Model-Based Parameter Estimation for Fault Detection using Multiparametric Programming

Ernie Che Mid and Vivek Dua*

Department of Chemical Engineering, Centre of Process System Engineering (CPSE), University College London, London, United Kingdom

KEYWORDS

Parameter estimation, fault detection, multiparametric programming

ABSTRACT

Fault detection has become increasingly important for improving the reliability and safety of process systems. This paper presents a model-based fault detection methodology for nonlinear process systems. The objective of this work is to detect faults by estimating the model parameters using multiparametric programming. The parameter estimates are obtained as an explicit function of the measurements by using multiparametric programming. The diagnosis of fault is carried out by monitoring the changes of the residual of model parameters. Case studies of fault detection for single stage evaporator system and quadruple tank system are presented. A number of faulty and fault free scenarios are considered to show the effectiveness of the presented approach. The proposed approach successfully estimates the model parameters and detects the faults through simple function evaluation of explicit functions.

1. INTRODUCTION

Fault can be defined as an unpermitted deviation of at least one characteristic of a variable from an acceptable behavior while failure can be characterized as a permanent interruption of a system's ability to perform a required function under specified operating conditions.[1] Faults may take place in any system component such as actuators, sensors, plant components, or a combination of these. In any case, the fault is the primary cause of changes in the system parameters and may result in fatal damage and economic loss if the faults are not properly handled. These issues provide the motivation to develop an efficient and timely response to detecting faults and the faulty equipment to be accurately located so that corrective control action can be taken before the faults turn into a catastrophic failure. Hence, fault detection techniques have been developed for the diagnosis problem for process systems due to the increasing demand for high performance, efficiency, reliability, and safety. These requirements must be taken into account when designing, operating and maintaining the process systems to ensure that potential common mode failures are minimized and system reliability is optimized.[2-4]

Model-based approaches for fault detection (FD) have received much attention over the last decades. Model-based methods utilize a mathematical model of the process in order to detect a fault as early as possible and as accurately as possible and utilizing the concept of analytical redundancy. Analytical redundancy techniques are more cost effective compared with hardware redundancy but are more challenging due to environmental noise / disturbance and modelling error.[5] Analytical redundancy is achieved through a comparison between measured signals with its estimation from the mathematical model of the system. Hence, the basic idea in model-based fault detection scheme is to compare the available system measurements with a priori information represented by the system's mathematical model through generation of residual quantities. In the residual evaluation, an evaluated residual is compared with a threshold and fault existence decision is made if the former exceeds the latter.[6,7] Most of the model-based FD techniques are based on observer-based, parity relation, parameter estimation, or a combination of the three.[8–11] Among all the model-based FD schemes, the observer-based FD is one of the widely used techniques in fault detection. The basic idea of observer approach is to estimate the outputs of the system from the measurements by using some type of observer, and then construct the residual by an output estimate error.[12–14] The parity relation approach uses the parity check on the consistency of parity equation to generate residuals (parity vector). The inconsistency in the parity relations indicates the presence of faults.[15-17] On the other hand, the parameter estimation approach is based on the assumption that the faults are reflected in the physical system parameters and only the model structure is needed to be known.[18-20] The parameters of the actual process are repeatedly estimated on-line and the results are compared with the reference model. The parameter estimation approach is straightforward if the model parameters have an explicit mapping with the physical coefficients. The key limitation however is that it requires solving an optimization online at regular time interval. The solution of an online optimization problem is not only time consuming, but also that the solution may not converge in a reasonable time.

Work in the area of fault detection and diagnosis in the context of controller design for closed-loop system has also been reported in the literature in [21,22]. In [23], a closed-loop structure with a feedback control law is designed to decouple the dependency between certain state variables for fault detection and isolation. While in [24], distributed model predictive control systems is utilized to ensure the system stability when actuator faults occur.

In this work, a new model-based fault detection scheme using parameter estimation is proposed (Section 2). The objective is to detect faults by estimating the model parameters using multiparametric programming (MPP) to obtain the parameter estimates as an explicit function of the measurements. The estimates of model parameter are thus obtained without the need to solve an online optimization problem. The diagnosis of fault is carried out by monitoring the changes of the residual of model parameters. Any substantial discrepancy between estimated model parameters and the observed model parameters indicates changes in the process and may be interpreted as a fault. The application of the methodology is illustrated for a single stage evaporator system and quadruple tank system case studies (Section 3). A number of faulty and fault free scenarios are considered to show the effectiveness of the presented approach. Concluding remarks are presented in Section 4.

2. METHODOLOGY

2.1. General formulation for fault detection using parameter estimation

Fault detection using parameter estimation techniques were amongst the first methodologies to be considered for the purpose of performing early fault detection and diagnosis for critical systems.[3,25-26] This method relies on the principle that possible faults in the monitored system can be associated with specific parameters and the mathematical model of the system represented by nonlinear ordinary differential equations (ODEs). Parameter estimation method for fault detection can be successful if (a) the mathematical model of process system is accurate; (b) the experimental data is available; (c) and the model parameters are related to physical system parameters of the equipment or process fluids.[7] The objective of fault detection is to solve the following:

Problem 1:

$$\varepsilon_{FD} = \min_{\boldsymbol{\theta},\mathbf{x}(t)} \sum_{j\in J} \sum_{i\in I} \left\{\hat{x}_j(t_i) - x_j(t_i)\right\}^2 \tag{1}$$

Subject to:

$$\frac{dx_j(t)}{dt} = f_j(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}, t),\ j \in J \tag{2}$$

$$x_j(t = 0) = x_j^0, j \in J \tag{3}$$

$$t \in \left[0, t_f\right] \tag{4}$$

where $x(t)$ is the *J*-dimensional vector of state variables in the given ODE system, $\hat{x}_j(t_i)$ represents the measurements of the state variables at the time points t_i and $\boldsymbol{\theta}$ is the vector of parameters that must be estimated such that the error, ε_{FD}, between the measurements, $\hat{x}_j$ and model predicted values of state variables, x_j is minimized. There are several approaches for solving the parameter estimation problem, and can be categorized as decomposition approaches[27-29] and sequential/simultaneous approaches[30-33]. However, these methods are computationally expensive to implement for on-line fault detection method.[34-38] To overcome this problem, we present the parameter estimates for fault detection using multiparametric programming.

2.2. Discretization of Ordinary Differential Equation

Consider that the ODE initial value problem (2)–(3) is to be solved on the interval, $t \in \left[0, t_f\right]$ and step size is given by Δt, the Euler method provides

$$x_j(k+1) = x_j(k) + \Delta t f_j(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) \tag{5}$$

For $I = 1$, substituting (5) in Problem 1, the fault detection is given by the following Nonlinear Program (NLP):

Problem 2:

$$\varepsilon_{MPP} = \ min_{\theta, x_j(k=1)} \sum_{j \in J} \left\{ \hat{x}_j(k+1) - x_j(k+1) \right\}^2 \tag{6}$$

Subject to:

$$h_j = x_j(k+1) - \ x_j(k) - \Delta t f_j(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) = 0, \ j \in J \tag{7}$$

$$x_j(k=0) = x_j^0, j \in J \tag{8}$$

where h_j represents the set of nonlinear algebraic equations obtained by discretizing the ODEs given by equation (2).

2.3. Parameter estimation using multiparametric programming

Various researchers have employed different methods for estimating the model parameters, $\boldsymbol{\theta}$. One key issue in fault detection is to estimate the model parameters precisely (accuracy) and as fast as possible (speed). Accuracy is important to avoid false-positives, while speed ensures that the faults are identified quick enough to be able to take corrective actions in a timely manner. In this work, we will focus on developing a solution technique for fault detection using multiparametric programming (MPP) to estimate the model parameters, $\boldsymbol{\theta}$. Multiparametric programming provides the optimization variables as an explicit function of the parameter.[39-44] In this work, the model parameters, $\boldsymbol{\theta}$ are considered as optimization variables and the measurements, $\hat{x}_j(k+1)$ as the parameters, in the context of multiparametric programming. To obtain this, the Karush-Kuhn-Tucker (KKT) conditions for Problem 2 are first obtained as follows.

The Lagrangian function is given by

$$L = g + \sum_{j \in J} \lambda_j h_j \tag{9}$$

where

$$g = \sum_{j \in J} \{\hat{x}_j(k+1) - x_j(k+1)\}^2 \tag{10}$$

$$h_j = x_j(k+1) - x_j(k) - \Delta t f_j(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) = 0, \; j \in J \tag{11}$$

The KKT conditions are given by the Equality Constraints as follows

$$\nabla_\theta L = \nabla_\theta g + \nabla_{\theta_j} \sum \lambda_j h_j \tag{12}$$

$$h_j{=}0 \tag{13}$$

Equations (12) and (13) represent a square system of multiparametric nonlinear algebraic equations. In this work, these equations are solved symbolically using Mathematica[45] to obtain $\boldsymbol{\theta}$ as an explicit function of $\hat{\boldsymbol{x}}$ i.e. $\boldsymbol{\theta}(\hat{\boldsymbol{x}})$ is obtained. This allows computation of parameter estimates, $\boldsymbol{\theta}$, by simple function evaluation of $\boldsymbol{\theta}(\hat{\boldsymbol{x}})$. The algorithm for model-based parameter estimation for fault detection using multiparametric programming is summarized in Table 1 and fault detection based upon the estimates obtain is described in the next section.

2.4. Fault detection analysis

Fault detection and diagnosis (FDD) is carried out by monitoring the changes of the residual of model parameters. In order to define the residual generator for the aforementioned analysis, the residual, $\boldsymbol{r}$ which is a scalar or vector valued signal containing information on the time and location of the occurrence of the fault is designed. The residual for fault detection method is defined as

$$\boldsymbol{r} = |\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}| \tag{14}$$

The estimated model parameters, $\boldsymbol{\theta}$, should be close to "true" model parameters, $\hat{\boldsymbol{\theta}}$, when no fault is present. The abnormal condition can be detected by comparing the residual with a decision or threshold function, $\boldsymbol{T}$. Any substantial discrepancy indicates changes in the process and may be interpreted as a fault. A fault is then declared if $\boldsymbol{r}$ surpasses a certain threshold, $\boldsymbol{T}$, as follows:

$\boldsymbol{r} < \boldsymbol{T} \Rightarrow$ no fault has occurred (15)

$\boldsymbol{r} \geq \boldsymbol{T} \Rightarrow$ a fault has occurred (16)

3. CASE STUDIES

In this section, two case studies are presented to demonstrate the applicability of the approach proposed in the previous section.

3.1 Single Stage Evaporator System

In this section, we present single stage evaporator system[46] as a case study to illustrate the solution steps. A mathematical model of single stage evaporator system is described as:

$$\frac{dW}{dt} = F - (\delta W + E_C) - V \tag{17}$$

$$\frac{dT}{dt} = \frac{\beta F x_F + (V-F)(T-T_B)}{W} \tag{18}$$

where

$$V = \left(\frac{UA(T_S-T) - FC_p(T-T_F) - Q_L}{\Delta H_V}\right) \tag{19}$$

W and T are the state variables representing the holdup and temperature respectively and the model parameters for this process system are heat transfer coefficient, UA and composition of feed, x_F. V is the vapor flowrate from evaporator, F is the feed flowrate, T_s is the steam temperature, T_B is the temperature for normal boiling point of solvent, T_F is the temperature of the feed system, C_p is the heat capacity of the solution, Q_L is the rate of heat loss to the surroundings and ΔH_v is the heat of vaporization of the solvent. The diagram of the evaporator plant is shown in Figure 1. The parameter values used for simulation of the reactor are shown in Table 2. In this system, the evaporator operation is assumed to be at 1 atm (101.3 kPa) of pressure. The temperature for normal boiling point of solvent, T_B is 100 ˚C and boiling point, β is 8.33 ˚C per 10% solute. The rate of heat loss to the surroundings, Q_L is 400 kJ/min and a steam temperature, T_s is 136 ˚C to correspond to a typical low-pressure steam line of about 200 kPa gauge.

