Sensor validation and reconstruction: Experiences with commercial technology

Kevin Brooks* and Margret Bauer**

* BluESP, 53 Platina Street, Randburg, South Africa, +27 11 251 5900,
(e-mail: kevin.brooks@bluesp.co.za)
** Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Lynwood Road, Pretoria, South Africa, +27 82 720 4621, (e-mail: margret.bauer@up.ac.za),
Corresponding author

Abstract:

Detecting the failure of a sensor in industrial processes is important to avoid the use of incorrect measurements. When a sensor fails the missing measurement are reconstructed, using the measurements of other sensors and inferring the missing or incorrect measurement. Although this technology has been developed more than 20 years ago, there are few commercial solutions available today. One of these few solutions uses principal component analysis, based on an algorithm originally developed by Qin and Li (1999). In this paper, this solution is applied to operating data from a minerals processing plant with persistent sensor problems. Somewhat surprisingly, poor results are obtained, despite numerous attempts to improve reconstructability. Analysis indicates that the challenges are not about the algorithm but rather about choices that need to be made in the application of data-driven analysis tools to new data sets. These include data selection, filtering and interpreting which results are useful. It is suggested that together with any new algorithm presented. researchers should provide practical guidelines in choosing appropriate data, and any pre-processing that may be required.

Keywords: Sensor validation; sensor reconstruction; commercial software tools, principal component analysis; fault diagnosis; fault detection; process control.

1 Introduction

Operating an industrial process relies predominantly – if not entirely – on the measurements taken in the field using industrial sensors. These sensors are assumed to give the actual state of the process but in reality there are many issues that can affect the sensor: the sensor can show a drift, give a temporarily incorrect measurement or fail entirely for extended periods of time.

Faulty sensors may cause the performance of a process to deteriorate, result in the need for process shut-down or – in a worst-case scenario – even accidents. Sensor faults should therefore be detected as well as diagnosed and, if possible, the measurement value should be reconstructed using the best knowledge based on past and present measurements in the process. This area is often referred to as sensor validation and is related to the area of soft sensors (Kadlec et al., 2009).

Sensor validation ties in with the wider scope of control loop performance monitoring where all types of faults, not only those originating from the sensor, are detected. Performance monitoring, however, usually stops after diagnosing the fault and does not include the reconstruction of missing or incorrect values. Recently, several excellent books on process performance monitoring using statistical methods such as PCA and PLS have been published (Jelali, 2012; Ge and Song, 2012; Krüger and Xie, 2012; Huang and Shah, 2012). These compilations provide an overview of statistical methods that have been successfully employed for process monitoring. However, many of the methods have only been applied in academic single case studies – simulated or industrial – and are not commercially available (Bauer et al., 2016).

Common sensor faults are often categorised as bias, drift, precision degradation, gain and complete failure (Kullaa, 2013). There are also spike faults or outliers and nonlinearity or excessive noise (Lo, 2014), which are particularly important in wireless sensor networks. For some of these faults it is impossible to ascertain whether it is the sensor that causes the noise, or another component of the process or control loop, such as a sticky valve or a poorly tuned controller. For example, as described in Jelali (2012), sensor faults can lead to oscillations that resemble the behaviour of a poorly tuned controller.

The underlying justification for sensor reconstruction is that most processing plants have many sensors that measure related quantities and contain a certain level of redundant information (Jiang, 2011). For example, there will be several measurements along a distillation column that show similar features and there is a good chance of estimating the temperature at the bottom tray from the one a few trays higher up.

Several approaches have been developed to detect faulty sensors by inspecting all sensor measurements in a process section. The key idea is to build a data-driven model from the set of process data that was captured during normal operation conditions. A fault detection index is computed from the discrepancy between the model under normal conditions and during run-time operation. Work on the application of principal component analysis (PCA) for sensor validation in the process industries was done in the late 1990s and early 2000s by Qin and co-workers, with the earlier work focusing on detailing the method (Dunia and Qin, 1997; Dunia et al., 1996) and later works looking into more robust and automated approaches (Qin and Li, 1999; Qin and Li, 2001). These methods have been patented (Qin and Guiver, 2003) and are implemented in an industrial software package. More recent studies of the fault sensitivity of are described by Ding and co-workers (Ding et al., 2010). PCA has also been applied to air handling units (Padilla, 2015) and for vibration monitoring (Kerschen et al., 2005). Adaptations of PCA for sensor validation include structured partial PCA with nonlinear extensions (Huang et al., 2000), neural network adopted PCA model (Zhu et al., 2009) and adaptive kernel PCA (Chouaib et al., 2013).

The drawbacks of PCA – such as the problems that arise from multiple faults and other complex fault scenarios – are discussed in Lieftucht et al. (2006, 2009), who proposed an improved method using regression-based reconstruction. Again, this has to the authors' knowledge, not been included in a commercial tool and is not easily available to the practitioner. Other data-driven approaches involve neural networks (Upadhyaya, 1992), for example auto-associative neural networks (Xu et al., 1999) or Bayesian belief networks (Mehranbod et al., 2003).

One advantage of using data-driven models is that no process model is required. There are other approaches that require a model in form of state-space equations, either from first principles or from numerical system identification. For example, Da and Lin (1995) use a bank of Kalman filters to detect sensor failures while Simani et al. (2000) identify ARX models for Kalman filters. Li and Shah (2002) use state-space equations from system identification to compute structured residuals on the vectors of an observed state-space model.

The work by Qin and co-workers has been implemented within a commercial software package. This package also includes modules for developing and implementing soft-sensor models based on a variety of modelling tools. The package was employed to detect faulty analyser sensors and reconstruct the measurements if a fault has been detected; that is, the aim was to replace missing data from the analyser using other process measurements.

Questions that control practitioners are faced with when validating sensor data are unanswered in the literature, such as:

- How to choose the process measurements to be included in the analysis?
- How many process measurements are necessary and are too many measurements obscuring the result?
- How to select data time frames?

It is important to remember that two sets of data are required, one for model building and one during run mode. Guidelines for the selection of both are needed. Some of these questions were brought up in a recent paper (Qin, 2014), which states "while it was possible to require clean and accurate data in small data samples, we might have to live with messiness of the data and contain the errors with massive data. Robust methods in statistical machine learning are effective ways to handle messy data, although some level of pre-processing is always helpful"

The approach taken in this paper is to gather a large set of process and sensor data, and to apply the method using the limited guidelines available, together with the author's experience in choosing a typical sub-set of the data by visual inspection. It will be demonstrated that this approach fails, and that attempts to improve the situation are also fruitless.

In this article, the authors use an existing method but follow the entire industrial application and implementation to highlight practical approaches, but also pitfalls, when using data-driven methods. There are aspects that are often neglected when first presenting a new methodology. These issues are not unique to sensor validation but are also crucial when dealing with poorly tuned loops, valve stiction, process faults and asset or maintenance management using data-driven methods. As big-data technology is increasingly available in the process industries, these questions have to be addressed.

The work presented here does not shy away from reporting when the algorithm does not work and the typical problems one can expect when using data-driven methods. In particular, complete sensor failure and the reconstruction of the measurement from other process measurements when an analytical sensor is not available is addressed.

This article is structured as follows: Section 2 provides an outline of the sensor validation technology into which the sensor validation method is embedded. Section 3 describes the industrial problem to which the sensor validation approach was applied, including the process description and the data available for analysis. Section 4 outlines the findings of the data-driven analysis and how the data had to be modified and treated to give the sufficiently good results. Conclusions from these findings are summarised in Section 5.

2 Sensor validation technology

The contribution of this article is to verify, discuss and interpret the results of applying an industrial software package for sensor fault identification and validation using process data. The single most important aim of this work is to find an approximated sensor measurement when the sensor is faulty, that is, reconstruction of the sensor measurement. This value should be estimated from the other measurements that are known to be accurate when the fault occurs.

A significant contribution of this work lies in outlining the workflow and discussing the results and especially the case where the results are unsatisfactory. Reasons for incorrect and inadequate results are given and an improved method is suggested as a result in later sections. This section first gives the steps involved in sensor validation and discusses the use of historical process data as well as the possibilities of automating the methods. Fault classification as used in the industrial software package, as well as in the literature, is described.

2.1 Workflow of data-driven analysis

One could argue that the implementation of the data validation algorithm and the way it is presented to the user is the most important aspect for the success of such a tool. The users are generally process engineers and/or control engineers looking after the algorithm, with little time on their hands. Thus, the algorithm has to be highly automated and the user should not have to adjust any parameters. It is important to note that control engineering practitioners usually do not have the resources available to implement algorithms in development environments such as Matlab or even Microsoft Excel. Existing packages currently do not allow the programming of add-ons or interference with the implemented method. The implemented code is proprietary and encrypted. The workflow is reflecting the software package studied here, but is generic for most applications of data-driven analysis and similar if not identical workflows are used in academic studies.

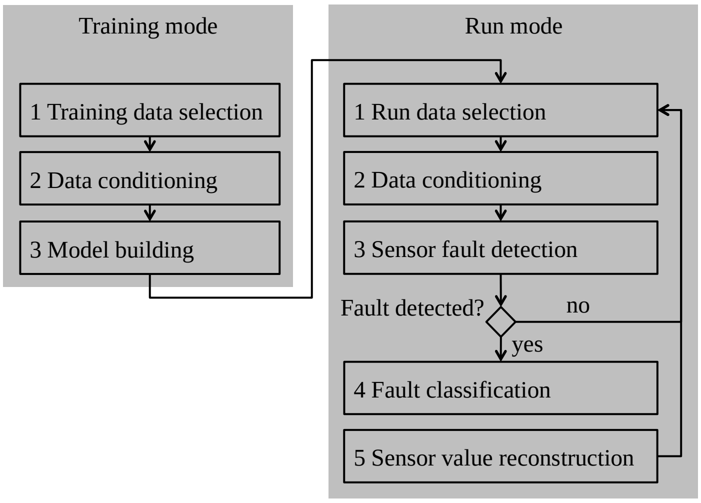

Figure 1. Workflow of the sensor validation software package.

