I&EC research
Industrial & Engineering Chemistry Research

ARTICLE

pubs.acs.org/IECR

# Software Accuracy-Based Sensor Network Design and Upgrade in Process Plants

DuyQuang Nguyen, Sanjay K. Sugumar, and Miguel J. Bagajewicz*

School of Chemical, Biological and Materials Engineering, University of Oklahoma, Norman, Oklahoma 73069

**ABSTRACT:** In this paper we present a new method for linear and nonlinear sensor network design with applications to variable estimation based on the use of accuracy and residual accuracy. Accuracy has been shown recently to be a better substitute for existing data-reconciliation network properties, such as precision, residual precision, gross error detectability and resilience as pointed out in our previous works (the first two references). We explore the numerical difficulties associated to different ways of calculating accuracy and different methods used to optimize the cost.

## 1. INTRODUCTION

Improved process monitoring via data reconciliation and appropriate gross error detection are achieved and/or improved by proper systematic location of sensors in a process plant. This idea gave rise to the problem of sensor location, which has evolved through several conceptual phases in the last 30 years. In all cases, the notion was to take advantage of the process relations, presented in the form of a mathematical model, to infer the value of certain variables, called key variables, by using measurements on other variables to aid in the process. Most of the published work aimed to find an optimal sensor network that satisfies the specifications on the precision level, observability, and redundancy of measurement of variables of interest, the challenges being finding efficient computational methods to solve the problem. We now highlight the most relevant work (the reader is referred to the textbook by Bagajewicz[3] for a complete list up to year 2000). Vaclavek and Loucka[4] were the first to explore the problem and sought to achieve the observability requirements of a multicomponent network. Kretsovalis and Mah[5] minimized a weighted sum of estimation error and measurement cost using a combinatorial search algorithm. Madron and Veverka[6] used multiple Gauss Jordan elimination to achieve observability of all key variables at minimum sensor cost. Bagajewicz[7] was the first to formulate a sensor network problem as a mixed-integer programming model (the model is MINLP) using binary variables to indicate whether a variable/stream is measured or not, sought to obtain minimum sensor cost, which was later reformulated into a MILP problem by using the notion of fake sensors for unmeasured variables.[8] Other efforts using tree search methods include the work by Bagajewicz and Sanchez.[9−12] Chmielewski et al.[13] showed that an unmeasured variable can be modeled in a data reconciliation formulation using a fake sensor with very high variance and used branch and bound method with linear matrix inequalities (LMI) transformation to obtain a solution. Most recently, Gala and Bagajewicz[14,15] presented a branch and bound method based on cutsets of a process graph for solving linear problems. This method has been proven to be remarkably faster, especially after a decomposition technique is used. Later, Nguyen and Bagajewicz[16] extended Gala and Bagajewicz's method to solve nonlinear problems, in which the tree search is built using model equations instead of cutsets. In other efforts, genetic algorithms were used by Sen et al.,[17] Carnero et al.,[18,19] and Heyen et al.[20] To this date, only the branch and bound methods that were used by Bagajewicz,[7] Gala and Bagajewicz,[14] Nguyen and Bagajewicz,[16] and Chmielewski et al.[13] guarantee optimality.

Ali and Narasimhan[21] introduced the concept of reliability of estimation of variables and proposed a graph-theoretic algorithm to maximize the reliability in linear systems, which was later extended to bilinear systems.[22] Raghuraj et al.[23] and Bhushan and Rengaswamy[24−27] designed a sensor network using fault diagnosis criteria. The authors used graph-theoretic approaches to warrant satisfactory process faults detection capability of the network, or maximize the reliability of the fault monitoring system. Bagajewicz et al.[28] designed a sensor network for simultaneous process monitoring and fault detection/resolution purposes using a MILP formulation.

All the published work on a process monitoring-focused sensor network considered precision, residual precision, error detectability, and resilience as performance targets. Recognizing that these targets do not represent best the network performance as far as its ability to filter gross errors, Bagajewicz[1] introduced the concept of software accuracy. He also explained its relationship to the aforementioned performance measures. It makes sense then to show how software accuracy as well as residual software accuracy, a concept we are introducing in this paper, might perform for sensor network design.[2]

The paper is organized as follows: We first review briefly the concept of software accuracy, we then introduce the concept of residual software accuracy. We later discuss a few issues about the theory of equivalency of gross errors, which was introduced by Bagajewicz and Jiang,[29] and its impact on the calculation of accuracy. We then move on to discuss issues about unboundedness and how

**Special Issue:** Puigjaner Issue

**Received:** March 9, 2010
**Accepted:** June 8, 2010
**Revised:** June 1, 2010
**Published:** June 25, 2010

dx.doi.org/10.1021/ie100566m | *Ind. Eng. Chem. Res.* 2011, 50, 4850–4857

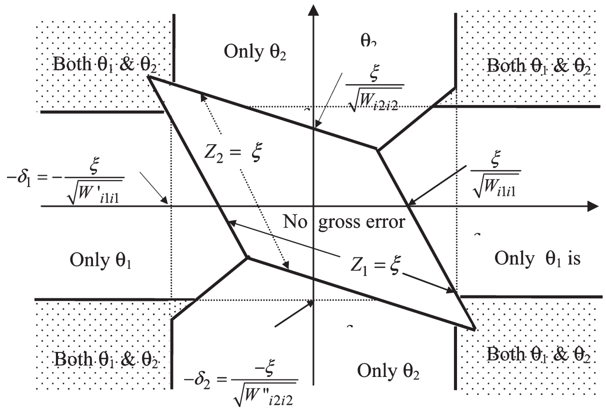

Both θ1 & θ2
Only θ2
θ2
$\frac{\xi}{\sqrt{W_{i2i2}}}$
Both θ1 & θ2
$Z_2 = \xi$
$-\delta_1 = -\frac{\xi}{\sqrt{W'_{i1i1}}}$
$\frac{\xi}{\sqrt{W_{i1i1}}}$
No gross error
Only θ1
$Z_1 = \xi$
Only θ1 is
Both θ1 & θ2
$-\delta_2 = \frac{-\xi}{\sqrt{W''_{i2i2}}}$
Only θ2
Both θ1 & θ2