The formulation and solution of the parameter estimation problem using multiparametric programming for single stage evaporator is summarized as follow:

i) The nonlinear ODE model (17)-(19) is discretized and reformulated as the following algebraic equations:

$$W(k+1) = W(k) + \Delta t(F - (\delta W(k) + E_C) - V) \tag{20}$$

$$T(k+1) = T(k) + \Delta t\left(\frac{\beta F x_F + (V-F)(T(k) - T_B)}{W(k)}\right) \tag{21}$$

where

$$V = \left(\frac{UA(T_S - T(k)) - FC_p(T(k) - T_F) - Q_L}{\Delta H_V}\right) \tag{22}$$

(ii) The discrete-time fault detection problem is formulated as the following NLP:

Problem 3:

$$\varepsilon_{MPP} = \min_{UA, x_F} \sum_{k=0}^{K} \left\{\left(\widehat{W}(k+1) - W(k+1)\right)^2 + \left(\hat{T}(k+1) - T(k+1)\right)^2\right\} \tag{23}$$

Subject to:

$$h_1 = W(k+1) - W(k) - \Delta t(F - (\delta W(k) + E_C) - V) = 0 \tag{24}$$

$$h_2 = T(k+1) - T(k) - \Delta t\left(\frac{\beta F x_F + (V-F)(T(k) - T_B)}{W(k)}\right) = 0 \tag{25}$$

$$k \in [0{,}500] \tag{26}$$

(iii) Substituting (22),(24) and (25) into (23), we obtain the following:

$$g = \left(\widehat{W}(k+1) - W(k) - \Delta t\left(F - (\delta W(k) + E_C) - \left(\frac{UA(T_S - T(k)) - FC_p(T(k) - T_F) - Q_L}{\Delta H_V}\right)\right)\right)^2 +$$

$$\left(\hat{T}(k+1) - T(k) - \Delta t\left(\frac{\beta F x_F + (\left(\frac{UA(T_S - T(k)) - FC_p\left(T(k) - T_F\right) - Q_L}{\Delta H_V}\right) - F)(T(k) - T_B)}{W(k)}\right)\right)^2 \tag{27}$$

The gradient of g with respect to UA and x_F is given by

$$\frac{\partial g}{\partial UA} = -(-((\frac{1}{\Delta H_V}) - \Delta H_V \Delta t E_c + \Delta H_V \Delta t F + \Delta t Q_L + C_p \Delta t F T(k) - C_p \Delta t F T_F + \Delta t T(k) UA -$$

$$\Delta t T_s UA + \Delta H_V W(k) - \Delta H_V \Delta t \delta W(k))(\frac{1}{\Delta H_V})(2(\Delta t T(-\Delta t T_s) + \widehat{W}(k+1))) -$$

$$(\frac{1}{\Delta H_V W(k)})(2\Delta t(T(k) - T_B)(-T(k) + T_s)(\hat{T}(k+1) - \Delta t(\frac{T(k)}{\Delta t} +$$

$$\frac{(T(k) - T_B)\left(-F + \frac{-Q_L - C_p F\left(T(k) - T_F\right) + (-T(k) + T_S)UA}{\Delta H_V}\right) + \beta F x_F}{W(k)}))) = 0 \tag{28}$$

$$\frac{\partial g}{\partial x_F} =$$

$$-(\frac{1}{W(k)})(2\beta \Delta t F(\hat{T}(k+1) - dt(\frac{T(k)}{\Delta t} + \left(\frac{1}{W(k)}\right)((T(k) - T_B)\ (-F +$$

$$\frac{-Q_L - C_p F(T(k) - T_F) + (-T(k) + T_s)UA}{\Delta H_V}) + \beta F x_F)))) = 0 \tag{29}$$

(iv) Equality Constrains in (28) and (29) are solved analytically in Mathematica and the solution is given by

$$UA(k) = -\frac{-\Delta H_V \Delta t E_c + \Delta H_V dtF + \Delta t Q_L + C_p \Delta t F T(k) - C_p \Delta t F T_F + \Delta H_V W(k) - \Delta H_V \Delta t \delta W(k) - \Delta H_V \widehat{W}(k+1)}{\Delta t(T(k) - T_s)} \tag{30}$$

$$x_F(k) = -(\frac{1}{\beta \Delta t F})(-\Delta t E_c T(k) + \Delta t E_c T_B + 2T(k)W(k) - \hat{T}(k+1)W(k) - T_B W(k) -$$

$$\Delta t T(k)\delta W(k) + \Delta t T(k)\beta \delta W(k) - T(k)\widehat{W}(k+1) + T(k)\beta \widehat{W}(k+1)) \tag{31}$$

(v) The estimated model parameters, (UA and x_F) are calculated using the measurements ($\widehat{W}$ and $\widehat{T}$) and control variable, F and the value of the residuals of model parameters is monitored for fault detection and diagnosis.

The model parameters, UA and x_F are obtained as explicit functions of the measurements and control variable, ($\widehat{W}(k)$, $\widehat{T}(k)$ and F) as in Equations (30) and (31). Simple function evaluation can be carried out to estimate the model parameter and detect faults without the need to solve the online optimization problem.

Fault-free scenario

In the fault-free scenario, the measured value and model predicted value of state variables for holdup, W and temperature, T are shown in Figure 2 and Figure 3 respectively. Noise has been added to the system as random data to evaluate the effectiveness of the proposed method using multiparametric programming. The model parameters are calculated using the measurements and step size, $\Delta t = 1$ min. The model parameters are only estimated after state variables have reached the steady state value at 50 min. The evaluation of parameter estimation for step size, $\Delta t = 1$ min are shown in Figure 4 and Figure 5 for UA and x_F. It can be seen from Figure 4 and Figure 5 that the estimated model parameters are close to true model parameters. The diagnosis of fault is carried out by monitoring the value of the residuals of model parameters. The result is shown in Figure 6 and Figure 7. No fault was detected since the residual is less than the threshold. Threshold is chosen as 5% from the nominal system parameter values.

Faulty scenario

An investigation for faulty scenario was implemented for this case study. To demonstrate the application of parameter estimation for the evaporator, the model parameter (UA and x_F) are changed as shown in Table 3 and the model is simulated to obtain data for parameter estimation.

In this faulty scenario, the state variable for holdup, W and temperature, T are simulated based on faulty condition as described in Table 3. Noise has been added to the system as random data to evaluate the effectiveness of the proposed method using multiparametric programming. Figure 8 and Figure 9 show the measured value and model predicted value of state variables for faulty scenario for holdup and temperature. The model parameters are only estimated after state variables have reached the steady state value at 50 min with step size,$\Delta t = 1$ min. Figure 10 and Figure 11 show the evaluation of estimated model parameters, UA and x_F respectively. From these figures, we can see that the estimated parameter, UA decreases from 40.548 kJ m/min ˚C at 75 min to 36.50 kJ m/min ˚C (at 375 min). Estimated model parameter for x_F also changes from 0.032 mass fraction (at 165 min) to 0.025 mass fraction (at 285 min).

The diagnosis of fault is carried out by monitoring the value of the residuals of model parameters and the result is shown in Figure 12 and Figure 13. Figure 12 shows that from 75 min to 375 min, percentage of residual for UA increases slowly up to 10 % and fault is declared from 225 min to 375 min since the residual for UA is more or equal to 5% of threshold value. As shown in Figure 13, fault for x_F is declared at 165 min to 285 min as the percentage of residual for x_F is 20 %. Multiparametric programming based parameter estimation is thus able to accurately and quickly identify the faults in the evaporator system.

3.2 Quadruple Tank System

The implementation of fault detection using multiparametric programming is further discussed using quadruple tank system. A simulation study on the quadruple-tank system[47] with faults event was simulated. The system consists of four interacting tanks, two pumps and two valves as shown in Figure 14. The system aims at controlling the liquid levels in the lower tanks. By adjusting the bypass valves of the system, the proportion of the water pumped into different tanks can be changed to adjust the degree of interaction between the pump throughputs and the water levels. The mathematical model of quadruple tank system is described as:

$$\frac{dH_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gH_1} + \frac{a_3}{A_1}\sqrt{2gH_3} + \frac{\gamma_1\kappa_1}{A_1}\,v_1 \tag{32}$$

$$\frac{dH_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gH_2} + \frac{a_4}{A_2}\sqrt{2gH_4} + \frac{\gamma_2\kappa_2}{A_2}\,v_2 \tag{33}$$

$$\frac{dH_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gH} + \frac{(1-\gamma_2)\kappa_2}{A_3}\,v_2 \tag{34}$$

$$\frac{dH_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gH_4} + \frac{(1-\gamma_1)\kappa_1}{A_4}\,v_1 \tag{35}$$

For tank i, A_i is the cross –section of the tank, a_i is the cross-section of the outlet hole, and H_i is the water level. The voltage applied to pump i is v_i, and the corresponding flow is $\kappa_i v_i$. Tank leakage faults are considered to test the proposed fault detection. These leaks are assumed to be produced by holes at the bottom of the tanks, such that the outflow is lost. The parameter values of the quadruple tank system are given in Table 4.

The formulation and solution of the parameter estimation problem using multiparametric programming for quadruple tank system is summarized as follow:

i) The nonlinear ODE model (32)-(35) is discretised and reformulated as the following algebraic equations:

$$H_1(k+1) = H_1(k) + \Delta t\left(-\frac{a_1}{A_1}\sqrt{2gH_1(k)} + \frac{a_3}{A_1}\sqrt{2gH_3(k)} + \frac{\gamma_1\kappa_1}{A_1}\,v_1\right) \tag{36}$$

$$H_2(k+1) = H_2(k) + \Delta t\left(-\frac{a_2}{A_2}\sqrt{2gH_2(k)} + \frac{a_4}{A_2}\sqrt{2gH_4(k)} + \frac{\gamma_2\kappa_2}{A_2}\,v_2\right) \tag{37}$$

$$H_3(k+1) = H_3(k) + \Delta t\left(-\frac{a_3}{A_3}\sqrt{2gH_3(k)} + \frac{(1-\gamma_2)\kappa_2}{A_3}\,v_2\right) \tag{38}$$

$$H_4(k+1) = H_4(k) + \Delta t\left(-\frac{a_4}{A_4}\sqrt{2gH_4(k)} + \frac{(1-\gamma_1)\kappa_1}{A_4}\,v_1\right) \tag{39}$$

(ii) The discrete-time fault detection problem is formulated as the following NLP:

Problem 4:

$$\varepsilon_{MPP} = \min_{a_1,a_2,a_3,a_4} \sum_{k=0}^{K}\Big\{(\hat{H}_1(k+1) - H_1(k+1))^2 + (\hat{H}_2(k+1) - H_2(k+1))^2 +$$

$$(\hat{H}_3(k+1) - H_3(k+1))^2 + (\hat{H}_4(k+1) - H_4(k+1))^2\Big\} \tag{40}$$

Subject to:

$$h_1 = H_1(k+1) - H_1(k) - \Delta t\left(-\frac{a_1}{A_1}\sqrt{2gH_1(k)} + \frac{a_3}{A_1}\sqrt{2gH_3(k)} + \frac{\gamma_1\kappa_1}{A_1}\,v_1\right) = 0 \tag{41}$$

$$h_2 = H_2(k+1) - H_2(k) - \Delta t\left(-\frac{a_2}{A_2}\sqrt{2gH_2(k)} + \frac{a_4}{A_2}\sqrt{2gH_4(k)} + \frac{\gamma_2\kappa_2}{A_2}\,v_2\right) = 0 \tag{42}$$

$$h_3 = H_3(k+1) - H_3(k) - \Delta t\left(-\frac{a_3}{A_3}\sqrt{2gH_3(k)} + \frac{(1-\gamma_2)\kappa_2}{A_3}\,v_2\right) = 0 \tag{43}$$

$$h_4 = H_4(k+1) - H_4(k) - \Delta t\left(-\frac{a_4}{A_4}\sqrt{2gH_4(k)} + \frac{(1-\gamma_1)\kappa_1}{A_4}\,v_1\right) = 0 \tag{44}$$

$$k\epsilon[0,650] \tag{45}$$

(iii) Substituting (41)-(44) into (40), we obtain the following:

$$g = (\hat{H}_1(k+1) - H_1(k) - \Delta t\left(-\frac{a_1}{A_1}\sqrt{2gH_1(k)} + \frac{a_3}{A_1}\sqrt{2gH_3(k)} + \frac{\gamma_1\kappa_1}{A_1}\,v_1\right))^2 +$$

$$(\hat{H}_2(k+1) - H_2(k) - \Delta t\left(-\frac{a_2}{A_2}\sqrt{2gH_2(k)} + \frac{a_4}{A_2}\sqrt{2gH_4(k)} + \frac{\gamma_2\kappa_2}{A_2}\,v_2\right))^2 + (\hat{H}_3(k+1) -$$

$$H_3(k) - \Delta t\left(-\frac{a_3}{A_3}\sqrt{2gH_3(k)} + \frac{(1-\gamma_2)\kappa_2}{A_3}\,v_2\right))^2 +$$

$$(\hat{H}_4(k+1) - H_4(k) - \Delta t\left(-\frac{a_4}{A_4}\sqrt{2gH_4(k)} + \frac{(1-\gamma_1)\kappa_1}{A_4}\,v_1\right))^2 \tag{46}$$