The workflow for sensor validation module is shown in Figure 1. There are two modes or stages of sensor validation using data-driven approaches: training mode and run mode. The first stage is the training mode. The assumption is that in training mode all sensors are operating correctly and the process is showing normal operating behaviour. Selecting the data is arguably the most important step for using data-driven methods, since all results rely on the quality of the data. The first task in the data selection step is to choose the variables to be included in the analysis. Selecting normal operational data is not as straightforward as it seems, because all measurements to be included in the analysis should not experience any unusual behaviour. At the same time, there should be enough fluctuations to cover a range of possible behaviour.

After selecting the data, data conditioning can be applied to eliminate unwanted features such as high frequency noise, low frequency biases or missing data points. Great care has to be taken when conditioning the data because the result may change significantly depending on the chosen conditioning. The data conditioning has to be noted and this information should be stored for the run operation. Also, the variables to be included in the analysis can be re-adjusted and some of the variables can be de-selected.

After the steps of data selection and conditioning, the model is built. This step is described in Appendix A and based on the PCA algorithm by Qin and Li (1999). There is no option to change or adjust the algorithm nor is there information on the parameter setting of the algorithm, such as the number of principal components. It is possible to set a parameter known as the Minimal Reconstructability which determines if a particular value is included in the sensor validation model.

In the second stage, the run mode, sensor data is compared to the training data to detect and remedy faults. Again, the first step is to select the data from which a PCA model will be built. The data can be conditioned and it is advisable to choose the same data conditioning carried out as in the training set. For example, if all outliers were removed in the training model then outliers should also be removed from the run data. Fault detection as described in the Appendix is carried out to detect if there is a fault present in the signal. The faults are then classified, as will be described in Section 2.3. The last and arguably the most significant step is the reconstruction of the sensor value if a fault has been detected. The reconstructed value is then used instead of the measured sensor value for control or monitoring purposes. The run mode is implemented online and recursively, that is, the analysis is carried out on an on-going basis. If no fault is detected, as well as after the missing measurement has been reconstructed, the procedure returns to Step 1 of the run mode operation

2.2 Sensor fault detection and reconstruction

On performing tests on the plant described in the next section, it was observed that the analysers would fail repeatedly and consistently. However, redundant information was present in the system as analysers for three different materials were available on one input and two output streams according to mass balance analysis (Crowe, 1996). In addition, flows and level measurements were available, including reagent dosages. While no formal redundancy analysis was performed, a carefully constructed correlation analysis indicated that sensor models should be possible. The use of sensor validation and measurement reconstruction was therefore investigated.

A sensor fault occurs when a sensor gives an incorrect measurement or fails entirely. Validation of the measured sensor data involves three steps:

- Detection that one of the sensors is showing a fault;
- Identification of the faulty sensors;
- Reconstruction of the faulty data once the fault has been detected.

It is possible to detect a sensor fault by comparing the current measurement with historical data captured during normal production in the past. One approach, which uses a model derived from process data captured during normal operation, has been proposed by Qin and Li (1999). The approach discussed in this paper has been coded in commercially available software; it is this software that has been used in this study. The approach will be recaptured in this paper focusing on the implementation code and discrete time representation.

Sensor validation is not an exact science. This is partly because sensor degradation can be a gradual process. In this case study fault detection is helped by the fact that the measurement value of the analytical sensor drops to zero as it fails entirely. At the same time, the task of reconstruction becomes more challenging because there is no information at all about the sensor value.

When using data-driven methods for sensor validation the following important questions need to be answered:

1. How much data is required for training purposes?
2. How many samples need to be available and does the sampling rate affect the training model? This is particularly relevant for analytical sensors, which may give measurements at infrequent intervals.
3. What data is appropriate for training purposes, i.e. what are measures to identify ‘normal’ data?
4. How many sensor measurements should be included in the runtime analysis? This question has several aspects:
    a. Will the inclusion of more sensors give better results? Or is the opposite true?
    b. Is there a recommended procedure to choose the sensors to be included in the analysis?

The answers to these questions will affect the outcome of the analysis. Few guidelines are available, apart from the examples given in the paper that developed the method. For instance, Qin and Li (1999) give a boiler example with seven variables. Over 630 data points were collected at a five-minute sampling interval and the authors state “the data were collected during a period of significant change in the boiler throughput so as to cover a wide range of the process behaviour”. Chiang et al (2000) used a training set of 500 samples at a 3-minute frequency, together with 52 variables to study fault detection in the Tennessee Eastman process.

For industrial data, the data length and number of variables is often predetermined by the nature of the process. In the author’s experience as well as in the following case study normal operating data is more rare than expected on many processes.

Here, it is shown how the choice of variables and data affects the data-driven analysis result. Researchers proposing these methods need to provide clear guidelines for the data selection to ensure that the work will be replicable. The focus here is on sensor value reconstruction because faults are easy to detect because the sensor value simply drops to zero if the sensor fails. As a baseline comparison, the reconstruction has to be better than the alternative case where the last correctly measured value is held until the sensor is operating again.

2.3 Classification of Faults

Having identified one or more faulty sensors, it is useful to be able to classify the type of sensor fault. In the industrial package, five different fault categories are identified using statistical testing. These categories are as follows and reflect the faults in the literature, see e.g. Kullaa, 2013.

Bias: A bias fault occurs when there is a large, persistent change in the indicated value, without an underlying process reason.
Drift: Drift refers to a slow change in one direction of the measured value, again without there being a process reason.

Precision: Loss of precision is associated with an increase of the noise associated with the measurement.
Frozen: A frozen measurement shows no change, despite there being changes in the process, which can be observed in other process variables. It is generally associated with complete failure of the analysis system.
Unknown: Any other fault that does not clearly show any of the previously described patterns is considered to be unknown.

For the purposes of the study described in the next section, fault classification is not critical because the faults in question are complete failures with the measurement dropping to zero, that is, there is a complete loss of the measurement signal. A fault can be identified by a simple test if the measurement is zero. Nevertheless, fault classification is carried out as it is a part of the sensor validation implementation. The results will be noted as they highlight the potential for incorrect fault classification.

3 Problem description

The sensor validation concept and methodology is applied to an industrial process, which will be described in this section. Here, the specific task of sensor fault detection and reconstruction as required for this particular process is described before giving a process description and the process data that is analysed.

Sensors in the process industry can be divided into at least two categories, namely (i) standard sensors to measure temperature, pressure, level and flow and (ii) analytical sensors that measure other quantities such as density, pH, or conductivity. Standard sensors have a quick response time and measurements are available every second or even more frequently. In addition, these sensors are comparatively inexpensive so if a measurement is important, a redundant sensor is put into place to ensure that the measurement is accurate. Analytical measurements, on the other hand, are expensive in comparison and may only be available infrequently. Analytical measurements can be taken either online or offline. Offline measurement samples are manually taken to a laboratory, and it may take hours before the results are available. Online measurements also take time to process and are usually available within several minutes. For example, in the case study described here the online measurements are available in up to 15 minutes. Each analytical sensor can provide several measurements for different physical quantities.

In the described industrial process, both standard process sensors as well as analytical sensors are present. Sensor faults occur frequently in analytical sensors and to a lesser extent also in the process sensors. Since the online model predictive controllers make decisions predominantly based on the analytical measurements it is important to replace missing measurements with their estimates. In this section, first the industrial process is described followed by the operating data available for training and testing.

3.1 Process description

The process under consideration consists of the first four cells of a flotation plant that separates a zinc sulphide ore from other minerals (see Figure 2). These four cells form what is termed the ‘rougher bank’. The aim of the rougher bank is to perform a rough cut of the feed, thus separating out fast floating components. This is achieved by feeding the crushed, ground ore mixed with water to the plant where flow instrument FI1 and AI1 are located. A surfactant chemical (sodium ethyl xanthate, SEX) is added to the slurry, together with $CuSO_4$ and naphthalene sulphonate (NS). Air is blown into the slurry to produce bubbles. Particles attach to these bubbles and rise to the top creating froth. The froth is then skimmed from the top of the first cell as a concentrate and processed further via the pump box. The remaining slurry is passed on to subsequent flotation cells where the same mechanism creates a froth, which is skimmed again.

