This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Kettunen, Markus; Zhang, P.; Jämsä-Jounela, Sirkka-Liisa

An embedded fault detection, isolation and accommodation system in a model predictive controller for an industrial benchmark process

Published in:
Computers and Chemical Engineering

Published: 01/01/2008

Document Version
Peer reviewed version

Please cite the original version:
Kettunen, M., Zhang, P., & Jämsä-Jounela, S-L. (2008). An embedded fault detection, isolation and accommodation system in a model predictive controller for an industrial benchmark process. *Computers and Chemical Engineering*, 32(12), 2966-2985.

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

An embedded fault detection, isolation and accommodation system in a model predictive controller for an industrial benchmark process

Markus Kettunen^a, Ping Zhang^b, Sirkka-Liisa Jämsä-Jounela^a

^a Helsinki University of Technology, Laboratory of Process Control and Automation, Kemistintie 1, FI-02150 Espoo, Finland

^b University of Duisburg-Essen, Institute for Automatic Control and Complex Systems, Bismarckstr. 81 BB, 47057 Duisburg, Germany

Abstract

Fault detection and isolation (FDI) for industrial processes has been actively studied during the last decades. Traditionally, the most widely implemented FDI methods have been based on model-based approaches. In modern process industry, however, there is a demand for data-based methods due to the complexity and limited availability of the mechanistic models. The aim of this paper is to present a data-based, fault tolerant control (FTC) system for a simulated industrial benchmark process, Shell control problem. Data-based FDI systems, employing principal component analysis (PCA), partial least squares (PLS) and subspace model identification (SMI) are presented for achieving fault tolerance in cooperation with controllers. The effectiveness of the methods is tested by introducing faults in simulated process measurements. The process is controlled by using model predictive control (MPC). To compare the effectiveness of the MPC, the FTC system is also tested with a control strategy based on a set of PI controllers.

© 2008 Elsevier Ltd. All rights reserved.

Keywords

FTC, PCA, PLS, SMI, MPC, Shell control problem

1 Introduction

During the last decades, applications of fault detection and isolation (FDI), as well as MPC, have been the most studied areas in the control field, especially in the process industries. Some applications of FDI connected to active fault tolerant control (FTC) strategies have recently been reported (Patwardhan, Manuja, Narasimhan, & Shah, 2006; Prakash, Patwardhan, & Narasimhan, 2002; Pranatyasto & Qin, 2001). Interest in integrating FDI and MPC has also increased. Traditionally, the most widely implemented FDI methods have been based on model-based approaches. In the modern process industries, however, there has been an increase in the demand for data-based methods due to the complexity and limited availability of mechanistic models.

Fault diagnosis is generally considered to be one of the most important actions in process plant supervision. Fault diagnosis consists of fault detection, isolation and identification components (Frank, Ding, & Marcu, 2000) and, when a fault tolerant control component is added, these are considered to be the main functions of a supervisory fault tolerant process (McAvoy et al., 2004). Without a proper fault detection, isolation and accommodation, the process is vulnerable to variable faults (Frank et al., 2000), which may easily render the process unprofitable, unstable and even unusable. Therefore fault detection plays a crucial role in active fault tolerant control: without proper detection of the fault, an active fault tolerant control strategy

cannot be activated and the fault cannot be accommodated. Despite the importance of fault diagnosis, fault detection and identification are still among the remaining tasks which a human operator must usually accomplish manually (Venkatasubramanian, Rengaswamy, & Kavuri, 2003a). The need for automated FDI is further emphasized by the finding that approximately 70% of industrial accidents are caused by human errors (Venkatasubramanian et al., 2003a). As a result, financial and safety reasons have been the major driving forces for continuous development in this research area.

Isolation of a fault is the second important part of fault diagnosis. Without knowledge of the fault location, accommodation of the fault cannot be accomplished. Generally, faults can be divided into the following categories: (1) structural changes in the process, (2) actuator faults, (3) sensor faults, (4) gross parameter changes in a model, and (5) external faults (Frank et al., 2000; Venkatasubramanian et al., 2003a). Fault identification attempts to identify the fault type, the magnitude of the fault, and the direction of the fault. Without proper fault identification, it is impossible to adjust the controller to counter the effects of the faults.

FDI has an important role in a fault tolerant control system. The FDI system has to successfully detect and identify different faults so that the controller can react and adapt to faulty situations. Several different FDI methods have been developed over the years, and Venkatasubramanian et al. (2003a) classified FDI methods into two categories: (1) methods based on first principle models, and (2) process history-based methods. These categories also have qualitative and quantitative method sub-categories. A number of FDI methods have recently been presented and reviewed in articles by Venkatasubramanian et al. (2003a), Venkatasubramanian, Rengaswamy, and Kavuri (2003b) and Venkatasubramanian, Rengaswamy, and Kavuri (2003c).

The popularity of the process history-based methods over the first principle models has been increasing because the modern processes are becoming more complex and therefore increasingly more challenging to model. The performance of the model-based methods in an industrial environment is also not as good as that with statistical methods. The downside with statistical methods is that an excessive amount of process data is required to make the process history-based methods work efficiently. The most popular process history-based FDI methods in current use are based on principal component analysis (PCA), as presented by Jackson (1979), and partial least squares (PLS) by Gerlach, Kowalski, and Wold (1979). The assumption in the basic versions of the methods is that the process behaviour is static. However, real processes usually have dynamical characteristics that often affect the accuracy of the static methods. To better adapt the methods for real systems, a number of variants of the basic methods have been developed: dynamic PCA by Ku, Storer, and Georgakis (1995), nonlinear PCA by Dong and McAvoy (1996), recursive PCA by Li, Yue, Valle-Cervantes, and Qin (2000) and recursive PLS by Qin (1998).

The dynamic behaviour of the process can be captured directly from the process history data by using identification methods like subspace model identification (SMI). During the last decade a number of different approaches have been proposed for SMI: canonical variate analysis (Larimore, 1990), N4SID (Van Overschee & deMoor, 1994), MOESP (Verhaegen, 1994), and simplified SMI (Hyötyniemi, 2001) and a PCA-based approach by Wang and Qin (2002).

The use of the statistical FDI methods has been studied intensively and many successful industrial applications have been reported. For instance, Dunia, Qin, Edgar, and McAvoy (1996), Chen, McAvoy, and Piovoso (1998) and Yoon and MacGregor (2004) demonstrated the use of statistical methods for fault detection and identification in the process industries. Recently also Komulainen, Sourander, and Jämsä-Jounela (2004) and Vermasvuori, Vatanski, and Jämsä-Jounela (2005) successfully used different statistical diagnostic methods for the FDI of an industrial dearomatisation process.

