Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

A New Fault Diagnosis Algorithm that Improves the Integration of Fault Detection and Isolation

Vicenç Puig, Frank Schmid, Joseba Quevedo, Belarmino Pulido

Abstract— **This work proposes a new model-based fault diagnosis method that improves the integration of the fault detection and isolation tasks. A new interface between fault detection and fault isolation is presented that contains information about the degree of fault signal activation and the occurrence time of fault signals. A combination of five fault signature matrices is used for the fault isolation process. The matrices store knowledge about faulty system behavior: boolean fault signal occurrence, signs of residual violation, sensitivities, time of fault signal activation and fault signal occurrence order. Finally, the new method is applied to the well-known two-tanks benchmark problem.**

I. INTRODUCTION

In model-based fault diagnosis, fault detection and fault isolation tasks are considered separately. The typical interface between these two modules is through binary codifying the test of violation of each residual and generating a fault signature. Then, fault isolation consists on matching the actual fault signature with some of the theoretical fault signatures. It is well known [1] that the performance of the whole fault diagnosis system can be highly augmented by improving such interface between fault detection and isolation modules. In particular, such interface can be improved taking into account the following information:

- residual signs - faults can cause positive or negative residual values.
- the size of the residual value - big violation of the threshold or only a small fault signal-activation.
- the sensitivity of a residual expression with respect to a certain fault.
- the time pattern of fault signal occurrence.
- the order of fault signal occurrence.

The goal of this paper is to present a new algorithm that improves such integration by taking the best of existing approaches, and considering the previous information. Considering the time pattern and the order of fault signal occurrence, generalized assumptions of existing methods about all the residuals sensitive to a given fault should be activated can be removed. This is specially important in dynamic systems where residuals can present different activation times after the fault occurrence that can lead to the fact that not all the residuals are activated simultaneously. Moreover,because of the different sensitivity of each individual residual to a fault some residual may not be activated in practice. Taking into account the size of the residual and its sensitivity with respect to a certain fault such problem can also be solved. Finally, the sign of the residual will increase the fault isolability between faults that without considering sign will not be distinguishable.

The structure of the remainder of the paper is the following: in Section II, assumptions associated to the existent methods in model-based fault isolation methods are presented. In Section III, a new algorithm that improves the integration of fault detection and isolation modules is presented. Finally, in Section IV the proposed algorithm is compared with existing methods in order to compare his performance using the well-known two-tanks benchmark problem.

This work is supported by CICYT (DPI2002-03500, DPI2002-01809) and by Research Commission of the "Generalitat de Catalunya" (group SAC ref.2001/SGR/00236)

V. Puig, F. Schmid and J.Quevedo are with Automatic Control Department, Campus de Terrassa, Technical University of Catalonia, Rambla Sant Nebridi, 10, 08222 Terrassa, Spain, e-mail: vicenc.puig@upc.edu

B. Pulido is with Departamento de Informática, Universidad de Valladolid. 47011 Valladolid, Spain

II. MODEL-BASED FAULT DETECTION

Model-based fault detection tests are based on the evaluation of a set of fault indication signals $r_i(k)$, obtained through analytical redundancy relations (ARRs) or residuals generated by comparing measurement of physical variables $y_i(k)$ of the process with their estimation $\hat{y}_i(k)$ using a model:

$$r_i(k) = y_i(k) - \hat{y}_i(k) \tag{1}$$

A fault detection task decides if a ARR is violated at a given instant or not generating a *fault signal* s_i according to:

$$s_i = \begin{cases} 0, & \text{if } |r_i(k)| < \tau_i \text{ (no fault)} \\ 1, & \text{if } |r_i(k)| \geq \tau_i \text{ (fault)} \end{cases} \tag{2}$$

where τ_i is the threshold associated to the ARR $r_i(k)$.

The *actual fault signature* of the system $\mathbf{s}(k) = [s_1(k), s_2(k), \ldots s_n(k)]$ is provided to the fault isolation module which will try to isolate the fault and give a diagnosis based on $\mathbf{s}(k)$.

