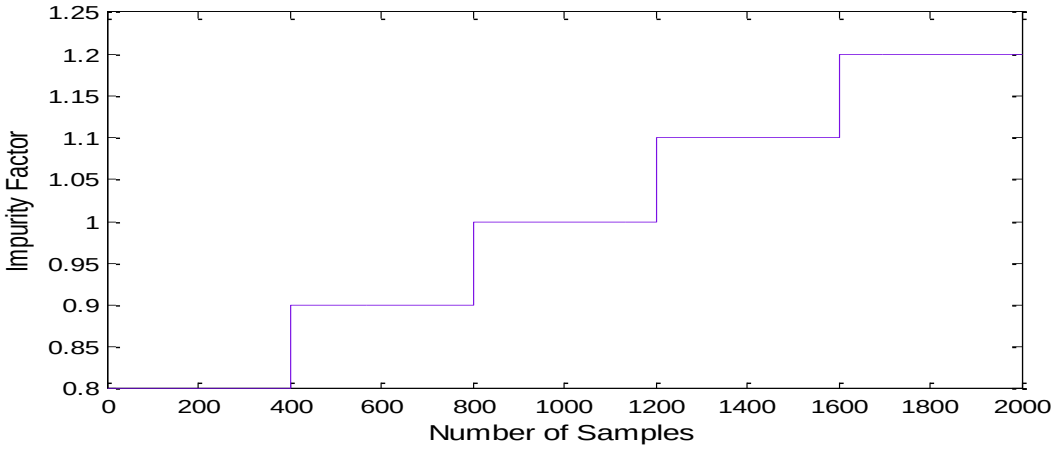


Figure 5.8 Impurity factor

The filtered model prediction errors are shown in figure 5.9-5.12. The first model prediction error of jacket input temperature is shown in figure 5.9 and that for jacket output temperature and reactor temperature are shown in figure 5.10 and 5.11 respectively. It can be observed that the independent network model output is not influenced by any type of fault. Therefore, it can be clearly noticed that all faults have been clearly detected, since the faults are over the chosen thresholds. Here in this section the thresholds are chosen as (+0.1/-0.1) for all cases. Moreover, no false alarms were thereby produced, so this verifies that the proposed scheme has shown excellent diagnostic performance. The model prediction errors of the FD are slightly bigger than the modelling prediction errors of training the neural model. The mean absolute error (MAE) for the jacket input temperature, jacket output temperature and reactor temperature are 0.004, 0.0054and 0.0072, respectively.

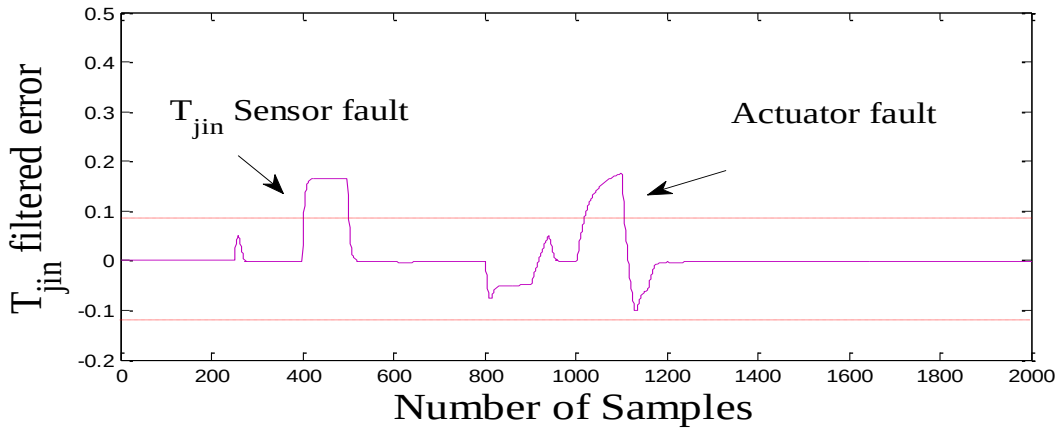


Figure 5.9 Filtered residual model prediction error of Tjin

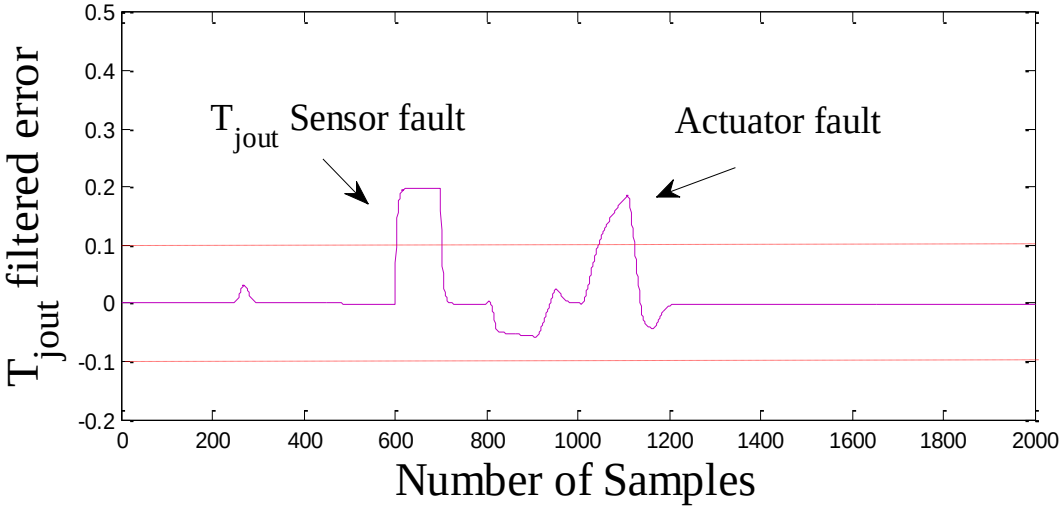


Figure 0.10 Filtered residual model prediction error of Tjout

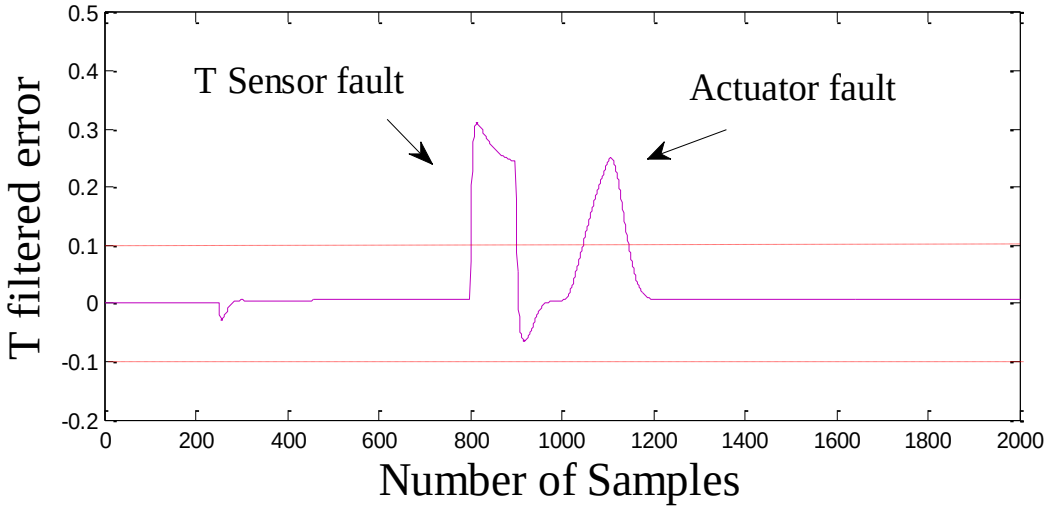


Figure 5.11 Filtered residual model prediction error of T

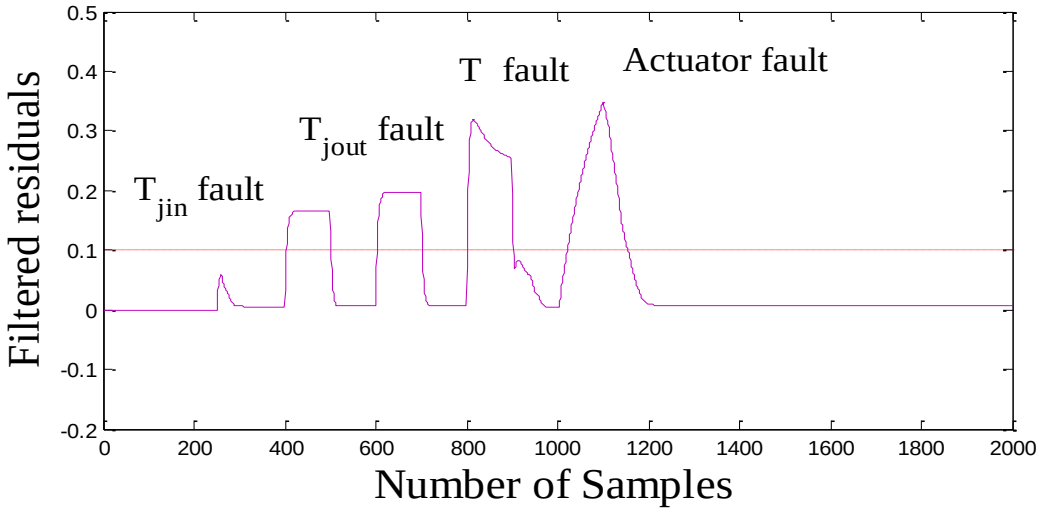


Figure 5.12 Filtered sum-squared residuals

Since the independent model does not use past faulty measurements as inputs. It is observed that the neural model outputs did not track the faulty system outputs. Thus, the residuals are sensitive to these faults, and consequently can be used to detect faults in the presence of noise and modelling errors. A pre-specified threshold ρ is marked in figure 5.9-5.12, the value of ρ is determined according to the specific application and is directly related to the noise level in the system and the level of modelling error in nominal condition. A lower value of the threshold will increase the false-alarm rate, while a higher value will reduce detection sensitivity. It can be clearly noticed in figure 5.9-5.12 that all faults have been clearly detected, and no false alarm was thereby produced. So, this verifies that the proposed scheme has shown excellent detection performance and robustness against disturbance and time-varying parameters. Fault magnitudes other than 10% have also been tested and the detection results are similar. The results therefore are not presented here for limited space.

5.5. Fault Isolation

In fault detection, a residual is generated to report a fault occurring. However, it is difficult to identify which fault has occurred among all pre-specified possible faults using

the residual, due to the fact that the residual is a scalar does not carry direct information about fault types. In this work, it is proposed to isolate faults according to model prediction errors. The model prediction errors are three-dimensional in this work. Different faults will have different impacts on these vectors in three-dimension vector space. Classification of these features will lead to classification of different faults. Therefore, the faults that have occurred can be isolated. According to the above arguments, the fault isolation scheme is developed in this work and is displayed in figure 5.13.

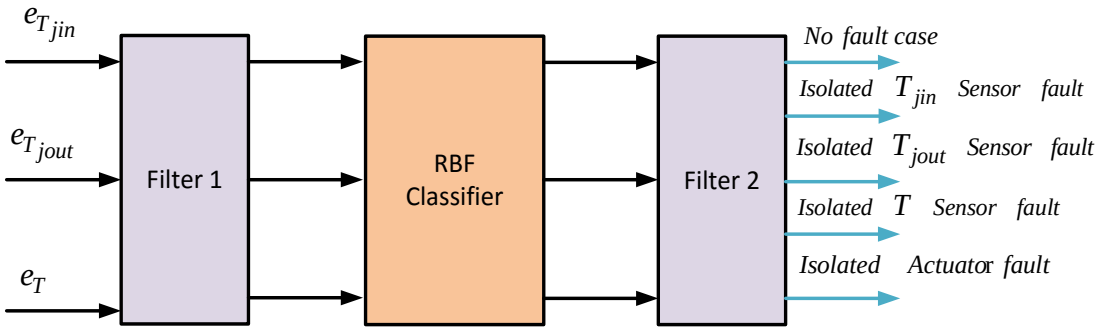


Figure 5.13 Block diagram of fault isolation

The isolation is achieved in the following way. The three model prediction error signals are used as the inputs of the classifier. The classifier has 5 outputs with each of the first 4 outputs dedicated to one fault, and the fifth output for no-fault case. The training data set contains 5 parts, with each part of the first 4 including data with one fault occurring and the fifth part for no-fault data. The training target is arranged that for each part of training data with a fault, the target for the dedicated output is “1”, while that for all the other 4 outputs are “0”. So, each output of the classifier is trained sensitive to only its corresponding fault and insensitive to the other pre-defined faults. After training, the classifier is used on-line to receive the three model prediction error signals. When the fifth output is “1” and all the other outputs are “0”, it indicates the system is healthy. If

any output among the first 4 is “1” while the others are “0”, it indicates the fault associated to this output occurs.

The network training for classification is different from that for modelling. The centres of the classifier were chosen again using the K-means clustering algorithm, so that the sum squared distance of each input data from the centre is minimized. The widths were chosen using p-nearest-neighbours. In the updating of the classifier weights, the recursive least squares (RLS) algorithm was used. The parameters of the RLS algorithm are selected as follows: μ=0.99999 , $w(0) = 10^{-6} * U(n_h,5)$ and $p(0) = 10^{6} * I(n_h)$, where μ is the forgetting factor, I is an identity matrix, U is the element unity matrix, and n_h is the number of hidden layer nodes. As the classifier was trained to classify a number of different patterns statically, a bigger number of centres than that of model were needed.in this study, different numbers of hidden nodes, such as 51, 151, and 251 were used. Finally, 51 hidden layer nodes are selected and the centres are chosen as 51. In addition to the optimization of weights using RLS algorithm, both centre locations and amplitude of width have also been optimized. As the objective function is nonlinearly related to both the centre and the width, a nonlinear optimization algorithm, the gradient descent method is employed for this task. The samples arranged for fault occurrence are illustrated in Table 3. Moreover, the target is set such that all four outputs are set as zero for the healthy condition data, and one output is set as 1 for a specific fault, with the others remaining at zero. Thus, once the first output is 1 and the other outputs are zero, this means that the jacket input temperature sensor fault occurred. In the same way, the jacket output temperature sensor fault is believed to have fault when the second output is 1 and the other outputs remain at zero. Similarly, the reactor temperature sensor fault and the valve actuator fault occurred when the third and the forth outputs are 1. After training, the RBF network classifier is tested with another set of faulty data with the same fault arrangement.

To ensure the reliable performance, the developed network classifier was tested many times with different sets of faulty data. The samples arranged for fault occurrence have been different from those of the training data. For the simulated faults shown in table 3, the four outputs of the neural classifier after use of a filter are displayed in figure 5.14-5.17. It is clearly that all faults are isolated. In isolation part here, the thresholds are chosen as 0.4 for all cases

Table 5.1 Classification of faults with respect to number of samples

Faults	Number of samples
No fault	0 ~ 400
T_{jin} sensor fault	401 ~ 400
No fault	501 ~ 600
T_{jout} sensor fault	601 ~ 700
No fault	701 ~ 800
T sensor fault	801 ~ 900
No fault	901 ~ 1000
Actuator fault	1001 ~ 1100
No fault	1101 ~ 2000

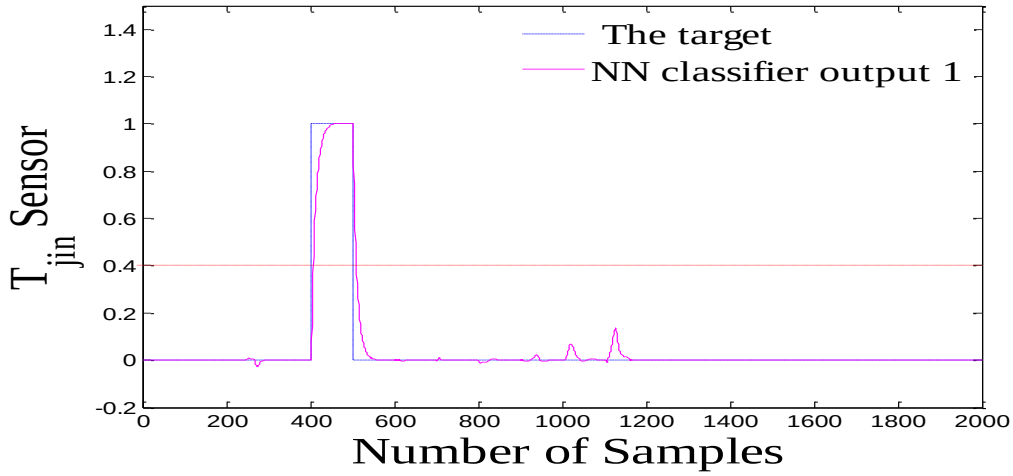


Figure 5.14 Classifier output 1

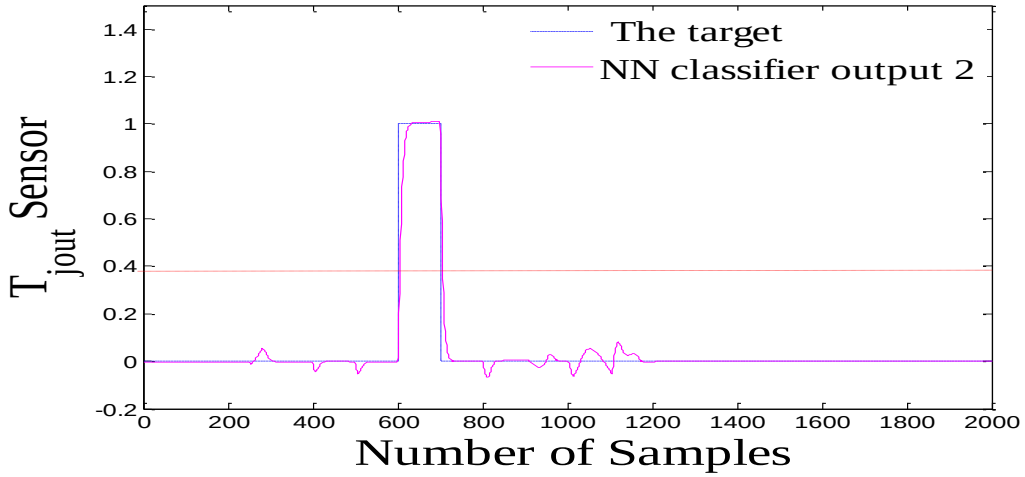


Figure 5.15 Classifier output 2

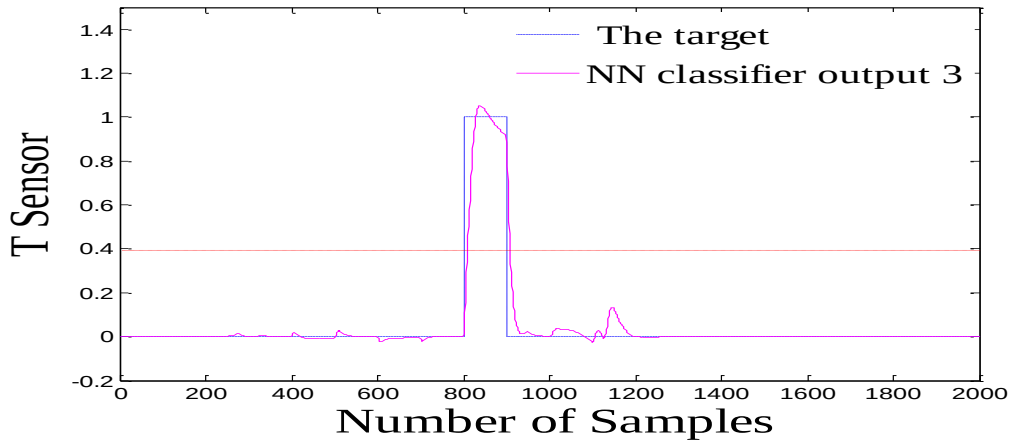


Figure 5.16 Classifier output 3

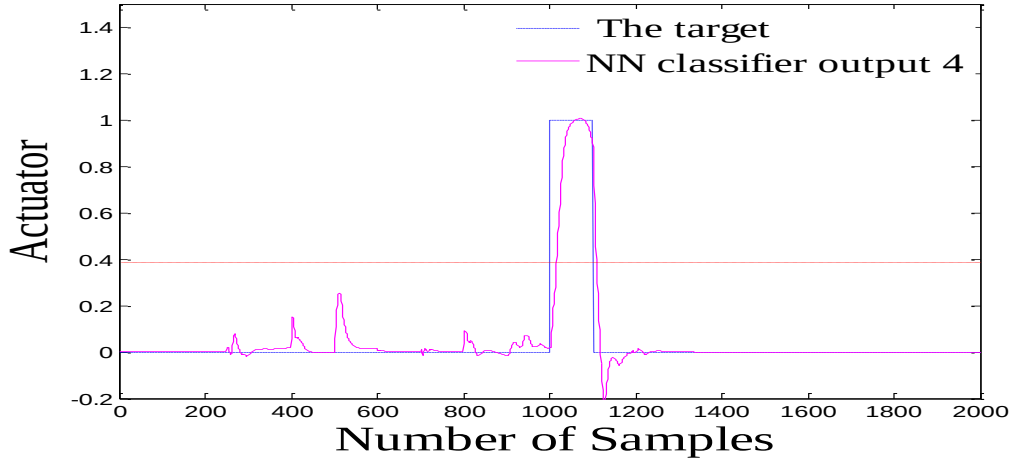


Figure 5.17 Classifier output 4

5.6. Discussion

Figure 5.14-5.17 illustrate the fault isolation results for the four faults. The classifier outputs were filtered to get rid of specks before they were used to indicate isolated fault. It is noticed that all faults have been clearly detected and isolated. Robustness of a fault detection system indicates its ability to distinguish between faults and model uncertainties or disturbances. When the disturbances come in the system it will not affect the report of the fault, and will not increase false alarm rate. False alarm is that where there is no fault but fault is reported or when there is a fault but it is not reported. In this research, the training data is acquired with all disturbances and time-varying parameters simulated. Therefore, the trained RBF model generates residual that is insensitive to these disturbances and time-varying parameters. It is observed from simulation results that all faults have been clearly detected and isolated, and no false alarm was produced. This verifies that the proposed scheme has shown an excellent performance.

5.7. Summary

A new robust fault diagnosis scheme has been developed for a Chylla-Haase reactor under closed-loop control using an independent RBF neural network model and a RBF classifier. Due to the increased difficulty in training an independent RBF model compared with the

dependent model, the network weights were updated using the ROLS algorithm. 10% changes on the three sensor outputs and one actuator output were simulated in the Chylla-Haase reactor Simulink model. Moreover, the disturbance such as the monomer feed rate, the time-varying parameters such as the fouling factor and impurity factor, and measurement noise were simulated and used. Consequently, the robustness of the fault detection to these disturbances and time-varying parameters was achieved. RBF classifier was implemented for fault isolation, where three dimension vectors of model prediction errors were used as the input for the network classifier. The different ways of faults affecting the model prediction error vector was classified, so that the occurring fault was identified. Optimisation of centre location and magnitude of the width significantly increased the classifying ability. The simulation results confirmed that the simulated faults have been clearly detected and isolated with zero false alarm rate. The research indicates the feasibility of the developed scheme applied to industrial systems, especially chemical and biochemical processes, for which the mathematical model is difficult to develop.

Chapter 6

MLP NN Based FDI for Open-Loop and Closed-Loop Systems

6.1. Multilayer Perceptron Neural Networks (MLPNN)

The MLP neural networks became the most commonly used type of feedforward neural networks after Rumelhart developed a training algorithm called back error propagation or BP algorithm (David and James, 1987, Lippmann, 1987). Typically, A MLP consists of an input layer, several hidden layers, and an output layer. Each layer contains a number of node, a neuron, which is the basic element of a neural network. A neuron is modelled as shown in figure 6.1.

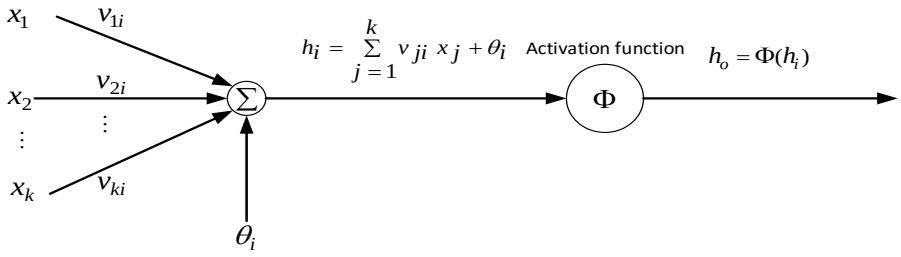


Figure 6.1 Neuron modelling

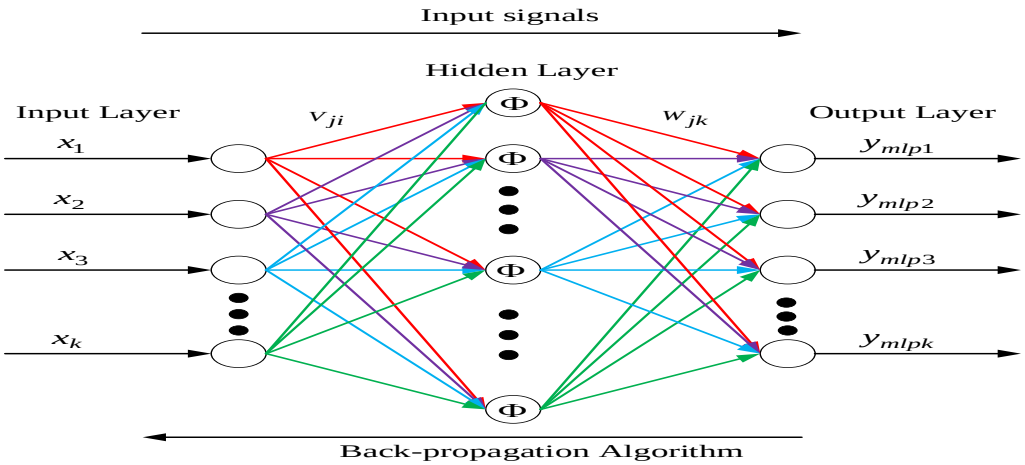


Figure 6.2 MLP NN structure

The activation h_i and the output signal h_o are obtained by Lippmann (1987).

$$h_i = \sum_{j=1}^{k} v_{ji}\, x_j + \theta_i \tag{6.1}$$

$$h_0 = \Phi(h_i) \tag{6.2}$$

Where h_i is output from each hidden neuron, $x=[x_1\ x_2 \ldots x_k]^T$ is a n by 1 input vector, $v=[v_1\ v_2 \ldots v_k]^T$ is n by 1 weight vector which connecting the input vector with the hidden layer inputs, where T denotes the transpose operation, θ is an additive bias, and Φ is the activation function. In this work the tangent sigmoid activation function is used which is defined as

$$h_o = \Phi(h_i) = \left(\frac{2}{(1+\exp(-h_i))-1} \right) \tag{6.3}$$

The output of network is given by:

$$\hat{y}_{mlp} = w_{jk}\, h_o \tag{6.4}$$

Where w_{jk} is the weight connecting the output layer and the output of the hidden neuron .

6.1.1. Learning algorithm

Since the network architecture is modelled, the first step to apply the neural network is to train the network. In this section we will discuss the development of the learning algorithm, we will briefly introduce the popular back-propagation learning algorithm.