The aim of FTC is to maintain the system under control in a healthy state by using either passive or active approaches (Staroswiecki & Gehin, 2001). Traditionally FTC has been applied using passive FTC strategies with hardware redundancy, or by designing the control strategies in a robust way to tolerate faults to some degree. Active fault tolerant control strategies attempt to enhance the availability of a plant by integrating an FDI system in the FTC strategy. When using an active fault tolerant control strategy, a fault accommodation step is implemented in addition to fault detection, isolation and identification (Chiang, Russell, & Braatz, 2001).

MPC has firmly established its position in petroleum refineries, and the use of MPC has increased in chemical plants as well, as stated by McAvoy et al. (2004) in their milestone report. The earliest MPC formulations include model predictive heuristic control (MPHC) by Richalet, Rault, Testud, and Papon (1978), and dynamic matrix control (DMC) by Cutler and Ramaker (1980). MPC has been under constant development, and the focus in this area has recently been on solving the problems related to nonlinear process models. For instance, Venkateswarlu and Venkat Rao (2005) have been studying the use of an MPC equipped with a neural network model to control an unstable nonlinear process. Recent reviews on the MPC technology have been presented by Lee and Cooley (1997), Rawlings (2000) and Qin and Badgwell (2003).

Research on FTC systems based on MPCs has recently increased. For instance, Maciejowski (1999) encourages the use of MPC in FTC systems. Also, Pranatyasto and Qin (2001) have been studying a PCA-based FTC system with MPC controlling a simulated fluid catalytic cracking (FCC) unit. Prakash et al. (2002) have developed a fault tolerant control system (FTCS) based on the generalized likelihood ratio (GLR) and a standard MPC controller. This FTC system is then applied to a simulated, non-isothermal, continuously stirred tank reactor (CSTR) system. Since then, Patwardhan et al. (2006) have improved the system developed by Prakash et al. (2002) and applied it to a laboratory-scale continuously stirred tank heater (CSTH) system and to a simulated heavy oil fractionator process, Shell control problem (SCP).

The aim of this paper is to present a FTC system with embedded FDI algorithms for the control of the SCP in the presence of measurement faults. Three approaches are used for the FDI design: PCA-, PLS- and SMI-based systems. The control system is based on MPC. Fault compensation is carried out using the measurement replacement and the measurement reconstruction methods. A comparative study is carried out with a set of PI controllers to measure the effectiveness of the MPC. The novelty of this paper comes from the embedding of each of the closed-loop trained PCA, PLS and SMI monitoring methods in the MPC-based control system. Testing is carried out with a simulated MIMO process, with measurement faults and process noise.

This paper is organized as follows: in Section 2 the Shell control problem is introduced, Section 3 describes the control objectives of the process and the controllers, Section 4 describes the structure of the FTC scheme, Section 5 describes the FDI methods, Section 6 presents the simulation results with the proposed FTC system also in the presence of disturbances, and Section 7 ends the paper with the conclusions.

2 Process description of the heavy oil fractionator

In oil refineries heavy oil fractionators are used for initially fractionating crude oil into different product draws by cooling down the mixed-phase oil feed. Usually there are several fractionators in series to fractionate different products from different product draws. This kind of multi-input/multi-output (MIMO) system is particularly susceptible to measurement faults. Even small measurement errors can lead to huge financial losses and to process disturbances further on in the refining process.

At the Shell process control workshop Pretti and Morari (1987) presented the Shell control problem: a linear model of a heavy oil fractionator with a set of control objectives and constraints to act as a performance test for new control strategies. The simulated process includes one reactor section, four heat exchangers,

one side stripper, one product feed and three product draws. Hot, mixed-phase oil is fed to the unit and then cooled down using reflux flows located at the side of the fractionator. These reflux flows remove heat so that the separation procedure in the fractionator can be carried out. Product separation is based on the condensation properties of the products. The target process used in the study is described in Fig. 1.

Fig. 1. The Shell control problem by Prett and Morari (1987)

The linear, first-order transfer function model presented by Prett and Morari (1987) is shown in Table 1, and it has been reported to be able to satisfactorily describe the dynamical behaviour of the heavy oil fractionator. The model has been frequently implemented and used for successful control studies in the literature for instance by Yu, Lee, & Morari (1994), Camacho and Bordons (2000), Vlachos, Williams, and Gomm (2002) and Bordeau (2003).

The control constraints for the inputs, outputs and variable change rates are set according to the specifications given by Prett and Morari (1987) and are presented in Table 2.

Random noise is also added to the measurement signals in order to reflect the situation with the real process measurements. The magnitude of the noise varies between -0.025 and 0.025 , which is 5% of the range of the value of measurement y_1 .

Table 1

The model of the Shell control problem by Prett and Morari (1987)

	Top draw flow rate, u_1	Side draw flow rate, u_2	Bottom reflux head Transfer rate, u_3	Intermediate reflux heat transfer rate, l_2	Upper reflux heat transfer rate, l_1
Top end point, y_1	$\frac{4.05e^{-27s}}{50s+1}$	$\frac{1.77e^{-28s}}{60s+1}$	$\frac{5.88e^{-27s}}{50s+1}$	$\frac{1.20e^{-27s}}{45s+1}$	$\frac{1.44e^{-27s}}{40s+1}$
Side end point, y_2	$\frac{5.39e^{-18s}}{50s+1}$	$\frac{5.72e^{-27s}}{60s+1}$	$\frac{6.90e^{-15s}}{40s+1}$	$\frac{1.52e^{-15s}}{25s+1}$	$\frac{1.83e^{-15s}}{20s+1}$
Top temperature, y_3	$\frac{3.66e^{-2s}}{9s+1}$	$\frac{1.65e^{-14s}}{30s+1}$	$\frac{5.53e^{-2s}}{40s+1}$	$\frac{1.16e^{-0s}}{11s+1}$	$\frac{1.27e^{-0s}}{6s+1}$
Upper reflux temperature, y_4	$\frac{5.92e^{-11s}}{12s+1}$	$\frac{2.54e^{-12s}}{27s+1}$	$\frac{8.10e^{-2s}}{20s+1}$	$\frac{1.73e^{-0s}}{5s+1}$	$\frac{1.79e^{-0s}}{19s+1}$
Side draw temperature, y_5	$\frac{4.13e^{-5s}}{8s+1}$	$\frac{2.38e^{-7s}}{19s+1}$	$\frac{6.23e^{-2s}}{10s+1}$	$\frac{1.31e^{-0s}}{2s+1}$	$\frac{1.26e^{-0s}}{22s+1}$
Intermediate reflux temp., y_6	$\frac{4.06e^{-8s}}{13s+1}$	$\frac{4.18e^{-4s}}{33s+1}$	$\frac{6.53e^{-1s}}{9s+1}$	$\frac{1.19e^{-0s}}{19s+1}$	$\frac{1.17e^{-0s}}{24s+1}$
Bottom reflux temp., y_7	$\frac{4.38e^{-20s}}{33s+1}$	$\frac{4.42e^{-22s}}{44s+1}$	$\frac{7.20e^{-0s}}{19s+1}$	$\frac{1.14e^{-0s}}{27s+1}$	$\frac{1.26e^{-0s}}{32s+1}$

Table 2

The control constraints of the Shell control problem process

Variable	Lower limit	Upper limit
y_1	-0.5	0.5
y_2	-	-
y_7	-0.5	-
u_1, u_2, u_3	-0.5	0.5
$\Delta u_1, \Delta u_2, \Delta u_3$	-0.05	0.05

3 Control of the Shell heavy oil fractionator process

Two different control strategies are commonly used for the control of a heavy oil fractionator: an approach consisting of a set of PI controllers and an MIMO approach using a single MPC. In the following, these two control strategies are tested.

