A New Cost-Optimal and Fault-Tolerant Instrumentation Sensor Network Design Methodology for Chemical Plants

V. Mohammadnia *, K. Salahshoor** 
Sh. Salehi***

* (Tel: 98-912-5839419; e-mail: v.mohammadnia@gmail.com).
** (e-mail: salahshoor@put.ac.ir)
*** (e-mail: shahnaz.salehi@yahoo.com)

Abstract: In this paper, an optimal redundant instrumentation sensor network design methodology is presented for complex chemical process plants using a combinatorial particle swarm optimization search (CPSO) engine. The approach, which is more flexible and general in comparison with previous works, aims to minimize cost as a main design factor, similar to the usual trend in the literature. Besides, it caters for fault-tolerance issue as a crucially important feature in the design procedure which has not been addressed sufficiently in the reported research works. For this purpose, weak redundancy degree (WRD) and sensor network reliability (R) are incorporated in the proposed design scheme as three evaluating measures. This enables the designer to maintain a desired fault-tolerant redundancy in the proposed sensor network to cope with a possible set of sensor failures. Thus, the developed CPSO engine searches in a diverse variety of sensor networks to adopt the most fitted one based on the imposed fault-tolerant design constraints. This facilitates the network realization of the fault-tolerance as the most attractive feature which is practically very demanding. Implementation of the proposed design methodology is illustrated in a simulated continuous stirred tank reactor (CSTR) as a benchmark process plant used in a large-scale design to show its effective capabilities.

1. INTRODUCTION

Measurements of all process variables are not practically cost-effective and yet operationally feasible in complex industrial plants. Accordingly, only a limited number of process variables are decided to be measured directly and hence reconciliation techniques could be beneficial for estimating the non-measured variables using the process model dynamics. Generally, a sensor network design methodology mainly deal with location or/and precision of sensors in large-scale plants so that some desired criteria viz.: observability (Vaclaveck and Loucka, 1976), precision (Muslin et al., 2005) and (Bagajewicz and Cabrera, 2002), reliability of estimation of variables (Ali and Narasimhan, 1993), (Ali and Narasimhan, 1995) and gross and error detectability (Bhushan and Rengaswamy, 2000a), (Raghuraj et al., 1999) are satisfied.

Bagajewicz (1997) used a tree type enumeration procedure to design a minimal cost network subject to constraints on precision, availability, resilience and error detectability. He proposed a design strategy that incorporates these criteria simultaneously for linear systems and suggested a MINLP to solve the problem. Further, Bagajewicz and Sanchez (1999) showed that problem of minimizing the variance subject to cost constraint can be converted to the problem of minimizing the cost subject to the variance constraints via determining measurement locations in linear networks. Bagajewicz and Cabrera (2002) presented a new MILP formulation, replacing the previous tree search solution procedures for minimizing cost subject to explicit constraints of precision, error detectability, resilience and availability. Although their method works well for small and medium problems, for large size problems the challenge exists. Sen et al. (1998) integrated graph theory and genetic algorithm concepts to develop a generalized sensor network design algorithm for non-redundant linear mass flow processes. In comparison with graph-theoretic algorithms (Ali and Narasimhan, 1993), GA-based method (Sen et al.,1998) provides more near optimal solutions (Sen et al.,1998). Bagajewicz et al. (2004) developed an instrumentation network design scheme that could reflect the potential benefit of adding sensors in networks and used value and cost concepts separately and applied them both in the integrated design, enabling to satisfy fault detection, material accounting and control criteria simultaneously. Kotecha et al. (2008) proposed a duality between the precision and reliability problems for non-redundant sensor network design in linear processes. This method enables one to convert any reliability design measure to precision framework and use explicit optimization algorithm, which was already developed for precise design (Bagajewicz and Cabrera, 2002), to design sensor network in the precision domain, satisfying reliability constraints specified in the design.