6.1.2. Back-propagation learning algorithm

The BP learning algorithm which is a gradient decent algorithm is designed to minimize the cost function iteratively equal to the mean square difference between the desired output and actual network output (Lippmann, 1987, Yu Chang et al., 1994, David and James, 1987).the desired output of all nodes is typically "low" $(0\, or <0.1)$. The network is trained by initially selecting small random weights and then presenting all training data repeatedly. Weights are adjusted after every trail using side information specifying correct class until all weights converge and the cost function is reduced to an acceptable value. An essential component of the algorithm is the iterative method that propagates error terms required to adapt weights back from nodes in output layer to nodes in lower layer. After training pattern is added the neural network weights at each layer are updated according to the following rule(Lippmann, 1987).Use a recursive algorithm starting at the output nodes and working back to hidden layer, weigh adjusted by

$$w_{ij}(t+1)=w_{ij}(t)+\mu\,\Delta w_{ij} \tag{6.5}$$

Where μ is the learning rate and Δw_{ij} is the gradient of the error with respect to the network weights. To ensure the convergence and stability of the BP training algorithm. The initial network weights w_{ij} is set to be small random value and the learning rate μ is selected to be $(0<\mu<1)$. The above formula can be rewritten as following:

$$w_{ij}(t+1) = w_{ij}(t) + \mu h_{oi}\delta_i \tag{6.6}$$

In this equation h_0 is the output of the hidden node, and δ is an error term for node j, if node j is an output node, then

$$\delta_j = y_j(1-y_j)(d_j - y_j) \tag{6.7}$$

Where d_j is the desired output of node j and y_j is the actual output. If node j is an hidden node then,

$$\delta_j = h_o(1-h_o)\sum_k \delta_k w_{jk} \tag{6.8}$$

Convergence is sometimes faster if a momentum term is added and weights changes are smoothed by

$$w_{ij}(t+1) = w_{ij}(t) + \mu h_{oi}\delta_i - \alpha(w_{ij}(t) - w_{ij}(t-1)) \tag{6.9}$$

Were $0 < \alpha < 1$

6.2. An independent mode of MLP

An independent MLP is applied here for modelling a non-linear dynamic system as shown:

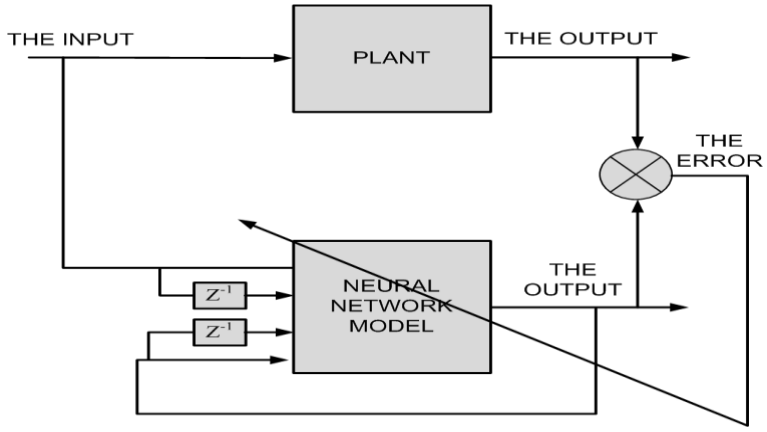


Figure 6.3 Independent mode

The nonlinear dynamic plant to be modelled is presented by the non-linear autoregressive with exogenous inputs (NARX) model as shown in equation (6.10) below:

$$y(t) = f\left[y(t-1), \cdots, y(t-n_y), u(t-1-d), \cdots, u(t-n_u-d)\right] + e(t) \qquad (6.10)$$

Where $u \in \Re^m$ and $y \in \Re^p$ are plant input and output respectively. $e \in \Re^p$ is random noise, m and p are the number of plant inputs and outputs respectively, n_y and n_u are the maximum lags in the model output and input, respectively, d is the time delay in inputs, and $f(*)$ is a vector valued non-linear function.

The independent mode of the network model can be represented by equation (6.11), which is referred to an **independent mode as the** prediction uses the past process outputs in the network input and therefore, the model cannot run independent of the process.

$$\hat{y}(t) = \hat{f}\left[\hat{y}(t-1), \cdots, \hat{y}(t-n_y), u(t-1-d), \cdots, u(t-n_u-d)\right] \qquad (6.11)$$

6.3. Fault Detection

6.3.1. Data Acquisition for open-loop reactor model

Training data is obtained here in the same way that described in chapter (4) by designing a set of random amplitude signals (RAS) for the five inputs to the open-loop reactor model: monomer feed rate, fouling factor, ambient temperature, impurity factor, and valve position. The network input vector consists of the past values of the five system inputs and the past values of the three system outputs. While in closed-loop reactor model, when acquiring training data, the excitation signal should be designed such that the training data has the persistently exciting property and should span over the entire network input space in every dimension, which can provide a good network model interpolation property and good generalization. The network input variables here in closed-loop system consists of the input vector and output vector. The input vector was determined to include the five

process inputs: monomer feed rate, fouling factor, ambient temperature, impurity factor and the fifth input is the controller output. A total data set of 2000 samples is collected from the system Simulink model, and 4s are used as the sampling time. The first 1500 samples are used for training the network model, and the remaining 500 samples are used for testing the network model. Before training and testing, the raw data is scaled linearly into the range of [0 1] using the following formulae

$$u=\begin{bmatrix} m_M \\ 1/h_f \\ T_{amb} \\ i \\ c \end{bmatrix}, \quad y=\begin{bmatrix} T_{jin} \\ T_{jout} \\ T \end{bmatrix} \tag{6.12}$$

$$u_{scaled}(k)=\frac{u(k)-u_{\min}}{u_{\max}-u_{\min}} \quad y_{scaled}(k)=\frac{y(k)-y_{\min}}{y_{\max}-y_{\min}} \tag{6.13}$$

In order to get a good training result with minimum modelling error, several numbers of maximum lags in the outputs and inputs, and several numbers of the maximum time delay in the inputs are tried. The maximum lags in the output were selected as 2, the maximum lags in the input is selected as 4, and the maximum time delay in the inputs is selected as 2, as described in equation (6.14). Thus, the MLP model is designed to have 26 inputs and 3 outputs as shown in figure 6.7. Several number of hidden layer nodes are tried. The initial value of weight connecting network input and the input of hidden layer nodes is selected as $v=0.1\times rand(n_h,n)$, where n_h is the number of hidden layers and n number of inputs to the proposed network. The initial value of weights connecting network output and then output of hidden layer nodes is chosen as $p=0.1\times rand(p,n_h)$, where p is the number of network outputs. μ The learning rate is selected here as 0.0001, and the training epochs is chosen as 15000. The back-propagation training algorithm is used here for training the proposed network as described in the previous section.

$$x(t)=[y(t-1)\ y(t-2)\ \ u(t-k-1)\ u(t-k-2)\ u(t-k-3)u(t-k-4)]^T \qquad (6.14)$$

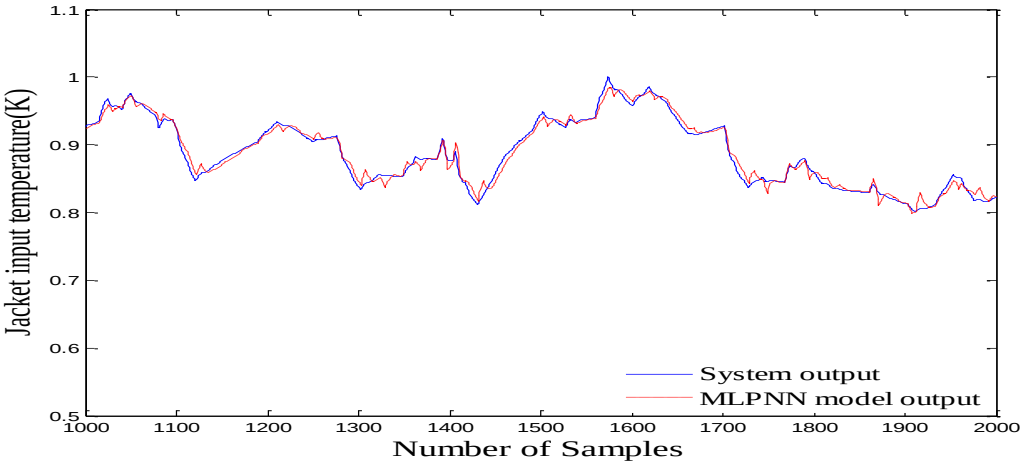


Figure 6.4 Jacket input temperature with MLPNN model for open-loop

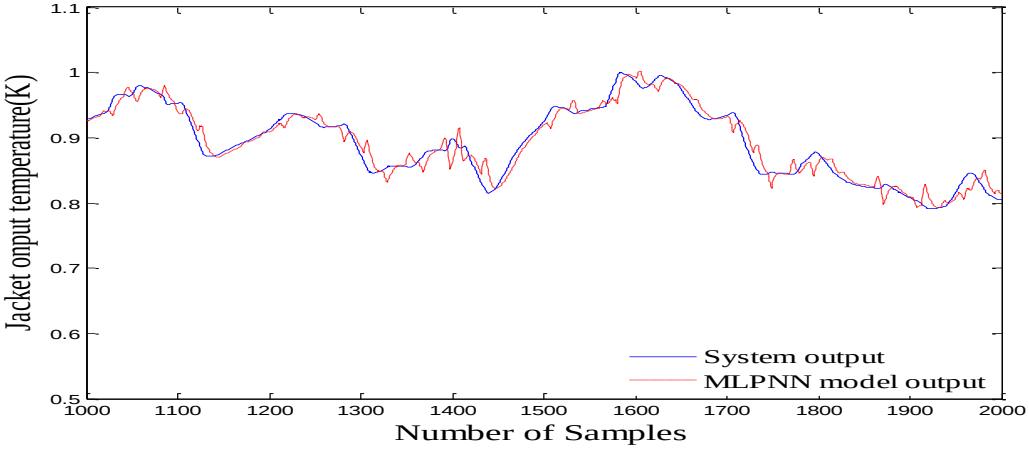


Figure 6.5 Jacket output temperature with MLPNN model for open-loop

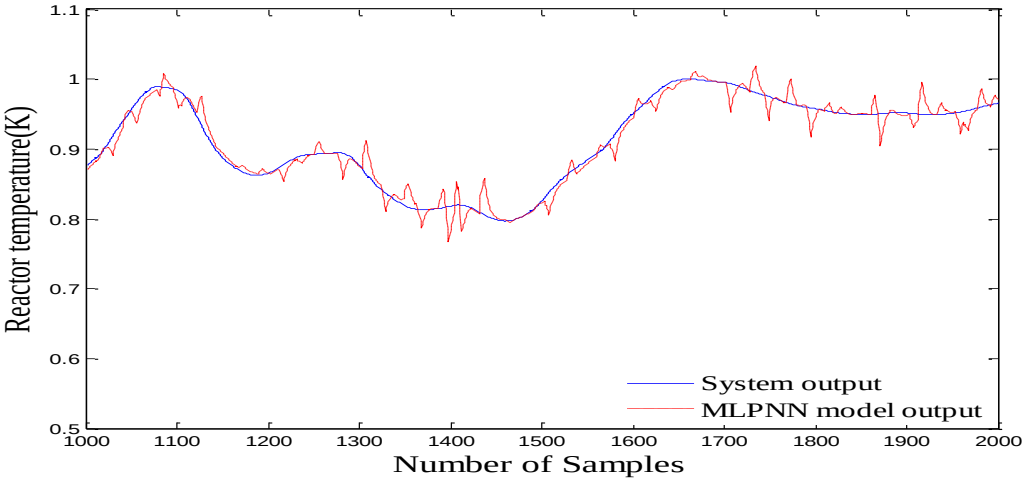


Figure 6.6 Reactor temperature with MLPNN for open-loop

Figure.6.4, 6.5, and 6.6 show the last 500 sample intervals in the training data set and the first 500 sample intervals in the testing data set. It can be clearly seen that the model outputs track the system output with a small modelling error.

6.3.2. Data Acquisition for closed-loop reactor model

Similar to acquiring training data for closed-loop reactor in chapter (5), the excitation signal is designed such that the training data has the persistently exciting property and should span over the entire network input space in every dimension, which can provide a good network model interpolation property and good generalization. A set of modified random amplitude signals (RAS) were designed for monomer feed rate, fouling factor, ambient temperature, impurity factor, and valve position setpoint. Then the network input variables is determined. The network input variables consist of the input vector and output vector. The input vector was determined to include the five process inputs: monomer feed rate, fouling factor, ambient temperature, impurity factor and the fifth input is the controller output. Here in this study the network input vector is designed such that, the maximum lags in the output were selected as 2, the maximum lags in the input is selected as 2, and the maximum time delay in the inputs is selected as 2, as described in equation. (6.15). Thus, the MLP model is designed to have 16 inputs and 3 outputs as shown in figure 6.7. Several number of hidden layer nodes are tried. The initial value of weight connecting network input and the input of hidden layer nodes is selected as $v = 0.3 \times rand(n_h, n)$ .The initial value of weights connecting network output and then output of hidden layer nodes is chosen as $p = 0.3 \times rand(p, n_h)$. μ The learning rate is selected here as 0.01 , and the training epochs is chosen as 10000 . The back-propagation training algorithm is used here for training the proposed network as described in the previous section.

$$x(t) = [y(t-1)\, y(t-2)\, u(t-k-1)\, u(t-k-2)]^T \tag{6.15}$$

6.4. Simulating Faults

In this study, after training the independent RBF network model with healthy data, the model will be used to detect faults that occurred in the system, i.e. generate residual when the system is subjected to any fault. The faulty data is obtained by simulating different faults in the proposed reactor. The classification and structure of faults are done in the same patter in previous chapter 4 and chapter 5.

6.5. Residual generation for open-loop reactor model

An independent model is implemented in parallel with the system to generate the residuals for detecting the sensor and actuator faults in the reactor. After training the network model with healthy random data, as described in the previous section, all four faults were simulated to the reactor model. Then, with another set of 2000 samples, faulty square data is collected. These faulty data are collected by designing a set of square waves for all inputs. The back-propagation training algorithm is used here for testing the proposed network as described in the previous section. Figure 6.7 demonstrates the fault detection approach. Figure 6.8, 6.9, and 6.10 demonstrate the residuals after using a low pass filter. The first model prediction error of jacket input temperature is shown in figure 6.8 and that for jacket output temperature and reactor temperature are shown in figure 6.9 and figure 6.1, respectively. It can be observed that the independent network model output is not influenced by any type of fault, and the thresholds are chosen here as 0.1 for all cases. The residuals of testing the neural model are slightly bigger than the residuals of training the neural model. The MAE for the jacket input temperature, jacket output temperature and reactor temperature are 0.0061, 0.0044and 0.0068, respectively.

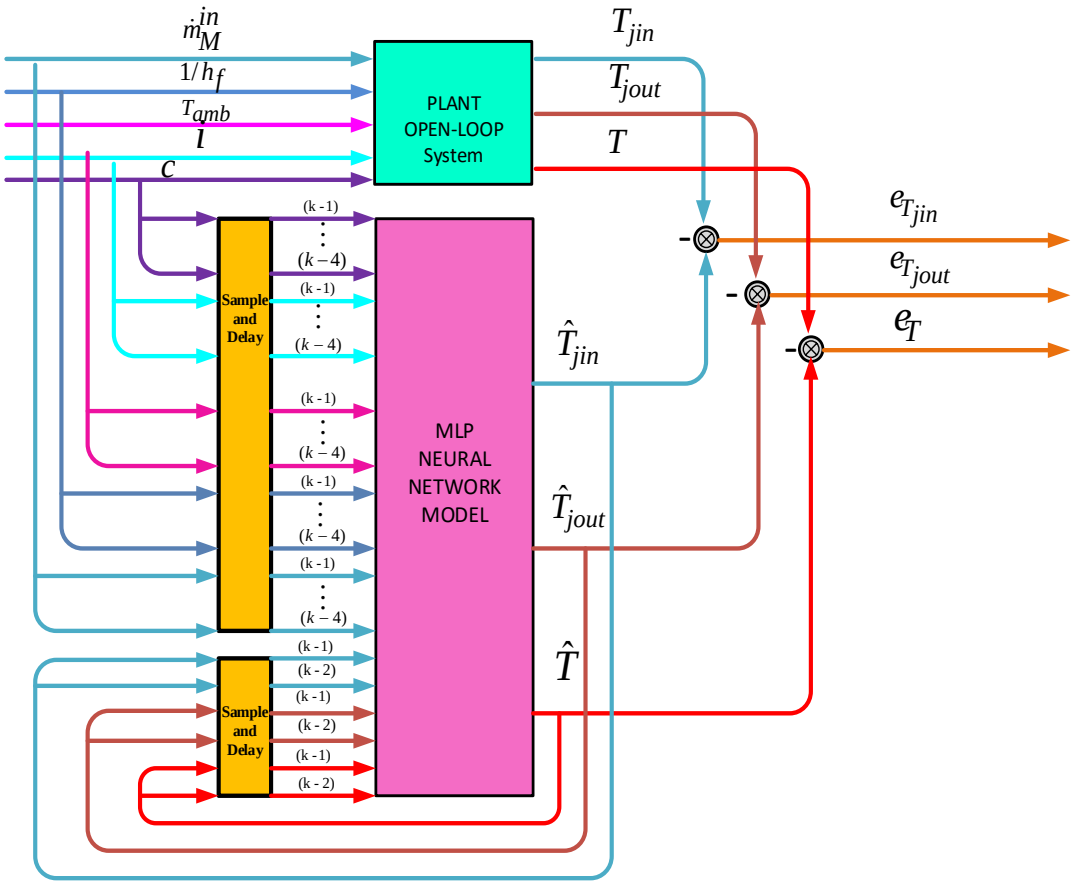


Figure 6.7 Structure of FD using an independent MLPNN for open-loop system

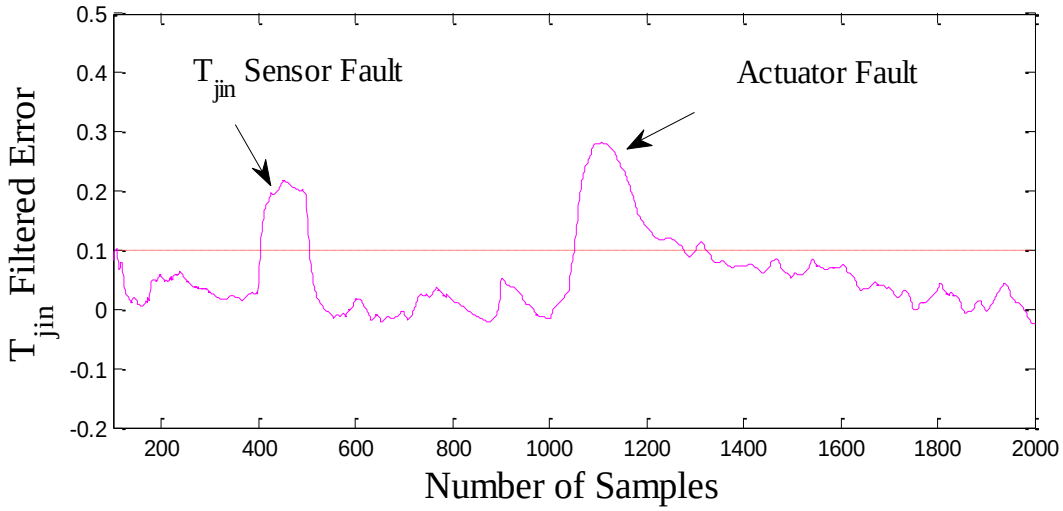


Figure 6.8 Filtered residual model prediction error of Tjin for open-loop system

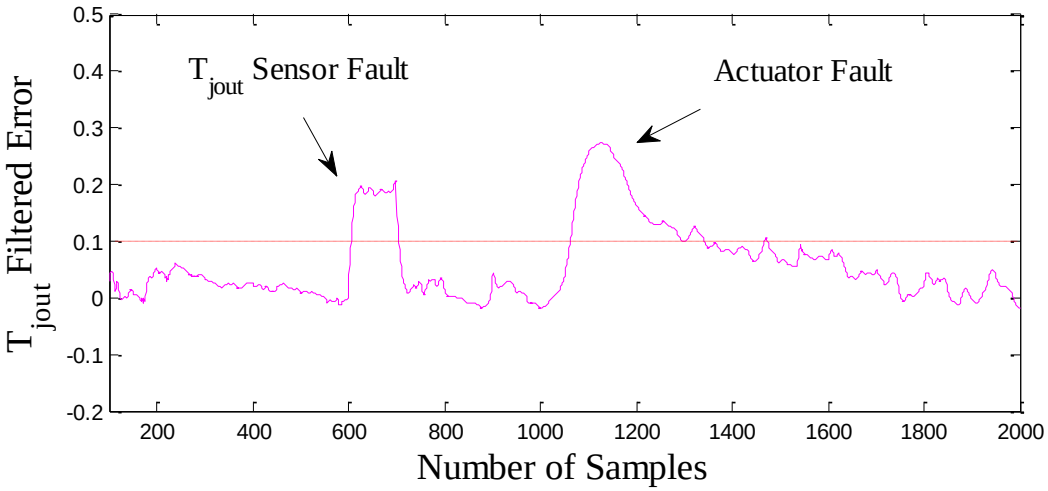


Figure 6.9 Filtered residual model prediction error of Tjout for open-loop system

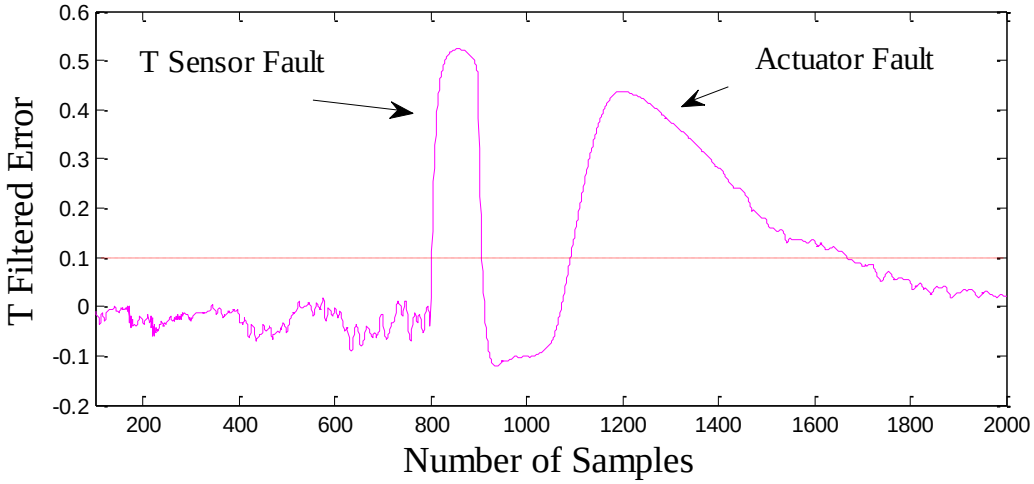


Figure 6.10 Filtered residual model prediction error of T for open-loop system

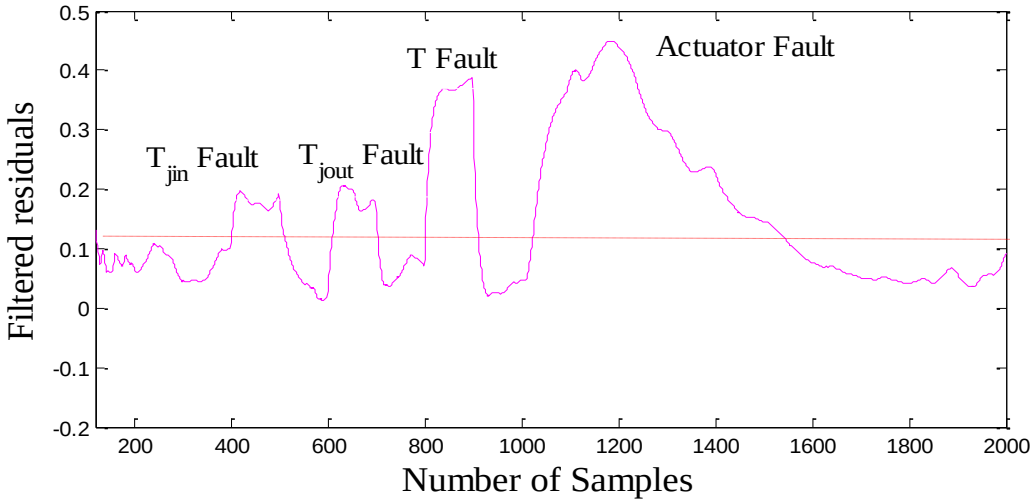


Figure 6.11 Filtered sum-squared model prediction errors

6.6. Fault Isolation

In this section, Radial basis neural network is used for fault classification. The network training for classification is similar from that for modelling in chapter (4). The centres of the classifier were chosen using the K-means clustering algorithm, so that the sum squared distance of each input data from the centre is minimized. The widths were chosen using p-nearest-neighbours. In the updating of the classifier weights, the recursive least squares (RLS) algorithm was used. The parameters of the RLS algorithm are selected as follows μ=0.99999 , $w(0)=10^{-8}*U(n_h,5)$ and $p(0)=10^{8}*I(n_h)$. Where μ is the forgetting factor, I is an identity matrix, U is the element unity matrix, and n_h is the number of hidden layer nodes. Different numbers of hidden nodes, such as $151, 251$ and 500 were used. Finally 151 hidden layer nodes are selected. Figure 6.12 shows the block diagram for fault isolation using RBFNN classifier. The samples arranged for fault occurrence and the isolation methodology are described in chapter (4) and (5). The outputs of the RBFNN classifier are displayed in Figures 6.13-6.16. the thresholds are chosen as 0.4 for all cases. It can be observed that all faults are isolated.

