Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Process Control 15 (2005) 629–638
JOURNAL OF PROCESS CONTROL
www.elsevier.com/locate/jprocont

Instrumentation design based on optimal Kalman filtering

Estanislao Musulin [a], Chouaib Benqlilou [a], Miguel J. Bagajewicz [b], Luis Puigjaner [a,*]

[a] *Department of Chemical Engineering, Universitat Politécnica de Catalunya, Diagonal 647, 08028-Barcelona, Spain*
[b] *University of Oklahoma, 100 E. Boyd T-335, Norman, OK 73019, USA*

Received 21 December 2004; accepted 1 March 2005

Abstract

This paper presents a methodology for locating sensors in dynamic systems. It aims to maximize Kalman filtering performance by using accuracy as its main performance index. To accomplish this task, both the measurement noise and the observation matrices are manipulated. The method has been applied in two academic case studies and in the Tennessee Eastman Challenge Problem and has shown promising results.

Keywords: Sensor placement; Dynamic processes; Kalman filter

1. Introduction

Several techniques have been developed for designing and upgrading instrumentation for process plants in which monitoring tools are used that assume a steady-state [1–3]. Most of these, dating up to the year 2000, are reviewed in [4]. Extending these steady-state sensor placement procedures to handle dynamic processes relies mainly on correctly denning the threshold limit based on the performance of the sensor network (precision and/or accuracy) and on classifying variables of dynamic processes.

Background equations of different dynamic data reconciliation techniques can be used to calculate the sensor network precision without any knowledge of the actual measurements. The Kalman filter is not the only procedure that can estimate the measurement error covariance matrix of estimated process variables. Alternative dynamic data reconciliation (DDR) techniques [7,8] can also be applied. However, in some cases, the Kalman filter provides superior performance in terms of variance reduction and dynamic tracking, as discussed in [9]. Additionally, by providing the observability requirement, the Kalman filter is able to estimate all states using an incomplete and noisy measurement set.

In a previous work [18], a systematic search was used to obtain a maximum performance of the sensor network. Optimization was split into two individual problems: minimizing cost and maximizing performance. Therefore, by looking at the spectrum of solutions it was possible to decide the best trade-off between performance and cost. Moreover, the Pareto optimum space over the different objectives could also be determined. Muske and Georgakis [1] proposed such a Pareto optimal approach, in which the error of the variance-covariance matrix is used as a means to determine quality.

In this work, the state-space model identification is addressed from the perspective of sensor placement. An analysis of observability is performed and used in the formulation of sensor placement. The optimal design is obtained using a genetic algorithm. The methodology is finally applied to several scenarios and compared with other methods.

* Corresponding author. Tel.: +34 93 401 66 78; fax: +34 93 401 09 79.
E-mail address: luis.puigjaner@upc.es (L. Puigjaner).

0959-1524/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2005.03.002

Nomenclature

$\mathbf{A}_k$	state transition matrix
$\mathbf{B}_k$	control gain matrix
$C^{\max}$	margin cost
c_i	capital cost of sensor type i
$d^j_{p,1}$	Kalman filter performance indicator of j averaging $[\mathbf{P}_{k/k}]_j$ over n
$d^j_{p,2}$	Kalman filter performance indicator of j using asymptotic $[\mathbf{P}_{k/k}]_j$ value
$\mathbf{d}_{p,b}$	"best" sensor network performance
$\mathbf{d}_{p,c}$	current sensor network performance
$d_{p,u}$	upper bound on accuracy design specification
$\mathbf{H}_k$	observation matrix
G	number of generation used to evaluate early GA stop
L_{ind}	individual length
N_{G}	total number of generations
N_{ind}	number of individuals in each generation
P_{c}	crossover probability
P_{m}	mutation probability
$\mathbf{P}_{k/k}$	error covariance matrix
$\mathbf{P}_0$	covariance matrix of primary variable estimation errors
$\mathbf{P}_k$	Kalman filter gain matrix
$\mathbf{Q}$	covariance model error matrix
$\mathbf{R}$	covariance measurement error matrix
$\mathbf{S}$	set of process variables of interest
$\mathbf{u}_k$	vector of control input variable at sampling time k
$\mathbf{x}_k$	vector of state variable at sampling time k
$\mathbf{x}^*_k$	state variable equal to $[\mathbf{u}_k, \mathbf{x}_k]$
$\mathbf{y}_k$	vector of measurements variables at sampling time k
$\mathbf{v}_k$	vector of measurement noise
$\mathbf{w}_k$	vector of process model noise

Operators

cov(·)	covariance operator
E[·]	expectation operator
obsv(·)	function calculating the observability matrix
rank(·)	function calculating the rank of a matrix

Greek symbols

σ^2_i	variance of sensor type i
Φ	fitness function
ψ	observability requirement function
$\mu_{\min}$	minimum fitness increment

Subscripts

i	sensor type
I	number of sensor types
j	process variable
J	number of variables that can be measured
k	time instant
n	time-horizon

2. Proposed methodology

2.1. Kalman filtering algorithm

The Kalman filter [10,11] is a recursive technique for estimating state variables and their associated error variances. The algorithm uses the following discrete state-space dynamic model and measurement model:

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{1}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

where k represents a sample time, $\mathbf{x}_k$ is the n_x dimensional vector of state variables, $\mathbf{u}_k$ is the n_u dimensional vector of manipulated input variables and $\mathbf{y}_k$ is the n_y dimensional vector of measured variables. The state transition matrix $\mathbf{A}_k$, the control gain matrix $\mathbf{B}_k$ and the observation matrix $\mathbf{H}_k$ are matrices of an appropriate dimension and if their coefficients are time-independent the subscript k can be dropped.