**Figure 1.** Regions of detected and undetected gross errors.[32]

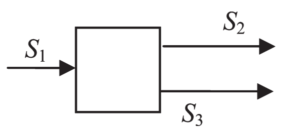


**Figure 2.** Example of equivalency of gross errors.

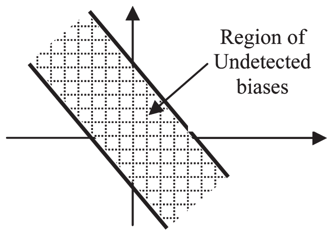
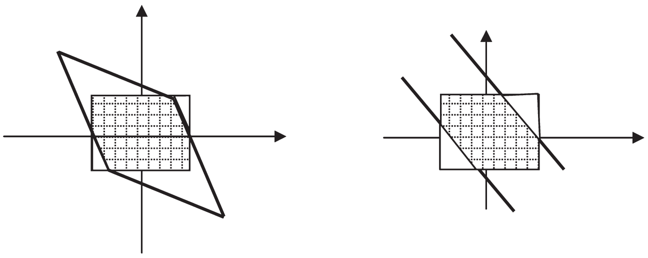


**Figure 3.** Illustration of biases equivalency.



**4a**. Normal case

**4b**. Equivalency of gross errors

**Figure 4.** Box and rhombus constraints defining undetected biases.

to handle them. We also review stochastic accuracy. We finish presenting two examples, one for a linear case and the other for a nonlinear case, both solved for different specifications and using different methods.

## 2. SOFTWARE ACCURACY

Accuracy has been defined as precision (standard deviation) plus absolute value of systematic bias.[30] Since true value of bias is hardly known unless hardware redundancy exists that enables the quantification of the bias, Bagajewicz[1] proposed to use the bias induced by undetected errors when data reconciliation and a sort of gross error detection techniques are used, called the induced bias. The associated accuracy is termed software accuracy ($\hat{a}_i$) and is defined as summation of precision and induced bias, that is $\hat{a}_i = \hat{\sigma}_i + \delta_i^*(p)$, where $\hat{\sigma}_i^2$ is the variance of the estimator and $\delta_i^*(p)$ the induced bias for a level of confidence $p$, which depends on undetected biases in the system. Bagajewicz[1] proposed to use the maximum possible value of undetected induced bias and derived formulas to calculate this maximum value assuming the maximum power measurement test is used to detect gross error. Another approach, which is the more realistic one, is calculating the induced bias as the expected value of all possible values. This was done via Monte Carlo simulation tool by Bagajewicz and Nguyen.[2]

## 3. GROSS ERROR EQUIVALENCY AND BOUNDEDNESS

Bagajewicz et al.[31] pointed out that the accuracy can undertake unbounded values. To explain this, they resorted to equivalency theory.[29] They pointed out that the region of undetected gross errors is a hyper-rhombus.[32] Such a rhombus is shown in Figure 1 for two gross errors ($n_T = 2$). The rhombus is composed of positive and negative constraints of the problem that arises from the absolute value. We recognize that the solution lies in one of the four corners, which depends on how the two gross errors in question contribute to the induced bias.

Bagajewicz et al.[31] also illustrated how equivalency theory[29] can explain the fact that the maximum induced bias can be unbounded, that is, two errors can cancel each other when they have opposite signs. This is illustrated in Figure 3 for the system shown in Figure 2. Indeed, assume that one wants to compute the induced bias in stream $S_1$ by considering two biases in $S_2$ and $S_3$. This leads to two parallel lines and therefore, any combination of errors in between the lines will go undetected by the maximum power test.

This phenomenon is not unlikely. In fact, it can take place often for two gross errors, as Bagajewicz et al.[31] illustrated for a small case. Gross errors are not, however, unbounded. If a bias in a measurement passes a certain threshold, which is usually a certain percentage of the normal value of the variable, by common sense the operators can tell that there is bias in the measurement. The threshold above which the bias is detected by the operators' judgment is used as the (upper) limit for bias. Figure 4 shows such limits for both cases.

Thus, the accuracy calculations in eq 2 need to be made using additional box constraints on each variable.

$$-\delta_{\max\ ,s} \le \delta^{(p)}_{\text{crit},s} \le \delta_{\max\ ,s} \qquad \forall\ s \qquad (1)$$

## 4. ACCURACY UNDER A LOWER NUMBER OF GROSS ERRORS

Assume that the number of measurements is kept constant, and instead of $k$ gross errors one has a smaller number of gross errors. Then (i) the power to detect the smaller number of biases (if any) decreases (the rhombus in Figure 1 expands and the maximum values of undetected biases increase) and (ii) the coefficient $(\text{SW})_{is}$ changes in value and probably in sign as well.

The first effect increases the accuracy value while the other two effects may increase or decrease the accuracy because the induced bias (hence the accuracy) depends simultaneously on the value and sign of both the undetected biases and the coefficients $(\text{SW})_{is}$, not to mention that a different number of undetected biases may be considered in the two cases (before and after removal of measurements).

Thus, in trying to design a system for accuracy one is faced with the following issue: When checking for accuracy under the presence of $n_T$ gross errors it makes sense to also check for the accuracy under the presence of any number of errors smaller than $n_T$, simply because when $n_T$ errors are expected, a smaller number is as likely. This is the approach taken in this paper.

## 5. STOCHASTIC ACCURACY

Bagajewicz[1] recognized the extreme condition that this induced bias represents, and proposed to look for a much realistic estimate. He proposed the new concept of stochastic accuracy, which is based on the calculation of the expected values of undetected gross errors instead of the maximum ones as presented above. To address the issue of providing a more realistic value, Bagajewicz and Nguyen[2] showed methods to calculate stochastic accuracy using Monte Carlo simulation. In that work, Monte Carlo simulation was used to simulate the status (failed, functioning, or being repaired) of all sensors in the system. The sampling procedure is briefly described as follows.[16]

Failure times and bias sizes for every sensor in the system are sampled and recorded until the end of time horizon is reached. The sensor reliability function is used to sample sensor's failure time and the distribution function of bias is used to sample bias size of the measurement.