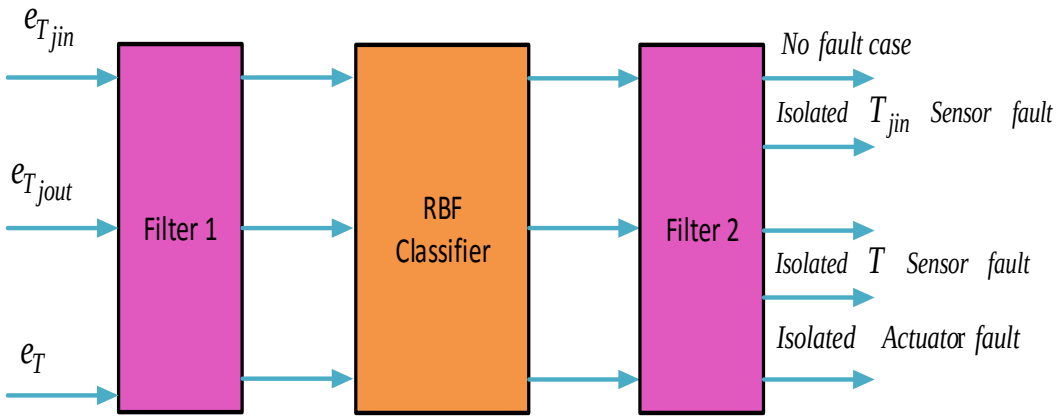


Figure 6.12 Block diagram of fault isolation

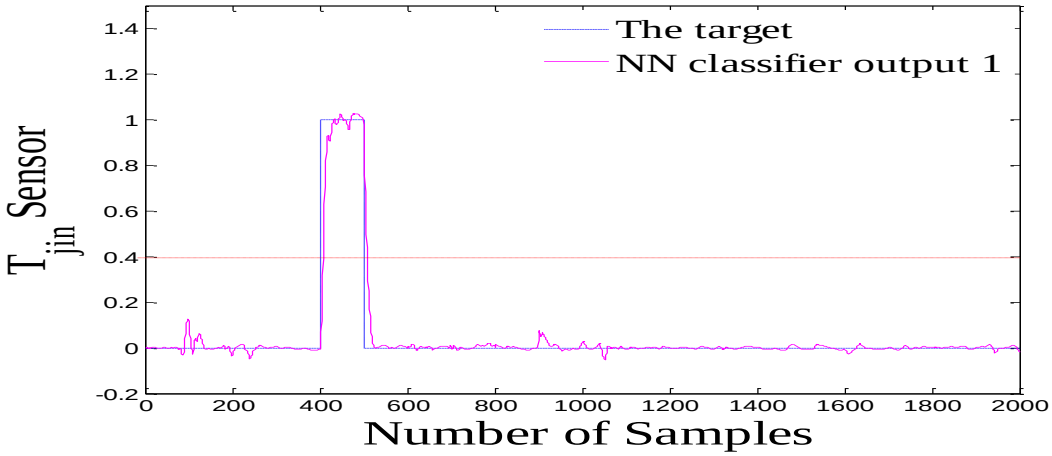


Figure 6.13 Classifier output 1

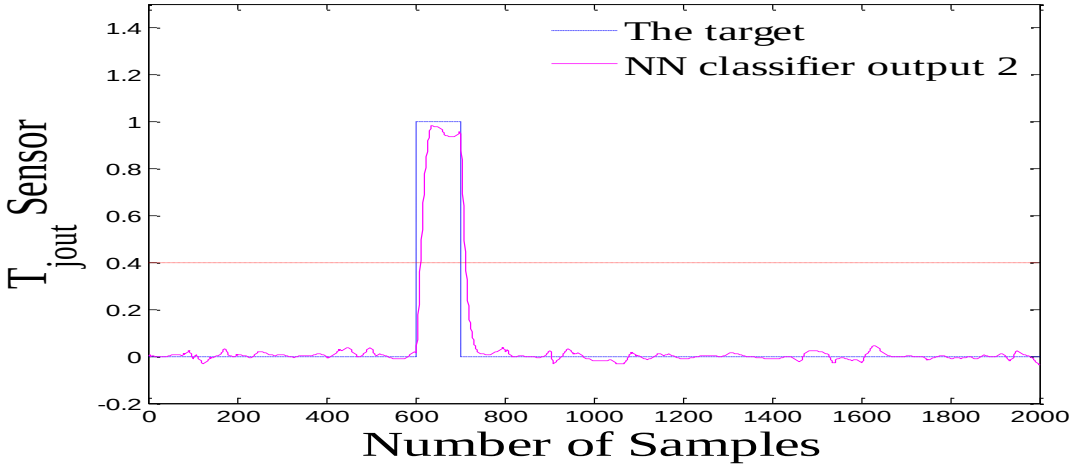


Figure 6.14 Classifier output 2

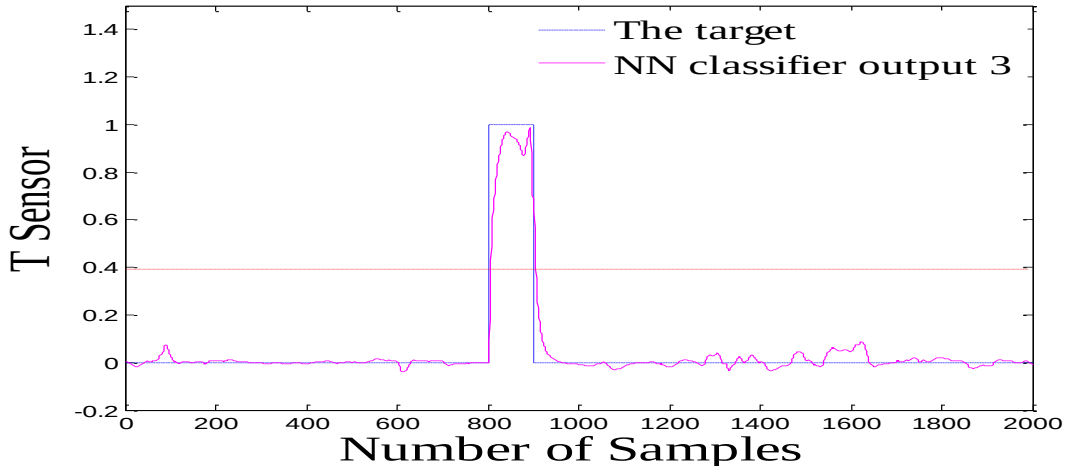


Figure 6.15 Classifier output 3

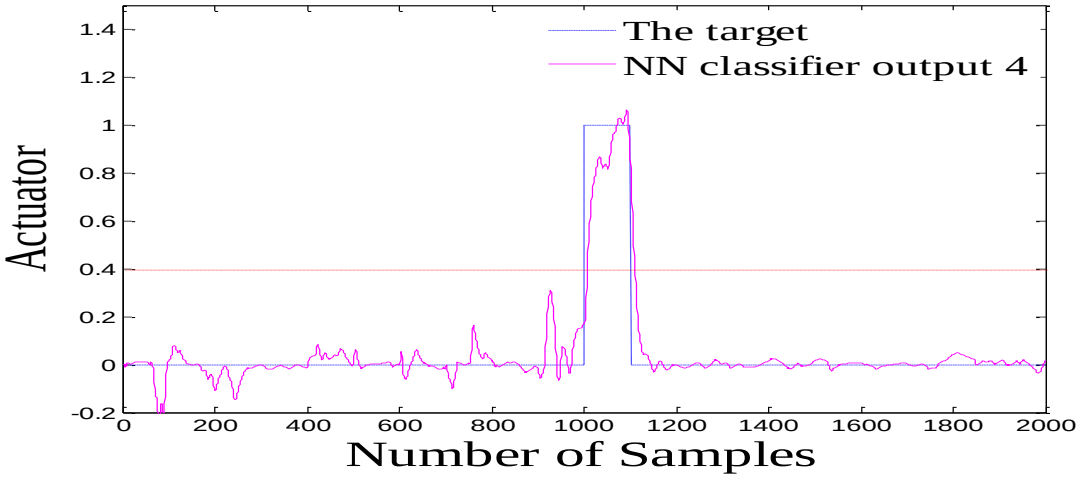


Figure 6.16 Classifier output 4

6.7. Residual generation for closed-loop reactor model

In order to produce polymer of desired quality a very tight temperature control is essential for the reactor. The controller should be able to keep the reactor temperature T within an interval of $\pm 0.6K$ around the desired set-point under all operating conditions and disturbances. Commonly used for a chemical reactor is a PI cascade control structure. The block diagram of the cascade PI control and the parameters of the conventional cascade PI controllers are shown in chapter (5). Figure 6.17 illustrates the fault detection approach, an independent MLP neural network is implemented to generate the residuals for the detection task. After training the network model with healthy random data, as described in the previous section, all four faults are simulated to the reactor model. Then, the fault detection is conducted with the network model using another set of 2000 samples faulty square data. These faulty data were collected when the system is given a set of designed square waves for monomer feed rate, fouling factor, ambient temperature and impurity factor. To simulate the realistic situation in the practical applications, a smaller amplitude signal is added to the fifth input of the system which is the controller output to excite the dynamics in different frequencies. Again the independent is tested using back-propagation training algorithm as described in previous training section. Figure 6.18,

6.19, and 6.20 demonstrate the residuals after using a low pass filter. The first model prediction error of jacket input temperature is shown in figure 6.18 and that for jacket output temperature and reactor temperature are shown in figure 6.19 and figure 6.20, respectively. It can be observed that the independent network model output is not influenced by any type of fault, and all fault signal are over the thresholds setting. Here in this section the thresholds are chosen as 0.1. The MAE for the jacket input temperature, jacket output temperature and reactor temperature are 0.0033, 0.0031and 0.0053, respectively.

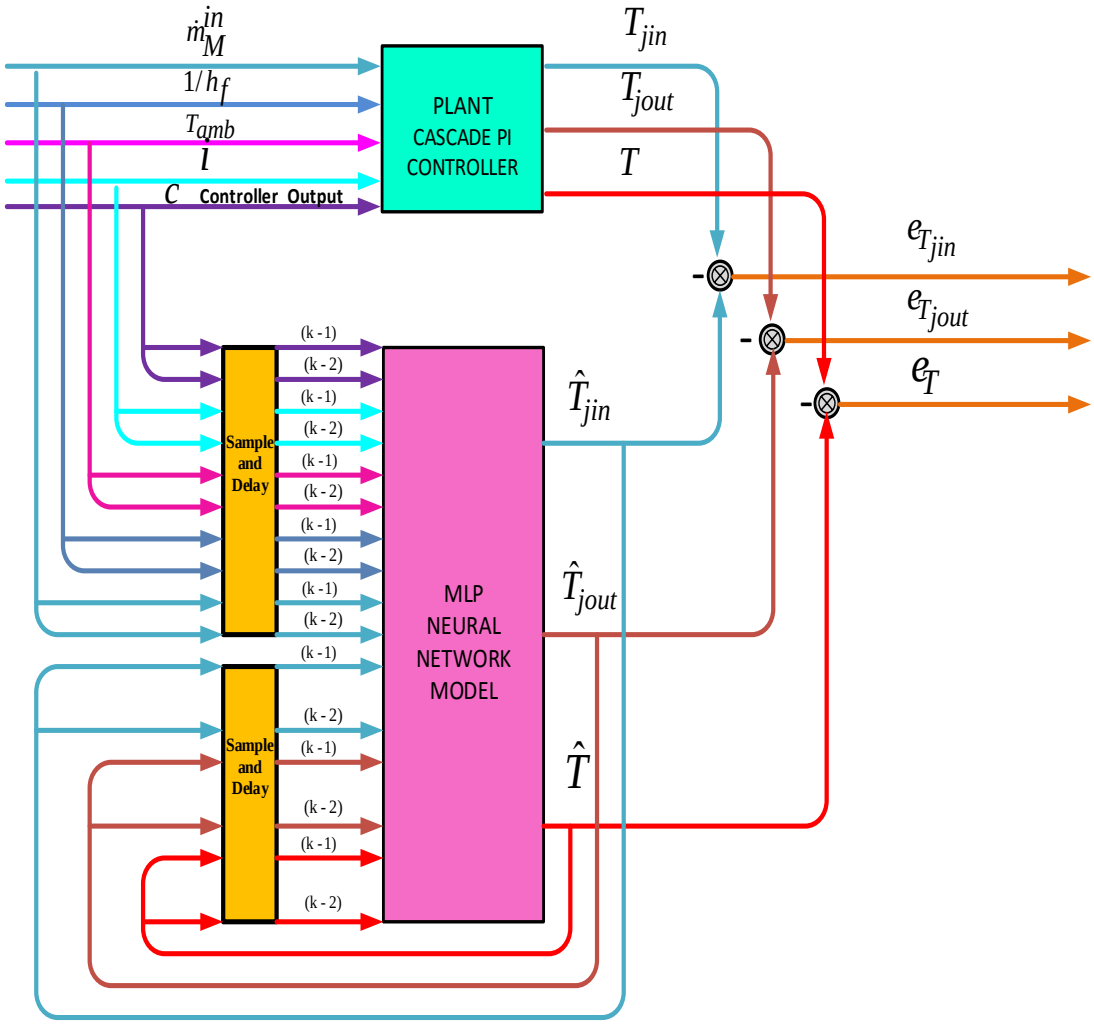


Figure 6.17 Structure of FD using an independent MLPNN for closed-loop system

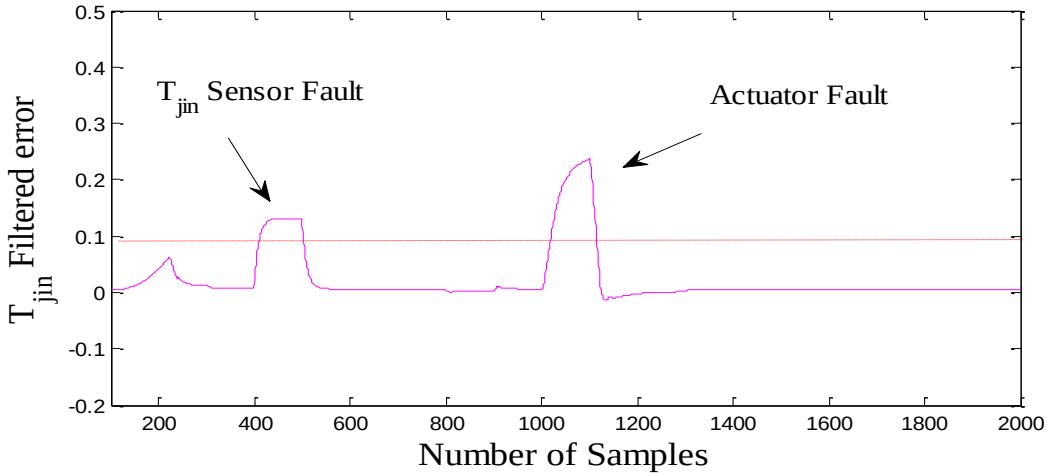


Figure 6.18 Filtered residual model prediction error of Tjin for closed-loop system

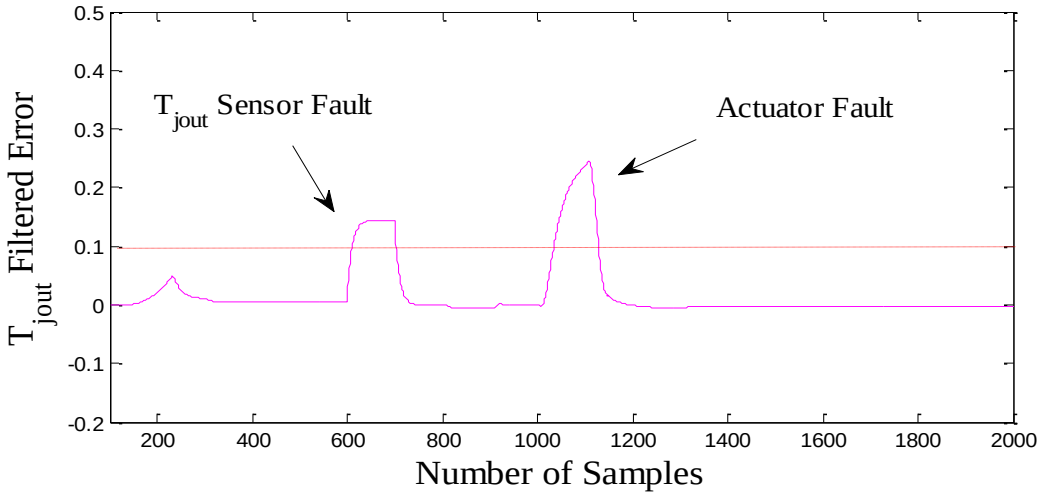


Figure 6.19 Filtered residual model prediction error of Tjout for closed-loop system

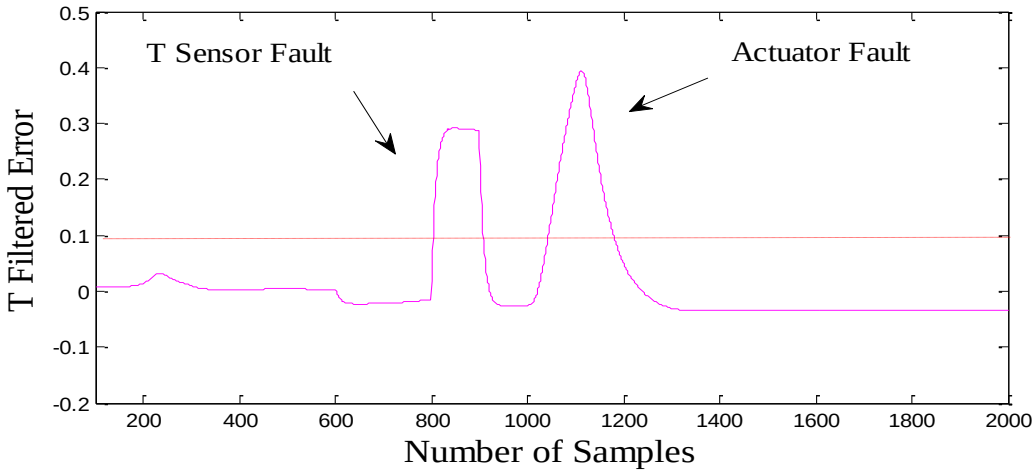


Figure 6.20 Filtered residual model prediction error of T for closed-loop system

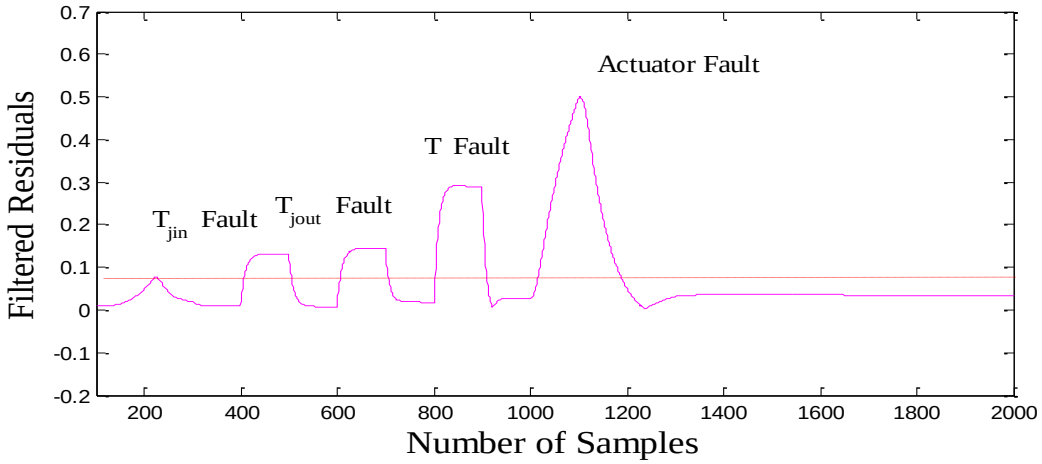


Figure 6.21 Filtered residual sum-squared model prediction errors

6.8. Fault isolation

Here again similar to the previous section a RBFF neural network is implemented to isolate faults and perform as a classifier . for more details see chaptre (4) and (5). The classifier outpurs are displayed in Figures 6.22-6.25. it can be observed that all faults have been isolated. The threshoulds here are chosen as 0.4 for all cases.

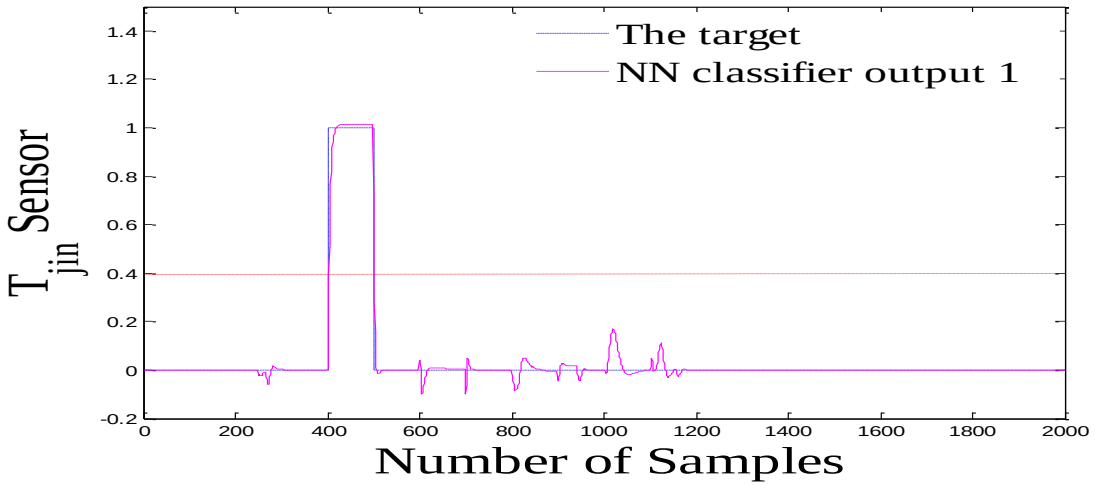


Figure 6.22 Classifier output 1

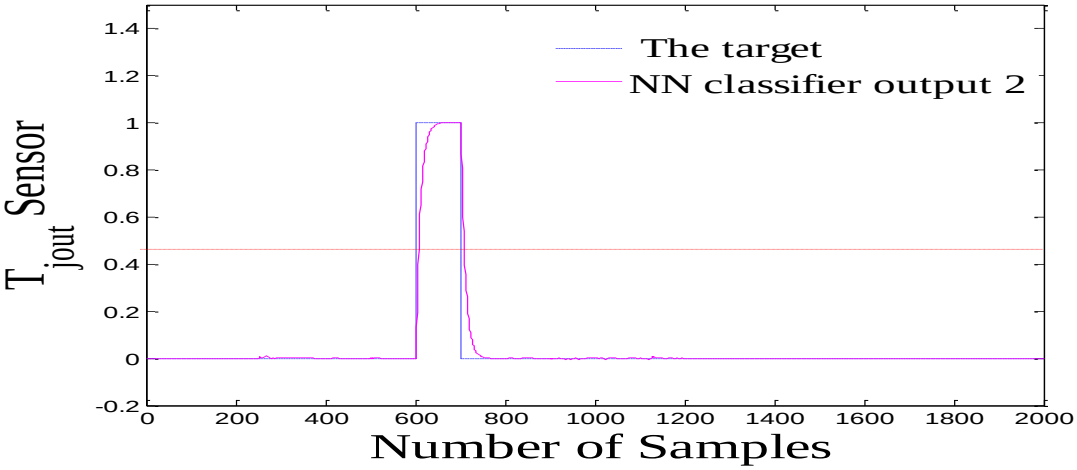


Figure 6.23 Classifier output 2

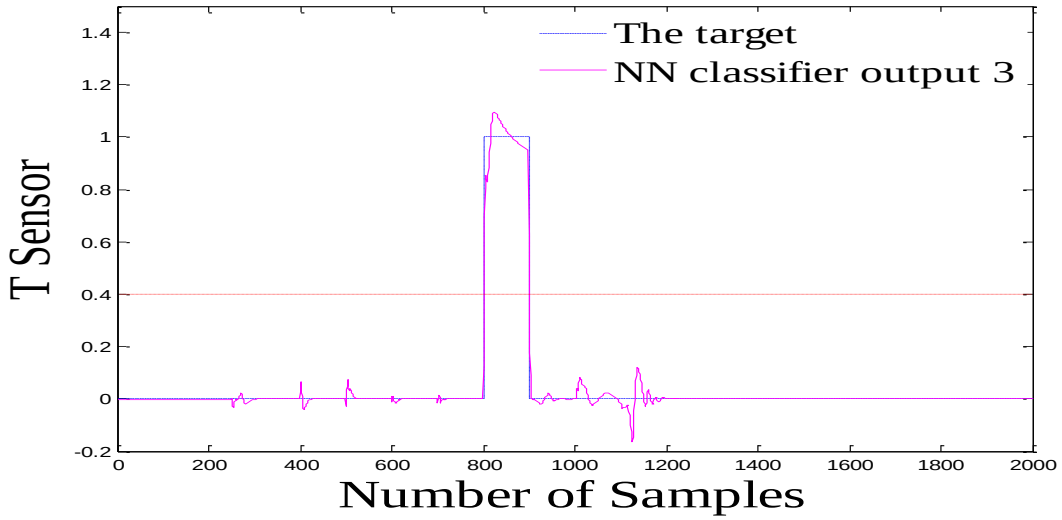


Figure 6.24 Classifier output 3

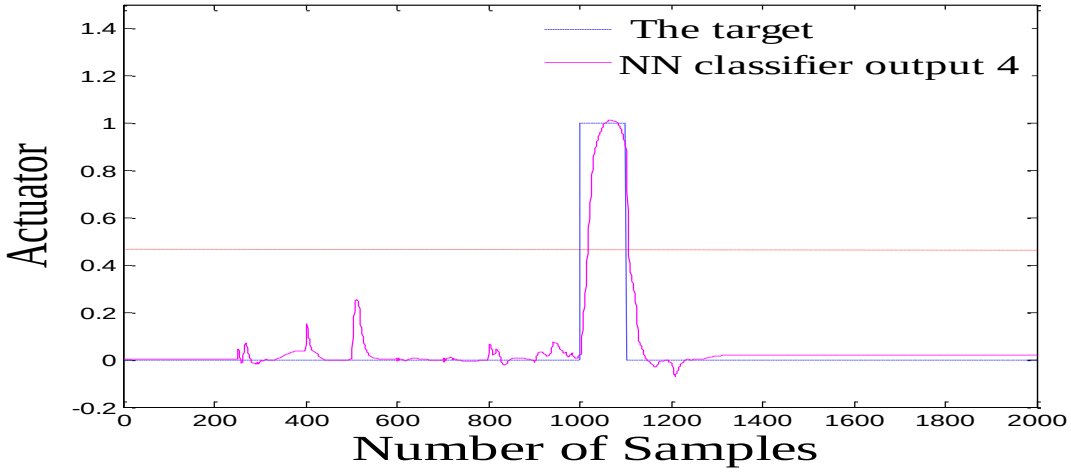


Figure 6.25 Classifier output 4

6.9. A comparison and discussion

In this work an independent MLP neural network is implemented to generate residuals for detection task. In the independent model, the past model output is fed back as part of the network input. Therefore, the model can operate independently from the process. Obviously, the dependent model can predict the process output for one step ahead only, while the independent model can predict the process output for multi-step ahead and can also operate as a simulation model independent of the process. The simulation results show that the independent network model output is not influenced by any type of fault. Therefore, it can be clearly noticed that all faults have been clearly detected. Moreover, no false alarms were thereby produced, so this verifies that the proposed scheme has shown excellent diagnostic performance. Since the independent model does not use past faulty measurements as inputs. It is observed that the neural model outputs did not track the faulty system outputs. However, one of most important criteria in fault diagnosis is the length of training time. The RBF network is implemented in chapter (4) and (5) for fault diagnosis, and is used because of its advantages over the multi-layer perceptron (MLP) of short training time. The comparison of two neural network architectures (MPL and RBF) has shown that RBF configuration trained by (RLS) algorithm have several advantages. The first one is greater efficiency in finding optimal weights for field strength prediction in complex dynamic systems. The RBF configuration is less complex network that results in faster convergence. The training algorithms (RLs and ROLS) that used for training RBFNN in chapter (4) and(5) have proven to be efficient, which results in significant faster computer time in comparison to backpropagation one.

6.10. Summary

An independent MLP neural network is implemented here to generate residuals for detection task. And another RBF is applied for isolation task performing as a classifier.

The fault diagnosis scheme is developed for a Chylla-Haase reactor under open-loop and closed-loop control system. The simulation results confirmed that the simulated faults have been clearly detected and isolated with zero false alarm rates. So this verifies that the proposed scheme has shown excellent diagnostic performance. . The research indicates the feasibility of the developed scheme applied to industrial systems, especially chemical and biochemical processes, for which the mathematical model is difficult to develop.