The Kalman filter assumes errors in the process model and in the measured data. The process noise $\mathbf{w}_k$ represents errors in the state transition model. This noise is assumed to be white, with zero mean and a variance $\mathbf{Q}$. $\mathbf{v}_k$ represents a measurement noise with a variance $\mathbf{R}$.

Using process measurements, the error covariance matrix $\mathbf{P}_{k/k-1}$ associated with the estimated state vector $\hat{\mathbf{x}}_{k/k}$ is updated as follows:

$$\mathbf{P}_{k/k} = \mathbf{P}_{k/k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k/k-1} \tag{3}$$

where $\mathbf{K}_k$ is the Kalman filter gain given by

$$\mathbf{K}_k = \mathbf{P}_{k/k-1} - \mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_{k/k-1}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1} \tag{4}$$

To implement the algorithm, the following initial conditions should be estimated:

$$\begin{aligned} &E\langle\mathbf{x}_0\rangle = \hat{\mathbf{x}}_0 \\ &E\langle[\mathbf{x}_0 - \bar{\mathbf{x}}_0][\mathbf{x}_0 - \bar{\mathbf{x}}_0]^{\mathrm{T}}\rangle = \mathbf{P}_0 \end{aligned} \tag{5}$$

2.2. Measurement of instrument performance

2.2.1. Process variable estimation performance measurement

The computation of the error covariance matrix $\mathbf{P}_{k/k}$ is independent of the process variable measurements, as is apparent from Eq. (3). Indeed, the parameters re-

quired for computing $\mathbf{P}_{k/k}$ are $\mathbf{R}$ and $\mathbf{Q}$, and the initial value of $\mathbf{P}_{k/k}$ (say $\mathbf{P}_0$), and these variables can be evaluated without any knowledge of the process variable values. The measurement error covariance matrix $\mathbf{R}$ is given by the intrinsic quality of the measuring devices; thus, the elements of $\mathbf{R}$ can be chosen as the design parameters for sensor placement. Assuming that measurement errors are independent (null covariance), the matrix $\mathbf{R}$ has a diagonal form. Determining the process noise covariance $\mathbf{Q}$ is generally a more complex process due to difficulties entailed in directly observing the process noise. Usually, the diagonal elements of $\mathbf{Q}$ are assumed to be positive and fixed and the off-diagonal elements are set to zero [11].

Therefore, $\mathbf{P}_{k/k}$ is selected as a basis for assessing the accuracy of the Kalman filter estimation. If the measurement errors $\mathbf{R}$ are independent of time and normally distributed, the Kalman filter is an optimal unbiased estimator (of all possible unbiased estimators it is the one with the smallest variance). However, if the measurement errors are not normally distributed the Kalman filter cannot be used because it is biased [10].

Since $\mathbf{P}_{k/k}$ is not constant over the time-horizon $k = 0,\ldots,n$, the Kalman filter performance of process variable j can be calculated by averaging $[\mathbf{P}_{k/k}]_j$ the entire time-horizon as presented in Eq. (6):

$$d^j_{p,1} = \frac{1}{n}\left(\sum_{k=0}^{n}[\mathbf{P}_{k/k}]_j\right) \tag{6}$$

Usually (see Section 2.2.2) the Kalman filter converges; the gain $\mathbf{K}_k$ and $\mathbf{P}_{k/k}$ reach a constant value in a few iterations for any initial conditions $\mathbf{P}_0$ and $\hat{\mathbf{x}}_0$. Therefore, the asymptotic value of $\mathbf{P}_{k/k}$ can also be used as a performance measure as shown:

$$d^j_{p,2} = \lim_{k\to n}([\mathbf{P}_{k/k}]_j) \tag{7}$$

In fact, when the Kalman filter is applied to a system that is continuous and dynamic, Eq. (6) is preferred, whereas when conditions reflect short-lived batch systems Eq. (7) is more appropriate.

2.2.2. Kalman filter initialization and convergence considerations

In the evaluation of the KF performance at any time k, an important issue to be considered is the fact that in practical applications the initial KF conditions ($\mathbf{x}_0$ and $\mathbf{P}_0$) are guessed, and the KF may not converge for any initial conditions.

The discrete time Riccati equation provides a useful means to evaluate the asymptotic behavior of $\mathbf{P}_{k/k}$. It can predict two types of theoretical divergencies due to

- the *natural behavior* of the process dynamics,
- process non-observability with the given measurements.

The Riccati equation can be used to demonstrate that "theoretical divergence" does not occur in controllable and observable systems, independently from the initial conditions.

However, sometimes the algorithm does not converge under "theoretical convergence" conditions due to

- bad data,
- numerical problems,
- *Mismodeling*: unmodeled state variables, unmodeled process noise, errors in the transition matrix ($\mathbf{A}$), overlooked non-linearities.

Divergence control methods are widely treated in the literature [12–14], being many of them alternate forms or extensions of the Kalman filter algorithm. In the next sections convergence conditions are assumed. Specifically, in the presented case studies the value of $n = 15$ has been sufficient to reach convergence.

2.2.3. System performance

Different sensor networks lead to different values of d^j_p. The performance measure presented in Eq. (7) corresponds to the estimation of a particular process variable. Although the objective is to maximize the performance of the filter for all variables, maximizing the performance for one particular variable may not be compatible with maximizing it for others, which can cause conflict.

In this work, we propose evaluating the performance of the whole system by selecting a function that relates the individual variable performance measurements given by Eq. (7). One simple alternative is to select the variable with the lowest performance value. A more elaborate option is based on evaluating the distance of sensor network ($d_{p,c}$) from the "best" sensor network. Thus, the "best" sensor network ($d_{p,b}$) is defined as the one in which all the process variables are completely measured with the most accurate devices available. When only a few process variables are of interest, only those ones are considered.

