International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:12, 2015

Dynamic Fault Diagnosis for Semi-Batch Reactor under Closed-Loop Control via Independent Radial Basis Function Neural Network

Abdelkarim M. Ertiame, D. W. Yu, D. L. Yu, J. B. Gomm

***Abstract*—**In this paper, a 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.

***Keywords*—**Robust fault detection, cascade control, independent RBF model, RBF neural networks, Chylla-Haase reactor, FDI under closed-loop control.

I. Introduction

IN recent years, the task of monitoring complex nonlinear processes has been intensively studied. Fault detection and isolation techniques have attracted much interest due to the increasing demand for good performance and higher standards of safety and reliability of technical plants [16], [21]. FDI has become a critical issue in the operation of high-performance chemical plants, nuclear plants, airplanes, ships, submarines, and space vehicles, etc. [5], [11], [21]. 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 [11], [21]. However, early detection of faults can help avoid all these major consequences in [5], [8], [9], [22] authors illustrated that, FD system must avoid two kinds of errors, false alarms and missed alarms.

Abdelkarim.M.Ertiame, Ding-Li Yu are with the Control Research Group, Liverpool John Moores University, Liverpool, UK (e-mail: Abdelkarim.M.Ertiame@2011.ljmu.ac.uk, D.Yu@ljmu.ac.uk).

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 [9], [19], [20], [23], [25].

The application of neural networks (NN) for FDI has been intensively studied over the last two decades [13], [16], [17], [21], [24]. In [18], FDI using a multi-layer perceptron (MLP) network is proposed for open-loop nonlinear dynamic process. Another method was studied in [1] used recurrent neural network (RNN) to model the process dynamics. In [26] sensor fault diagnosis is proposed for open-loop chemical process via dependent RBF neural networks. In [10] researchers studied fuzzy logic and neural network applications for fault diagnosis, their paper introduced a dependent neural network for residual generation and fuzzy logic for residual evaluation. In [6], [7], FDI using an independent RBFNN is proposed for open-loop chylla-haase reactor.

In practice the semi-batch polymerization reactor works under closed-loop. FDI for the proposed reactor is challenging due to its high nonlinear nature. The reason is the process outputs in closed loop system will be fed back and this will affect the sensor faults. The novelty of this work lies in using an independent RBFNN to model Chylla-Haase polymerization reactor that is under cascade PI control. Firstly, an independent RBFNN is employed to predict the process output on-line. The K-means clustering algorithm is used to choose the centers for the RBF networks, and a P-nearest-neighbors algorithm is used to choose the widths. Moreover, a recursive orthogonal least squares (ROLS) algorithm is used to train the weights of the independent RBFNN. Then, a second neural network is used as a classifier to isolate these faults. Several actuator and sensor faults are simulated to the Chylla-Haase benchmark model. The detection and isolation of these faults using the developed scheme was demonstrated by Matlab/Simulink simulation, and the results indicate the effectiveness of the method and the feasibility of applying to the practical chemical plants. The paper is organized as follows: In Section II, process description of Chylla-Haase and the dynamic model is presented. Section III presents modelling of the system dynamics using RBF network. Fault detection scheme is given in Section IV. Section V present fault isolation scheme, finally conclusion is discussed in Section VI.

II. The Chylla-Haase Polymerization Reactor and the Closed-Loop Control

In the chemical industry, the most commonly used reactors are batch and semi-batch reactors. In this research, a semi-batch polymerization reactor is considered which is described in [4] and used as a benchmark for process control applications. The schematic diagram of the semi-batch polymerization reactor is shown in Fig. 1. 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 by 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 injected, and when the jacket controller output is between 50% and 100%, the valve is opened and steam is inserted [2], [4], [14], [15].

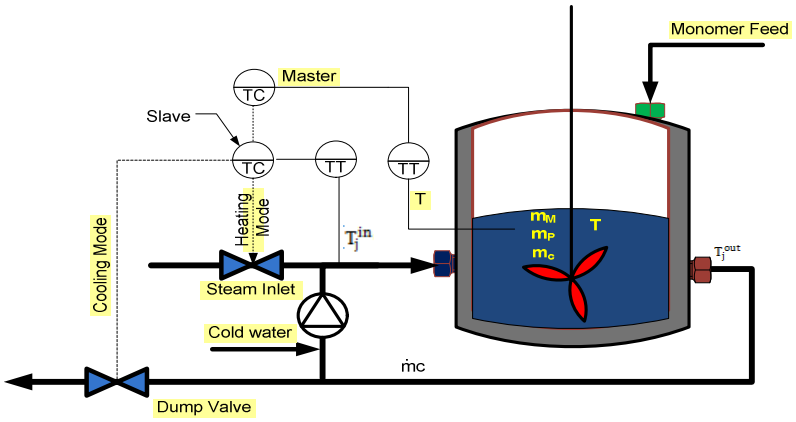


Fig. 1 Chylla-Haase reactor schematic

A. Chylla-Haase Reactor Dynamic Model

The mathematical model of the Chylla-Haase reactor is described by a set of five ordinary differential equations which come from material and heat balances inside the reactor:

$$\frac{dm_M}{dt} = \dot{m}_M^{in}(t) + \frac{Q_{rea}}{\Delta H} \tag{1}$$

$$\frac{dm_P}{dt} = -\frac{Q_{rea}}{\Delta H} \tag{2}$$

$$\frac{dT}{dt} = \frac{1}{\sum_i m_i C_{P,i}}[\dot{m}_M^{in}(t) C_{P,M}(T_{amb} - T) - UA(T - T_j) - (UA)_{loss}(T - T_{amb}) + Q_{rea}] \tag{3}$$

$$\frac{dT_{jout}}{dt} = \frac{1}{m_C C_{p,C}}[\dot{m}_C C_{P,C}(T_{jin}(t-\theta_1) - T_{jout}) + UA(T - T_j)] \tag{4}$$

$$\frac{dT_{jin}}{dt} = \frac{dT_{jout}(t-\theta_2)}{dt} + \frac{T_{jout}(t-\theta_2) - T_{jin}}{\tau_P} + \frac{K_P(c)}{\tau_P} \tag{5}$$