3.1 Control objectives

The control objective of the fractionator is to keep the top draw product end point y_1 , the side draw product end point y_2 and the bottom reflux temperature y_7 at the setpoint values by manipulating the top draw flow rate u_1 , the side draw flow rate u_2 and the heat transfer rate u_3 of the bottom reflux. The heat transfer rate u_3 is further adjusted using a control loop with the hot steam flow rate as a control variable. There are also two measured disturbances in the system: the heat transfer rate of the upper reflux l_1 and the intermediate reflux l_2 . These flows remove the heat from the system and are re-boiled in other sections of the plant.

3.2 Control strategy based on a set of PI controllers

First, the control structure for the PI-controlled system is selected by calculating the relative gain array (RGA) (Bristol, 1966) presented in Table 3.

Based on the calculated RGA matrix, the top draw product end point y_1 is recommended to be controlled with the top draw product flow rate u_1 , the top draw product end point y_2 with the side draw product flow rate u_2 , and the bottom reflux temperature y_7 with the bottom reflux heat transfer rate u_3 .

Table 3

The RGA matrix of the Shell heavy oil fractionator

	u_1	u_2	u_3
y_1	2,0757	-0,7289	-0,3468
y_2	3,4242	0,9343	-3,3585
y_3	-4,4999	0,7946	4,7053

After selecting the appropriate control structure, the control strategy is tested in the Matlab Simulink environment by adding the three PI controllers to the system, each controlling one of the controlled variables (CVs) y_1 , y_2 and y_7 . A low-pass filter is set for each controller input to damp the effect of the noise on the control performance.

The required PI parameters are determined by using the internal model control (IMC) tuning rules (Rivera, Morari, & Skogestad, 1986). The resulting values are presented in Table 4. Next, the PI-controlled system performance is evaluated by performing step tests. First, a setpoint change of 0.4 is introduced at time step $t = 100$ min to the setpoint of the top draw product end point y_1 . The response of the PI controller is slow but stable, as can be seen from Fig. 2. The variable y_1 reaches the given end point value 0.4 at time step $t = 550$ min. An effect on the other measurements is also seen.

Table 4

The PI controller parameters

	K	T_t
Controller 1	0.4	72
Controller 2	0.8	89
Controller 3	1.8	22

3.3 Control strategy based on the MPC

The MPC-based control strategy is next developed by using the Matlab MPC toolbox of Bemporad, Morari, and Ricker (2006). The real-time optimization of a cost function is carried out by using quadratic programming for solving the constrained optimization problem, and analytical methods for the unconstrained case. As some of the states of the process cannot be directly measured, a state estimator is required for the controller. The MPC toolbox uses a Kalman filter for the state estimation as a default, which is used for state estimation in this study.

The MPC parameters are adjusted according to the dynamics of the simulated process. The prediction horizon p is set long enough to be able to react to most situations occurring in the simulated process. Since the dead times in the process vary between 0 and 28 min and the time constants between 6 and 60 min, the prediction horizon is set to 120 min. The control horizon m is also set to a large value. But, because an increase in the control horizon also increases the calculation time, the length of the control horizon is set to 40min. The sample time with the process and with the MPC is 1min.

Fig. 2. The controlled and manipulated variables of the plant when the process is controlled by a set of PI controllers and a step change with a magnitude of 0.4 is introduced to a setpoint of y_1 at time step $t=100$ min

The weights of the controlled variables and the manipulated variables (MVs) are set in order to adjust the performance and behaviour of the MPC. Weights for the controlled variables y_1 , y_2 and y_7 are set to 45. The MV weights are set to 0.01. The weights for the MV rates are set to 1000 for Δu_1 , Δu_2 and Δu_3 to dampen the effect of the noise and sudden changes in the output values. These weight value settings provide more stable and reliable control actions than with the lower values. The MPC parameter values are summarized in Table 5.

Table 5

The MPC parameters for controlling the Shell heavy oil fractionator

Parameter	Values
Prediction horizon, p	120
Control horizon, m	40
Weights, CV [y_1 y_2 y_7]	[45 45 45]
Weights, MV	[0.01 0.01 0.01]
Weights, MV rates	[1000 1000 1000]

Finally, the control performance of the MPC-based system is tested. The results of the step response testing with the MPC, when a setpoint change of 0.4 is introduced at time step $t=100$ min to the setpoint of the top

draw product end point y_1 , are presented in Fig. 3. The performance of the MPC is good; the setpoint 0.4 for the top draw product end point is reached at time step $t = 264$ min. There are also some effects on the other variables, but the changes in the process are smooth and the variable values are quickly returned back to the setpoint values, as can be seen from Fig. 3.

Fig. 3. The controlled and manipulated variables of the plant when the process is controlled by the MPC and a step change with a magnitude of 0.4 is introduced to a setpoint of y_1 at time step $t=100$ min

3.4 Comparison of the control performance of the PI controllers and the MPC

A properly tuned MPC shows much better performance in terms of accuracy and response time than the control strategy based on PI controllers. The MPC is also able to reach the setpoint faster than the control with the PI controllers, as can be seen from Figs. 3 and 4. The other benefits of MPC versus PI control include, according to Camacho and Bordons (2000) and verified during our experiments, the ease of use and tuning procedure, the versatility of the method for a wide range of processes, built-in compensation for the dead times due to the process model, and intrinsic handling of multivariable control and the measured disturbances. The performance of the MPC is mainly affected by the accuracy of the model: the more accurate the model, the better the control performance of the system.

Fig. 4. The three different fault compensation approaches: measurement replacement, measurement signal reconstruction and reference trajectory matching

4 Structure of the FTC system

The proposed FTC system consists of three parts: the FDI-part for detecting the fault, the control part for controlling the process, and the supervisory part for carrying out the necessary actions to minimise the effects of the fault. Three different methods for minimizing the effects of the faults are studied: measurement replacement, measurement reconstruction and reference trajectory matching. The reference trajectory matching is only suitable for an MPC-based control system. In this study, only the measurement replacement and the measurement reconstruction strategies are implemented so that it is possible to compare the results between different control systems. All three approaches are presented in Fig. 4.

