Sensor location optimization for fault diagnosis with a comparison to linear programming approaches

M. A. Rahim[1], Haris M. Khalid[2] and Muhammad Akram[2]

Abstract The critical importance of sustaining fault diagnosis, as a major system tool, is unquestionable if the high performance and reliability of increasingly-complex engineering systems is to be sustained over time and across a wide operating range. However, it is quite difficult to retain the joint ability of fault detection and isolation as it requires a strong system architecture. That is why, before designing an industrial supervision system, the determination of a system's monitoring ability based on technical specifications is important, as finding the source of the failure is not trivial in systems with a large number of components and complex component relationships. This paper presents an efficient and cost-effective fault detection and isolation (FDI) scheme that evolved from an earlier work [1]. FDI specifications are translated into constraints of the optimization problem considering that the whole set of analytical redundancy relations has been generated, under the assumption that all candidate sensors are installed and later on tested by an optimization algorithm using binary and relaxed versions of linear and non-linear programming. By doing so, the critical information about the presence or absence of a fault is gained in the shortest possible time, with not only confirmation of the findings, but also an accurate unfolding-in-time of the finer details of the fault, thus completing the overall diagnostic picture of the system under test. The proposed scheme is evaluated extensively on a two-tank process used in industry, exemplified by a benchmarked laboratory scale coupled-tank system.

Keywords Sensor location. Optimization. Fault detection and isolation. Analytical redundancy relations. Linear programming. Benchmarked laboratory-scaled two-tank system

1 Introduction

Process faults, if undetected, have a serious impact on process economy, product quality, safety, and productivity and pollution levels. In order to detect, diagnose and correct these abnormal process behaviors, efficient and advanced automated diagnostic systems are of great importance to modern industries. Fault diagnosis and process supervision are an increasingly important topic in many industrial applications and also in an active academic research area. Considerable research has gone into the development of such diagnostic systems [2]. Most approaches for fault detection and isolation (FDI) in some sense involve the comparison of the observed behavior of the process to a reference model. While probing into this research area more, the problem of monitoring the quality of the plant through diagnosis of unprecedented changes has been the subject of intense research [3] as well as finding an optimal control while tackling such non-linear constrained procedures [4].

The process behavior is inferred using sensors measuring the important variables in the process. Hence, the efficiency of the diagnostic approach depends critically on the location of sensors used for monitoring process variables. The emphasis of most of the work on model-based fault diagnosis has been more on procedures to perform diagnosis given a set of sensors and less on the actual location of sensors for efficient identification of faults. The problem of sensor placement for FDI

[1]M.A. Rahim
Faculty of Business Administration,
University of New Brunswick, Fredericton,
E3B 5A3, New Brunswick, Canada
email: rahim@unb.ca

[2]Haris M. Khalid (Corresponding Author), M. Akram
Department of Systems Engineering,
King Fahd University of Petroleum and Minerals,
Dhahran 31261, Saudi Arabia
email: mhariskh@kfupm.edu.sa

consists of determining the optimal set of instruments such that a predefined set of faults are detected and isolated. In many cases, this set is defined in order to design some remedial actions such that the control loop is able to continue operating even in the presence of a fault (fault-tolerant control).

NOMENCLATURE	
Q_i is the initial flow of water	β is associated with the leakage $q_0 = \beta h_2$
Q_l is the increment in the nominal (leakage-free) of flow Q_l^0	q_v is the Valve Flow
Q_0 is the output flowrate	q_p is the Pump Flow
$\varphi(u)$ is the dead-band and saturation type of non-linearity	u_v is the Valve Control Input
a_m is the motor system	u_p is the Pump Control Input
b_m is the pump system	f_1 is the tank 1 leak
u is the input to the motor	f_2 is the tank 2 leak
Q_l is the leakage in Tank 1	f_{h1} is the wrong tank 1 level sensor reading
C_0 is the Discharge coefficient of output valves	f_{h2} is the wrong tank 2 level sensor reading
C_{12} is the Discharge coefficient of Inter tank	f_{qv} is the wrong valve flow sensor reading
A_1 is the Cross-Sectional Area of Tank1	f_{qp} is the wrong sensor flow sensor reading
A_2 is the Cross-Sectional Area of Tank2	f_{uv} is the wrong valve control input sensor reading
H_1 is the Height of Tank1	f_{up} is the wrong pump control input sensor reading
H_2 is the Height of Tank2	q is the vector of binary elements
$g=980\ cm/sec^2$ is the gravitational constant	m is the total number of candidate sensors
h_1 is the increment in the nominal (leakage-free) of height H_1^0	w_j is the cost of sensor s_j comprising purchase, maintenance
h_2 is the increment in the nominal (leakage-free) of height H_2^0	y_{si} is the measured sensor output
q_0 is the increment in Q_0	y_{si}^0 is the true or fault-free output
q_i is the increment in Q_i	υ_i *is the additive noise*
q_l is the increment in Q_l	k_{si} is the gain in sensor-network
k_p is the proportional gain in PI Controller	G_i is the subsystem monitoring the sensor-outputs y_i
k_I is the Integral gain in PI Controller	y_i^{ss} is the steady state values of the sensor output y_i
α is associated with the leakage $q_l = \alpha h_1$	$\hat{M}_{ik}$ and $\hat{M}_{il}$ are the matrices generated from the F_D^I and F_D^{II}
f_p are the process faults	ρ_i be the binary ARR selector
f_s are the sensor faults	

2 Related works

In sensor location optimization, the usual objective to minimize in the sensor placement problem is the sensor cost. There are several articles devoted to the study of the design of sensor networks using goals corresponding to normal monitoring operation. Aside from cost, other objective functions, such as precision, reliability, or simply observability were used. Different techniques were also used, such as graph theory, mathematical programming, genetic algorithms and multi-objective optimization, among others. The problem has also been extended to incorporate upgrade considerations and maintenance costs. In [5], it is noticed that the problem of sensor placement in the model-based FDI community is still an open problem. However, some contributions have already been made in this direction [6-12], among others.