The reactor model includes the material balances (1) and (2) for the monomer mass $m_M(t)$ and the polymer mass $m_P(t)$, the energy balance (3) with the reactor temperature $T(t)$, plus the energy balances (4) and (5) 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 [15]:

$$y = [T, T_{jin}, T_{jout}]^T \tag{6}$$

The heating/cooling function $K_P(c)$ is influenced by an equal-percentage valve with valve position $c(t)$ as shown in (7):

$$K_P(c) = \begin{cases} 0.8\times 30^{-c/50\%}(T_{inlet} - T_{jin}(t)), & c<50\% \\ 0, & c=50\% \\ 0.15\times 30^{(c/50\%-2)}(T_{steam} - T_{jin}(t)), & c>50\% \end{cases} \tag{7}$$

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. Moreover, the variables and the parameters of the reactor model are listed in Table I. [15].

B. 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 ±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}.
- Measurement noise is added to the temperature measurements (6) with the standard deviation $\sigma(y)=0.05K$.

Table II describes the empirical relations for the polymerization rate, the jacket heat transfer area, and the overall heat transfer coefficient [15]. See [15] for the parameters values of the model and the polymers.

TABLE I
VARIABLES AND PARAMETERS OF THE REACTOR MODEL

Symbol	Quantity
$\dot{m}_M^{in}(t)$	Monomer feed rate $[kg/s]$
$Q_{rea}=-\Delta H.R_P$	Reaction heat $[kW]$
R_P	Rate of polymerization $[kg/s]$
$-\Delta H$	Reaction enthalpy $[kJ\,kg^{-1}]$
U	Overall heat transfer coefficient $[kWm^{-2}K^{-1}]$
A	Jacket heat transfer area $[m^2]$
$(UA)_{loss}$	Heat loss coefficient $[kW/K]$
$C_{p,M}, C_{p,P}, C_{p,C}$	Specific heat at constant pressure $[kJ\,kmol^{-1}K^{-1}]$
θ_1, θ_2	Transport delay $[s]$
$T_j=(T_{jin}+T_{jout})/2$	Average cooling jacket temperature $[K]$
$K_p(c)$	Heating/cooling function $[K]$
τ_p	Heating/cooling time constant $[s]$

C. Closed-Loop Control System Design and Performance

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 Fig. 2. 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.

TABLE I
EMPIRICAL RELATIONS, THE JACKET HEAT TRANSFER ARE, AND THE OVERALL HEAT TRANSFER COEFFICIENT

i	Impurity factor $[-]$
$k=k_0\exp(-E/RT).(k_1\mu)^{k_2}$	kinetic constant $[s^{-1}]$
$\mu=c_0\exp(c_1 f).10^{c_2(a_0/T-c_3)}$	Batch viscosity $[kg\,m^{-1}s^{-1}]$
$f=m_P/(m_M+m_P+m_C)$	Auxiliary variable $[-]$
$k_0, k_1, E, R, a_0, c_0, c_1, c_2, c_3$	Constants
R	Natural Gas $[kJ\,kmol^{-1}K^{-1}]$
$A=(\frac{m_M}{\rho_M}+\frac{m_P}{\rho_P}+\frac{m_W}{\rho_W})\frac{P}{B_1}+B_2$	Jacket heat transfer area $[m^2]$
ρ_M, ρ_P, ρ_W	Densities $[kg\,m^{-3}]$
B_1	Reactor bottom area $[m]$
P	Jacket perimeter $[m]$
B_2	Jacket bottom area $[m^2]$
$U=1/(h^{-1}+h_f^{-1})$	Heat transfer coefficient $[kWm^{-2}K^{-1}]$
$h=d_0\exp(d_1\mu_{wall})$	Film Heat transfer $[kWm^{-2}K^{-1}]$
$\mu_{wall}=c_0\exp(c_1 f).10^{c_2(a_0/T_{wall}-c_3)}$	Wall viscosity $[kg\,m^{-1}s^{-1}]$
$T_{wall}=(T+T_j)/2$	Wall temperature $[K]$
h_f^{-1}	Fouling factor $[m^2K/kW]$
d_0, d_1	Constants

The parameters of the conventional cascade PI controllers have been tuned in simulation studies a $K_P=21$ s, $K_I=0.08$ 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 $4s$. Fig. 3 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 ± 0.6 K 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.