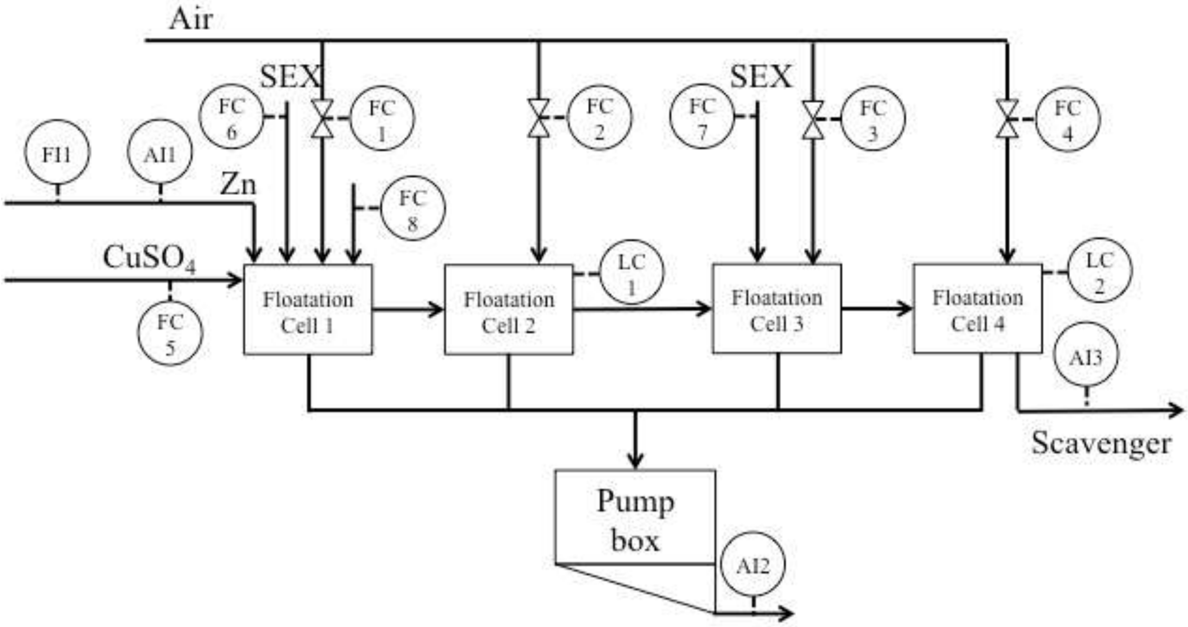

Figure 2. Process schematic of flotation rougher bank.

A description of the sensors – both controlled variables and indicators - is given in Table 1, the location of which can be seen in Figure 2. There are eight flow control loops, four for the gaseous air inlet and four measuring the inflow of chemical additives. The feed rate of the slurry feed is also measured but not controlled. The liquid level is measured and controlled for the second and fourth cell. Analytical sensors AI1 to AI3 are placed at the feed, the pump box and the scavenger respectively. Each analytical sensor measures three quantities namely zinc, lead, and iron contents. In this work, these are referred to as, for example, AI1.ZN, AI1.PB and AI1.FE.

All measurements are avaialble on the plant distributed control system (DCS). All variables are also written to the plant historian, where they are historised at one-minute frequency.

The work described here forms part of a larger advanced process control (APC) project, aimed at installing online control and optimisation of the flotation unit. As a result, dynamic models are available covering many of the process variables, see Brooks and Koorts, 2017.

For the purposes of closed loop control and optimisation of the flotation circuit, the two most important measurements are the zinc in the concentrate (AI2.ZN) and the tails (AI3.ZN). Should these measurements not be available for an extended length of time, the advanced process control system, which depends on them, would need to be turned off. This has significant financial implications for the operation (Bauer and Craig, 2008). If we consider an APC system having benefits of, for example, \$1,000,000 per annum, turning off the system for ten percent of the time due to faulty analysers would represent a loss of \$100,000 per year. An investment in software and the engineering effort to implement can be easily justified.

Table 1: Process measurements and analysers in the flotation rougher bank.

Tag	Description	Type	Units	Frequency
LC1	Liquid Level in Cell 2 (PID loop)	Ultrasonic	mm	1 second
LC2	Liquid Level in Cell 4 (PID loop)	Ultrasonic	mm	1 second
FC1	Air to Cell 1 (PID loop)	Pitot	m_n^3/hour	1 second
FC2	Air to Cell 2 (PID loop)	Pitot	m_n^3/hour	1 second
FC3	Air to Cell 3 (PID loop)	Pitot	m_n^3/hour	1 second
FC4	Air to Cell 4 (PID loop)	Pitot	m_n^3/hour	1 second
FC5	Copper Sulphate Addition (PID loop)	Coriolis	m^3/hour	1 second
FC6	Sodium ethyl xanthate (SEX) to cell 1 (PID loop)	Coriolis	m^3/hour	1 second
FC7	Sodium ethyl xanthate (SEX) to cell 3 (PID loop)	Coriolis	m^3/hour	1 second
FC8	naphthalene sulphate to cell 1 (PID loop)	Magnetic	m^3/hour	1 second
FI1	Feed volumetric flow	Magnetic	m^3/hour	1 second
AI1	Feed online X-ray fluorescence (XRF) analyser	XRF	% Zn, Pb, Fe	10 minutes
AI2	Concentrate XRF analyser	XRF	% Zn, Pb, Fe	10 minutes
AI3	Tails XRF analyser	XRF	% Zn, Pb, Fe	10 minutes

3.2 Process Data

The process measurements are available at high frequency, that is, less than one second sampling rate. The XRF analysers measure zinc, lead and iron use a sampling system. The scan rates of the devices vary from 10 to 30 minutes. Due to the reasonably slow nature of the process, the XRF measurements are suitable for online advanced process control, provided that the analyses can be trusted.

Six weeks of data at ten-minute sampling rate was collected from the plant historian and made available to the authors. The data is available on the web repository hosted by the South African Council of Automation and Control (SACAC): https://sacac.org.za/resources/. The dataset thus consists of 6029 samples of eleven process values and nine analyser values. The data are plotted in Figure 3 and Figure 4. It can be observed from Figure 3 that the analytical sensor data can fall away, seemingly at random intervals. When this happens, the measured value drops to zero and all three measurements of zinc (ZN), lead (PB) and iron (FE) are lost for a particular analyser. The data is available on-line at sacac.org.za/resources.

Figure 3 and Figure 4 illustrate the complexity of data selection for the sensor validation. The analysers fail frequently and there are only short time-windows where all three analysers are giving correct readings at the same time. This is by no means unusual. Online analysers are notorious for being unreliable. The process measurements of flow and level are more reliable though no information whether the readings are correct is available. In addition, there are periods of process upsets towards the end of the data set and these periods should not be included in the training set. In the remaining sections, there is a frequent complete failure of all analytical sensors, which complicates the analysis and validation further.

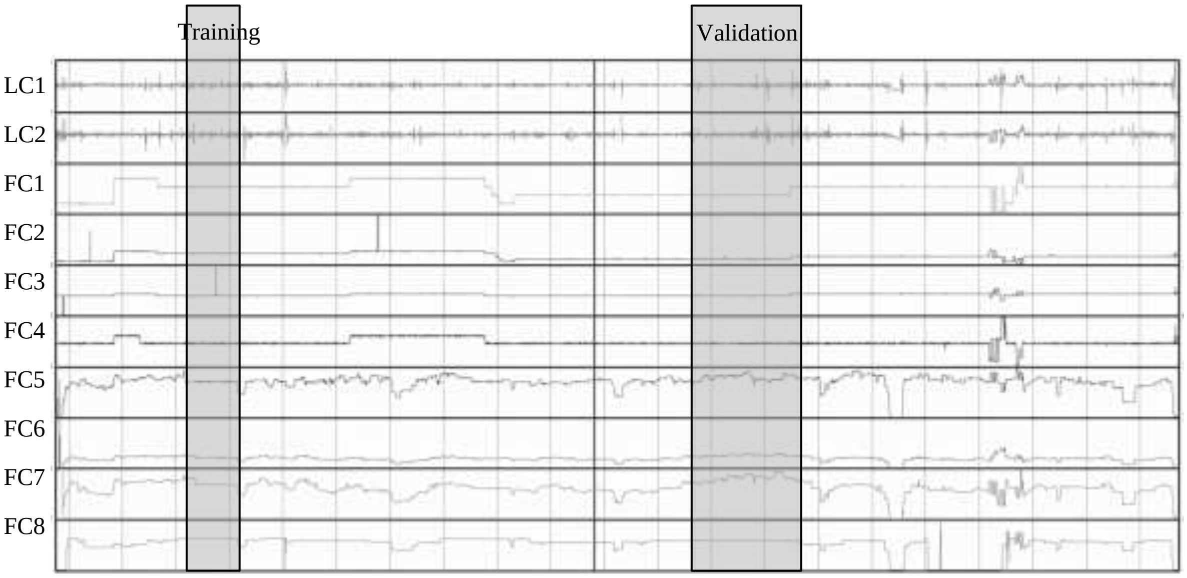

Figure 3. Process measurement trends of flotation process – six weeks data with 10-minute sampling interval; grey areas indicate data selection for training and validation.

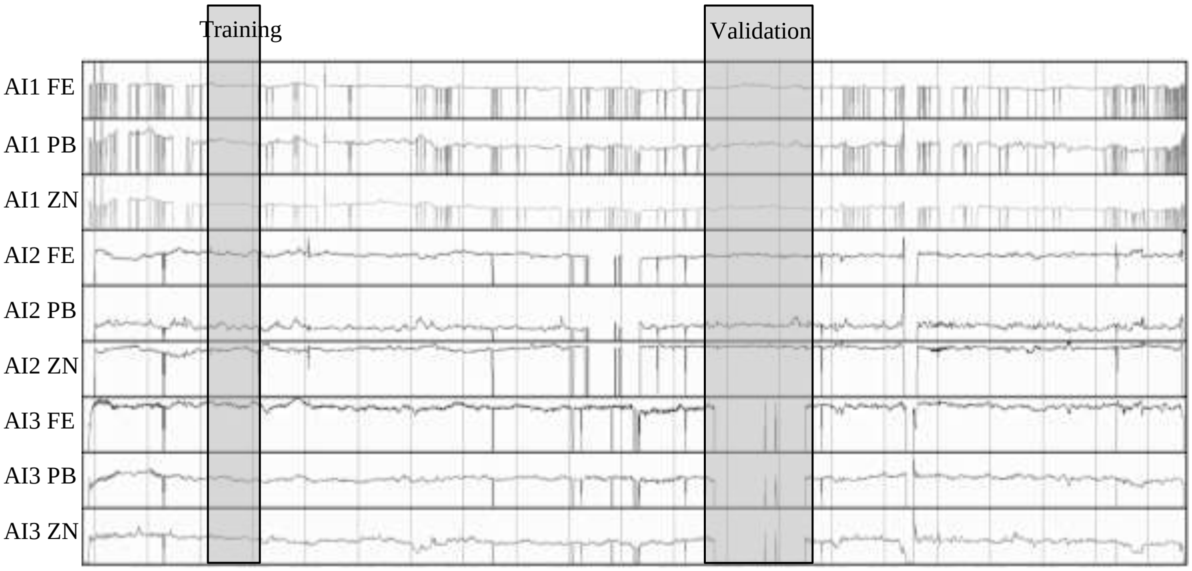

Figure 4: Sensor measurements from three analysers – six weeks data with 10-minute sampling interval; grey areas indicated data selection for training and validation.