In model-based FDI, faults are modeled as deviations of parameter values or unknown signals, and diagnostic models are, in such cases, often brought back to a residual form. For works based on continuous differential/difference-equation-based models (e.g., see [2] and [13] and the references therein for discrete-event models [14-15] and for diagnosis of hybrid systems [16]. To be able to perform model-based supervision, some redundancy is needed, and this redundancy is typically provided by sensors mounted on the process. Scientific attention has mainly been devoted to design a diagnosis system, given a model of a process equipped with a set of sensors. Special attention is needed to decide which sensors to include in the process. Deciding where to put sensors correctly, which makes it possible to meet a given diagnosis performance specification, is the topic of this paper. There are many types of performance measures in diagnosis, for example, detection performance, false alarm probabilities, time to detection, etc. In this paper, sensors are placed such that maximum isolability is possible, i.e., faults in different components should, as far as possible and desired, be able to be isolated from each other. Since sensor placement is often done early in the design phase, possibly before a reliable process model can be developed, the method developed in this paper is based on a structural process model.

The main approaches to construct residuals are based on using analytical redundancy relations (ARRs) generated either using the parity space [17] or observer approaches [18]. In [19] the sensor placement problem is solved by the analysis of a set of possible ARRs using algorithms of cycle generation in graphs. Some other results devoted to sensor placement for diagnosis using graph tools can be found in [20-24]. All these works use a structural model-based approach and define different diagnosis specifications to solve the sensor placement problem. In [25], the sensor placement problem is solved by the analysis of a set of possible ARR using algorithms of cycle generation in graphs.

In [1], an optimal sensor placement is introduced for model-based FDI, which requires finding the set of all possible ARRs and considering that all possible candidate sensors are installed. Then, a set of sensors that minimizes the total cost of the network is selected such that the resulting ARRs satisfy that a pre-established set of faults can be detected and isolated. For sensor placement, it is required to use an ARR generation algorithm that is complete. Otherwise, the sensor placement could exclude from consideration some sensor configurations just because some ARRs were not generated. Excluded configurations could provide better FDI results that the ones that were generated. Or, even in some dramatic cases, the sensor placement could not find a solution because of this lack of completeness, whereas, in fact, if all ARRs were generated a solution would have been found.

The structure of the paper is as follows. Section 1 provides the introduction and section 2 gives the details of the related works. In section 3, the sensor location optimization problem statement is presented. In section 4, the implementation and results are shown. Finally, some conclusions and extensions are suggested in section 5.

3 Sensor location optimization problem statements

A most critical and important issue surrounding the design of automatic control systems with successively increasing complexity, is guaranteeing a high system performance over a wide operating range and meeting the requirements of system reliability and dependability. To have an effective and optimal implementation of this performance, an optimal sensor placement is required.

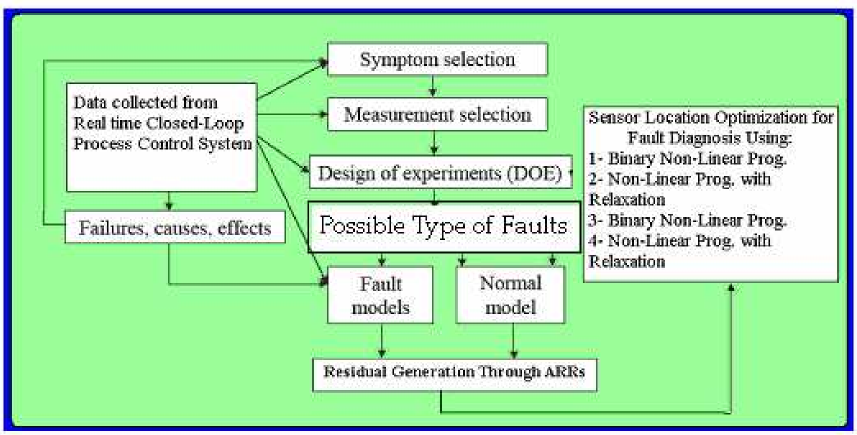

Fig. 1 Proposed scheme

In this paper, a sensor location optimization approach is proposed to meet the requirements for a quick and reliable fault detection and isolation scheme and thus promoting to a FDI Optimization. The tasks of our fault diagnosis scheme (See Fig.1) are executed by the implication of binary and relaxed versions of linear and non-linear programming, by targeting the optimal sensor placement and an optimal objective cost value. The proposed scheme has been evaluated on a laboratory scaled based two-tank system. It is the most used prototype applied in the wastewater treatment plant, the petro-chemical plant, and the oil/gas systems.

3.1 Model of the coupled tank system

The physical system under evaluation is formed of two tanks connected by a pipe. The leakage is simulated in the tank by opening the drain valve. A DC motor-driven pump supplies the fluid to the first tank and a PI controller is used to control the fluid level in the second tank by maintaining the level at a specified level, as shown in the figure (see Fig. 2).

A step input is applied to the DC motor- pump system to fill the first tank. The opening of the drainage valve faults introduces a leakage in the tank. Various types of leakage are introduced and the liquid height in the second tank, H_2 and the inflow rate, Q_i are both measured. The National Instruments LABVIEW package is employed to collect these data.