At each failure time, the MP measurement test is performed and the sensors that are detected being biased are singled out.

If the MP measurement cannot detect any bias, no action is needed and the next time interval is then investigated. Otherwise (the MP flags the presence of biases), for each sensor with detected bias, the failure events are updated starting from that time interval. The sensors with detected bias are repaired, then they resume work and the next failure event is sampled (if the time is still within time horizon).

With the information obtained from the Monte Carlo simulation, the accuracy within a time interval is calculated according to the status of sensors within that interval, for example if all sensors are at normal state, then accuracy is equal to precision value. The more common case is that some sensors are taken off-service for repair while some others contain undetected bias, then accuracy is equal to residual precision plus induced bias. The expected value of accuracy ($E[a]$) is then calculated as the mean value of the accuracy values within time intervals ($a_i$) with weight being the duration of time intervals ($T_i$), that is, $E[a] = {}^1/_T\Sigma^i a_i T_i$, where $T$ is the simulation time horizon (note that $T = \Sigma T_i$) and $i$ is the index of time interval.

Bagajewicz and Nguyen[2] showed that the expected value of accuracy is a realistic one while the maximum one is too conservative. Even in the case that the gross errors equivalency phenomenon is observed, which leads to infinite maximum undetected bias, the expected value is still a finite and reasonable one. In fact, if bias is normally distributed with zero mean and small variance, the expected values of bias and the accuracy are small no matter how bad the system's capability to detect bias is. Thus, bounds (box constraints) on values of undetected biases are not needed. Another point to make is that all possibilities are realized in the Monte Carlo simulation: from extremes cases such as all sensors are at normal state or all contain undetected biases to realistic and more common cases like some are functioning properly, some contain undetected biases, while some are taken off-line for repair. The accuracy associated with the mentioned realistic cases is the residual accuracy of order $k > 0$. Thus, stochastic accuracy encompasses all the possible values of accuracy, and residual accuracy needs not to be calculated separately.

One problem with the Monte Carlo simulation approach is that it takes a long time to obtain accurate results. As shown in Bagajewicz and Nguyen,[2] the number of simulation trials needs be at least 1000 in order for the expected value to converge. In the Madron example shown below, if 1000 simulations are executed to compute the expected value, a network consisting of 12 sensors takes roughly 30 s to compute while the network consisting of 24 sensors take roughly 2 min 30 s (on a 2.8 GHz CPU, 1028 MB RAM PC). To be able to solve large scale problems, the time to compute expected value of accuracy in each node of the search tree needs to be limited to not more than a few seconds. To reduce computational time, we propose two solutions: (i) reduce the number of simulation attempts to 100 at the expense of degraded quality of solution, (ii) use an approximate method to calculate expected value of accuracy (described in the next section).

## 6. APPROXIMATE METHOD TO CALCULATE STOCHASTIC ACCURACY

We resort to the same approximate method previously used[32] to calculate expected value of accuracy. First, we discuss the expected value of accuracy, then we present the approximation scheme to calculate this value.

The formulas to calculate the expected value of accuracy are derived at the same fashion as the equations to calculate the downside expected financial loss presented in Nguyen et al., 2006. More specifically, one needs to accommodate the effect of all the possibilities regarding the biases sizes and biases' locations on the value of the accuracy of interest, then one performs integration over all possible biases sizes (at known biases' location) to obtain the expected value corresponding to this known biases' location. Then, the expected value of accuracy is found as the "average" value of these expected values at all possible biases' locations. The expected value corresponding to a set of biases at known location, $E[a_i]|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$, is then multiplied by the fraction of time the system is in that specific state (i.e., containing that set of biases), $\Psi|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$, the summation of these products gives the final expected value of accuracy (hence the expected value of accuracy is the average value of the expected values at known biases location with weight being the associated fractions of time). This is shown next:

$$E[a_i] = \Psi^0 E[a_i]|^0 + \sum_{\forall\theta_1} \Psi|^1_{\theta_1} E[a_i]|^1_{\theta_1} + \sum_{\forall\theta_1,\theta_2} \Psi|^2_{\theta_1,\theta_2} E[a_i]|^2_{\theta_1,\theta_2} + \ldots + \sum_{\forall\theta_1,\theta_2,\ldots,\theta_n} \Psi|^n_{\theta_1,\theta_2,\ldots,\theta_n} E[a_i]|^n_{\theta_1,\theta_2,\ldots,\theta_n} \quad (2)$$

The fraction of time the system is in a specific state $\Psi|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$ can only be known through simulation, which consumes computational time. So we use another measure to replace the $\Psi|_{\theta_1,\theta_2,\ldots,\theta n}{}^n$, which is the probability that the system is in the specific state at time $t$:$\varphi(t)|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$; the expected value is then calculated by

$$E(t)[a_i] = \varphi(t)^0 E[a_i]|^0 + \sum_{\forall\theta_1} \varphi(t)^1_{\theta_1} E[a_i]|^1_{\theta_1} + \sum_{\forall\theta_1,\theta_2} \varphi(t)|^2_{\theta_1,\theta_2} E[a_i]|^2_{\theta_1,\theta_2} + \ldots + \sum_{\forall\theta_1,\theta_2,\ldots,\theta_n} \varphi(t)|^n_{\theta_1,\theta_2,\ldots,\theta_n} E[a_i]|^n_{\theta_1,\theta_2,\ldots,\theta_n} \quad (3)$$

The probabilities $\varphi(t)|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$ can be readily calculated using the failure probability of each sensor at time t, $f_i(t)$, which is