Given a set of ARRs and a set of considered faults $f_1, f_2, \ldots f_m$, a *theoretical fault signature matrix* FSM can be defined by binary codifying the influence of a fault on every residual.

This matrix has as many rows as residuals and as many columns as considered faults. An element FSM_{ij} of this matrix being equal to 1 means that the j^{th} fault appears in the expression of the i^{th} residual. Otherwise it is equal to 0. Assuming classical FDI fault hypotheses, i.e, single faults and no-compensation (exoneration) [2], fault isolation will consist in looking for a column of the fault signature matrix Σ that matches the actual fault signature $\mathbf{s}(k)$. Therefore,

0-7803-9568-9/05/$20.00 ©2005 IEEE

TABLE I

INFLUENCE ON THE CREDIBILITY OF A FAULT HYPOTHESIS FOR EACH OF THE FOUR CASES

	Column Reas.	Row Reas.	DMP Meth.	New Meth.
(1) expected fault signal	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$
(2) missing fault signal	$\searrow$	$-$	$-$	$\searrow$
(3) unexpected fault signal	$\searrow$	$\searrow$	$-$	$\searrow$
(4) expected absence fault signal	$\nearrow$	$-$	$-$	$-$

this classic approach in the FDI-community is also known as Column Reasoning [3]. This simplified purely boolean view works well if the following premises are satisfied:

Assumption 1: All theoretical fault signals associated to a fault must be activated.

Assumption 2: All those fault signals must be activated *at the same time* and persist during the whole fault isolation process.

Assumption 1 fails essentially in case of small-size faults that are not strong enough to activate all theoretical fault signals. Missing fault signals are a problem for Column Reasoning. Row Reasoning - the classic approach in the DX-community [2] - tries to avoid that by reasoning only with the activated fault signals and the corresponding rows of FSM.

Assumption 2 cannot hold in case of dynamic systems with transient residuals that lead to different fault signals occurence times. This has already been stated by many other authors (e.g. [1]) and some efforts have been made to improve fault isolation for dynamic systems.

Some approaches focus on the problems connected to Assumption 1 (see Section II-A), others try to solve those related to Assumption 2 (see Section II-B). The present paper tries to improve the efficiency of the fault isolation process by combining both aspects in one approach.

A. Reasoning with static fault signals

When dealing with the problems connected to Assumption 1, four main questions arise:

Case 1 - expected fault signal: what happens, when a theoretical 'one' meets an observed 'one'?

Case 2 - missing fault signal: what happens when a theoretical 'one' meets an observed 'zero'?

Case 3 - unexpected fault signal: what happens when a theoretical 'zero' meets an observed 'one'?

Case 4 - expected absence of fault signal: what happens when a theoretical 'zero' meets an observed 'zero'?

Every case should have a different influence on the credibility of a fault hypothesis. That influence differs from method to method as illustrated in Table I. '$\nearrow$' means, that this case supports the fault hypothesis, '$\searrow$' that not, and '$-$' means, that this case is not considered by that method. In Section IV, Column Reasoning[3], Row Reasoning[2], the DMP-algorithm [4] and the new method proposed in Section III are compared.

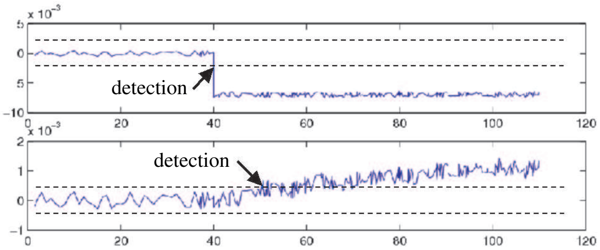

Fig. 1. Transient residuals with delayed detection

B. Reasoning with dynamic fault signals

Fault signals with different occurrence times represent a problem for many fault isolation methods. Often, the diagnostic decision at time k only depends on the observed fault signals $\mathbf{s}(k)$ at time k. The past values of $\mathbf{s}$ are not regarded. That can cause erroneous diagnostic results.