Only a few works have addressed the sensor network design by determining both type and location of sensor simultaneously; Muslin et al. (2005) discussed both location and type of sensors in precise linear sensor network designs. If type of sensors is not a consideration in design procedure,
number of possible networks that can be constructed via
given set of sensors decreases drastically. Observing this
point, i.e. neglecting the variety of sensors in the design
procedure can lead to a substantial saving of design time.
Subsequently, designer can take advantage of the saved time
to study on designs developed by enumeration methods.
Enumeration methods try to examine all possible candidates
based on a logical algorithm, and suggest the most optimal
solution whose optimality is guaranteed because of their
inherit analytical behavior. This may be the main reason
behind previous works that have ignored sensor variety in
their presented design.

Staroswieck et al (2004) addressed the problem of fault
tolerant estimation and the design of fault tolerant sensor
networks. They defined fault tolerance with respect to a
principle that a given functional of the system state should
remain observable when sensor failures occur. All sensor sets
were shown in an automaton which contains all the subsets of
sensors such that the estimation objective can be achieved.
They introduced three criteria evaluating the system fault
tolerance with respect to sensor failures when a
reconfiguration strategy is used: weak redundancy degree
(RD), sensor network reliability (R), and mean-time to non-
observability (MTTNO). Sensor networks are designed by
finding redundant sensor sets whose RD and/or R and/or
MTTNO are larger than some specified values. Their
regressive method works well on small designs in which
design algorithm only cares about the existence of a sensor
on the variable. However, when it comes to designing of
networks with multiple sensors available to measure a
variable, due to the drastic increase of number of possible
solutions, the calculation effort highly increases and it fails to
work. In addition, a main criterion in instrumentation design
procedure, i.e. cost of instrumentation, has been neglected in
this approach.

In order to address the mentioned issues altogether, we have
presented a new instrumentation design methodology for
cost-optimal and fault-tolerant sensor networks which is
more comprehensive, flexible and practical than other
designs given in the literature. In the proposed method,
instead of following the regressive method which uses a
regular and determined approach to check all the possible
nodes that fulfill the constraints, a search engine is used.
In this method, search engine does not examine all solutions to
find the most optimal one, so time consumption decreases
considerably. This is a benefit that allows designer to involve
more variety of sensors in design in addition to facilitating
the design to be applied in large-scale designs for the sake of
saved time caused by using the CPSO. Moreover, cost-
related considerations as well as the fault tolerance criteria
have been involved in the design procedure.

In this paper, first the necessary terms, e.g. redundancy
degree and reliability of networks, are briefly introduced
(Staroswieck et al, 2004). The proposed fault tolerant
estimation design algorithm is presented in the next section.
Finally, the presented algorithm is implemented in a CSTR
case study including a set of 15 process variables to illustrate
its specified capabilities.

## 2. FAULT TOLERANCE ASSESSMENT

### 2.1 Minamility and Redundancy

Consider the continuous time deterministic system:

\[
\dot{x} = f(x(t), u(t))
\]

\[
y(t) = g(x(t))
\]

\[
z(t) = h(x(t))
\]

where \( x \in \mathbb{R}^n \) is the state vector, \( u \in \mathbb{R}^m \) is the control
input, \( y \in \mathbb{R}^p \) is the measurement vector, and \( z \in \mathbb{R}^q \) is the
functional of the state which is to be estimated. The inputs
\( u(t) \) are assumed to be sufficiently differentiable and \( f, g, h \)
are sufficiently smooth vector fields. Let \( J \subseteq \mathbb{R} \) be a subset of
the system sensors, and introduce the notation \( \text{obsv}(z/J) \)
where (for a given definition of observability):

\[
\text{obsv}(z/J) = \begin{cases} 1 & \text{if } z \text{ is observable with } J \\ 0 & \text{otherwise} \end{cases}
\]