Let $\mathbf{S}$ be the set of variables of interest. Then, the system performance can be constructed by comparing the current system performance ($d^j_{p,c}$) and the "best" system performance ($d^j_{p,b}$) for all the process variables j, as follows:

$$d^{\mathbf{S}}_p = \left(d_0 + \sum_{j\in\mathbf{S}}|d^j_{p,c} - d^j_{p,b}|\right)^{-1} \tag{8}$$

where d_0 is introduced to avoid singularities: if the value of $d_0 = 1$, the maximum performance value is normalized and the best performance corresponds to $d^{\mathbf{S}}_p = 1$.

2.3. Sensor placement

Clearly, at a design phase, measurement errors ($\mathbf{v}_k$) are mainly due to measuring device errors. A possible connection between sensor placement and the Kalman filter can be performed via the measurement error variance matrix **R**, by assigning a large value to unmeasured variables in this matrix. The design procedure should choose the value of the diagonal of **R**. In classical control, manipulated variables **u** are assumed to be known. Thus, the elements of **P** are the variance of state variables **x**. However, from a monitoring perspective, the estimated value and the variance of variables are considered. In this case, the distinction between state and manipulated variables is avoided. Therefore, to include variable set **u** in the Kalman filter algorithm, these variables have to be considered as state variables, which leads to a new n_z dimensional state variables, $\mathbf{x}_k^* = [\mathbf{u}_k, \mathbf{x}_k]$. This change involves an update of the model identification (**A**, **B** and **H**). Indeed, since the control input variables are considered to be state variables, matrix **B** is dropped. Next, if we assume that all process variables are potentially measured, matrix **H** is the identity matrix of size n_z. This assumption is very important in the placement algorithm for analytical calculation of Kalman filter gain, as shown in Eq. (4). To identify the state transition matrix it is assumed that input variables during a given time period are correlated with the input during the previous time period, as follows:

$$\mathbf{u}_k \approx \mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{9}$$

Using the state variable $\mathbf{x}_k^*$ and Eqs. (1) and (9) provide the following transition state matrix $\mathbf{A}^*$:

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{I}^{n_u \times n_u} & \mathbf{0}^{n_u \times n_x} \\ \mathbf{B}^{n_x \times n_u} & \mathbf{A}^{n_x \times n_x} \end{bmatrix} \tag{10}$$

giving

$$\mathbf{x}_k^* = \mathbf{A}^* \mathbf{x}_{k-1}^* + \mathbf{w}_{k-1}^* \tag{11}$$

$$\mathbf{y}_k^* = \mathbf{H}^* \mathbf{x}_k^* + \mathbf{v}_{k-1}^* \tag{12}$$

2.4. Observability analysis

It is clear that the system performance $d_p^{\mathbf{S}}$ can lead to acceptable values for any set of sensors if the process variables belonging to **S** are all strictly observable either independently or through the model equations. Therefore, the values of $d_p^{\mathbf{S}}$ depend not only on the dynamic data reconciliation technique (in this case the Kalman filter) but also on the sensor network selected.

From **S**, those variables that are redundant are adjusted and their variances are fed back by the Kalman filter. The estimation of unmeasured but observable variables and their variances is obtained by means of coaptation techniques (the Kalman filter can directly solve the incomplete and noisy state observation). In this way it is possible to find the variances of all the process variables in **S** in order to calculate the value of $d_p^{\mathbf{S}}$.

Therefore, observability analysis is an important aid in the design of instrumentation schemes. An unmeasured variable is defined as unobservable if it cannot be uniquely determined through the measured variables. A variable is defined as non-redundant if the deletion of its measurements will make it unobservable. Thus, both definitions are based on the uniqueness of determining a variable value. These fundamental properties will serve for steady-state as well as for dynamic processes.

Albuquerque and Biegler [6] developed an efficient method for classifying the variables of a dynamic process. After discretizing the differential equations using implicit Runge–Kutta, their method calls for linearizing the process model. Finally, they apply the properties of observability and redundancy to derive the tools necessary for such a classification using sparse LU decomposition. Their approach to observability analysis is similar to the one developed by [15], except for that they use it on dynamic systems. An important result presented in their work is that redundancy analysis in steady-state differs from that used in dynamic systems.

It is important to mention that the procedure explained above *variable classification* yields the same results as those presented by the Kalman filter when using the observability matrix. To ensure that the Kalman filter converges to an acceptable and unique value the observability constraints must be satisfied. Therefore, variable classification has a direct impact on the Kalman filter performance. The error variance of the unobservable variable j will tend towards infinity and thus the Kalman filter gain becomes equal to zero. This means that the second term on the right hand side of Eq. (3) is null while the first term is continuously increasing at each iteration of the Kalman filter. Moreover, if the pair $(\mathbf{A}^*, \mathbf{H}^*)$ is completely observable, the unmeasured but observable variables are given a unique error covariance matrix.

As was explained, unmeasured variables are made evident in the sensor placement model formulation by assigning infinite variance to the corresponding positions of **R**. It is additionally necessary to explicitly handle the unmeasured variables by modifying $\mathbf{H}^*$. The columns of $\mathbf{H}^*$ corresponding to unmeasured variables are set to zero-columns so that the Kalman filter results can be validated by requesting that the rank of the observability matrix be made equal to the number of state variables:

$$\operatorname{rank}(\operatorname{obsv}(\mathbf{A}^*, \mathbf{H}^*)) = n_z \tag{13}$$

If the hypothesis of observability is not fulfilled, that is, the observability matrix of the pair $(\mathbf{A}^*, \mathbf{H}^*)$ is not full row rank, the state covariance error of the Kalman filter will not converge to a unique value from different initial conditions [11].