4 Case study results

This section applies the methodology described in Sec. 3 to the industrial process of Sec. 2 and is structured as follows. In the training mode, a PCA model is generated from a period of 'normal' operating data. The normal operating data is defined as a period where none of the sensors fail and the process measurements are mostly stationary as described previously. The validity of the PCA model is then verified with the training data. To further test the methodology, a single, artificial fault is introduced to the **same data** as used for training to show that the fault detection and reconstruction works for this data range. In run mode, the model, fault detection and reconstruction is applied to another section of the data different to the training data with faults present. The results are discussed and different ways to improve the result are suggested.

4.1 PCA model generation: Training mode

As described in Section 3.2, the PCA model is generated using normal operating data, that is, data where no fault is present and no strong disturbances affect the process. Many possible choices can be made when deciding on what data should be used to create the underlying PCA model required for the technique. Some questions include:

1. Is more data necessarily better than less, that is, how should the time frame be selected?
2. Should obviously bad data (particularly for the analysers) be cut? Alternatively, can the data be interpolated?

In this particular case study, there are three components for each analyser, namely zinc, lead and iron. This is a common situation for analysers where several quantities are measured at the same time. For the sensor validation algorithm, this occurrence leads to a third question:

3. Should the data for all three components (Zn, Pb, Fe) be used, given that when the analyser fails all three values are lost?

Qin and Li (2001) give little advice on the choice of data. In the authors' experience the data selection is usually not included in publications on data analysis algorithms. It may be even suggested that what often appears to be a random selection of data is a careful, manual selection of data, possibly after some analysis was conducted.

The data used in this study shows extensive periods of time where one or another analyser are not reading, so as a first step, a period of data was chosen for which the majority of the sensors were giving reasonable readings. This dataset consists of 301 samples or 50 hours and is marked in Figure 3 and Figure 4 as the left-hand rectangle. The corresponding dates in the data are 29-Sep-15 02:00 to 01-Oct-15 04:00. This data selection is referred to as the training set. It is important to point out here that the selection was done by visual inspection. Automatically and generically selecting the training data is a difficult task and depends significantly on the process dynamics. Methods based, for example, on Gaussianity testing are available but none of them have been implemented in the software package used.

No data conditioning was performed before using this data in the PCA model development; the software used applies a first order filter. The software implements a minimum reconstructability parameter; a threshold value that, when a fault condition is detected during model training, serves as a measure of how much better the value computed by the sensor validation model is than the average value for the variable. Variables for which the reconstructability falls below the minimum reconstructability value are not incorporated in the sensor validation model. Reconstructability is based on the variance of the reconstruction error when compared with the variance that results from using average values rather than the model estimate. The reconstructability index equals to one if the variance is identical and zero if the variance of the reconstruction error is zero.

For the default reconstructability index of 0.5, flows FC1, FC2, FC3, FC4, FC5, FC8, FI1 and the levels LC1 and LC2 were not considered to be reconstructible. Only the SEX flows remain in the models, together with the analyser values. The statistics for the PCA fits for the remaining variables are given in Table 2.

Table 2. Statistics for fits of variables in the training data set.

Tag	Average	Square of correlation coefficient	Mean Square Error (MSE)	MSE/ Average (%)
AI1FE	13.41	0.87	0.11	0.8%
AI1PB	1.33	0.97	0.01	1.0%
AI1ZN	9.89	0.91	0.12	1.2%
AI2FE	8.54	0.95	0.09	1.1%
AI2PB	2.12	0.81	0.12	5.6%
AI2ZN	41.94	0.92	0.37	0.9%
AI3FE	15.43	0.94	0.15	1.0%
AI3PB	1.15	0.93	0.02	1.5%
AI3ZN	3.48	0.72	0.09	2.5%
FC6	4.56	0.96	0.08	1.8%
FC7	2.73	0.96	0.05	1.7%

The model fits are, in the most part, very good as evidenced by the low values of the MSE relative to the average, as well as the high correlation coefficients. This indicates that the PCA model has potential for detecting faults and reconstructing data. ***Error! Reference source not found.***Figure 5 shows the fits for AI2.ZN and AI3.ZN which are the most important variables from an application point of view. Visually the fits are good.

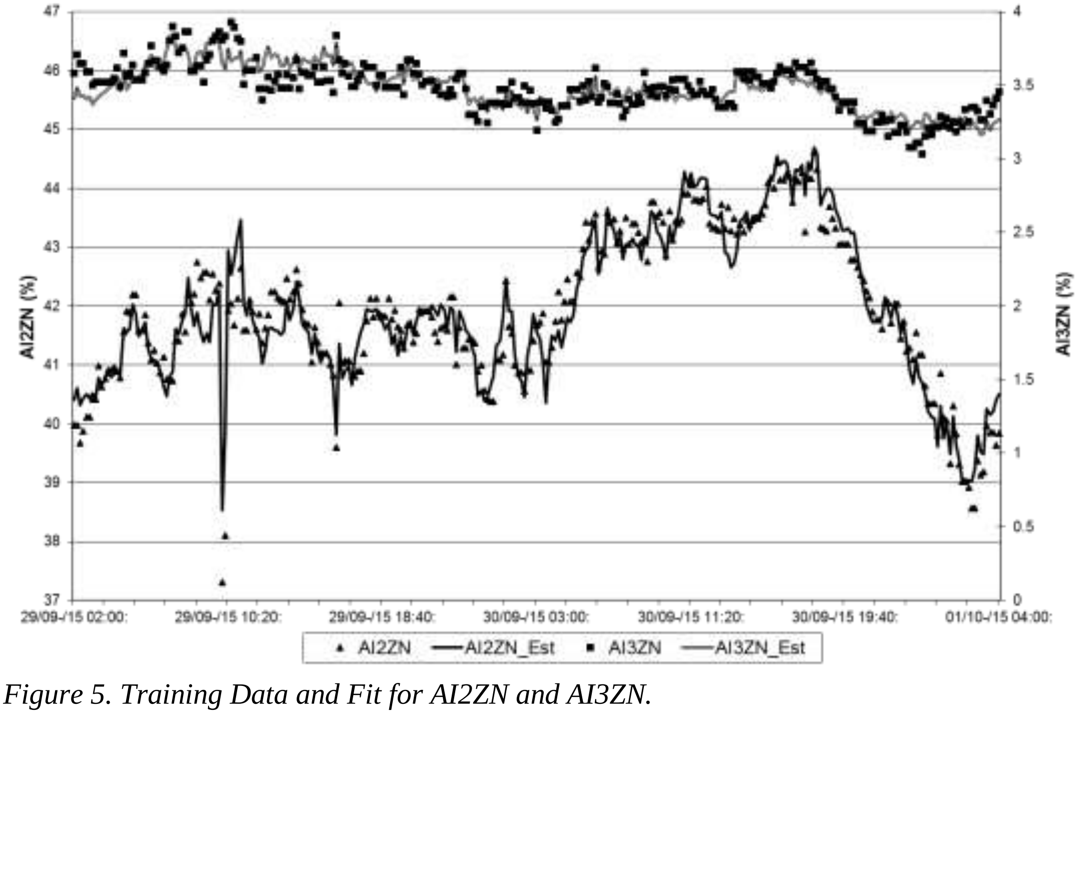

Figure 5. Training Data and Fit for AI2ZN and AI3ZN.

4.2 Reconstruction for induced single faults

Fault detection approaches can be tested by introducing artificial faults in the training data set (Qin and Li, 1999). A strong feature of the software package used is that the results can be tested by running the detection routines with simulated or real data. The software reports detected faults, and the calculated replacement value.

In this case a failure for AI3.ZN on 29-Sep-15 at 18:30, to a value of zero for 100 samples, was introduced. The fault detection routine recognises the fault and categorises it as “unknown” for four samples, before correctly re-categorising it as “frozen”. When the fault is removed at sample number 200, it is categorised as a “bias” error for two samples and then “unknown” for one. This behaviour is shown in Figure 6.