4.1 FTC structure based on measurement replacement

The first FTC structure is based on replacement of the measurement value provided by the model if the measurement is declared faulty. The model is acquired by applying the subspace identification methods to a training data set. The decision to use the subspace identified model output instead of the measurement is made in the supervisory unit. The performance of the method is dependent on the accuracy of the model which, in turn, is dependent on the quality of the training data. The structure of the FTC approach based on measurement replacement is shown in Fig. 4 with a solid line.

4.2 FTC structure based on measurement reconstruction

The second FTC structure is based on reconstruction of the measurement signal. In this method, the faulty signal is compensated with an approximate correction value derived using an iterative method based on the optimization of the squared prediction error (SPE) value. To be able to calculate an SPE, the PCA model is constructed using a training data set. The compensation itself is carried out in the supervisory unit. This approach is shown in Fig. 4 with a dashed line.

4.3 FTC structure based on reference trajectory matching

The third FTC structure is based on the reference trajectory matching. This approach is similar to measurement signal reconstruction with the exception that the reference trajectory of the MPC is changed using the correction value acquired through optimization of the SPE. The change in the reference trajectory

is carried out by the supervisory unit. The FTC strategy based on the reference trajectory matching is shown in Fig. 4 with a dotted line.

5 Fault detection and isolation

In this study three approaches are used for the FDI: an FDI system based on principal component analysis, partial least squares and subspace model identification. In addition to the FDI methods, a cumulative sum mechanism is implemented in the system in order to avoid false alarms. The cumulative sum system requires that the fault is detected at least five time steps before the fault is declared. After the five steps the fault compensation is started with gradually increasing compensation value reaching the final compensation value after five more time steps.

5.1 PCA

Fault detection with the PCA is based on the SPE detection threshold, Q_α , which is calculated using the equations of Jackson (1979), with the approximation that the probability distribution is normally distributed:

$$Q_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} + 1 \right]^{1/h_0} \quad (1)$$

where c_α is the normal density distribution corresponding to the upper $(1 - \alpha)$ percentile of the normal deviate, and θ_i is calculated using Eq. (2):

$$\theta_i = \sum_{j=k+1}^K \lambda_j^i, \quad i = 1, 2, 3 \quad (2)$$

where k is the number of selected principal components and K is the total number of principal components. h_0 is calculated with Eq. (3):

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \quad (3)$$

The Hotelling T_{limit}^2 calculation is based on Eq. (4):

$$T_{\text{limit}}^2 = \frac{k(N-1)}{N-k} F(k, N-k, \alpha) \quad (4)$$

where N is the number of observations, $F(k, N-k, \alpha)$ corresponds to the probability point on the F distribution with $(k, N-k)$ degrees of freedom, and α represents the user-defined confidence level.

For fault detection purposes, the new data are first processed using the standard procedure of Jackson (1991), after which the SPE index value and the Hotelling T^2 values are calculated. The SPE values are calculated using Eq. (5):

$$Q = (x_{\text{scaled}} - \hat{x})^T (x_{\text{scaled}} - \hat{x}) \quad (5)$$

where x_{scaled} contains the current autoscaled measurement values and $\hat{x}$ the predicted measurement values. The Hotelling T^2 values are calculated at each time step as follows:

$$T^2 = x_{\text{scaled}} \cdot P_k \cdot \Lambda_k^{-1} \cdot P_k^T \cdot x_{\text{scaled}}^T \quad (6)$$

where P_k contains the selected k first principal components and Λ_k is the matrix containing the k first eigenvalues.

When the SPE and Hotelling T^2 values are calculated, the values are compared to the SPE and Hotelling T^2 limits, respectively. If the SPE values are higher than the limit, then a fault is declared on the variable with the highest SPE score over the limit. The Hotelling T^2 values are only used for verification and comparison purposes, not for fault declaration itself.

The direction and the required fault compensation magnitudes are estimated using an iterative procedure based on minimization of the SPE value with a change to the measurement value.

5.2 PLS

For fault detection and isolation purposes with PLS, the PLS regression model is constructed with a training data set.

Fault detection is carried out using the root mean square of the prediction error (RMSEP) index:

$$\text{RMSEP} = \sqrt{\frac{\sum_{i=1}^m (\hat{y}_i - y_i)^2}{m}} \quad (7)$$

where m is the number of measurements.

The RMSEP values are calculated for each variable, and the variable with the highest RMSEP value over the limit is selected to be faulty. Fault magnitude estimation is carried out by calculating a residual between the PLS model output and the measurement value.

5.3 SMI

Subspace model identification is used in the study to identify a model for FDI. For instance, [Hyötyniemi \(2001\)](#) used the same SMI approach as is used in the study to identify the state-space matrices for the FDI.

The FDI with the SMI is carried out by calculating the residual between the predictions of the SMI model and the process measurements. If the absolute residual between the measurement and the predicted output is higher than the limit, then a fault is declared in that variable. The magnitude of the fault is estimated as the difference between the outputs of the model and the measurement.

6 Testing of the FTC system

In order to effectively compare the performances of the FTC systems, a testing of the FDI and FTC is carried out. First the PCA, PLS and SMI-based FDI is tested and, finally, the results of the combination of the FDI systems with the fault compensation strategies are compared.

6.1 Analyzer faults and faulty testing data

Two different kinds of fault are common in the oil refining process analyzers and sensors: abrupt bias faults and slowly increasing or decreasing drift faults. The bias faults are usually caused by contamination of the

analyzer sample, while the drift faults can be caused by the slow accumulation of substances in the sensors, analyzers or sample lines.

The data set used for testing consists of 800min of the simulated process data including the measurement faults. The bias and drift faults are introduced into the simulated process measurements. In the test setting, a positive bias fault with a magnitude of 0.5 is introduced into the top draw product quality variable y_1 at the time step 100 min, and the fault is ended at the time step 300 min. Another fault, a positive drift fault, is introduced into the top draw product quality variable y_1 at the time step 100 min, and the fault is ended at the time step 300min at which the fault magnitude is 0.5. The fault magnitude 0.5 is the maximum hard constraint allowed for the top draw product end point y_1 .

6.2 Testing the FDI

First, a set of training data is created using the original process model for training the FDI methods. Next, the trained PCA-, PLS and SMI-based FDI methods are tested and the results recorded.

6.2.1 Training the FDI methods

Training the PCA-, PLS- and SMI-based FDI methods is carried out with the closed-loop non-faulty training data set, with the disturbance variables upper reflux heat duty, l_1 and intermediate reflux heat duty l_2 varying between -0.5 and 0.5 . This will excite the input and output variables of the process and allow the FDI methods to capture the normal behaviour of the process. The controlled variables are the top draw product end point y_1 , the side draw product end point y_2 and the temperature measurement y_7 . The manipulated variables are the top draw flow rate u_1 , the side draw flow rate u_2 and the bottom reflux heat transfer rate u_3 .