2.5. Sensor placement procedure

The objective is to maximize the system performance given in Eq. (8) as much as possible by varying the diagonal elements of matrix **R** subject to a cost bound $C^{\max}$ and a set of additional constraints $\psi(\mathbf{S})$ which are related to observability. From the diagonal elements of **R**, the placement and type (e.g., level-meter, flowmeter, etc.) of sensors are directly obtained.

Assume s_{ij} is an integer variable corresponding to the placement of sensor type i, at network location j. Additionally, the sensor type is given by its variance error (σ_i^2). When a variable is not measured, a "dummy" sensor with ($\sigma_i^2 \to \infty$) is selected, which has a null cost. This directly affects **R** and $\mathbf{H}^*$. Therefore, the optimal sensor network problem is formulated as follows:

$$\begin{aligned} &\max_{s_{ij}} \ (d_p^{\mathbf{S}}) \\ &\text{s.t.} \quad \sum_i (c_i \cdot s_{ij}) \leqslant C^{\max} \\ &\qquad \psi(\mathbf{S}) = 1 \end{aligned} \tag{14}$$

where $\psi(\mathbf{S})$ is an algorithmic constraint. Its value is equal to one when the set of feasible sensor networks allows one to observe all the process variables belonging to the set **S**, and zero otherwise. In practice, this constraint should ensure that the variable value belonging to **S** should be able to be inferred, although some redundancy should also be allowed to validate those inferences. In other words, the function $\psi(\mathbf{S})$ should ensure a minimum redundancy in order to provide an acceptable precision for the estimation of variables.

In addition, a threshold value of cost $C^{\max}$ is used to build a Pareto curve. Since the relationship between investment and performance is generally non-linear, increasing $C^{\max}$ by a small amount might lead to a considerable performance improvement and vice versa.

It is important to note that the value of **R** is directly related to the Kalman filter convergence. A reduced measurement error noise (i.e., low values of **R**) leads to quicker convergence of the algorithm. Therefore, an accurate sensor network will reduce the measurement error and consequently the convergence time, which allows for a better tracking of the process behavior. Thus, a lower sensor network cost can provide accurate solutions for slow control systems, whereas a fast changing control system will require a more expensive network in order to obtain an optimal solution and to ensure a good tracking of the process variables.

2.6. Genetic algorithm approach

The different ways in which a catalogue of I sensors can be arranged in J location positions is $(I+1)^J$. Additionally, in most real life problems, near optimal solutions that can be generated quickly are more desirable than optimal solutions which require a huge amount of time. Therefore, a Genetic Algorithm (GA) [19–22] technique has been chosen to solve this highly combinatorial optimization problem.

In classical GA formalism a set of N_{ind} candidate solutions (population) are generated randomly. The potential solution in the multidimensional search space (individual) is coded as a vector, called a chromosome. The goodness of each individual in the population is evaluated by using a pre-specified fitness criterion. Upon assessing the fitness of all the chromosomes in the population, a new generation of individuals is created from the current population by using crossover and mutation operators.

In the proposed approach, each gene in the chromosome corresponds to a variable that can be measured once, using a single sensor selected from a set with different cost and accuracy.

Therefore, the length of the chromosome is equal to the number of variables that can be measured J. A dummy sensor with null cost and a very low accuracy is included in the sensor catalogue in order to take into account the location of no sensors. Then, the value of each gene can vary from 1 to $I+1$, where I is the number of sensors in the catalogue.

The assessment of the fitness is made in two stages: Firstly, the sensor network feasibility (i.e., the restriction of Eq. (14)) is evaluated. If the individual involves any unfeasibility the objective function value is considered infinite and the sensor network fitness Φ is set to zero. If the individual is feasible, the fitness is evaluated using the performance criterion presented in Eq. (14) (i.e., $\max_{s_{ij}}(d_p^{\mathbf{S}})$). During the feasibility and fitness evaluation, the sensor type is associated with its characteristics (i.e., cost and accuracy).

To generate new populations, individuals are selected from the initial population and crossed by means of roulette wheel selection, two point crossover and mutation operators. The algorithm is terminated using one of two stopping criteria: either the number of generations reaches a predefined maximum value N_{G} or the current population does not yield sufficient improvement $\mu_{\min}$ compared with the performance reached G generations before.

The Matlab genetic algorithm Toolbox developed by the University of Sheffield [17] was used in the following case studies. All the examples were solved using an AMD XP2500 processor with 512MB RAM running on Microsoft Windows XP.

3. Results and discussion

In the following case studies, the number of possible measurements considered and the number of states is the same, therefore following a practical initialization of the

Kalman filter algorithm, the value of $\mathbf{P}_0$ is selected such that it equals $\mathbf{R}$.

3.1. Case study I

Fig. 1 shows a process network used as a motivating example in order to evaluate the proposed sensor placement methodology. It is taken from [16] (eight streams and four storage tanks are considered). The aim of this example is to monitor the flowrate and mass hold-ups. It is possible to place flowrate sensors in all streams and a level sensor in all nodes (i.e., $J = 12$). The dynamic mass balance is adopted for supporting the instrumentation design schemes. Table 1 gives the catalogue of sensors and their characteristics, including a dummy sensor that represents the location of no sensors.

The matrices $\mathbf{A}^*$, $\mathbf{B}$, $\mathbf{H}^*$ for this case study are as follows:

$$\mathbf{A}^* = \begin{pmatrix} \mathbf{I}^{8\times 8} & \mathbf{0}^{8\times 4} \\ \mathbf{B}^{4\times 8} & \mathbf{I}^{4\times 4} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \end{pmatrix}$$

$$\mathbf{H}^* = \mathbf{I}^{12\times 12} \tag{15}$$

and the state vector $\mathbf{x}^*$ results,

$$\mathbf{x}^* = [F1, F2, F3, F4, F5, F6, F7, F8, M1, M2, M3, M4] \tag{16}$$

The GA has been applied with $N_{\text{ind}} = 30$, $N_{\text{G}} = 50$, $G = 15$, $\mu_{\min} = 1^{-10}$, $P_{\text{c}} = 0.7$ and $P_{\text{m}} = P_{\text{c}}/L_{\text{ind}} = 0.0538$. With these parameters, the algorithm converges in 1 m and 40 s.