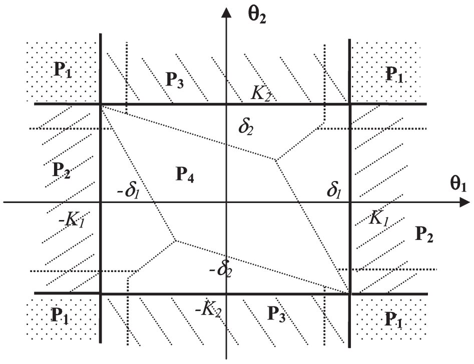


**Figure 5.** Regions used in analytical calculation.

the probability that the sensor $i$ fails at time $t$ (hence the probability that the sensor is functioning properly is $1 - f_i(t)$). For example, the probability that the system contains two biases $\theta_1$,$\theta_2$ in the two sensors 1 and 2 $\varphi(t)|_{\theta_1,\theta_2}{}^2$ is the product of the probabilities that the two sensors 1, 2 fail at time $t$ while the others are working properly, that is

$$\varphi(t)|^2_{\theta_1,\theta_2} = f_1(t)f_2(t)\prod_{s\neq 1,s\neq 2}[1 - f_s(t)]$$

The expected value $E[a_i]$ defined above is equivalent to the stochastic accuracy presented in section 6 but the expected value $E(t)[a_i]$ is not because the reliability measures $f_i(t)$ and $\varphi(t)|_{\theta_1,\theta_2,\ldots,\theta n}{}^n$ and the accuracy value $E(t)[a_i]$ depends on time: the longer the age of the sensors, the worse the accuracy value $E(t)[a_i]$. Despite the difference, the two calculation procedures should give values at the same order of magnitude, which is confirmed by calculation results.

The expected value corresponding to a set of biases at known location (e.g., $E[a_i]|_{\theta_1,\theta_2}{}^2$) is calculated using the same approximation scheme as the one used in Nguyen et al.[32] More specifically, to calculate $E[a_i]|_{\theta_1,\theta_2}{}^2$ the space of variable is partitioned into four regions shown in Figure 5.

With the partitioned space shown in Figure 5, the value $E[a_i]|_{\theta_1,\theta_2}{}^2$ is then calculated by

$$\begin{aligned}
E[a_i]|^2_{\theta_1,\theta_2} &= \textstyle\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\\
&= \textstyle\int_{\mathrm{P1}} F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 + \int_{\mathrm{P2}} F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 + \int_{\mathrm{P3}} F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\\
&\quad + \int_{\mathrm{P4}} F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\\
&= \{\int_{-\infty}^{-K_1} + \int_{K_1}^{\infty}\}\{\int_{-\infty}^{-K_2} + \int_{K_2}^{\infty}\}F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\ (E_1)\\
&\quad + \{\int_{-\infty}^{-K_1} + \int_{K_1}^{\infty}\}\int_{-K_2}^{K_2} F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\ (E_2)\\
&\quad + \int_{-K_1}^{K_1}\{\int_{-\infty}^{-K_2} + \int_{K_2}^{\infty}\}F\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\ (E_3)\\
&\quad + \int_{-K_1}^{K_1}\int_{-K_2}^{K_2} F(\theta_1,\theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\ (E_4)\}
\end{aligned} \tag{4}$$

The term $E_1$ is calculated analytically while $E_2$, $E_3$, and $E_4$ are calculated approximately at the same fashion as done in Nguyen et al.[32]

We limit the calculation of $E[a_i]|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$ up to two biases; The calculations of $E[a_i]|_{\theta_1,\theta_2,\ldots,\theta_n}{}^n$ for more than two biases consume significant time, so we extrapolate the value for more than two biases using the corresponding values when no more than two biases are present ($E[a_i]|^0$, $E[a_i]|_{\theta_1}{}^1$ and $E[a_i]|_{\theta_1,\theta_2}{}^2$). This introduces another layer of approximation with the benefit of reduced computational time.

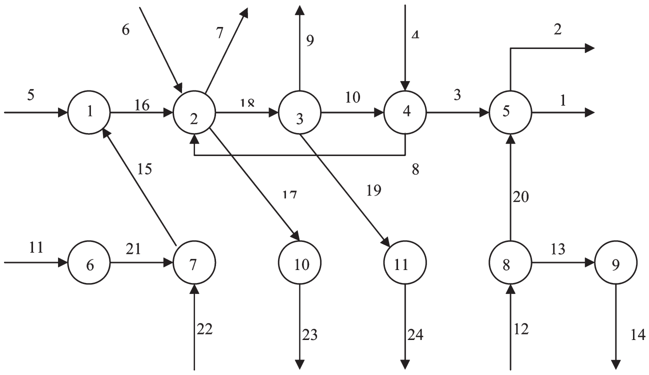


**Figure 6.** Madron and Veverka[33] example.

The failure probability of sensor $f_i(t)$ takes the default value of 0.2 for all sensors, but it can be changed by the user. The biases are assumed to follow normal distributions $h_k(\theta_k,\bar{\delta}_k,\rho_k)$ with zero mean and variance of 4.0 (default values).

## 7. SENSOR NETWORK DESIGN METHOD

The optimization model to design minimum cost sensor network using accuracy as a constraint is as follows:

$$\begin{aligned}
&\min \sum_{\forall i} c_i q_i\\
&\text{s.t.}\\
&a_i(q) \le a_i^* \quad \forall i \in M_S\\
&q_i = 0, 1 \quad \forall\ i
\end{aligned} \tag{5}$$

where $q_i$, an element of vector $q$, is a binary variable indicating that a sensor is used to measured variable $i$, $c_i$ is the cost of such a sensor, and $M_s$ represents the set of variables where a performance specification is required (variables of interest or "key" variables). This performance specification can be precision of the estimators. Thus, $a_i(q)$ is the accuracy of variable $i$ corresponding to the set of sensors represented by $q$ and $a_i^*$ the corresponding threshold value. Other specifications could include residual accuracy or reliability.

We use the tree enumeration method using a list of fundamental units, which can be variables,[7] cutsets,[14] or equations.[16] The inverse tree method using individual variables[16] can also be used, depending on the requirements on the system.

At each node, software accuracy is evaluated using one of the three options: (i) accuracy defined by maximum undetected bias, (ii) stochastic accuracy using Monte Carlo simulation, (iii) expected value of accuracy using the approximate method.