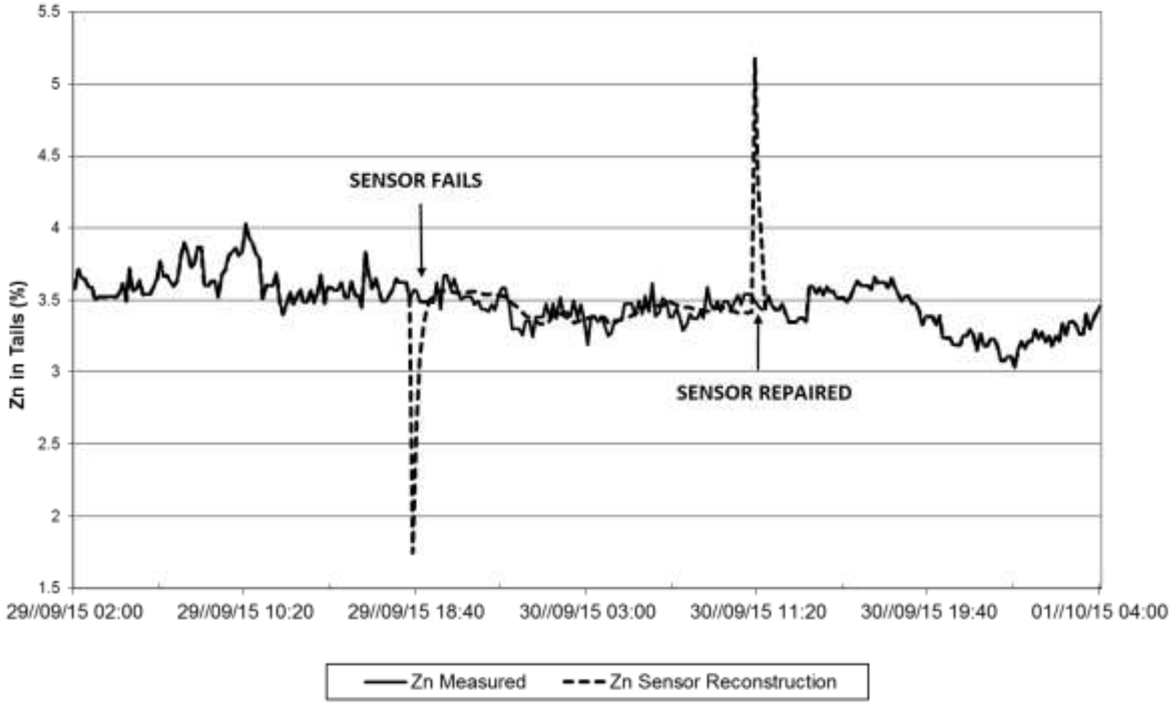

Figure 6. Artificial fault in AI3.ZN and reconstructed values.

The sensor reconstruction shows spikes when the fault is introduced and removed. The reconstructed value, which is simply the PCA fit, is good in between these events. With some validation, and possibly filtering, the behaviour above would be acceptable for use in an online control application.

4.3 Detecting and reconstructing real sensor faults

The ability of this model to detect faults – and provide reconstructed values where necessary – was tested using data that was not part of the training set. This is the data labelled “validation” in Figure 3 and Figure 4. This data includes a period of three and a half days where, apart from two short-term outliers, analyser AI3 was not reading at all. The corresponding dates in the data are 18-Oct-15 03:32 to 23-Oct-15 00:12. The faults detected by the routine, and the number of occurrences is detailed in Table 3.

Table 3: Fault classification for the validation set.

Tag	Bias	Drift	Precision	Frozen	Unknown
AI1PB					1
AI1ZN	74	88		4	40
AI2ZN					6
AI3FE		1	1	11	5
AI3ZN		8	5	10	19

The results in Table 3 are puzzling. The simplest way to detect failure of the analysers used in this study is by means of a range check. A value of less than 0.1 certainly indicates a failure. Using this criterion on the validation set reveals that each of the AI3 analysers have 491 values that indicate failure, and the AI1 and AI2 analysers each have only one failure.

Having detected the failures, the algorithm proceeds to reconstruct the sensor values. Figure 7 gives the measurement and reconstructed values (labelled AI3.ZN_SV) for AI3.ZN. Note that should the algorithm not detect a failure, that the reconstructed value is simply equal to the measured value.

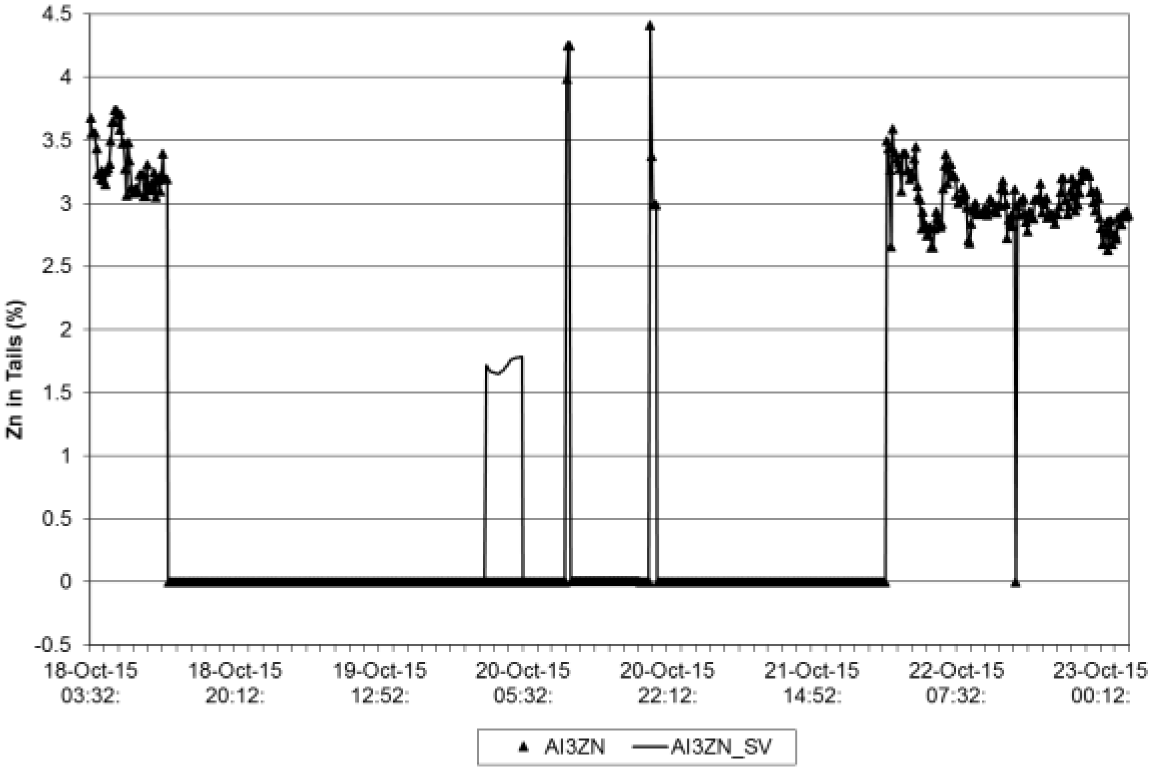

Figure 7. Original and reconstructed data for AI3.ZN.

The algorithm has clearly failed to identify the periods for which the analyser is faulty. The reconstructed value is zero where it is certain that the sensor has failed. It was expected that the measurement should have been replaced by a percentage value around 3%, as measured before and after the sensor failure.

Surprisingly, as indicated in Table 3, the majority of the faults are found in AI1.ZN, a signal that visually, apart from one zero value, appears to be giving consistent results. The original and reconstructed values of this analyser are shown in Figure 8. The routine has generated a number of false fault detections, and the replacement value is inaccurate.

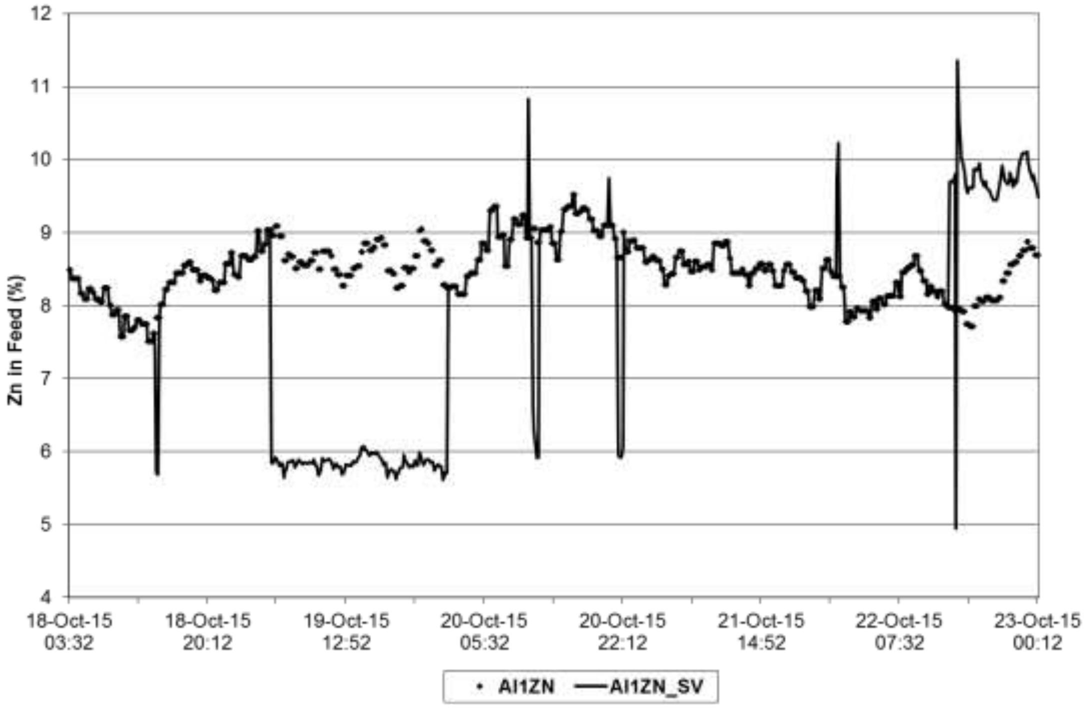

Figure 8. Original and reconstructed data for AI1.ZN.