The low computational time required by the algorithm to reach the solution allows it to scan the entire restriction space, from $C^{\max} = \$1000$ to \$30,000, in steps of \$1000. The scan was executed 10 times taking a total time of 8 h and 37 min.

To improve the computation performance, the search is biased by including the best sensor network obtained for a given $C^{\max}$ in the initial population of the computation with a $C^{\max}$ immediately superior. The resulting profiles for the system performance (defined as in Eq. (8)) are presented in Fig. 2, including the upper and lower bounds obtained during the 10 scans and the mean values. Note that $C^{\max} = \$30{,}000$ allows the location of the more expensive sensor to measure each variable.

It has been found that the variance in the solutions obtained in the different scanning experiments can be reduced by changing the algorithm parameters (increasing N_{ind} and N_{G}) but at the cost of an increase in the calculation time.

It is clear that higher investment leads to a better sensor network. However, in the curve it is difficult to find a point that presents a clear best solution from the point of view of the cost-performance trade-off. The sensor network corresponding to the break in the curve at a margin cost of \$20,000 with a performance of 42.74% was selected since it gives an acceptable performance at a relatively low cost (see Table 2).

To check the results given by the algorithm, an exhaustive search was performed for $C_{\max} = \$16{,}000$, the computation time being 51 h and 21 min. From the $3^{13} = 531{,}441$ possible solutions only 36% are feasible. They are presented in Fig. 3 along with the best and worst solutions reached by the GA. The optimum corresponds to only one sensor network and there are 44 solutions better than the one found by the GA. Table 3 presents the characteristics of the optimum sensor network for a margin cost of \$16,000 and the sensor networks obtained with GA for margin costs of \$16,000

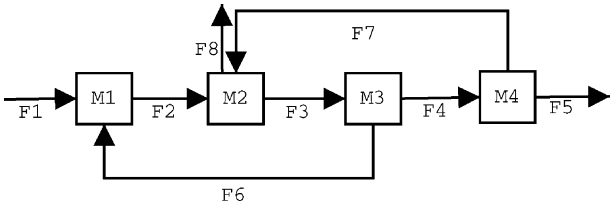

Fig. 1. Case study I: Scheme of the process network.

Table 1
Case study I: Available sensors in the catalogue

Sensor type	Cost	Accuracy (%)
1	2500	1
2	1800	5
3	0	1000

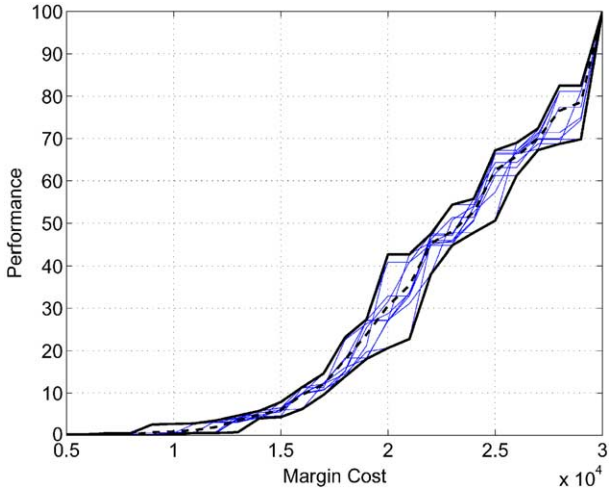

Fig. 2. Case study I: Performance obtained in several restriction space scanning.

Table 2
Case study I: Sensor network characteristics for the best cost-accuracy trade-off

$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$M1$	$M2$	$M3$	$M4$	MC	RC	d_p^S
–	1%	–	1%	1%	1%	–	–	1%	1%	1%	1%	$20,000	$20,000	42.74

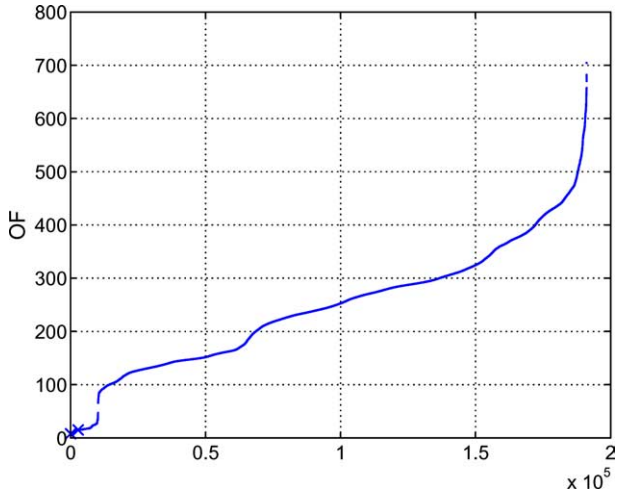

Fig. 3. Case study I: Sorted OF values obtained through an exhaustive search. The crosses represent the best and worst solutions obtained by the GA.

and $20,000. Note that the solutions obtained by the GA are near the optimum, and were obtained very quickly.

3.2. Case study II

Let us consider the sensor network design of the linear dynamic process taken from [5], and reproduced in Fig. 4. It consists of one tank (Node 2) linked by four flow streams. Node 1 is a splitter that was considered as a storage tank, but without any accumulation. This hypothesis is equivalent to having a measuring device in Node 1 of a high precision and at zero cost. The sensor options for each stream are assumed to be [1%, 2%, 3%] of relative error with costs of $2500, $1500 and $800, respectively, while the options for hold-ups are assumed to be [16 kg, 64 kg] of relative error with costs of $2600 and $500, respectively.