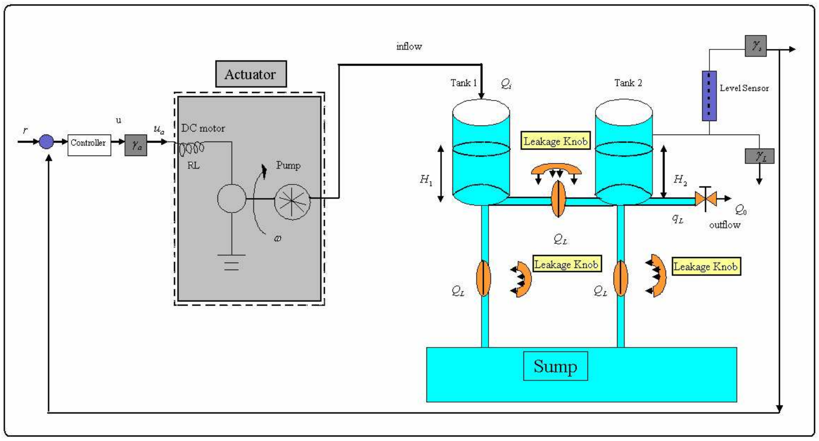

Fig. 2 Two-tank model

As mentioned earlier, various types of leakages were introduced by opening the drainage valve and the liquid height profiles in the second tank were subsequently analyzed. Three variables being measured in this process are hydraulic height, hydraulic flow and the control output. In all, there are four internal variables and two input variables in the system, as summarized (see Table 1). So the candidate sensor set comprises up to six sensors $S = \{h_u, h_l, q_v, q_p, u_v, u_p\}$. Eight hypothetical faults are considered in the system (see Table 2): leaks in the tank 1 and tank 2, and wrong readings of each candidate sensor. So the fault sets are $F = F_P \cup F_S = \{f_u, f_l\} \cup \{f_{hu}, f_{hl}, f_{qv}, f_{qp}, f_{uv}, f_{up}\}$ where F_p stands for Process faults and F_s stands for sensor faults.

Table 1 Variables of the coupled tank system

Variable	Description
h_1	*Tank 1 level*
h_2	*Tank 2 level*
q_v	*Valve flow*
q_p	*Pump flow*
u_v	*Valve control input*
u_p	*Pump control input*

Table 2 Hypothetical faults of the system

Variable	Description
f_1	*Tank 1 leak*
f_2	*Tank 2 leak*
f_{h1}	*Wrong tank 1 level sensor reading*
f_{h2}	*Wrong tank 2 level sensor reading*
f_{qv}	*Wrong valve flow sensor reading*
f_{qp}	*Wrong sensor flow sensor reading*
f_{uv}	*Wrong valve control input sensor reading*
f_{up}	*Wrong pump control input sensor reading*

A benchmark model of a cascade connection of a DC motor and a pump relating the input to the motor, *u,* and the flow, Q_i, is a first-order system:

$$\dot{Q}_i = -a_m Q_i + b_m \phi(u) \qquad (1)$$

Where a_m and b_m are the parameters of the motor-pump system and $\phi(u)$ is a dead-band and saturation type of nonlinearity. It is assumed that the leakage Q_ℓ occurs in tank 1 and is given by:

$$Q_\ell = C_{d\ell}\sqrt{2gH_1} \qquad (2)$$

With the inclusion of the leakage, the liquid level system is modeled by:

$$A_1\frac{dH_1}{dt} = Q_i - C_{12}\varphi\left(H_1 - H_2\right) - C_\ell\varphi\left(H_1\right) \qquad (3)$$

$$A_2\frac{dH_2}{dt} = C_{12}\varphi\left(H_1 - H_2\right) - C_0\varphi\left(H_2\right) \qquad (4)$$

Where $\varphi(.) = sign(.)\sqrt{2g(.)}$, $Q_\ell = C_\ell\varphi\left(H_1\right)$ is the leakage flow rate, $Q_0 = C_0\varphi\left(H_2\right)$ is the output flow rate, H_1 is the height of the liquid in tank 1, H_2 is the height of the liquid in tank 2, A_1 and A_2 are the cross-sectional areas of the 2 tanks, g=980 $cm/\sec^2$ is the gravitational constant, C_{12} and C_o are the discharge coefficient of the inner-tank and output valves, respectively.

The model of the two-tank fluid control system, shown above in Fig. 2, is of a second order and is nonlinear with a smooth square-root type of nonlinearity. For design purposes, a linearized model of the fluid system is required and is given below in (5) and (6):

$$\frac{dh_1}{dt} = b_1 q_i - \left(a_1 + \alpha\right)h_1 + a_1 h_2 \qquad (5)$$

$$\frac{dh_2}{dt} = a_2 h_1 - \left(a_2 - \beta\right)h_2 \qquad (6)$$

Where h_1 and h_2 are the increments in the nominal (leakage-free) heights H_1^0 and H_2^0 :

$$b_1 = \frac{1}{A_1}, \quad a_1 = \frac{C_{db}}{2\sqrt{2g(H_1^0 - H_2^0)}}, \quad \beta = \frac{C_0}{2\sqrt{2gH_2^0}}, a_2 = a_1 + \frac{C_{do}}{2\sqrt{2gH_2^0}} \quad \alpha = \frac{C_{d\ell}}{2\sqrt{2gH_1^0}}$$

and the parameter α indicates the amount of leakage.

A PI controller, with gains k_p and k_I, is used to maintain the level of the tank 2 at the desired reference input r .

Where q_i , q_ℓ , q_0 , h_1 and h_2 are the increments in Q_i , Q_ℓ , Q_o , H_1^0 and H_2^0, respectively, the parameters a_1 and a_2 are associated with linearization, whereas the parameters α and β are respectively associated with the leakage and the output flow rate, i.e. $q_\ell = \alpha h_1$, $q_o = \beta h_2$.

4 Implementation and results

4.1 Generation of ARR table using fuzzy rules

The main approaches to construct the residuals using ARRs generated are either using parity space or observer-based approaches. The approach developed over here for the ARR generations is by using the observer-based technique improvised by using a sensor network (see Fig. 3).