While not proven here, the issue in reconstructing the data here appears to revolve around the fact that multiple faults are presented, but, as shown in Table 3, not necessarily identified. The reconstruction can therefore be using values that by inspection are known to be faulty.

4.4 Data conditioning and selection experiments

Since the result of the sensor validation and reconstruction algorithm as implemented in the software package was unsatisfactory, further experiments were conducted to see if the results could be improved. These experiments are based on careful considerations and many years of experience working with process data for monitoring and control purposes. These kinds of experiments, though ad-hoc, are usually conducted by practitioners but rarely, if ever, reported in academic publications.

These experiments show the complexities inherent in sensor validation: There are a great deal of options to choose from in conditioning the data, which might improve the validation result. The conditioning and selection experiments together with the expected and observed results are given in Table 4. Key to conducting these ad-hoc experiments is to determine the expected results before conducting the experiments. Both expected and observed results were discussed with the industrial partner.

Table 4. Data conditioning and selection experiments.

Data conditioning experiment	Description	Expected result	Observed result
1 – Introduce low limits on AI3	Bad data forced manually by adding low limits of 0.1 on AI3 values	Bad data immediately recognised.	Bad data is recognised, but not consistently reconstructed correctly
2 – Training with only Zn analyses and process data	Remove all Pb and Fe analyses from the data set	Better reconstructability due to fewer multiple faults	Only AI1ZN meets the reconstructability criterion
3 – Training with only Zn and Pb analyses and process data	Remove all Fe analyses from the data set	Improved reconstructability due to fewer multiple faults	Same variables chosen as for base case. Reconstruction not satisfactory
4 – De-spiking the data	Data for A1.ZN de-spiked by interpolation to get rid of data drop out	PCA model will pick up drop out in original data and reconstruct well	Method detects dropout, but reconstruction inconsistent
5 – Using more samples to train the PCA model	1870 samples used to train the model.	Better fit and improved reconstructability	PCA fit is good, but reconstructability remains poor

For Experiment 1, low limits were specified for the AI3 values. This has the effect that the algorithm does not have to detect the bad data associated with values falling to zero, but rather that the data is immediately flagged as bad. The PCA model was then rerun on the validation set. The result is that when running the validation data, the AI3 analyser is correctly flagged as faulty. The algorithm then reconstructs the data as shown in Figure 9.

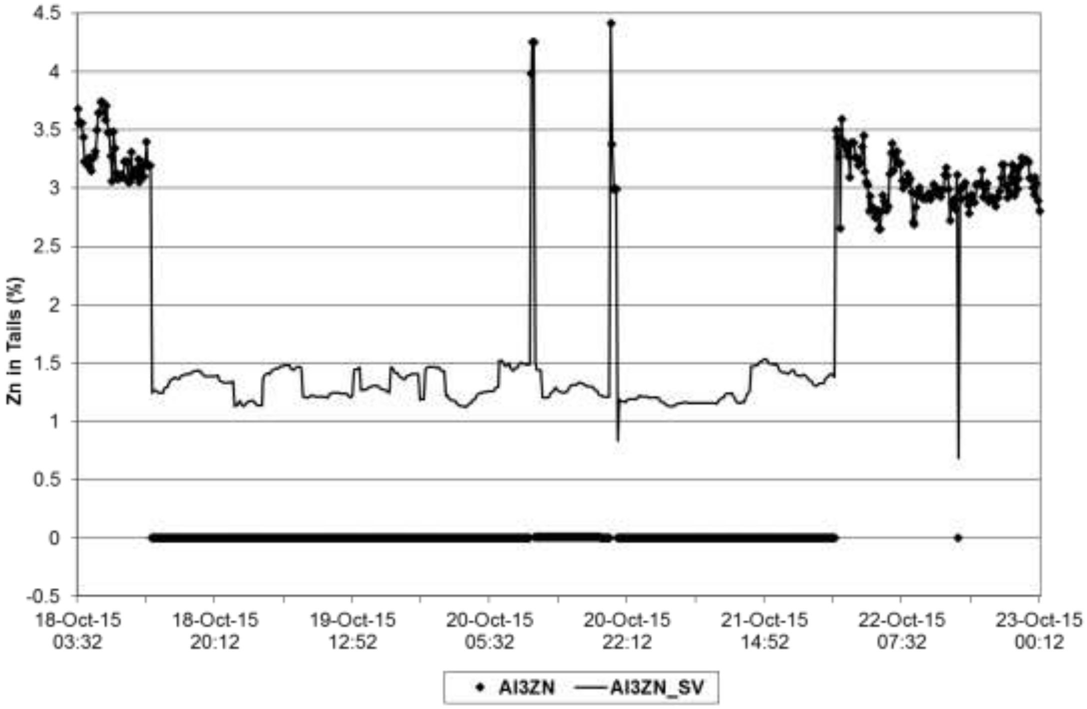

Figure 9: Original and reconstructed data for AI3.ZN with low limit specified.

Compared to Figure 8 this is a better result; although the absolute value of the reconstructed data shows an offset, the trend appears reasonable. A straightforward bias correction would make this result useable in practice. The same cannot be said for the result for AI3.PB. As shown in Figure 10, the reconstruction of this value is very poor.

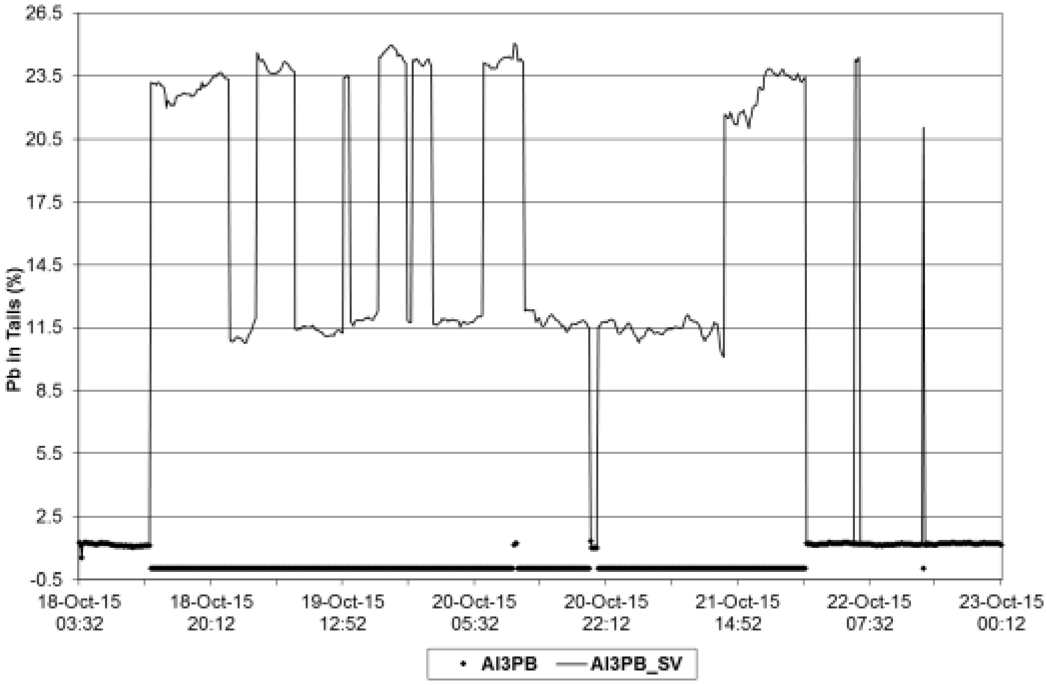

Figure 10. Original and reconstructed data for AI3PB with low limit specified.

In no case were the reconstructed sensor values in the validation set acceptable from a visual perspective. The conclusion is that on real data this method fails to detect faulty sensors correctly as shown throughout this section, and even if low or high limits are imposed, the reconstructed sensor values are not reliable. As shown in Table 4, many other data conditioning strategies were tried. None were successful.

5 Discussion and recommendations

In the industrial case study presented here, a set of sensors in a minerals processing facility was studied. The focus was on analytical sensors, which fail during the operation. That is, no measurement is recorded for extended period of time for the analytical sensor. The aim of the study was to reconstruct a missing measurement from other process data. An industrial software package, that implements the method described in Qin and Li (1999) and Qin and Li (2001), was applied. It is based on PCA and derives a process model from process data. The PCA model was first used to detect that a fault has occurred because the collected data deviates from the model. In a second step, the data that was detected as faulty was reconstructed using the remaining process measurements and the PCA model.

From a process perspective, there clearly is a connection between the different analytical sensors. The analytical sensors measure zinc, lead and iron at different stages in the process. These measurements are directly related. Also, the process measurements are related to the analytical measurements. Therefore, it would be expected that some form of prediction of a missing measurement from the other sensors should be achievable.

There are several challenges that have to be noted, and which should be considered when developing and improving existing sensor validation and reconstruction algorithms. These challenges are discussed along with the different stages of fault detection and reconstruction.

Selecting the training data

It is critical to select the correct data for training and here several issues come to the fore.

