Research Journal of Applied Sciences, Engineering and Technology 5(24): 5657-5664, 2013
DOI:10.19026/rjaset.5.4252
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: December 12, 2012 Accepted: January 21, 2013 Published: May 30, 2013

Research Article

Testable Subsystems Generation for Fault Detection and Isolation Using a Structural Matching Rank Algorithm Testability of an Electrical Circuit

Alem Saïd, Mokhtari Hicham, Ouziala Mahdi and Benazzouz Djamel
Solid Mechanics and Systems Laboratory (LMSS), University M'Hamed Bougara Boumerdes, Algeria

Abstract: In this study, an advanced way of dealing with testable subsystems and residual generation for fault detection and isolation based on structural analysis is presented. The developed technique considers execution issues; therefore, it has a more realistic point of view compared to classical structural approaches available in the literature. First, theoretical aspects of structural analysis are considered and introduced. Then the way of incorporating them to test the structural proprieties is explained. Finally, we show how the proposed (upgraded) matching rank algorithm can be used in order to choose the most suited matching that leads to computational sequences and detection tests. The method is demonstrated using an electrical circuit.

Keywords: Testable subsystems, Structural Analysis, Bipartite graph, Analytic Redundancy Relations ARR, Fault Detection and Isolation FDI

INTRODUCTION

In complex systems consisting of a great number of sensors, the obtained information starting from the various parameters and the signals must be controlled with an aim of the faults detection. This became a rigorous task with the increase in the number of the subsystems, i.e., actuators and sensors.

The structural approach constitutes a general plan to provide information when the system becomes complex. The main objective of the application of the structural approach is to identify the subsystems that present redundancy.

This study investigates the structural properties of dynamical systems by analyzing their structural model. The structural model of a system is an abstraction of its behavior model in the sense that only the structure of the constraints, i.e. the existence of links between variables and parameters, is considered and not the constraints themselves. The links are represented by a bipartite graph, which is independent of the nature of the constraints and variables (quantitative, qualitative, equations, rules, etc.) and of the value of the parameters. This indeed represents a qualitative, very low level, easy to obtain, model of the system behavior.

Structural analysis is concerned with the properties of the system' structure model, which resorts to the analysis of its bipartite graph. As this graph is independent of the value of the system parameters, structural properties are true almost everywhere in the system parameter space.

The FDI community, Staroswiecki and Declerck (1989), adopted the analysis of system structure, originally developed for the decomposition of large systems of equations for their hierarchical resolution and structural concepts were used for the analysis of system monitorability using complete matchings on a graph. A recent overview of structural analysis can be found in Blanke *et al.* (2006), which also provides essential references to the field. Research on properties of residual generators for linear differential algebraic systems were treated in Nyberg and Frisk (2006) and Combined structural and polynomial methods were pursued in Krysander (2006) and Krysander and Nyberg (2005). Techniques for active fault isolation were treated in Niemann (2006) and suggested in a structural context in Blanke *et al*. (2006). Structural isolability properties were investigated and exemplified in Düstegör (2005) and Düstegör *et al*. (2006).

In spite of their simplicity, structural models can provide many useful information for fault diagnosis and fault-tolerant control design, since structural analysis is able to identify those components of the system which are-or are not-monitorable, to provide design approaches for analytic redundancy based residuals, to suggest alarm filtering strategies and to identify those components whose failure can be tolerated through reconfiguration.

Corresponding Author: Alem Saïd, Solid Mechanics and Systems Laboratory (LMSS), University M'Hamed Bougara Boumerdes, Algeria

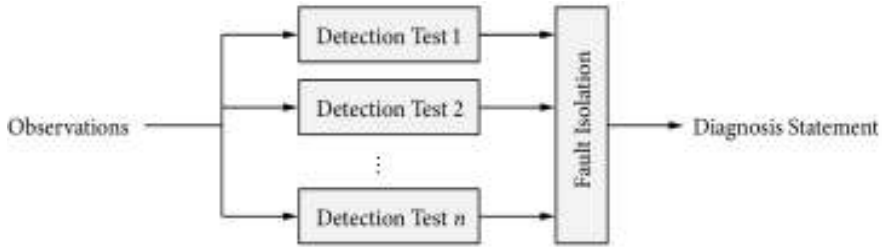

Fig. 1: A typical FDI-system

FAULT DETECTION AND SYSTEM ANALYSIS

A typical FDI-system consists of a set of fault detection tests associated with a fault isolation scheme (Fig. 1). The input to the FDI-system is a set of observations, i.e., measurements, from the monitored system and the output is a diagnosis statement. The diagnosis statement contains a collection of faults that can be used to explain the observations.