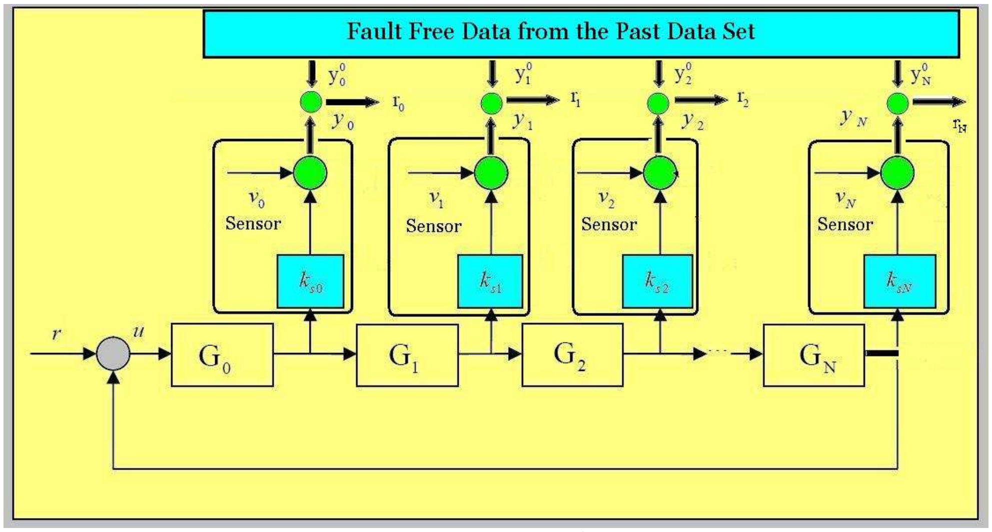

Fig. 3 Sensor network

A sensor is modeled by a gain and an additive noise, as given below:

$$y_i = k_{si} y_i^0 + v_i \tag{7}$$

Where y_{si} , y_{si}^0 and v_i are the measured sensor output, true or fault-free output and additive noise, respectively.

Here the gain is such that $0 \le k_{si} \le 1$, with the degree of the fault ranging from no fault at all for $k_{si} = 1$ to a complete failure for $k_{si} = 0$. The subsystems such as actuators, processors and controllers are denoted by transfer functions, G_i . Many systems consisting of several closed loops, each with its own reference input, can be viewed as a sensor network that can be described by a ring-type topology.

The objective of the sensor network is to diagnose faults in both the sensors, through the gains k_{si} and in the subsystems G_i by monitoring the sensor outputs y_i .

The mathematical relations governing the sensor outputs y_i to the input to G_0 , denoted by e are:

$$\begin{aligned} y_1 &= G_0 k_{s0} e + v_0 \\ y_2 &= G_0 G_1 k_{s1} e + v_1 \\ y_3 &= G_0 G_1 G_2 k_{s2} e + v_2 \\ &. \\ &. \\ y_i &= G_0 G_1 G_2 ... G_{i-1} k_{s(i-1)} e + v_{i-1} \end{aligned} \tag{8}$$

Where $e = r - y$.

The fuzzy rules are being defined by using the steady-state values of the sensor outputs, y_i , denoted by y_i^{ss} . A change in the gain k_{si} , or a change in the steady-state gain of the transfer function G_i , denoted by G_i^{ss} , is indicative of a fault in the i-th sensor and i-th subsystem, respectively (see Fig.3). Assuming that the noise term is subsumed in the fuzzy membership function, the steady-state model takes the form:

$$y_i^{ss} = G_0^{ss} G_1^{ss} G_2^{ss} ... G_{i-1}^{ss} k_{s(i-1)} e \tag{9}$$

Let us now define linguistic variables such as *zero*, and *non-zero*. For simplicity, we will consider the case where only one device can be faulty at any given time, i.e. the fault is assumed to be simple. In this case, the fuzzy rules may take the following form:

Rule 1: If y_i^{ss} is non-zero, then there is a fault in the steady-state gain G_0^{ss} or G_1^{ss} or G_2^{ss} or…or G_i^{ss} or ith sensor gain k_{si}

Rule II: If y_i^{ss} is zero, then there is no fault in the subsystem's steady-state gain G_0^{ss} or G_1^{ss} or G_2^{ss} or…or G_i^{ss} or ith sensor gain k_{si}

Rule III: If y_i^{ss} is zero and $y_{s(i+1)}$ is non-zero then there is a fault in subsystem G_{i+1}^{ss} or sensor $k_{s(i+1)}$

Rule IV: If y_i^{ss} is non-zero and $y_{s(i+1)}$ is zero then there is a fault in sensor k_{si}

Note: These rules may be generalized to multiple faults.

4.2 Optimization Problem formulation

As in [1], the optimal sensor placement problem can be formulated as the following optimization problem, as shown in equation (10). Let q be a vector of binary elements that denotes which candidate sensors is installed or not. $q_j = 1$ means that sensor $s_j \in S$ is installed, whereas $q_j = 0$ means that s_j is not.

$$\min : J(q) = \sum_{j=1}^{m} w_j q_j \tag{10}$$

subject to

F_D *is detectable (Fault diagnosis used for detectibility of the fault)*

F_D *is isolable (Fault diagnosis used for the isolability of the fault)*

where m is the total number of candidate sensors and w_j is the cost of sensor s_j comprising purchase, maintenance, installation and reliability costs.

Problem (1) will be solved for two general cases:

- Case I: $F_D{}^I = F_P$
- Case II: $F_D{}^{II} = F_P \cup F_S$

In CASE I, the *Target Fault Set* is known *a priori*, before solving the optimization problem. In CASE II, this is not true, since F_{S^*} will be known *a posteriori*, after the optimization problem is solved. Considering these cases, the following constraint equations are being used:

$F_D{}^I : F_D{}^I = F_P$

$$F_D{}^I \textit{ is detectable} \leftrightarrow \sum_{r_i \in \hat{R}} \hat{M}_{ik}\, \rho_i \geq 1, \;\; \forall\; f_k \in F_P \tag{11}$$

Note: F_P *contains* f_1, f_2, f_3, f_4.