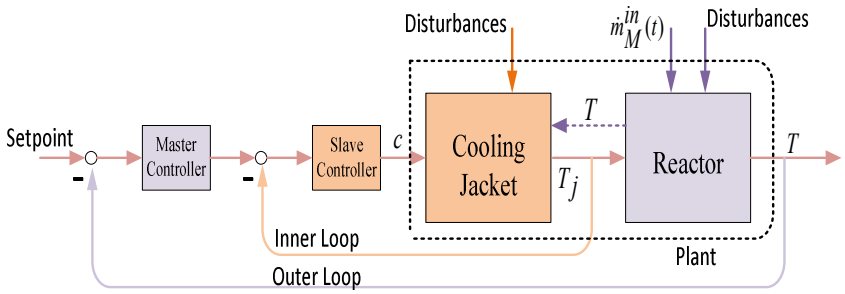
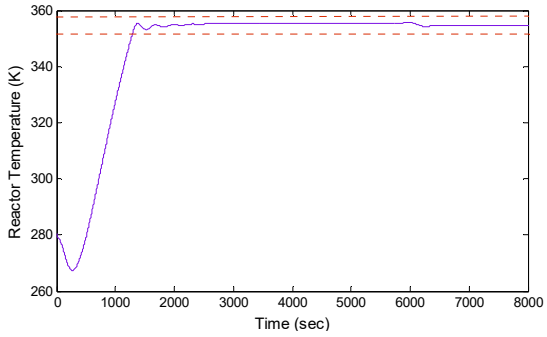
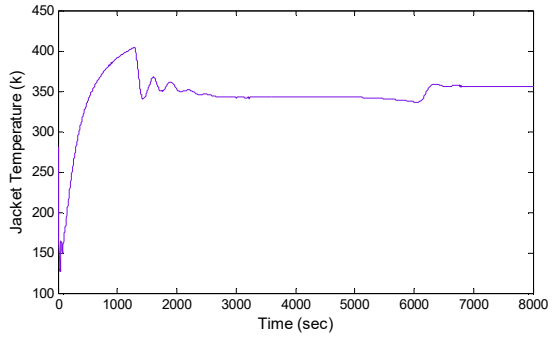


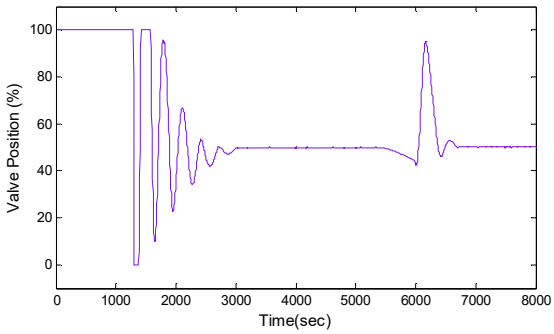
Fig. 2 Block diagram of the cascade control scheme



(a)

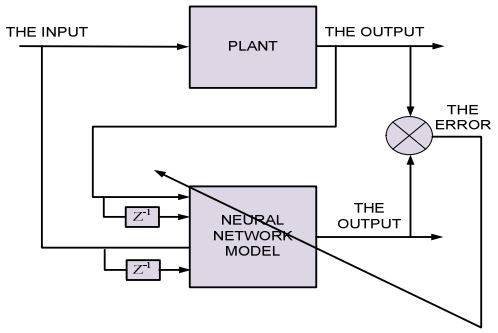


(b)

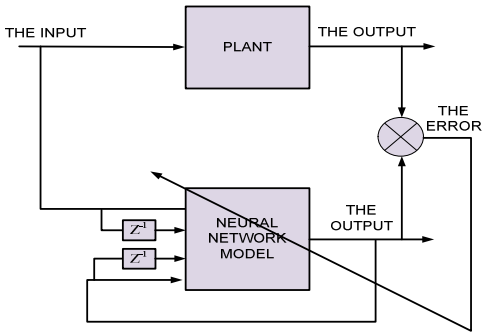


(c)

Fig. 3 Cascade PI control results: (a) reactor temperature, (b) jacket temperature, (c) valve position



(a)



(b)

Fig. 4 The structures of the dependent mode (a) and the independent mode (b)

III. Independent RBF Neural Network Model

A. Independent Mode of RBF Modelling

Using a static neural network to model a non-linear dynamic system can be classified into two modes: a dependent mode and an independent mode. In the dependent mode the process output and its delayed values are used as part of network input, and therefore the model is dependent on the process output and cannot operate independently from the process. 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 [5], [16], [18]. 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 structures of the two modes are displayed in Fig. 4.

B. RBF Network Structure

The nonlinear dynamic plant to be modelled is presented by the non-linear autoregressive with exogenous inputs (NARX) model as shown in (8):

$$y(t)=f[y(t-1),\cdots,y(t-n_y),u(t-1-d),\cdots,u(t-n_u-d)]+e(t) \quad (8)$$

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 (9), which is referred to dependent mode as the prediction uses the process output and therefore, the model cannot run independent of the process.

$$\hat{y}(t)=\hat{f}[y(t-1),\cdots,y(t-n_y),u(t-1-d),\cdots,u(t-n_u-d)] \quad (9)$$

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 (10), then the model referred to an independent model.

$$\hat{y}(t)=\hat{f}[\hat{y}(t-1),\cdots,\hat{y}(t-n_y),u(t-1-d),\cdots,u(t-n_u-d)] \quad (10)$$