Given a set of observations, y, the outcome of a detection test τ_i is a binary fault detection result, d_i, equal to, for instance,1 if the test has alarmed, or equal to 0, otherwise.

To enable fault isolation, different detection tests typically monitors different faults and thus different parts of the system. Each fault detection test typically utilizes a subset of the observations in order to determine if any fault is present in its monitored part of the system.

Common traditional approaches for construction of fault detection tests are for example *limit checking*, i.e., to check if a sensor is within its normal operating range, or to employ hardware redundancy. For instance, if two sensors are used to measure the same physical quantity, it is possible to test if one of the sensors is faulty by comparing the values of the sensors. Another approach, providing potentially increased diagnosis performance and in which the need of additional, redundant, hardware is avoided, is to use detection tests based on *residuals*. Detection tests based on residuals will be generated further in the next section using a structural approach.

STRUCTURAL MODEL

This section provides a brief overview of the main concepts in structural analysis. It provides a baseline for further extensions to enhanced structural faults detection and isolation.

The essential idea in analytic fault diagnosis is to establish relations to test whether measured and other known variables satisfy all relations that describe the system's normal behavior. If this is not the case, some violation of the normal behaviour has occurred, i.e., one or more faults are present in the system. Relations that can be used for such testing are referred to as redundancy relations.

Let a system be described by a set X of unknown variables, a set K of known variables and a set of constraints C on these variables.

Then there may exist a set $Cm \subseteq C$ from which all variables in X can be determined. A system that has this property is said to have a complete matching on the unknown variables. If any constraints exist that were not used to obtain such matching, the set of unmatched constraints $Cum \subset C$ may be used to test the consistency between known variables and the system's normal behavior. Hence, redundancy relations are obtained from the unmatched constraints.

Solving for unknown variables in a nonlinear system can be rather complex if done directly on the analytical form of the constraints. Structural analysis offers a significant shortcut. It is a method to determine possible ways to solve a set of constraints without actually doing so. Making a graph representation of the relations between constraints and unknown variables makes it possible to seek through a graph to determine how one could solve for unknown variables. The result of structural analysis is a receipt that, in a symbolic form, describes how unknown variables could be calculated from known variables, using the system constraints. Analytical expressions are not used until a complete structural solution is found. This dramatically reduces the complexity of finding parity equations for fault diagnosis.

The salient feature of the structural analysis approach is that graph theory exists which can be employed to find all possible ways the set of system constraints can be matched to unknown variables (Dulmage and Mendelsohn, 1959). As sets of unmatched constraints, in general, differ from matching to matching, structural analysis can determine the entire set of possible parity relations.

Being very useful as a first step of the analysis, the results of structural analysis are, however, only indicative of the existence of the associated analytical results. The existence of a structural parity relation does not guarantee the existence of an analytic counterpart. Non-existence in the structural domain does, however, imply non-existence also in the analytical domain.

Structural concepts were studied early in the applied mathematics community and various theoretical algorithms were developed in Hopcroft and Karp (1973) and Izadi-Zamanabadi *et al*. (2003). Structural analysis was and is used intensively in chemical

engineering for solving large sets of equations (Leitold and Hangos, 2001; Unger *et al*., 1995). The structural approach and the features it offers for analyzing monitoring and diagnosis problems were first introduced in Staroswiecki and Declerck (1989). Extensions to the analysis of reconfigure ability and fault-tolerance emerged in Izadi-Zamanabadi and Staroswiecki (2000) and Staroswiecki and Gehin (2000).

