Determination of Possible Minimal Conflict Sets using Components Clusters and Gröbner Bases

R. Ceballos and **M.T. Gómez** and **R. M. Gasca** and **S. Pozo**[1]

Abstract.
In engineering applications many models use polynomial constraints. The models are based on the knowledge of the behavior of the system to diagnose. Inputs and outputs of components are represented as variables of those constraints, and they can be observable and non-observable depending on the situation of the sensors in the system.

In this work, we propose a new approach to automate and to improve the determination of possible minimal conflict sets. This approach has two phases. In the first phase, we determine components clusters in the system in order to reduce drastically the number of contexts to consider. This is specially necessary in high density systems where components compose independent sets in themselves. In the second phase, we construct a reduced context network with the possible minimal conflicts. In this phase we use Gröbner basis reduction in the relevant contexts of each components cluster. This process is totally offline and the results obtained are very promising. This article shows these results applied to a heat exchangers system.

1 Introduction

Diagnosis allows to determine why a system correctly designed does not work as it was expected. It is based on the monitorization of a system that uses sensors integrated and which is supposed to work correctly. The diagnosis aim is to detect and to identify the reason of this unexpected behavior, in other words, to identify the parts which fail in a system. In order to explain a wrong behavior, the diagnosis process uses a determined set of observations and a model of the system. These faults have to be avoided if we want to keep a system within the desired production and security level.

Two communities work in parallel and usually separated in diagnosis: FDI (from Automatic Control) and DX (Artificial Intelligence). Nevertheless, the integration of FDI theories with the DX community has been shown in recent works (as [2] and [8]).

Both communities are based on the use of models. In the area of DX, the first work related to diagnosis was presented with the aim of identifying faults in the component systems based on its structure and its behavior [3]. DART [9] and GDE [15] were the first implementations to perform diagnosis; both detect possible faults using different inference mechanisms. In [18] and [14] it was proposed a general theory to explain the discrepancies between the observed and correct behaviors that the mechanisms subject to the diagnosis process (logical-based diagnosis) have. These two papers presented the diagnosis formalization.

Most DX approaches for components characterize the diagnosis of a system as a collection of minimal sets of failing components which explain the observed behaviors (symptoms). A conflict is a set of assumptions where at least one must be false. The assumptions are about behavioral modes of components. GDE [15] coupled with an ATMS [13] as inference engine uses previously discovered conflicts to constrain the search in the candidate space. In this approach, conflicts are identified in the process of constraint propagation through recording dependencies of predicted values given the system description and the observations. A diagnostic hypothesis must not contain a conflict. A conflict is minimal, if none of its subsets is a conflict. The main problem of using it is the big number of possible conflicts 2^n, where *n* is the number of components.

In this work, we propose a new approach to automate and to improve the determination of possible conflicts. This work is based on two advantages:
- A structural pretreatment in order to reduce drastically the computational complexity, specially in high density systems where components compose independent sets in themselves.
- The reduction of possible conflicts using Gröbner Bases reduction.

These two advantages can be done in an off-line process. Finding all minimal conflict has been a problem active enough at last years using CS-Tree [11]. In this line another work is proposed with a different approach [4]. This paper investigates how to improve the calculation of all minimal unsatisfiable subsets by preprocessing the system, reducing the size and the number of sets of constraints, and also using incremental satisfaction of constraints. In [5] new algorithms guided by structural properties are presented. The aim of these algorithms is to identify conditions under which the diagnosis are tractable. They specifically proposed to focus on the determination of the structural limitations and on algorithms to eliminate variables of the system.

Many techniques exploit the topological structure of the system using a constraint graph of the problem. For example, in [16], in order to reduce the computational complexity, they propose a two-steps approach: First, the system is analyzed to find overdetermined submodels, and then, all these submodels are transformed into consistency relations.

Gröbner bases theory is the origin of many symbolic algorithms used to manipulate multiple variable polynomies. For an introduction to Gröbner bases [1] and [12] can be consulted. Having a set of equality polynomial constraints Gröbner bases produces an equivalent system which is generally easier to solve.