Chapter 7

Extended Kalman Filter (EKF) Based FD

A fault diagnosis (FD) scheme is developed in this section for an exothermic semi-batch polymerization reactor. The scheme includes two parts: the first part is to generate residual using an extended Kalman filter (EKF), and the second part is the decision making to report fault using a statistical method. The reactor is a multivariable nonlinear dynamic process and is subjected to several major disturbances. A mathematical model is developed for the reactor with some model parameters identified from the input/output data, and then the developed continuous model is discretized into a discrete model. Three sensor faults and one actuator fault are simulated on the reactor and are detected using the developed method. Moreover, several practical disturbances and system uncertainties, such as significant changes in monomer feed rate, fouling factor, impurity factor and ambient temperature, as well as measurement noise are also simulated. The FD simulation results are presented to demonstrate the effectiveness of the proposed method.

7.1. Extended Kalman Filter (EKF)

In this section the filtering problem in nonlinear dynamic systems is addressed. The EKF algorithm in discrete-time form is reviewed. Consider the following nonlinear system, described by difference equation and the observation model with additive noise:

$$X_{k-1} = f(X_{k-1}, u_{k-1}) + W_{k-1} \tag{7.1}$$

$$Y_{k-1} = h(X_k) + V_k \tag{7.2}$$

Where:

X_k State vector

Y_k Observation vector

W_k Process noise vector

V_k Measurement noise vector

$f(*)$ Process nonlinear vector function

$h(*)$ Observation nonlinear vector function

The EKF uses a 2 step prediction-correction algorithm. The first step involves projecting both the most recent state estimate and an estimate of the error covariance (from the previous time period) forwards in time to compute a predicted (or a-priori) estimate of the states at the current time. The second step involves correcting the predicted state estimate calculated in the first step by incorporating the most recent process measurement to generate an updated (or a-posteriori) state estimate. However, due to the non-linear nature of the process being estimated the covariance prediction and update equations cannot use f and h directly. Rather the Jacobean of f and h will be used.

Predict and update equations

Predict

Predict state

$$\hat{X}_{k|k-1} = f(\hat{X}_{k-1|k-1}, u_{k-1}) \tag{7.3}$$

Predict estimate covariance

$$P_{k|k-1} = F_{k-1}\ P_{k-1|k-1}\ F_{k-1}^{T} + Q_k \tag{7.4}$$

Update

Innovation or measurement residual

$$\gamma_k = Y_k - h(\hat{X}_{k|k-1}) \tag{7.5}$$

Innovation or residual covariance

$$S_k = H_k\ P_{k|k-1}\ H_k^{T} + R_k \tag{7.6}$$

Optimal Kalman gain

$$K_k = P_{k|k-1}\ H_k^{T}\ S_k^{-1} \tag{7.7}$$

Update state estimate

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k\ \gamma_k \tag{7.8}$$

Update estimate covariance

$$P_{k|k} = (I - K_k\ H_k)\ P_{k|k-1} \tag{7.9}$$

Where w_k and v_k are the process and measurement noise which assumed to be zero mean Gaussian noise with covariance Q_k and R_k, and they are given as:

Diagonal process noise covariance matrix

$$Q_k = E[w_k\ \ w_k^{T}] \tag{7.10}$$

Diagonal measurement noise covariance matrix

$$R_k = E[v_k\ \ v_k^{T}] \tag{7.11}$$

Where F and H are the Jacobean matrixes that allowing the linearization of the reactor model and they are given by the following equations:

F is the Jacobean matrix of partial derivatives of $f(*)$ with respect to x:

$$F_k = \left[\partial f(x,k) / \partial x \right] \tag{7.12}$$

H is the Jacobean matrix of partial derivatives of $h(*)$ with respect to x:

$$H_k = \left[\partial h(x,k) / \partial x \right] \tag{7.13}$$

Where x_k is defined as $[T, T_{jin}, T_{jout}]$.

7.2. Discretization of reactor model

The Cyhlla-Haase reactor is described by a set of continuous time differential equations. However, the extended Kalman filter requires a set of discrete equations. Hence for use within an extended Kalman filter the reactor model equations must be discretized. The simple and efficient approach for discretization is to use a backward Euler method.

Euler method is the simplest method for solving differential equations numerically; it's also called backward Euler method or explicit method. It was developed by truncate the Tylor series at first term and neglecting the high order terms.

$$y' = f(x, y) \tag{7.14}$$

$$y_k = y_{k-1} + h\, f(x_{k-1}, y_{k-1}) \tag{7.15}$$

Where h is a step size.

7.3. Online states and parameters estimations

An online estimation of the states and parameters with an EKF requires a simplified reactor model, which is still accurate enough to obtain reliable estimation. In order to obtain a more accurate model, all the empirical relations for the polymerization rate,

jacket heat transfer area, and overall heat transfer coefficient are substituted. Then the simplified reactor model equations are discretized using first order backward Euler method described in (7.14) -(7.15) as following:

Material balance for monomer mass

$$\frac{dm_M}{dt} = \dot{m}_M^{in}(t) + \frac{Q_{rea}}{\Delta H}$$

Where, $Q_{rea} = -\Delta H * R_p$, $R_p = i * k * m_M$, $k = k_0 \exp(-E/RT)(k_1\mu)^{k_2}$,

$f = {m_P}/{(m_P + m_M + m_C)}$ $\mu = c_0 \exp(c_1 f) * 10^{c_2(a_0/T - c_3)}$,

So, then

$$m_{M(k)} = m_{M(k-1)} + T_s \times (\dot{m}_M^{in}) - T_s \times \left([i * m_{M(k-1)}] \times [k_0 \exp(-E/RT_{(k-1)}) * \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2}] \right) \quad (7.17)$$

Material balance for polymer mass

$$\frac{dm_P}{dt} = -\frac{Q_{rea}}{\Delta H}$$

$$m_{P(k)} = m_{P(k-1)} + T_s \times [i * m_{M(k-1)}] \times T_s \times \left([k_0 \exp(-E/RT_{(k-1)}) * \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2}] \right) \quad (7.18)$$

Energy balance for reactor temperature

$$\frac{dT}{dt} = \frac{1}{\sum_i m_i C_{P,i}} [\dot{m}_M^{in}(t) C_{P,M}(T_{amb} - T) - UA(T - T_j) - (UA)_{loss}(T - T_{amb}) + Q_{rea}]$$

Where,

$$T_{wall} = {(T+T_j)}/{2} \quad A = (\frac{m_M}{\rho_M} + \frac{m_P}{\rho_P} + \frac{m_W}{\rho_W}) \frac{P}{B_1} + B_2$$

So, then

$$T_k = T_{(k-1)} + T_s \times \left(\frac{\left(-\dot{m}_M^{in} * C_{p,M} * T_{(k-1)}\right) + \left(-UA_{loss} * T_{(k-1)}\right)}{(m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W})} \right) +$$

$$T_s \times \left(\frac{((\frac{m_{M(k-1)}}{\rho_M} + \frac{m_{P(k-1)}}{\rho_P} + \frac{m_W}{\rho_W}) \frac{P}{B_1} + B_2) * (-T_{(k-1)})}{\frac{1}{d_0 \exp(d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)})} + \frac{1}{h_f}} \right) +$$

$$T_s \times \left(\frac{1}{(m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W})} \right) \times \left[\left(\dot{m}_M^{in} * C_{p,M} * T_{amb}\right) + \left(UA_{loss} * T_{amb}\right) \right] +$$

$$T_s \times \left(\frac{1}{(m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W})} \right) \times$$

$$\left[\left(\frac{((\frac{m_{M(k-1)}}{\rho_M} + \frac{m_{P(k-1)}}{\rho_P} + \frac{m_W}{\rho_W}) \frac{P}{B_1} + B_2) * (\frac{T_{jin(k-1)}}{2})}{\frac{1}{d_0 \exp(d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)})} + \frac{1}{h_f}} \right) \right] +$$

$$T_s \times \left(\frac{1}{(m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W})} \right) \times$$

$$\left[\left(\frac{((\frac{m_{M(k-1)}}{\rho_M} + \frac{m_{P(k-1)}}{\rho_P} + \frac{m_W}{\rho_W}) \frac{P}{B_1} + B_2) * (\frac{T_{jout(k-1)}}{2})}{\frac{1}{d_0 \exp(d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)})} + \frac{1}{h_f}} \right) \right] +$$

$$T_s \times \left(\frac{1}{(m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W})} \right) \times \left[(-\Delta H * i * m_{M(k-1)}) * \right.$$

$$\left. (k_0 \exp(-E/RT_{(k-1)}) * \left(k_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2}) \right]$$

(7.19)

Energy balance for jacket output temperature

$$
\begin{aligned}
T_{jout(k)} = T_{jout(k-1)} + \\
T_s \times \left[\left(\frac{\left(\left(\frac{m_{M(k-1)}}{\rho_M} + \frac{m_{P(k-1)}}{\rho_P} + \frac{m_W}{\rho_W} \right) \frac{P}{B_1} + B_2 \right) * (T_{(k-1)}) * \left(\frac{1}{m_C \, C_{p,C}} \right)}{\frac{1}{d_0 \exp\left(d_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} \right)} + \frac{1}{h_f}} \right) \right] + \\
T_s \times \left[\left(\frac{\left(\left(\frac{m_{M(k-1)}}{\rho_M} + \frac{m_{P(k-1)}}{\rho_P} + \frac{m_W}{\rho_W} \right) \frac{P}{B_1} + B_2 \right) * \left(\frac{-T_{jin(k-1)}}{2} \right) * \left(\frac{1}{m_C \, C_{p,C}} \right)}{\frac{1}{d_0 \exp\left(d_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_{C(k-1)}} \right) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} \right)} + \frac{1}{h_f}} \right) \right] + \\
T_s \times \left[\left(\frac{\left(\left(\frac{m_{M(k-1)}}{\rho_M} + \frac{m_{P(k-1)}}{\rho_P} + \frac{m_W}{\rho_W} \right) \frac{P}{B_1} + B_2 \right) * \left(\frac{-T_{jout(k-1)}}{2} \right) * \left(\frac{1}{m_C \, C_{p,C}} \right)}{\frac{1}{d_0 \exp\left(d_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_{C(k-1)}} \right) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} \right)} + \frac{1}{h_f}} \right) \right] + \\
T_s \times \left[\left(\frac{1}{m_C \, C_{p,C}} \right) * \left(\dot{m}_C C_{p,C} \, T_{jin} (t - \theta_1) \right) \right] - T_s \times \left[\left(\frac{1}{m_C \, C_{p,C}} \right) * \left(T_{jout(k-1)} \right) \right]
\end{aligned}
$$

(7.19)

Energy balance for jacket input temperature

$$
\frac{dT_{jout}(t-\theta_2)}{dt} = \frac{T_{jout} - T_{jout}(t-\theta_2)}{\theta_2}
$$

$$
T_{jin(k)} = T_{jin(k-1)} + h \times \left(\frac{T_{jout(k-1)} - T_{jout}(t-\theta_2)}{\theta_2} + \frac{T_{jout}(t-\theta_2) - T_{jin(k-1)}}{\tau_P} + \frac{K_P(c)}{\tau_P} \right)
$$

(7.20)

The overall estimation vector of EKF is as following:

$$\hat{x}=[\hat{m}_M \;\; \hat{m}_P \;\; \hat{T} \;\; \hat{T}_{jin} \;\; \hat{T}_{jout}]^T \tag{7.21}$$

With the initial conditions:

$$\hat{x}(0) = \begin{bmatrix} m_{M(0)} & m_{P(0)} & T_{amb} & T_{amb} & T_{amb} \end{bmatrix}^T \tag{7.22}$$

Where F and H are the Jacobian matrixes that allowing the linearization of reactor model, and the linearized system matrixes can be written as following:

Where F is the Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to x:

$$F_k = \frac{\partial f(x(k))}{\partial x(x(k))} = \begin{bmatrix} F_{1,1}(k) & F_{1,2}(k) & F_{1,3}(k) & F_{1,4}(k) & F_{1,5}(k) \\ F_{2,1}(k) & F_{2,2}(k) & F_{2,3}(k) & F_{2,4}(k) & F_{2,5}(k) \\ F_{3,1}(k) & F_{3,2}(k) & F_{3,3}(k) & F_{3,4}(k) & F_{3,5}(k) \\ F_{4,1}(k) & F_{4,2}(k) & F_{4,3}(k) & F_{4,4}(k) & F_{4,5}(k) \\ F_{5,1}(k) & F_{5,2}(k) & F_{5,3}(k) & F_{5,4}(k) & F_{5,5}(k) \end{bmatrix} \tag{7.23}$$

$F_{1,1}$ is the partial derivative of the monomer mass equation with respect to m_M:

$$F_{1,1} = 1 + T_s \times (\dot{m}_M^{in}) - T_s \times \left(\begin{array}{l} \left(i * m_{M(k-1)} * k_0 \exp(-E / RT_{(k-1)}) \right) * \\ \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2} \end{array} \right)$$

$$= 1 + T_s \times (\dot{m}_M^{in})$$

$$- T_s \times \left(\begin{array}{l} \left(i * k_0 \exp(-E / RT_{(k-1)})\right) * \\ \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2} + \\ \left(i * m_{M(k-1)} * k_0 \exp(-E / RT_{(k-1)})\right) * \\ k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2 - 1} * \\ \left(\begin{array}{l} k_1 c_0 10^{c_2 (a_0 / T_{(k-1)} - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} * \\ \exp\left(\frac{c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)}\right) \end{array} \right) \end{array} \right) \tag{7.24}$$

$F_{1,2}$ is the partial derivative of the monomer mass equation with respect to m_P:

$$F_{1,2} =$$

$$- T_s \times \left(\begin{array}{l} \left(i * m_{M(k-1)} * k_0 \exp(-E / RT_{(k-1)})\right) * \\ k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2 - 1} * \\ \left(\begin{array}{l} k_1 c_0 10^{c_2 (a_0 / T_{(k-1)} - c_3)} * \frac{c_1 * (-c_1 m_{P(k-1)}) * (m_{M(k-1)} + m_{P(k-1)} + m_C)}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} * \\ \exp\left(\frac{c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)}\right) \end{array} \right) \end{array} \right) \tag{7.25}$$

$F_{1,3}$ is the partial derivative of the monomer mass equation with respect to T:

$$F_{1,3} =$$

$$-T_s \times \left(\begin{array}{l} \left(i * m_{M(k-1)} * k_0 \exp(-E/RT_{(k-1)}) * (-E/RT_{(k-1)})^2 \right) * \\ \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2} + \\ \left(i * m_{M(k-1)} * k_0 \exp(-E/RT_{(k-1)}) \right) * \\ k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2 - 1} * \\ \left(\begin{array}{l} k_1 c_0 10^{c_2(a_0/T_{(k-1)} - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)} * \\ \frac{-c_2 a_0}{(T_{(k-1)} - c_3)^2} * 10^{c_2(a_0/T_{(k-1)} - c_3)} * Ln(10) \end{array} \right) \end{array} \right) \tag{7.26}$$

$F_{1,4}$ is the partial derivative of the monomer mass equation with respect to T_{jout}:

$$F_{1,4} = 0 \tag{7.27}$$

$F_{1,5}$ is the partial derivative of the monomer mass equation with respect to T_{jin}:

$$F_{1,5} = 0 \tag{7.28}$$

Next, the partial derivatives of the material balance polymer mass equation with respect to all states will be derived.

$F_{2,1}$ is the partial derivative of the polymer mass equation with respect to m_M:

$$F_{2,1} =$$

$$-T_s \times \left(\begin{array}{l} \left(i * k_0 \exp(-E / RT_{(k-1)})\right) * \\ \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2(a_0 / T_{(k-1)} - c_3)} \right)^{k_2} + \\ \left(i * m_{M(k-1)} * k_0 \exp(-E / RT_{(k-1)})\right) * \\ k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2(a_0 / T_{(k-1)} - c_3)} \right)^{k_2 - 1} * \\ \left(\begin{array}{l} k_1 c_0 10^{c_2(a_0 / T_{(k-1)} - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} * \\ \exp\left(\frac{c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)}\right) \end{array} \right) \end{array} \right) \tag{7.29}$$

$F_{2,2}$ is the partial derivative of the polymer mass equation with respect to m_P:

$$F_{2,2} = 1 +$$

$$-T_s \times \left(\begin{array}{l} \left(i * m_{M(k-1)} * k_0 \exp(-E / RT_{(k-1)})\right) * \\ k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}\right) * 10^{c_2(a_0 / T_{(k-1)} - c_3)} \right)^{k_2 - 1} * \\ \left(\begin{array}{l} k_1 c_0 10^{c_2(a_0 / T_{(k-1)} - c_3)} * \frac{c_1 * (-c_1 m_{P(k-1)}) * (m_{M(k-1)} + m_{P(k-1)} + m_C)}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} * \\ \exp\left(\frac{c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)}\right) \end{array} \right) \end{array} \right) \tag{7.30}$$

$F_{2,3}$ is the partial derivative of the polymer mass equation with respect to T:

$$F_{2,3} =$$

$$-T_s \times \left(\begin{array}{l} \left(i * m_{M(k-1)} * k_0 \exp(-E/RT_{(k-1)}) * (-E/RT_{(k-1)})^2 \right) * \\ \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2} + \\ \left(i * m_{M(k-1)} * k_0 \exp(-E/RT_{(k-1)}) \right) * \\ k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)^{k_2 - 1} * \\ \left(\begin{array}{l} k_1 c_0 10^{c_2(a_0/T_{(k-1)} - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)} * \\ \frac{-c_2 a_0}{(T_{(k-1)} - c_3)^2} * 10^{c_2(a_0/T_{(k-1)} - c_3)} * Ln(10) \end{array} \right) \end{array} \right) \tag{7.31}$$

$F_{2,4}$ is the partial derivative of the polymer mass equation with respect to T_{jout}:

$$F_{2,4} = 0 \tag{7.32}$$

$F_{2,5}$ is the partial derivative of the polymer mass equation with respect to T_{jin}:

$$F_{2,5} = 0 \tag{7.33}$$

Next, the partial derivatives of the energy balance for reactor temperature equation with respect to all states will be derived.

$F_{3,1}$ is the partial derivative of the reactor temperature equation with respect to m_M:

Let $F_{3,1} = a + b + c$, where

$$a = T_s \times \left[\begin{array}{l} \left(\left(-C_{p,M} * T_{(k-1)} * \dot{m}_M^{in} \right) * \left(\frac{-C_{p,M}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) + \\ \left(\left(C_{p,M} * T_{amb} * \dot{m}_M^{in} \right) * \left(\frac{-C_{p,M}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) + \\ \left(\left(-T_{(k-1)} * UA_{loss} \right) * \left(\frac{-C_{p,M}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) + \\ \left(\left(T_{amb} * UA_{loss} \right) * \left(\frac{-C_{p,M}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) \end{array} \right] \quad (7.34)$$

$$b = T_s \times \left[\begin{array}{l} \left(\left(k_0 \exp(-E / RT_{(k-1)}) \right) * \right. \\ \left. \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2} \right) * \\ \left(\frac{\left((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}) \right) (-\Delta H * i) + \left(\Delta H * m_{M(k-1)} C_{p,M} \right)}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \end{array} \right] + T_s \times \left[\begin{array}{l} \left(\frac{\left(\Delta H * i * m_{M(k-1)} \right)}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))} \right) * \\ \left(k_0 \exp(-E / RT_{(k-1)}) * k_2 \left(k_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2 - 1} \right) * \\ \left(k_1 * c_0 * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} * \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * \left(\frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} \right) \right) \end{array} \right] \quad (7.35)$$

For more simplicity the reactor temperature equation is more simplified as following:

$$h = d_0 \exp(d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)})$$

$$\beta = \left[\left(\frac{((\frac{P*m_{M(k-1)}}{B_1*\rho_M} + \frac{P*m_{P(k-1)}}{B_1*\rho_P} + \frac{P*m_W}{B1*\rho_W}) + B_2)*(-T_{(k-1)} + T_{jin(k-1)}/2 + T_{jout(k-1)}/2)}{((m_{M(k-1)}C_{p,M}) + (m_{P(k-1)}C_{p,P}) + (m_W C_{p,W}))} \right) \right]$$

So then,

$$c = T_S \times \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\frac{\left(\left(-T_{(k-1)} + T_{jin(k-1)}/2 + T_{jout(k-1)}/2 \right) * \left(PB_1\rho_M / (B_1\rho_M)^2 \right) * \left(((m_{M(k-1)}C_{p,M}) + (m_{P(k-1)}C_{p,P}) + (m_W C_{p,W})) \right) \right)}{((m_{M(k-1)}C_{p,M}) + (m_{P(k-1)}C_{p,P}) + (m_W C_{p,W}))^2} \right) \right] +$$

$$T_S \times \left[\frac{(\beta) * \left((h)^{-2} * \left((h) * \left(d_1 c_0 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)} \right) * \exp\left(c_1 * m_P / ((m_{M(k-1)}C_{p,M}) + (m_{P(k-1)}C_{p,P}) + (m_W C_{p,W})) \right) * \left(-c_1 * m_P / ((m_{M(k-1)}C_{p,M}) + (m_{P(k-1)}C_{p,P}) + (m_W C_{p,W}))^2 \right) \right) \right)}{(h^{-1} + h_f^{-1})^2} \right]$$

(7.36)

$F_{3,2}$ is the partial derivative of the reactor temperature equation with respect to m_P:

Let $F_{3,2} = a + b + c$, where

$$a = T_s \times \left[\begin{array}{l} \left(\left(-C_{p,M} * T_{(k-1)} * \dot{m}_M^{in} \right) * \left(\dfrac{-C_{p,P}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) + \\ \left(\left(C_{p,M} * T_{amb} * \dot{m}_M^{in} \right) * \left(\dfrac{-C_{p,P}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) + \\ \left(\left(-T_{(k-1)} * UA_{loss} \right) * \left(\dfrac{-C_{p,P}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) + \\ \left(\left(T_{amb} * UA_{loss} \right) * \left(\dfrac{-C_{p,P}}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \right) \end{array} \right] \tag{7.37}$$

$$b = T_s \times \left[\begin{array}{l} \left(\left(k_0 \exp(-E / RT_{(k-1)}) \right) * \right. \\ \left. \left(k_1 * c_0 \exp\left(\dfrac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2} \right) * \\ \left(\dfrac{\left(\Delta H * i * m_{M(k-1)} C_{p,P} \right)}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))^2} \right) \end{array} \right] + T_s \times \left[\begin{array}{l} \left(\dfrac{-\left(\Delta H * i * m_{M(k-1)} \right)}{((m_{M(k-1)} C_{p,M}) + (m_{P(k-1)} C_{p,P}) + (m_W C_{p,W}))} \right) * \\ \left(k_0 \exp(-E / RT_{(k-1)}) * k_2 \left(k_1 * c_0 \exp\left(\dfrac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} \right)^{k_2 - 1} \right) * \\ \left(k_1 * c_0 * 10^{c_2 (a_0 / T_{(k-1)} - c_3)} * \exp\left(\dfrac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * \right. \\ \left. \left(\dfrac{c_1 (m_{M(k-1)} + m_{P(k-1)} + m_C) - c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} \right) \right) \end{array} \right] \tag{7.38}$$

$$c = T_S \times \left[\left(\frac{1}{h^{-1}+h_f^{-1}} \right) * \left(\frac{\left(\left(-T_{(k-1)} + {T_{jin(k-1)}}/{2} + {T_{jout(k-1)}}/{2} \right) * \left({PB_1\rho_M}/{(B_1\rho_M)^2} \right) * \left(((m_{M(k-1)}C_{p,M})+(m_{P(k-1)}C_{p,P})+(m_W C_{p,W})) \right) \right)}{((m_{M(k-1)}C_{p,M})+(m_{P(k-1)}C_{p,P})+(m_W C_{p,W}))^2} \right) \right] +$$

$$T_S \times \left[\frac{(\beta) * \left((h)^{-2} * \left((h) * \left(d_1 c_0 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} \right) * \exp\left(\frac{c_1 * m_{P(k-1)}}{((m_{M(k-1)}C_{p,M})+(m_{P(k-1)}C_{p,P})+(m_W C_{p,W}))} \right) * \left(\frac{c_1(m_{M(k-1)}+m_{P(k-1)}+m_C)-c_1 * m_{P(k-1)}}{((m_{M(k-1)}C_{p,M})+(m_{P(k-1)}C_{p,P})+(m_W C_{p,W}))^2} \right) \right)}{(h^{-1}+h_f^{-1})^2} \right]$$

(7.39)

$F_{3,3}$ is the partial derivative of the reactor temperature equation with respect to T :

Let $F_{3,3} = a+b+c+d$

To simplify the equation we let:

$$h = d_0 \exp(d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)}+m_{P(k-1)}+m_C}) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)})$$

$$\alpha = \left[\left(\frac{((\frac{P*m_{M(k-1)}}{B_1*\rho_M}+\frac{P*m_{P(k-1)}}{B_1*\rho_P}+\frac{P*m_W}{B_1*\rho_W})+B_2)}{((m_{M(k-1)}C_{p,M})+(m_{P(k-1)}C_{p,P})+(m_W C_{p,W}))} \right) \right]$$

$$\mu = \left((m_{M(k-1)}C_{p,M})+(m_{P(k-1)}C_{p,P})+(m_W C_{p,W}) \right)$$

$$\omega = \left(k_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * 10^{c_2(a_0/T_{(k-1)} - c_3)} \right)$$

$$\lambda = \Delta H * i * k_0 * m_{M(k-1)} \exp(\frac{-E}{RT_{(k-1)}})$$

So, from the discretised reactor temperature equation we can find $F_{3,3}$ as following:

$$a = \left(\frac{(\mu) * (-\dot{m}_M^{in} * C_{p,M})}{\mu^2} \right) + \left(\frac{\mu * UA_{loss}}{\mu^2} \right) + \frac{\left[\left(-\lambda * (\omega * Ln10 * (-c_2 a_0 / (T_{(k-1)} - c_3)^2) * k_2 (\omega)^{k_2 - 1} \right) + \left((\omega)^{k_2} * \lambda * (-ER / (RT_{(k-1)})^2 \right) \right]}{\mu}$$

(7.40)

$$b = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\alpha * (-T_{(k-1)}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * (\alpha) \right] + \left(h^{-2} * h \right) * \left(\begin{array}{l} d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * \\ 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)} * Ln10 * \\ \frac{-8 c_2 a_0}{(2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} - 4c_3)^2} \end{array} \right) \left(\alpha * (-T_{(k-1)}) \right) * \frac{}{h^{-2} + h_f^{-2}}$$

(7.41)

$$c = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\alpha * (\frac{T_{jout(k-1)}}{2}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * (0) \right] +$$

$$\left(h^{-2} * h \right) * \left(\begin{array}{l} d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * \\ 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)} * Ln10 * \\ \frac{-8c_2 a_0}{(2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} - 4c_3)^2} \end{array} \right)$$

$$\left(\alpha * (\frac{T_{jout(k-1)}}{2}) \right) * \frac{}{h^{-2} + h_f^{-2}}$$