The structural analysis approach was presented in a digested form in Düstegör *et al*. (2004). Structural analysis has hence evolved during several decades. However, the salient features of the theory and the possibilities it offers have only become apparent to a larger community in the field of automation and automatic control over the last few years (Åström *et al*., (2001; Izadi-Zamanabadi *et al*., 2003) with applications reported in, e.g., Izadi-Zamanabadi and Staroswiecki (2000).

Structure as a bi-bipartite graph : The model structural represents the links between a set of variables and a set of constraints. It is an abstraction of the behavior model because it describes simply which variables are connected to which constraints. Hence, the structural model presents the basic features and properties of a system which are independent of its parameters. The behavior model of a system is defined by a pair (C, Z) where Z is a set of variables and parameters and C is a set of constraints (Blanke *et al*., 2006; Düstegör *et al*., 2004).

This structure can be represented by a graph bipartite. A graph is bipartite if its vertices can be separated in two disjoin sets C and Z in such a way that every edge has one endpoint in C and the other one in Z.

To note that the graph bipartite is undirected graph, which can be interpreted as follows: All the variables and parameters connected with a given constraint vertex has to satisfy the equation or rule this vertex represents. This graph allows representing the structure of rather general models including both differential and algebraic constraints.

In the bipartite graphs the vertices of Z are represented by circles while constrained the vertices of C will be represented by bars.

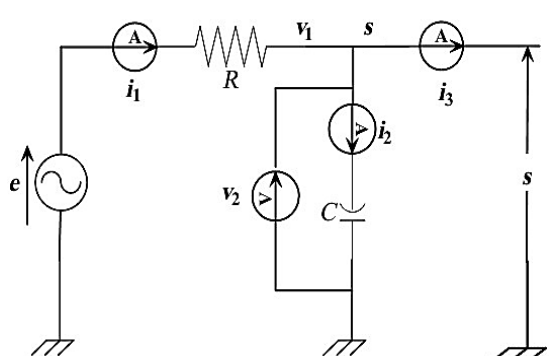

Fig. 2: Electrical circuit

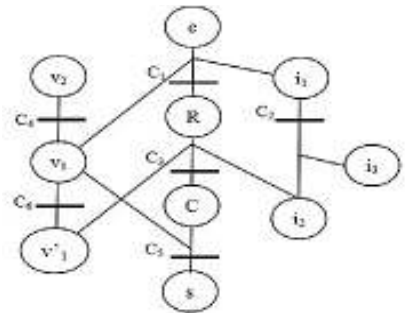

Fig. 3: Structure graph of the electrical system

Table 1: Incidence matrix of the system

Variables/ constraints	Unknown variables				Known variables					
	R	C	$\dot{V}_1$	s	e	v_1	v_2	i_1	i_2	i_3
C_1	1				1	1		1		
C_2								1	1	1
C_3		1	1						1	
C_4						1	1			
C_5				1		1				
C_6			1			1				

Example 1: For a better comprehension, the example above will be used, (Ploix *et al*., 2005) (Fig. 2).

To consider an electrical system where the courant and the tension are the variables propagating in the circuit.

The system consists of the components {Resistor R, capacitor C, Output charge s, Generator e, Sensors i, v). A continuous model with variable in continuous time is given by the following constraints:

- Resistor C_1: $e - v_1 = Ri_1$
- Node rule C_2 : $i_1 = i_2 + i_3$
- Capacitor C_3 : $i_2 = C\frac{d}{dt}v_1$
- Tensions rule C_4: $v_1 = v_2$
- Charge C_5: $v_1 = s$
- Derivative constraint C_6 : $\dot{V}_1 = \frac{d}{dt}v_1$

The graph bipartite of our system is represented in the Fig. 3 as follows:

We can also represent our system in the form of incidence matrix, Table 1. Every column of the matrix correspond to a circle-vertex and every row correspond to a bar-vertex, the variables that are connected to a constraint is marked by 1 in its box as is given in the following:

Known and unknown variables: System variables and parameters can be decomposed into known (Internal) and unknown (external). System inputs and outputs are examples of variables that are usually known. In the same way, parameters that are entries for the model or were previously identified are certainly known. Thus, known variables are available in real-time and they can directly be employed in the faults diagnosis.