The gradient of g with respect to a_1, a_2, a_3 and a_4 is given by

$$\frac{\partial g}{\partial a_1} = \left(\frac{1}{A_1}\right)\left(2.829\mathit{\Delta} t\sqrt{gH_1(k)}\left(\widehat{H}_1(k+1) - \mathit{\Delta} t\left(\frac{H_1(k)}{\mathit{\Delta} t} - \frac{\sqrt{2gH_1(k)}a_1}{A_1} + \frac{\sqrt{2gH_3(k)}a_3}{A_1} + \frac{\gamma_1\kappa_1 v_1}{A_1}\right)\right)\right) = 0 \qquad (47)$$

$$\frac{\partial g}{\partial a_2} = (\frac{1}{A_2})(2.829\mathit{\Delta} t\sqrt{gH_2(k)}(\widehat{H}_2(k+1) - \mathit{\Delta} t(\frac{H_2(k)}{\mathit{\Delta} t} - \frac{\sqrt{2gH_2(k)}a_2}{A_2} + \frac{\sqrt{2gH_4(k)}a_4}{A_2} + \frac{\gamma_2\kappa_2 v_2}{A_2}))) = 0 \qquad (48)$$

$$\frac{\partial g}{\partial a_3} = -(\frac{1}{A_1})(2.829\mathit{\Delta} t\sqrt{gH_3(k)}(\widehat{H}_1(k+1) - \mathit{\Delta} t(\frac{H_1(k)}{\mathit{\Delta} t} - \frac{\sqrt{2gH_1(k)}a_1}{A_1} + \frac{\sqrt{2gH_3(k)}a_3}{A_1} + \frac{\gamma_1\kappa_1 v_1}{A_1}))) - 2\mathit{\Delta} t(0 - \frac{\sqrt{2gH_3(k)}}{A_3})(\widehat{H}_3(k+1) - \mathit{\Delta} t(\frac{1.H_3(k)}{\mathit{\Delta} t} - \frac{a_3\sqrt{2gH_3(k)}}{A_3} + \frac{\kappa_2 v_2(1-\gamma_2)}{A_3})) = 0 \qquad (49)$$

$$\frac{\partial g}{\partial a_4} = -(\frac{1}{A_2})(2.829\mathit{\Delta} t\sqrt{gH_4(k)}(\widehat{H}_2(k+1) - \mathit{\Delta} t(\frac{H_2(k)}{\mathit{\Delta} t} - \frac{\sqrt{2gH_2(k)}a_2}{A_2} + \frac{a_3\sqrt{2gH_3(k)}}{A_2} + \frac{\gamma_2\kappa_2 v_2}{A_2}))) - 2\mathit{\Delta} t(0 - \frac{\sqrt{2gH_4(k)}}{4})(\widehat{H}_4(k+1) - \mathit{\Delta} t(\frac{1.H_4(k)}{\mathit{\Delta} t} - \frac{a_4\sqrt{2gH_4(k)}}{A_4} + \frac{\kappa_1 1(1-\gamma_1)}{A_4})) = 0 \qquad (50)$$

(iv) The Equality Constrains in (47)-(50) are solved analytically in Mathematica and the solution is given by

$$a_1 =$$

$$-(\frac{1}{0.-\frac{16\Delta t^4(gH_1(k))^{1.}(gH_3(k))^{1.}}{A_1^2 A_3^2}})(1.(-1.(\frac{4\Delta t^2(gH_3(k))^{1.}}{A_1^2} -$$

$$(\frac{1}{A_3})(2.829\Delta t^2\sqrt{gH_3(k)}(0.-\frac{\sqrt{2gH_3(k)}}{A_3})))(-\frac{2.829\Delta t H_1(k)\sqrt{gH_1(k)}}{A_1} + \frac{2.829\Delta t\sqrt{gH_1(k)}\widehat{H}_1(k+1)}{A_1} -$$

$$\frac{2.829\Delta t^2\sqrt{gH_1(k)}\gamma_1\kappa_1 v_1}{A_1^2}) - (\frac{1}{A_1^2})(4\Delta t^2\sqrt{gH_1(k)}\sqrt{gH_3(k)}(\frac{2.829\Delta t H_1(k)\sqrt{gH_3(k)}}{A_1} -$$

$$\frac{2.829\Delta t\widehat{H}_1(k+1)\sqrt{gH_3(k)}}{A_1} + 2.\Delta t H_3(k)(0. - \frac{\sqrt{2gH_3(k)}}{A_3}) - 2\Delta t(0. - \frac{\sqrt{2gH_3(k)}}{A_3})\widehat{H}_3(k+1) +$$

$$\frac{2.829\Delta t^2\sqrt{gH_3(k)}\gamma_1\kappa_1 v_1}{A_1^2} + \frac{2.\Delta t^2(0.-\frac{\sqrt{2gH_3(k)}}{A_3})\kappa_2 v_2(1-\gamma_2)}{A_3}))) \qquad (51)$$

$$a_2 =$$

$$-(\frac{1}{A_2^2\Delta t\left(gH_2(k)\right)^{1.5}\left(gH_4(k)\right)^{1.}})(0.707(0. - A_2^3 H_2(k)(gH_2(k))^{1.}(gH_4(k))^{1.} + A_2^3(gH_2(k))^{1.}\widehat{H}_2(k+$$

$$1)(gH_4(k))^{1.} - 1.A_2^2A_4(gH_2(k))^{1.}H_4(k)(gH_4(k))^{1.} + 1.A_2^2A_4(gH_2(k))^{1.}(gH_4(k))^{1.}\widehat{H}_4(k+$$

$$1) - 1.A_2^2\Delta t(gH_2(k))^{1.}(gH_4(k))^{1.}\kappa_1v_1 + 1.A_2^2\Delta t(gH_2(k))^{1.}(gH_4(k))^{1.}\gamma_1\kappa_1v_1 -$$

$$A_2^2\Delta t(gH_2(k))^{1.}(gH_4(k))^{1.}\gamma_2\kappa_2v_2)) \qquad (52)$$

$$a_3 = -(\frac{1}{A_1^2\Delta t\left(gH_1(k)\right)^{1}\left(gH_3(k)\right)^{1.5}})(0.707(0. -2.22\times 10^{-16}A_1^3H_1(k)\left(gH_1(k)\right)^{1.}\left(gH_3(k)\right)^{1.} +$$

$$2.22\times 10^{-16}A_1^3\left(gH_1(k)\right)^{1.}\widehat{H}_1(k+1)\left(gH_3(k)\right)^{1.} - 1.A_1^2A_3\left(gH_1(k)\right)^{1.}H_3(k)\left(gH_3(k)\right)^{1.} +$$

$$1.A_1^2A_3\left(gH_1(k)\right)^{1.}\left(gH_3(k)\right)^{1.}\widehat{H}_3(k+1) - 1.A_1^2\Delta t\left(gH_1(k)\right)^{1.}\left(gH_3(k)\right)^{1.}\kappa_2v_2 - 2.22\times$$

$$10^{-16}.A_1^2\Delta t(gH_1(k))^{1.}(gH_3(k))^{1.}\gamma_1\kappa_1v_1 - 1.A_1^2\Delta t(gH_1(k))^{1.}(gH_3(k))^{1.}\gamma_2\kappa_2v_2)) \qquad (53)$$

$$a_4 = -(\frac{1}{0.-\frac{16\Delta t^4\left(gH_2(k)\right)^{1.}\left(gH_4(k)\right)^{1.}}{A_2^2A_4^2}})\ 1.((\frac{1}{A_2^2})(4\Delta t^2\sqrt{gH_2(k)}\sqrt{gH_4(k)}(-\frac{2.829\Delta tH_2(k)\sqrt{gH_2(k)}}{A_2} +$$

$$\frac{2.829\Delta t\sqrt{gH_2(k)}\widehat{H}_2(k+1)}{A_2} - \frac{2.829\Delta t^2\sqrt{gH_2(k)}\gamma_2\kappa_2v_2}{A_2^2})) + \frac{1}{A_2^2}(4\Delta t^2(gH_2(k))^{1.}(\frac{2.829\Delta tH_2(k)\sqrt{gH_4(k)}}{A_2} -$$

$$\frac{2.829\Delta t\sqrt{gH_4(k)}\widehat{H}_2(k+1)}{A_2} + 2.\Delta tH_4(k)(0. - \frac{\sqrt{2gH_4(k)}}{A_4}) - 2.\Delta t(0. - \frac{\sqrt{2gH_4(k)}}{A_4})\widehat{H}_4(k+1) +$$

$$\frac{2.\Delta t^2(0.-\frac{\sqrt{2gH_4(k)}}{A_4})\kappa_1v_1(1.-\gamma_1)}{A_4} + \frac{2.829\Delta t^2\sqrt{2gH_4(k)}\gamma_2\kappa_2v_2}{A_2^2})) \qquad (54)$$

(v) The estimated model parameters, a_i is evaluated using the measurements, $\widehat{H}_i(k)$ and control variables, (V_1 and V_2) and the value of the residuals of model parameters is monitored for fault detection and diagnosis.

The model parameters,a_i are obtained as explicit functions of the measurements, $\widehat{H}_i(k)$ and control variables, (V_1 and V_2) as in Equations (51)-(54). Simple function evaluation can be carried out to estimate the model parameter without the need to solve the online optimization problem.

Fault-free scenario

In the fault-free scenario, the measured value and model predicted value for water level, H_i is shown in Figure 15. Noise has been added to the system as random data to evaluate the effectiveness of the proposed method using multiparametric programming. The model parameters, a_i are calculated using the measurements and step size, $\Delta t = 5$ s. The evaluation of model parameters parameter estimation for step size, $\Delta t = 5$ s are shown in Figure 16. As shown in Figure 16, the estimated model parameters, a_i are close to true model parameters. The diagnosis of fault is carried out by monitoring the value of the residuals of model parameters. The result is shown in Figure 17 and no leakage was detected since the residual is less than the threshold. Threshold is chosen as 5% from the nominal system.

Faulty scenario

An investigation for faulty scenario was implemented for this case study. It is assumed that the fault takes place due to leak of Tank 1 and Tank 2, resulting in changes in cross section of outlet holes, a_1 and a_2, in Tank 1 and Tank 2. The faults considered are modelled as changes in model parameters as shown in Table 5. In this faulty scenario, the state variables for water level tank, H_i are simulated based on faulty condition as described in Table 5. Noise has been added to the system as random data to evaluate the effectiveness of the proposed method using multiparametric programming. Figure 18 shows the noisy measured value and model predicted value for water level, H_i used to evaluate the model parameters.

Figure 19 shows the evaluation of estimated model parameter value, a_1. We can see that the estimated parameter for a_1 have increased from 0.071 cm^2 to 0.08165 cm^2 at 50 s to 150 s and increased from 0.071 cm^2 to 0.0781 cm^2 from 350 s to 450 s. While in estimated the model parameter for a_2, the result shows that from 200 sec to 300 sec, there is an increased of model parameter, a_2 from 0.057 cm^2 to 0.06556 cm^2 and increased from 0.057 cm^2 to 0.0627 cm^2 from 350 s to 450 s. There are no changes in a cross section of output holes in Tank 3 and Tank 4 as the estimated model parameter for a_3 and a_4 shows no difference.

The residual of model parameters is monitored for FDD and the result is shown in Figure 20. This figure shows that the fault is declared for Tank 1 as residual for a_1 achieves threshold value at 50 s to 150 s and 350 s to 450 s while fault in Tank 2 is declared at 200 s to 300 s and 350 s to 450 s. These results indicate that there are leakages at Tank 1 and Tank 2 at specified times as discussed above. The figure also shows that no leakages are detected in Tank 3 and Tank 4.

4. CONCLUDING REMARKS

In this work, we proposed the multiparametric programming method to estimate the parameter estimation for fault detection. In this method, a square system of parametric nonlinear algebraic equations is solved symbolically to obtain model parameter as an explicit function of measurements. A limitation of the proposed approach is that the symbolic solution of the parametric nonlinear algebraic equations may not always be possible and in this work Mathematica© was used for obtaining the solution. The model parameters are thus computed efficiently by performing simple function evaluations, without the need to solve an online optimization problem. An implementation of proposed methods is performed by two case studies (single stage evaporator and quadruple tank system). The results show that the implementation of multiparametric programming for parameter estimation successfully obtained model parameters as an explicit function of measurements. Noise was added to the system as random data to test the effectiveness of the proposed method. This method is able to accurately estimate the model parameters and quickly identify fault. In multiparametric programming, the online computational burden is replaced by simple function evaluations. Replacing the optimization problem of fault detection with simple and efficient computations has given multiparametric programming significant advantages in estimating model parameters. The proposed fault detection approach using multiparametric programming thus provides quick and accurate fault detection. The issues pertaining fault tolerant control will be the subject of future work.

AUTHOR INFORMATION

Corresponding Author

*E-mail: v.dua@ucl.ac.uk

ACKNOWLEDGMENTS

The authors thank MoHE (Ministry of Higher Education) Malaysia and UniMAP (University Malaysia Perlis) for the financial support.