In DX,[7] and [8], symbolic processing algorithms (Gröbner bases) of the initial model are used, and they generate the possible minimal conflict sets of the model according to its structure and behavior. Another proposition [17] related to our work about the objec-

[1] Computer Engineering Superior Technical School of Seville, Spain email: ceballos,mayte,gasca,sergio@lsi.us.es

Table 1. System Polynomial Model of the System of Heat Exchangers

C.	Constraints	C.	Constraints	C.	Constraints
N_{11}	$\mathbf{f}_{11}$-$\mathbf{f}_{12}$-$\mathbf{f}_{13}$	N_{23}	$\mathbf{f}_{27}$-f_{28}-f_{29}	E_3	$\mathbf{f}_{26}$-$\mathbf{f}_{27}$
	$\mathbf{f}_{11}\cdot\mathbf{t}_{11}$-$\mathbf{f}_{12}\cdot\mathbf{t}_{12}$-$\mathbf{f}_{13}\cdot\mathbf{t}_{13}$		$\mathbf{f}_{27}\cdot\mathbf{t}_{27}$-$f_{28}\cdot t_{28}$-$f_{29}\cdot t_{29}$		$\mathbf{f}_{31}$-f_{32}
N_{12}	f_{14}+f_{15}-$\mathbf{f}_{16}$	N_{24}	f_{210}+f_{211}-$\mathbf{f}_{212}$		$\mathbf{f}_{26}\cdot\mathbf{t}_{26}$-$\mathbf{f}_{27}\cdot\mathbf{t}_{27}$+$\mathbf{f}_{31}\cdot\mathbf{t}_{31}$-$f_{32}\cdot t_{32}$
	$f_{14}\cdot t_{14}$+$f_{15}\cdot t_{15}$-$\mathbf{f}_{16}\cdot\mathbf{t}_{16}$		$f_{210}\cdot t_{210}$+$f_{211}\cdot t_{211}$-$\mathbf{f}_{212}\cdot\mathbf{t}_{212}$	E_4	$\mathbf{f}_{16}$-$\mathbf{f}_{17}$
N_{13}	$\mathbf{f}_{17}$-$\mathbf{f}_{18}$-$\mathbf{f}_{19}$	E_1	$\mathbf{f}_{12}$-f_{14}		f_{32}-$\mathbf{f}_{33}$
	$\mathbf{f}_{17}\cdot\mathbf{t}_{17}$-$\mathbf{f}_{18}\cdot\mathbf{t}_{18}$-$\mathbf{f}_{19}\cdot\mathbf{t}_{19}$		f_{22}-f_{24}		$\mathbf{f}_{16}\cdot\mathbf{t}_{16}$-$\mathbf{f}_{17}\cdot\mathbf{t}_{17}$+$f_{32}\cdot t_{32}$-$\mathbf{f}_{33}\cdot\mathbf{t}_{33}$
N_{14}	f_{110}+f_{111}-$\mathbf{f}_{112}$		$\mathbf{f}_{12}\cdot\mathbf{t}_{12}$-$f_{14}\cdot t_{14}$+$f_{22}\cdot t_{22}$-$f_{24}\cdot t_{24}$	E_5	$\mathbf{f}_{18}$-f_{110}
	$f_{110}\cdot t_{110}$+$f_{111}\cdot t_{111}$-$\mathbf{f}_{112}\cdot\mathbf{t}_{112}$	E_2	$\mathbf{f}_{13}$-f_{15}		f_{28}-f_{210}
N_{21}	$\mathbf{f}_{21}$-f_{22}-f_{23}		f_{23}-f_{25}		$\mathbf{f}_{18}\cdot\mathbf{t}_{18}$-$f_{110}\cdot t_{110}$+$f_{28}\cdot t_{28}$-$f_{210}\cdot t_{210}$
	$\mathbf{f}_{21}\cdot\mathbf{t}_{21}$-$f_{22}\cdot t_{22}$-$f_{23}\cdot t_{23}$		$\mathbf{f}_{13}\cdot\mathbf{t}_{13}$-$f_{15}\cdot t_{15}$+$f_{23}\cdot t_{23}$-$f_{25}\cdot t_{25}$	E_6	$\mathbf{f}_{19}$-f_{111}
N_{22}	f_{24}+f_{25}-$\mathbf{f}_{26}$				f_{29}-f_{211}
	$f_{24}\cdot t_{24}$+$f_{25}\cdot t_{25}$-$\mathbf{f}_{26}\cdot\mathbf{t}_{26}$				$\mathbf{f}_{19}\cdot\mathbf{t}_{19}$-$f_{111}\cdot t_{111}$+$f_{29}\cdot t_{29}$-$f_{211}\cdot t_{211}$