The accuracy calculation for the maximum undetected bias definition is based on two inputs: (1) the positions of the sensors (given by a binary vector $q$); (2) the maximum number of gross errors ($n_{\mathrm{T}}$) the user is willing to consider in the definition of accuracy.

Assume that $n$ is the number of sensors in the node (the number of nonzero elements in $q$). Then, if $n_{\mathrm{T}} > n$, the maximum number of gross errors is assumed to be $n$,

Table 1. Flow Rate, Precision, and Sensor Costs for Example 1 (Madron and Veverka[33])

| stream | flow | sensor cost | sensor precision (%) | stream | flow | sensor cost | sensor precision |
|---|---|---|---|---|---|---|---|
| 1 | 140 | 19 | 2.5 | 13 | 10 | 12 | 2.5 |
| 2 | 20 | 17 | 2.5 | 14 | 10 | 12 | 2.5 |
| 3 | 130 | 13 | 2.5 | 15 | 90 | 17 | 2.5 |
| 4 | 40 | 12 | 2.5 | 16 | 100 | 19 | 2.5 |
| 5 | 10 | 25 | 2.5 | 17 | 5 | 17 | 2.5 |
| 6 | 45 | 10 | 2.5 | 18 | 135 | 18 | 2.5 |
| 7 | 15 | 7 | 2.5 | 19 | 45 | 17 | 2.5 |
| 8 | 10 | 6 | 2.5 | 20 | 30 | 15 | 2.5 |
| 9 | 10 | 5 | 2.5 | 21 | 80 | 15 | 2.5 |
| 10 | 100 | 13 | 2.5 | 22 | 10 | 13 | 2.5 |
| 11 | 80 | 17 | 2.5 | 23 | 5 | 13 | 2.5 |
| 12 | 40 | 13 | 2.5 | 24 | 45 | 13 | 2.5 |

Table 2. Results for Example 1

|  | design case | stochastic accuracy (Monte Carlo simulation method) | accuracy calculated using the approximate method |
|---|---|---|---|
| requirements | 1 | six key variables 1, 4, 6, 11, 12, 24; thresholds: 30% |  |
|  | 2 | six key variables 1, 4, 5, 11, 14, 24; thresholds: 15% |  |
| optimal sensor network, measured streams | 1 | 1, 4, 6, 12, 21, 24 | 1, 4, 6, 12, 21, 24 |
|  | 2 | 1, 4, 5, 11, 13, 14, 15, 16, 24 | 1, 4, 5, 14, 21, 24 |
| number of sensors in optimal network | 1 | 6 | 6 |
|  | 2 | 9 | 6 |
| optimal cost | 1 | 82 | 82 |
|  | 2 | 146 | 96 |
| accuracies of key variables | 1 | $S_1 = 4.0$, $S_4 = 8.1$, $S_6 = 8.4$, $S_{11} = 5.5$, $S_{12} = 8.8$, $S_{24} = 8.2$ | $S_1 = 3.1$, $S_4 = 4.3$, $S_6 = 4.1$, $S_{11} = 3.4$, $S_{12} = 4.2$, $S_{24} = 3.9$ |
|  | 2 | $S_1 = 3.9$, $S_4 = 7.0$, $S_5 = 13.5$, $S_{11} = 4.7$, $S_{14} = 2.4$, $S_{24} = 6.7$ | $S_1 = 3.0$, $S_4 = 4.1$, $S_5 = 8.9$, $S_{11} = 3.4$, $S_{14} =8.9$, $S_{24} = 3.9$ |
| number of nodes explored | 1 | 5224 | 5224 |
|  | 2 | 58988 | 10275 |
| computational time | 1 | 1 h 24 min | 14 min 11 s |
|  | 2 | 33 h 11 min | 29 min 57 s |
| average computational time per node explored | 1 | 0.97 s | 0.16 s |
|  | 2 | 2.0 s | 0.17 s |

that is $n_T = n$. When $n_T < n$, all combination of $n_T$ biases in the $n$ sensors need to be evaluated.

The other two options to calculate accuracy need one input, which is the sensor placement (vector $q$) in each node, in addition to two user input parameters: the failure probability $f_i(t)$ or failure rate of sensor and the distributions of biases.

The branching and stopping criteria of all the methods discussed above are maintained the same as in previous work.[16]

## 8. EXAMPLES

We consider two examples: one middle scale linear problem and one industrial large scale nonlinear problem. The calculation methods are implemented on Fortran running on 2.8 GHz Pentium CPU, 1028 MB RAM PC computer.

**8.1. Example 1.** Consider the network proposed by Madron and Veverka[33] (Figure 6). Madron and Veverka[33] did not report flow rates and so the values shown in Table 1 were chosen from Bagajewicz[7] where the cost of the sensors is also given. Only flow rate variables are considered, hence the system is linear.

Two design case studies are considered for this example. The problem was solved using the tree enumeration method making use of cutsets and the decomposition technique.[14,15] In each node, the accuracies of the six key variables are calculated and compared with the threshold values. The accuracy is calculated using one of the three accuracy definitions: (i) stochastic accuracy obtained by Monte Carlo simulation, and (ii) stochastic accuracy obtained by the approximation method.

In the Montecarlo approach to stochastic accuracy, a failure event of sensor is assumed to follow exponential distribution with constant failure rate = 0.01/year; the number of simulation trials is 100. In the case of stochastic accuracy obtained using the approximate method, the failure probability of sensor is 0.2 for all sensors. In the last two cases, the biases are assumed to follow normal distributions with zero mean and a variance value of 4.0.