Table 1. Parameter estimation using multiparametric programming algorithm

Step 1.	Discretize nonlinear ODE model (2) to algebraic equations as given in (5)
Step 2.	Formulate fault detection optimization problem as a nonlinear programming (NLP) problem as given in (6) - (8)
Step 3.	Formulate KKT conditions for (6) - (8) as given (9) - (13)
Step 4.	Solve the Equality Constrains (12) - (13) of the KKT conditions parametrically to obtain Lagrange multiplies and model parameters, $\boldsymbol{\theta}(\hat{\boldsymbol{x}})$ as a function of measurements, $\hat{\boldsymbol{x}}$
Step 4.	Screen the solutions obtained in the previous step and ignore solutions with imaginary parts
Step 5.	Calculate the estimated model parameters, $\boldsymbol{\theta}$ using the measurement, $\hat{\boldsymbol{x}}$ by simple evaluation of $\boldsymbol{\theta}(\hat{\boldsymbol{x}})$

Table 2. Model parameters for single-stage evaporator system

Parameter	Value	Description
U	43.6 kJ/(min m ˚C)	heat transfer coefficient
A	0.93 m2	area of heat transfer
x_F	0.032 mass fraction	composition of the feed
T_s	136 ˚C	steam temperature in the steam chest
T_B	100 ˚C	normal boiling point of solvent
C_p	4.18 kJ/(kg ˚C)	heat capacity of the solution
T_F	88 ˚C	temperature of the feed system
Q_L	400.0 kJ/min	rate of heat loss to the surroundings
ΔH_v	2240 kJ/kg	heat of vaporization of the solvent
β	8.33 ˚C	boiling point elevation per mass fraction of solute
δ	0.06 (kg/min)/kg holdup	constant
E_c	0.0454 kg/min	constant
F	2.27 kg/min	feed flowrate

Table 3. Faulty scenario for single stage evaporator system

Fault parameter	UA^F	x_F^F
% change in value	-10.0	-20.0
Type of change	ramp	step
Starting time of change (min)	75	165
Stop time of change (min)	375	285

Table 4. Model parameters for the quadruple tank system

Parameters	Values	Units	Description
A_1; A_2; A_3; A_4	28; 32; 28; 32	cm^2	Cross-section of Tank i
a_1; a_2; a_3; a_4	0.071; 0.057; 0.071; 0.057	cm^2	Cross-section of the outlet hole
v_1 ; v_2	3.00; 3.00	V	Input voltage
κ_1; κ_2	3.33; 3.35	cm^3 / Vs	
γ_1; γ_2	0.7; 0.6	-	
g	981.0	cm / s^2	Acceleration of gravity

Table 5. Faulty scenario for quadruple tank system

Time	Fault parameter	a_1^F	a_2^F
50 s – 150 s	% change in value	+ 15.0	0
200 s – 300 s	% change in value	0	+ 15.0
350s – 450 s	% change in value	+ 10.0	+ 10.0

REFERENCES

[1] Isermann, R.; Ballé, P. Trends in the Application of Model Based Fault Detection and Diagnosis of Technical Processes. *Control Eng. Pract.* **1997**, *5* (5), 709.

[2] Isermann, R. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*; 2006.

[3] Isermann, R. Process Fault Detection Based on Modeling and Estimation Methods - A Survey. *Automatica* **1984**, *20* (4), 387.

[4] Yu, X.; Jiang, J. A Survey of Fault-Tolerant Controllers Based on Safety-Related Issues. *Annu. Rev. Control* **2015**, *39*, 46.

[5] Passino, K. M. Bridging the Gap between Conventional and Intelligent Control. *IEEE Control Syst.* **1993**, *13* (3), 12.

[6] Frank, P. M. Analytical and Qualitative Model-Based Fault Diagnosis – A Survey and Some New Results. *Eur. J. Control* **1996**, *2* (1), 6.

[7] Isermann, R. Model-Based Fault-Detection and Diagnosis - Status and Applications. *Annu. Rev. Control* **2005**, *29*, 71.

[8] Gao, Z.; Cecati, C.; Ding, S. X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part I: Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2015**, *62* (6), 3768.

[9] Dai, X.; Gao, Z. From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis. *IEEE Trans. Ind. Informatics* **2013**, *9* (4), 2226.

[10] Hwang, I.; Kim, S.; Kim, Y.; Seah, C. . A Survey of Fault Detection , Isolation , and Reconfiguration Methods. *IEEE Trans. Control Syst. Technol.* **2010**, *18* (3), 636.

[11] Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A Review of Process Fault Detection and Diagnosis Part I: Quantitative Model-Based Methods. *Comput. Chem. Eng.* **2003**, *27* (3), 293.

[12] Mhaskar, P.; Gani, A.; El-farra, N. H.; Mcfall, C.; Christofides, P. D.; Davis, J. F. Integrated Fault-Detection and Fault-Tolerant Control of Process Systems. *AIChE J.* **2006**, *52* (6), 2129.

[13] Mhaskar, P.; McFall, C.; Gani, A.; Christofides, P. D.; Davis, J. F. Isolation and Handling of Actuator Faults in Nonlinear Systems. *Automatica* **2008**, *44* (1), 53.

[14] Du, M.; Mhaskar, P. Isolation and Handling of Sensor Faults in Nonlinear Systems. *Automatica* **2014**, *50* (4), 1066.

[15] Gertler, J. Fault Detection and Isolation Using Parity Relations. *Control Eng. Pract.* **1997**, *5* (5), 653.

[16] Chow, E. Y.; Willsky, A. S. Analytical Redundancy and the Design of Robust Failure Detection Systems. *IEEE Trans. Autom. Control* **1984**, *29* (7), 603.

[17] Gertler, J. Survey of Model-Based Failure Detection and Isolation in Complex Plants. *Control Syst. Mag. IEEE* **1988**, *8* (6), 3.

[18] Isermann, R. Fault Diagnosis of Machines via Parameter Estimation and Knowledge processing—Tutorial Paper. *Automatica* **1993**, *29* (4), 815.

[19] Garatti, S.; Bittanti, S. A New Paradigm for Parameter Estimation in System Modeling. *Int. J. Adapt. Control Signal Process.* **2012**, *27*, 667.

[20] Huang, B. Detection of Abrupt Changes of Total Least Squares Models and Application in Fault Detection. *IEEE Trans. Control Syst. Technol.* **2001**, *9* (2), 357.

[21] Mhaskar, P.; Liu, J.; Christofides, P. D. *Fault-Tolerant Process Control: Methods and Applications*; Springer-Verlag: London, England, 2013.

[22] McFall, C.; Pena, D. M. de la; Ohran, B.; Christofides, P. D.; Davis, J. F. Fault Detection and Isolation for Nonlinear Process Systems Using Asynchronous Measurements. *Ind. Eng. Chem. Res* **2008**, *47*, 10009–10019.

[23] Ohran, B.; Pena, D. M. de la; Christofides, P. D.; Davis, J. F. Enhancing Data-Based Fault Isolation Through Nonlinear Control. *Aiche J.* **2008**, *54*, 223–241

[24] Chilin, D.; Liu, J.; Chen, X.; Christofides, P. D. Fault detection and isolation and fault tolerant control of a catalytic alkylation of benzene process. *Chem. Eng. Sci.* **2012**, *78*, 155–166

[25] Park, S.; Himmelblau, D. M. Fault Detection and Diagnosis via Parameter Estimation in Lumped Dynamic Systems. *Ind. Eng. Chem. Process Des. Dev.* **1983**, 482–487.

[26] Pouliezos, A.; Stavrakakis, G.; Lefas, C. Fault detection using parameter estimation. Qual. Reliab. Eng. Int. **1989**, 5 (4), 283–290.

[27] Varah, J.M. A Spline Least Squares Method for Numerical Parameter Estimation in Differential Equations. *SIAM J. Sci. Stat. Comput* **1982**, *3* (1), 28.

[28] Michalik, C.; Chachuat, B.; Marquardt, W. Incremental Global Parameter Estimation in Dynamic Systems. *Ind. Eng. Chem. Res.* **2009**, *48*, 5489.

[29] Dua, V. An Artificial Neural Network Approximation Based Decomposition Approach for Parameter Estimation of System of Ordinary Differential Equations. *Comput. Chem. Eng.* **2011**, *35* (3), 545

[30] Hwang, M.; Seinfeld, J. H. A New Algorithm for the Estimation of Parameters in Ordinary Differential Equations. *AIChE J.* **1972**, *18* (1), 90.

[31] Tjoa, I.-B.; Biegler, L. T. Simultaneous Solution and Optimization Strategies for Parameter Estimation of Differential-Algebraic Equation Systems. *Ind. Eng. Chem. Res.* **1991**, *30* (1981), 376.

[32] Bilardello, P.; Joulia, X.; Le Lann, J. M.; Delmas, H.; Koehret, B. A General Strategy for Parameter Estimation in Differential-Algebraic Systems. *Comput. Chem. Eng.* **1993**, *17* (5-6), 517.

[33] Dua, V.; Dua, P. A Simultaneous Approach for Parameter Estimation of a System of Ordinary Differential Equations, Using Artificial Neural Network Approximation. *Ind. Eng. Chem. Res.* **2011**, *51* (4), 1809.

[34] Papamichail, I.; Adjiman, C. S. Global Optimization of Dynamic Systems. *Comput. Chem. Eng.* **2004**, *28* (3), 403.

[35] Espie, D. M.; Macchietto, S. The Optimal Design of Dynamic Experiments. *Exp. Des.* **1989**, *35* (2), 223.

[36] Papamichail, I.; Adjiman, C. S. A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations. *J. Glob. Optim.* **2002**, *24* (1), 1.

[37] Sakizlis, V.; Perkins, J. D.; Pistikopoulos, E. N. Parametric Controllers in Simultaneous Process and Control Design. *Comput. Aided Chem. Eng.* **2003**, *15* (C), 1020.

[38] Vassiliadis, V. S., Sargent, R. W. H.,Pantelides, C. C. Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints. *Ind. Eng. Chem. Res.* **1994**, *33* (9), 2111.

[39] Dua, V.; Pistikopoulos, E. N. Algorithms for the Solution of Multiparametric Mixed-Integer Nonlinear Optimization Problems. *Ind. Eng. Chem. Res.* **1999**, *38* (10), 3976.

[40] Pistikopoulos, E. N.; Dua, V.; Bozinis, N. A.; Bemporad, A.; Morari, M. On-line optimization via off-line parametric optimization tools. *Comput. Chem. Eng.* **2002**, *26* (2), 175–185

[41] Pistikopoulos, E. N. Perspectives in Multiparametric Programming and Explicit Model Predictive Control. *AIChE J.* **2009**, *55*, 1918–1925

[42] Oberdieck, R.; Diangelakis, N. A.; Papathanasiou, M. M.; Nascu, I.; Pistikopoulos, E. N. POP - Parametric Optimization Toolbox. *Ind. Eng. Chem. Res.* **2016**, *55* (33), 8979–8991

[43] Efstratios N. Pistikopoulos; Georgiadis, M.; Dua, V. *Multi-Parametric Programming: Theory, Algorithms, and Applications. Volume 1*. Process Systems Engineering Series, Wiley-VCH,Weinheim 2007.

[44] Pistikopoulos, E. N.; Georgiadis, M.; Dua, V. *Multi-parametric Model-Based Control: Theory and Applications. Volume 2*; Process Systems Engineering Series, Wiley-VCH,Weinheim, 2007.

[45] Dua, V. Mixed Integer Polynomial Programming. *Comput. Chem. Eng.* **2015**, *72*, 387.

[46] Dalle Molle, D. T.; Himmelblau, D. M. Fault Detection in a Single-Stage Evaporator via Parameter Estimation Using the Kalman Filter. *Ind. Eng. Chem. Res.* **1987**, *26* (12), 2482.

[47] Johansson, K. H. The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero. *IEEE Trans. Control Syst. Technol.* **2000**, *8* (3), 456.