Let \( 2^\mathbb{R} \) be the set of all subsets of \( \mathbb{R} \); then (4) induces a two-
class partition:

\[
2^R^+ = \{ J \subseteq \mathbb{R} ; \text{obsv}(z/J) = 1 \}
\]

\[
2^R^- = \{ J \subseteq \mathbb{R} ; \text{obsv}(z/J) = 0 \}
\]

The class \( 2^R^+ \) contains all the subsets of sensors by which \( z \) is
observable, and it is assumed that \( R \in 2^R^+ \); i.e. the system is
observable by the whole set of sensors. Accordingly, minimal
sensor set and redundant sensor sets are defined a following:

A subset of sensors \( J \in 2^R^+ \) is minimal (MSS), if

\[
\forall k \subseteq J \quad k \not\in 2^R^+
\]

and a subset of sensors \( J \in 2^R^- \) is redundant (RSS), iff it is not
minimal.

### 2.2 Interpretation of fault tolerance

Assume that one or several sensor failures occur at time \( t_f \)
so that the set of sensors \( J \) can be decomposed into the normal
and the faulty ones: \( J = J_n \cup J_f \). Therefore, the
measurement equations can be written

\[
y_n(t) = g_n(x(t))
\]

\[
y_f(t) = g_f(x(t))
\]

where \( y_n \) (resp. \( y_f \) ) represent the normal (resp. the faulty)
outputs of the sensor network \( J \) and \( g_n \) (resp. \( g_f \) ) are the
normal (resp. the faulty) measurement equations. The fault
tolerance problem used in this paper can be interpreted as
follows: the faulty sensors \( J_f \) are switched off, and the
problem is to assess the possibility of still estimating the
functional \( z \) by using the remaining set of sensors \( J_n \) which is
indeed true, provided system is still observable. This method
that is named reconfiguration strategy only needs fault
detection and isolation (fault estimation is not necessary), and that the fault tolerance property is a structural one, since it is associated with triple (1), (7) and (8), it does not depend on the type of fault which affects the sensors $J_f$. In this paper, we consider only the reconfiguration strategy. Now that the interpretation of fault tolerance was presented, the redundancy degree and reliability are defined in the next sections.

2.3 Weak Redundancy Degree

Let $J \subseteq I$ be any subset of the sensors, i.e. some state of the system instrumentation. Let $K \in \text{MSS}(J)$, then the quantity $|J/K|$ represents the maximal number of sensors which can be lost while $z$ can still be estimated by $K$. In the ‘best’ situation, as many sensor losses as

$$|J| - \min_{K \in \text{MSS}(J)} |K|$$

(9)
can be accepted. The weak redundancy degree evaluates the size of this ‘best’ situation.

The weak redundancy degree associated with the pair $(z, J)$ is

$$\text{WRD}(z, J) = |J| - \min_{K \in \text{MSS}(J)} |K|$$

(10)

From the interpretation of $\text{WRD}(z, J)$ it follows that the following statement is true:

$$\exists J' \subseteq J \text{ such that } |J \setminus J'| = \text{WRD}(z, J)$$

and $J' \in \text{MSS}(J)$

(11)

Of course, in many cases, $z$ will no longer be observable after less than $\text{WRZ}(z, J)$ sensors are lost.

2.4 Availability of the Estimation Service

Let $t_0=0$ be the time at which the system operation was started, and let J(t) be the subset of the non-faulty (available) sensors at time t. Let $J_0 = \{0\}$, assuming such data to be available, the fault tolerance of the $z$-estimation process can be evaluated by the probability for the estimation of $z$ to be possible during the given time interval $[0,t]$ assuming that it was possible using the set $J_0$ at time 0, $P(z/J_0)$. Let $J \subseteq J_0$ be any subset of sensors. The probability for the estimation of $z$ to be possible during the time interval $[0,t]$ using $K$ is given by (12):

$$R(z/J,t) = P(z/K).R(K,t)$$

(12)

where $P(z/K) = 1$ if $K$ is a MSS or a RSS and $P(z/K) = 0$ otherwise, and $R(K,t)$ is the reliability of the set of sensors $K$; which is defined as the probability that no sensor of $K$ fails during the interval $[0,t]$. If sensor failures are independent, i.e. there is no common mode failure, one has

$$R(K,t) = \prod_{k \in K} R_k(t) \prod_{k \in K} (1 - R_k(t))$$

(13)

where $R_k(t)$ is sensor $k$ reliability. The reliability of such individual components is often modeled using the Poisson distribution:

$$R_k(t) = e^{-\lambda_k t}$$

(14)

where $\lambda_k$ is sensor $k$ failure rate, which is supposed to be constant.