Unknown variables are not directly measured, although there might exist some way to compute their value from the values of known ones (*calculability*). In

the electrical circuit example, the last five columns of the incidence matrix {e, v_1, v_2, i_1, i_2, i_3} correspond to known variables, while the first four ones correspond to unknown variables {R,C, s,$\dot{V}_1$}, (the components and the derivative state).
Variable set is portioned into:

$$Z = k\ U\ X$$

Or: K (known variable) and X (unknown variable)
And: $C = C_K\ U\ C_X$

C_K constraint, which links only the known variables and C_X includes constraints in which at least one unknown variables appears.

MATCHING ON A BI-PARTITE GRAPH

The basic tool for the structural analysis is the concept of matching founded on a bipartite graph. In other words, a matching is a causal assignment, which associates unknown variables of the system with the constraint of system from which they can be calculated. The unknown variables that cannot be matched cannot be calculated. The variables, which can be matched several ways, can be determined by various ways, which provide means for fault detection and a possibility for reconfiguration (Düstegör *et al*., 2004).

On a Bipartite graph, the matching is a set of arcs/edges such that any two edges have no common node (neither in C nor in Z) and it is represented by an arc in Gras. This is given as follows, in Fig. 4, an example of two possible matchings for the electrical circuit system:

In the incidence matrix, the matching is represented by one ①. Only one ① is authorized in same column and line.

Matching algorithm: We can count many algorithms to find the maximal matching, like the classical maximal matching algorithm, the maximal flow matching, Hungarian method (Dulmage and Mendelsohn, 1963; Hopcroft and Karp, 1973).

However, we will use an efficient and upgraded algorithm based on ranks to find the matchings considering his effectiveness and his simplicity of application on a system. See for basics, Blanke *et al*. (2006). In our ameliorated version of the algorithm, it is possible to generate all non-matched constraints in each iteration, this is useful for diagnosis and for tests generation.

Ranking algorithm
Given: an incidence matrix or a structural graph

- Mark all know variables
- i = 0
- Fine all constrain in the current table with exactly one unmarked variable. Associate them the rank i and mark these constraints as well as the corresponding variable.
- Set i = i + 1
- If there are unmarked constraints where all variables with rank i, mark them and link them to the pseudo-variable ZERO
- If there are unmarked variables or constraints, continue with step2

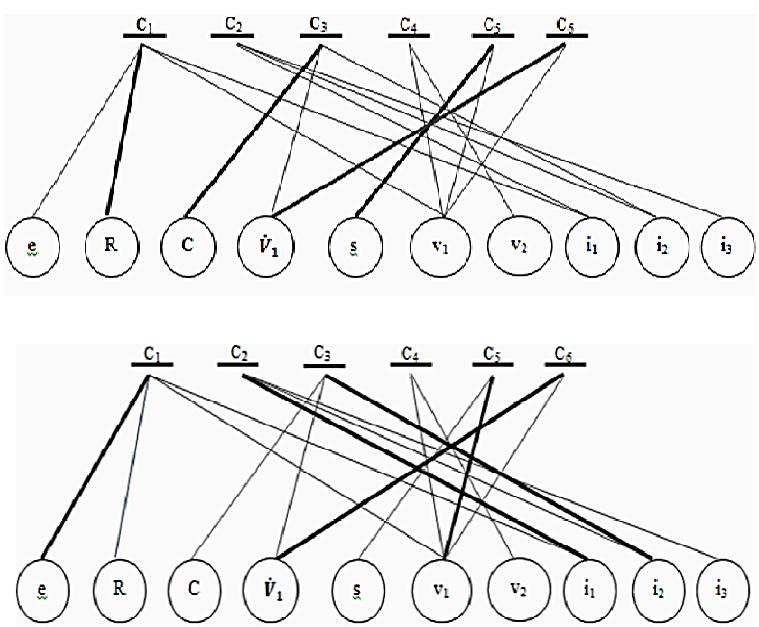

Fig. 4: Possible matching on the bipartite graph of the electrical system

The constraint propagation algorithm applied to the electrical circuit system is exhaustively given as follows:

Since, {e, v_1, v_2, i_1, i_2 and i_3} are known, only the variable set {R,$\dot{V}_1$, C, s} has to be matched.