6.2.1.1 Principal component analysis

The principal component analysis is carried out with three separate PCA models. Each of the PCA model contains delay-compensated input variables and a controlled variable. The model structures are $\text{PCA1} = [y_1 u_1 u_2 u_3 l_1 l_2]^T$, $\text{PCA2} = [y_2 u_1 u_2 u_3 l_1 l_2]^T$ and $\text{PCA7} = [y_7 u_1 u_2 u_3 l_1 l_2]^T$. This way the PCA models are able to take the process disturbances into account while detecting faults in the system. For each PCA model, three principal components are used with 84.1, 96.5 and 83.8% variance captured. For FDI purposes, squared prediction error and Hotelling T^2 limits are calculated using a 95% confidence for both. The SPE and Hotelling T^2 limits are calculated using the procedure presented in the previous section.

6.2.1.2 Partial least squares

The PLS is trained using the same data as with the PCA. Three different PLS models are used for FDI: PLS1 contains delay-compensated measured disturbances l_1 and l_2 and three control inputs u_1 , u_2 and u_3 as input variables and the current y_1 as an output variable, PLS2 contains the same input variables with the correct delay-compensation and y_2 as an output variable and finally PLS7 containing the same input variables with the correct delay-compensation and y_7 as an output variable. The PLS latent variables (LVs) are calculated using the NIPALS algorithm of [Wold \(1975\)](#). In the final models PLS1, PLS2 and PLS7 there are three latent variables which captured 79.7, 94.6 and 80.3% of the input variance and 35.8, 83.1 and 31.3% of the output variance, respectively.

6.2.1.3 Subspace model identification

The identified subspace model is trained using the same training data as with the PLS. The inputs for the subspace identified system are the two measured disturbances l_1 and l_2 and three control inputs u_1 , u_2 and u_3 . The outputs are the seven outputs, y_1 , y_2 , y_3 , y_4 , y_5 , y_6 and y_7 . When creating the state-space models, the order of the model is reduced from a 35th order to a 10th order model for optimization purposes.

6.2.2 Results of the PCA-based FDI

The PCA-based FDI consists of the detection and isolation part and the identification part. The SPE index is used for detecting and isolating the faults in the controlled variables, if one SPE value is higher than the SPE limit, a fault is declared. An iterative method based on the SPE value is used for fault identification. In addition to the SPE index, Hotelling T^2 index is used for reference and verification purposes.

In the first FDI testing the PCA-based FDI is able to detect both the bias- and drift-shaped faults. For the bias fault, both the SPE and the Hotelling T^2 detect the fault at time step $t = 105$ min, i.e. 5min later than the fault has entered the system. The drift fault is detected by the SPE at time step $t = 143$ min with the PI controller-based control strategy, and at time step $t = 138$ min with the MPC-based control strategy. With the Hotelling T^2 the fault is not detected until at time step $t = 232$ min. Based on the results, the SPE has a significantly faster detection rate than with the Hotelling T^2 when using the same confidence levels for both of the methods. The SPE index and the Hotelling T^2 indices are presented in Fig. 5 for bias fault and in Fig. 6 for the drift fault.

Fig. 5. The Hotelling T^2 and the SPE indices and limits for the bias-shaped fault in the measurement y_1

Fig. 6. The Hotelling T^2 and the SPE indices and limits for the drift-shaped fault in the measurement y_1

The fault isolation is based on the largest SPE value higher than the detection limit. If more than one SPE value is higher than the detection limit, the one with the highest SPE value is selected to be the faulty one.

Finally, the fault identification is carried out at each time step by using an iterative procedure based on minimisation of the SPE value. In the procedure, the number of required iteration depends on the first iteration step size and the iteration limit. In the study, the starting step is set to 0.08 and the residual limit for ending the iteration is set to 5×10^{-6} . The magnitude of the faulty measurement is changed during each iteration step and a new SPE value is calculated and compared to the SPE value of the original measurement. This difference is then compared to the difference acquired during the $k - 1$ iteration step, where k is a current iteration step. If the difference between these two is smaller than the stopping criteria the iteration is ended and the approximated fault magnitude is acquired. In essence, the stopping criteria corresponds to the accuracy of the fault estimation: the smaller the criteria, more accurate the estimation, but longer iteration time. The limit used in the study is small enough to achieve as accurate values for the correction as possible with as short iteration time as possible. y_1 iteration steps for the Δy and the SPE score for each iteration at time step $t = 240$ min are presented as an example case of a bias fault in the measurement in Figs. 7 and 8, respectively.

Fig. 7. Iteration steps for the Δy_1 at the time step $t=240$ min when a bias fault is present in the output y_1

Fig. 8. Iterations steps for the SPE score at the time step $t=240$ min when a bias fault is present in the output y_1

6.2.3 Results of the PLS-based FDI

The PLS-based FDI has the RMSEP as a fault detection index. This index measures the residual between the model outputs and the measurements.

First, a bias fault is affecting the top draw distillation end point measurement at time step $t = 100$. The limit for detecting the faults is set to 2.5, which is clearly above the noise level of the process. The bias fault is detected and isolated at time step $t = 105$ min, only after a delay caused by the cumulative sum algorithm. The faulty variable is isolated from the RMSEP plots, the highest value being selected as the faulty one. Fault magnitude is determined by comparing the PLS model predictions to the current value of Y . The difference of the predicted and measured Y is selected as the fault magnitude. Next, the drift fault affecting the measurement y_1 is detected at time step $t = 139$ min with the PI controller-based control strategy, and at time step $t = 138$ min with the MPC-based control strategy. The values of the RMSEP index can be seen in Fig. 9 for the bias fault and in Fig. 10 for the drift fault.

Fig. 9. The RMSEP values for the measurements in the case of a bias fault in the output y_1

Fig. 10. The RMSEP values for the measurements in the case of a drift fault in output y_1

6.2.4 Results of the SMI-based FDI

In the case of the SMI-based FDI, the faults are detected by comparing the SMI model outputs with the measurement outputs. If the residual between the measurements and the SMI model outputs value is higher than a fault threshold, then a fault is detected and isolated to that specific measurement. The fault threshold used in the study is set to 0.07 to be high enough to exceed the noise level of the process, but also to be as low as possible to detect the faults as soon as possible. The FDI system is able to detect the faults in the measurements successfully; the bias fault is detected at time step $t = 105$ min with both the MPC and PI controller-based control strategies. The drift fault is detected after a delay of 28 min, at time step $t = 128$ min when using either one of the control strategies. These results are presented in Fig. 11 for the bias fault, and in Fig. 12 for the drift fault.