Figure 1 shows a fault that affects two residuals in a different way. Due to the dynamics of the system, the residual values pass the thresholds at different times k, and therefore, the fault signals are detected at different times. In order to provide consistent and reliable diagnosis results, a fault isolation method should perform an incremental way of reasoning considering those possible time-delays, as for example proposed in [5] Dynamic-Table-of-States-method (DTS), or in [6], [7].

III. ARCHITECTURE OF THE NEW METHOD FOR FAULT ISOLATION

A. Introduction to the new method

The method presented here tries to include mechanisms to deal with both problems described in Section II-A and II-B. This leads to an architecture with three different components as shown in Figure 2. The first component is the interface between the fault detection and fault isolation modules. It is based on a memory that stores information about the fault signal occurrence history and is updated cyclically by the fault detection module.

The pattern comparison component compares the memory to the stored fault patterns. This process includes different tasks.

The last component represents the decision logic part of the method. Its goal is to exclude fault hypotheses (if inconsistent with the memory) and to determine the most probable from the lasting ones.

B. The Memory Component

The memory consists of a table in which events in the residual history are stored. For each row, the first column stores the *occurrence time* t_i, the second one stores the *maximum activation value* $a_{i,max}$, and the third one stores the *sign* of the residual. If the fault detection component detects a new fault signal, it updates the memory by filling out the three fields. In general, any detection algorithm can be used for that purpose.

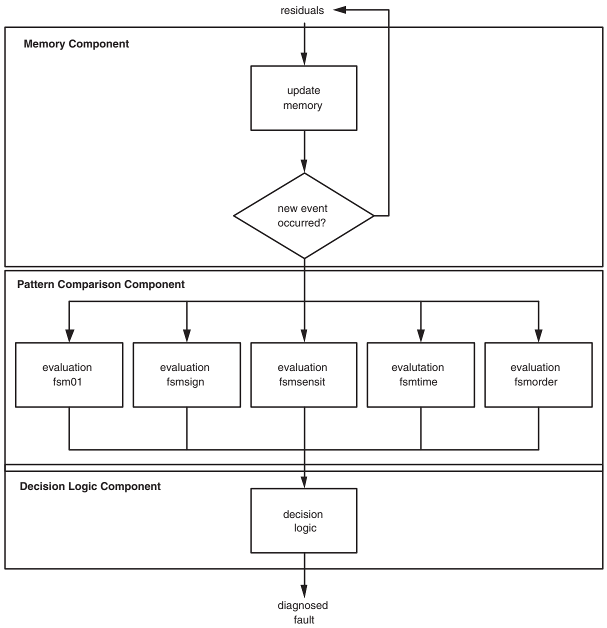

Fig. 2. Block diagram of the new method

The activation value $a_i(k)$ for every fault signal is calculated as in the DMP-approach [4] using the function of Kramer:

$$a_i(k) = sgn(r_i(k)/\tau_i)\,\frac{(r_i(k)/\tau_i)^4}{1+(r_i(k)/\tau_i)^4} \tag{3}$$

In this way, residuals are normalised to a metric between -1 and 1 which indicates the degree to which each equation is satisfied: 0 for perfectly satisfied, 1 for severely violated high and -1 for severely violated low. Moreover, the Kramer function avoids decision instability (chattering) due to noise that the boolean test in equation (2) presents, thanks to introduced grading.

The problem of different time instants of fault signal appearance is solved not allowing an isolation decision until a prefixed waiting time (T_w) has elapsed, from the first fault signal appearance. This (T_w) is calculated from the larger transient time response (T_{lt}) from non-faulty situation to any faulty situation. After this time has been elapsed, a diagnosis is proposed and the memory component is reset being ready to start the diagnosis of a new fault. Following [1], inside this diagnosis window, the maximum activation value of the memory-table $a_{i,max}$ is updated at the time k_0 only if the actual activation $a_i(k_0)$ is superior to the previous ones:

$$a_{i,max} = \max_{\forall k<k_0 \,\wedge\, |a_i(k)|>0.5}(|a_i(k)|) \tag{4}$$

Due to the max-operator the activation values only can rise and not fall again. Fault signals with $|a_{i,max}| < 0.5$ are filtered out. That implies a double advantage:
1.) The effect of noise is partially filtered out. That leads to smoother diagnosis results without flickering.
2.) The persistence of fault signals does not matter since just the peaks of activation are stored.