(7.42)

$$d = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\alpha * (\frac{T_{jin(k-1)}}{2}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * (0) \right] +$$

$$\left(h^{-2} * h \right) * \left(\begin{array}{l} d_1 * c_0 \exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C}) * \\ 10^{c_2(a_0/2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)}/4 - c_3)} * Ln10 * \\ \frac{-8c_2 a_0}{(2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} - 4c_3)^2} \end{array} \right)$$

$$\left(\alpha * (\frac{T_{jin(k-1)}}{2}) \right) * \frac{}{h^{-2} + h_f^{-2}}$$

(7.43)

$F_{3,4}$ is the partial derivative of the reactor temperature equation with respect to T_{jout}:

Let $F_{3,4} = a + b + c$, where

$$a=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\alpha*(\frac{T_{jout(k-1)}}{2})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\alpha*(1/2)\right)\right]+$$

$$\left(h^{-2}*h\right)*\left(\begin{array}{l} d_1*c_0\exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)}+m_{P(k-1)}+m_C})* \\ 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10* \\ \frac{-4c_2a_0}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2} \end{array}\right)$$

$$\left(\alpha*(\frac{T_{jout(k-1)}}{2})\right)*\frac{}{h^{-2}+h_f^{-2}}$$

(7.44)

$$b=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\alpha*(\frac{T_{jin(k-1)}}{2})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*(0)\right]+$$

$$\left(h^{-2}*h\right)*\left(\begin{array}{l} d_1*c_0\exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)}+m_{P(k-1)}+m_C})* \\ 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10* \\ \frac{-4c_2a_0}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2} \end{array}\right)$$

$$\left(\alpha*(\frac{T_{jin(k-1)}}{2})\right)*\frac{}{h^{-2}+h_f^{-2}}$$

(7.45)

$$c = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\alpha * (-T_{(k-1)}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * (0) \right] +$$

$$\left(\alpha * \left(\frac{T_{jin(k-1)}}{2} \right) \right) * \frac{\left(h^{-2} * h \right) * \left(\begin{array}{l} d_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * \\ 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * Ln10 * \\ \frac{-4c_2 a_0}{(2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} - 4c_3)^2} \end{array} \right)}{h^{-2} + h_f^{-2}}$$

(7.46)

$F_{3,5}$ is the partial derivative of the reactor temperature equation with respect to T_{jin}:

Let $F_{3,5} = a + b + c$, where

$$a = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\alpha * \left(\frac{T_{jin(k-1)}}{2} \right) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\alpha * (1/2) \right) \right] +$$

$$\left(\alpha * \left(\frac{T_{jin(k-1)}}{2} \right) \right) * \frac{\left(h^{-2} * h \right) * \left(\begin{array}{l} d_1 * c_0 \exp\left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right) * \\ 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * Ln10 * \\ \frac{-4c_2 a_0}{(2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} - 4c_3)^2} \end{array} \right)}{h^{-2} + h_f^{-2}}$$

(7.47)

$$b=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\alpha*(\frac{T_{jout(k-1)}}{2})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*(0)\right]+$$

$$\left(h^{-2}*h\right)*\left(\begin{array}{l}d_1*c_0\exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)}+m_{P(k-1)}+m_C})*\\ 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10*\\ \frac{-4c_2a_0}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\end{array}\right)$$

$$\left(\alpha*(\frac{T_{jout(k-1)}}{2})\right)*\frac{}{h^{-2}+h_f^{-2}}$$

(7.48)

$$c=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\alpha*(-T_{(k-1)})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*(0)\right]+$$

$$\left(h^{-2}*h\right)*\left(\begin{array}{l}d_1*c_0\exp(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)}+m_{P(k-1)}+m_C})*\\ 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10*\\ \frac{-4c_2a_0}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\end{array}\right)$$

$$\left(\alpha*(\frac{T_{jin(k-1)}}{2})\right)*\frac{}{h^{-2}+h_f^{-2}}$$

(7.49)

Now we will find $F_{4,1}$ is the partial derivative of the jacket output temperature equation with respect to m_M:

Let $F_{4,1}=a+b+c$, where

$$\eta = \left[\left(\frac{\left(\left(\frac{P*m_{M(k-1)}}{B_1*\rho_M} + \frac{P*m_{P(k-1)}}{B_1*\rho_P} + \frac{P*m_W}{B_1*\rho_W} \right) + B_2 \right)}{(m_C C_{p,C})} \right) \right] \quad , \quad \xi = \left[\left(\frac{\left(\frac{P}{B_1*\rho_M} \right)*(m_C C_{p,C})}{(m_C C_{p,C})^2} \right) \right]$$

$$\psi = \left(\frac{c_1 m_{P(k-1)}}{m_{M(k-1)} + m_{P(k-1)} + m_C} \right)$$

$$a = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\eta * (T_{(k-1)}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\xi * T_{(k-1)} \right) \right] +$$

$$\left[\left(\eta * (T_{(k-1)}) \right) * \left(\frac{\left(h^{-2} * h \right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} \right)}{h^{-2} + h_f^{-2}} \right) \right]$$

(7.50)

$$b = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\eta * \left(\frac{-T_{jout(k-1)}}{2} \right) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\xi * \left(\frac{-T_{jout(k-1)}}{2} \right) \right) \right] +$$

$$\left[\left(\eta * \left(\frac{-T_{jout(k-1)}}{2} \right) \right) * \left(\frac{\left(h^{-2} * h \right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} \right)}{h^{-2} + h_f^{-2}} \right) \right]$$

(7.51)

$$c = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\eta * (\frac{-T_{jin(k-1)}}{2}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\xi * (\frac{-T_{jin(k-1)}}{2}) \right) \right] +$$

$$\left[\left(\eta * (\frac{-T_{jin(k-1)}}{2}) \right) * \left(\frac{\left(h^{-2} * h \right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * \frac{-c_1 m_{P(k-1)}}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} \right)}{h^{-2} + h_f^{-2}} \right) \right]$$

(7.52)

Now we will find $F_{4,2}$ is the partial derivative of the jacket output temperature equation with respect to m_P:

Let $F_{4,2} = a + b + c$, where

$$a = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\eta * (T_{(k-1)}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\xi * T_{(k-1)} \right) \right] +$$

$$\left[\left(\eta * (T_{(k-1)}) \right) * \left(\frac{\left(h^{-2} * h \right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * \frac{-c_1 m_{P(k-1)} + c_1 (m_{M(k-1)} + m_{P(k-1)} + m_C)}{(m_{M(k-1)} + m_{P(k-1)} + m_C)^2} \right)}{h^{-2} + h_f^{-2}} \right) \right]$$

(7.53)

$$b = \left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\eta * (\frac{-T_{jout(k-1)}}{2})\right)$$

$$= \left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\xi * (\frac{-T_{jout(k-1)}}{2})\right)\right] +$$

$$\left[\left(\eta * (\frac{-T_{jout(k-1)}}{2})\right) * \left(\frac{\left(h^{-2} * h\right) * \left(\begin{array}{c} d_1 * c_0 \exp(\psi) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} * \\ \frac{-c_1 m_{P(k-1)} c_1 (m_{M(k-1)}+m_{P(k-1)}+m_C)}{(m_{M(k-1)}+m_{P(k-1)}+m_C)^2} \end{array}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.54)

$$c = \left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\eta * (\frac{-T_{jin(k-1)}}{2})\right)$$

$$= \left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\xi * (\frac{-T_{jin(k-1)}}{2})\right)\right] +$$

$$\left[\left(\eta * (\frac{-T_{jin(k-1)}}{2})\right) * \left(\frac{\left(h^{-2} * h\right) * \left(\begin{array}{c} d_1 * c_0 \exp(\psi) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} * \\ \frac{-c_1 m_{P(k-1)} c_1 (m_{M(k-1)}+m_{P(k-1)}+m_C)}{(m_{M(k-1)}+m_{P(k-1)}+m_C)^2} \end{array}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.55)

Now we will find $F_{4,3}$ is the partial derivative of the jacket output temperature equation with respect to T :

Let $F_{4,3} = a + b + c$, where

$$a = \left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\eta * (T_{(k-1)})\right)$$

$$= \left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right) * (\eta)\right] +$$

$$\left[(\eta * (T_{(k-1)})) * \left(\frac{\left(h^{-2} * h\right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} * Ln10 * (-8c_2 a_0)}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.56)

$$b = \left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\eta * \left(\frac{-T_{jout(k-1)}}{2}\right)\right)$$

$$= \left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right) * (0)\right] +$$

$$\left[\left(\eta * \left(\frac{-T_{jout(k-1)}}{2}\right)\right) * \left(\frac{\left(h^{-2} * h\right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} * Ln10 * (-8c_2 a_0)}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.57)

$$c = \left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\eta * \left(\frac{-T_{jin(k-1)}}{2}\right)\right)$$

$$= \left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right) * (0)\right] +$$

$$\left[\left(\eta * \left(\frac{-T_{jin(k-1)}}{2}\right)\right) * \left(\frac{\left(h^{-2} * h\right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} * Ln10 * (-8c_2a_0)}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.58)

Now we will find $F_{4,4}$ is the partial derivative of the jacket output temperature equation with respect to T_{jout}:

Let $F_{4,4} = a + b + c + d$, where

$$a = \left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\eta * \left(\frac{-T_{jout(k-1)}}{2}\right)\right)$$

$$= \left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right) * \left(\frac{-\eta}{2}\right)\right] +$$

$$\left[\left(\eta * \left(\frac{-T_{jout(k-1)}}{2}\right)\right) * \left(\frac{\left(h^{-2} * h\right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)} * Ln10 * (-4c_2a_0)}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.59)

$$b=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\eta*(\frac{-T_{jin(k-1)}}{2})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*(0)\right]+$$
$$\left[\left(\eta*(\frac{-T_{jin(k-1)}}{2})\right)*\left(\frac{\left(h^{-2}*h\right)*\left(\frac{d_1*c_0\exp(\psi)*10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10*}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.60)

$$c=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\eta*(T_{(k-1)})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*(0)\right]+$$
$$\left[\left(\eta*(T_{(k-1)})\right)*\left(\frac{\left(h^{-2}*h\right)*\left(\frac{d_1*c_0\exp(\psi)*10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10*}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.61)

$$d=-\left(\frac{1}{m_C*C_{p,C}}\right) \tag{7.62}$$

Now we will find $F_{4,5}$ is the partial derivative of the jacket output temperature equation with respect to T_{jin}:

Let $F_{4,5}=a+b+c+d$, where

$$a=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\eta*(\frac{-T_{jin(k-1)}}{2})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\frac{-\eta}{2}\right)\right]+$$

$$\left[\left(\eta*(\frac{-T_{jin(k-1)}}{2})\right)*\left(\frac{\left(h^{-2}*h\right)*\left(\frac{d_1*c_0\exp(\psi)*10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10*}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.63)

$$b=\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*\left(\eta*(\frac{-T_{jout(k-1)}}{2})\right)$$

$$=\left[\left(\frac{1}{h^{-1}+h_f^{-1}}\right)*(0)\right]+$$

$$\left[\left(\eta*(\frac{-T_{jout(k-1)}}{2})\right)*\left(\frac{\left(h^{-2}*h\right)*\left(\frac{d_1*c_0\exp(\psi)*10^{c_2(a_0/2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}/4-c_3)}*Ln10*}{(2T_{(k-1)}+T_{jin(k-1)}+T_{jout(k-1)}-4c_3)^2}\right)}{h^{-2}+h_f^{-2}}\right)\right]$$

(7.64)

$$c = \left(\frac{1}{h^{-1} + h_f^{-1}} \right) * \left(\eta * (T_{(k-1)}) \right)$$

$$= \left[\left(\frac{1}{h^{-1} + h_f^{-1}} \right) * (0) \right] +$$

$$\left[\left(\eta * (T_{(k-1)}) \right) * \left(\frac{\left(h^{-2} * h \right) * \left(\frac{d_1 * c_0 \exp(\psi) * 10^{c_2 (a_0 / 2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} / 4 - c_3)} * Ln10 * \frac{-4c_2 a_0}{(2T_{(k-1)} + T_{jin(k-1)} + T_{jout(k-1)} - 4c_3)^2} \right)}{h^{-2} + h_f^{-2}} \right) \right]$$

(7.65)

$$d = \left(\frac{\dot{m}_C \left(T_{jin(k-1)} - T_{jin}(k - \theta_1 \right) / \theta_1}{m_C} \right) \tag{7.66}$$

Now we will find $F_{5,1}$ is the partial derivative of the jacket input temperature equation with respect to m_M :

$$F_{5,1} = 0 \tag{7.67}$$

Now we will find $F_{5,2}$ is the partial derivative of the jacket input temperature equation with respect to m_P :

$$F_{5,2} = 0 \tag{7.68}$$

Now we will find $F_{5,3}$ is the partial derivative of the jacket input temperature equation with respect to T :

$$F_{5,3} = 0 \tag{7.69}$$

Now we will find $F_{5,4}$ is the partial derivative of the jacket input temperature equation with respect to T_{jout}:

$$F_{5,4}=\left(\frac{1}{\theta_2}\right)-\left[\left(\frac{1}{\theta_2}+\frac{1}{\tau_P}\right)*\left(\frac{T_{jout(k-1)}-T_{jout}(k-\theta_2}{\theta_2}\right)\right] \tag{7.70}$$

Now we will find $F_{5,5}$ is the partial derivative of the jacket input temperature equation with respect to T_{jin}:

$$F_{5,5}=-\left(\frac{1}{\tau_P}\right) \tag{7.71}$$

Next step is to find the Jacobian matrix of partial derivatives of $h(\bullet)$ with respect to x:

$$H_k=\left[\partial h(x,k)\middle/\partial x\right]=\begin{bmatrix}1 & 1 & 1 & 1 & 1\end{bmatrix} \tag{7.72}$$

In this work, the uncertainties of the system are modelled as process noise in the covariance matrix of the EKF as following:

$$Q_k=10^{-5}\times diag\left(\begin{pmatrix}0 & 0 & 3 & 3 & 3\end{pmatrix}\right) \tag{7.73}$$

The diagonal covariance matrix of the measurement noise is set with standard deviation $\sigma(y)=0.05k$ and is given as:

$$R_k=10^{-3}\times diag\left(\begin{pmatrix}5 & 5 & 5\end{pmatrix}\right) \tag{7.74}$$

The state covariance matrix is set as:

$$P_k=10^{-7}\times diag\left(\begin{pmatrix}1 & 1 & 1 & 1 & 1\end{pmatrix}\right) \tag{7.75}$$

Moreover, the measurement vector is selected as:

$$y_{EKF} = \left[m_M \ , m_P \ , T \ , T_{jout} \ , T_{jin} \right]^T \tag{7.76}$$

The choice of these covariance matrixes will affect the performance and the convergence of the EKF. The tuning of the EKF involves an iterative modification of the covariance in order to yield the best estimates of the states. Changing the covariance matrices Q and R affects both transient and steady state operation of the filter. After implementing the EKF and in order to get a better estimation of the states, the initial values of Q_k and R_k are selected randomly and tuned accordingly. From experience, the values of R_k matrix elements are higher than the values of Q_k matrix elements.

Figures (7.1)-(7.3) show the estimated value of the reactor temperature, jacket output temperature and jacket input temperature compared to the measured nominal values. These results obtained from simulation of EKF estimation. It can be clearly seen that the EKF achieves a good estimates of these variables under normal nominal mode.

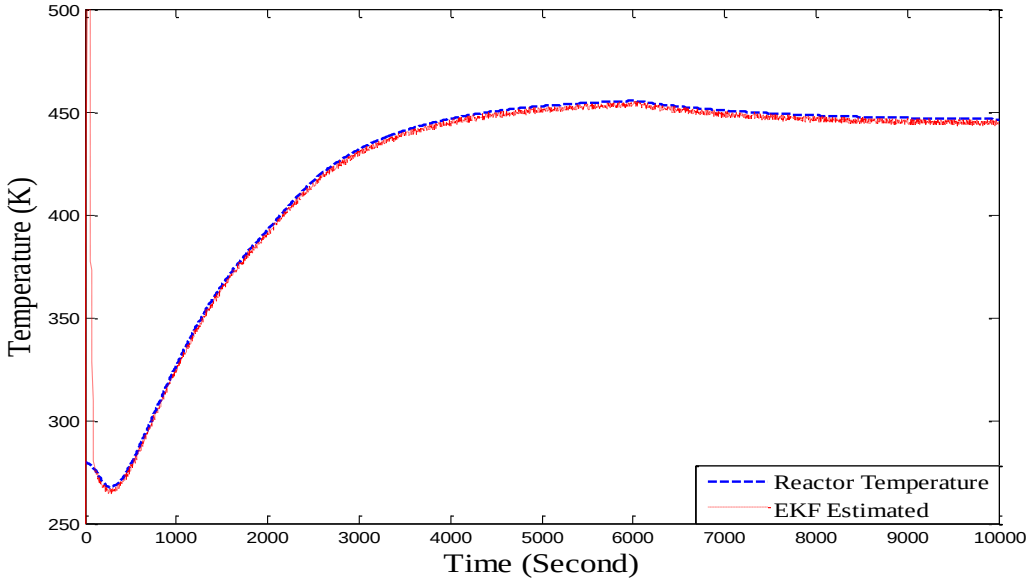


Figure 7.1 Reactor temperature estimation using EKF

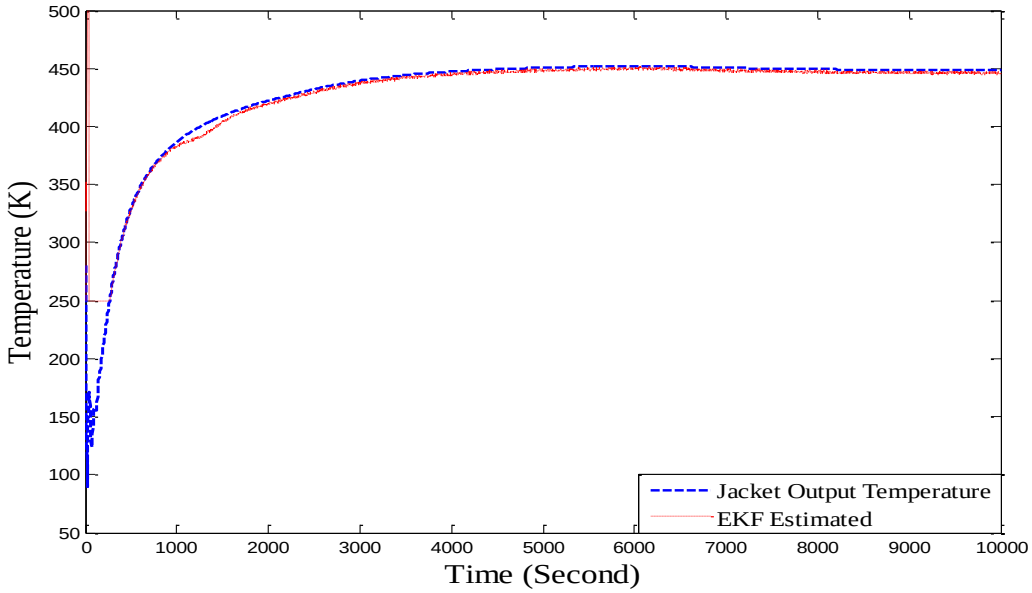


Figure 7.2 Jacket output temperature estimation using EKF

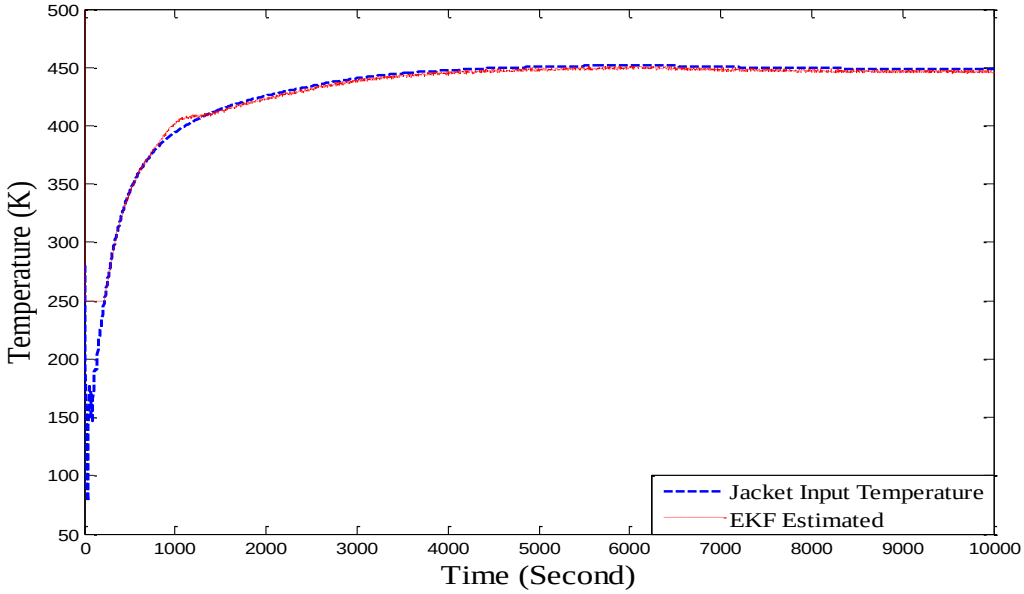


Figure 7.3 Jacket input temperature estimation using EKF

7.4. Fault Detection

The faulty data is obtained by simulating different faults in the proposed reactor. These faults are classified as three sensor faults and one actuator fault. The sensor faults are jacket input temperature sensor fault, jacket output temperature sensor fault, and reactor

temperature sensor fault, and the actuator fault is the inlet temperature. These faults are simulated as described in previous chapters.

Figure 7.4 demonstrates the fault detection method. An on-line implementation of the EKF is presented. During normal behaviour of the process and after EKF convergence, the innovation fluctuation is a small and a white noise sequence of a zero mean. When any fault occurs, the innovation will be influenced by the fault type. In addition, an on line implementation of the standardized hypotheses statistical test is presented in order to distinguish normal behaviour of the process from an abnormal behaviour.

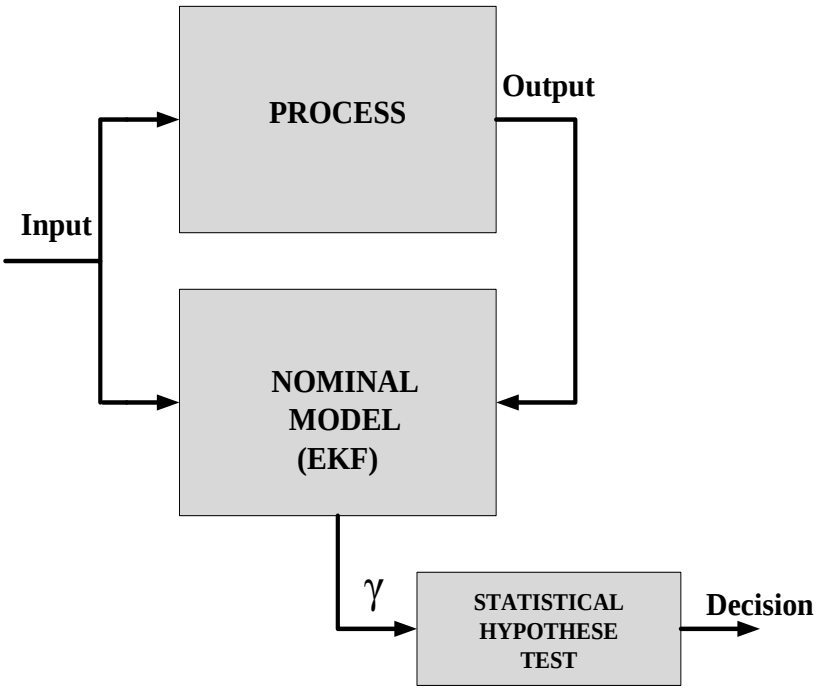


Figure 7.4 FD scheme by standardized hypotheses statistical test

7.5. Fault detection via hypothesis testing

After the residual is generated by the EKF estimation, the decision making is made by a statistical method called hypothesis testing.

7.5.1. Generation of innovation sequence

The innovation sequence is the difference between the measured output and estimated output and is defined as:

$$\gamma_k = Y_k - h * \left(\widehat{X}_{k|k-1}\right) \tag{7.77}$$

7.5.2. Statistics of the innovation sequence

In the normal healthy operating conditions, the innovation sequence γ_k is a zero mean Gaussian white noise sequence with covariance (Mehra and Peschon, 1971).

$$S_k = H_k \; P_{k|k-1} \; H_k{}^T + R_k \tag{7.78}$$

7.5.3. Hypothesis testing

The standardized innovation statistical parameters (mean and variance) that obtained under an abnormal behaviour will be compared on-line with those obtained under normal behaviour. So, two hypotheses are defined; hypothesis H_0 refers to the innovation statistics in the normal mode, and hypothesis H_1 refers to the innovation in an abnormal mode. Mehra and Peschon (1971) described that it's more appropriate to consider the standardized innovation sequence for the hypothesis testing purposes. This sequence is defined as

$$\eta_k = \left(H_k \; P_{k|k-1} \; H_k{}^T + R_k\right)^{-0.5} * \left(Y_k - h * \left(\widehat{X}_{k|k-1}\right)\right) \tag{7.79}$$

$$\eta_k = \left(H_k \; P_{k|k-1} \; H_k{}^T + R_k\right)^{-0.5} * \gamma_k \tag{7.80}$$

Then

$$E\left\{\eta_j \; \eta_k{}^T\right\} = I \; \delta_{jk} \tag{7.81}$$

Where I denotes the identity matrix.

In the normal operating conditions, η_k has a zero mean and a unit variance. So, when any fault occurs, the standardized innovation sequence η_k will depart from zero mean. In this study, the hypothesis testing on mean is applied. This test checks whether the observed standardized sequence has a zero mean or not. The mean of the standardized innovation sequence is estimated as

$$\hat{\bar{\eta}} = \frac{1}{N}\sum_{k=1}^{N}\eta_k \tag{7.82}$$

Where N is the sample size and $\bar{\eta}$ refers to the true mean. Under the null hypothesis H_0, $\hat{\bar{\eta}}$ has a Gaussian distribution with zero mean and covariance

$$E\left\{\hat{\bar{\eta}}\,\hat{\bar{\eta}}^T\right\} = I/N \tag{7.83}$$

Therefore at any given significance level of acceptance (hypothesis H_1), the null hypothesis H_0 is rejected whenever

$$\left|\hat{\bar{\eta}}\right| > 1.96 * I/\sqrt{N} \tag{7.84}$$

7.6. Simulating Faults

In this study, the faulty data is obtained by simulating different faults in the proposed reactor. The classification and structure of faults are done in the same patter in previous chapters 4-6.

7.7. Performances and discussion

Before applying the fault detection method, the knowledge model parameters and the initial state vector are correctly initialized. The EKF algorithm is tuned using process noise covariance matrix Q , and measurement noise covariance matrix R by hand until obtaining a compromise between quick detection and as small as possible of false alarm

rate. The knowledge of these two covariance matrices is the mean key to adjust the EKF. Figures 7.5 to7.7 demonstrate the evolution of the standardized hypothesis statistical test for the four simulated faults. It can be clearly noticed that all faults have been clearly detected. Moreover, no false alarms are thereby produced, so this verifies that the proposed scheme has shown excellent diagnostic performance. Figures 7.5 to 7.7 show two different regions, which are fault region and confidence region. The fault region is defined as at any given significance level of acceptance (hypothesis H_1), the null hypothesis H_0 is rejected whenever $|\bar{\eta}|>\alpha$. Where α is the threshold and is selected as 5%. In addition, it should be pointed out from Figure 7.7 that the actuator fault (inlet temperature fault) occurs at $2800\ s$, and is detected using statistical test with $200\ s$. The dynamic error is defined as an initial value effects.