The following cases are considered:

*Design Case 1.* Flow rates of the following six streams are of interest with required accuracy threshold of 30%: 1, 4,

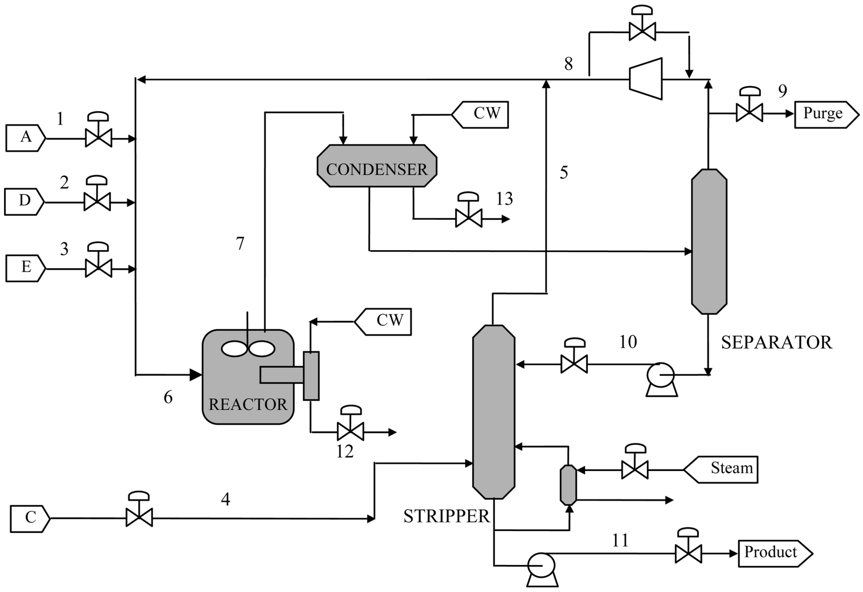


**Figure 7.** Process flow diagram of the Tennessee Eastman (following Ricker and Lee[34]).

**Table 3. Sensor Precision, Flow Rates, and Sensor Costs for the TE Process**

| stream no. | flow value | sensor precision | sensor cost | flow rate/ composition/ variable of interest | stream no. | flow value | sensor precision | sensor cost | flow rate/ composition/ variable of interest |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.8899 | 2 | 300 | $F_6$ | 24 | 0.023 | 2 | 730 | $y_{F,8}$ |
| 2 | 0.322 | 2 | 770 | $y_{A,6}$ | 25 | 0.048 | 2 | 740 | $y_{G,8}$ |
| 3 | 0.089 | 2 | 780 | $y_{B,6}$ | 26 | 0.023 | 2 | 750 | $y_{H,8}$ |
| 4 | 0.264 | 2 | 730 | $y_{C,6}$ | 27 | 0.33 | 2 | 720 | $y_{A,9}$ |
| 5 | 0.069 | 2 | 740 | $y_{D,6}$ | 28 | 0.138 | 2 | 730 | $y_{B,9}$ |
| 6 | 0.187 | 2 | 750 | $y_{E,6}$ | 29 | 0.24 | 2 | 740 | $y_{C,9}$ |
| 7 | 0.016 | 2 | 760 | $y_{F,6}$ | 30 | 0.013 | 2 | 750 | $y_{D,9}$ |
| 8 | 0.035 | 2 | 810 | $y_{G,6}$ | 31 | 0.186 | 2 | 760 | $y_{E,9}$ |
| 9 | 0.017 | 2 | 820 | $y_{H,6}$ | 32 | 0.023 | 2 | 770 | $y_{F,9}$ |
| 10 | 1.4752 | 2 | 300 | $F_7$ | 33 | 0.048 | 2 | 780 | $y_{G,9}$ |
| 11 | 0.272 | 2 | 750 | $y_{A,7}$ | 34 | 0.023 | 2 | 790 | $y_{H,9}$ |
| 12 | 0.114 | 2 | 760 | $y_{B,7}$ | 35 | 0.2586 | 2 | 200 | $F_{10}$ |
| 13 | 0.198 | 2 | 700 | $y_{C,7}$ | 36 | 0.002 | 2 | 700 | $X_{D,10}$ |
| 14 | 0.011 | 2 | 710 | $y_{D,7}$ | 37 | 0.136 | 2 | 710 | $X_{E,10}$ |
| 15 | 0.177 | 2 | 720 | $y_{E,7}$ | 38 | 0.016 | 2 | 720 | $X_{F,10}$ |
| 16 | 0.022 | 2 | 730 | $y_{F,7}$ | 39 | 0.472 | 2 | 720 | $X_{G,10}$ |
| 17 | 0.123 | 2 | 780 | $y_{G,7}$ | 40 | 0.373 | 2 | 730 | $X_{H,10}$ |
| 18 | 0.084 | 2 | 790 | $y_{H,7}$ | 41 | 0.2113 | 2 | 200 | $F_{11}$ |
| 19 | 0.33 | 2 | 780 | $y_{A,8}$ | 42 | 0.537 | 2 | 730 | $X_{G,11}$ |
| 20 | 0.138 | 2 | 770 | $y_{B,8}$ | 43 | 0.438 | 2 | 740 | $X_{H,11}$ |
| 21 | 0.24 | 2 | 760 | $y_{C,8}$ | 44 | 2.806 | 2 | 100 | $P_r$ |
| 22 | 0.013 | 2 | 750 | $y_{D,8}$ | 45 | 393.6 | 2 | 500 | $T_r$ |
| 23 | 0.186 | 2 | 740 | $y_{E,8}$ | 46 | 2.7347 | 2 | 100 | $P_s$ |
|  |  |  |  |  | 47 | 353.3 | 2 | 500 | $T_s$ |

6, 11, 12, and 24 (these streams are either input or output streams).

*Design Case 2.* Flow rates of the following six streams are of interest with required accuracy threshold of 15%: 1, 4, 5, 11, 14, and 24 (these streams are either input or output streams).