The sensor placement optimization problem presented above was inverted to minimize the capital cost subject to the requirement on certain performance limit

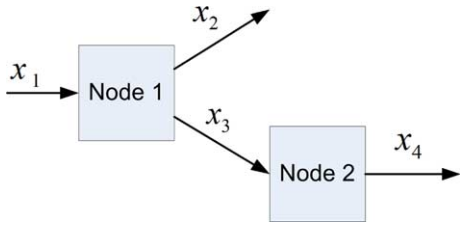

Fig. 4. Case study II: Process flow diagram.

($d_{p,u}^S$) and system observability. The resulting problem can be stated as follows:

$$\begin{aligned} &\min_{s_{ij}} \quad \left(\sum_i c_i \cdot s_{ij}\right) \\ &\text{s.t.} \quad d_{p,c}^j \leqslant d_{p,u}^j \quad \forall j \in \mathbf{S} \\ &\qquad \psi(\mathbf{S}) = 1 \end{aligned} \tag{17}$$

Therefore, the incidence matrix **B** for this process is given by Eq. (18), where the columns represent variables x_1, x_2, x_3, x_4 and x_5, respectively and the rows represent the dynamic mass balance around Nodes 1 and 2, respectively. For both stream and hold-up measuring device options a "dummy" sensor is added to emulate unmeasured variables.

$$\mathbf{B} = \begin{pmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \end{pmatrix} \tag{18}$$

The sensor placement problem is solved using the GA presented in Section 2.6. In fact, this is a small problem ($4^3 \times 3^2 = 574$ possible sensor networks) that would not require the use of GA, but it has been included here for purposes of comparison with the results obtained in other studies.

The individuals are coded using base four for the flow-meters and base three for the level-meters. The fitness function is the opposite of the objective function in Eq. (17) and the feasibility is checked by means of the constraints in Eq. (17). The GA configuration parameters are the following: $N_{\text{ind}} = 20$, $N_G = 10$, $G = 10$, $P_c = 0.7$, $P_m = 0.7/5$ and $\mu_{\min} = 10^{-10}$.

Table 3
Case study I: Comparison between the sensor network characteristics of the solutions given by the GA and the optimum found through an exhaustive search for MC = $16,000

Algorithm	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$M1$	$M2$	$M3$	$M4$	RC	d_p^S	CPU time
GA (best solution)	–	1%	–	1%	1%	1%	–	–	1%	1%	1%	1%	$158,007	11.39	1.66 min
GA (worst solution)	1%	5%	–	–	–	–	–	1%	5%	1%	1%	5%	$15,400	6.24	1.66 min
Optimum solution	5%	–	–	5%	5%	5%	–	–	5%	1%	5%	1%	$15,800	11.66	3080 min

Table 4
Case study II: Comparing the proposed approach with the findings of Chmielewski et al. [5]

	s_1	s_2	s_3	s_4	s_5	Cost
[5]	–	3%	1%	2%	64	$5300
GA	3%	–	–	3%	64	$2100

In order to test the algorithm, the optimization problem was solved 10 times. The costs obtained in each run were as follows:

[2800, 2100, 2100, 2800, 2900, 2900, 2900, 2900, 2800, 2100]

The sensor network corresponding to the solution, given a cost of $2100, is presented in Table 4. The mean time for each optimization was 27.7 s. To check the results, an exhaustive search was executed, which showed

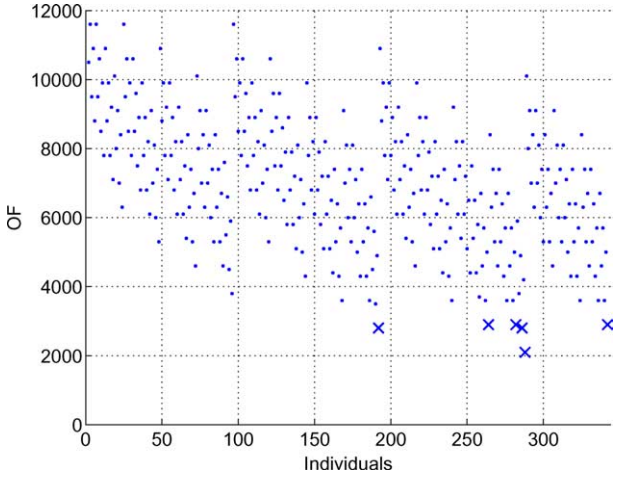

Fig. 5. Case study II: OF values of the feasible sensor networks. The crosses represent the solutions obtained by the GA.

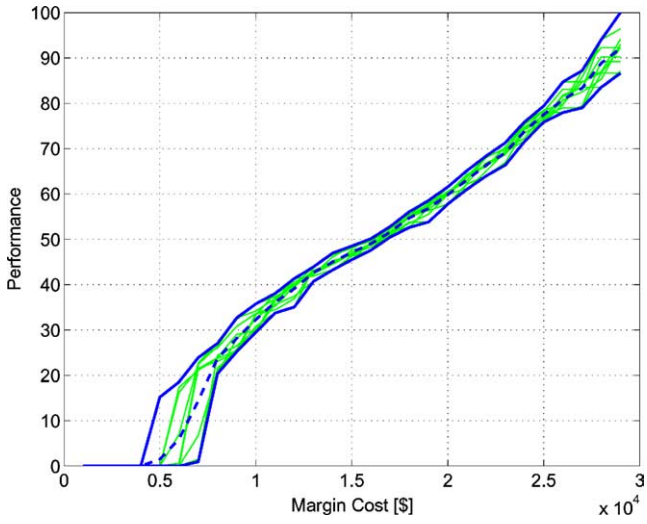

Fig. 7. Case study III: Performance vs. margin cost. The dashed line represents the mean. The bold lines show the upper and lower bounds.