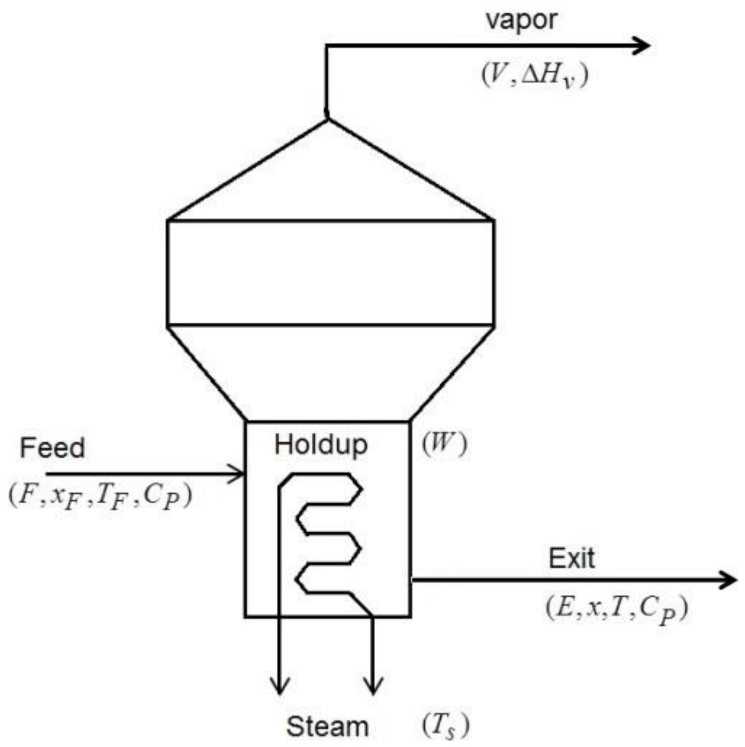

Figure 1. Evaporator configuration and notation

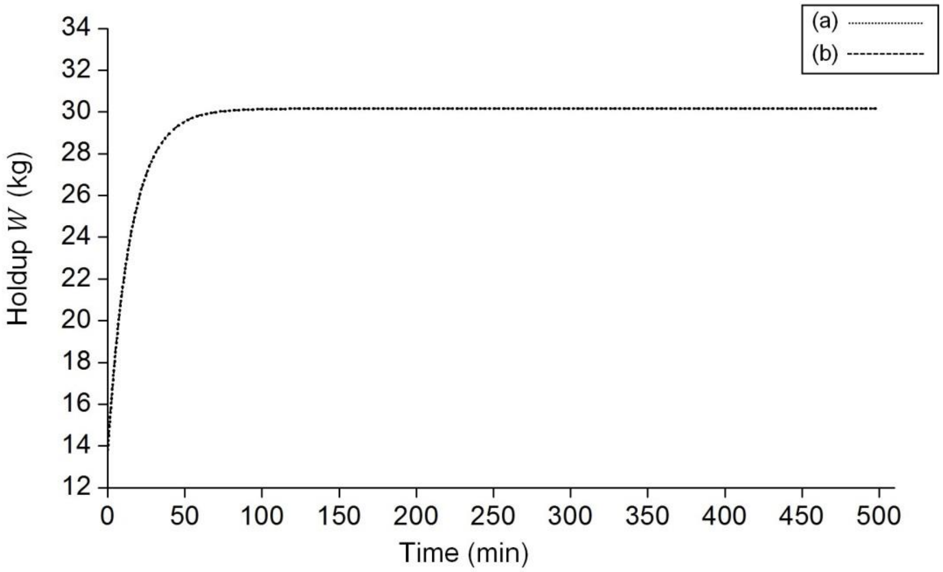

Figure 2. (a) $\widehat{W}$, measured value of state variable, W (b) W, model predicted value of state variable, W

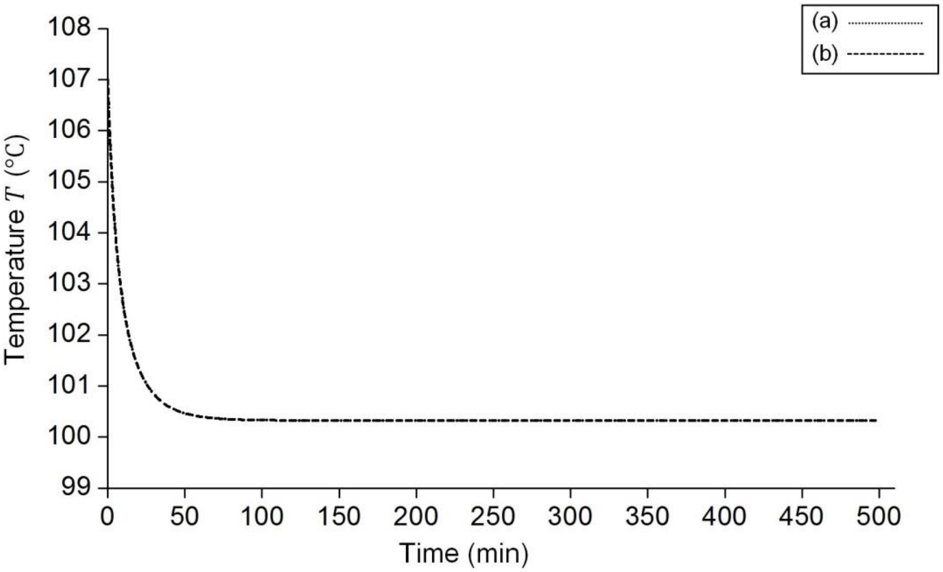

Figure 3. (a) $\widehat{T}$, measured value of state variable, T (b) T, model predicted value of state variable, T

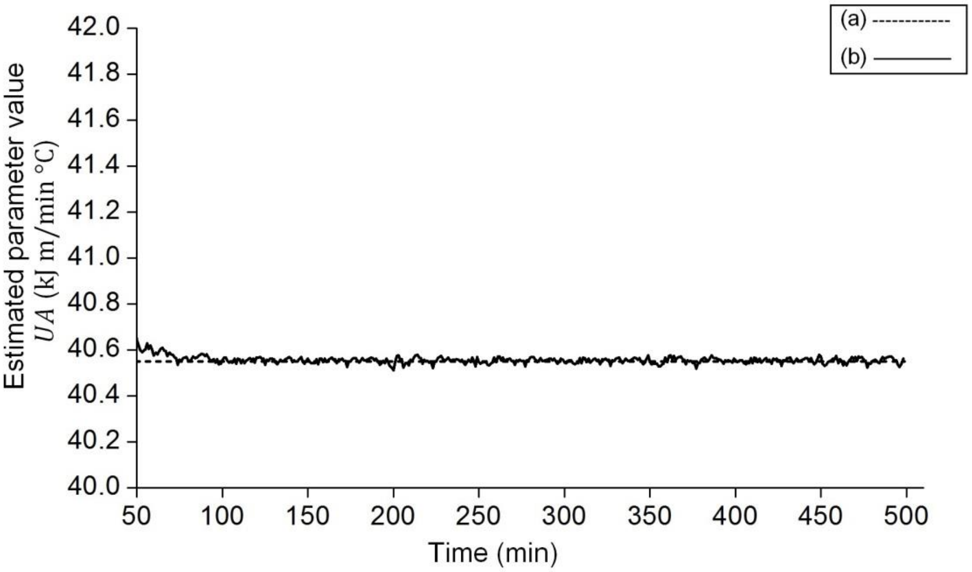

Figure 4. Model parameter values: (a) $\widehat{UA}$, true value of UA (b) UA, estimated model parameter value of UA

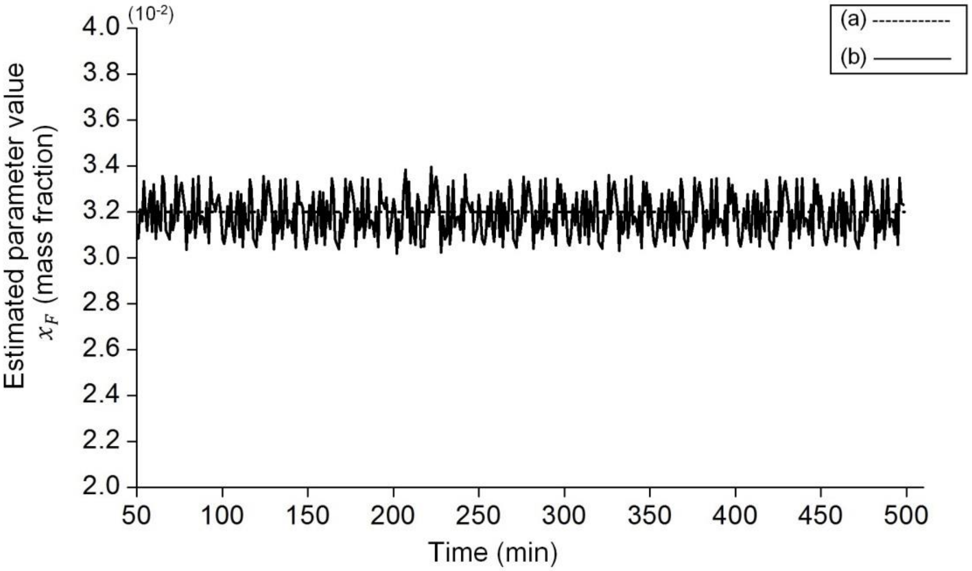

Figure 5. Model parameter values: (a) $\hat{x}_F$, true value of x_F (b) x_F, estimated model parameter value of x_F

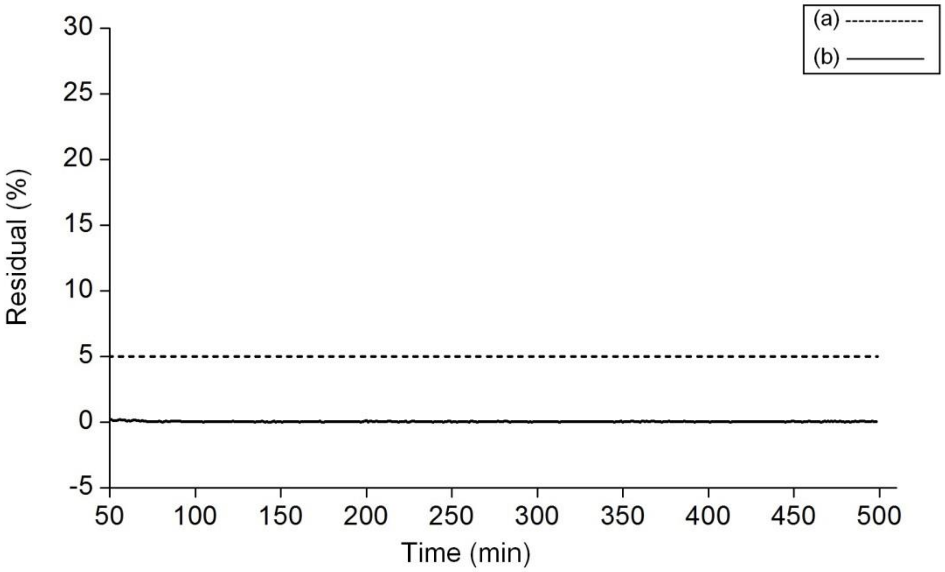

Figure 6. (a) Threshold value (b) Residual evaluation of estimated model parameters for UA

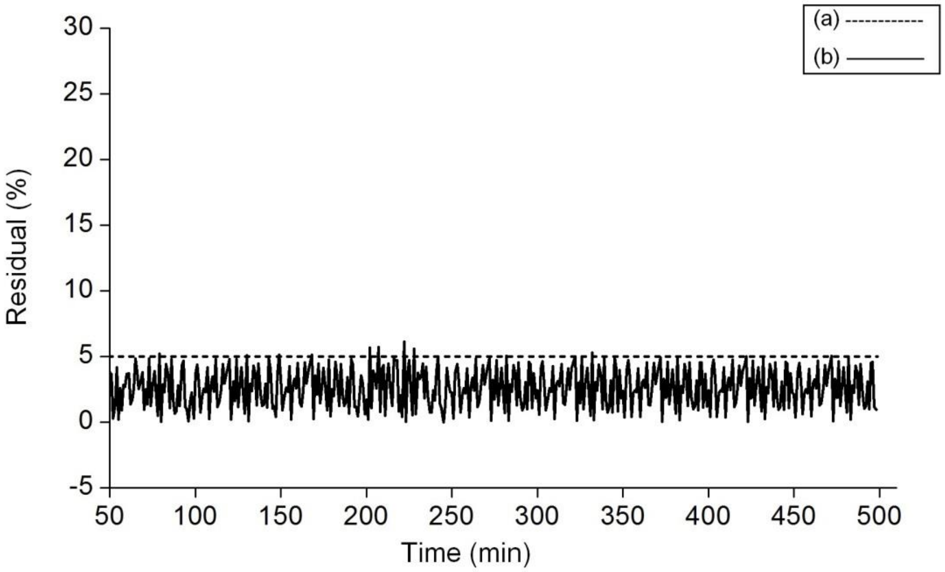

Figure 7. (a) Threshold value (b) Residual evaluation of estimated model parameters for x_F

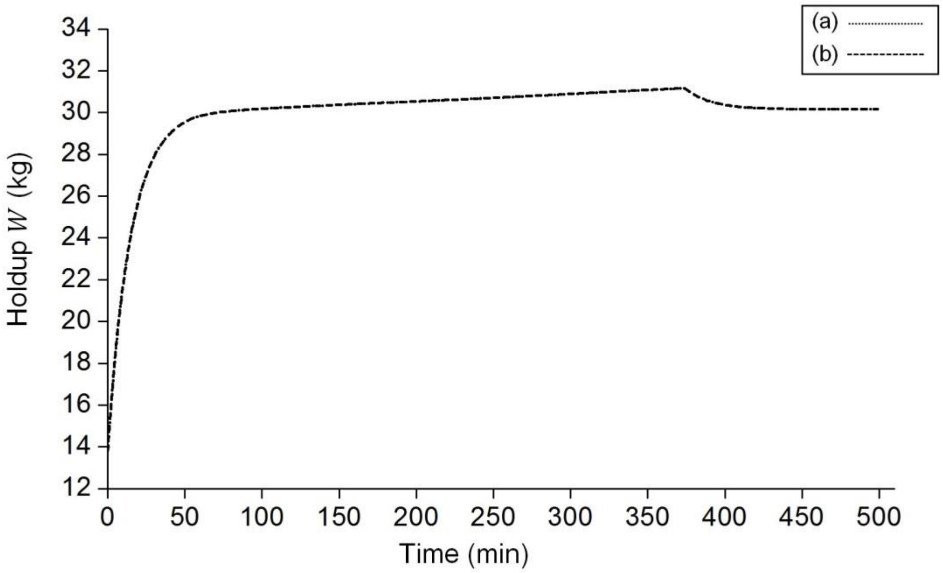

Figure 8. Faulty scenario (a) $\widehat{W}^{\mathrm{F}}$, measured value of state variable, W (b) W, model predicted value of state variable, W