Now, considering the whole set $J_0$, it follows from the fact that all its subsets $K$ are exclusive, that the probability for the estimation of $z$ to be possible during the time interval $[0,t]$ is given by

$$R(z/J_0,t) = \sum_{K \subseteq J_0} P(z/K).R(K,t)$$

(15)

In (15), $P(z/K)$ is 1 if subset $K$ is observable and 0 if not.

3. DESIGN PROCEDURE

Cost, precision and reliability are fundamental characteristics of an instrumentation network. Accordingly, different models can be constructed by employing any combination of these networks. One common model which has been used commonly in the literature minimizes cost of constructing satisfying network reliability and fault tolerance constraints. This model is called minimum cost model and can be shown by:

$$\min \sum_{j} (C_j S_{ji})$$

$$\text{s.t. } \begin{cases} \sum_j R_j(z/K) \geq R^* \\ \text{WRD} \geq \text{WRD}^* \end{cases}$$

(16)

where $S_{ji}$ represents the integer number showing the placement of the variable of sensor type $j$ at network location $i$. The reliability of the network should be evaluated according to the specified variables which should be observed. These variables can be either whole or part of the network states. If the case study is not large, the best solution to these types of optimization problems is using the bottom to top algorithm introduced in (Staroswiecki et al., 2004). In large-scaled or medium-scaled plants, where there are a lot of topologies to study, implementing such algorithms takes a lot of time and fails to be successful. In order to solve this problem, a search engine is used to investigate possible solutions and determine which network is the most optimized solution to the model (16). Note that although the algorithm suggested in (Staroswiecki et al., 2004) for fault tolerant instrumentation is not directly stated as the optimization problem, it can be considered a special case of model (16). Consider that the price of all sensors is equal.

$$C_j = C \text{ for } j = 1, 2, ..., n$$

(17)

In this case, overall cost of a network which has $n$ variables to be measured is:

$$\text{Total Cost} = \sum_j (C_j S_{ji}) = n.C$$

(18)

Therefore, minimizing the number of sensors ($n$) used in (Staroswiecki et al., 2004) is equivalent to minimizing the cost of sensors, i.e. the model used in (Staroswiecki et al., 2004) can be considered as a special case of model (16).
To solve the presented model in (16), a combinatorial search engine, CPSO (Jaboui et al., 2004), is proposed to search for the best solution satisfying the optimization problem. The block diagram of such an algorithm has been depicted in Fig. 1. CPSO will check observability, redundancy degree and reliability of the network for every particle in each iteration. If these values fulfill the problem requirements, CPSO allows the particle to survive in the engine; otherwise, it will put away the particle and choose another one as a substitute. This scenario goes on until the best solution is suggested by CPSO.

Fig. 1. Block Diagram of the design algorithm aims to search a network satisfying (16)

4. IMPLEMENTATION ON THE CASE STUDY AND RESULTS

The case study used in this paper is the CSTR that Bhushan and Rengaswamy (2000b) introduced in their article (Fig. 1). This process involves an exothermic liquid-phase reaction. The model parameters along with their nominal operating values are presented in (Bhushan and Rengaswamy, 2000b).