$F_D{}^{II} : F_D{}^{II} = F_P \cup F_S$

$$F_D{}^{II} \textit{ is detectabl} \leftrightarrow \sum_{r_i \in \hat{R}} \hat{M}_{ik}\, \rho_i \geq \begin{Bmatrix} 1 & if \;\; f_k \in F_P \\ q_k & if \;\; f_k \in F_S \end{Bmatrix} \;\; \forall\; f_k \in F \tag{12}$$

Note: F_P *contains* f_1, f_2, f_3, f_4 *and* F_S *contains* f_5, f_6, f_7, f_8.

$$F_D{}^{I} \text{ is ISOLABLE} \quad (13)$$

$$\leftrightarrow \sum_{r_i \in \hat{R}} \left| \hat{M}_{ik} - \hat{M}_{il} \right| \rho_i \geq 1, \ \forall \ f_k, f_l \in F_P, f_k \neq f_l$$

$$F_D{}^{II} \text{ is isolable} \quad (14)$$

$$\leftrightarrow \sum_{r_i \in \hat{R}} \left| \hat{M}_{ik} - \hat{M}_{il} \right| \rho_i \geq \begin{Bmatrix} 1 \quad if \ \ f_k, f_l \in F_P \\ q_k \ \ if \ \ f_l \in F_P \ and \ f_k \in F_S \\ q_l \ \ if \ \ f_k \in F_P \ and \ f_l \in F_S \\ q_k q_l \ \ if \ \ f_k, f_l \in F_S \end{Bmatrix} \ \forall \ f_k \in F$$

Let ρ_i be the binary ARR selector denoting whether ARR r_i is valid (ρ_i =1) or not (ρ_i = 0) and $\hat{M}_{ik}$ and $\hat{M}_{il}$ are the matrices generated from the $F_D{}^{I}$ and $F_D{}^{II}$.

4.3 Implementation of fuzzy rules on the coupled tank system to generate the ARR table

We will use a set of fuzzy logic rules to detect a leakage. The fuzzy IF and THEN rules for the two-tank fluid system are derived using the sensor network shown in Fig.3. For the fault diagnosis problem, the equivalent of Fig. 3, is shown in Fig. 4 whose various sub-systems and sensor blocks are all explained below. First, note that the first two blocks in Fig. 4, i.e. G_0 and $G_1 = G_1^0 \gamma_a$, represent the controller and the actuator sub-systems, respectively.

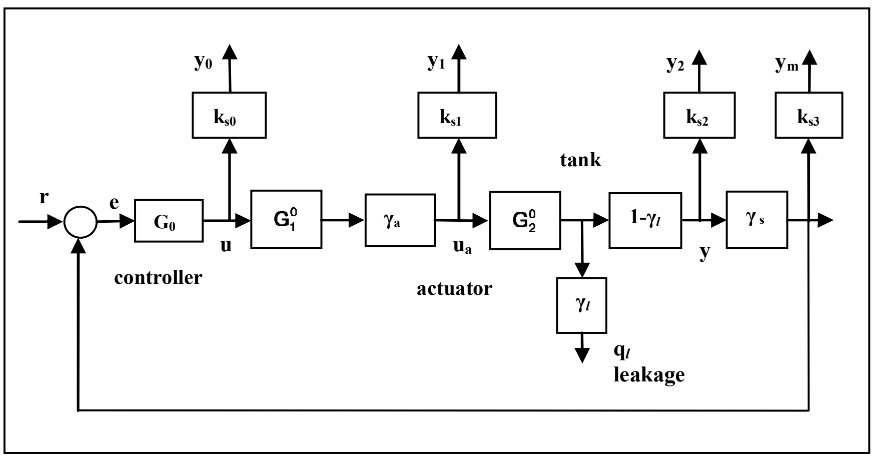

Fig. 4 Fluid system subject to a leakage

As shown in Fig. 4, the leakage is modeled by the gain γ_ℓ, which is used to quantify the amount of flow lost from the tank[1]. Thus the net outflow is quantified by the gain ($1-\gamma_\ell$). Since the two blocks G_2^0 and ($1-\gamma_\ell$) cannot be dissociated from each other, they are fused into a single block labeled $G_2 = G_2^0(1-\gamma_\ell)$. The feedback sensor, modeled by the gain k_{sf}, is used to feed the plant output y back to the controller, and is modeled by the last block G_3 in Fig. 3, where $G_3 = k_{sf}$. An additional sensor, termed as the redundant sensor of gain k_{s2}, is used here to discriminate between faults in the height sensor and feedback sensor. Even though the control input u does not necessitate a separate sensor to monitor its output, as it is freely available from the digital controller (G_0), a separate unit gain, labeled $k_{s0} = 1$, is attributed to it. Similarly, the last sensor, used to monitor the feedback sensor output, is also attributed a unit gain, i.e. $k_{s3} = 1$. The reason for adding these two unit gains to Fig. 4 is motivated by our desire to make the overall sensor network structure for the leakage detection problem fit in well within the general sensor network. By doing so, the two fuzzy rules (Rules 1 and 2 given earlier) can be readily applied to Fig. 4. The four residuals, r_0, r_1, r_2 and r_3, are the deviations between the fault-free and fault-bearing measurements of the control input , flow rate, height from the redundant sensor, and height from the feedback sensor, respectively.