- Starting set (rank 0): {$e, v_1, v_2, i_1, i_2, i_3$ }
- First step (rank 1): match $\dot{V}_1$ with C_6 *and* match s with C_5.
- Second step (rank 2): match C with C_3 and match
- R with C_1.

It is clear that all our unknown variables are matched.

System' canonical decomposition: The graph theory recalls a classical result from bipartite graph, (Dulmage and Mendelsohn, 1959), which states that any finite-dimensional graph can be decomposed into three sub-graphs with specific properties, respectively associated with an *over-constrained*, a *just-constrained* and an *under-constrained* subsystem. This decomposition is canonical, i.e., for a given system, it is *unique*. The three subsystems play a major role in the analysis of the system structural properties: Observability, Controllability and Monitorability.

A graph (C, Z) is called:

- **Over-constrained:** if there is a complete matching on the variables Z but not on the constraint C,
- **Just-constrained:** if there is a complete matching on the variables Z and on the constraints C,
- **Under-constrained:** if there is a complete matching on the constraints C but not on the variables Z.

PROPERTIES OF STRUCTURAL ANALYSIS

Observability: Observability informs us about the unknown variables of our system that can be checked and observed.

Structural Observability condition (Blanke *et al*., 2006):

- All the unknown variables are reachable from the known ones
- The over-constrained and the just-constraint subsystems are causal
- The under-constrained subsystem is empty

To simplify, to ensure that the system is observable, it is necessary that, it be over-constrained or just-constrained. Moreover, it is necessary that all the unknown variables are reachable (Blanke and Staroswiecki, 2006), i.e., can be calculated by means of the known variables.

Monitorability: A system is said to be monitorable, if it can be determined using only known variables, whether the system constraints are satisfied or not.

In addition, this is feasible through designing fault detection and isolation algorithms based on the analytical redundancy relation.

Analytical redundancy ARR based fault diagnosis tries to identify faults by comparing the actual behavior of the system, with the theoretical behavior described by the system constraints. They are given in the means of only known variables. ARRs are the constraints that express this redundancy. This redundancy is represented in the following with the over-constrained and unmatched constraints that denote the testable subsystems.

The ARR should have the following properties:

- **Robust:** i.e., insensitive to unknown input and unknown parameters.
- **Sensitive to faults:** this insures that they are not satisfied when faults are present, so that there is no missed detection.
- **Structured:** this insures that in the presence of a given fault, only subsets of the ARR are not satisfied, thus allowing recognizing, the fault which occurred.

The following two necessary conditions meant for which a fault to be monitorable are equivalent:

- X is structurally observable in the system
- It belong to the over-constrained part of the system (C, Z), i.e., that the unknown variables must be observable (adding sensors) and it belongs to an overstrained subset (complete matching on the variables Z**)**.

Design of analytic redundancy relations: Redundancy relations are sub-graphs of the structure graph, which are associated with complete causal matching of the unknown variables and associated with the over-constrained subsystem of a reduced bi-partite graph. Redundancy relations are composed of alternated chains, which start with known variables and end with non-matched constraints, here the chosen output is labeled ZERO.

Designing a set of residuals requests for building maximal matching on the given structural graph, under derivative causality and identifying the redundancy relations as the non-matched constraints in which all the unknowns have been matched. The algorithm to find maximal matching has been previously presented.

In our studied example, the incidence matrix obtained after applying the rank algorithm is in the form below, Table 2:

It is clear that constraints C_2 and C_4 are the non-matched constraints, where all the variables are matched.