In this case study, the measurable variables are V, CA, T, TC, P, Fp, Fg, Fq, Fv, Fc, Fi, Tu, TCi, Tci, F2, and F3. Thus, in order to incorporate variety in design, three different sets of sensors are considered for each sensor type. The failure rates along with corresponding costs of these sensors have been tabulated at Table 1. In the final step, an efficient modified CPSO algorithm is used (Jaboui et al., 2004). Parameters of search engine are set as follows: $\omega=1.1$, $c_1=0.6$, $c_2=0.5$, $v_{\text{max}}=2$, $v_{\text{min}}=-2$ and $\omega=1.2$. Moreover, 20 particles and 100 iterations have been considered to run the CPSO. Now, the proposed design procedure is conducted to solve the problem in model (16) with five different constraints. In all test runs, the WRD constraint is set four. The reliability constraints vary from 0.7 to 0.85, but the first case does not include any reliability constraint. The search is undertaken ten times so that whether the suggested solution satisfies all the design requirements or not. On the other hand, in these designs, 15 variables should be monitored; so many feasible networks can be constructed by combination of measured variables. Accordingly, performing an exhaustive search to verify the results is not practical. If we consider different possible types of sensors given in Table 1 to measure each variable, there will be 384,422,112 observable networks. The mentioned issues necessitate applying appropriate verification tests on the obtained results. In order to cope with mentioned issue, three tests are conducted to verify whether the obtained solutions satisfy our design requirements or not.

The overall time for conducting such an experiment was 3 h and 27 min. If we assume conducting experiment for one single network through MATLAB software and a machine with 2.5 GHz of CPU and 2 GB of RAM takes 0.1036 sec on average, performing an exhaustive search for such a plant will take more than 460 days!

In order to have a visual measure to compare the cost of all the solutions obtained by CPSO with each reliability constraint, a diagram showing the mutual relationship between network costs and their corresponding reliabilities has been provided in Fig. 3. The upper and lower lines that are shown in bold represent the worst and best solutions obtained by CPSO and the other lines shown in green, represent the other 9 solutions. As seen, there is almost a linear relationship between cost and reliability constraint. The diagram maintains its linearity until reliability reaches 0.85. For higher constraint values, there is a sharp jump. In other words, in order to have an increase of %2 in the reliability of the network at the end point of the diagram, there is an extra cost of $25000!

The proposed CPSO algorithm was utilized to offer the best possible networks enforced to fulfill the requirements of the design problem, but there is no guarantee for the optimality of the obtained solutions and there is no available tool that can assure us the obtained solutions are the optimal network among all the other practical networks that can be built by the given sets of sensors. On the other hand, it should be checked that whether the suggested solution satisfies all the design requirements or not. On the other hand, in these designs, 15 variables should be monitored; so many feasible networks can be constructed by combination of measured variables. Accordingly, performing an exhaustive search to verify the results is not practical. If we consider different possible types of sensors given in Table 1 to measure each variable, there will be 384,422,112 observable networks. The mentioned issues necessitate applying appropriate verification tests on the obtained results. In order to cope with mentioned issue, three tests are conducted to verify whether the obtained solutions satisfy our design requirements or not.

In order to have a visual measure to compare the cost of all the solutions obtained by CPSO with each reliability constraint, a diagram showing the mutual relationship between network costs and their corresponding reliabilities has been provided in Fig. 3. The upper and lower lines that are shown in bold represent the worst and best solutions obtained by CPSO and the other lines shown in green, represent the other 9 solutions. As seen, there is almost a linear relationship between cost and reliability constraint. The diagram maintains its linearity until reliability reaches 0.85. For higher constraint values, there is a sharp jump. In other words, in order to have an increase of %2 in the reliability of the network at the end point of the diagram, there is an extra cost of $25000!

The proposed CPSO algorithm was utilized to offer the best possible networks enforced to fulfill the requirements of the design problem, but there is no guarantee for the optimality of the obtained solutions and there is no available tool that can assure us the obtained solutions are the optimal network among all the other practical networks that can be built by the given sets of sensors. On the other hand, it should be checked that whether the suggested solution satisfies all the design requirements or not. On the other hand, in these designs, 15 variables should be monitored; so many feasible networks can be constructed by combination of measured variables. Accordingly, performing an exhaustive search to verify the results is not practical. If we consider different possible types of sensors given in Table 1 to measure each variable, there will be 384,422,112 observable networks. The mentioned issues necessitate applying appropriate verification tests on the obtained results. In order to cope with mentioned issue, three tests are conducted to verify whether the obtained solutions satisfy our design requirements or not.