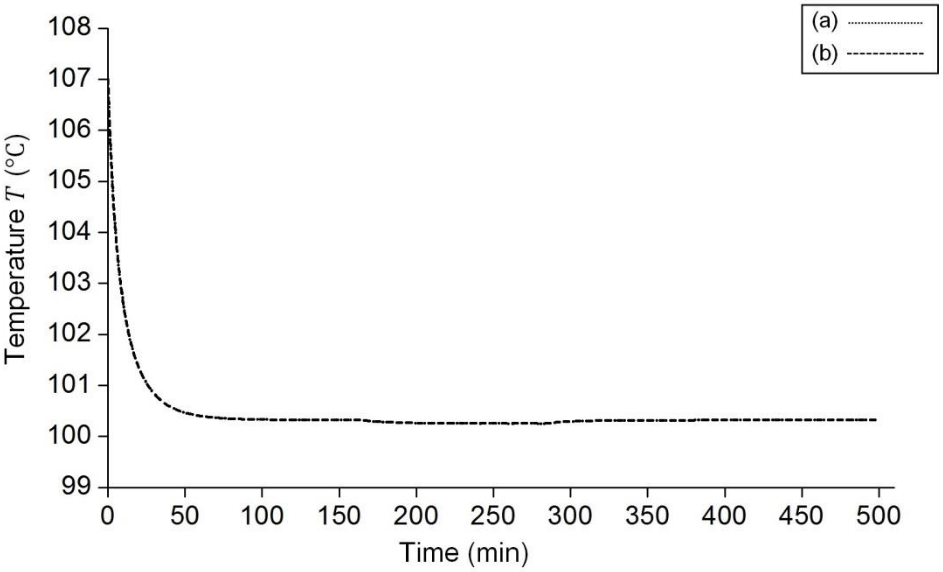

Figure 9. Faulty scenario (a) $\widehat{T}^{F}$, measured value of state variable, T (b) T, model predicted value of state variable, T

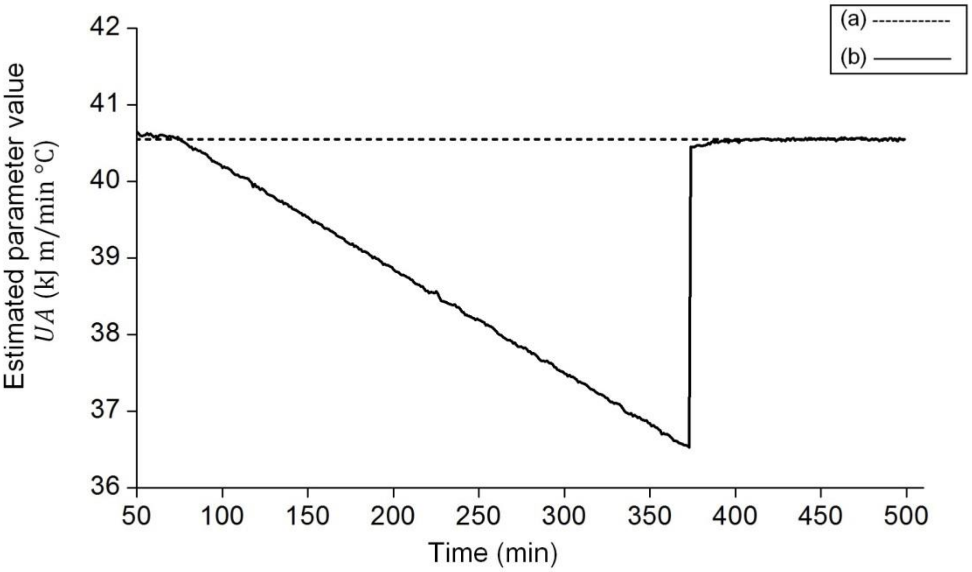

Figure 10. Model parameter values: (a) $\widehat{UA}$, true value of UA (b) UA^F, estimated model parameter value of UA

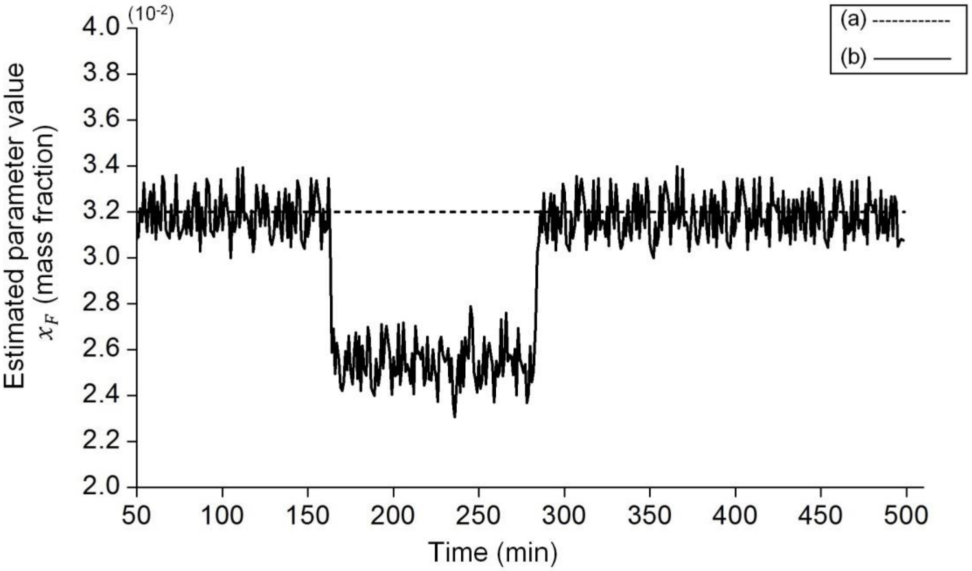

Figure 11. Model parameter values: (a) $\hat{x}_F$, true value of x_F (b) x_F^F, estimated model parameter value of x_F

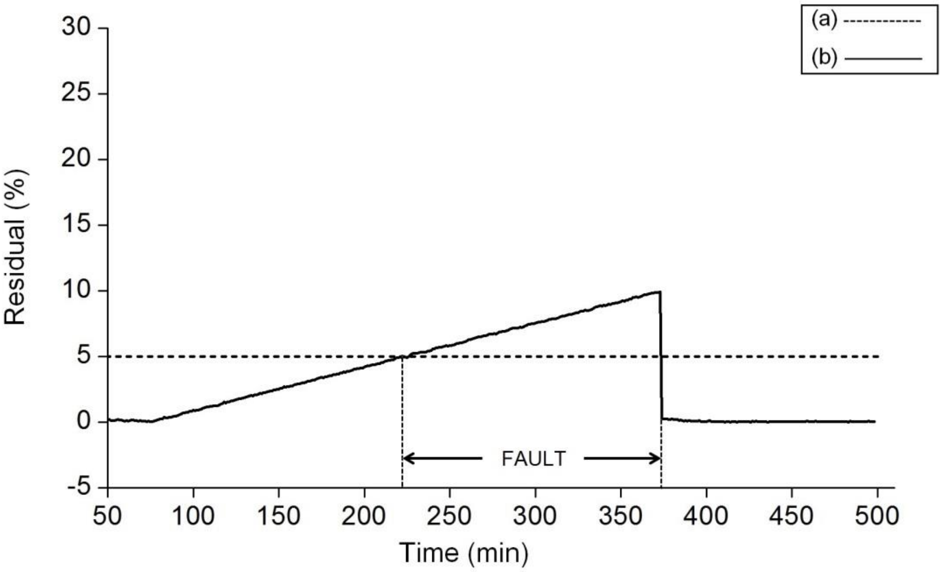

Figure 12. (a) Threshold value (b) Residual evaluation of estimated model parameters for UA^F

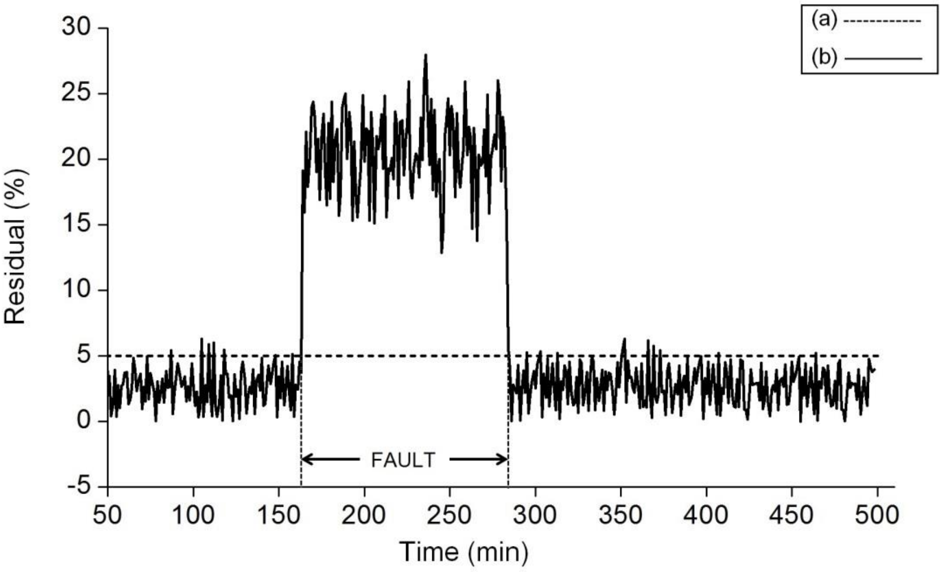

Figure 13. (a) Threshold value (b) Residual evaluation of estimated model parameters for x_F^F