Fig. 11. The SMI residual values and limit for the measurement y_1 in the case of a bias fault in output y_1

Fig. 12. The SMI residual values and limit for the measurement y_1 in the case of a drift fault in output y_1

6.3 Testing the fault compensation methods under the PI- or model predictive control

After successful fault detection, isolation and fault magnitude identification, a fault is compensated. The measurement reconstruction with the PCA- and PLS-based FDI systems and measurement replacement with the SMI-based FDI system are tested with a fault in the top draw product end point y_1 under the PI controller-based control strategy and model predictive control. Only the results for the measurement reconstruction and the measurement replacement strategies are presented, since the results and the performance of the FTC system with the reference trajectory matching are identical to the results obtained with the measurement reconstruction strategy.

6.3.1 Measurement reconstruction with the PCA

First, the measurement reconstruction-based fault compensation system with the PCA-based FDI is tested with the PI controller-based control strategy. The fault is detected at time step $t = 105$ min. The system is able to detect and counter the fault, causing only a minor effect on the other measurements owing to the small estimation error with the fault magnitude iteration. The results are presented in Fig. 13.

Fig. 13. Performance of measurement reconstruction with the PCA-based FDI using a PI controller-based control strategy in the case of a bias fault in y_1

Second, the PCA-based FDI with the measurement reconstruction system is tested under the control of the MPC. As with the PI controllers, the MPC-based system is able to detect the bias fault in the process measurement y_1 at time step $t = 105$ min. The fault is quickly compensated and only a small disturbance is caused in the other measurements, as can be seen in Fig. 14.

Third, the drift fault is introduced into the process output y_1 and the process controlled with the PI controller-based control strategy. This time the fault is detected and compensated at time step $t=135$ and 35 min after the fault had started to affect the measurement value. The system is able to compensate for the fault and is working efficiently, as can be observed in Fig. 15.

Finally, the drift fault is present in measurement y_1 while using the PCA-based FDI and under model predictive control. With these settings the fault is detected and compensated at time step $t=138$ and 38 min after the fault had entered the system. These results can be verified from Fig. 16.

In all of the cases with PCA-based FDI the fault is successfully compensated and the effect on the measurements is less than 1% of the maximum fault magnitude.

6.3.2 Measurement reconstruction with the PLS

Next, the measurement reconstruction compensation strategy is implemented with the PLS-based FDI. First, a bias fault is affecting the top draw product end point y_1 and the system is controlled with a set of PI controllers. The system is able to detect and counter the effects of the fault at time step $t = 105$ min, as can be seen in Fig. 17.

Fig. 14. Performance of measurement reconstruction with the PCA-based FDI using a model predictive controller in the case of a bias fault in y_1

Fig. 15. Performance of measurement reconstruction with the PCA-based FDI using a PI controller-based control strategy in the case of a drift fault in y_1

Fig. 16. Performance of measurement reconstruction with the PCA-based FDI using a model predictive controller in the case of a drift fault in y_1

Fig. 17. Performance of measurement reconstruction with the PLS-based FDI using a PI controller-based control strategy in the case of a bias fault in y_1

The MPC-based control strategy is then applied to the FTC system when the bias fault is affecting the measurement y_1 . Again, the fault is detected and compensated at time step $t = 105$ min and the fault has almost no effect on the measurements. The small spike in y_1 is caused by the delay in fault detection, which is implemented in order to eliminate the effect of random spikes in the measurements. The response of the system can be seen in Fig. 18.

Fig. 18. Performance of measurement reconstruction with the PLS-based FDI using a model predictive controller in the case of a bias fault in y_1

Third, the system is tested with the drift fault affecting the y_1 . Now the system is controlled by using the PI controller-based control strategy. The results are good and the system is able to detect and counter the effects of the fault at time step $t = 139$ min, with a delay of 39min, as is evident from Fig. 19.

Finally, the system is tested with the MPC as a control strategy and the drift fault affecting the measurement y_1 . The fault in the measurement is detected and the effects of the fault are countered at time step $t = 138$ min. The effect on the measurement value is small and the system is successfully kept under control. The performance of the FTC system can be seen in Fig. 20.

Fig. 19. Performance of the measurement reconstruction with the PLS-based FDI using a PI controller-based control strategy in the case of a drift fault in y_1

Fig. 20. Performance of the measurement reconstruction with the PLS-based FDI using a model predictive controller in the case of a drift fault in y_1

In all of the cases with PLS-based FDI, the fault is successfully compensated and the effect on the measurements is less than 10% of the maximum fault magnitude. The system with MPC as control strategy is slightly faster and more stable than the system based on PI controller-based control strategy.

6.3.3 Measurement replacement with the SMI

Finally, the fault compensation method based on measurement replacement is tested with the SMI-based FDI. The SMI model output is used instead of the measurement value, when a fault is present in one of the measurements. During testing the fault is affecting the top draw product end point y_1 . After the cessation of the fault, the process measurement is used instead of the model output and the process is returned back to the normal state after the delay caused by the cumulative sum algorithm. Both the PI controllers and the MPC are used for controlling the process with the bias or the drift fault.

First, the measurement replacement algorithm is implemented with the PCA-based FDI and the PI controller-based control strategy. With the bias fault in y_1 , the fault is detected at time step $t = 105$ min. There is no further effect on the measurement values and the fault compensation is successful. The results are shown in Fig. 21.

Fig. 21. Performance of measurement reconstruction with the SMI-based FDI using a PI controller-based control strategy in the case of a bias fault in y_1

Second, as with the control strategy based on the PI controllers, a bias fault in y_1 is compensated at time step $t = 105$ min when the process is under control of the MPC. The FTC combination of the measurement

replacement compensation and the SMI-based FDI is functioning efficiently and there are no effects on the measurements due to the fault compensation, as is clearly seen in Fig. 22.

Fig. 22. Performance of measurement replacement with the SMI-based FDI using a model predictive controller in the case of a bias fault in y_1

Next, a drift fault is introduced into the process output y_1 and the system is under PI controller-based control strategy. This time the SMI-based FDI and the measurement replacement compensation strategy are able to detect and counter the fault at time step $t = 133$ and 33 min after the fault had first appeared in the measurement. The effect on the other measurements caused by the FTC is small; there is less than 1% disturbance in the output values. The response of the system is presented in Fig. 23.

Finally, with the FTC system under control by an MPC, the drift fault in the top draw end point y_1 is detected and compensated at time step $t = 137$ and 37 min after the fault has entered the system. The results are shown in Fig. 24.

With the SMI-based FDI, the results show that the system is able to detect and counter the introduced faults rapidly and effectively: less than 1% of the maximum fault magnitude disturbance is caused in the measurements. Both the MPC-and PI controller-based control strategies performed equally this time.