The overall time for conducting such an experiment was 3 h and 27 min. If we assume conducting experiment for one single network through MATLAB software and a machine with 2.5 GHz of CPU and 2 GB of RAM takes 0.1036 sec on average, performing an exhaustive search for such a plant will take more than 460 days!

In order to have a visual measure to compare the cost of all the solutions obtained by CPSO with each reliability constraint, a diagram showing the mutual relationship between network costs and their corresponding reliabilities has been provided in Fig. 3. The upper and lower lines that are shown in bold represent the worst and best solutions obtained by CPSO and the other lines shown in green, represent the other 9 solutions. As seen, there is almost a linear relationship between cost and reliability constraint. The diagram maintains its linearity until reliability reaches 0.85. For higher constraint values, there is a sharp jump. In other words, in order to have an increase of %2 in the reliability of the network at the end point of the diagram, there is an extra cost of $25000!

The proposed CPSO algorithm was utilized to offer the best possible networks enforced to fulfill the requirements of the design problem, but there is no guarantee for the optimality of the obtained solutions and there is no available tool that can assure us the obtained solutions are the optimal network among all the other practical networks that can be built by the given sets of sensors. On the other hand, it should be checked that whether the suggested solution satisfies all the design requirements or not. On the other hand, in these designs, 15 variables should be monitored; so many feasible networks can be constructed by combination of measured variables. Accordingly, performing an exhaustive search to verify the results is not practical. If we consider different possible types of sensors given in Table 1 to measure each variable, there will be 384,422,112 observable networks. The mentioned issues necessitate applying appropriate verification tests on the obtained results. In order to cope with mentioned issue, three tests are conducted to verify whether the obtained solutions satisfy our design requirements or not.

The overall time for conducting such an experiment was 3 h and 27 min. If we assume conducting experiment for one single network through MATLAB software and a machine with 2.5 GHz of CPU and 2 GB of RAM takes 0.1036 sec on average, performing an exhaustive search for such a plant will take more than 460 days!
Table 1. Available sensors for instrumentation design: Volume; C: Concentration; T: Temperature; P: Pressure; F: Flow

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Failure Rate (×10^{-2})</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_1</td>
<td>1/81</td>
<td>3000</td>
</tr>
<tr>
<td>V_2</td>
<td>1/50</td>
<td>2200</td>
</tr>
<tr>
<td>V_3</td>
<td>1/28</td>
<td>1600</td>
</tr>
<tr>
<td>C_1</td>
<td>1/80</td>
<td>2500</td>
</tr>
<tr>
<td>C_2</td>
<td>1/50</td>
<td>1800</td>
</tr>
<tr>
<td>C_3</td>
<td>1/28</td>
<td>800</td>
</tr>
<tr>
<td>T_1</td>
<td>1/88</td>
<td>1500</td>
</tr>
<tr>
<td>T_2</td>
<td>1/49</td>
<td>400</td>
</tr>
<tr>
<td>T_3</td>
<td>1/23</td>
<td>1000</td>
</tr>
<tr>
<td>P_1</td>
<td>1/90</td>
<td>1400</td>
</tr>
<tr>
<td>P_2</td>
<td>1/48</td>
<td>800</td>
</tr>
<tr>
<td>P_3</td>
<td>1/22</td>
<td>4500</td>
</tr>
<tr>
<td>F_1</td>
<td>1/82</td>
<td>1400</td>
</tr>
<tr>
<td>F_2</td>
<td>1/44</td>
<td>1000</td>
</tr>
<tr>
<td>F_3</td>
<td>1/2</td>
<td>3000</td>
</tr>
</tbody>
</table>