The RBF network performs nonlinear mapping, and is used because of its advantages over the multi-layer perceptron (MLP) of short training time. The RBFNN in this research consists of three layers: an input layer, a hidden layer, and an output layer. The hidden layer contains a number of RBF neurons; each of them represents a single radial basis function, with associated center and width, and calculates the Euclidean distance between center c and RBF network input vector x defined by $\|x(t)-c_j(t)\|$ where $c_j(t)$ is j^{th} center, and $x(t)$ is the neural network input vector which is given as shown in (11):

$$x(t)=f[y(t-1),\cdots,y(t-n_y),u(t-1-d),\cdots,u(t-n_u-d)] \quad (11)$$

Then, the output of the hidden layer nodes 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.

$$\phi_j(t)=\exp\left(-\frac{\|x(t)-c_j(t)\|^2}{{\sigma_j}^2}\right),\ \mathrm{j}=1,\cdots,n_h \quad (12)$$

where σ_j is a positive scalar called a width and n_h is the number of centers. The network outputs are then computed as a linear weighted sum of the hidden node outputs and bias as shown in (13):

$$\hat{y}_i(t)=\sum_j^{n_h}\phi_j(t)^T w_{ji}\ ,\ i=1,\cdots,q \quad (13)$$

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

C. Training Algorithm

In this work the centers of the RBFNN are set by the K-means clustering method [3],whose objective is to minimize the sum squared distances from each input data to its closest center so that the data is adequately covered by the activation functions $\phi(t)$. Moreover, the widths are computed by the p-nearest neighbor's method [3]. 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 center to the p nearest centers as given by:

$$\sigma_i=\frac{1}{P}\sum_{d=1}^{P}\|c_i(t)-c_d(t)\|\ \ i=1,\cdots,n_h \quad (14)$$

The value of p is chosen in this work to be $p=3$.

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 [13]. Training of the RBF network weights with the ROLS algorithm is as follows. Considering (13) 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

$$Y(k)=\hat{Y}(k)+E(k)=\Phi(k)W(k)+E(k) \quad (15)$$

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 (15) can be solved for $W(k)$ using the recursive MIMO Least Squares algorithm to minimize the following time-varying cost function,

$$J(k)=\left\|\begin{bmatrix}\sqrt{\lambda}Y(k-1)\\ ----\\ y^T(k)\end{bmatrix}-\begin{bmatrix}\sqrt{\lambda}\Phi(k-1)\\ ----\\ \phi^T(k)\end{bmatrix}W(k)\right\|_F \quad (16)$$

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 in [13] that minimizing (16) is equivalent to minimizing the following cost function,

$$J(k) = \left\| \begin{bmatrix} \sqrt{\lambda}\hat{Y}(k-1) \\ ---- \\ y^T(k) \end{bmatrix} - \begin{bmatrix} \sqrt{\lambda}R(k-1) \\ ---- \\ \phi^T(k) \end{bmatrix} W(k) \right\|_F \quad (17)$$

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

$$\begin{bmatrix} \sqrt{\lambda}R(k-1) \\ ---- \\ \phi^T(k) \end{bmatrix} = Q(k) \begin{bmatrix} R(k) \\ --- \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \hat{Y}(k) \\ --- \\ \eta^T(k) \end{bmatrix} = Q^T(k) \begin{bmatrix} \sqrt{\lambda}\hat{Y}(k-1) \\ ---- \\ y^T(k) \end{bmatrix} \quad (18)$$

where Q is an orthogonal matrix. Combining (17) and (18) and considering that the *F*-norm is preserved by orthogonal transformation, the following equivalent cost function is obtained,

$$J(k) = \left\| \begin{bmatrix} \hat{Y}(k) - R(k)W(k) \\ -------- \\ \eta^T(k) \end{bmatrix} \right\|_F \quad (19)$$

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

$$R(k)W(k) = \hat{Y}(k) \quad (20)$$

and leaves the residual at sample interval k as $\left\|\eta^T(k)\right\|_F$.

Since $R(k)$ is an upper triangular matrix, $W(k)$ can be easily solved from (20) by backward substitution.

The decomposition in (18) can be achieved efficiently by applying Givens rotations to an augmented matrix to obtain the following transformation [13]:

$$\begin{bmatrix} \sqrt{\lambda}R(k-1) & \sqrt{\lambda}\hat{Y}(k-1) \\ \phi^T(k) & y^T(k) \end{bmatrix} \rightarrow \begin{bmatrix} R(k) & \hat{Y}(k) \\ 0 & \eta^T(k) \end{bmatrix} \quad (21)$$

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 (21). Then solve $W(k)$ in (20) with $R(k)$ and $\hat{Y}(k)$ obtained in (21). 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 [13]. For use in off-line mode, the Givens rotations can be computed to realize the transformation in (21) continuously to the end of training, and then W is solved finally from (20). 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 [13].

D.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, impurity factor, and valve position setpoint as shown in Fig. 5. The second step in developing the RBF model of the process is to determine the network input variables. The network input variables 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 c as defined in (22). 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 (22):

$$u = \begin{bmatrix} m_M \\ 1/h_f \\ T_{amb} \\ i \\ C \end{bmatrix}, \quad y = \begin{bmatrix} T_{jin} \\ T_{jout} \\ T \end{bmatrix} \quad (22)$$

Before training and testing, the input vector and output vector were scaled linearly into the range of [0 1] using (23). Then, in order to implement the proposed network in an independent mode, the network input vector x used the past value of the model outputs rather than the plant outputs together with the five inputs of the system as designed in (10) and shown in Fig. 8. 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 (24). 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 centers being chosen using the *K*-means clustering algorithm. Moreover, the *P*-nearest-neighbors 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 sec 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.