The results are shown in Table 2. Several conclusions are drawn from the results shown in Table 2 (and other testing results that were obtained but not shown here):

(1) The stochastic value captures the status of the system during the whole simulated time horizon, while the expected

Table 4. Results for the TE Example

| | stochastic accuracy (Monte Carlo simulations) | accuracy calculated using the approximate method |
|---|---|---|
| accuracy requirement | $S_{27}$ = 20%, $S_{29}$ = 27%, $S_{41}$ = 5%, $S_{42}$ = 8.5%, $S_{43}$ = 8.5% | |
| optimal sensor network, measured streams | 27, 35, 39, 40, 41, 42 | 27, 33, 41, 42, 43 |
| optimal cost | 3300 | 3170 |
| accuracies of key variables | $S_{27}$ = 6.7, $S_{29}$ = 9.3, $S_{41}$ = 1.8, $S_{42}$ = 1.8, $S_{43}$ = 4.7 | $S_{27}$ = 3.9, $S_{29}$ = 5.2, $S_{41}$ = 2.6, $S_{42}$ = 8.0, $S_{43}$ = 5.5 |
| number of nodes explored | 844 | 1439 |
| computational time | 6 h 31 min | 11 min 21 s |

value obtained by using eq 3 corresponds to a specific point in time since the probability of failure $f_i(t)$ depends on time. It can be said that the stochastic value is the average value in time dimension of the expected value. If the sensors are relatively new, the probability of failure is small and the expected value would be smaller than the stochastic value as confirmed by calculation results; as sensors age with time, the probability and the expected value increase. We have calculated to find that if the probability of failure $f_i(t)$ is in range 0.5−0.6, the expected value is comparable to the stochastic value.

(2) In general, adding sensors would improve the stochastic and the expected value of accuracy as seen in column 4: both of the two optimal solutions in two design cases contain sensors for the four streams 1, 4, 11, 24 and increasing the number of sensors from 6 (design case 1) to 9 (design case 2) improves the accuracy of the estimators for these four streams.

**8.2. Tennessee Eastman Process.** The flow diagram for the Tennessee Eastman (TE) process is shown in Figure 7 following the diagram shown in Downs and Vogel,[33] who first introduced the TE process as an industrial challenge problem. Table 3 below shows the flow values, sensor costs, and the precision of the sensors. The example has 47 variables and 28 nodes. The incidence matrix is obtained by linearizing the nonlinear system of equations as explained in Nguyen and Bagajewicz.[16]

Only one design case is considered for this example: software accuracy for streams 27, 29, 41, 42, 43 is requested with threshold values 20%, 27%, 5%, 8.5%, and 8.5%, respectively. The results are shown in Table 4.

For the case of stochastic accuracy, since we have to use a small number of simulation runs in the Monte Carlo simulation (100) to reduce computational time, the solutions found have to be verified using a larger number of simulation runs (at least 1000). The reported solution in option 2 (shown in column 2) is the third best solution located by the tree search procedure. The best solution and second best solution marginally satisfies the accuracy requirement for stream 42 using 100 Monte Carlo simulation runs. After verifying by using 1000 Monte Carlo simulation runs, these two solutions are found to be infeasible (not satisfying the accuracy requirement for stream 42) and have to be discarded. As already seen in example 1, in this example the option 2 asks for more sensors than the option 3, which is attributed to the fact that a relative small probability of failure (0.2) is used in calculating the expected value. Three key variables 27, 41, 42 are contained in all the solutions.

As software accuracy contains the undetected induced bias in its definition, the requirement on accuracy value encompasses the requirements on precision, gross errors delectability, and gross errors resilience. More specifically, a sensor network that renders good (small) software accuracy for variables of interest needs to possess good precision for key variables, enough redundancy to detect biases so that undetected biases would have small magnitudes (this property is directly related to gross errors delectability), and a smearing effect of undetected biases on estimators of key variables that is limited (this property is directly related to gross errors resilience). These needed network capabilities generally require a good level of redundancy (i.e., more sensors than the number of key variables) as can be seen in the obtained results. Reducing the accuracy threshold asks for more sensors and, in the extreme case, no feasible solution is found.

## 9. CONCLUSIONS

We investigated two methods to calculate software accuracy for the design of linear and nonlinear sensor networks: a Monte Carlo simulation (stochastic accuracy) and an approximate method to calculate stochastic accuracy to design sensor networks using accuracy constraints. Methods to calculate accuracy require relatively short computational time that enables solving a general problem of accuracy targeted sensor network design in an acceptable time. Only in the most challenging problems, the large scale nonlinear problems like the TE process, do the methods show limitation in terms of computational efficiency: they are able to locate the optimal or near optimal solution within an acceptable time but require a long time to prove global optimality of the obtained solution.

## ■ AUTHOR INFORMATION

### Corresponding Author

*E-mail: bagajewicz@ou.edu.

## ■ NOMENCLATURE

$\hat{a}_i$ = accuracy of stream $i$
$p$ = confidence level
$Z_{\text{crit}}^{(p)}$ = maximum power measurement statistics for confidence level $p$
$\delta_i$ = maximum induced bias in a stream $i$
$\delta_{\text{crit},i}^{(p)}$ = gross error in stream $i$ at confidence level $p$
$\delta_{\max,\, s}$ = gross error in stream $s$
$\hat{\sigma}_i$ = precision of the estimator for stream $i$

## ■ REFERENCES

(1) Bagajewicz, M. On the Definition of Software Accuracy in Redundant Measurement Systems. *AiChE J.* **2005**, *51* (4), 1201–1206.

(2) Bagajewicz, M.; Nguyen, D. Stochastic-Based Accuracy of Data Reconciliation Estimators for Linear Systems. *Comput. Chem. Eng.* **2008**, *32* (6), 1257–1269.

(3) Bagajewicz, M. *Design and Upgrade of Process Plant Instrumentation*; Technomic Publishers: Lancaster, PA, 2000.

(4) Vaclavek, V.; Loucka, M. Selection of Measurements Necessary to Achieve Multicomponent Mass Balances in Chemical Plant. *Chem. Eng. Sci.* **1976**, *31*, 1199.

(5) Kretsovalis, A.; Mah, R. S. H. Effect of Redundancy on Estimation Accuracy in Process Data Reconciliation. *Chem. Eng. Sci.* **1987**, *42*, 2115.

(6) Madron, F.; Veverka, V. Optimal Selection of Measuring Points in Complex Plants by Linear Models. *AIChE J.* **1993**, *38* (2), 227–236.