The memory table makes the residual history accessible for later computation by explicitly storing that data. In this way, time aspects of fault isolation can be treated in a very easy and straight-forward way.

C. The Pattern Comparison Component

The pattern comparison component compares the memory to the stored fault patterns. This process includes five different evaluation tasks, each of them based on a different aspect of the fault patterns: boolean fault signal activation, fault signal signs, residual sensitivity to faults, fault signal occurrence time and fault signal occurrence order. Each evaluation task uses a separate fault signature matrix containing the necessary information to perform the task. A summary of the meaning for each evaluation factor is provided in Table II.

	Meaning
factor01	counts the number of fault signals for each fault, excludes faults with unexpected fault signals
factor$\pm$	excludes faults if signs are in conflict with the theoretical signs
factor$sensit$	gives a measure for the fault probability, based on sensitivity
factor$time$	gives a measure for the fault probability, based on time-patterns
factor$order$	excludes faults if order does not correspond to theoretical values

TABLE II
THE MEANING OF ALL FACTORS - AN OVERVIEW

1) ***FSM***01*: Evaluation of Fault Signal Appearance:* This evaluation component works with a simple, boolean fault signature matrix ***FSM***01 (see Table III). Assuming that there are n residuals, ***factor***01_j is calculated for the j^{th} fault hypothesis in the following way:

$$\boldsymbol{factor}01_j = \frac{\sum_{i=1}^{n}(boolean(a_{i,max}) * \boldsymbol{FSM}01_{ij})}{\sum_{i=1}^{n}\boldsymbol{FSM}01_{ij}} * zvf_j \tag{5}$$

with

$$boolean(a_{i,max}) = \begin{cases} 0, & \text{if } a_{i,max}=0 \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

and the zero-violation-factor as

$$zvf_j = \begin{cases} 0, & \text{if } \exists\, i \in \{1,\dots n\} \text{ with } \boldsymbol{FSM}01_{ij}=0 \\ & \text{and } a_{i,max} \neq 0 \\ 1, & \text{otherwise} \end{cases} \tag{7}$$

That leads to the following behavior regarding the four cases in Section II-A: Expected fault signals support a hypothesis, unexpected fault signals eliminate it due to equation (7). Missing fault signals influence the supportability of a hypothesis indirectly via the denominator in equation (5). Case 4 is not considered.

*2) **FSM**$sign$: Evaluation of Fault Signal Signs:* The ***FSM***$sign$-table contains the theoretical sign-patterns that faults produce in the residual equations. Those patterns can be codified using the values 0 for no influence, $+1/-1$ for positive/negative deviation for every $\boldsymbol{FSM}sign_{ij}$.

The ***factor***$sign_j$ is calculated comparing theoretical signs to the signs stored in the memory. This comparison is done only for the subset of activated fault signals ($a_{i,max} \geq 0.5$):

$$\boldsymbol{factor}sign_j = \begin{cases} 1, & \text{if } sign(a_i) = \boldsymbol{FSM}sign_{ij} \\ & \quad and\ a_{i,\max} \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

*3) **FSM**$sensit$: Evaluation of Fault Signal Activation Values:* This evaluation component uses the fault signal activation values $a_{i,max}$ from the memory table and computes ***factor***$sensit$ using the sensitivity-based ***FSM***$sensit$-table for weighting the activation values.