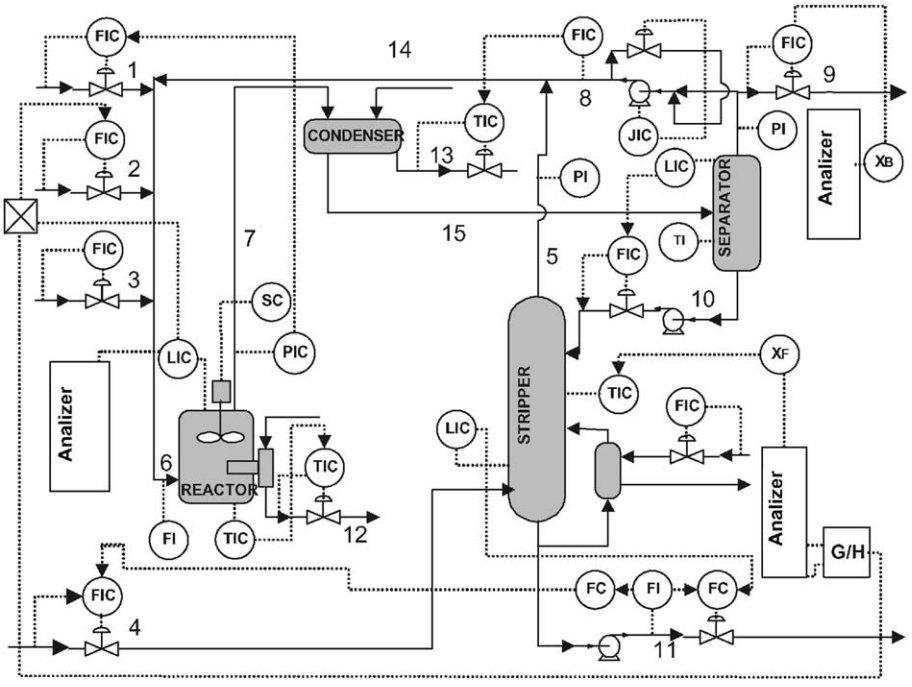

Fig. 6. Case study III: Tennessee Eastman process flowsheet.

Table 5
Case study III: Sensor network accuracy (d_p^S) and cost (RC) for the best (B) and worst (W) solutions given by the GA under different restrictions (MC)

	1	2	3	4	5	6	7	8	9	10	11	14	15	$L1$	$L2$	$L3$	MC ($)	RC ($)	d_p^S (%)
B	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1	29,000	29,000	100
W	.25	.1	.1	.25	.1	.1	.5	.1	.1	.1	.1	.1	.1	.1	.1	.1	29,000	27,200	86.60
B	.5	.1	.5	.5	.1	.1	–	.1	.1	.5	.1	.5	.5	.1	.1	.1	20,000	19,800	61.55
W	.5	.1	.1	.25	2	1	.25	.1	.1	.25	.5	.5	1	.1	.1	.1	20,000	19,750	57.79
B	1	2	1	1	.5	–	.5	.5	1	.5	.5	1	2	.1	.1	.1	10,000	10,000	35.79
W	1	.5	.25	2	.5	.5	–	–	.5	1	1	1	2	.1	.5	.5	10,000	10,000	29.56

that out of the 574 possible sensor networks a total of 343 turned out to be feasible. The computation time for evaluating all the feasible solutions was 1 min 16 s. The objective function (OF) values obtained for each individual are presented in Fig. 5. The solutions yielded for the GA in the different executions are marked with a cross. The exhaustive optimization shows that there exists one local optimum corresponding to the best sensor network that was found by the GA in three of the ten runs.

It is worthy of note that although the GA does not always yield the optimum, it yields a good solution in a short time. Obviously, the application of the presented GA is recommended for higher combinatorial problems.

3.3. Case study III

The Tennessee Eastman (TE) process [23] involves the production of two products, G arid H, from four reactants, A, C, D and E. There are 41 measured process variables and 12 manipulated variables. The process is depicted in Fig. 6. There are five main units: an exothermic 2-phase reactor, a product condenser, a flash separator, a reboiled stripper, and a recycle compressor.

The design of a sensor network with high accuracy and at a minimum cost has been performed. The application focuses in reconciling signals corresponding to material flows and units inventory. For this purpose, two sets of sensors, one containing flow-meters and other containing level-meters, were proposed. The flow-meter variances are [0.1%, 0.25%, 0.5%, 1%, 2%] and their corresponding costs are [$2000, $1700, $800, $500, $250]. In turn, the level-meter variances and costs are [0.1%, 0.5%, 1%, 2%] and [$1000, $800, $500, $300], respectively.

The potentially measured variables are 13 streams (1–11, 14 and 15) and the hold-ups of the reactor ($L1$), separator ($L2$) and stripper ($L3$). According to the proposed sensor placement problem statement, the number of sensor network alternatives is combinatorially high ($6^{13}\times5^3 = 16.32\text{E}+12$). In this case the application of an exhaustive search is unpractical (it would take years!!). The optimization problem was solved using the approach presented in Section 3.1. The GA parameters were set to $N_{\text{ind}} = 50$, $N_{\text{G}} = 200$, $G = 15$, $P_{\text{c}} = 0.7$, $P_{\text{m}} = 0.7/16$ and $\mu_{\min} = 10^{-10}$.