(7) Bagajewicz, M. Design and Retrofit of Sensors Networks in Process Plants. *AIChE J.* **1997**, *43* (9), 2300–2306.

(8) Bagajewicz, M.; Cabrera, E. A New MILP Formulation for Instrumentation Network Design and Upgrade. *AIChE J.* **2001**, *48* (10), 2271–2282.

(9) Bagajewicz, M.; Sánchez, M. Duality of Sensor Network Design Models for Parameter Estimation. *AIChE J.* **1999**, *45a* (3), 661–664.

(10) Bagajewicz, M.; Sánchez, M. Design and Upgrade of Nonredundant and Redundant Linear Sensor Networks. *AIChE J.* **1999**, *45b* (9), 1927–1939.

(11) Bagajewicz, M.; Sanchez, M. Reallocation and Upgrade of Instrumentation in Process Plants. *Comput. Chem. Eng.* **2000**, *24a*, 1945–1959.

(12) Bagajewicz, M.; Sánchez, M. Cost-Optimal Design of Reliable Sensor Networks. *Comput. Chem. Eng.* **2000**, *23b*, 1757–176211/12, pp.

(13) Chmielewski, D.; Palmer, T.; Manousiouthakis, V. On the Theory of Optimal Sensor Placement. *AIChE J.* **2002**, *48* (5), 1001–1012.

(14) Gala, M.; Bagajewicz, M. Efficient Procedure for the Design and Upgrade of Sensor Networks Using Cutsets and Rigorous Decomposition. *Ind. Eng. Chem. Res.* **2006**, *45a* (21), 6687–6697.

(15) Gala, M.; Bagajewicz, M. Rigorous Methodology for the Design and Upgrade of Sensor Networks Using Cutsets. *Ind. Eng. Chem. Res.* **2006**, *45b* (21), 6679–6686.

(16) Nguyen, D. Q.; Bagajewicz, M. Design of Nonlinear Sensor Networks for Process Plants. *Ind. Eng. Chem. Res.* **2008**, *47* (15), 5529–5542.

(17) Sen, S.; Narasimhan, S.; Deb, K. Sensor Network Design of Linear Processes Using Genetic Algorithms. *Comput. Chem. Eng.* **1998**, *22*, 385–390.

(18) Carnero, M.; Hernandez, J.; Sanchez, M.; Bandoni, A. An Evolutionary Approach for the Design of Nonredundant Sensor Networks. *Ind. Eng. Chem. Res.* **2001**, *40*, 5578–5584.

(19) Carnero, M.; Hernandez, J.; Sanchez, M.; Bandoni, A. On the Solution of the Instrumentation Selection Problem. *Ind. Eng. Chem. Res.* **2005**, *44*, 358–367.

(20) Heyen, G. Dumont, M. Kalitventzeff, B. Computer-Aided Design of Redundant Sensor Networks. In *Proceeding of 12th European Symposium on Computer-aided Process Engineering*; Grievink, J., van Schijndel, J., Eds.; Elsevier Science: Amsterdam, 2002; pp 685—690.

(21) Ali, Y.; Narasimhan, S. Sensor Network Design for Maximizing Reliability of Linear Processes. *AIChE J.* **1993**, *39* (5), 820–828.

(22) Nguyen Thanh, D. Q.; Siemanond, K.; Bagajewicz, M. On the determination of downside financial loss of instrumentation networks in the presence of gross errors. *AIChE J.* **2006**, *52* (11), 3825–3841.

(23) Raghuraj, R.; Bhushan, M.; Rengaswamy, R. Locating Sensors in Complex Chemical Plants Based on Fault Diagnostic Observability Criteria. *AIChE J.* **1999**, *45* (2), 310–322.

(24) Bhushan, M.; Rengaswamy, R. Design of Sensor Network Based on the Signed Directed Graph of the Process for Efficient Fault Diagnosis. *Ind. Eng. Chem. Res.* **2000**, *39*, 999–1019.

(25) Bhushan, M.; Rengaswamy, R. Comprehensive Design of Sensor Networks for Chemical Plants Based on Various Diagnosability and Reliabilty Criteria. I. Framework. *Ind. Eng. Chem. Res.* **2002**, *41a*, 1826–1839.

(26) Bhushan, M.; Rengaswamy, R. Comprehensive Design of Sensor Networks for Chemical Plants Based on Various Diagnosability and Reliabilty Criteria. II. Applications. *Ind. Eng. Chem. Res.* **2002**, *41b*, 1840–1860.

(27) Bhushan, M.; Narasimhan, S.; Rengaswamy, R. Robust Sensor Network Design for Fault Diagnosis. *Comput. Chem. Eng.* **2008**, *32*, 1067–1084.

(28) Bagajewicz, M.; Fuxman, A.; Uribe, A. Instrumentation Network Design and Upgrade for Process Monitoring and Fault Detection. *AIChE J.* **2004**, *50* (8), 1870–1880.

(29) Bagajewicz, M.; Jiang, Q. Gross Error Modeling and Detection in Plant Linear Dynamic Reconciliation. *Comput. Chem. Eng.* **1998**, *22* (12), 1789–1810.

(30) Miller, R. W. *Flow Measurement Engineering Handbook*; McGraw-Hill: New York, 1996.

(31) Bagajewicz, M.; Nguyen, D.; Sugumar, S. K. New Method for Sensor Network Design. *Comput-Aided Chem. Eng.* **2008**, *25*, 429–434.

(32) Nguyen Thanh, D. Q.; Siemanond, K.; Bagajewicz, M. On the Determination of Downside Financial Loss of Instrumentation Networks in the Presence of Gross Errors. *AIChE J.* **2006**, *52* (11), 3825–3841.

(33) Madron, F.; Veverka, V. Optimal Selection of Measuring Points in Complex Plants by Linear Models. *AIChE J.* **1992**, *38* (2), 227.

(34) Ricker, N. L.; Lee, J. H. Nonlinear Modeling and State Estimation for the Tennessee Eastman Challenge Process. *Comput. Chem. Eng.* **1995**, *19* (9), 983–1005.