That approach can be found as well in the DMP-method [4]. The following equations describe how to calculate the entries $\boldsymbol{FSM}sensit_{ji}$:

$$\boldsymbol{FSM}sensit_{ji} = S_{ji}\,\frac{1}{\mid \tau_i \mid} \tag{9}$$

with the sensitivity

$$S_{ji} = \frac{\partial r_i}{\partial f_j} \tag{10}$$

Since sensitivity depends on time in case of a dynamic system, here the steady-state value after a fault occurrence is considered as also suggested in [3]. The value of $\boldsymbol{FSM}sensit_{ji}$ describes, how easily a fault will cause a violation of the threshold of the i^{th} residual since the larger its partial derivative with respect to fault, the more sensitive that equation is to deviations of the assumption. Similarly, residuals with large detection thresholds are less sensitive as they are more difficult to violate. Therefore $\boldsymbol{FSM}sensit_{ji}$ can be used to weight the activation value of different fault signals:

$$\boldsymbol{factor}sensit_j = \frac{\sum_{i=1}^{n}(a_{i,max} * \boldsymbol{FSM}sensit_{ij})}{\sum_{i=1}^{n} \mid \boldsymbol{FSM}sensit_{ij} \mid} \tag{11}$$

*4) **FSM**$time$: Evaluation of Fault Signal Occurrence Time:* This part of the pattern comparison component compares the fault signal occurrence times from the memory with the data of the ***FSM***$time$-table. This table contains a fuzzy time interval for every fault-residual-combination. Entries in the table have the following format:

$$\boldsymbol{FSM}time_{ij} = [\alpha, \beta, \gamma, \delta] \tag{12}$$

Those values are interpreted as the description of a fuzzy membership function, as can be seen in Figure 3. Entries with $\mathbf{FSM}01_{ji} = 0$ are codified as $[-1, -1, -1, -1]$.

All time intervals refer to the apparition time of the first activated residual. The new method allows to force a residual to be the first one expected: the case is represented as a pseudo interval $[0, 0, 0, 0]$. Similarly, a symptom can be forced not to be the first one, using an interval $[\alpha, \beta, \gamma, \delta]$ with $\delta \geq \gamma \geq \beta \geq \alpha > 0$.

This means codifying an absolute order of fault signal occurrence in ***FSM***$time$ (in contrast to the relative order in ***FSM***$order$). A similar approach can be found in [7] and [6]. The fuzzy way of considering time in this method facilitates the inclusion of uncertainties that often exist about time intervals.

The fault signature matrix ***FSM***$time$ allows to use Column Reasoning for a exactly specified subset of fault signals, while Row Reasoning is used in all other cases.

The computation of ***factor***$time$ is given in equation (13):

$$\boldsymbol{factor}time_j = \min_{a_{i,max} \neq 0}(\mu(t_i, t_{ref}, \boldsymbol{FSM}time_{ij})) \tag{13}$$

where $\mu(t_i, t_{ref}, \boldsymbol{FSM}time_{ij})$ is a function that returns the truth value of the fuzzy distribution $\boldsymbol{FSM}time_{ij}$ for $t = t_i - t_{ref}$, the time span between the occurrence time of the i^{th} fault signal and the reference time (occurrence of the first fault signal).

Making use of ***factor***$time$, the new fault isolation method is able to cope with dynamically developing fault signal patterns. Additionally, the evaluation of occurrence times improves the diagnostic resolution and speeds up the fault isolation process if the feature of forcing one fault signal or a set of fault signals to be the first one is used.

*5) **FSM**$order$: Evaluation of Fault Signal Occurrence Order:* The output of this evaluation component is only '1', if all fault signals appear in the right order. Otherwise it is '0'.

The order of fault signals is codified using ordinal numbers, starting with '1'. If two fault signals appear at the same time or if they explicitly may commute their order, then they should share the same ordinal number. Fault signals that must not appear get the code '0'.

Special attention has to be paid to the fact that Row Reasoning is implemented as well for the order of fault signal occurrence. That means that the occurrence order of fault signals is a relative and not a absolute one. Therefore, the codes '1' and '3' define a relative position ('3' after '1') with an eventual but not necessary fault signal '2' in between.