V_{ob}=$\{f_{11},f_{12},f_{13},f_{16},f_{17},f_{18},f_{19},f_{112},f_{21},f_{26},f_{27},f_{212},f_{31},f_{33},t_{11},t_{12},t_{13},t_{16},t_{17},t_{18},t_{19},t_{112},t_{21},t_{26},t_{27},t_{212},t_{31},t_{33}\}$

V_{nob}=$\{f_{14},f_{15},f_{110},f_{111},f_{22},f_{23},f_{24},f_{25},f_{28},f_{29},f_{210},f_{211},f_{32},t_{14},t_{15},t_{110},t_{111},t_{22},t_{23},t_{24},t_{25},t_{28},t_{29},t_{210},t_{211},t_{32}\}$

tives, but not to the method, presents the concept of a possible conflict as an alternative to the use of pre-compiled dependency-recording. A method is developed to calculate the minimal chains which can be evaluated, and the minimal models which can also be evaluated. Also, the use of Gröbner bases in the FDI community has been already proposed in previous works [10] and [6]. In order to imporve these previous methodologies we present a novel methodology.

Our paper has been organized as follows. In section 2 it appears definitions and notations in order to clarify concepts for our approach. In section 3 we show an example of six heat exchangers. In section 4, we present the structural pretreatment and its usefulness. Then, in section 5, we give a description of the relevant context of the context network. Afterwards, section 6 describes the determination of the possible minimal conflict contexts. Finally, conclusions and future works are presented.

2 Definitions and Notation

In order to clarify the diagnosis process we need to expose some definitions and notation based on the concepts proposed on the diagnosis community (DX). Model-based diagnosis requires a system model which represents the behavior of the system and each model components.

Definition 1. System Polynomial Model (SPM): It can be defined as a finite set of polynomial equality constraints (*P*) which determine the system behavior. This is done by means of the relations between the system non-observable variables (V_{nob}) and the system observable variables (V_{ob}) which are directly obtained from sensors that are supposed to work correctly. The representation of a SPM is a tuple (P,V_{ob},V_{nob}).

Definition 2. Context Set (CS): It is a collection of components which compose the system. The possible context set will be 2^{comp}, where *comp* is the number of components of the system.

Definition 3. Context Network (CN): It is a graph is formed by all the elements of the context set of the system according to the way proposed by ATMS[13]. The CN has a natural structure of oriented graphs for set inclusion.

3 System example: A System of Heat Exchangers

In order to explain our methodology, we will use the following system which is a well-known example in the bibliography concerning model-based diagnosis. This system proposed in [10], consists of six heat exchangers, three flows f_i which come in at different temperatures t_i. This example defines three different subsystems, each one formed by two exchangers: E1, E2, E3, E4 and E5, E6. Each of the six exchangers and each of the eight nodes of the system are considered as components to verify their correct functioning. The normal functioning of the system can be described by means of polynomial constraints coming from three different kinds of balances:

$\sum_i f_i = 0$: mass balance at each node
$\sum_i f_i \cdot t_i = 0$: thermal balance at each node
$\sum_{in} f_i \cdot t_i - \sum_{out} f_j \cdot t_j = 0$: enthalpic balance for each heat exchanger

The table 1 represents the SPM that corresponds to the system of heat exchangers. This example is necessary in the following section in order to show the improvement obtained with our approach.