1) As Kerschen (2005) states: “It is first assumed that the reference data set contains enough information to cover normal process operation.” It is not straight forward to choose good data for most industrial processes. To generically establish what constitutes ‘normal process operation’ is inherently complex and inevitably requires expert knowledge. In the case study described in this paper, there are very limited periods of normal operation where all analytical sensors give a result, as can be seen in Figure 3 and Figure 4. The intervals where all three analysers were functional were selected for training purposes, but these intervals are limited and not all flow sensors were giving correct readings during this time. In the authors’ experience, this situation is not unique but rather reflects the standard situation. Techniques and/or expert knowledge are required to identify normal operating conditions.
2) Assuming that the process is operating as desired and all sensors are fully functional, the question is which sensors to include. There is a manifold of sensors in most industrial processes and not all will give required interconnected information. Also, most processes to be investigated are subsections of a larger production process. For computational but also for interpretational reasons it is not helpful to include a large number of measurements. Instead, measurements close to critical process variables that need reconstruction should be included. There is a need to develop guidelines on how to select sensors for data-driven sensor validation and reconstruction.
3) A third issue is, once normal operating conditions are identified and fulfilled, how to select the correct time frame and how much data to select for training purposes. The question is whether more data is always better than less as was assumed in this study. Here, the longest period of uninterrupted ‘normal’ operating conditions was selected. It consisted of 300 samples, which appears to contain enough information.
4) Once the data is selected, the option of data conditioning arises and there need to be guidelines whether the data should be pre-conditioned to remove outliers, periods of dropout etc. In the authors’ experience removing outliers increases the validity of the model but further investigation is required. Also, filtering can be applied to the data to focus on specific faults. Again, the selection of filters and other pre-conditioning of the data requires expert knowledge. Guidelines are needed for their application.

Detecting the sensor fault

Similar issues as for the training data selection arise once a model has been derived from training data and sensor faults should be detected. Generally, the same selection as for the training data should be applied for the run data, that is, the same variables should be selected and the same data conditioning should be applied. This appears obvious but this guideline can easily be violated when selecting the variables in practice. When using the PCA model approach fault detection occurs on a sample-by-sample basis so the selection of the time frame is not required.

Reconstruction of missing sensor measurement

Any measurement value reconstructed with a data-driven method, such as using the PCA model, has to be better than the simplest of reconstruction: holding the last correct value that was recorded and replacing any new measurement with this value. In an industrial case study, the correct value of the missing sensor measurement is usually unknown but artificial faults can be easily introduced and the deviation from the actual value can be estimated and an even better estimation of the quality of the reconstructed result can be achieved.

The PCA based method as described in Section 3 did not prove to be robust when applied to real plant data, with its drift and un-modelled disturbances. The results of Section 4 show that the approach could deal with artificially introduced faults but not with real faulty process data for both sensor fault detection and sensor value reconstruction. Data conditioning and selecting experiments were conducted and the results are highly unsatisfactory. It reveals that there are pitfalls when implementing data-driven methodologies. The method has been successfully applied in the past (Qin and Li, 1999, 2001) and clearly worked for certain data sets. However, the selection of the data and case studies where the PCA model can be applied appear to be critical.

From a practitioner's perspective, a simple regression model, together with rules on sensor validity, and a bias update, can provide a workable solution to sensor validation for a limited number of sensors. For situations that require validation of a large number of sensors, this method is clumsy and time consuming. In an era where handling of "big data" has become a keyword, it is hoped that methods can be developed that simplify this task. In the field of inferential sensors, an update is often applied to the model to improve its accuracy. By analogy, a method of introducing bias to the PCA model predictions may improve its usability. This could be a simple additive bias or might employ a Kalman filter. This is an area for on-going research.

6 Conclusions

In this paper, a commercial software tool for sensor validation and reconstruction has been tested using sensor data from the operation of a froth flotation plant. The dataset consists of eleven process measurements and nine analyser values from three sensors. The PCA model was trained using three hundred samples, collected at ten-minute frequency. At a minimal reconstructability of 0.5, all nine analyser values are included in the model, together with two reagent additions. The resulting PCA model shows good fitting statistics. The model successfully detects a simulated fault in the training data and provides reasonable reconstructed values. However, when the model is used on a validation set of data that includes a large period in which one analyser has failed, the results are not acceptable. This case study shows that, for the applied process data, the sensor faults could neither be detected nor reconstructed.

From a practitioner's viewpoint, the PCA model together with the structured residual approach to fault detection is appealing as it uses data can be fully automated. Modern plants produce a large amount of reasonably high frequency data, which should be suitable for producing practical solutions to solutions to important problems. This work demonstrates that guidelines are needed on:

- How best to choose the data set to train the model.
- What conditioning of the data is both required and desirable
- How data collected at varying frequencies should be managed

Leaning on recent work in data management, it should be possible to use an algorithm to precondition the data. This would move work in this field more towards the "big data" handling methods that are moving into the processing world. As Qin (204) points out "the data-mining and machine-learning communities have developed robust methods that use imperfect data as the norm rather than as the exception … while it was possible to require clean and accurate data in small data samples, we might have to live with messiness of the data and contain the errors with massive data”. Our recommendation is that attention be paid to developing guidelines that enable the practitioner to easily use the powerful tools that have been and are currently being developed.

Appendix A: Fault detection, isolation and sensor validation

A sensor fault is detected, isolated and validated using the same data-driven model based on principal component analysis (PCA). Deriving the PCA model is described in Qin and Li (2001). Here, the method and focus on how the implementation in discrete time as required for an industrial software package is recapped. The process model is derived from data captured during a period defined as normal operation. At each time instance *k* a measurement *x** is captured for all *m* sensors.

$$\mathbf{x}^*[k] \tag{1}$$

There are a total of *N* time instances or samples. Correlation coefficients define the statistical relationship between two measurements x_i^* and x_j^* over all times *k*:

$$r_{x_i^* x_j^*} = \frac{\sum_{k=1}^{N}\left(x_i^*[k]-\bar{x}_i^*\right)\left(x_j^*[k]-\bar{x}_j^*\right)}{\sqrt{\sum_{k=1}^{N}\left(x_i^*[k]-\bar{x}_i^*\right)^2 \sum_{k=1}^{N}\left(x_j^*[k]-\bar{x}_j^*\right)^2}} \tag{2}$$

The correlation matrix of time measurement vector **x*** investigates the relationship between all *m* sensor measurements and is defined as follows:

$$\mathbf{X}^* = \begin{bmatrix} r_{x_1^* x_1^*} & \cdots & r_{x_1^* x_m^*} \\ \vdots & \ddots & \vdots \\ r_{x_m^* x_1^*} & \cdots & r_{x_m^* x_m^*} \end{bmatrix} \tag{3}$$

The correlation matrix **X*** will be exploited to predict the relationships between the measurements in the future. If the relationships are different from the relationships under normal operating conditions then a fault is detected. Once the relationship has been defined, a further step in generating the prediction model is to carry out an eigen-decomposition of the correlation matrix. The eigen-decomposition finds the principal components under normal operating conditions:

$$\mathbf{X}^*\mathbf{q} = \lambda\mathbf{q} \tag{4}$$

The vectors $\mathbf{q}_1 ... \mathbf{q}_m$ are the *m* eigenvectors of **X*** and $\lambda_1 \ldots \lambda_m$ are the corresponding eigenvalues. Eigenvalues and eigenvectors are sorted according to the largest eigenvalue in descending order. The (*m-n)* principal components with the highest eigenvalues are organized into a matrix where *n* is the number of principal components with the lowest eigenvalues.

$$\mathbf{P} = [\mathbf{q}_1 \ldots \mathbf{q}_{m-n}] \in \mathcal{R}^{m,m-n} \tag{5}$$

The principal components explain most of the variability in the data under normal operating conditions. The remaining principal components which contain least variability are grouped into the n residual eigenvectors.

$$\widetilde{\mathbf{P}} = [\mathbf{q}_{m-n+1} \ldots \mathbf{q}_m] \in \mathcal{R}^{m,n} \tag{6}$$

P are the principal components and **t** are the scores or projections. Now, the sensor measurement values of matrix **x*** can be expressed in terms of principal components **t** and residual components $\tilde{\mathbf{t}}$:

$$\mathbf{x}^* = \mathbf{Pt} + \widetilde{\mathbf{P}}\tilde{\mathbf{t}} \tag{7}$$

Qin and Li (1999) state that "[…] when *m*-*n* is chosen to be the number of equality constraints on the measurement vector, then the *n* residual components should contain random noise only." Equality constraints are the direct relationships between the physical variables measured by the sensor. For example, the level in a tank is directly related to the outflow. This means that *n* has to be adjusted for each process according to the relationships of the measurements.

Multiplying Eq. (7) from the left with $\widetilde{\mathbf{P}}^T$ from the right as follows

$$\widetilde{\mathbf{P}}^T\mathbf{x}^* = \widetilde{\mathbf{P}}^T\mathbf{Pt} + \widetilde{\mathbf{P}}^T\widetilde{\mathbf{P}}\tilde{\mathbf{t}} \tag{8}$$

gives the process model equation as

$$\widetilde{\mathbf{P}}^T\mathbf{x}^* = \tilde{\mathbf{t}} \tag{9}$$

because $\widetilde{\mathbf{P}}$ and **P** are orthogonal so that $\widetilde{\mathbf{P}}^T\mathbf{Pt} = \mathbf{0}$ and $\widetilde{\mathbf{P}}^T\widetilde{\mathbf{P}} = \mathbf{I}$. This can be expressed as the process model equation:

$$\mathbf{Bx}^*[k] = \mathbf{e}^*[\mathrm{k}] \tag{10}$$

with $\mathbf{B} = \widetilde{\mathbf{P}}^T$ as the matrix representing the model the and $\mathbf{e}^* = \tilde{\mathbf{t}}$ as the model residual which contains measurement noise, process noise and model errors.