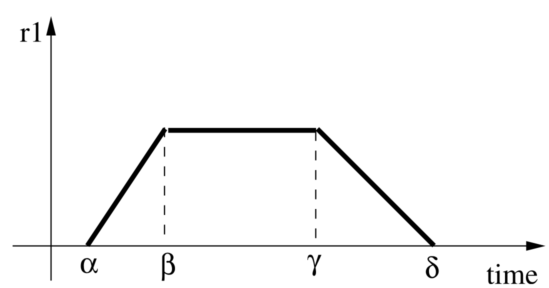

Fig. 3. Characterizing symptom r_1 apparition times through a fuzzy time interval. r_1 must not be the first symptom to appear.

D. The Decision Logic Component

The decision logic component's task is to isolate the fault. It bases its reasoning on all of the previously explained evaluation factors.

Decision logic can be divided into two steps:
1.) *Exclusion*: Using the results of the pattern comparison component, a big part of the faults can be excluded from the set of possible faults. That is the case, if any of the factors from the pattern comparison component is 'zero'. Every factor represents some kind of filter, that only lets slip through the possible fault hypotheses.
2.) *Confidence calculation and support*: The minimum operator is applied to ***factor***$sensit$ and ***factor***$time$ for every remaining fault hypothesis:

$$p_j = \min(\textbf{\textit{factor}}sensit_j, \textbf{\textit{factor}}time_j) \tag{14}$$

The result gives a measure for the confidence of this fault hypothesis.

IV. Comparison of the Proposed Method to Existing Ones on a Benchmark Problem

A. Definition of Performance Criteria

The diagnosis DGN of a fault isolation system is an expression of the form

$$DGN = \{(fault\ 1, p_1), ...(fault\ m, p_m)\} \tag{15}$$

where p_j are the confidence values assigned by the isolation method to each of m faults. In optimal case, DGN only contains one single element with $p_j \neq 0$: the correct diagnosis.

Looking at the content of DGN two performance criteria can be deduced:

Definition 1: *'Diagnostic resolution'*
The diagnostic resolution can be defined as

$$diagnostic\,resolution = \frac{1}{L}\sum_{j=1}^{L}\sum_{i=1}^{m} p_{ji} \tag{16}$$

where L is the number of executed performance tests, m is the number of faults considered and p_{ji} the confidence associated to i^{th} fault in the j^{th} performance test as defined in Eq. (14).

The diagnostic resolution represents the average number of valid fault hypotheses per diagnosis, and the best possible value is 1.

Definition 2: *'Error rate'*
The error rate of a diagnosis system is defined as the average percentage of wrong diagnoses. A diagnosis is supposed to be wrong, if $p_j = 0$ is assigned to the correct fault hypothesis.

An error rate of 0 is desirable. Thus, the optimal point in the error rate/diagnostic resolution-plane is (0/1).