$$u_{scaled}(k) = \frac{u(k) - u_{\min}}{u_{\max} - u_{\min}}$$

$$y_{scaled}(k) = \frac{y(k) - y_{\min}}{y_{\max} - y_{\min}} \quad (23)$$

$$x(t) = [y(t-1)\ y(t-2)\ y(t-3)\ u(t-k-1)\ u(t-k-2)\ ]^{\mathrm{T}} \quad (24)$$

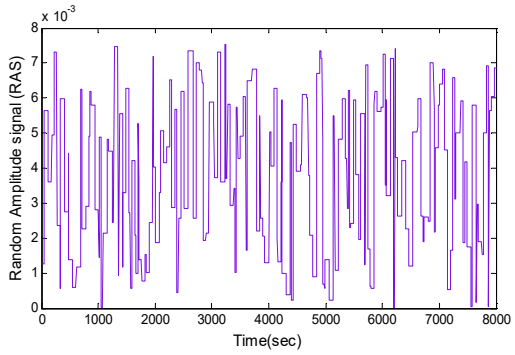
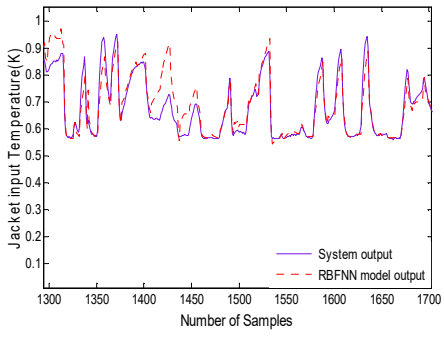
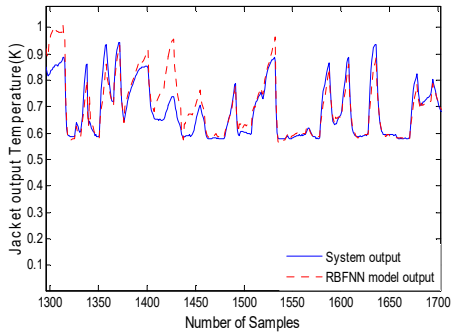


Fig. 5 RAS Signal

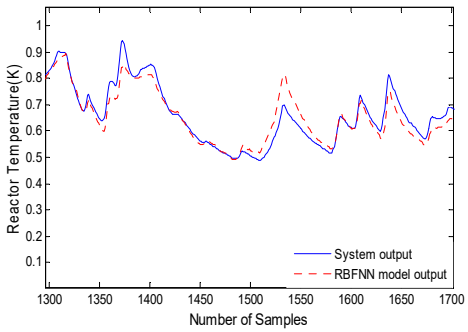
Fig. 6 displays the last 200 samples of the trained data and the first 200 samples of tested data. It can be seen that the model prediction error is quite small, and these data are with all possible disturbance and parameter uncertainties. The mean absolute error (MAE) of the scaled data for the jacket input temperature, jacket output temperature and reactor temperature are 0.0029, 0.0019, and 0.0024, respectively.



(a)



(b)



(c)

Fig. 6 Process outputs data compared with RBF model outputs during training and testing: (a) Jacket input temperature, (b) jacket output temperature, and (c) reactor temperature

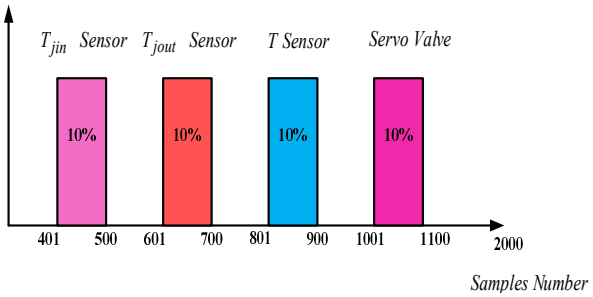


Fig. 7 Faults amplitude and occurring time

IV. Fault Detection

A. 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 possible faults here are 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. The actuator fault is the malfunction of the servo valve controlling the cooling water flow rate. The three sensor faults for jacket input temperature, jacket output temperature and the reactor temperature are simulated by

superimposing with a 10% change of the temperature at the samples t = 400 to t = 500, t = 600 to t = 700 and t = 800 to t = 900, respectively, as shown in Fig. 7. These faults are implemented to the reactor Simulink model.

The heating/cooling is controlled by an equal-percentage valve with valve position change. When the valve position $c<50\%$, cooling water with temperature of 278.71 K is injected into the cooling jacket. When the valve position $c>50\%$, steam with temperature of 449.82 K is injected into the recirculating water stream to heat up the coolant. It is assumed here that a malfunction in servo valve happened so that the valve position has a bias, which leads to increase in the temperature by 10% change of the measured inlet temperature. This fault is simulated from the sample number 1000 to 1100, as shown in Fig. 7. The actuator fault is also implemented in the Simulink model.