The assumption for this detection method is that the signal is sufficiently stationary. In case of transients affecting the process, a dynamic process model should be used. To alleviate the effect of temporary transients, low pass filtering can be applied such as EWMA filters. Now that the process model has been derived for normal operating conditions the residuals **e**[*k*] can be computed from the operating data **x**[*k*] during which time a fault may be present. The fault detection index is defined as

$$d[k] = \mathbf{e}^T[k]R_e^{-1}\mathbf{e}[k] \tag{11}$$

where the inverse covariance matrix of $\mathbf{e}^*[k]$ is

$$R_e^{-1} = E\{\mathbf{e}^*[k]\mathbf{e}^{*T}[k]\} = \begin{bmatrix} \sigma^2_{e_1^*e_1^*} & \cdots & \sigma^2_{e_1^*e_m^*} \\ \vdots & \ddots & \vdots \\ \sigma^2_{e_m^*e_1^*} & \cdots & \sigma^2_{e_m^*e_m^*} \end{bmatrix} \tag{12}$$

with

$$\sigma^2_{x_i^*x_j^*} = \frac{1}{N}(e_i^*[k] - \bar{e}_i^*)(e_j^*[k] - \bar{e}_j^*). \tag{13}$$

We here deal with **e*** and not **e** as this refers to the measured value. If *d*[*k*] exceeds a predefined limit based on the Chi-square distribution, then a fault is detected. Again, this interpretation is based on the assumption that **x** is normally distributed.

In PCA, the squared prediction error (SPE) along with Hotelling's T^2 statistic is used to express the deviation from the PCA model with one single value. Here, the SPE can be used for fault detection and is defined as:

$$SPE[k] = \mathbf{x}^T[k]\widetilde{\mathbf{P}}\widetilde{\mathbf{P}}^T\mathbf{x}[k] \qquad (14)$$

In the software implementation package, $d[k]$ is used instead of SPE because the SPE assumes a normal distribution of the measurements x. The PCA model is, strictly speaking, not required for fault detection in the case study provided because the sensor faults of the analytical sensors are straightforward. The sensor output drops to zero when the sensor is faulty, that is, it fails entirely. The same PCA model, however, is vital for reconstructing the sensor signal when a fault has been detected.

Once a fault has been detected it is often required to replace the incorrect measurement with a better estimate. This estimate should be based on the other measurements where no fault is present (Dunia and Qin, 1998). In the industrial solution, the PCA model under normal operating condition is compared to the collected run data.

Before this can be done the question of reconstructability has to be answered, that is: Is sufficient variation of a specific process variable during normal operation explained with the PCA model? If not, then the PCA model cannot be used to reconstruct that particular variable. Here, a measure for variability is the absolute value of the loads in the principal components in the PCA decomposition. A threshold has to be defined above which the variable is deemed reconstructable. In the industrial solution the threshold is set to 0.5 by default but can be adjusted by the user to a different value between 0 and 1. Reconstructability and selection of the threshold is carried out during training of the PCA model with normal operating data.

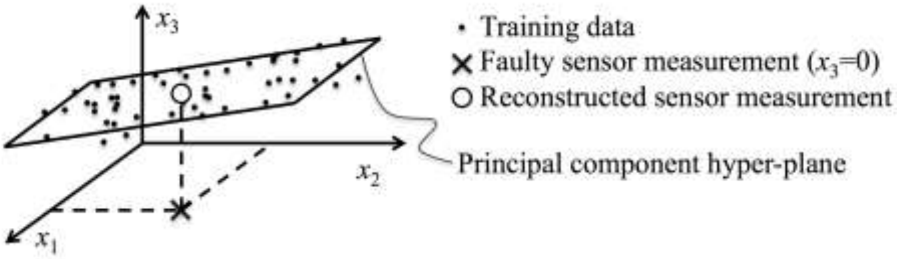
x_3
• Training data
✕ Faulty sensor measurement (x_3=0)
○ Reconstructed sensor measurement
x_2
Principal component hyper-plane
x_1

Figure 11. Geometrical interpretation of the sensor validation reconstruction.

If the reconstructability criterion is fulfilled then a faulty sensor measurement can be corrected using the normal operating condition PCA model. The idea is to minimize the distance between the training data set and the faulty measurement vector where $x_j[k]$ is known to be incorrect [Kerschen et al 2005].

$$\min_{x_j} J \ \text{ with} J = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \qquad (15)$$

Eq. 15 is a geometrical interpretation of this correction is shown in Figure 11. In a sensor measurement vector the third variable x_3 is incorrectly recorded as zero. A PCA model with two principal components has been generated using the training data. In the reconstruction step the closest point in the PC hyper-plane is found by the optimization problem of Eq. 15.

References

Bauer, M., & Craig, I. K. (2008). Economic assessment of advanced process control–a survey and framework. *Journal of process control*, *18*(1), 2-18.

Brooks, K. S., & Koorts, R. (2017). Model Predictive Control of a Zinc Flotation Bank Using Online X-ray Fluorescence Analysers. *IFAC-PapersOnLine, 50*(1), 10214-10219.

Chiang, L.H., Russell, E.L. and Braatz, R.D., (2000). Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and intelligent laboratory systems*, *50*(2), pp.243-252.

Chouaib, C., Mohamed-Faouzi, H., & Messaoud, D. (2013, June). Adaptive kernel principal component analysis for nonlinear dynamic process monitoring. *In Control Conference (ASCC), 2013 9th Asian*. IEEE.

Crowe, C. M. (1996). Data reconciliation—progress and challenges. *Journal of process control*, *6*(2-3), 89-98.

Da, R., & Lin, C. F. (1995). Sensor failure detection with a bank of kalman filters. In *American Control Conference, Proceedings of the 1995* (Vol. 2, pp. 1122-1126). IEEE.

Ding, S., Zhang, P., Ding, E., Yin, S., Naik, A., Deng, P., & Gui, W. (2010). On the application of PCA technique to fault diagnosis. *Tsinghua Science & Technology*, 15 (2), 138-144.

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.

Dunia, R., & Qin, S. J. (1997). Multi-dimensional fault diagnosis using a subspace approach. In *American Control Conference.*

Dunia, R., & Qin, S. J. (1998). Joint diagnosis of process and sensor faults using principal component analysis. *Control Engineering Practice*, 6 (4), 457-469.

Ge, Z., & Song, Z. (2012). *Multivariate statistical process control: Process monitoring methods and applications*. Springer Science & Business Media.

Huang, B., & Shah, S. L. (2012). *Performance assessment of control loops: theory and applications*. Springer Science & Business Media.

Huang, Y., Gertler, J., & McAvoy, T. J. (2000). Sensor and actuator fault isolation by structured partial PCA with nonlinear extensions. *Journal of Process Control*, 10 (5), 459-469.

Jelali, M. (2012). *Control performance management in industrial automation: assessment, diagnosis and improvement of control loop performance*. Springer Science & Business Media.

Jiang, L. (2011). *Sensor fault detection and isolation using system dynamics identification techniques*. Doctoral dissertation, The University of Michigan.

Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, *33* (4), 795-814.

Kerschen, G., De Boe, P., Golinval, J.C. & Worden, K (2005). Sensor validation using principal component analysis. *Smart Materials and Structures* (14), 36-42.

Kruger, U., & Xie, L. (2012). *Advances in statistical monitoring of complex multivariate processes: with applications in industrial process control*. John Wiley & Sons.

Kullaa, J. (2013). Detection, identification, and quantification of sensor fault in a sensor network. *Mechanical Systems and Signal Processing*, *40* (1), 208-221.

Li, W., & Shah, S. (2002). Structured residual vector-based approach to sensor fault detection and isolation. *Journal of Process Control*, *12* (3), 429-443.

Lieftucht, D., Kruger, U., Irwin, G. W., & Treasure, R. J. (2006). Fault reconstruction in linear dynamic systems using multivariate statistics. *IEE Proceedings-Control Theory and Applications*, *153* (4), 437-446.

Lieftucht, D., Völker, M., Sonntag, C., Kruger, U., Irwin, G. W., & Engell, S. (2009). Improved fault diagnosis in multivariate systems using regression-based reconstruction. *Control Engineering Practice*, *17*(4), 478-493.

Lo, C. (2014). *Efficient Sensor Fault Diagnosis in Wireless Sensor Networks*. Doctoral dissertation, The University of Michigan.

Mehranbod, N., Soroush, M., & Panjapornpon, C. (2005). A method of sensor fault detection and identification. *Journal of Process Control*, *15*(3), 321-339.

Padilla, M., & Choinière, D. (2015). A combined passive-active sensor fault detection and isolation approach for air handling units. *Energy and Buildings*, *99*, 214-219.

Qin, S. J. (2014). Process data analytics in the era of big data. *AIChE Journal*, *60*(9), 3092-3100.

Qin, S. J., & Guiver, J. P. (2003). *U.S. Patent No. 6,594,620*. Washington, DC: U.S. Patent and Trademark Office.

Qin, S. J., & Li, W. (1999). Detection, identification, and reconstruction of faulty sensors with maximized sensitivity. *AIChE Journal*, *45* (9), 1963-1976.

Qin, S. J., & Li, W. (2001). Detection and identification of faulty sensors in dynamic processes. *AIChE Journal*, *47* (7), 1581-1593.

Simani, S., Fantuzzi, C., & Beghelli, S. (2000). Diagnosis techniques for sensor faults of industrial processes. *IEEE Transactions on Control Systems Technology*, *8*(5), 848-855.

Upadhyaya, B. R., & Eryurek, E. (1992). Application of neural networks for sensor validation and plant monitoring. *Nuclear Technology*, *97* (2), 170-176.

Xu, X., Hines, J. W., & Uhrig, R. E. (1999). Sensor validation and fault detection using neural networks. In *Proc. Maintenance and Reliability Conference (MARCON 99)*, 10-12.

Zhu, D., Bai, J., & Yang, S. X. (2009). A multi-fault diagnosis method for sensor systems based on principle component analysis. *Sensors*, *10* (1), 241-253.