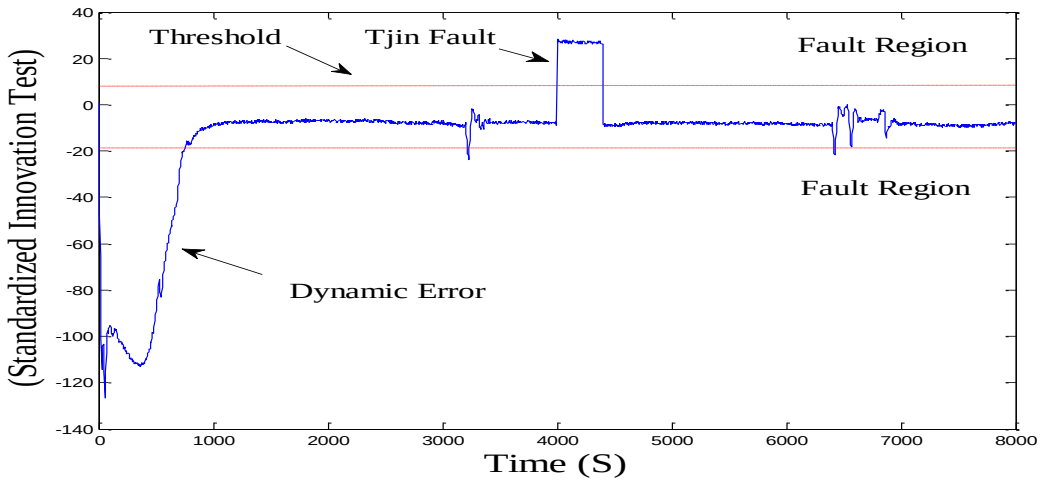


Figure 7.5 Evolution of Tjin using standardized hypotheses test

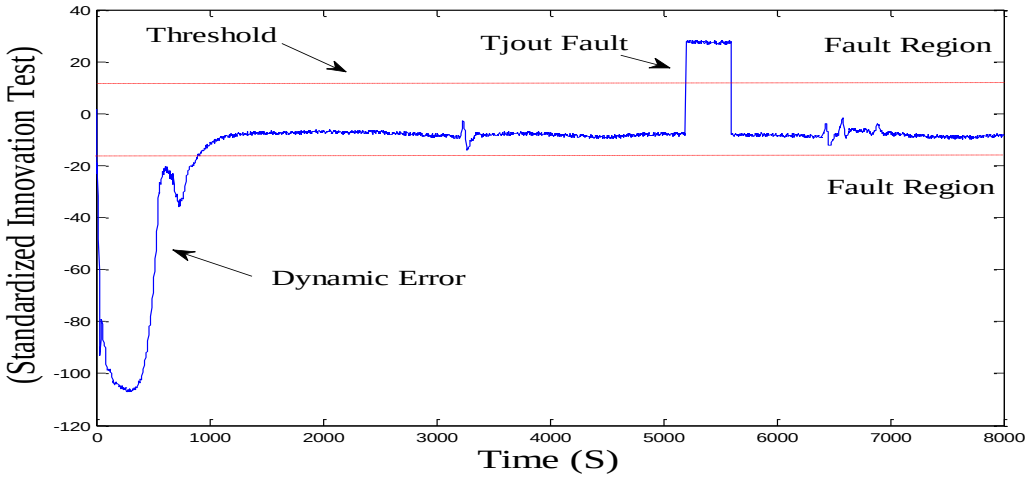


Figure 7.6 Evolution of Tjout using standardized hypotheses test

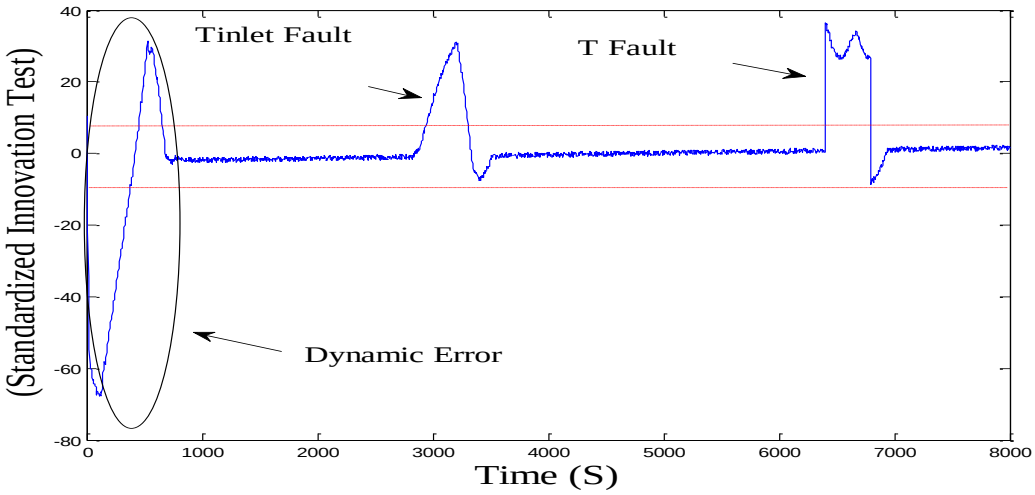


Figure 7.7 Evolution of T using standardized hypotheses test

7.8. Summary

In this study, a fault detection scheme based on standardized hypothesis of statistical tests (Mehra test) generated by extended kalman filter (EKF) was developed. The proposed method is applied for online fault detection in exothermic semi-batch polymerization reactor. The simulation results show that all faults were clearly detected. Moreover, no false alarms are thereby produced, so this verifies that the proposed scheme has shown excellent detection performance. In order to implement the proposed method, the EKF

algorithm was tuned using process noise covariance matrix Q , and measurement noise covariance matrix R by hand until obtaining a compromise between quick detection and as small as possible of false alarms. The knowledge of these two covariance matrices was the mean key to adjust the EKF. Moreover, the ability of the EKF to estimate both the states and parameters was the main key for applying the proposed method successfully. However, the implementation of the proposed method requires a prior knowledge of the model, and the linearization technique for the nonlinear model is a challenging task in order to implement EKF.

Chapter 8

Nonlinear Observer Based FD

In the recent engineering issues, the task of the detection and diagnosis of the system failures are being intensively more significant. This section presents a robust fault diagnosis scheme for abrupt and incipient faults in nonlinear dynamic system. A general framework is developed for model-based fault detection and diagnosis using on-line approximators and adaptation/learning schemes. In this framework, neural network models constitute an important class of on-line approximators. The changes in the system dynamics due to fault are modelled as nonlinear functions of the state, while the time profile of the fault is assumed to be exponentially developing. The changes in the system dynamics are monitored by an on-line approximation model, which is used for detecting the failures. A systematic procedure for constructing nonlinear estimation algorithm is developed, and a stable learning scheme is derived using Lyapunov theory. Simulation studies are used to illustrate the results and to show the effectiveness of the fault diagnosis methodology.

8.1. Modelling A non-adiabatic Continues Stirred Tank Reactor

A common chemical system encountered in the process industry is the continuously stirred tank reactor (CSTR). Here we will study a jacketed non-adiabatic tank reactor, the vessel is assumed to be perfectly mixed, and a single first-order exothermic and irreversible reaction, $A \rightarrow B$ take place. A schematic diagram of the vessel and the surrounding cooling jacket is shown in figure 8.1. It can be noticed that in the reality the coolant flow is normally surrounding whole jacket. A model of the CSTR is required for

more advanced control approaches. The inlet stream of reagent A is fed into the tank (the volume V in the reactor tank is kept constant). The control strategy requires that the jacket temperature $u_3(t)$ is manipulated in order to keep the concentration of reagent A $y_1(t)$ at the desired level, in spite of disturbances arising from the inlet feed stream concentration and temperature (inputs $u_1(t)$ and $u_2(t)$). As the temperature in the tank $y_2(t)$ can vary significantly during operation of the reactor.

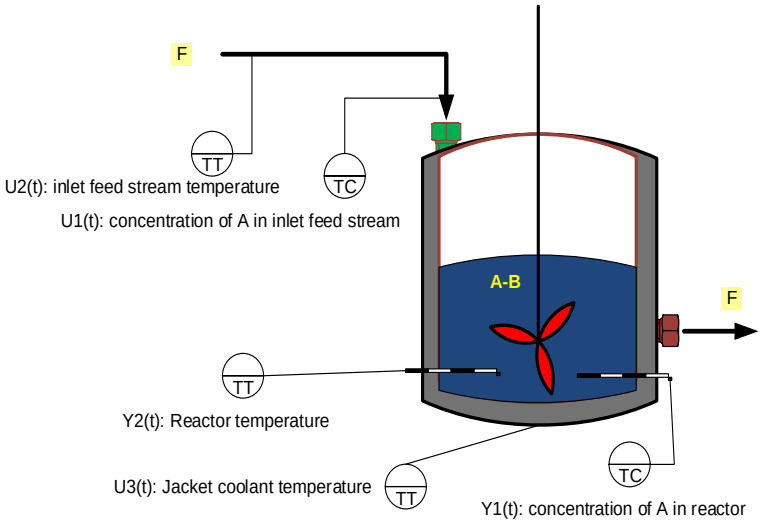


Figure 8.1 Schematic diagram of CSTR (Zhai and Ma,2012).

The CSTR system is modelled using basic accounting and energy conservation principles. The change of the concentration of reagent A in the vessel per time unit can be modelled as:

$$dCA/dt = F/V * (CAf(t) - CA(t)) - r(t) \quad (8.1)$$

Where the first term expresses concentration changes due to differences between the concentration of reagent A in the inlet stream and in the vessel, and the second term expresses concentration changes that occurs due to the chemical reaction in the vessel. The reaction rate per unit volume is described by Arrhenius rate law:

$$r(t)=k*\exp\left(-E/(R*T(t))\right)*CA(t) \quad (8.2)$$

Which states that the rate of chemical reaction increases exponentially with the absolute temperature. k Is here an unknown non-thermal constant, E is the activation energy, R Boltzmann's ideal gas constant and $T(t)=y_2(t)$ the temperature in the reactor. Similarly, using the energy balance principle (assuming constant volume in the reactor), the temperature change per time unit in the reactor can be modelled as:

$$dT(t)/dt=F/V(Tf(t)-T(t))-(H/(cp*rho))*r(t)-(U*A)/(cp*rho*V)*(T(t)-T_j(t) \quad (8.3)$$

Where the first and third terms describe changes due to that the feed stream temperature $Tf(t)$ and the jacket coolant temperature $T_j(t)$ differ from the reactor temperature. The second term is the influence on the reactor temperature caused by the chemical reaction in the vessel. In this equation, H is a heat reaction parameter, cp a heat capacity term, rho a density term, U an overall transfer coefficient and A the area for the heat exchange (coolant/vessel area) (Zhai and Ma, 2012).

The CSTR has three input signals:

$u_1(t)=CAf(t)$ Concentration of A in inlet feed stream $[kg\ mol\ m^3]$

$u_2(t)=Tf(t)$ Inlet feed stream temperature $[K]$

$u_3(t)=T_j(t)$ Jacket coolant temperature $[K]$

And two output signals:

$y_1(t)=CA(t)$ Concentration of A in reactor tank $[kg\ mol\ m^3]$

$y_2(t)=T(t)$ Reactor temperature $[K]$

After lumping together some of the original parameters we end up with eight different model parameters as given follows:

Table 8.1 parameter values

F	Volume flow rate	$1\ [m^3/h]$
V	Volume in reactor	$1\ [m^3]$
k	Non-thermal factor	$3.5\times e^7\ \ [1/h]$
E	Activation energy	$11850\ \ [kcal/kgmol]$
R	Boltzmann's gas constant	$1.98589\ \ [kcal/(kgmol*K)]$
H	Heat of the reaction	$-5960\ [kcal/kgmol]$
$HD=cp*rho$	Heat capacity time density	$480\ \ [kcal/(m^3*K)]$
$HA=U*A$	Overall heat times tank area	$145\ \ [kcal/(K*h)]$

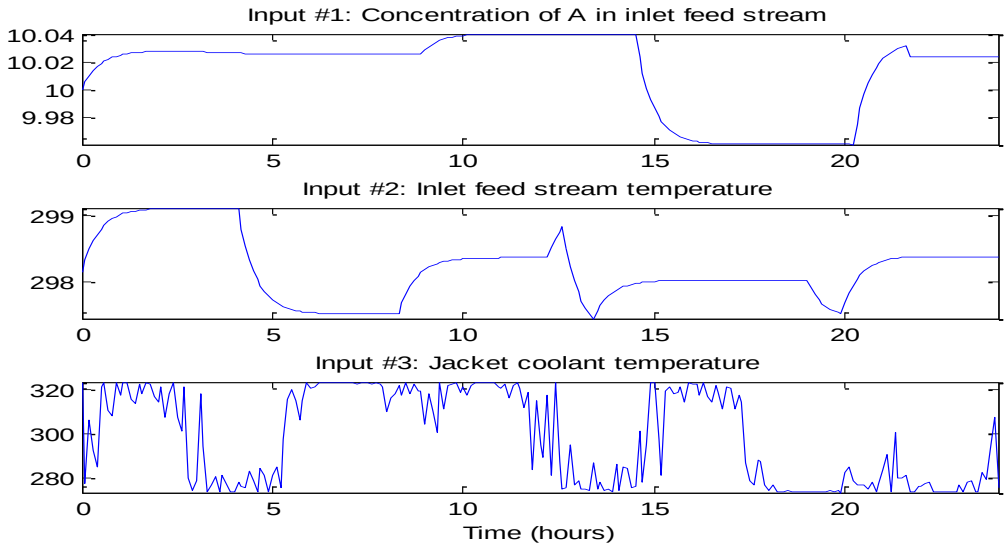


Figure 8.2 Inputs of CSTR

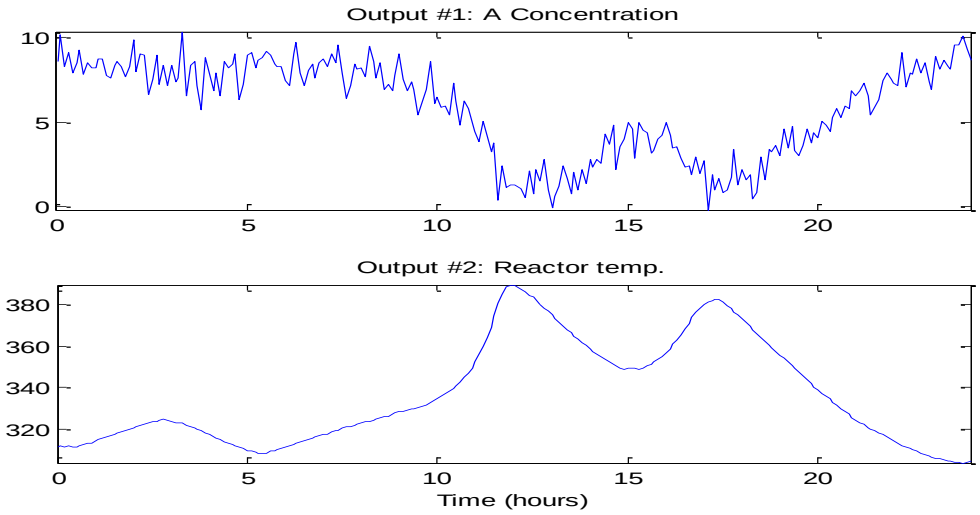


Figure 8.3 Outputs of CSTR

8.2. Fault diagnosis scheme

8.2.1. Representation of Failures

The class of dynamical systems under study is described by

$$\dot{x}(t) = \xi(x(t), u(t)) + \beta(t-T)\, f(x(t), u(t)) \tag{8.4}$$

Where $x \in \Re^n$ is the state vector, $u \in \Re^m$ is the input vector, $\xi, f : \Re^n * \Re^m \to \Re^m$ are smooth vector fields, $T \geq 0$ is the beginning time of the failure, and β is a square $n*n$ matrix function representing the time profiles of failures. We consider incipient and abrupt faults that are modelled by

$$\beta(t-T) = diag(\beta_1(t-T), \beta_2(t-T), \cdots, \beta_n(t-T)) \tag{8.5}$$

Where

$$\beta_i(\tau) = \begin{cases} 0 & if\ \tau \leq 0 \\ 1 - e^{-\rho i \tau} & if\ \tau \geq 0 \end{cases} \qquad i = 1,2,\ldots,n \tag{8.6}$$

And $\rho_i > 0$ is unknown constant that represent the rate at which the failure in state x_i evolves. For large value of ρ_i, the time profile function β_i approaches a step function ,

which models abrupt failures. The objective is to design a fault diagnosis scheme that processes input and state information to determine the presence and characteristics of any incipient and abrupt faults. Since this task does not address fault accommodation, below we make the standard assumption that the control input u and the state vector x remain bounded prior and after the occurrence of a fault:

The "healthy" system in the absence of any faults is described by

$$\dot{x}h(t)=\xi(xh(t),u(t)):=\xi^{*}(xh(t),u(t))+\tilde{\xi}(xh(t),u(t)) \tag{8.7}$$

Where ξ^{*} represents the nominal dynamics (known) and $\tilde{\xi}$ characterizes any discrepancy between the actual plant and nominal model that may occur due to modelling errors. It is well known in the fault diagnosis literature that the presence of modelling errors, in general, increases the probability of false alarms. During the last few years the designs of so-called robust fault diagnosis schemes have resulted in a variety of tools for dealing with such modelling uncertainties. An intuitive approach is to use a small threshold in the residual error to account for modelling uncertainties; in this case fault is declared if the residual error is greater than the selected threshold. Another approach attempts to decouple the effects of faults and modelling errors as a way of improving robustness. In this work we first consider the ideal case where $\xi=0$ and then the case where $|\tilde{\xi}(x,u)|\leq\xi_0$ for all $(x,u)\in(x*u)$, where ξ_0 is a known constant. In general, the design and analysis of robust diagnosis architectures based on nonlinear modelling techniques requires further investigation.

In many system applications there are more state variables than sensors. Therefore, the availability for measurement of the full state vector is a critical and limiting assumption. The design and analysis of fault diagnosis schemes using OLA approach for input-output systems becomes considerably more complex. The separation principle which for linear

systems allows combination of state-feedback controllers with state observers does not hold for nonlinear systems.

8.2.2. Nonlinear Estimator

The failure representation described by (8.4) provides a framework for characterizing a wide class of faults. Demetriou and Polycarpou (1998), Polycarpou and Helmicki (1995) In general, the magnitude of faults in practical applications depends on the state of the system as well as the system input. The nonlinear fault representation (8.4) captures these dependencies of f on the state x and input u, furthermore, since the above nonlinear fault representation is function of the control input u, the fault detection scheme works even in the case where the feedback control compensates the effect of small incipient faults on the system output. The price that one has to pay for the potential to model a larger class of failures is the need to approximate unknown nonlinear functions, which leads to nonlinear fault diagnosis techniques. This can be realized by utilization of parameterized OLA structure with adjustable parameters. Such an adaptive nonlinear estimator is given by

$$\hat{x}=W(s)[z] \tag{8.8}$$

$$Z=\xi(x,u;\hat{\theta}) \tag{8.9}$$

$$\dot{\hat{\theta}}=\eta(x,u,\hat{x};\hat{\theta}) \tag{8.10}$$

Where $W(s)$ is $n*n$ stable filter matrix, (8.8) and (8.9) represent an observer-based nonlinear estimation scheme, and (8.10) is the adaptive law of the adjustable parameters. Next we proceed to the design of $W(s)$, ξ and η.

Now we consider the construction of a nonlinear estimator for modelling deviation in system dynamics due to failure. Based on the system representation described by (1), we choose an estimated model of the form

$$\dot{\hat{x}}=A\hat{x}(t)+\xi^{*}(x(t),u(t))+\hat{f}(x(t),u(t);\hat{\theta}(t))-Ax(t) \qquad (8.11)$$

Where $\hat{x}\in\Re^{n}$ is the estimated state vector, $\hat{f}$ represents an online approximation (OLA) model, $\hat{\theta}$ is a vector of adjustable parameters or weights, and A is a constant square matrix of dimension $n*n$, we choose eigenvalues lie in the left-half complex, A is a stability matrix. The initial value of the estimated parameter vector for the estimated model (8.11) is $\hat{\theta}(0)=\hat{\theta}^{0}$ chosen such that $\hat{f}(x,u;\hat{\theta})=0$ for all (x,u), corresponding to the case of no failure (healthy condition), while the initial value of the estimated state vector is selected as $\hat{x}(0)=x(0)$. Starting from these initial conditions, the main objective is to adjust (using input/output information) the parameter estimate $\hat{\theta}(t)$ at each time t so that $\hat{f}(x,y,\hat{\theta})$ approximates the unknown function $\beta(t-T)\ f(x,u)$ as closely as possible. Polycarpou and Helmicki (1995), Demetriou and Polycarpou (1998) explained that, once this is achieved then the output of the online approximator $\hat{f}$ can be used to detect and diagnose as well as accommodate any system failures. The online approximator, denoted by $\hat{f}$ represents the adjustable component of the estimation model.

Demetriou and Polycarpou (1998), Polycarpou and Helmicki (1995) proved that, to construct the estimated model (8.11) the following assumptions need to be made:

- The state vector x is available for measurement. This is a critical assumption that limits the applicability of this approach. Removal of this assumption requires the use of nonlinear observers. It is noted that the time derivative of the state vector $\dot{x}$

is not assumed to be available for measurement so that the overall learning scheme is free of any differentiators.

- The nominal system (in the absence of any failures) described by $\xi^*(x(t),u(t))$ is known. In practice, the inevitable presence of modelling errors will cause some discrepancy between the actual plant and the nominal model. The issue of robustness is further investigated in the development of learning schemes and in the simulation example (both described in details below) where modelling inaccuracies and measurement noise are included.

A block diagram representation of the estimated model (8.11) is described in figure 8.4. The construction of an accurate nonlinear model-based estimator, able to follow any variations in the physical system, is a crucial component of the overall learning scheme.

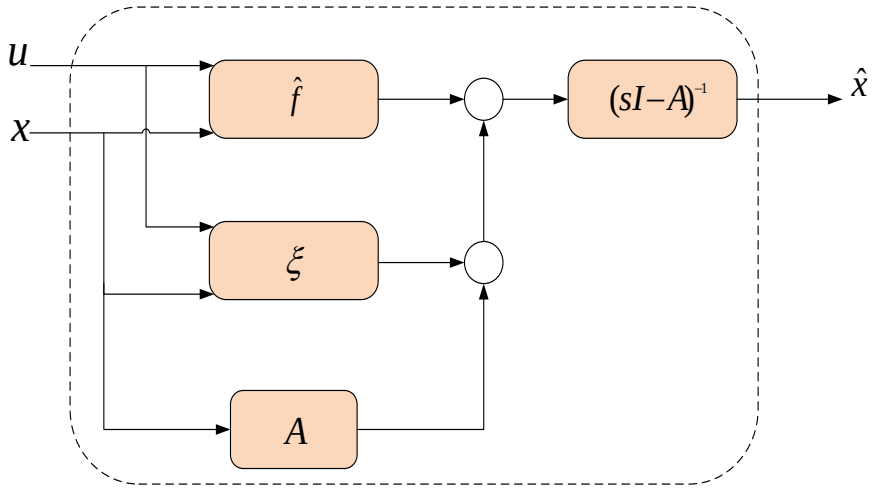


Figure 8.4 Block diagram representation of estimation model (Polycarpou and Vemuri, 1995).

8.2.3. On-line approximators

Polycarpou and Vemuri (1995) explained that, the adjustable component of the estimated model (8.11) is the on-line approximator $\hat{f}$,depending on its specific structure, $\hat{f}$ is also referred to as an approximation model or network. In the network formulation, the $(n+m)$

dimensional vector $z:=(x,u)$ is the input to the network, $\hat{\theta}\in\Re^q$ is a set of adjustable parameters or weights in vector form, and $\phi:=\hat{f}(z;\hat{\theta})$ is the output of the network. By changing the value of $\hat{\theta}$ it is possible to change the input/output $z\mapsto\phi$ response of the network $\hat{f}$ and hence monitor the physical system for different kinds of failures.

From an analytical viewpoint it is convenient to distinguish between linearly and nonlinearly parameterized approximation methods. In the case of linearly parameterized approximators, $\hat{f}$ is of the form:

$$\hat{f}(z;\hat{\theta})=\Omega(z)^T\ \hat{\theta} \tag{8.12}$$

In this work a class of radial basis function (RBF) neural networks is used as an online approximator for detection. The output of RBF networks is of the form:

$$\hat{f}_n(z;\hat{\theta}):=\left\{\sum_{i=1}^{n}\hat{\theta}_i\ w_i(z)\ :\hat{\theta}_i\in\Re\right\} \tag{8.13}$$

Where w_i is the output of i^{th} the basis function. The Gaussian function

$$w_i(z):=\exp\left(-|z-c_i|^2\Big/\sigma_i^2\right) \tag{8.14}$$

Where c_i and σ_i are the $i-th$ centre and width respectively, is usually chosen as the basis function. RBF networks are also capable of universal approximation. The approximation properties of ERBF networks are similar to those of spline functions. For example if the centre and width are kept fixed then RBF networks are linearly parameterized approximators; if they are allowed to vary then RBF networks become nonlinearly parameterized.

8.3. Learning Schemes

In the presence of system faults, changes in the dynamics cause a mismatch in the behaviours of the estimated and the nominal system model. The objective of a learning schemes is to develop an adaptive procedure that not only detects changes in the dynamics but also able to learn these changes for the purposes of identifying and correcting the fault. Therefore, learning is an inherent component of a FDA architecture, especially for unanticipated faults. We now describe a methodology for designing and analysing learning schemes based on the on-=line approximation approach discussed above.