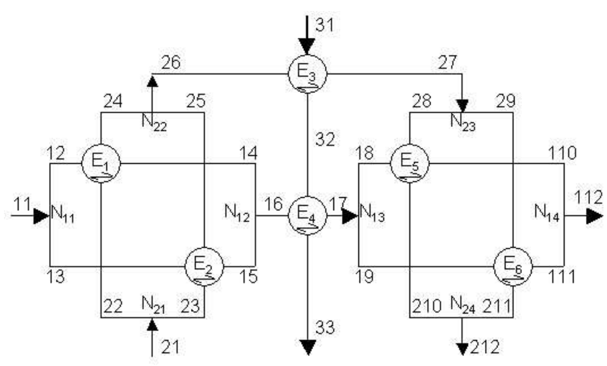

Figure 1. System of Heat Exchangers

4 Structural pretreatment: Identification of components clusters

In our methodology, the first step is to isolate independent subsystems. This structural pretreatment will give us a partition of the system into independent subsystems. The independence between subsystems guarantees us that the possible minimal conflict sets of the system can be obtained by the conflicts of all independent subsystems. The subsystems obtained are much smaller than the whole system, and therefore the computational complexity to detect conflicts from each subsystem is lower compared to the whole system. The partition of the system guarantees a smaller computational cost. In order to clarify the following steps we need the following definitions:

Definition 5. Components cluster (CC): A set of components *C* belonging to the complete system is a components cluster, if the following predicates are true:
- For all non-observable inputs and outputs of each component of *C*, these inputs and outputs are always linked to only components of *C*.
- It does not exist another set *C'* with less elements than *C* which validates the first predicate and it is included in *C*.

With the first predicate we are looking for the independence between conflicts of different components clusters. This predicate guarantees us that we are able to detect a conflict in a components cluster without information about other components clusters. This is possible because, in a components cluster, all the non-observable inputs and outputs are between components of the same cluster, and therefore, there is not any connection with any other component which is not monitored. Each components cluster is a set of components where we can detect conflicts.

We are looking for the division of our system into the biggest possible number of clusters in order to obtain a smaller computational cost. The second predicate guarantees us that the components clusters will be as small as possible, because it avoids that a set of components (components cluster) is composed of two or more independent sets of components. In this predicate we guarantee that a set *C'* will not exist inside *C*, because if *C'* exists, then another independent set *C"* with components $C \setminus C'$ could exist.

Example: For example, component E_3 is not completely monitored because we are not able to know the value of outputs f_{32} and t_{32}. Likewise, E_4 is not completely monitored because we are not able to know the value of inputs f_{32} and t_{32}. But we can monitor these two components if we think of these two components as if they were a subsystem with the same observable inputs and outputs that they had separately.

Algorithm: The following pseudo-code (see figure 3) defines the function *clustersIdentification(C)* which takes *C*, it consists of all the components of the system, and returns *A*, the set of components clusters. The algorithm will previously store in the set *E* all the pairs of components which have an common non-observable variable. The algorithm begins creating as many sets as *n*, where *n* is the number of components of the system. All these sets have one component. Then, for each element of *E*, which is a connection between two components $x \in S_1$ and $y \in S_2$, where S_1 and $S_2 \in A$, the algorithm merges sets S_1 and S_2. When the process is finished all components have assigned one components cluster.

Table 2. Components Clusters

CC	Constraints	CC	Constraints
1	$\{N_{11}\}$	4	$\{N_{14},N_{23},N_{24},E_5,E_6\}$
2	$\{N_{13}\}$	5	$\{E_3,E_4\}$
3	$\{N_{12},N_{21},N_{22}, E_1,E_2\}$		

```
clustersIdentification(C) return A
  E = {}
  A = {}
// Detect all connections between components
  foreach x ∈ C
    foreach y ∈ C
      if x ≠ y ∧ nonObsVar(x) ∩ nonObsVar(y) ≠ {}
        E = E ∪ {{x,y}}
      endif
    endforeach
  endforeach
// Generate clusters with only one component
  foreach x ∈ C
    A = A ∪ {{x}}
  endforeach
// Detect all components clusters
  foreach {x,y} ∈ E
    if ∃ S1,S2 | S1 ∈ A ∧ S2 ∈ A ∧ S1 ≠ S2
          ∧ x ∈ S1 ∧ y ∈ S2
      A = A \ S1
      A = A \ S2
      A = A ∪ {S1 ∪ S2}
    endif
  endforeach
```

Figure 3. Algorithm to select the components clusters

Auxiliary function of the algorithm:

- nonObsVar(x): This function returns the set of non-observable variables of a component *x*.

For the example presented in section 3, we obtained five components clusters which appear in table 2.

5 Reduction of the Constraint Network: Identification of the relevant context

The model which reflects the system structure and behavior presents the constraints that link the system inputs and outputs; but many times some intermediate variables are not observable and they do not allow to determine whether there are faults in components in a direct way. The idea is to produce an equivalent constraints model which has the same solution as the original, but without non-observable variables. This process is explained in following subsections.