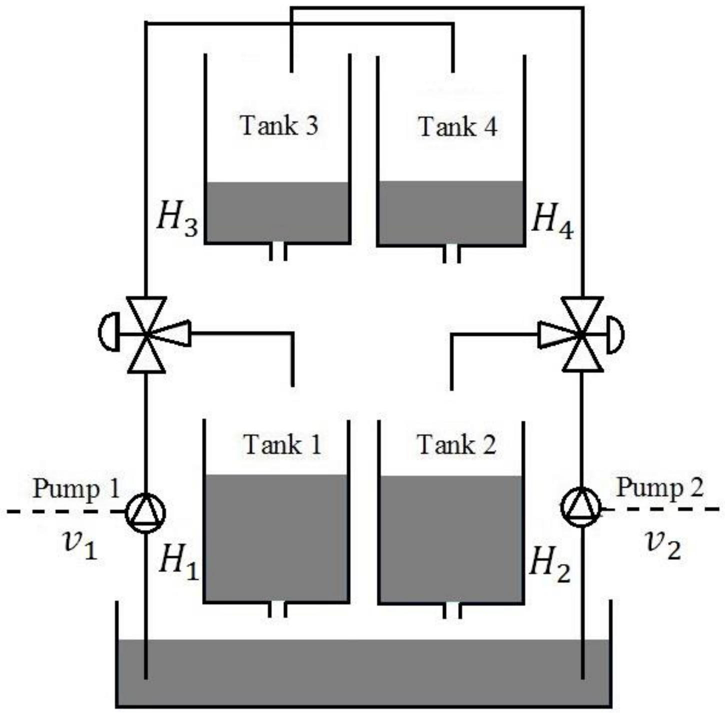

Figure 14. Quadruple tank process

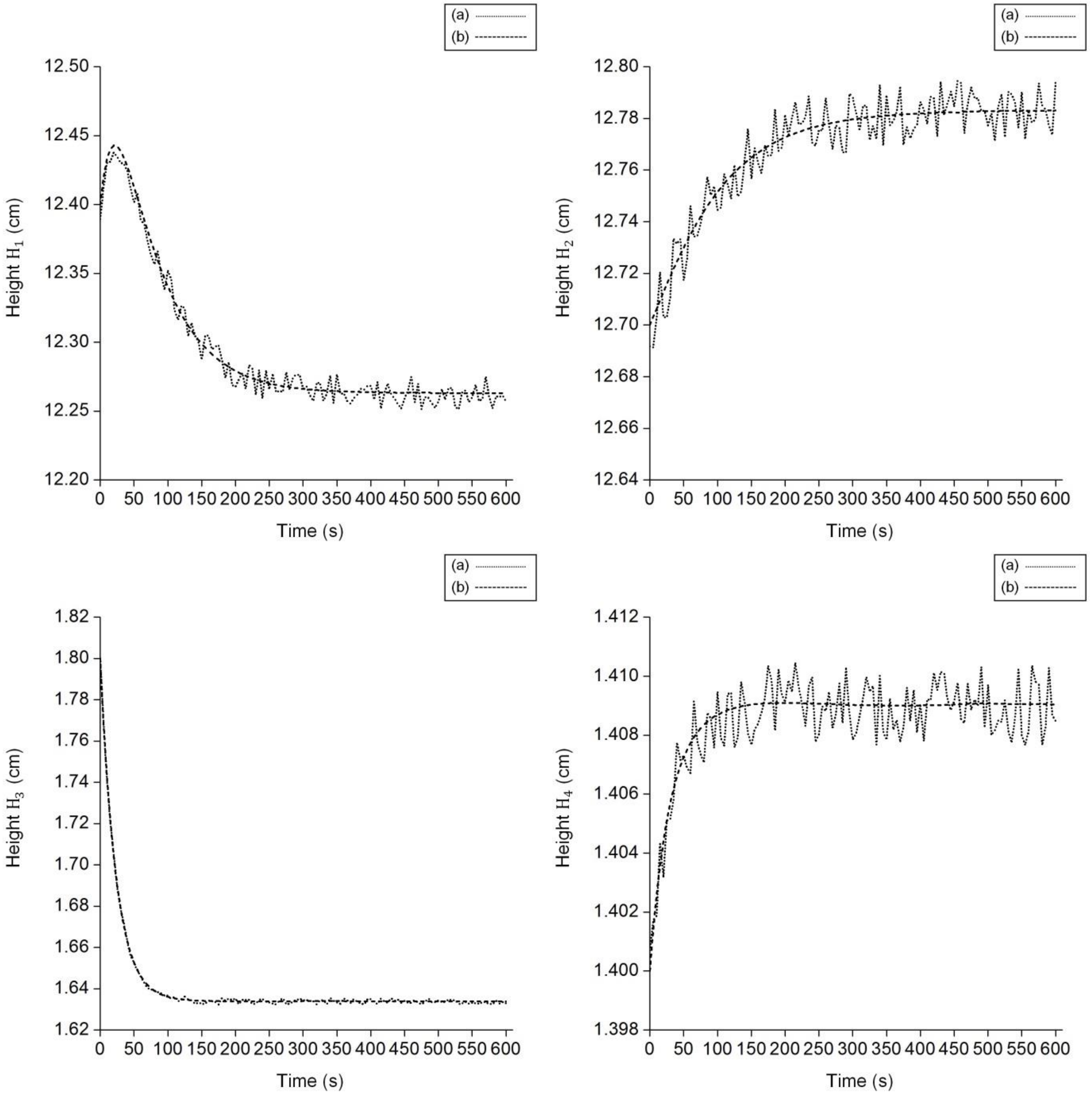

Figure 15. (a) $\widehat{H}_i$, measured value of water level (b) Model predicted value of state variable, H_i

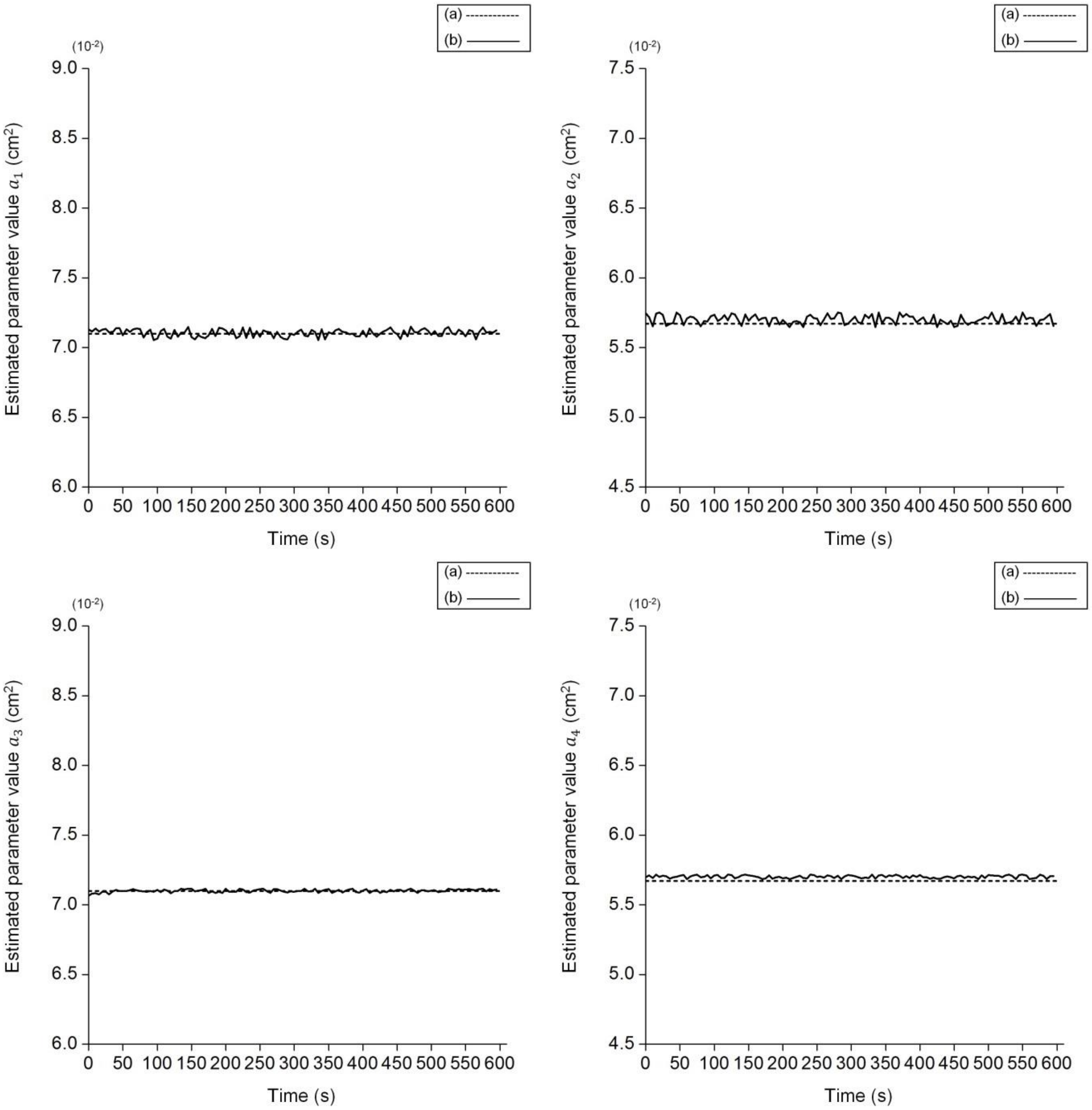

Figure 16. Model parameter values: (a) $\hat{a}_i$, true value of a_i (b) a_i, estimated model parameter value of a_i

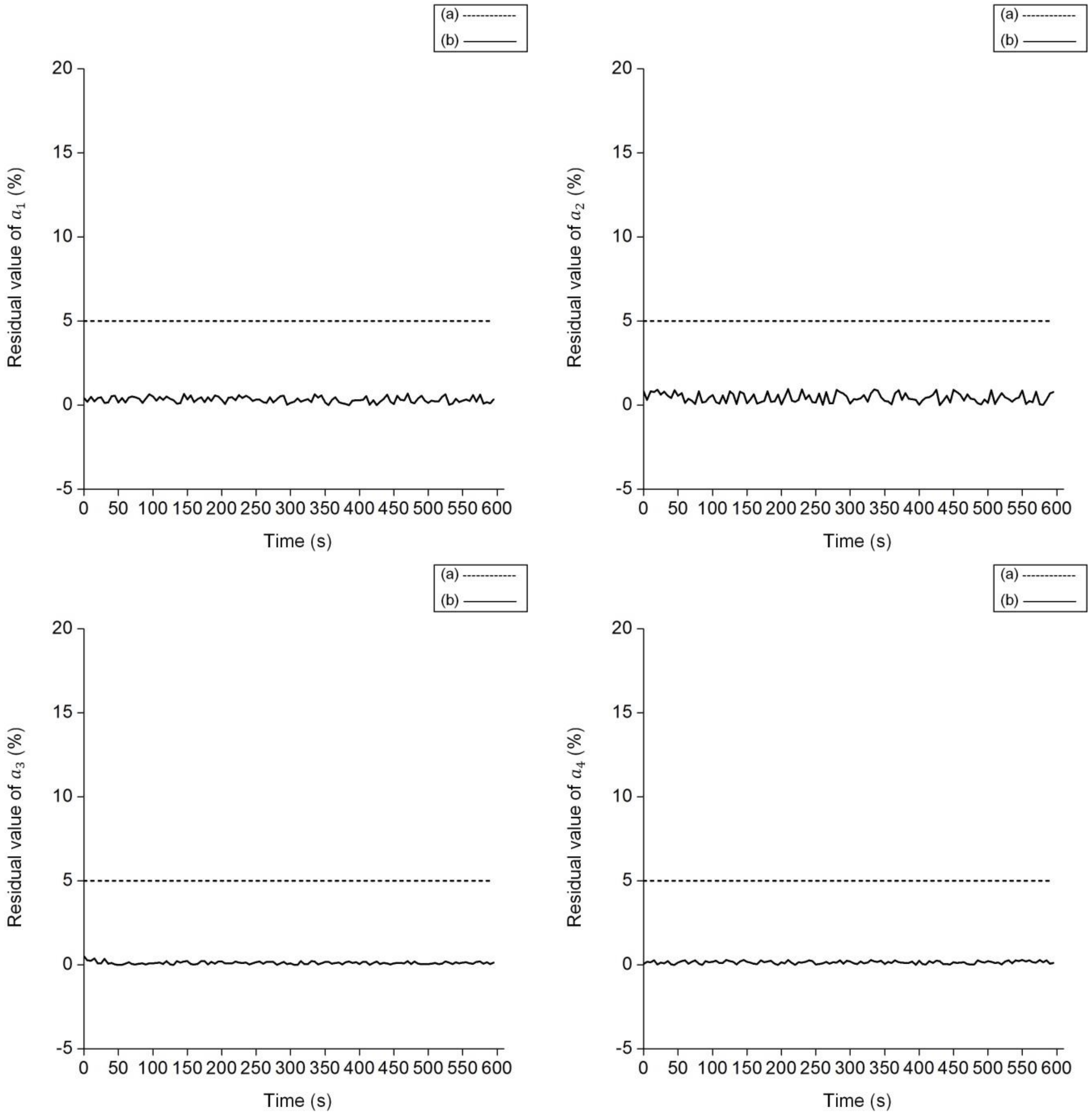

Figure 17. (a) Threshold value (b) Residual evaluation of estimated model parameters for a_i