We start by rewriting $\tilde{f}$ as:

$$\begin{aligned}\tilde{f}(x,u,\hat{\theta},t) &= \beta(t-T)\, f(x,u) - \hat{f}(x,u;\hat{\theta}) \\ &= \beta(t-T)\, \hat{f}(x,u;\hat{\theta}^*) - \hat{f}(x,u;\hat{\theta}) + v(t)\end{aligned} \tag{8.15}$$

Where $\hat{\theta}^* \in \Re^q$ is a constant parameter vector, and v is denotes the approximation error, given by:

$$v(t) := \beta(t-T)[\, f(x(t),u(t)) - \hat{f}(x(t),u(t);\hat{\theta}^*)] \tag{8.16}$$

The approximation error v is a critical quantity, representing the minimum possible deviation between the unknown function f and the output of the on-line approximator $\hat{f}$. Ideally we would like to have $v(t)=0$; in other words, we wish to approximate the function f by exactly letting $\hat{\theta}=\hat{\theta}^*$, where $\hat{\theta}^*$ is some "optimal" parameter estimate. Unfortunately, this is not always possible, and a residual approximation error is something that needs to be dealt with. The type of on-line approximator, the number of nodes, and the number of network layers are some of the factors that influence the value of v (Polycarpou and Helmicki, 1995, Demetriou and Polycarpou, 1998)

The optimal parameter vector $\hat{\theta}^*$ is an "artificial" quantity required only for analytical purposes. We choose $\hat{\theta}^*$ as the value of $\hat{\theta}$ that minimizes the distance between $f(x,u)$ and $\hat{f}(x,y,\hat{\theta})$ overall (x,u) in some compact (i.e., closed and bounded) learning domain D, subject to the restriction that $\hat{\theta}^*$ belongs to a compact, convex region $M_{\hat{\theta}} \subset \Re^q$ i.e.

$$\hat{\theta}^* := \arg \min_{\hat{\theta} \in M_{\hat{\theta}}} \left\{ \sup_{(x,u) \in D} |f(x,u) - \hat{f}(x,u;\hat{\theta})| \right\} \tag{8.17}$$

In the development of the adaptive law, the parameter estimate vector $\hat{\theta}$ is also restricted within $M_{\hat{\theta}}$, using a projection algorithm. By doing so, we avoid any numerical problems that may otherwise arise due to very large parameter values. More importantly, the projection algorithm prevents parameter drift, a phenomenon that may occur with standard adaptive laws in the presence of modelling uncertainty. One of the problems associated with the projection algorithm is the selection of an appropriate region $M_{\hat{\theta}}$ in the parameters space $\Re^q$. In general $M_{\hat{\theta}}$ should be selected such that it contains the "optimal" parameter vector $\hat{\theta}^*$, which is the reason $\hat{\theta}^*$ is restricted within the region $M_{\hat{\theta}}$ in (8.17). This restriction may undermine the approximation power of $\hat{f}$ by increasing the approximation error v, however, by selecting the "size" of $M_{\hat{\theta}}$ sufficiently large, the increase will be negligible.

Now using (8.11),(8.14), the output estimation error e satisfies the following differential equation:

$$\dot{e} = Ge + \beta(t-T)\,\hat{f}(x,u;\hat{\theta}^*) - \hat{f}(x,u;\hat{\theta}) + v \tag{8.18}$$

Demetriou and Polycarpou (1998) explained that, based on (8.13), we use the Lyapunov synthesis approach to derive the adaptive law for updating the parameter estimates. The

Lyapunov synthesis approach is based on choosing a Lyapunov function whose time derivative can be made negative semi-definite by an appropriate adaptive law. Hence, in this approach the adaptive law is derived by the Lyapunov function in contrast to optimization methods where minimization techniques such as gradient descent and least squares are used to derive the adaptive law. The appeal of the Lyapunov synthesis approach is that, when applicable, it guarantees the stability of the adaptive scheme.

In our case, the Lyapunov synthesis approach yields the following adaptive law for updating the parameter estimates:

$$\dot{\hat{\theta}} = P\{\Gamma Z e\} \tag{8.19}$$

Where

$$Z := \left[\partial \hat{f}(x, u; \hat{\theta}) \Big/ \partial \hat{\theta} \right]^T \quad , \quad \Gamma = \Gamma^{\mathrm{T}} > 0 \tag{8.20}$$

Where e is the estimation error, Γ is the positive definite matrix $\Gamma \in \Re^{q*q}$ is known as the learning rate matrix, while $Z \in \Re^{q*n}$ is the sensitivity function between the output of the network approximator and the adjustable or weights. The adaptive law derived by the Lyapunov synthesis approach is modified by the use of a projection algorithm P, so that the parameter estimates $\hat{\theta}$ remain within the bounded region $M_{\hat{\theta}}$. The projection algorithm goes into effect only if the parameter estimate vector $\hat{\theta}$ reaches the boundary of the region $M_{\hat{\theta}}$, denoted by $\overline{M}_{\hat{\theta}}$, and is directed outwards. In such situation, the algorithm projects the standard adaptive law (8.19) onto the tangent hyper plane of $\overline{M}_{\hat{\theta}}$, at the current value of $\hat{\theta}$, denoted by $\hat{\theta}_t$.

Polycarpou and Helmicki (1995) illustrated that, the projection algorithm is illustrated geometrically in figure 8.5. Thus, the overall adaptive law for updating the parameter estimates of the on-line approximator, using the Lyapunov synthesis approach and the projection modify cation, is given by:

$$\dot{\hat{\theta}}=\Gamma Ze-X^{*}\,\Gamma\left(\hat{\theta}\hat{\theta}^{T}\Big/_{|\hat{\theta}|^{2}}\right)\Gamma Ze \tag{8.21}$$

Where X^{*} denotes the indicator function given by:

$$X^{*}=\begin{cases}0, & \text{if } (|\hat{\theta}|<M_{\hat{\theta}}) \text{ or } (|\hat{\theta}|<\overline{M}_{\hat{\theta}} \text{ and } \hat{\theta}^{T}\Gamma Ze\le 0\\ 1, & \text{if } (|\hat{\theta}|<M_{\hat{\theta}}) \text{ and } \hat{\theta}^{T}\Gamma Ze>0\end{cases} \tag{8.22}$$

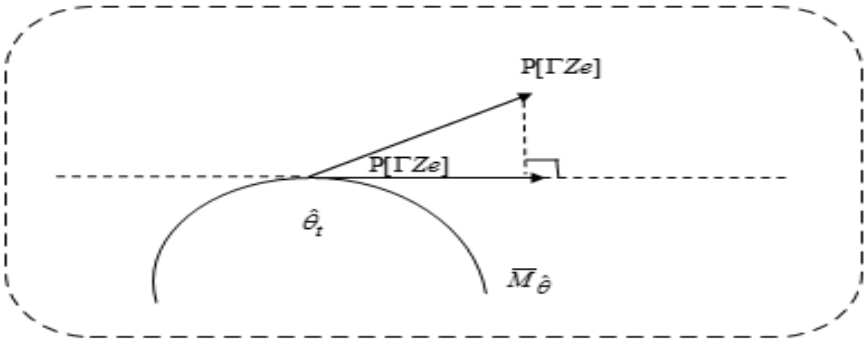


Figure 8.5 Geometric interpretation of projection algorithm (Polycarpou and Vemuri, 1995).

8.4. Stability and Robust Fault Diagnosis

Under ideal conditions of no modelling errors, a fault is declared whenever the output of the online approximator $y=\hat{f}(x,u;\hat{\theta})$ becomes nonzero. A straightforward and practical way of improving the robustness of the algorithm with respect to modelling uncertainties is to start adaptation whenever the state error is above a certain threshold. This approach to improving robustness is incorporated into the learning methodology developed above by modifying the adaptive law (8.19) as follows:

$$\dot{\hat{\theta}} = P\{\Gamma Z D[e]\} \tag{8.23}$$

Where $D[\bullet]$ is the dead-zone operator, defined as

$$D[e] := \begin{cases} 0, & if\ |e| \le \varepsilon \\ e, & if\ |e| > \varepsilon \end{cases} \tag{8.24}$$

Where $\varepsilon > 0$ is a design constant. The selection of the dead-zone size ε clearly induces a trade-off between reducing the possibility of false alarms (robustness) and improving the sensitivity to faults. In the next section we derive a value for the dead-zone size ε (in terms of modelling uncertainty bound ξ_0) that guarantees robustness in the presence of any modelling uncertainty satisfying the given bound (Demetriou and Polycarpou, 1998).

Demetriou and Polycarpou (1998) illustrated that, the online approximation approach has certain inherent robust properties: first, since this approach is formulated in a nominal modelling framework, it allows the use of nonlinear nominal models, hence minimizing any modelling inaccuracies that would otherwise be introduced due to linearization of the system. Second, as a result of its learning capability, the on-line approximator is able to update the nominal model during operation; this fine-tuning of the nominal model may improve the accuracy between the real system and the nominal model, which, in turn leads to better performance. Finally, the ability of the on-line approximator to learn the characteristics of the off-nominal system behaviour provides a means of comparing these characteristics to any known disturbance or any known failure modes, hence discriminating between disturbances and failures.

8.5. FD for CSTR__Case Study

8.5.1. State Space Model

The following state space representation is obtained for the CSTR

$$dx1(t)/dt = F/V * (u1(t) - x1(t)) - k_0 * \exp(-E/R * x2(t)) * x1(t) \quad (8.25)$$

$$dx2(t)/dt = F/V * (u2(t) - x2(t)) - k_0 * \exp(-E/R * x2(t)) * x1(t) - (HA/(HD*V)) * (x2(t) - u3(t)) \quad (8.26)$$

Where

$$y1(t) = x1(t) \quad (8.27)$$

$$y2(t) = x2(t) \quad (8.28)$$

State equation:

$$dx = \begin{bmatrix} F/V*(u(1)-x(1)) - k_0*\exp(-E/R*x(2))*x(1) \\ F/V*(u(2)-x(2)) - k_0*\exp(-E/R*x(2))*x(1) - (HA/(HD*V))*(x(2)-u(3)) \end{bmatrix} \quad (8.29)$$

Output equation:

$$y = \begin{bmatrix} x(1) \\ x(2) \end{bmatrix} \quad (8.30)$$

Using the methodology described in previous sections, an estimated model is constructed. This estimated model is described by the following state-space representation as following:

$$\dot{\hat{x}} = \begin{bmatrix} F/V*(u(1)-x(1)) - k_0*\exp(-E/R*x(2))*x(1) \\ F/V*(u(2)-x(2)) - k_0*\exp(-E/R*x(2))*x(1) - (HA/(HD*V))*(x(2)-u(3)) \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{f}(x_2, \hat{\theta}) \end{bmatrix} - p\begin{bmatrix} \hat{x}_1 - x_1 \\ \hat{x}_2 - x_2 \end{bmatrix} \quad (8.31)$$

Where $[x_1, x_2]^T = [y_1, y_2]^T$ is the state vector of the system; $\hat{x} = [\hat{x}_1, \hat{x}_2]^T$ is the estimated state vector; $p > 0$ is the pole location of the filter; $\hat{f}$ is the online approximator model used to monitor the system; and $\hat{\theta}$ is a vector of adjustable parameters.

8.6. Simulating faults

The classes of failures considered in this work are strictly related to the dispositioning of the hot and cool water valves of the cooling system. This leads to a temperature of the fluid entering the jacket different with respect to the command value. Hence, an actuator fault result in a faulty input temperature given by

$$u^*(t)=u(t)+f_a(t) \tag{8.32}$$

Where $u(t)$ is the command value. The time profile adopted for the fault function f_a is

$$f_a(t)=u_0(1-e^{-\mu(t-t_0)}) \quad , \; t\geq t_0 \tag{8.33}$$

Where u_0 is the maximum amplitude, t_0 is the fault occurrence time, and μ is the fault evolution rate. Parameter μ is used to simulate a desired time evolution: small value characterize slowly developing faults (incipient fault); large values are used to model step-like behaviours of the fault (abrupt fault).

The reactor temperature sensor fault is superimposed with 10% change of the measured reactor temperature and simulated from time $10s\,to\,15s$.

8.7. Residual generation

In this simulation example we use a class of neural networks, known as Radial Basis Function (RBF) networks, as the on-line approximator model. Specifically, we use Gaussian RBF networks which are described by

$$\hat{f}(x_2,\hat{\theta})=\sum_{i=1}^{N}\hat{\theta}_i \exp\left(-\|x_2-c_i\|^2 \Big/ \sigma^2\right) \tag{8.34}$$

We choose a uniform width $\sigma=0.6$ for the basis functions, and $N=19$ fixed centres c_i, which are evenly distributed in the interval $[-9\;9]$.

The standard adaptive law in this case is

$$\dot{\hat{\theta}} = P\{\Gamma\, Z\ \ e\} \tag{8.35}$$

$$\dot{\hat{\theta}} = \Gamma Z e - X^{*}\,\Gamma\left(\hat{\theta}\hat{\theta}^{T} \Big/ |\hat{\theta}|^{2}\right)\Gamma Z e \tag{8.36}$$

Where

$$Z := \left[\partial \hat{f}(x,u;\hat{\theta}) \Big/ \partial\hat{\theta}\right]^{T} \quad ,\ \ \Gamma = \Gamma^{\mathrm{T}} > 0 \tag{8.37}$$

Where X^{*} denotes the indicator function given by:

$$X^{*} = \begin{cases} 0, & \text{if } (|\hat{\theta}| < M_{\hat{\theta}}) \text{ or } (|\hat{\theta}| < \overline{M}_{\hat{\theta}} \text{ and } \hat{\theta}^{T}\Gamma Z e \le 0 \\ 1, & \text{if } (|\hat{\theta}| < M_{\hat{\theta}}) \text{ and } \hat{\theta}^{T}\Gamma Z e > 0 \end{cases} \tag{8.38}$$

The OLA output norm and state error norm may be used to monitor the system for failure detection. In Figure.6 the evolution of the output state estimation error norm is represented for the time interval $[0\,,60]\,s$. The rapid jump at $t=10$ provides a measure for detecting the system's failure. The evolution of the output estimation norm is given by

$$e_{N}(t) := \left[(x1(t)-\hat{x}1(t))^{2} + (x2(t)-\hat{x}2(t))^{2}\right]^{0.5} \tag{8.39}$$

$$\phi_{N}(t) = \left|\hat{f}(x1(t),\hat{\theta}(t))\right| \tag{8.40}$$

8.8. Performance and Discussion

In this simulation we use the adaptive law given by (8.35)-(8.38), where the learning rate is chosen as $\Gamma = \gamma I$ and $\gamma = 10$. The projection operator P is used to constrain $\hat{\theta}$ within $M_{\hat{\theta}} = \left\{\hat{\theta} \in \Re^{19} : |\hat{\theta}| \le 100\right\}$. Finally, the filter pole is set to $p=1$, and the initial parameter estimation vector is chosen as $\hat{\theta}(0)=0$, which corresponds to modelling a no-failure

situation. The online approximator (OLA) output norm is used as an indicator of a system failure. It can be clearly seen from figure (8.6)-(8.10) that, the output estimation error norm $e_N(t)$ increasing after the first failure occurs at t=10s and then to converge to zero after few seconds. Similarly, for the second failure the output estimation error norm increased after the second failure occurred at $t = 30s$ and then converge to zero after few seconds. Therefore, the OLA output norm $\phi_N(t)$ provides a good measure for detecting system failure. This indicates that, the estimated model approximates well post-failure system. It is noted that the initial nonzero value of the output estimation error norm is due to a simulated difference in initial conditions between the physical system and the estimated model. This is because the OLA is trying to learn the deviation between the dynamics of the real system and the nominal model, which is nonzero as a result of modelling uncertainty

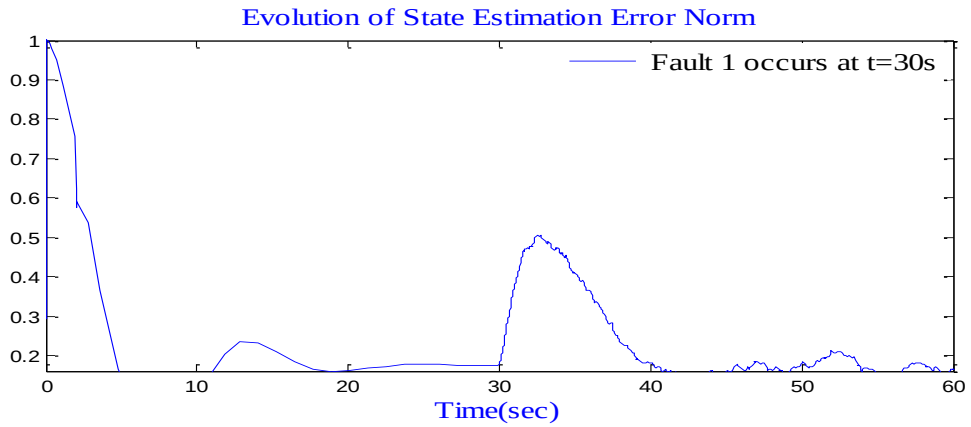


Figure 8.6 Evolution of output estimation error norm with actuator fault occurred at t=30s

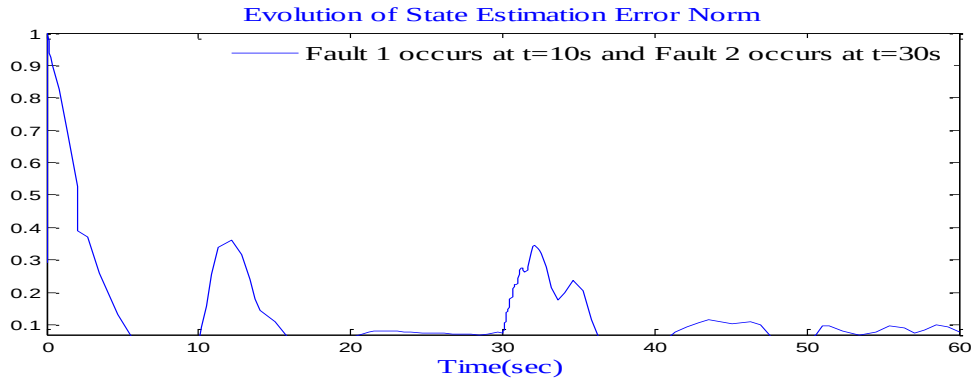


Figure 8.7 Evolution of output estimation error norm with sensor fault at t=10s and incipient actuator fault at t=30s

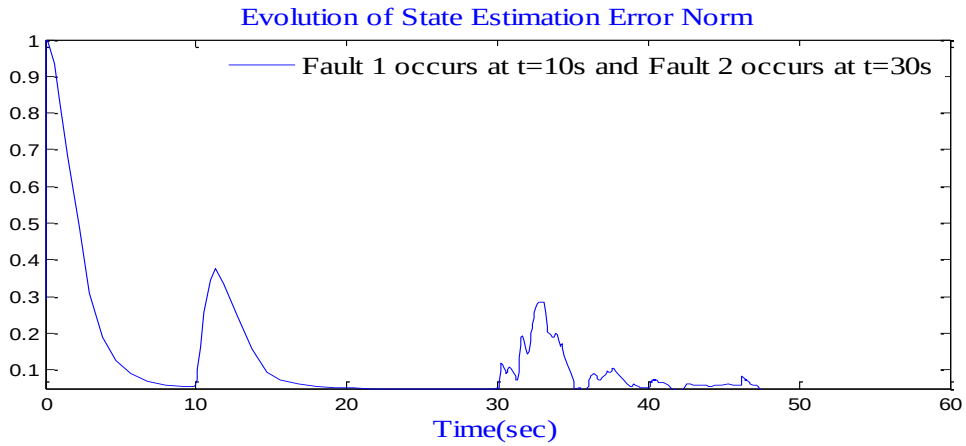


Figure 0.8 Evolution of output estimation error norm with sensor fault at t=10s and incipient actuator fault at t=30s

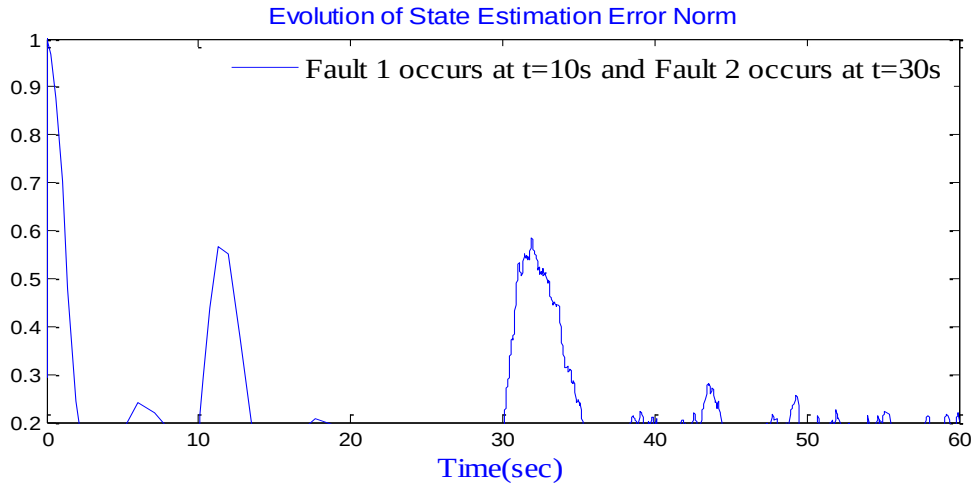


Figure 0.9 Evolution of output estimation error norm with sensor fault at t=10s and abrupt fault at t=30s

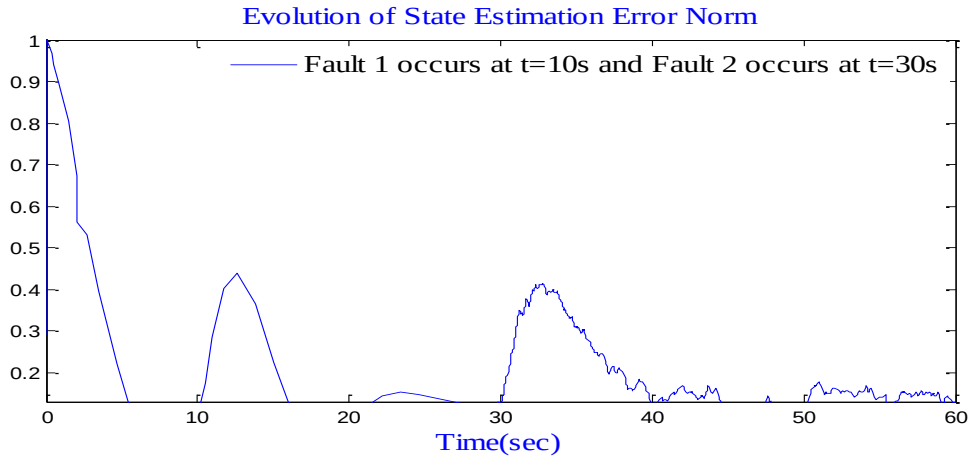


Figure 0.10 Evolution of output estimation error norm with sensor fault at t=10s and actuator fault at t=30s

8.9. Summary

In this work we have presented a general learning methodology and some preliminary analytical results concerning the use of neural networks and other on-line approximation models for diagnosis of failure in dynamical systems. Nonlinear modelling techniques have been employed for monitoring the dynamical system and for estimating any changes that may occur due to a failure. The estimated model is used for detecting failures in nonlinear systems. The main advantages of using nonlinear estimation techniques in FDA is the ability to model a larger and more practically realistic class of failure. The methodology developed in this work is based on analytical redundancy techniques. In particular, we have assumed that the nominal model provides an accurate description of the physical system in the absence of any failures. In the presence of modelling uncertainty, the FDA learning scheme may perceive this uncertainty as a change in the system dynamics, thus confusing the effect of faults and possibly leading to false alarms. Another assumption that we have made is the availability for measurement of all states. These assumptions and investigation of the effect of modelling uncertainty on the performance of the learning scheme, are main topics for future work.

Chapter 9

Conclusions and Future Work

9.1. Conclusions

This research investigates the potential use of Fault diagnosis methods for multivariable dynamic processes such as Chylla-Haase polymerization reactor and continues stirred tank reactor. In this section the main contributions and key results of this research project are summarised.

The main aim of first section of this research is focused on the understanding of the nonlinear dynamic behaviour of the Chylla-Haase polymerization reactor. In this chapter, the mathematical model of the proposed reactor is described. The material and energy balances of the reactor are illustrated in more details. All the uncertainties and disturbances in the process is discussed. Moreover, all parameter values for polymer A and B and all the empirical relations for the polymerization rate, the jacket heat transfer area, and the overall heat transfer coefficient are represented. The Simulink model of the proposed reactor is set up using Simulink/MATLAB. The design of Simulink model is developed based on a set of ordinary differential equations that describe the dynamic behaviour of the proposed polymerization process. The Simulink block diagram of the proposed reactor is presented and discussed in more details. The simulation results of open-loop Polymerization process for both polymer A and B are presented and discussed.

In the next part of this research a fault diagnosis scheme is developed for open-loop Chylla-Haase reactor using an independent RBFNN. Three sensor faults and one actuator fault are simulated on the Chylla-Haase reactor. Moreover, the uncertainties and

disturbances in the process are simulated. Two different techniques to employ RBF neural networks for fault diagnosis are investigated. The first technique is implementing an independent RBNN for residual generation. Moreover, the generated residuals were used for detecting actuator and sensor faults. The second technique is applying an additional RBFNN as a classifier to perform the classification task for residual evaluation and therefore to diagnose and isolate the faults. The simulation results show that all faults were clearly detected and isolated. Moreover, no false alarms are thereby produced, so this verifies that the proposed scheme has shown excellent diagnosis performance. The main contribution of this work is to show how to apply an independent RBFNN to open-loop Chylla-Haase benchmark polymerization reactor fault diagnosis.so this proposed method can contribute to the safety of chemical reactors.

The third part of this research is focused on the development of robust fault diagnosis scheme for a Chylla-Haase reactor under closed-loop control using an independent RBF neural network model and a RBF classifier. In the independent model, the past model output is fed back as part of the network input. Therefore, the model can operate independently from the process. Due to the increased difficulty in training an independent RBF model compared with the dependent model, the network weights were updated using the ROLS algorithm. Moreover, the disturbances are simulated and used. Consequently, the robustness of the fault detection to these disturbance is achieved. RBF classifier is implemented for fault isolation. As in practice most of systems work under closed-loop control. One of vital problems in closed-loop identification is that, the controller output cannot be designed when the reactor is under closed-loop control. Most chemical processes operate as a part of a control configuration, and the control action will correct small changes of the states caused by faults. The proposed FDI strategy is dealt with this problem.

Fault detection and isolation methods are investigated here for open-loop and closed-loop Cylla-Haase polymerization reactor using an independent mode of MLPNN. An independent MLP neural network is implemented here to generate residuals for detection task. And another RBF is applied for isolation task performing as a classifier. The simulation results confirmed that the simulated faults are clearly detected and isolated with zero false alarm rates. So this verifies that the proposed scheme has shown excellent diagnostic performance. The main contribution of this work is using an independent MLP for open-loop and mainly for closed-loop control system.

In chapter seven, a fault detection scheme based on standardized hypothesis of statistical tests generated by extended kalman filter (EKF) is developed. The proposed method is applied for online fault detection in Cylla-Haase exothermic semi-batch polymerization reactor. The simulation results show that all faults were clearly detected. Moreover, no false alarms are thereby produced, so this verifies that the proposed scheme has shown excellent detection performance. The application of using online estimation by extended kalman filter for Chylla-Haase reactor is believed to be a new contribution to industrial process.

The final part of this study focused on the development of a robust fault diagnosis scheme for abrupt and incipient faults in nonlinear dynamic system. A general framework is developed for model-based fault detection and diagnosis using on-line approximators and adaptation/learning schemes. In this framework, neural network models constitute an important class of on-line approximators. The changes in the system dynamics due to fault are modelled as nonlinear functions of the state, while the time profile of the fault is assumed to be exponentially developing. The changes in the system dynamics are monitored by an on-line approximation model, which is used for detecting the failures. A systematic procedure for constructing nonlinear estimation algorithm is developed, and a

stable learning scheme is derived using Lyapunov theory. Simulation studies are used to illustrate the results and to show the effectiveness of the fault diagnosis methodology. The main contribution of this work is to apply this method to the proposed nonlinear continuous stirred tank reactor.