5.1 Gröbner Bases

Having the set of equality polynomial constraints of the form P=0, Gröbner bases produces an equivalent system G=0, which has the same solution as the original, which is generally easier to solve. The main idea is to transform the polynomial constraint set into a standard form for the resolution of problems.

This algorithm is a generalization of Gauss elimination for multivariable lineal equations and of Euclides algorithm for one-variable polynomial equations. Concerning the advantages that the use of Gröbner bases has for the system models subject to diagnosis, it can be said that:

- If the model is over-constrained and has redundant equations, these redundancies will disappear when a reduced Gröbner base is calculated.

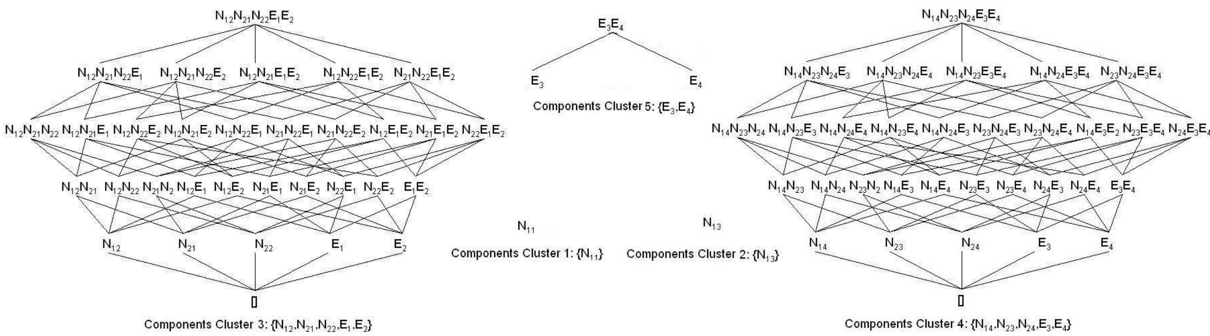

Figure 2. Context networks of the five components clusters

- If the model is over-restricted and inconsistent, one of the constraints which provides the algorithm will be 1=0, what is obviously inconsistent.

In this work, we supposed there is a function called GröbnerBases, which obtains the Gröbner Bases of a set of constraints. Let us consider, for instance, for the function called GröbnerBases($\{$x-a·c, y-b·d, z-c·e, f-x-y, g-y-z$\}$,$\{$a, b, c, d, e, f, g$\}$,$\{$x ,y ,z$\}$), the result would be the following system of polynomial constraints $\{$b·d+c·e-g=0, a·c-c·e-f+g=0$\}$.

5.2 Relevant contexts of the context network

For all the subsystems obtained in section 4, we will build a different and independent context network, as it appears in figure 2. Without this structural pretreatment, the number of nodes of the context network for the system of heat exchangers (as it appears in [8]) is 2^{14}-1. With the structural pretreatment the number of nodes are: (2^1-1)+(2^1-1)+(2^5-1)+(2^5-1)+(2^2-1) = 67

In our approach, a set of symbolic algorithms takes the constraints set of some of the context network in order to obtain a new constraints set without non-observable variables, Gröbner bases. In this new set of constraints, we know the values of all the variables.

In order to reduce the number of contexts to process, and therefore to improve the computational complexity, we will select which contexts are important to obtain conflicts in a system. These contexts will be the relevant contexts.

Definition 6. Irrelevant contexts (IC): It is a context which generates redundant constraints or does not generate any constraint. In other words, the set of constraints that Gröbner bases algorithm generates using an irrelevant context is empty or is included in the sets of constraints generated in other relevant contexts with less components.

Definition 7. Relevant contexts (RC): It is a context which can generate new constraints which can not be included in the set of constraints generated in other relevant contexts with less components.

We do not apply GröbnerBases function to irrelevant context, because this transformation will never generate new constraints. The set of constraints obtained using only relevant contexts is the biggest set of constraints that we can obtain. Irrelevant contexts are not important in order to obtain the minimal diagnosis of a system.

Example: The context $N_{12}N_{21}E_1E_2$ is irrelevant because if exists another context, $N_{12}E_1E_2$, with one less component (N_{21}) and which generates the same set of constraints than the first one. Another example of irrelevant context is E_1E_2, because GröbnerBases function is unable to obtain any constraint.