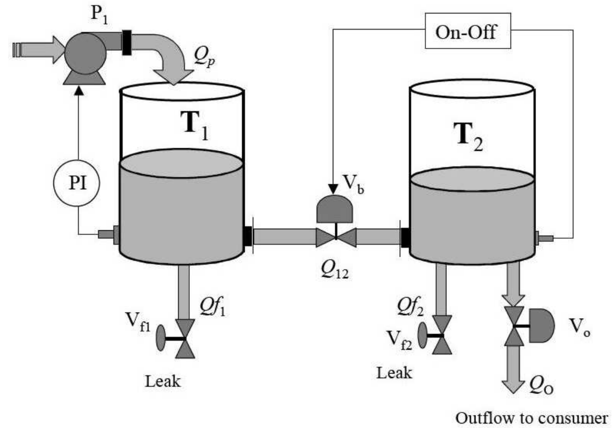

Fig. 4. A diagram of the two-tank model

B. The benchmark problem

In order to evaluate the performance of the new method, it is applied to the two-tank benchmark-problem (see Figure 4) and compared to Column Reasoning [3], Row Reasoning [2], DMP [4] and DTS [5]. This benchmark consists of two coupled tanks that provide a continuous water flow Q_0 to a consumer. The water levels in both tanks (y_1 and y_2) are measured and used for feedback control. In tank T_1, a PI-controller determines the inflow of water via pump P_1 (control signal U_p). The water level in Tank T_2 can be controlled using the valve V_b between both tanks (control signal U_b). Provided a higher water level in Tank T_1, water flows into Tank T_2 if the valve is open ($U_b = 1$). If the water level in Tank T_2 is kept near to the set point, a constant water flow Q_0 leaves the two-tank-system. Under normal conditions, the valve V_0 is always open ($U_0 = 1$).

The following possible fault scenarios are regarded:

- **Fault 1:** additive fault in pump P_1 (actuator fault).
- **Fault 2:** additive fault in level sensor y_1 (sensor fault).
- **Fault 3:** additive fault in level sensor y_2 (sensor fault).
- **Fault 4:** constant leak in tank T_1 (system fault).
- **Fault 5:** constant leak in tank T_2 (system fault).
- **Fault 6:** additive fault in sensor of controller output U_p in tank T_1 (sensor fault).

The boolean fault signature matrix of the problem is given in Table III. All the FSM-Tables are not included because of the lack of space but can be found in [8]. For a detailed description of the benchmark problem see [9] or [10].

TABLE III
The fault signature matrix for the benchmark system.

	f_1	f_2	f_3	f_4	f_5	f_6
r_1	0	1	1	1	0	0
r_2	0	1	1	0	1	0
r_3	0	0	0	0	0	1
r_4	1	0	0	0	0	1

C. The results

For the comparison, the six fault scenarios are tested twice: first with big faults and then with small faults. Using the

TABLE IV
THE PERFORMANCE OF DIFFERENT FDI METHODS ON THE TWO-TANK EXAMPLE

	Diagnosis resolution	Error rate
Column Reasoning	1.167	0.167
Row Reasoning	2.333	0.000
DMP method	2.182	0.000
DTS	0.917	0.417
new method	1.323	0.000

performance criteria from Section IV-A, the results shown in Table IV can be obtained. Figure 5 illustrates the advantages of the new method: It is reliable (error rate = 0) and has a diagnostic resolution close to one.

1) Diagnostic resolution: The diagnostic resolution differs from method to method due to a different influence of the four cases from Section II-A as shown in Table I. DTS provides the best diagnosis resolution, but the error rate is bigger than in the new method. Column Reasoning provides the second best diagnostic resolution, since observed 'zeros' are used to distinguish fault hypotheses. In this way, the closest fault hypothesis (using Hamming-distance) is taken, and a sharp, but not necessarily correct diagnosis result is given. In DMP and Row Reasoning, observed 'zeros' are not used to support the diagnosis result.

In Row Reasoning, only activated fault signals matter and cases 2 and 4 are not regarded. The cautious diagnosis DGN consists of all fault hypotheses that are not contradictory to the activated fault signals. That leads to error-free, but unsharp diagnostic results.

The confidence value p_i that DMP assigns to each fault hypothesis corresponds to $\boldsymbol{factor} sensit_i$. Only expected fault signals have an influence on p_i. Since the case of unexpected fault signals does not affect the confidence values of DMP, only a bad diagnostic resolution is achieved. In the new method, the fault hypotheses with unexpected fault signals are eliminated via the zero-violation-factor from equation (7).

2) Error rate: In case of big faults, no method gives an erroneous diagnosis because all theoretical fault signals are activated. However, for small faults, not all theoretical fault signals appear. Two methods commit errors in their diagnosis: Column Reasoning and DTS. In Column Reasoning, f_2/f_4 and f_3/f_5 are confused (see Table III). The DTS-method [5] is similar to Column Reasoning, but adds a retarding moment: The tests given in equation (2) are not immediately evaluated but only after a previously specified fault signal detection time has elapsed. If not all fault signals are activated after that time, DTS also gives an incorrect diagnosis. Sometimes, it is not able to give a diagnosis at all. Then all p_j are zero. That explains the diagnostic resolution below one.