Fig. 23. Performance of measurement reconstruction with the SMI-based FDI using a PI controller-based control strategy in the case of a bias fault in y_1

Fig. 24. Performance of measurement replacement with the SMI-based FDI using a model predictive controller in the case of a drift fault in y_1

6.4 Testing of the effects of disturbances to the control system

In this section disturbance rejection capability of the controller is tested. The two measured disturbances, l_1 and l_2 are manipulated and the effect of the disturbances to the controlled variables is observed. As stated in the Shell control problem description, the measured disturbances l_1 and l_2 vary between $[-0.5 \ 0.5]$; so if the system can survive the effect of these extreme values, the values in between will not cause any problems during normal operation. The disturbance patterns used by Vlachos et al. (2002) are utilized and the results are presented and discussed. The disturbance patterns used by Vlachos et al. (2002) are $d_1 = [0.5 \ 0.5]^T$, $d_2 = [-0.5 \ -0.5]^T$, $d_3 = [0.5 \ -0.5]^T$ and $d_4 = [-0.5 \ 0.5]^T$ for disturbances $[l_1 \ l_2]^T$. The disturbances appear in the system at time $t = 25\text{min}$ and the final values are reached at time $t = 50\text{min}$. The disturbance patterns are presented in Fig. 25. The effect of the disturbance patterns d_1 , d_2 , d_3 and d_4 is presented in Figs. 26–29, respectively.

Fig. 25. The disturbance patterns used in the study to test the disturbance tolerance of the FTC system

As can be seen from the figures, the worst case scenario, with both disturbance values being the same sign and at their maximum value (patterns d_1 and d_2), will cause the manipulated variable u_1 to approach its minimum or maximum constraint value. Nevertheless, the MPC is able to retain all three CVs close to their target trajectories. With the both extreme cases the value of y_1 is kept within 0.05 of the target trajectory, which is 10% of the maximum and minimum hard constraints for the top draw distillation end point value y_1 . The y_2 is kept within 0.15 of the target trajectory and y_7 within 0.05 of the target trajectory.

Finally, disturbance rejection capability of the system in presence of a fault is tested with the PLS-based FTC system with measurement signal reconstruction. The disturbance pattern d_2 is tested with an upward, drift-shaped fault with a final magnitude of 0.5 that is introduced to the top draw distillation end point analyzer y_1 . The RMSEP values are presented in Fig. 30 and the results of the experiment are presented in Fig. 31. As can be seen from the figures, the system is able to compensate for the fault and the overall performance of the system is very good despite the disturbances and a fault present in the system. Overall, the system handles the disturbances well even in the presence of a fault, and is able to smoothly keep the controlled variables close to the desired setpoint values.

Fig. 26. The effect of the disturbance pattern d_1 ($[0.5 \ 0.5]$) to the CVs and MVs

Fig. 27. The effect of the disturbance pattern d_2 ($[-0.5 \ -0.5]$) to the CVs and MVs

Fig. 28. The effect of the disturbance pattern d_3 ([0.5 -0.5]) to the CVs and MVs

Fig. 29. The effect of the disturbance pattern d_4 ([-0.5 0.5]) to the CVs and MVs

Fig. 30. The RMSEP values and limit for the drift-shaped fault in the measurement y_1 while the disturbance pattern d_2 $([-0.5 -0.5])$ is affecting the system

Fig. 31. The performance of the PLS-based FTC system with an upward drift-shaped fault in the measurement y_1 and the disturbance pattern d_2 $([-0.5 -0.5])$ is affecting the system

6.5 Analysis of the results and discussion

As heavy oil fractionators are located in crucial positions in the refineries, fault-free operation is essential in order to ensure a reliable supply of raw materials to other parts of the plant.

Based on the results, the performance of the tested FTC systems is good; the maximum deviation between the faulty and the compensated measurement values being less than 10% of the maximum of the fault magnitude in all cases. All three tested FTC systems based on PCA, PLS and SMI are able to effectively detect, isolate and compensate the faults introduced into the process measurements with variable detection times. The FTC system performed well with all the applied FDI methods; bias faults are detected only after a delay of 5 min and the drift faults after a delay of 33–39 min.

In general, the systems with the MPC as a control strategy performed had better performance. The reason for this is the high MV rate weight values, which effectively compensated for the effects of the noise.

The effect of disturbances is also tested on the FTC-controller and MPC control system is found to be tolerant to the presented disturbances. The system is able to detect faults successfully even in the presence of severe disturbances affecting the system.

The results clearly indicate that all the methods presented here have considerable potential to be used as an effective FTC system for a real industrial process. The control systems also work efficiently in combination with the fault detection, identification and compensation methods. It is therefore possible to implement the proposed FTC system to a wide variety of controller types. In the experiments, however, the MPC had a better performance than the PI controllers in terms of accuracy, rate and interaction handling with the other measurements.

The effect of these faults in a system without an FTC system can also be seen in the figures; generally the impact of faults is large, the magnitude of the disturbances being the maximum or near maximum of the hard constraint limit of variables. Such faults would cause problems and lead to additional disturbances, huge economical losses and even equipment damage if they remain undetected in a real industrial plant.

7 Conclusions

In this paper, the PCA, PLS and SMI methods are used for fault detection, isolation and identification of the measurements of the Shell control problem (Prett & Morari, 1987) in a simulated heavy oil fractionator in order to achieve fault tolerance with MPC. Three FTC systems based on the FDI methods and fault compensation strategies are successfully implemented for the detection, isolation, identification and accommodation of the faults in the measurements of the simulated shell heavy oil fractionator. In these FTC systems, a measurement reconstruction and a measurement replacement strategy are used for fault compensation purposes. A reference trajectory tracking FTC method is also presented in the paper, but not implemented during the experiments.

The performance of the FTC systems and the different control strategies is evaluated. Based on the results, the presented methods have been shown to be effective and the fault compensation methods are able to counter bias and drift faults in the simulated measurements of the SCP process. With the PCA-based system, the SPE is used for fault detection, Hotelling T^2 index only being used for comparison purposes. The RMSEP index based on PLS latent variables is used for fault detection with the PLS-based FDI. The SMI-based FDI used the residual between the identified model output and the measurement value to determine whether a fault is present in the measurement. A cumulative sum algorithm is also implemented to avoid false alarms.

The performance and the fault detection and fault compensation rates of the different FTC systems are good with both the MPC and the set of PI controllers. The bias faults are detected rapidly only after a delay caused by the cumulative sum algorithm. For drift shaped faults the fault detection rate varied between 33 and 39 min (17–20% of the time the fault is present in the system), depending on the method and the controller used. The results of the experiments suggest that the presented FTC systems are effective, fast and able to counter different kinds of fault irrespective of the control system used. However, during the experiments the MPC provided more flexibility in the handling of variables and therefore appears to be more suitable especially for large MIMO systems. One should also note that the accuracy of the MPC internal model affects the control performance, even though some model mismatch is handled and tolerated by the MPC itself.