Algorithm: The algorithm takes a context *C* and it returns (boolean variable (*b*)) if the context *C* is a relevant context. It has two phases:

- The first step is to eliminate all constraints non-observable variables, because these constraints are directly a part of Gröbner Bases. For example, the context N_{11} is a relevant context and all their constraints have only observable variables. GröbnerBases function generates the same set of constraints of the component N_{11}, therefore it is not necessary to call because we have the result without calling.
- The second step is to eliminate those non-observable variables which appear only once in all the set of constraints of all the components of the context. Because GröbnerBases function can not eliminate these variables. Therefore, sending this constraint to GröbnerBases function can not give us any information, so this constraint is unnecessary to obtain Gröbner Bases. We will eliminate the constraint where this variable appears, in order to eliminate this kind of non-observable variables.

When all of these kind of variables are eliminated, the context will be relevant if their obtained constraints verify the following predicate: All the components must have at least one constraint. If exists one component *x* in the context *C* without constraints, then it will exist another context *C'* with one less component (*x*) than context *C*, which will generate the same set of constraints as context *C*. Therefore, it's better to apply GröbnerBases function to this context *C'* which is smaller than *C*, and, that will give us the same set of constraints.

If this predicate is false, it will not be necessary to call GröbnerBases function, because this call can not give us new constraints to detect conflicts in the system.

Auxiliary functions of the algorithm:

- nonObsVar(x): This function returns the set of non-observable variables of a component *x*.
- constraints(x): This function returns the set of constraints associated to a component *x*.
- deleteConstraints(x,e): This function deletes the constraint *e* of the set of constraint of the componen *x*.
- existCompWithoutConstraints(C): This function returns a boolean

```
Pre: C={A context}
isItARelevantContext?(C) return B
// Delete constraints without non-observable variables
  foreach x ∈ C
    foreach e ∈ constraints(x)
      if nonObsVar(e) = {}
        deleteConstraints(x,e)
      endif
    endforeach
  endforeach
// Delete irrelevant constraints to obtain Gröbner B.
  b = true
  while b ∧ ¬ existCompWithoutConstraints(C)
    b = ∃ x ∈ C ∧ e ∈ constraints(x) ∧ v ∈ nonObsVar(e)
      such that ∀ y ∈ C ∧ f ∈ constraints(y) ∧ f ≠ e ∧
      nonObsVar(f) ≠ {} : {v} ∩ nonObsVar(f) = {}
    if b
      deleteConstraints(x,e)
    endif
  endwhile
  b = ¬ existCompWithoutConstraints(C)
Post: b={It's true when C is a relevant context}
```

Figure 4. Pseudocode to determinate if a context is relevant

value which is true when the context *C* has one or more components with none constraints associated.

The GröbnerBases function receives the finite set of polynomial constraints of the components of the context, the set of observable variables of the context and the set of non-observable variables of the context, and calculates the constraint set of the context. Let us consider, for instance, the context represented by the components $N_{12}E_1E_2$. The GröbnerBases function receives the three previous sets, GröbnerBases ($\{$contextConstraintsOf($N_{12}E_1E_2$)$\}$, $\{f_{16}, f_{12}, f_{13}, t_{16}, t_{12}, t_{13}\}$, $\{f_{14}, f_{15}, f_{13}, f_{22}, f_{24}, f_{23}, f_{25}, t_{14}, t_{15}, t_{13}, t_{22}, t_{24}, t_{23}, t_{25}\}$), and the result is $\{f_{12}+f_{13}-f_{16}\}$.

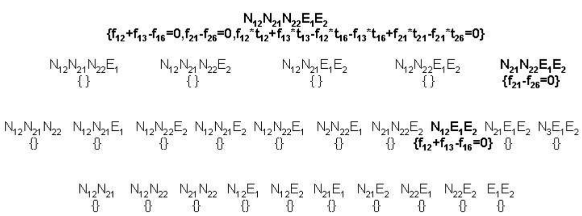

Figure 5. Context Network with symbolic constraint for the component cluster 3

For example, in figure 5 it appears the reduced context network for the component cluster 3 ($\{N_{12}, N_{21}, N_{22}, E_1, E_2\}$). It was obtained applying the corresponding GröbnerBases function to each relevant context of the context network of set 3. For this set GröbnerBases function is called only 3 times (it appears in boldface type). In the completed problem of the system heat exchanger, GröbnerBases function is called 7 times of all of 67 potential calls. Therefore, the search algorithm will avoid the computational treatment of these contexts, and will improve the efficiency in the search of possible conflicts. In table 3 it appears the differences between using all contexts (as in [7] and [8]) and to using our approach.