The novelty and unique contribution of this research to knowledge is divided into four sections. First section will be focus on developing a new FDI method for open-loop and closed-loop reactor using an independent RBFNN, which will be a new contribution to knowledge. The second section will be focus on developing a new FDI method for open-loop and closed-loop reactor using an independent MLPNN, which will be a new contribution to knowledge. The third section is to develop a new FD method for reactor using EKF to against disturbances, which also will be a major challenge and a new contribution to knowledge. Finally, developing and designing an adaptive nonlinear observer based fault detection for reactor using a learning methodology is a new contribution to knowledge. These proposed methods are robust against the disturbances and can also cope with high nonlinearities of the reactor. The application of all proposed fault detection and isolation strategies for monitoring reactor. Thus the originality of the proposed research stands.

9.2. Recommendation for future work

In this section, some recommendations for future work will be given. These recommendations will significantly improve the performance of the developed FDI schemes. The recommended future works are (1) design fault isolation scheme for detected fault by using EKF, (2) development of adaptive nonlinear observer based fault detection using learning methodology for Chylla-Haase reactor.

- The developed fault detection method is based on the using of Extended Kalman Filter (EKF) and statistical test. Although the method presented in this thesis have

demonstrated the effectiveness of the proposed approach, it has some limitation in estimating the parameters of the reactor due to its high nonlinearity. The further improvement will be by designing The EKF to estimate on-line to the state of reactor and the overall heat transfer coefficient (U). The diagnosis method will be based on a probabilistic neural network classifier. The Inputs of the probabilistic classifier are the input-output measurements of reactor and the parameter U estimated by EKF, while the outputs of the classifier are fault types in reactor.

- In this thesis we have presented a general learning methodology and analytical results concerning the use of RBF neural networks as on-line approximation model for diagnosis of failure in dynamical systems. Although the method presented in this thesis have demonstrated the effectiveness of the proposed approach applied to CSTR reactor. It has some limitation when applied to Chylla-Haase reactor, due to the high nonlinear dynamics and unsteady operating conditions of the reactor, the application of these techniques are very challenging task to implement. Moreover, the full state measurements and an accurate knowledge of parameters of Chylla-Haase reactor are hardly available. Obtaining a nominal model of Cyhlla-Haase reactor that provides an accurate description of the physical system in the absence of any failures will be a challenging task. The future work will be designing an adaptive nonlinear observer based FD using learning methodology for Chylla-Haase reactor.
- Due to the nonlinearities present both within the plant and within the neural network, suitable stability based training rules for on-line approximator based on the RBF network. Therefore, in the future work, gradient-based adaptive control laws, will be employed, utilizing the backpropagation algorithm to determine the gradients of some suitable cost function with respect to each weight in the network.

The weight vector can then be adapted, using the gradient descent update. Hence it is necessary to develop a technique by which the error at the output of the plant could be fed back to provide a suitable descent direction at the output of the neural network. If the Jacobian of the plant is known, the gradient of the cost with respect to each input is then readily determined. In some cases, a nonlinear model of the plant may be available for analytic differentiation to provide the necessary Jacobian as the output of a sensitivity model. Differentiation of these equations yields the sensitivity model which can be executed in parallel with the nonlinear system to provide a continuous estimate of the Jacobian, required for on-line approximators. Differentiation of these equations will be one of the future work.

- After designing a detection and approximation estimator for online monitoring. Once a fault is detected, a bank of isolation estimators will be activated for the purpose of fault isolation. A key design issue of the proposed fault isolation scheme is the adaptive residual threshold associated with each isolation estimator. A fault that has occurred can be isolated if the residual associated with the matched isolation estimator remains below its corresponding adaptive threshold, whereas at least one of the components of the residuals associated with all the other estimators exceeds its threshold at some finite time. A bank of nonlinear adaptive estimators are used in the proposed FDI scheme, One of the nonlinear adaptive estimators is the fault detection and approximation estimator (FDAE) used to detect faults. The remaining ones are fault isolation estimators (FIEs) that are used for isolation purposes only after a fault has been detected. Each FIE corresponds to a particular type of fault.

References

AKHOONDZADEH, M. 2013. A MLP neural network as an investigator of TEC time series to detect seismo-ionospheric anomalies. *Advances in Space Research,* 51**,** 2048-2057.

BALLESTEROS-MONCADA, H., HERRERA-LÓPEZ, E. J. & ANZUREZ-MARÍN, J. 2015. Fuzzy model-based observers for fault detection in CSTR. *ISA Transactions,* 59**,** 325-333.

BARTON, R. S. & HIMMELBLAU, D. M. Online prediction of polymer product quality in an industrial reactor using recurrent neural networks. Neural Networks,1997., International Conference on, 9-12 Jun 1997 1997. 111-114 vol.1.

BENKOUIDER, A. M., BUVAT, J. C., COSMAO, J. M. & SABONI, A. 2009. Fault detection in semi-batch reactor using the EKF and statistical method. *Journal of Loss Prevention in the Process Industries,* 22**,** 153-161.

BENKOUIDER, A. M., KESSAS, R., YAHIAOUI, A., BUVAT, J. C. & GUELLA, S. A hybrid approach to faults detection and diagnosis in batch and semi-batch reactors by using EKF and neural network classifier. *Journal of Loss Prevention in the Process Industries*.

BEYER, M.-A., GROTE, W. & REINIG, G. 2008. Adaptive exact linearization control of batch polymerization reactors using a Sigma-Point Kalman Filter. *Journal of Process Control,* 18**,** 663-675.

BONVIN, D. 1998. ADCHEM '97 IFAC Symposium: Advanced Control of Chemical ProcessesOptimal operation of batch reactors—a personal view. *Journal of Process Control,* 8**,** 355-368.

CACCAVALE, F., PIERRI, F., IAMARINO, M. & TUFANO, V. 2009. An integrated approach to fault diagnosis for a class of chemical batch processes. *Journal of Process Control,* 19**,** 827-841.

CHEN, S., BILLINGS, S. A., COWAN, C. F. N. & GRANT, P. M. 1990. Non-linear systems identification using radial basis functions. *International Journal of Systems Science,* 21**,** 2513-2539.

CHEN, T., MORRIS, J. & MARTIN, E. 2007. Particle Filters for Dynamic Data Rectification and Process Change Detection A2 - Zhang, Hong-Yue. *Fault Detection, Supervision and Safety of Technical Processes 2006.* Oxford: Elsevier Science Ltd.

CHETOUANI, Y. 2004. Fault detection by using the innovation signal: application to an exothermic reaction. *Chemical Engineering and Processing: Process Intensification,* 43**,** 1579-1585.

CHYLLA, R. W. & HAASE, D. R. 1993. Temperature control of semibatch polymerization reactors. *Computers & Chemical Engineering,* 17**,** 257-264.

CLARK, J. Y. & WARWICK, K. Detection of faults in a high speed packaging machine using a multilayer perceptron (MLP). Innovations in Manufacturing Control Through Mechatronics, IEE Colloquium on, 22 Nov 1995 1995. 7/1-7/3.

CLARKE-PRINGLE, T. & MACGREGOR, J. F. 1997. Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors. *Computers & Chemical Engineering,* 21**,** 1395-1409.

CORRADINI, M. L., CRISTOFARO, A. & PETTINARI, S. 2012. Design of robust fault detection filters for linear descriptor systems using sliding-mode observers. *IFAC Proceedings Volumes,* 45**,** 778-783.

DALLE MOLLE, D. T. & HIMMELBLAU, D. M. 1987. Fault detection in a single-stage evaporator via parameter estimation using the Kalman filter. *Industrial & Engineering Chemistry Research,* 26**,** 2482-2489.

DASH, R. N., SUBUDHI, B. & DAS, S. A comparison between MLP NN and RBF NN techniques for the detection of stator inter-turn fault of an induction motor. Industrial Electronics, Control & Robotics (IECR), 2010 International Conference on, 27-29 Dec. 2010 2010. 251-256.

DAVID, E. R. & JAMES, L. M. 1987. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing:Explorations in the Microstructure of Cognition: Foundations.* MIT Press.

DE LIRA, S., PUIG, V., QUEVEDO, J. & HUSAR, A. 2011. LPV observer design for PEM fuel cell system: Application to fault detection. *Journal of Power Sources,* 196**,** 4298-4305.

DE VALLIE`RE, P. & BONVIN, D. 1989. Application of estimation techniques to batch reactors—II. Experimental studies in state and parameter estimation. *Computers & Chemical Engineering,* 13**,** 11-20.

DEIBERT, R. & ISERMANN, R. 1992. Examples for fault detection in closed loops. *Annual Review in Automatic Programming,* 17**,** 235-240.

DEMETRIOU, M. A. & POLYCARPOU, M. M. 1998. Incipient fault diagnosis of dynamical systems using online approximators. *Automatic Control, IEEE Transactions on,* 43**,** 1612-1617.

ERTIAME, A. M., DINGLI, Y., FENG, Y. & GOMM, J. B. Fault detection and isolation for open-loop Chylla-Haase polymerization reactor. Automation and Computing (ICAC), 2013 19th International Conference on, 13-14 Sept. 2013 2013. 1-6.

ERTIAME, A. M., YU, D., YU, F. & GOMM, J. B. 2015. Robust fault diagnosis for an exothermic semi-batch polymerization reactor under open-loop. *Systems Science & Control Engineering,* 3**,** 14-23.

ERTIAME, A. Y., D. ; YU, D. ; GOMM, J. 2015. Dynamic Fault Diagnosis for Semi-Batch Reactor under Closed-Loop Control via Independent Radial Basis Function Neural Network. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering,* 9**,** 1208-1220.

FABRIZIO CACCAVALE, MARIO IAMARINO, FRANCESCO PIERRI & TUFANO, V. 2011. *Control and Monitoring of Chemical Batch Reactors,* London, Springer-Verlag London Limited.

FABRIZIO CACCAVALE, M. L., FRANCESCO PIERRI,VINCENZO TUFANO 2011. *Control and Monitoring of Chemical Batch Reactors,* London, Springer-Verlag London Limited.

FERRARI, R. M. G., PARISINI, T. & POLYCARPOU, M. M. A robust fault detection and isolation scheme for a class of uncertain input-output discrete-time nonlinear systems. American Control Conference, 2008, 11-13 June 2008 2008. 2804-2809.

FRANK, P. M. 1990. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. *Automatica,* 26**,** 459-474.

FRANK, P. M. 1996. Analytical and Qualitative Model-based Fault Diagnosis – A Survey and Some New Results. *European Journal of Control,* 2**,** 6-28.

FRANK, P. M. & DING, X. 1997. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control,* 7**,** 403-424.

FRANK, P. M. & KÖPPEN-SELIGER, B. 1997. Fuzzy logic and neural network applications to fault diagnosis. *International Journal of Approximate Reasoning,* 16**,** 67-88.

FU, K., ZHANG, D., TANG, P., TANG, Z. & HE, W. Adaptive extended Kalman filter for a red shift navigation system. Control Conference (CCC), 2015 34th Chinese, 28-30 July 2015 2015. 5194-5199.

GERTLER, J. 1997. Fault detection and isolation using parity relations. *Control Engineering Practice,* 5**,** 653-661.

GERTLER, J. J. 1988. Survey of model-based failure detection and isolation in complex plants. *Control Systems Magazine, IEEE,* 8**,** 3-11.

GOLOVKO, V., SAVITSKY, Y. & MANIAKOV, N. Modeling nonlinear dynamics using multilayer neural networks. Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, International Workshop on, 2001., 2001 2001. 197-202.

GOMM, J. B., WILLIAMS, D., EVANS, J. T., DOHERTY, S. K. & LISBOA, P. J. G. 1996. Enhancing the non-linear modelling capabilities of MLP neural networks using spread encoding. *Fuzzy Sets and Systems,* 79**,** 113-126.

GOMM, J. B. & YU, D. L. 2000. Selecting radial basis function network centers with recursive orthogonal least squares training. *Neural Networks, IEEE Transactions on,* 11**,** 306-314.

GRAICHEN, K., HAGENMEYER, V. & ZEITZ, M. Adaptive Feedforward Control with Parameter Estimation for the Chylla-Haase Polymerization Reactor. Proceedings of the 44th IEEE Conference on Decision and Control, 12-15 Dec. 2005 2005a. 3049-3054.

GRAICHEN, K., HAGENMEYER, V. & ZEITZ, M. Adaptive Feedforward Control with Parameter Estimation for the Chylla-Haase Polymerization Reactor. Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on, 12-15 Dec. 2005 2005b. 3049-3054.

GRAICHEN, K., HAGENMEYER, V. & ZEITZ, M. 2006. Feedforward control with online parameter estimation applied to the Chylla–Haase reactor benchmark. *Journal of Process Control,* 16**,** 733-745.

HATAMI, E., SALARIEH, H. & VOSOUGHI, N. 2014. Design of a fault tolerated intelligent control system for a nuclear reactor power control: Using extended Kalman filter. *Journal of Process Control,* 24**,** 1076-1084.

HE, H., LIU, Z. & HUA, Y. 2015. Adaptive Extended Kalman Filter Based Fault Detection and Isolation for a Lithium-Ion Battery Pack. *Energy Procedia,* 75**,** 1950-1955.

IFTIKHAR, K., KHAN, A. Q. & ABID, M. 2015. Optimal fault detection filter design for switched linear systems. *Nonlinear Analysis: Hybrid Systems,* 15**,** 132-144.

ISERMANN, R. 1984. Process fault detection based on modeling and estimation methods—A survey. *Automatica,* 20**,** 387-404.

ISERMANN, R. 1993. Fault diagnosis of machines via parameter estimation and knowledge processing—Tutorial paper. *Automatica,* 29**,** 815-835.

ISERMANN, R. 1997. Supervision, fault-detection and fault-diagnosis methods — An introduction. *Control Engineering Practice,* 5**,** 639-652.

ISERMANN, R. 2005. Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control,* 29**,** 71-85.

JASSEMI-ZARGANI, R. & NECSULESCU, D. 2002. Extended Kalman filter-based sensor fusion for operational space control of a robot arm. *IEEE Transactions on Instrumentation and Measurement,* 51**,** 1279-1282.

JOHNSON, C., VENAYAGAMOORTHY, G. K. & MITRA, P. 2009. Comparison of a spiking neural network and an MLP for robust identification of generator dynamics in a multimachine power system. *Neural Networks,* 22**,** 833-841.

JUNG-WOOK, P., HARLEY, R. G. & VENAYAGAMOORTHY, G. K. Comparison of MLP and RBF neural networks using deviation signals for on-line identification of a synchronous generator. Power Engineering Society Winter Meeting, 2002. IEEE, 2002 2002. 274-279 vol.1.

KABBAJ, N., DONCESCU, A. & AGUILAR-MARTIN, J. 2009. Parity Relations based on elimination theory for Fault Detection in a bioprocess. *IFAC Proceedings Volumes,* 42**,** 1270-1275.

KAI, X., WEI, C. & LIU, L. 2010. Robust Extended Kalman Filtering for Nonlinear Systems With Stochastic Uncertainties. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* 40**,** 399-405.

KELIRIS, C., POLYCARPOU, M. M. & PARISINI, T. 2015. A robust nonlinear observer-based approach for distributed fault detection of input–output interconnected systems. *Automatica,* 53**,** 408-415.

KHANESAR, M. A., KAYACAN, E., TESHNEHLAB, M. & KAYNAK, O. 2012. Extended Kalman Filter Based Learning Algorithm for Type-2 Fuzzy Logic Systems and Its Experimental Evaluation. *IEEE Transactions on Industrial Electronics,* 59**,** 4443-4455.

KOPPEN-SELIGER, B. & FRANK, P. M. Fault detection and isolation in technical processes with neural networks. Decision and Control, 1995., Proceedings of the 34th IEEE Conference on, 13-15 Dec 1995 1995. 2414-2419 vol.3.

LI, R. & OLSON, J. H. 1991. Fault detection and diagnosis in a closed-loop nonlinear distillation process: application of extended Kalman filters. *Industrial & Engineering Chemistry Research,* 30**,** 898-908.

LIPPMANN, R. 1987. An introduction to computing with neural nets. *IEEE ASSP Magazine,* 4**,** 4-22.

LIU, W. 1999. An extended Kalman filter and neural network cascade fault diagnosis strategy for the glutamic acid fermentation process. *Artificial Intelligence in Engineering,* 13**,** 131-140.

LORON, L. & LALIBERTE, G. Application of the extended Kalman filter to parameters estimation of induction motors. Power Electronics and Applications, 1993., Fifth European Conference on, 13-16 Sep 1993 1993. 85-90 vol.5.

MAHMUD, M. N., IBRAHIM, M. N., OSMAN, M. K. & HUSSAIN, Z. Comparison of MLP network training algorithms for fault classification in transmission lines. Control System, Computing and Engineering (ICCSCE), 2014 IEEE International Conference on, 28-30 Nov. 2014 2014. 208-213.

MAKI, Y. & LOPARO, K. A. 1997. A neural-network approach to fault detection and diagnosis in industrial processes. *Control Systems Technology, IEEE Transactions on,* 5**,** 529-541.

MEDVEDEV, A. 1995. Fault detection and isolation by a continuous parity space method. *Automatica,* 31**,** 1039-1044.

MEHRA, R. K. & PESCHON, J. 1971. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica,* 7**,** 637-640.

MENAA, M., TOUHAMI, O. & IBTIOUEN, R. Estimation of the rotor resistance in induction motor by application of the spiral vector theory associate to extended Kalman filter. System Theory, 2003. Proceedings of the 35th Southeastern Symposium on, 16-18 March 2003 2003. 211-216.

MRUGALSKI, M. & KORBICZ, J. 2007. Application of the MLP Neural Network to the Robust Fault Detection A2 - Zhang, Hong-Yue. *Fault Detection, Supervision and Safety of Technical Processes 2006.* Oxford: Elsevier Science Ltd.

NAIK, A. S., YIN, S., DING, S. X. & JEINSCH, T. 2009. Recursive identification algorithm for parity space based fault detection systems. *IFAC Proceedings Volumes,* 42**,** 312-317.

ODENDAAL, H. M. & JONES, T. 2014. Actuator fault detection and isolation: An optimised parity space approach. *Control Engineering Practice,* 26**,** 222-232.

OUHROUCHE, M. A., LEFEBVRE, S. & DO, X. D. Application of an extended Kalman filter to rotor speed and resistance estimation in induction motor vector control. Electrical and Computer Engineering, 1998. IEEE Canadian Conference on, 24-28 May 1998 1998. 297-300 vol.1.

PANDEY, P. C. & BARAI, S. V. 1995. Multilayer perceptron in damage detection of bridge structures. *Computers & Structures,* 54**,** 597-608.

PARLOS, A. G., CHONG, K. T. & ATIYA, A. F. 1994. Application of the recurrent multilayer perceptron in modeling complex process dynamics. *IEEE Transactions on Neural Networks,* 5**,** 255-266.

PATTON, R. J. 1997. Robustness in model-based fault diagnosis: The 1995 situation. *Annual Reviews in Control,* 21**,** 103-123.

PATTON, R. J. & CHEN, J. 1992a. On-line residual compensation in robust fault diagnosis of dynamic systems. *Annual Review in Automatic Programming,* 17**,** 221-227.

PATTON, R. J. & CHEN, J. Robustness in quantitative model-based fault diagnosis. Intelligent Fault Diagnosis - Part 2: Model-Based Techniques, IEE Colloquium on, 26 Feb 1992 1992b. 4/1-417.

PATTON, R. J. & CHEN, J. 1997. Observer-based fault detection and isolation: Robustness and applications. *Control Engineering Practice,* 5**,** 671-682.

PATTON, R. J., CHEN, J. & SIEW, T. M. Fault diagnosis in nonlinear dynamic systems via neural networks. Control, 1994. Control '94. International Conference on, 21-24 March 1994 1994. 1346-1351 vol.2.

PIERRI, F. & PAVIGLIANITI, G. Observer-based actuator fault detection for chemical batch reactors: A comparison between nonlinear adaptive and H∞- based approaches. Control & Automation, 2007. MED '07. Mediterranean Conference on, 27-29 June 2007 2007. 1-6.

PIERRI, F., PAVIGLIANITI, G., CACCAVALE, F. & MATTEI, M. 2008. Observer-based sensor fault detection and isolation for chemical batch reactors. *Engineering Applications of Artificial Intelligence,* 21**,** 1204-1216.

POLYCARPOU, M. M. & HELMICKI, A. J. 1995. Automated fault detection and accommodation: a learning systems approach. *Systems, Man and Cybernetics, IEEE Transactions on,* 25**,** 1447-1458.

POLYCARPOU, M. M. & VEMURI, A. T. 1995. Learning methodology for failure detection and accommodation. *Control Systems, IEEE,* 15**,** 16-24.

SALAHSHOOR, K. & MOSALLAEI, M. 2008. Process Fault Monitoring Using Data Fusion Based on Extended Kalman Filter Incorporated with Time-Delayed Measurements. *IFAC Proceedings Volumes,* 41**,** 8896-8901.

SENJYU, T., KINJO, K., URASAKI, N. & UEZATO, K. 2003. High efficiency control of synchronous reluctance motors using extended Kalman filter. *IEEE Transactions on Industrial Electronics,* 50**,** 726-732.

SOUAHLIA, S., BACHA, K. & CHAARI, A. 2012. MLP neural network-based decision for power transformers fault diagnosis using an improved combination of Rogers and Doernenburg ratios DGA. *International Journal of Electrical Power & Energy Systems,* 43**,** 1346-1353.

SRINIVASAN, B., PALANKI, S. & BONVIN, D. 2003. Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Computers & Chemical Engineering,* 27**,** 1-26.

TRUNOV, A. B. & POLYCARPOU, M. M. 2000. Automated fault diagnosis in nonlinear multivariable systems using a learning methodology. *Neural Networks, IEEE Transactions on,* 11**,** 91-101.

VEMURI, A. T. & POLYCARPOU, M. M. Input-output systems robust nonlinear fault diagnosis. American Control Conference, 1997. Proceedings of the 1997, 4-6 Jun 1997 1997. 927-931 vol.2.

VENKATASUBRAMANIAN, V., RENGASWAMY, R. & KAVURI, S. N. 2003a. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Computers & Chemical Engineering,* 27**,** 313-326.

VENKATASUBRAMANIAN, V., RENGASWAMY, R., KAVURI, S. N. & YIN, K. 2003b. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering,* 27**,** 327-346.

VENKATASUBRAMANIAN, V., RENGASWAMY, R., KAVURI, S. N. & YIN, K. 2003c. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering,* 27**,** 327-346.

VENKATASUBRAMANIAN, V., RENGASWAMY, R., YIN, K. & KAVURI, S. N. 2003d. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering,* 27**,** 293-311.

WALKER, B. K. & HUANG, K.-Y. 1995. FDI by extended Kalman filter parameter estimation for an industrial actuator benchmark. *Control Engineering Practice,* 3**,** 1769-1774.

WANG, S., JIANG, Y., LI, Y. & LIU, D. 2015. Reliable observer-based H∞ control for discrete-time fuzzy systems with time-varying delays and stochastic actuator faults via scaled small gain theorem. *Neurocomputing,* 147**,** 251-259.

WANG, S. W., YU, D. L., GOMM, J. B., PAGE, G. F. & DOUGLAS, S. S. 2006. Adaptive neural network model based predictive control for air–fuel ratio of SI engines. *Engineering Applications of Artificial Intelligence,* 19**,** 189-200.

WANG, Z. & SHANG, H. 2015. Kalman filter based fault detection for two-dimensional systems. *Journal of Process Control,* 28**,** 83-94.

WEI, Z. & YANG, S. X. Neural network based extended Kalman filter for localization of mobile robots. Intelligent Control and Automation (WCICA), 2011 9th World Congress on, 21-25 June 2011 2011. 937-942.

WEN, Y., POZNYAK, A. S. & SANCHEZ, E. N. Dynamic multilayer neural networks for nonlinear system on-line identification. Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on, 2000 2000. 25-30.

WILLSKY, A. S. 1976. A survey of design methods for failure detection in dynamic systems. *Automatica,* 12**,** 601-611.

XIAODONG, Z. 2011. Sensor Bias Fault Detection and Isolation in a Class of Nonlinear Uncertain Systems Using Adaptive Estimation. *Automatic Control, IEEE Transactions on,* 56**,** 1220-1226.

XIAODONG, Z., POLYCARPOU, M. M. & PARISINI, T. 2002. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *Automatic Control, IEEE Transactions on,* 47**,** 576-593.

YOUNG-MOON, P., MYEON-SONG, C. & LEE, K. Y. 1996. An optimal tracking neuro-controller for nonlinear dynamic systems. *IEEE Transactions on Neural Networks,* 7**,** 1099-1110.

YU CHANG, T., CHEN, K. S., WEN-LIANG, K. & FUNG, A. K. 1994. A dynamic learning neural network for remote sensing applications. *IEEE Transactions on Geoscience and Remote Sensing,* 32**,** 1096-1102.

YU, D. L., GOMM, J. B. & WILLIAMS, D. 1999. Sensor fault diagnosis in a chemical process via RBF neural networks. *Control Engineering Practice,* 7**,** 49-55.

ZAREI, J. & SHOKRI, E. 2014. Robust sensor fault detection based on nonlinear unknown input observer. *Measurement,* 48**,** 355-367.

ZHAI, Y. & MA, S. Controller design for continuous stirred tank reactor based on feedback linearization. Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on, 15-17 July 2012 2012. 378-381.

ZHANG, J., SHI, P. & LIN, W. 2016. Extended sliding mode observer based control for Markovian jump linear systems with disturbances. *Automatica,* 70**,** 140-147.

ZHANG, P., YE, H., DING, S. X., WANG, G. Z. & ZHOU, D. H. 2006. On the relationship between parity space and approaches to fault detection. *Systems & Control Letters,* 55**,** 94-100.

ZHANG, X., POLYCARPOU, M. M. & PARISINI, T. 2010. Fault diagnosis of a class of nonlinear uncertain systems with Lipschitz nonlinearities using adaptive estimation. *Automatica,* 46**,** 290-299.

ZHONG, M., SONG, Y. & DING, S. X. 2015. Parity space-based fault detection for linear discrete time-varying systems with unknown input. *Automatica,* 59**,** 120-126.

ZHOU, Y., HAHN, J. & MANNAN, M. S. 2003. Fault detection and classification in chemical processes based on neural networks with feature extraction. *ISA Transactions,* 42**,** 651-664.

ZHU, F. & CEN, F. 2010. Full-order observer-based actuator fault detection and reduced-order observer-based fault reconstruction for a class of uncertain nonlinear systems. *Journal of Process Control,* 20**,** 1141-1149.

ZHUANG, G., XIA, J., CHU, Y. & CHEN, F. 2014. H∞ mode-dependent fault detection filter design for stochastic Markovian jump systems with time-varying delays and parameter uncertainties. *ISA Transactions,* 53**,** 1024-1034.

Appendix A: Author Publication List

Ertiame, A. M., Dingli,Yu., Feng,Yu., J.B.Gomm, . (2013). Fault detection and isolation for open-loop Chylla-Haase polymerization reactor. Automation and Computing (ICAC), 2013 19th International Conference on.

Ertiame, A. M., Dingli,Yu., Feng,Yu., J.B.Gomm, (2015). "Robust fault diagnosis for an exothermic semi-batch polymerization reactor under open-loop." Systems Science & Control Engineering 3(1): 14-23.

Ertiame, A. Y., D.; Yu, D.; Gomm, J. (2015). "Dynamic Fault Diagnosis for Semi-Batch Reactor under Closed-Loop Control via Independent Radial Basis Function Neural Network." International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering 9(12): 1208-1220.