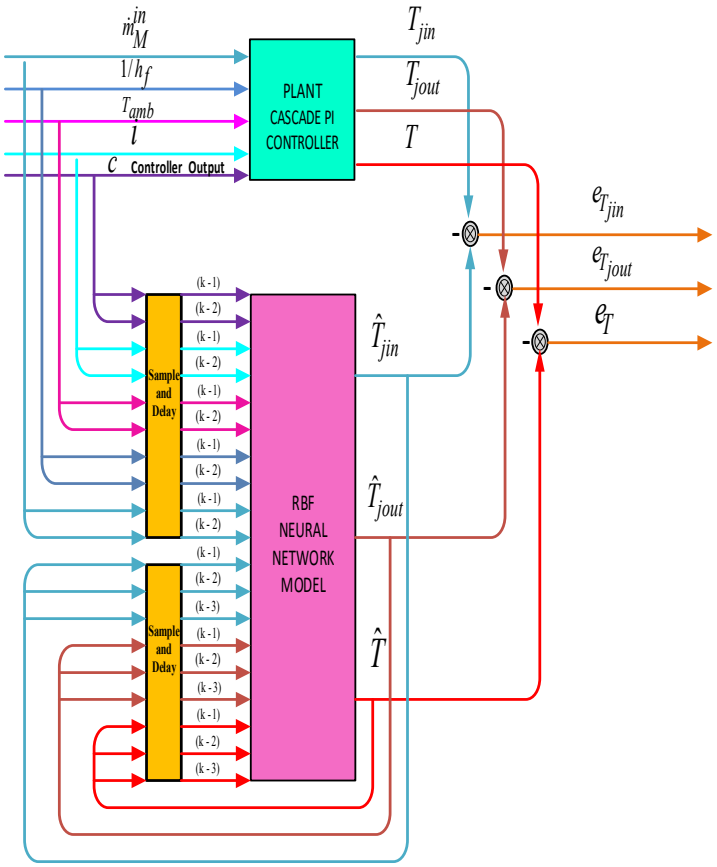


Fig. 8 The structure of FD using an independent RBFNN

B. Residual Generation

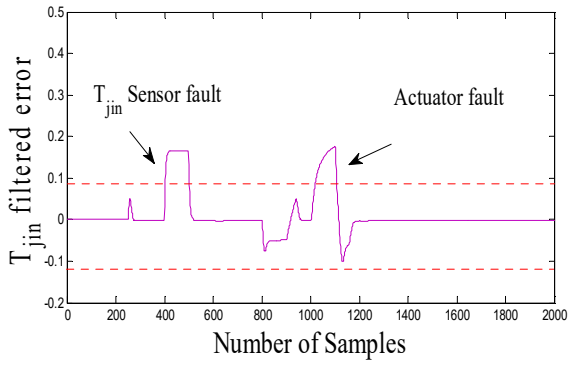
Fig. 8 demonstrates the fault detection approach. An independent model is implemented in parallel with the system to generate the residuals for detecting the sensor and actuator faults in the reactor. After training the network model with healthy random data, as described in the previous section, all four faults 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 input vector x of the independent network used the past value of the model outputs rather than the plant outputs together with the five inputs of the system as designed in (10) and shown in Fig. 8. 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. The filtered model prediction errors are shown in Fig. 10. The first model prediction error of jacket input temperature is shown in Fig. 9 (a) and that for jacket output temperature and reactor temperature are shown in Figs. 9 (b) and (c) respectively. In this study, the residual $\mathcal{E}$ is generated as the sum-squared filtered modelling error as:

$$e(t)=[y(t)-\hat{y}(t)] \tag{25}$$

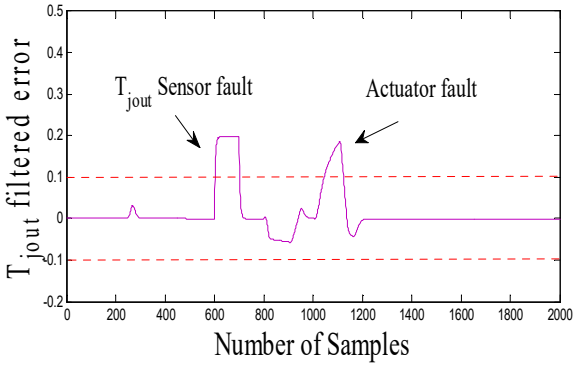
$$\varepsilon(t) = \sqrt{(e_{T_{jin}})^2 + (e_{T_{jout}})^2 + (e_T)^2} \quad (26)$$

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, 0.0054, and 0.0072, respectively. Fig. 10 demonstrates the model prediction errors after using a low pass filter. 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. Moreover, no false alarms were thereby produced, so this verifies that the proposed scheme has shown excellent diagnostic performance.

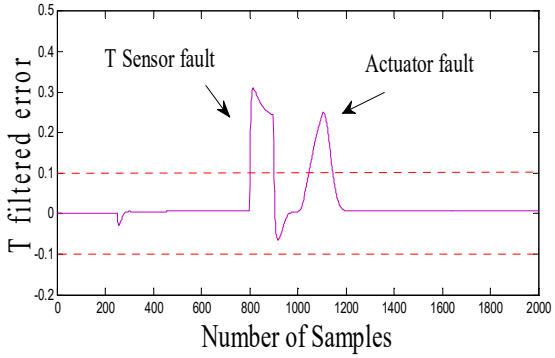
Since the independent model does not use past faulty measurements as inputs. It is observed that the neural model outputs did not track the faulty system outputs. Thus, the residuals are sensitive to these faults, and consequently can be used to detect faults in the presence of noise and modelling errors. A pre-specified threshold ρ is marked in Fig. 9. 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 Fig. 10 that all faults have been clearly detected, and no false alarm was thereby produced.