[1] *Comments*: The physical two-tank fluid system is nonlinear, including dead-band nonlinearity and has fast dynamics. The identified model order is different from that of the model derived from the physical laws. This makes it difficult to employ the conventional parameter identification technique [6], as the function $\varphi(.)$ is difficult to obtain. Performing a number of offline experiments on the physical system, by varying the detection parameters, captures the influence of the detection parameters on the input-output behavior reliably.

Table 3 Example of an ARR table

	h_1	h_2	q_p	q_v	u_p	u_v
ARR_{s1}	0	1	0	1	1	1
ARR_{s2}	0	1	0	1	1	1
ARR_{s3}	0	1	0	1	0	1
ARR_{s4}	0	1	0	1	1	0
ARR_{s5}	0	0	0	1	1	1
ARR_{s6}	0	1	0	0	1	1

Table 4 Example of a fault signature matrix

	f_1	F_2	f_{h2}	f_{qv}	f_{up}	f_{uv}
ARR_{s1}	0	1	1	1	1	1
ARR_{s2}	1	0	1	1	1	1
ARR_{s3}	1	1	1	1	0	1
ARR_{s4}	1	1	0	1	1	1
ARR_{s5}	1	1	0	1	1	1
ARR_{s6}	1	1	1	0	1	1

Applying the exhaustive ARR generation algorithm described in [22] a full ARR table and a full fault signature matrix was created, a sample of which is shown in Table 3 and Table 4, where for example ARR_{s1} denotes the 1st sample generated.

4.4 Optimization results

Cost distribution table of six sensors, as per the fault signature matrix, was generated as follows (see Table 5):

Table 5 Cost distribution table

Variable	Description	Cost Distribution of the six sensors						
h_1	*Tank 1 level*	10		X	X		X	X
h_2	*Tank 2 level*	100			X	X		X
q_v	*Valve flow*	10	X			X	X	X
q_p	*Pump flow*	10	X	X	X	X	X	X
u_v	*Valve control input*	10	X	X	X	X		X
u_p	*Pump control input*	100	X	X			X	

Various techniques have been employed, which are a) binary non-linear programming technique b) non-linear programming with relaxation c) binary linear programming and d) linear programming with relaxation. The cost function is given as follows:

$$Min\ 10q_{hu}+100q_{hl}+10q_{qv}+10q_{qp}+10q_{uv}+100q_{up} \quad (15)$$

4.5 Constraints for binary non-linear programming:

A sample of non-linear constraints is shown as below (see equation 16-22):

Note : During constraint formulation, $q_{hu}=q_3$, $q_{hl}=q_4$, $q_{qv}=q_5$, $q_{qp}=q_6$, $q_{uv}=q_7$, $q_{up}=q_8$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7+q_4*q_5*q_8+q_5*q_7*q_8+q_4*q_7*q_8>=1 \quad (16)$$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7+q_4*q_5*q_8+q_5*q_7*q_8+q_4*q_7*q_7>=1 \quad (17)$$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7+q_4*q_5*q_8+q_5*q_7*q_8+q_4*q_7*q_8>=1 \quad (18)$$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7*q_8+q_4*q_5*q_7+q_4*q_5*q_8+q_4*q_7*q_8=q_4 \quad (19)$$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7*q_8+q_4*q_5*q_7+q_4*q_5*q_8+q_5*q_7*q_8>=q_5 \quad (20)$$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7*q_8+q_4*q_5*q_7+q_5*q_7*q_8+q_4*q_7*q_8>=q_7 \quad (21)$$

$$q_4*q_5*q_7*q_8+q_4*q_5*q_7*q_8+q_4*q_5*q_8+q_5*q_7*q_8+q_4*q_7*q_8>=q_8 \quad (22)$$

4.6 Constraints for linear programming with LP relaxation:

In the linearized version of programming, the constraints are linearized in the following manner, for the following non-linear constraints:

$$q_4*q_5*q_7*q_8 \quad (23)$$

The linearized version of equation (23) can be written as follows (see equation 24-28):

$$q_4+q_5+q_7+q_8<=x_{11}+1+1+1 \; ; \quad (24)$$

$$x_{11}<=q_4 \; ; \quad (25)$$

$$x_{11}<=q_5 \; ; \quad (26)$$

$$x_{11}<=q_7 \quad (27)$$

$$x_{11}<=q_8 \; ; \quad (28)$$

Note: The remaining equations and inequalities can be seen in the Appendix A.

A sample of detailed results for linear programming with relaxation technique is as follows (see Table 6):

Table 6 Sample of detailed results for linear programming with relaxation

Variables	LB	Relaxed	UB	Marginal
---- VAR q3	.	.	+INF	10.000
---- VAR q4	.	0.707	+INF	.
---- VAR q5	.	1.000	+INF	.
---- VAR q6	.	.	+INF	10.000
---- VAR q7	.	1.000	+INF	.
---- VAR q8	.	0.707	+INF	.
---- VAR x1	.	0.707	+INF	.
---- VAR x2	.	0.707	+INF	.
---- VAR x3	.	1.000	+INF	.
---- VAR x4	.	0.414	+INF	.
---- VAR x5	.	0.707	+INF	.
---- VAR x6	.	0.707	+INF	.
---- VAR x7	.	0.707	+INF	.
---- VAR x8	.	0.414	+INF	.
---- VAR x9	.	0.414	+INF	.
---- VAR x10	.	0.707	+INF	.
---- VAR x11	.	0.414	+INF	.
---- VAR f	-INF	161.421	+INF	.