Since the GA converges quickly, it allows one to scan the restriction space, and in doing so to solve the problem for $C^{\max}$ from $1000 to $29,000 in steps of $1000. Such a scan was undertaken 10 times in order to check the feasibility of the algorithm. The trade-off between cost and performance is shown in Fig. 7 for all the scans. The time for all the scans was 33 h, 35 min and 7 s, which corresponds to a mean time of 6 min and 57 s for each solution to converge. Using a Pareto optimal analysis it is possible to detect the best performance according to the desired investment. The most suitable investment situation corresponds to an increase in performance at a low cost, which presents a break in the curve. In this case, an investment of $10,000 is sufficient to yield an acceptable performance of 35.79% and the corresponding sensor network is shown in Table 5.

It should be noted that the real cost (investment) obtained by adding the value of all sensors placed does not necessarily coincide with the margin cost $C^{\max}$, which is a restriction value. Additionally, the fact that the real costs of the sensor networks obtained by the GA appear to be near to the margin cost restriction lends additional support to the method proposed.

Finally, note that the only known optimum corresponds to an investment of $29,000 and the placement of sensors at all locations. This optimal solution was found by the GA, but not on all the runs. However, the worst solution found by the GA is very near the optimum, as can be seen in Table 5.

4. Conclusions

The greatest contribution of this work is the establishment of a methodology for optimally placing a minimum number of measuring devices so as to satisfy the sensor network accuracy performance of a dynamic system. The solution strategy has been implemented in academic and industrial case studies and has shown promising results. The profile of the relative increase in system performance along the sensor network and the associated investment margin gives the designer all alternatives. The proposed approach can be extended to take

on non-linear systems by using the extended Kalman filter.

Acknowledgments

The financial support received from the *Generalitat the Catalunya* through the FI program and the additional partial support received from the European Union (contracts No. G1RD-CT-2001-00466 and MRNT-CT-2004-512233) is graciously acknowledged. We would also like to acknowledge financial support for Dr. Bagajewicz's sabbatical stay at ETSEIB, provided by the Spanish Ministry of Education.

References

[1] K.R. Muske, C. Georgakis, Optimal measurement system design for chemical processes, AIChE J. 49 (6) (2003) 1488–1495.
[2] C. Benqlilou, M. Graells, E. Musulin, L. Puigjaner, Design and retrofit of reliable sensor networks, Ind. Eng. Chem. Res. 43 (2004) 8026–8036.
[3] C. Benqlilou, M. Graells, L. Puigjaner, Decision-making strategy and tool for sensor network design and retrofit, Ind. Eng. Chem. Res. 43 (2004) 1711–1722.
[4] M.J. Bagajewicz, Process Plant Instrumentation. Design and Upgrade, Technomic Publishing Company, 2000, ISBN 1-56676-998-1, Available from: <http://www.techpub.com>, Now CRC Press: <http://www.crcpress.com>.
[5] D.J. Chmielewski, T. Palmer, V. Manousiouthakis, On the theory of optimal sensor placement, AIChE J. 48 (5) (2002) 1001–1012.
[6] J.S. Albuquerque, L.T. Biegler, Data reconciliation and gross-error detection for dynamic systems, AIChE J 42 (10) (1996) 2841–2856.
[7] D.K. Rollins, D. Devanathan, Unbiased estimation in dynamic data reconciliation, AIChE J. 39 (8) (1993) 1330–1334.
[8] M. Bagajewicz, Q. Jiang, Integral approach to plant linear dynamic reconciliation, AIChE J. 43 (1997) 2546–2558.
[9] C. Benqlilou, M. Bagajewicz, A. Espuña, L.A. Puigjaner, Comparative study of linear dynamic data reconciliation techniques. In: 9th Mediterranean Congress of Chemical Engineering, 26–29 November 2002, pp. 8–31.
[10] S. Narasinham, C. Jordache, Data Reconciliation and Gross Error Detection, Gulf Publishing Company, 2000.
[11] P.S. Maybeck, Stochastic models, estimation, and control series, Mathematics in Science and Engineering, vol. 141, Academic Press, New York, 1979.
[12] G. Minkler, J. Minkler, Theory and Application of Kalman Filtering, Magellan Book Company, 1993.
[13] S.M. Grewal, A.P. Andrews, Kalman Filtering, Theory and Practice Using MATLAB, Wiley-Interscience Publication, 2001.
[14] A.J. Laub, Numerical linear algebra aspects of control design computations, IEEE Trans. Automat. Control. AC-30 (2) (1985) 97–108.
[15] C.M. Crowe, Observability and redundancy of process data for steady state reconciliation, Chem. Eng. Sci. 44 (1989) 2909–2917.
[16] M. Darouach, M. Zasadzinski, Data reconciliation in generalized linear dynamic systems, AIChE J. 37 (1991) 193–201.
[17] Genetic Algorithm Toolbox [on-line]. Consulted 1 August 2003. Available from: <http://www.shef.ac.uk/uni/projects/gaipp/ga-toolbox/>.
[18] C. Benqlilou, M. Bagajewicz, A. Espuña, L. Puigjaner, Sensor-placement for dynamic systems, in: J. Grievink, J. van Schijndel (Eds.), Proceeding of 13th European Symposium on Computer-aided, Process Engineering, Elsevier Science, Amsterdam, 2003, pp. 371–376.
[19] G. Heyen, M. Dumont, B. Kalitventzeff, Computer-aided design of redundant sensor networks, in: J. Grievink, J. van Schijndel (Eds.), Proceeding of 12th European Symposium on Computer-aided Process Engineering, Elsevier Science, Amsterdam, 2002, pp. 685–690.
[20] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[21] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[22] E. Musulin, M.J. Bagajewicz, J.M. Nougués, L. Puigjaner, Design and upgrade of sensor networks for principal components analysis monitoring, Ind. Eng. Chem. Res. 43 (2004) 2150–2159.
[23] J. Downs, E. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (1993) 245–255.