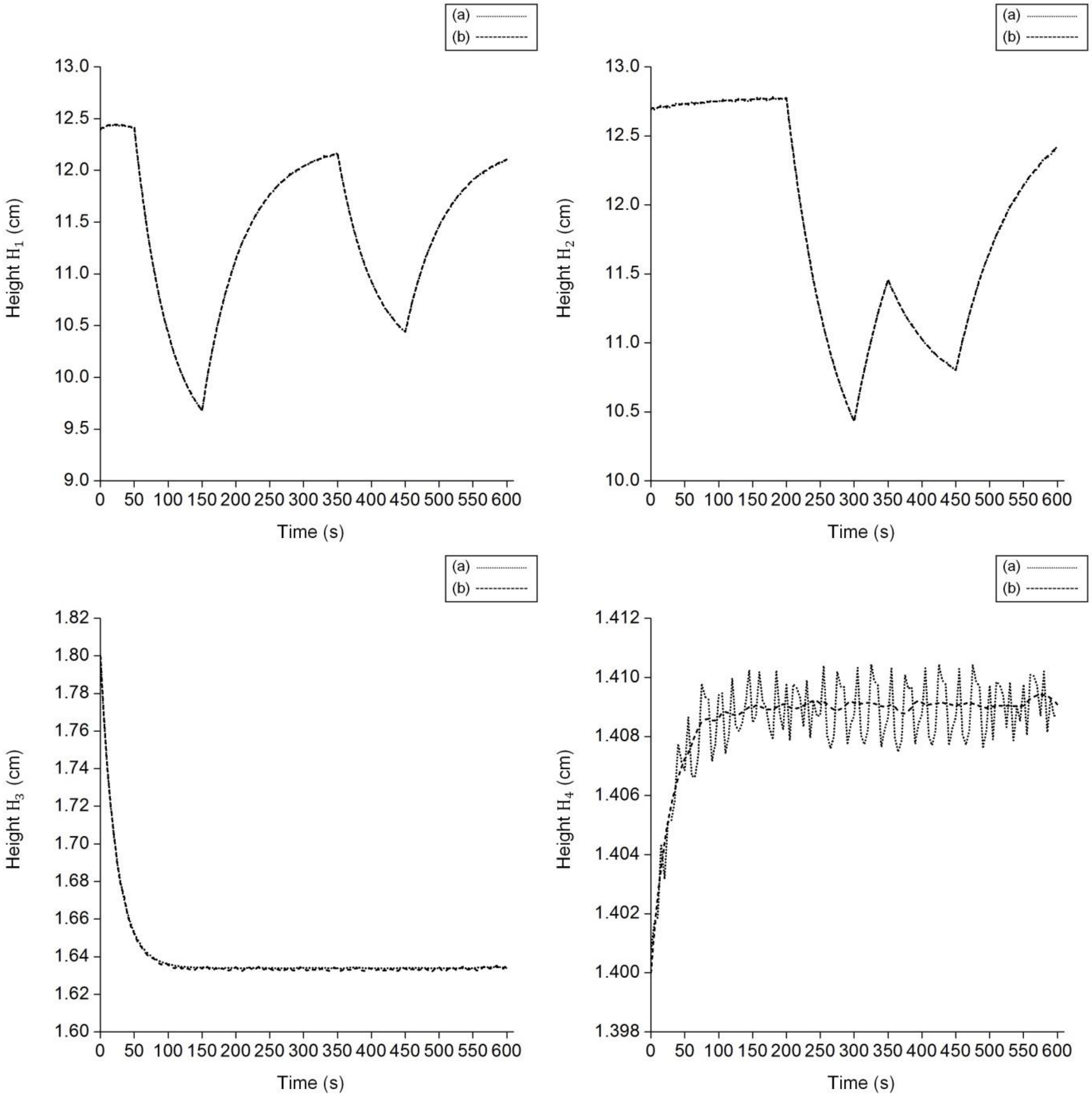

Figure 18. Faulty scenario (a) $\widehat{H}_i^F$, measured value of water level (b) Model predicted value of state variable, H_i^F

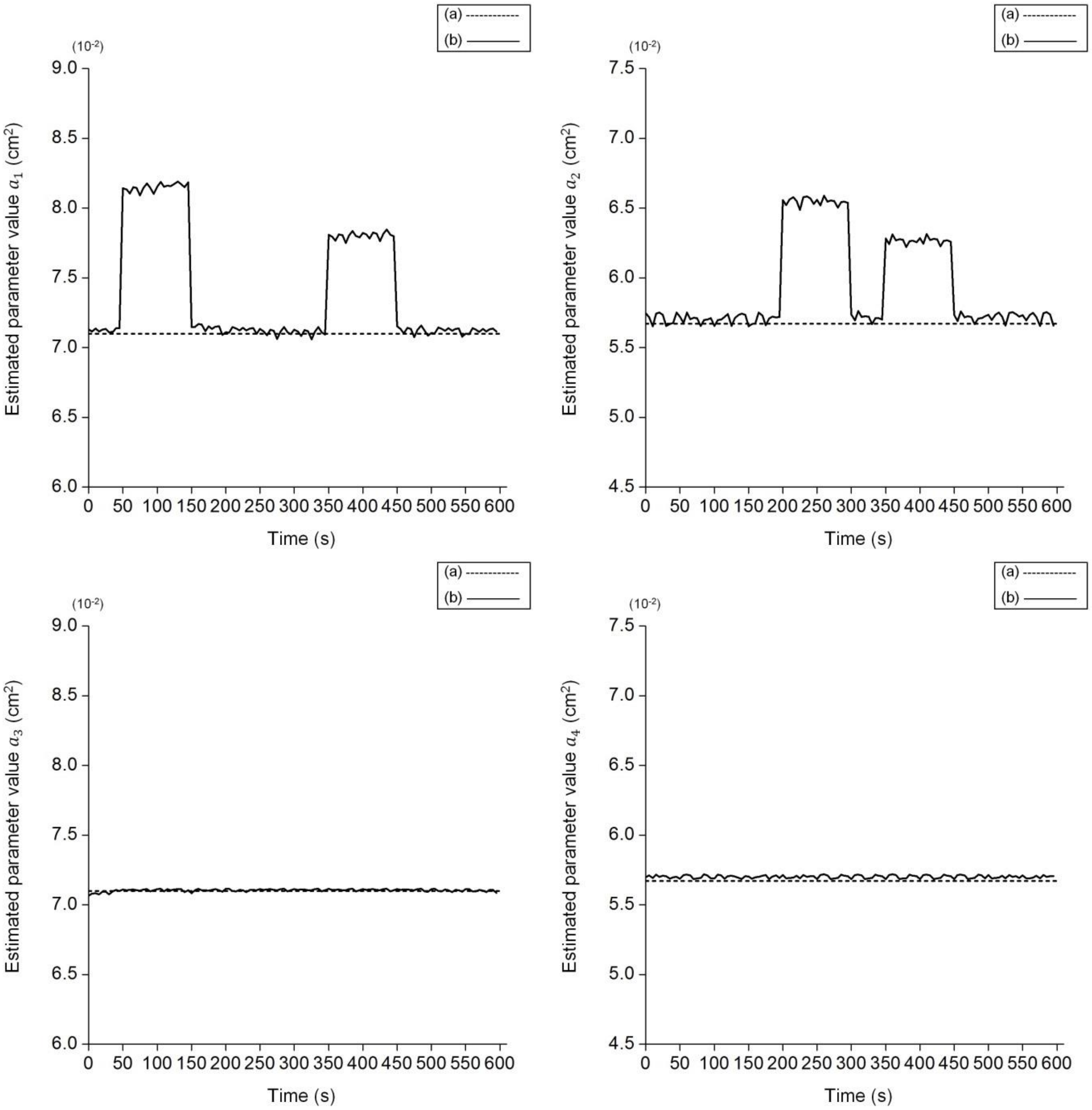

Figure 19. Model parameter values: (a) $\hat{a}_i$, true value of a_i (b) a_i^F, estimated model parameter value of a_i

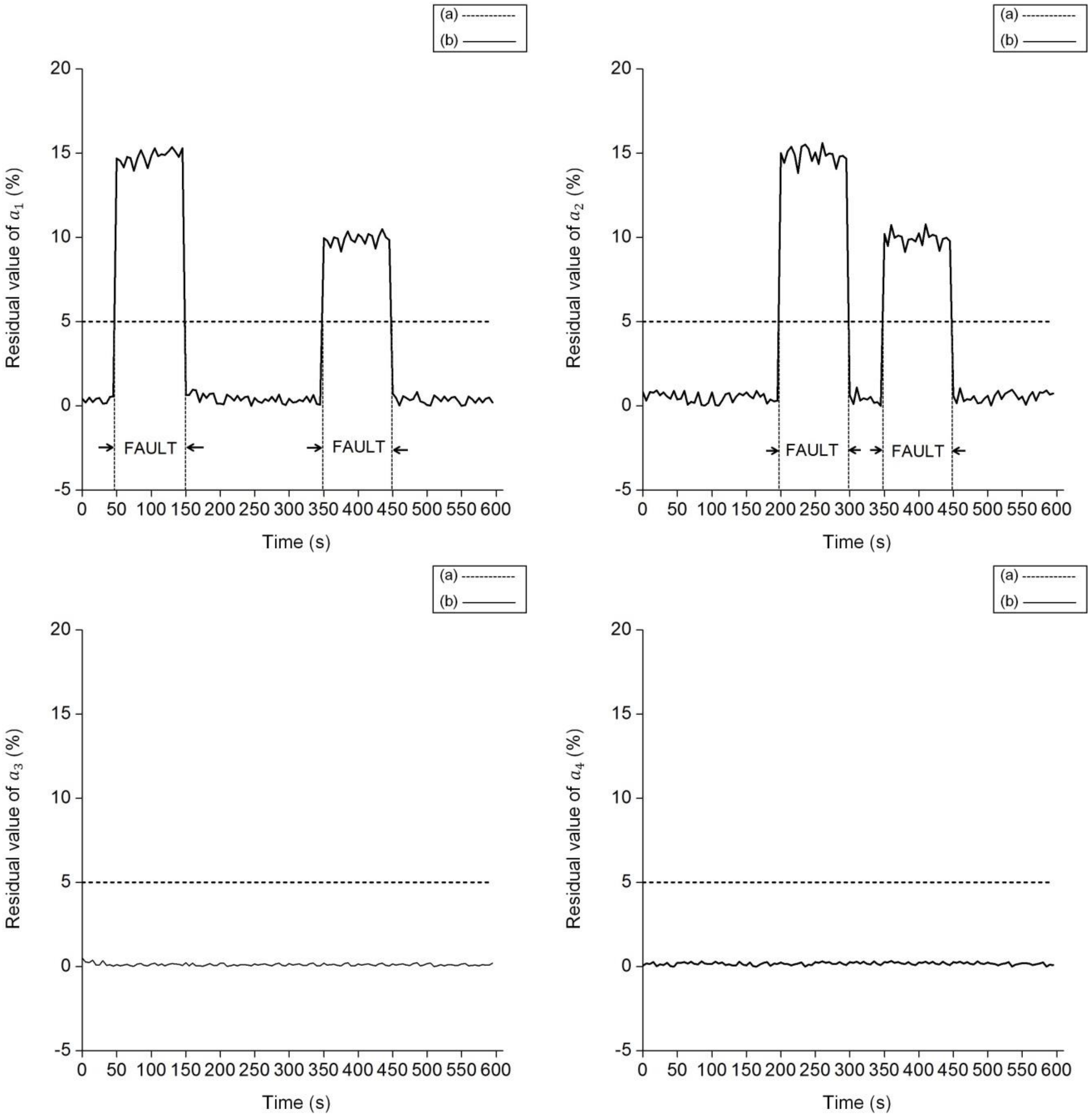

Figure 20. (a) Threshold value (b) Residual evaluation of estimated model parameters for a_i^F

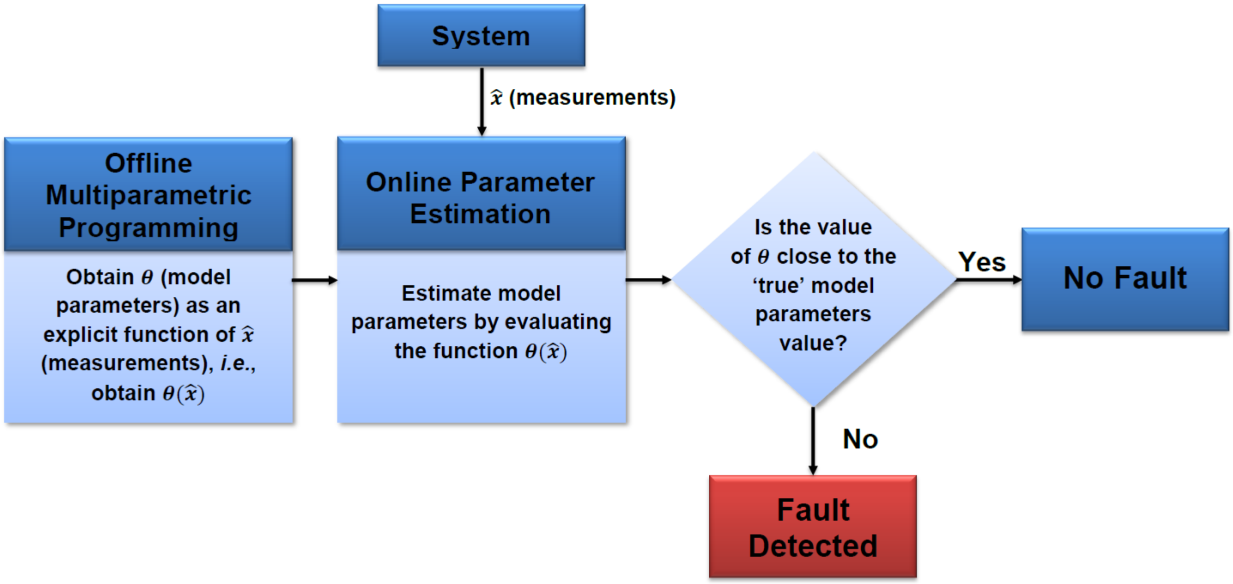
System
$\hat{x}$ (measurements)
Offline Multiparametric Programming
Obtain θ (model parameters) as an explicit function of $\hat{x}$ (measurements), *i.e.*, obtain $\theta(\hat{x})$
Online Parameter Estimation
Estimate model parameters by evaluating the function $\theta(\hat{x})$
Is the value of θ close to the 'true' model parameters value?
Yes
No Fault
No
Fault Detected

Figure 21. For Table of Contents Only