Table 2: Incidence matrix after matching

Variables/ constraints	Internal variables				External variables						Rank
	R	C	$\dot{V}_1$	s	e	v_1	v_2	i_1	i_2	i_3	
C1	1				1	1		1			2
C_2								1	1	1	Zero
C_3		1	1						1		1
C_4						1	1				Zero
C_5				1		1					0
C_6			1			1					0

Constraint C_2 and C_4 are used for the design of redundancy relations by replacing all the matched variables by the known variables, the results are as following:

The given calculation order of unknown variables from knows is as follow:

$$\underline{C4}: v_1 = v_2 \rightarrow \underline{C5}: v_1 = s \rightarrow \underline{C2}: i_1 + i_2 + i_3 \rightarrow \underline{C6}: \dot{V} = \frac{d}{dt} v_1 \rightarrow \underline{C3}: i_2 = C \frac{d}{dt} v_1 \rightarrow \underline{C1}: e - v1 = R\, i1 \rightarrow \text{Zero}$$

By replacing the known variables we have:

$$C_1{:}0 = Cv_1 + R\, i_1 - e = r(t)$$

where, r(t): is a residual obtained by the analytical redundancy relation, the residual is an evaluation of the ARR, it is equal to 0 if the system is free-fault and to a non-zero value in the presence of a fault (fault detection). It is s-called also, the fault indicator or fault signature, if two faults have the same fault signature, they are not structurally distinguishable (fault isolation).

We can note that Kirchhoff laws are generally the redundant constraints in circuit analysis and they are commonly used to calculate residues.

The order of operations on constraints is*(coming from the last)* C_1 (C_3 (C_6 (v_1)),C_2 (i_2, i_3),C_3 (C_4 (v_2))) → zero

We remark that v_2 and v_1 are redundant sensors and it is similar for i_1 and (i_2,i_3). Hence, it is well known that hardware redundancy is generally the commonly used approach in circuit tests, the next section deals with redundancy elimination in testable subsystems generated using a structural methodology.

GENERATION OF TESTABLE SUBSYSTEMS

A testable subsystem is a set of constraints, which leads to an Analytical Redundancy Relation (ARR). An ARR is a static or dynamic time evolution constraint that contains only known variables and it is equal to zero (0) when the system operates according to its normal operation model.

Any given subsystem model cannot always be checked because testing the consistency of a subset of elementary models requires that some physical variables to be known. A test, also called a detection test, dedicated to a subsystem model is a Boolean function defined on a space of known variables, based on all the analytical relations of the elementary models belonging to the subsets.

In the following, a short survey on related works is given. Armengol *et al*. (2009) resumes all the methods used in the literature to generate the testable subsystems in the context of structural analysis.

Krysander *et al.* (2008) proposed an algorithm for finding all MSOs (minimal structurally over-constrained subsystems), i.e., testable subsystems TSS. This algorithm is based on the Dulmage-Mendelsohn decomposition, (Dulmage and Mendelsohn, 1959) and on a top-down approach. It starts with the entire model and then reduces the size of the model step by step until a TSS remains.

A handicap of this method is that, the same TSS can be found more than once. Therefore, this algorithm is not optimal in terms of efficiency. Krysander *et al*. (2008) presented improvements in order to increase the efficiency.

Even if the complexity of this method is lower than for the last approach, the main disadvantage of this method is that it cannot take into account the notion of deductibility/reachability of variables, Blanke and Staroswiecki (2006): all variables of the system are considered as deductibles and, therefore, some ARRs may not be achievable. Of course, systems with branchings cannot be managed.

In this study, we combine the rank algorithm with the approach proposed in Krysander *et al*. (2008) to ameliorate TSS generation. The main idea is using the elimination of the redundancy in the incidence matrix and reduces the number of constraints used to the minimal; this will generate the *basic testable subsystems* defined in Ploix *et al*. (2005).

Taking in account the following definition of deductibility (calculability):

Definition 1: Two constraints C1 and C2 defined on a set of physical variables V will be considered as *equivalent*$\sigma(C1) = \sigma(C2)$. It is denoted $C1 \Leftrightarrow C2$.