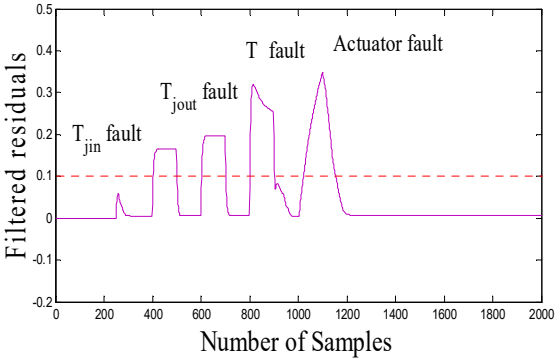
(a)



(b)



(c)



(d)

Fig. 9 (a), (b), and (c) Filtered model prediction errors. (d) sum-squared filtered modelling errors

V. Fault Isolation

The application of NNs for fault isolation has been used by many researchers. For example, in [6], [7], [17], [18], [26] used an RBF network. In [12] an MLP network used for 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 Fig. 10.

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.

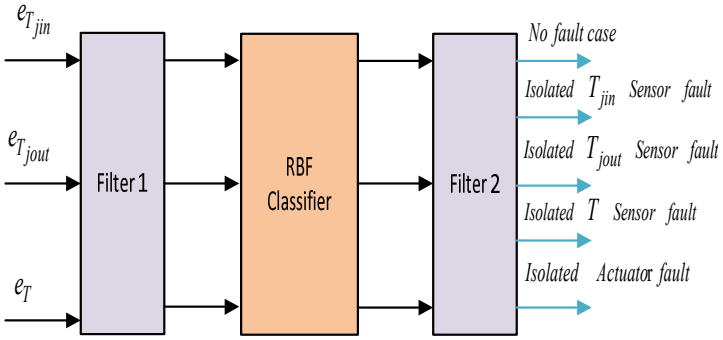


Fig. 10 Block diagram for fault isolation

The network training for classification is different from that for modelling. The centers of the classifier were chosen again using the K-means clustering algorithm, so that the sum squared distance of each input data from the center is minimized. The widths were chosen using p-nearest-neighbors. In the updating of the classifier weights, the recursive least squares (RLS) algorithm was used. The parameters of the RLS algorithm are selected as: $\mu = 0.999$, $w(0) = 10^{-6} \times U(n_h,5)$ and $p(0) = 10^{6} \times I(n_h)$, 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 centers 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 centers are chosen as 51. In addition to the optimization of weights using RLS algorithm, both center locations and amplitude of width have also been optimized. As the objective function is nonlinearly related to both the center 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 III. 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 III, the four outputs of the neural classifier after use of a filter are displayed in Fig. 11.

Fig. 11 illustrates 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.

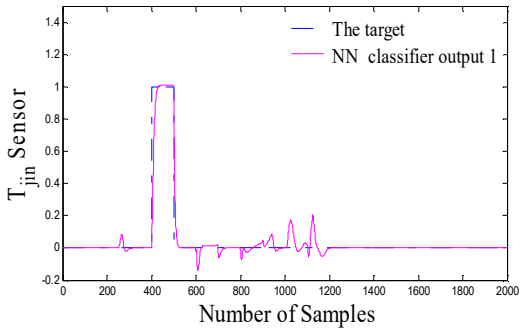
VI. Conclusion

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 disturbance 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

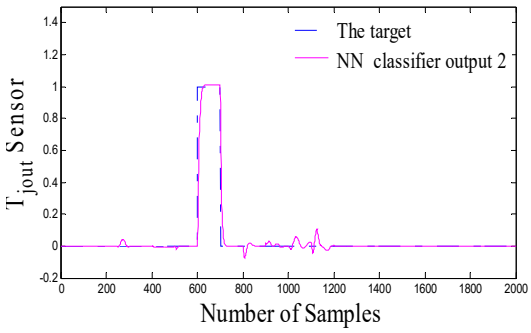
faults affecting the model prediction error vector was classified, so that the occurring fault was identified. Optimization of center location and magnitude of the width significantly increased the classifying ability. The simulation results confirmed that the simulated faults have been clearly detected and isolated with zero false alarm rates. 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.

TABLE III
CLASSIFICATION OF FAULTS WITH RESPECT TO NUMBER OF SAMPLES

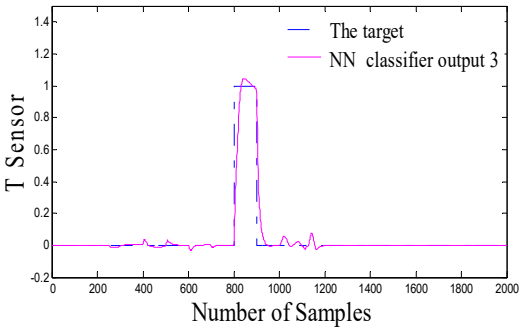
Faults	Number of samples
No fault	0 ~ 400
T_{jin} sensor fault	401 ~ 400
No fault	501 ~ 600
T_{jout} sensor fault	601 ~ 700
No fault	701 ~ 800
T sensor fault	801 ~ 900
No fault	901 ~ 1000
Actuator fault	1001 ~ 1100
No fault	1101 ~ 2000



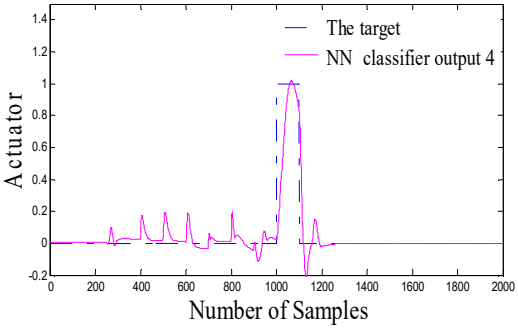
(a)



(b)



(c)



(d)

Fig. 11 (a), (b), (c), and (d) are the four filtered outputs of the network classifier

REFERENCES

[1] Barton, R. S. and Himmelblau, D. M. (1997) *Online prediction of polymer product quality in an industrial reactor using recurrent neural networks,* 114 vol.1.