Table 3. Improvement between no reduction or using components cluster and relevant context (This test have been carried out in a Pentium IV-2Ghz with 512 MB)

	No reduction	Using CC	Using CC and RC
Number of Contexts	2^{14}-1	67	67
Calls to GB. function	2^{14}-1	67	7
Obtained Constraints	64	14	14
Elapsed time	4'2 days	7 Seconds	1 Second

6 Determination of possible minimal conflict context

In order to determinate the possible minimal conflicts we apply a constraint-driven algorithm. The following definitions are necessary in this process:

Definition 8. Context Analytical Redundancy Constraint (CARC): It is a constraint derived from SPM, in such a way that only the observed variables are related. In our approach, the set of CARC of the system is the union of all the constraints which we was obtained in each subsystem, by applying the corresponding GröbnerBases function to each relevant context of the context network.

Definition 9. Possible Minimal Conflict Context (PMCC): A relevant context *C* is a possible minimal conflict contexts if it does not have an empty set of constraints and verifies one of the following predicates:
- All its subcontexts are not possible minimal conflict contexts
- One or some of its subcontexts are possible minimal conflict contexts, but the union of all the CARCs of its subcontexts does not include all the CARCs of the context *C*. In other words, it exists at least one CARCs in context *C* which is not included in any of its subcontexts which are possible minimal conflict contexts.

Table 4. CARCs

Index	CC	CARC
1	1	f11 - f12 - f13
2	1	-(f11 t11) + f12 t12 + f11 t13 - f12 t13
3	2	f17 - f18 - f19
4	2	-(f17 t17) + f18 t18 + f17 t19 - f18 t19
5	3	f12 + f13 - f16
6	3	f21 - f26
7	3	f12 t12 + f13 t13 - f12 t16 - f13 t16 + f21 t21 - f21 t26
8	4	f18 + f19 - f192
9	4	f27 - f292
10	4	f18 t18 + f19 t19 - f18 t192 - f19 t192 + f27 t27 - f27 t292
11	5	f26 - f27
12	5	f16 - f17
13	5	f31 - f33
14	5	f16 t16 - f17 t17 + f26 t26 - f27 t27 + f31 t31 - f31 t33

The set of CARCs of the system is the union of all the constraints which we obtained in section 5.2 in each subsystem, applying the corresponding GröbnerBases function to each relevant context of the context network. These CARCs have associated an index to carry out a more efficient search process. In table 4 it appears all these constraints grouped by sets. In these constraints, their truth value can be evaluated from the system observed variables through the corresponding monitorization.

In our system example (heat exchangers), all the relevant contexts are possible minimal conflict context. All the possible minimal conflict contexts are represented in figure 6. The numbers at the bottom of each context are the CARCs indexes which corresponds to this context.

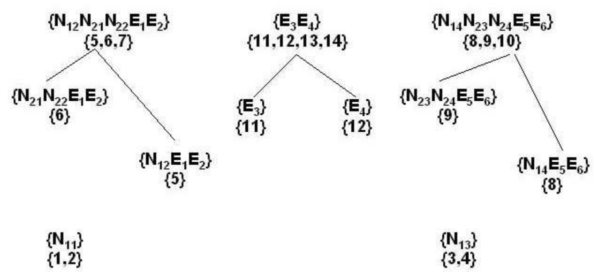

Figure 6. Possible Minimal Conflict Network of the System

In order to determine which relevant contexts are PMCC, we have to traverse the graph of the context network from the leaf nodes to the root node, in such a way that, if any of the upper contexts do not validate the definition of PMCC, it can not be considered as a PMCC.

For example, it can be observed that, when the graph is traversed from the context $N_{12}E_1E_2$ or context $N_{12}N_{22}E_1E_2$, the preceding context $N_{12}N_{21}N_{22}E_1E_2$ is a PMCC, because this bigger context has also other constraints (13 and 14). Therefore, these three contexts are PMCC.