The set of references $\sigma(C)$, also called constraint support, containing either the name of the constraint if it is related to only one component state, or, elsewhere, the names of all the constraints that compose C.

Definition 2: A constraint C2 overestimates (miscalculates) a constraint C1 if $\sigma(C1) \subset \sigma(C2)$, It is denoted $C1 \subset C2$.

Constraints overestimating others and equivalent constraints have to be removed because they are not minimal. This is the main idea in Krysander *et al*. (2008) and Yassine *et al*. (2008), they called it redundant constraints.

Recall the example 1, the incidence matrix for the reduced model of electrical circuit (without miscalculated and redundant constraints and variables) is as follow, Table 3:

Table 3: Incidence matrix after re-arrangement and reduction

Variables/ constraints	Internal variables				External variables			
	R	C	$\dot{V}_1$	s	e	v_1	i_1	i_2
\	1				1	1	1	
C_3		1	1					1
C_6			1			1		
C_5				1		1		

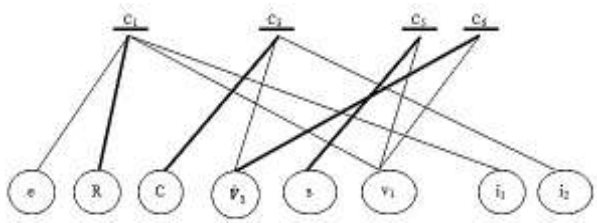

Fig. 5: Bipartite graph of reduced system with a maximal matching on variables

One finds that, by establishing all the possible matchings, it is possible to generate all the detection tests (basic tests), which will be used for the design of all Testable subsystems.

Figure 5, illustrates a maximal matching on the reduced model.

It is proposed that: by tracking all the linked edges on the graph, starting from a matched variable and passing by a constraint, allows us to tracing the order of calculability directly from the Bipartite graph and generate a testable subset (a basic subset),this is called a *mapping*. As a result, by establishing all the possible mappings on Fig. 5, one can get all the testable subsystems (the sets of all possible basic tests in different calculation orders) i.e., the over-determined subsets of constraints, which can be used directly in the FDI-scheme defined in Fig. 1.

CONCLUSION

The structural analysis enables the exploration of local redundancy of the system, it also allows highlighting the possibilities for monitoring through the structural properties such as, Monitorability and determination of calculation sequences, the result is a residue and allows generating all the testable subsystems. Analysis of residues structure has in purpose to assess the faults detectability and isolability, but it can also suggest what sensors should be implemented in order to change the status of the system components, to move a non-observable one to an observable (Alem and Benazzouz, 2013).

An ameliorated ranking algorithm is used to generate the testable subsets and produce detection tests by eliminating the miscalculated and equivalent (redundant) constraints in the structured model. The efficiency of the proposed method is confirmed along the paper using an electrical circuit system. The obtained results are clearly compared with those in Ploix *et al*. (2005). The paper gives a short investigation on related works and commonly utilized terminology on testable subsystems generation using a structural approach.

In future works, one can include some causal interpretations on computational sequences, i.e., derivative and algebraic loops. Amelioration on the proposed algorithm o fit with large-scale systems with dynamic behaviors will be also considered. It is possible to take into account the exponential complexity of the iterative algorithm and proposing a heuristic approach to deal with an optimization on the combination of all the obtained basic matchings in each iteration.

REFERENCES

Alem, S. and D. Benazzouz, 2013. Optimal sensor placement for fault detection and isolation by the structural adjacency matrix. Int. J. Phys. Sci., 8(6): 225-230.

Armengol, J., A. Bregon, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido and L. Trave-Massuyes, 2009. Minimal structurally over-determined sets for residual generation: A comparison of alternative approaches. Proceedings of the 7th IFAC-Safeprocess 09. Barcelona, Spain, pp: 1480-1485.

Åström, K., J. Albertos, P. Blanke and M. Isidori, 2001. A Control of Complex Systems. In: Schaufelberger, W. and R. Sanz (Eds.), Springer, London/New York.