References

- Bemporad, A., Morari, M., & Ricker, N. L. (2006). *Matlab model predictive control toolbox—user's guide*. http://www.mathworks.com/access/helpdesk/help/pdf_doc/mpc/mpc_ug.pdf (May 16, 2006)
- Bordeau, M. A. (2003). *Squared model predictive controller performance on the Shell standard control problem*. ISA conference, Houston, USA.
- Bristol, E. H. (1966). On a new measure for multivariable process control. *IEEE Transactions on Automatic Control*, 11, 133.
- Dong, D., & McAvoy, T. J. (1996). Nonlinear principal component analysis based on principal curves and neural networks. *Computers & Chemical Engineering*, 20, 65–78.
- Camacho, E. F., & Bordons, C. (2000). *Model predictive control* (2nd ed.). London: Springer-Verlag.
- Chen, G., McAvoy, T. J., & Piovoso, M. J. (1998). A multivariate statistical controller for online quality improvement. *Journal of Process Control*, 8, 139–149.
- Chiang, L. H., Russell, E. L., & Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. London: Springer-Verlag.
- Cutler, C. R., & Ramaker, B. L. (1980). Dynamic matrix control—a computer control algorithm. In *Joint automatic control conference*.
- Dunia, R., Qin, S. J., Edgar, T. F., & McAvoy, T. J. (1996). Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42(10), 2797–2812.
- Frank, P. M., Ding, S. X., & Marcu, T. (2000). Model-based fault diagnosis in technical processes. *Transactions of the Institute for Measurement and Control*, 22, 57–101.
- Gerlach, R. W., Kowalski, B. R., & Wold, H. (1979). Partial least squares modelling with latent variables. *Analytica Chimica Acta*, 112, 417–421.
- Hyötyniemi, H. (2001). *Multivariate regression—Techniques and tools* (Vol. 125). Internal report. Helsinki: Helsinki University of Technology.
- Jackson, J. E. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, 21, 341–349.
- Jackson, J. E. (1991). *User's guide to principal components*. New York: Wiley Interscience.
- Komulainen, T., Sourander, M., & Jämsä-Jounela, S.-L. (2004). An online application of dynamic PLS to a dearomatization process. *Computers & Chemical Engineering*, 28, 2611–2619.
- Ku, W., Storer, R. H., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30, 179–196.
- Larimore, W. (1990). Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of 29th IEEE conference on decision and control* (pp. 596–604).
- Lee, J. H., & Cooley, B. (1997). Recent advances in model predictive control and other related areas. In *Proceedings of the chemical process control—V, CACHE, AIChE* (pp. 201–216).
- Li, W., Yue, H. H., Valle-Cervantes, S., & Qin, S. J. (2000). Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10, 471–486.
- Maciejowski, J. M. (1999). Modelling and predictive control: Enabling technologies for reconfiguration. *Annual Reviews in Control*, 23, 13–23.

- McAvoy, T., Jämsä-Jounela, S.-L., Patton, R., Perrier, M., Weber, H., & Georgakis, C. (2004). Milestone report for area 7 industrial applications. *Control Engineering Practice*, 12, 113–119.
- Patwardhan, S. C., Manuja, S., Narasimhan, S., & Shah, S. L. (2006). From data to diagnosis and control using generalized orthonormal basis filters. Part II. Model predictive and fault tolerant control. *Journal of Process Control*, 16, 157–175.
- Prakash, J., Patwardhan, S. C., & Narasimhan, S. A. (2002). Supervisory approach to fault-tolerant control of linear multivariable systems. *Industrial & Engineering Chemistry Research*, 41, 2270–2281.
- Pranatyasto, T. N., & Qin, S. J. (2001). Sensor validation and process fault diagnosis for FCC units under MPC feedback. *Control Engineering Practice*, 9, 877–888.
- Prett, D. M., & Morari, M. (1987). *Shell process control workshop*. Stoneham: Butterworth Publishers.
- Qin, S. J. (1998). Recursive PLS algorithms for adaptive data modelling. *Computers & Chemical Engineering*, 22, 503–514.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11, 733–764.
- Rawlings, J. B. (2000). Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20, 38–52.
- Richalet, J., Rault, A., Testud, J. L., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14, 413–428.
- Rivera, D. E., Morari, M., & Skogestad, S. (1986). Internal model control: PID controller design. *Industrial & Engineering Chemistry Process Design and Development*, 25, 252–265.
- Staroswiecki, M., & Gehin, A.-L. (2001). From control to supervision. *Annual Reviews in Control*, 25, 1–11.
- Van Overschee, P., & de Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic–stochastic systems. *Automatica*, 30, 75–93.
- Venkatasubramanian, V., Rengaswamy, R., & Kavuri, S. N. (2003a). A review of process fault detection and diagnosis. Part I. Quantitative model-based methods. *Computers & Chemical Engineering*, 27, 293–311.
- Venkatasubramanian, V., Rengaswamy, R., & Kavuri, S. N. (2003b). A review of process fault detection and diagnosis. Part II. Qualitative models and search strategies. *Computers & Chemical Engineering*, 27, 313–326.
- Venkatasubramanian, V., Rengaswamy, R., & Kavuri, S. N. (2003c). A review of process fault detection and diagnosis. Part III. Process history based methods. *Computers & Chemical Engineering*, 27, 327–346.
- Venkateswarlu, Ch., & Venkat Rao, K. (2005). Dynamic recurrent radial basis function network model predictive control of unstable nonlinear processes. *Chemical Engineering Science*, 60, 6718–6732.
- Verhaegen, M. (1994). Identification of the deterministic part of MIMO state space models given in innovations form from input–output data. *Automatica*, 30, 61–74.
- Vermasvuori, M., Vatanski, N., & Jämsä-Jounela, S.-L. (2005). Data-based fault detection of the online analysers in a dearomatisation process. In *Proceedings of first NeCST workshop on networked control systems & fault tolerant control* (pp. 219–224).
- Vlachos, C., Williams, D., & Gomm, J. B. (2002). Solution to the Shell standard control problem using genetically tuned PID controllers. *Control Engineering Practice*, 10, 151–163.
- Wang, J., & Qin, S. J. (2002). A new subspace identification approach based on principal component analysis. *Journal of Process Control*, 12, 841–855.
- Wold, H. (1975). Path models with latent variables: The NIPALS approach. In *Quantitative sociology: International perspective on mathematical and statistical model building* (pp. 307–357).
- Yoon, S., & MacGregor, J. F. (2004). Principal-component analysis of multiscale data for process monitoring and fault diagnosis. *AIChE Journal*, 50(11), 2891–2903.
- Yu, Z., Lee, J. H., & Morari, M. (1994). State-estimation-based model predictive control applied to shell control problem—A case study. *Chemical Engineering Science*, 49, 285–301.