[2] Beyer, M.-A., Grote, W. and Reinig, G. (2008) 'Adaptive exact linearization control of batch polymerization reactors using a Sigma-Point Kalman Filter', *Journal of Process Control,* 18(7–8), 663-675.

[3] Chen, S., Billings, S. A., Cowan, C. F. N. and Grant, P. M. (1990) 'Non-linear systems identification using radial basis functions', *International Journal of Systems Science,* 21(12), 2513-2539.

[4] Chylla, R. W. and Haase, D. R. (1993) 'Temperature control of semibatch polymerization reactors', *Computers & Chemical Engineering,* 17(3), 257-264.

[5] Deibert, R. and Isermann, R. (1992) 'Examples for fault detection in closed loops', *Annual Review in Automatic Programming,* 17(0), 235-240.

[6] Ertiame, A. M., Dingli, Y., Feng, Y. and Gomm, J. B. (2013) *Fault detection and isolation for open-loop Chylla-Haase polymerization reactor,* 1-6.

[7] Ertiame, A. M., et al. (2014). "*Robust fault diagnosis for an exothermic semi-batch polymerization reactor under open-loop*." Systems Science & Control Engineering 3(1): 14-23.

[8] Fabrizio Caccavale, Mario Iamarino, Francesco Pierri and Tufano, V. (2011) *Control and Monitoring of Chemical Batch Reactors,* London: Springer-Verlag London Limited.

[9] Ferrari, R. M. G., Parisini, T. and Polycarpou, M. M. (2008) *A robust fault detection and isolation scheme for a class of uncertain input-output discrete-time nonlinear systems,* 2804-2809.

[10] Frank, P. M. and Köppen-Seliger, B. (1997) 'Fuzzy logic and neural network applications to fault diagnosis', *International Journal of Approximate Reasoning,* 16(1), 67-88.

[11] Gertler, J. J. (1988) 'Survey of model-based failure detection and isolation in complex plants', *Control Systems Magazine, IEEE,* 8(6), 3-11.

[12] Gomm, J. B., et al. (1996). "Enhancing the non-linear modelling capabilities of MLP neural networks using spread encoding." Fuzzy Sets and Systems 79(1): 113-126.
[13] Gomm, J. B. and Yu, D. L. (2000) 'Selecting radial basis function network centers with recursive orthogonal least squares training', *Neural Networks, IEEE Transactions on,* 11(2), 306-314.
[14] Graichen, K., Hagenmeyer, V. and Zeitz, M. (2005) *Adaptive Feedforward Control with Parameter Estimation for the Chylla-Haase Polymerization Reactor,* 3049-3054.
[15] Graichen, K., Hagenmeyer, V. and Zeitz, M. (2006) 'Feedforward control with online parameter estimation applied to the Chylla–Haase reactor benchmark', *Journal of Process Control,* 16(7), 733-745.
[16] Isermann, R. (1984) 'Process fault detection based on modeling and estimation methods—A survey', *Automatica,* 20(4), 387-404.
[17] Patton, R. J. and Chen, J. (1992) *Robustness in quantitative model-based fault diagnosis,* 4/1-417.
[18] Patton, R. J., Chen, J. and Siew, T. M. (1994) *Fault diagnosis in nonlinear dynamic systems via neural networks,* 1346-1351 vol.2.
[19] Pierri, F., Paviglianiti, G., Caccavale, F. and Mattei, M. (2008) 'Observer-based sensor fault detection and isolation for chemical batch reactors', *Engineering Applications of Artificial Intelligence,* 21(8), 1204-1216.
[20] Polycarpou, M. M. and Helmicki, A. J. (1995) 'Automated fault detection and accommodation: a learning systems approach', *Systems, Man and Cybernetics, IEEE Transactions on,* 25(11), 1447-1458.
[21] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N. and Yin, K. (2003) 'A review of process fault detection and diagnosis: Part III: Process history based methods', *Computers & Chemical Engineering,* 27(3), 327-346.
[22] Wang, S. W., Yu, D. L., Gomm, J. B., Page, G. F. and Douglas, S. S. (2006) 'Adaptive neural network model based predictive control for air–fuel ratio of SI engines', *Engineering Applications of Artificial Intelligence,* 19(2), 189-200.
[23] 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(5), 1220-1226.
[24] Xiaodong, Z., Polycarpou, M. and Parisini, T. (2001) *Fault isolation in a class of nonlinear uncertain input-output systems,* 1741-1746 vol.2.
[25] Xiaodong, Z., Polycarpou, M. M. and Parisini, T. (2002) 'A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems', *Automatic Control, IEEE Transactions on,* 47(4), 576-593.
[26] Yu, D. L., Gomm, J. B. and Williams, D. (1999) 'Sensor fault diagnosis in a chemical process via RBF neural networks', *Control Engineering Practice,* 7(1), 49-55.