Blanke, M. and M. Staroswiecki, 2006. Structural design of systems with safe behavior under single and multiple faults. Proceedings of the 16th IFAC Symposium Safeprocess. Beijing, China, pp: 511-516.

Blanke, M., M. Kinnaert, J. Lunze and M. Staroswiecki, 2006. Diagnosis and Fault-Tolerant Control. 2nd Edn., Springer, Berlin, Heidelberg.

Dulmage, A.L. and N.S. Mendelsohn, 1959. A structure theory of bipartite graphs of finite exterior dimension. Trans. Royal Soc. Canada Ser., 53: 1-13.

Dulmage, A.L. and N.S. Mendelsohn, 1963. Two algorithms for bipartite graphs. J. Soc. Indus. Appl. Math, 11(1): 183-194.

Düstegör, D., 2005. Structural analysis for FDI: Algorithmic issues. Ph.D. Thesis, University of Science and Technology of Lille (French), France.

Düstegör, D., E. Frisk, V. Cocquempot, M. Krysander and M. Staroswiecki, 2006. Structural analysis of fault isolability in the DAMADICS benchmark. Control Eng. Pract., 14(6): 597-608.

Düstegör, D., V. Cocquempot and M. Staroswiecki, 2004. Structural analysis for fault detection and identification: An algorithmic study. Proceedings of the 2nd Symposium on System Structure and Control (SSSC'04). Oaxaca, Mexico.

Hopcroft, J.E. and R.M. Karp, 1973. An n/sup 5/2 algorithm for maximal matchings in bipartite graphs. SIAM J. Comput., 2(4): 225-231.

Izadi-Zamanabadi, R. and M. Staroswiecki, 2000. A structural analysis method formulation for fault-tolerant control system design. Proceedings of the 39th IEEE Conference on Decision and Control. Sydney, Australia, pp: 4901-4902.

Izadi-Zamanabadi, R., M. Blanke and S. Katebi, 2003. Cheap diagnosis using structural modeling and fuzzy-logic based detection. Control Eng. Practice, 11(4): 415-422.

Krysander, M., 2006. Design and analysis of diagnosis systems using structural methods. Ph.D. Thesis, Linköping University, Sweden.

Krysander, M. and M. Nyberg, 2005. Fault isolability prediction of diagnostic models. Proceedings of 16th International Workshop on Principles of Diagnosis DX-05. Pacific Grove, CA, USA.

Krysander, M., J. Aslund and M. Nyberg, 2008. An efficient algorithm for finding minimal over-constrained subsystems for model-based-diagnosis. IEEE T. Syst. Man Cy. A, 38(1): 197-206.

Leitold, A. and K.M. Hangos, 2001. Structural solvability analysis of dynamic process models. Comput. Chem. Eng., 25(11-12): 1633-1646.

Niemann, H.H., 2006. A setup for active fault diagnosis. IEEE T. Automat. Contr., 51(9): 1572-1578.

Nyberg, M. and E. Frisk, 2006. Residual generation for fault diagnosis of systems described by linear differential-algebraic equations. IEEE T. Automat. Contr., 51(12): 1995-2000.

Ploix, S., M. Desinde and S. Touaf, 2005. Automatic design of detection tests in complex dynamic systems. Proceeding of 16th IFAC World Congress. Prague, Czech Republic, July 4-8.

Staroswiecki, M. and P. Declerck, 1989. Analytical redundancy in nonlinear interconnected systems by means of structural analysis. Proceedings of the IFAC Symposium on Advanced Information Processing in Automatic Control (AIPAC'89). Nancy, France, pp: 23-27.

Staroswiecki, M. and A.L. Gehin, 2000. From control to supervision. Proceedings of the IFAC Symposium Safeprocess'00. Budapest, Hungary, Vol. 1.

Unger, J., A. Kröner and W. Marquardt, 1995. Structural analysis of differential-algebraic equation systems-theory and applications. Comput. Chem. Eng., 19(8): 867-882.

Yassine, A., S. Ploix and J.M. Flaus, 2008. A method for sensor placements taking into account diagnosability criteria. Appl. Math. Comput. Sci., 18(4).