Moreover, it is shown that out of six sensors, i.e. $\{h_1, h_2, q_v, q_p, u_v, u_p\}$, the optimal sensor placement is of four sensors, which are h_2, q_v, u_v and u_p. Thus, the optimal sensor configuration for the 6 sensors being experimented is as follows:

$$S^* = \{h_2, q_v, u_v, u_p\} \quad (29)$$

The objective value and the computational time results for the four techniques employed are as follows (Result # 1- 4):

Result # 1:

BINARY NON-LINEAR PROGRAMMING
***** OBJECTIVE VALUE = 220.00*
GENERATION TIME = 0.031 SECONDS

Result # 2:

NON-LINEAR PROG. WITH RELAXATION
***** OBJECTIVE VALUE = 161.4214*
GENERATION TIME = 0.046 SECONDS

Result # 3:

BINARY LINEAR PROGRAMMING
***** OBJECTIVE VALUE = 220.00*
GENERATION TIME = 0.035 SECONDS

Result # 4:

LINEAR PROG. WITH RELAXATION
***** OBJECTIVE VALUE = 161.4214*
GENERATION TIME = 0.094 SECONDS

It can be seen from the results that the objective value of binary non-linear programming and linear programming are the same, but the linear programming is taking more computational time, which is required while linearization. However, with LP relaxation, the linear programming with relaxation took a slightly larger computational time. But, the objective value results for linear programming with relaxation are better than non-linear programming. One can also notice that the relaxed version has more optimal objective value cost, as the sensors are not bound to straddle only between 0 and 1.

5 Conclusions

The sensor location problem has been addressed in this paper. Considering [1], the detectability and isolability performance are considered for optimal sensor placement. It allows for the determination of the set of sensors that minimizes a pre-defined cost function, satisfying at the same time a pre-established set of FDI specifications, for a given set of faults. Sets of all possible ARRs have been generated through a set of fuzzy rules, considering all possible candidate sensors installed. The optimization techniques of linear and nonlinear programming have been applied, which shows an improved cost function, accompanied by a reduction in computational time.

Nevertheless, there are still some open issues that could be considered for further research. Firstly, the causality constraints involved in the structural modeling of dynamic equations are not taken into account. Secondly, faults that change the structure of the model are not considered either, only additive faults on measurable variables have been dealt with here. The variable values with the relaxation technique show a particular behavior that, if analyzed as per the behavior of the system, warrants further study. Fault detectability and isolability constraints have been formulated in this paper, but other specifications, such as fault identifiability, fault sensitivity, etc., could be easily included in the optimal sensor placement problem.

Acknowledgements Authors wish to acknowledge the valuable suggestions and useful comments of Editor-in-Chief and two anonymous reviewers. Their suggestions and comments have improved the quality of the paper immensely. Authors also acknowledge the financial support by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the King Fahd University of Petroleum and Minerals, in carrying out this work. Authors would also like to acknowledge the editorial and proof-reading assistances of Kim Wilson of University of New Brunswick, Canada.

References

1. Sarrate R, Puig V, Escobet T, Rosich A (2007) Optimal sensor placement for model-based fault detection and isolation. *Proceedings of the.* 46th *IEEE Conference on Decision and Control*, pages 2584–2589, New Orleans, USA, December 12–14.
2. Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2003) Diagnosis and fault-tolerant control. Springer.
3. Rahim M A, Khalid, H M, Akram M, Khoukhi A, Cheded L, Doraiswami R (2011) Quality monitoring of a closed-loop system with parametric uncertainties and external disturbances: a fault detection and isolation approach. International Journal of Advance Manufacturing and Technology. 55(1-4): 293--306.
4. Rahim M A, Khalid H M, Khoukhi A (2012) Nonlinear constrained optimal control problem: a PSO-GA-based discrete augmented Lagrangian approach. International Journal of Advance Manufacturing and Technology. Article in Press.
5. Bagajewicz M (2000) Design and upgrade of process plant instrumentation. Lancaster, PA: Technomic Publishers.
6. Spanache S, Escobet T, Trav´e-Massuy`es L (2004) Sensor placement optimization using genetic algorithms. Proceedings of the 15th International Workshop on Principles of Diagnosis (DX'04), Carcassonne, France, June 23–25.
7. Nejjari F, P´erez R, Escobet T, Trav´e-Massuy`es L (2006) Fault diagnosability utilizing quasi-static and structural modelling. Math Comput Mod 45:606–616.
8. Staroswiecki M, Hoblos G, Aitouche A (2004) Sensor network design for fault tolerant estimation. Int J Adapt Control Signal Process 18(1):55–72.
9. Maquin D, Luong M, Ragot J (1997) Fault detection and isolation and sensor network design. Europ J Autom 31(13):396–406.
10. Raghuraj R, Bhushan M, Rengaswamy R (1999) Locating sensors in complex chemical plants based on fault diagnostic observability criteria. AIChE J 45(2):310–322.
11. Commault C, Dion J-M (2007) Sensor location for diagnosis in linear systems: A structural analysis. IEEE Trans Automat Contr 52(2):155–169.
12. Frisk E, Krysander M (2007) Sensor placement for maximum fault isolability. Proceedings of the 18th International Workshop on Principles of Diagnosis (DX'07), Nashville,TN,USA, May 29–31.
13. Gertler J (1998) Fault detection and diagnosis in engineering systems. New York: Marcel Dekker.
14. Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1996) Failure diagnosis using discrete-event models. *IEEE Trans Control Syst Technol* 4(2):105–124.
15. Lunze J, Schroeder J (2004) Sensor and actuator fault diagnosis of systems with discrete inputs and outputs. *IEEE Trans Syst, Man, Cybern. B, Cybern* 34(2):1096–1107.
16. Narasimhan S, Biswas G (2007) Model-based diagnosis of hybrid systems. *IEEE Trans Syst, Man, Cybern, A, Syst, Humans* 37(3):348–361.
17. Staroswiecki M, Comtet-Varga G (2001) Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5):687–699.
18. Nikoukhah R (1998) A new methodology for observer design and implementation. *IEEE Trans Automat Contr* 43 (2):229–234.
19. Maquin D, Luong M, Ragot J (1997) Fault detection and isolation and sensor network design. *Europ J Autom* 31(13):396–406.
20. Raghuraj R, Bhushan M, Rengaswamy R (1999) Locating sensors in complex chemical plants based on fault diagnostic observability criteria. *AIChE J* 45(2):310–322.
21. Krysander M, Frisk E (2008) Sensor placement for fault diagnosis. *IEEE Trans Syst, Man, Cybern, A* 38(6):1398–1410.
22. Commault C, Dion J-M, Agha SY (2008) Structural analysis for the sensor location problem in fault detection and isolation. *Automatica* 44(8):2074–2080.
23. Yassine A A, Ploix S, Flaus J M. A method for sensor placement taking into account diagnosability criteria. *Int J Appl Math Comput Sci* 18(4):497–512.
24. Trav´e-Massuy`es L, Escobet T, Olive X (2006) Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Trans Syst, Man, Cybern, A*, 36(6):1146–1160.