The use of these minimal conflict contexts allows to establish a proposition for the minimal diagnosis as in [18].

7 Conclusions and future works

This paper proposes a new approach to automate and to improve the determination of possible minimal conflict sets. The determination of components clusters of the system drastically reduces the number of contexts to consider. The use of only the relevant contexts allow us to reduce the set of minimal possible conflict contexts. This process is totally offline and the results obtained are very promising. Extension to ODEs with polynomial constraints, in order to deal with dynamic systems, is our next objective.

As future works we want to improve our methodology using a constraint database in order to store polynomial constraints. Also, a constraint database will allow us to use the power of SQL in order to query the database.

ACKNOWLEDGEMENTS

This work has been funded by the Ministerio de Ciencia y Tecnología of Spain (DPI2003-07146-C02-01) and the European Regional Development Fund. (ERDF/FEDER).

REFERENCES

[1] B. Buchberger, 'Gröbner bases: An algorithmic method in polynomial ideal theory', *Multidimensional Systems Theory, N. K. Bose, ed.*, 184–232, (1985).

[2] M. Cordier, F. Lévy, J. Montmain, L. Travémassuyés, M. Dumas, M. Staroswiecki, and P. Dague, 'A comparative analysis of ai and control theory approaches to model-based diagnosis', *14th European Conference on Artificial Intelligence*, 136–140, (2000).

[3] R. Davis, 'Diagnostic reasoning based on structure and behavior', *In Artificial Intelligence 24*, 347–410, (1984).

[4] M. Garcia de la Banda, P. Stuckey, and J. Wazny, 'Finding all minimal unsatisfiable subsets', *Proc. Of the 5th ACM Sigplan Internacional*, (2003).

[5] R. Dechter, 'Diagnosis and planning using network-based algorithhms', *Final Report 1996-97*, (1984).

[6] E. Frisk, 'Residual generator design for non-linear, polynomial systems - a gröbner basis approach', *In Proc. IFAC Safeprocess, Budapest Hungary*, 979–984, (2000).

[7] R.M. Gasca, J.A. Ortega, M.Toro, and F. De la Rosa T, 'Diagnosis dirigida por restricciones simbólicas para modelos polinómicos', *Terceras Jornadas de trabajo sobre Metodologias Cualitativas Aplicadas a los Sistemas Socioeconomicos, Valladolid*, (2001).

[8] R.M. Gasca, C. Del Valle, R. Ceballos, and M. Toro, 'An integration of fdi and dx approaches to polinomial models', *DX*, (2003).

[9] M. Genesereth, 'The use of design descriptions in automated diagnosis', *Artificial Intelligence 24*, 411–436, (1984).

[10] C. Guernez, 'Fault detection and isolation on non linear polynomial systems', *15th IMACS World Congress on Scientific, Computation, Modelling and Applied Mathematics*, (1997).

[11] B. Han and S. Lee, 'Deriving minimal conflict sets by cs-trees with mark set in diagnosis from first principles', *IEEE Transcations on Systems, man and cybernetics- Part B: Cybernetics*, **29-2**, (1999).

[12] D. Kapur and Y. N. Lakshman, 'Elimination methods: An introduction, en symbolic and numerical computation for artificial intelligence', *D. Kapur and Mundy (eds.)*, (1992).

[13] J. De Kleer, 'An assumption-based truth maintenance system', *Artificial Intelligence 28*, **2**, 127–161, (1986).

[14] J. De Kleer, A. Mackworth, and R. Reiter, 'Characterizing diagnoses and systems', *Artificial Intelligence 56*, **2-3**, 197–222, (1992).

[15] J. De Kleer and B.C. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, (1987).

[16] M. Krysander and M. Nyberg, 'Structural analysis utilizing mss sets with application to a paper plant', *Proc. of the Thirteenth International Workshop on Principles of Diagnosis, Austria*, (2002).

[17] J.B. Pulido, 'Posibles conflictos como alternativa al registro de dependencias en lnea para el diagnstico de sistemas continuos', *Tesis Doctoral de la Universidad de Valladolid*, (2000).

[18] R. Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence 32*, **1**, 57–96, (1987).