Table 2. The best and worst solutions obtained for different reliability constraints

<table>
<thead>
<tr>
<th>Reliability Constraint</th>
<th>V</th>
<th>CA</th>
<th>Tc</th>
<th>P</th>
<th>F_4</th>
<th>F_vg</th>
<th>F_C</th>
<th>F_i</th>
<th>Ti</th>
<th>CA_i</th>
<th>Tci</th>
<th>F_2</th>
<th>F_3</th>
<th>WRD</th>
<th>R</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.45</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>×</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.5</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.55</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.6</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.65</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.7</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.75</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.8</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.85</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3. Weak redundancy degree verification for five design sets

<table>
<thead>
<tr>
<th>Reliability Constraint</th>
<th>V</th>
<th>CA</th>
<th>Tc</th>
<th>P</th>
<th>F_4</th>
<th>F_vg</th>
<th>F_C</th>
<th>F_i</th>
<th>Ti</th>
<th>CA_i</th>
<th>Tci</th>
<th>F_2</th>
<th>F_3</th>
<th>WRD</th>
<th>R</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.75</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.8</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.85</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 3. Cost of the networks versus reliability constraints, the two lines shown in bold are the worst and best solutions.

Obviously, it is not possible to perform a comprehensive verification for our design results. However, in order to assess the presented approach performance, 500 randomly chosen networks are shown with their corresponding costs and reliability values in Fig. 4. In this figure the circles represent the solutions suggested by the CPSO that all are below or as high as the dots. Although this type of verification can not completely approve the results, it can lend additional support to the performance of the presented method. Likewise, for reliability validation, we took a number of networks and let their sensors fail according to their reliability values to see whether the new obtained network is observable or not.

In reliability validation, the ratio of the observable network to the total number of networks determines the statistical reliability. The diagram of statistic reliability versus number of experiments for all searches is depicted in Fig. 5. It can be observed that as the number of experiments increases, reliability curve approaches the corresponding reliability constraints. Note that because number of repetitions is not large enough, the statistic reliability obtained with ten test trials cannot be valid for reliability validation, but as more experiments are performed the reliability approaches the expected value.

Reliability and cost validations have been investigated. Table 3 shows the verification results for weak redundancy degree.
For each design, two networks have been shown. The first is the initial one, indicating the main solution suggested by the CPSO algorithm, while the second one represents the network that has been obtained after four sensor failures in the initial network. Examining Table 3 infers that the networks have been remained observable after specified failures appear in the sensor sets. Of course, the networks with fewer failures that are located between these two networks are observable too. For instance, consider the solution suggested by the CPSO in Table 3 for the first design with no reliability constraint in which a sequence of four failures: S_{12}, S_5, S_3 and S_2 can occur in the initial network. In each sensor failure, a new observable network is obtained. This procedure goes on until it reaches a node that has the minimal number of sensors and hence no extra sensor failure can occur, indicating that sequence of failures ends at this node. The number of failures in this sequence determines the weak redundancy degree of initial network which is four in all the designs.

![Fig. 5. Statistic Reliability calculated to validate the results, as seen all diagrams approaches the constraints used in the designs as the number of experiments increases](image)

5. CONCLUSIONS

A new design methodology has been proposed in this paper based on the CPSO algorithm. This facilitates the employment of a sensor network design perspective to complex large-scale plants in which cost considerations are included as the main design objective while incorporating fault-tolerant properties, leading to a new optimal redundant instrumentation sensor network design. This enables to maintain a desired fault-tolerant redundancy characteristic in a specific industrial environment to cope with a possible set of sensor failures. Different test scenarios carried out in a CSTR benchmark problem illustrated the inherent capabilities of the proposed sensor network design methodology. However, this instrumentation design can be applied in other models with more variable to measure. Instrumentation design for fault tolerant and precise sensor networks is in our line of work at the moment.

REFERENCES