V. CONCLUSION

The new method combines two advantages: It is highly reliable and at the same time provides very sharp diagnosis results. This is achieved by a clever combination of Row and Column Reasoning.

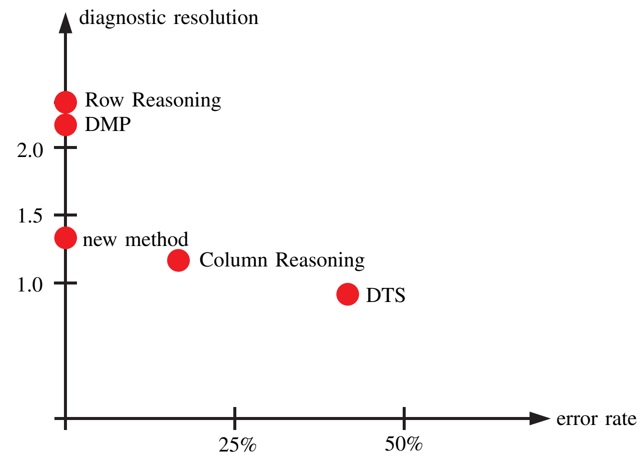

Fig. 5. Performance of different methods (two-tank-example)

The reliability of the new method is due to strict Row Reasoning, used in the evaluation of boolean fault signal occurrence, signs, sensitivities and fault signal occurrence order. Only the evaluation of occurrence times includes the possibility to apply Column Reasoning to some specific fault signals. They can be forced to be the first one or not the first one to appear. If those fault signals are well-chosen during the design of the $\boldsymbol{FSM} time$-table, the error-rate of the method is not affected and the diagnostic resolution can be refined.

REFERENCES

[1] C. Combastel, S. Gentil, and J. P. Rognon, "Toward a better integration of residual generation and diagnostic decision," in *Proceedings of IFAC Safeprocess'03*, Washington, USA, 2003.
[2] M. Cordier, P. Dague, F. Lvy, J. Montmain, M. Staroswiecki, and L. Trav-Massuys, ""conflicts versus analytical redundancy relations: a comparative analysis of the model-based diagnosis approach from the artificial intelligence and automatic control perspectives"," *"IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics"*, vol. 34, no. 5, pp. 2163–2177, 2004.
[3] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. M. Dekker, 1998.
[4] T. F. Petti, J. Klein, and P. S. Dhurjati, "Diagnostic model processor: Using deep knowledge for process fault diagnosis," *AIChE Journal*, vol. 36, p. 565, 1990.
[5] J. M. Kościelny, "Fault isolation in industrial processes by the dynamic table of states method," *Automatica*, vol. 31, pp. 747–753, 1995.
[6] J. Quevedo, V. Puig, F. Nejjari, T. Escobet, J. Aguilar, and B. Pulido, "An improvement of model based fault isolation based on a dynamic signature matrix for delayed systems," in *Workshop on Advanced Control and Diagnosis*, Duisburg, Germany, 2003.
[7] V. Puig, J. Quevedo, T. Escobet, and B. Pulido, "On the integration of fault detection and isolation in model based fault diagnosis," in *DX04, 15th International Workshop on Principles of Diagnosis*, Carcassonne, France, 2004.
[8] F. Schmid, "Model-based fault detection and isolation: A new approach for fault isolation in dynamic networks with time delays," Diploma Thesis, Dept. dEnginyeria de Sistemes,Automatica i Informatica Industrial (ESAII), Universitat Politecnica de Catalunya (UPC), Terrasa, Spain, 2004.
[9] B. Ould Bouamama, R. Mrani Alaoui, P. Taillibert, and M. Staroswiecki, "Diagnosis of a two-tank system," Intern Report of CHEM-project, USTL, Lille, France, 2001.
[10] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer, 2003.