25. Krysander M (2006) Design and analysis of diagnosis systems using structural methods. Ph.D. dissertation, Linköpings Universitet, Linköping, Sweden.

APPENDIX A

(a) Non-linear constraints

```
q4*q5*q7*q8+q4*q5*q7+q4*q5*q8+q5*q7*q8+q4*q7*q8>=1 ;
q4*q5*q7*q8+q4*q5*q7+q4*q5*q8+q5*q7*q8+q4*q7*q7>=1 ;

q4*q5*q7*q8+q4*q5*q7+q4*q5*q8+q5*q7*q8+q4*q7*q8>=1 ;
q4*q5*q7*q8+q4*q5*q7+q4*q5*q8+q5*q7*q8+q4*q7*q8>=1 ;
q4*q5*q7*q8+ q4*q5*q7*q8+q4*q5*q7+q4*q5*q8+q4*q7*q8>=q4 ;
q4*q5*q7*q8+ q4*q5*q7*q8+q4*q5*q7+q4*q5*q8+q5*q7*q8>=q5 ;
q4*q5*q7*q8+ q4*q5*q7*q8+q4*q5*q7+q5*q7*q8+q4*q7*q8>=q7 ;
q4*q5*q7*q8+ q4*q5*q7*q8+q4*q5*q8+q5*q7*q8+q4*q7*q8>=q8 ;

q4*q5*q7*q8+q5*q7*q8>=q4 ;
q4*q5*q7*q8+q5*q7*q8>=q4 ;
q4*q5*q7*q8+q4*q7*q8>=q5 ;
q4*q5*q7*q8+q4*q7*q8>=q5 ;
q4*q5*q7*q8+q4*q5*q8>=q7 ;
q4*q5*q7*q8+q4*q5*q8>=q7 ;
q4*q5*q7*q8+q4*q5*q7>=q8 ;
q4*q5*q7*q8+q4*q5*q7>=q8 ;

q4*q5*q7*q8+q5*q7*q8>=q4 ;
q4*q5*q7*q8+q5*q7*q8>=q4 ;
q4*q5*q7*q8+q4*q7*q8>=q5 ;
q4*q5*q7*q8+q4*q7*q8>=q5 ;
q4*q5*q7*q8+q4*q5*q8>=q7 ;
q4*q5*q7*q8+q4*q5*q8>=q7 ;
q4*q5*q7*q8+q4*q5*q7>=q8 ;
q4*q5*q7*q8+q4*q5*q7>=q8 ;

q4*q5*q7+q5*q7*q8>=q4*q8 ;
q4*q5*q7+q4*q5*q8>=q7*q8 ;
q4*q5*q7+q4*q7*q8>=q5*q8 ;
q5*q7*q8+q4*q5*q7>=q4*q8 ;
q4*q5*q8+q4*q7*q8>=q5*q7 ;
q4*q5*q8+q5*q7*q8>=q4*q7 ;
q4*q7*q8+q5*q7*q8>=q4*q5;
```

(b) Linear constraints

```
q4+q5<=x1+1 ;
x1<=q4 ;
x1<=q5 ;
q4+q7<=x2+1 ;
x2<=q4 ;
```

```
x2<=q7 ;
q5+q7<=x3+1 ;
x3<=q5 ;
x3<=q7 ;
q4+q8<=x4+1 ;
x4<=q4 ;
x4<=q8 ;
q5+q8<=x5+1 ;
x5<=q5 ;
x5<=q8 ;
q7+q8<=x6+1 ;
x6<=q7 ;
x6<=q8 ;
q4+q5+q7<=x7+1+1 ;
x7<=q4 ;
x7<=q5 ;
x7<=q7 ;
q4+q5+q8<=x8+1+1 ;
x8<=q4 ;
x8<=q5 ;
x8<=q8 ;
q4+q7+q8<=x9+1+1 ;
x9<=q4 ;
x9<=q7 ;
x9<=q8 ;
q5+q7+q8<=x10+1+1 ;
x10<=q5 ;
x10<=q7 ;
x10<=q8 ;
q4+q5+q7+q8<=x11+1+1+1 ;
x11<=q4 ;
x11<=q5 ;
x11<=q7 ;
x11<=q8 ;

x11+x7+x8+x10+x9>=1 ;
x11+x11+x7+x8+x9>=q4 ;
x11+x11+x7+x8+x10>=q5 ;
x11+x11+x7+x10+x9>=q7 ;
x11+x11+x8+x10+x9>=q8 ;
x11+x10>=q4 ;
x11+x9 >=q5 ;
x11+x8 >=q7 ;
x11+x7>=q8 ;
x11+x10>=q4 ;
x11+x9>=q5 ;
x11+x8>=q7 ;
x11+x7>=q8 ;
x7+x10>=x4 ;

x7+x8>=x6 ;
x7+x9 >=x5 ;
x10+x7>=x4 ;
x8+x9>=x3 ;
x8+x10>=x2 ;
x9+x10>=x1
```