

# STRUCTURAL ANALYSIS FOR FAULT DETECTION AND ISOLATION AND FOR FAULT TOLERANT CONTROL

**Marcel Staroswiecki**

*LAIL-CNRS UMR 8021, Polytech'Lille, University Lille I, France*

**Keywords:** System structure, bipartite graph, digraph, causal graph, fault detection and isolation, fault tolerance, fault accommodation, system reconfiguration

## Contents

- 1. Introduction
- 2. Structural model
  - 2.1. Structure as a Bipartite Graph
  - 2.2. Subsystems
  - 2.3. Structural Properties
  - 3. Matching on a bipartite graph
    - 3.1. Definitions
    - 3.2. Oriented Graph Associated with a Matching
  - 4. Causal interpretation
    - 4.1. Algebraic Constraints
    - 4.2. Differential Constraints
    - 4.3. Loops
  - 5. System Decomposition
    - 5.1. Canonical Decomposition
    - 5.2. Causal Subsystems
  - 6. Observability
    - 6.1. Structural Observability Conditions
    - 6.2. Graph Based Interpretation and Formal Computation
  - 7. Monitorability
    - 7.1. Analytical Redundancy Based Fault Detection and Isolation
    - 7.2. The Structurally Monitorable Subsystem
    - 7.3. The Design of Robust and Structured Residuals
  - 8. Fault tolerant estimation
  - 9. Controllability
    - 9.1. Structural Controllability Conditions
    - 9.2. Fault Tolerant Control
  - 10. A simple example
  - 11. Conclusion
- Glossary
- Bibliography
- Biographical Sketch

## Summary

This article introduces the structural model of a system, which 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 very low level, easy to obtain, model of the system behavior. In spite of their simplicity, structural models can provide many useful information for Fault Detection and Isolation (FDI) and Fault Tolerant Control (FTC) design, since structural analysis, which consists of the analysis of the system bipartite graph, 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 – or cannot – be tolerated through reconfiguration.

## 1. Introduction

The design of fault detection and isolation (FDI) and fault tolerant control (FTC) algorithms, like the design of control algorithms, is based on models. Unlike control, which addresses nominal (sometimes uncertain) systems, FDI and FTC address faulty ones.

However, detailed behavior models are seldom available in the first phases of system design, when complex processes, with hundreds of variables are considered, and simpler models have to be used. This article introduces the structural model of a system, which 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 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. Thus, structural properties are true almost everywhere in the system parameter space.

In spite of their simplicity, structural models can provide many useful information for FDI and FTC 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 – or cannot – be tolerated through reconfiguration.

In this article, structural properties of interest are connected with the design of FDI and FTC systems. Namely, one is interested in:

- the identification of the monitorable part of the system, i.e. the subset of the system components whose faults can be detected and isolated,
- the possibility to design residuals which meet some specific FDI requirements, namely which are robust (i.e. insensitive to disturbances and uncertainties), and structured (i.e. sensitive to certain faults and insensitive to others),
- the existence of reconfiguration possibilities in order to estimate (resp. to control)

some variables of interest in case of sensor, actuator or system component failures.

Answers to these questions are provided by the analysis of the system structural graph and its canonical decomposition. In order to introduce the canonical decomposition, matchings on a bipartite graph are first presented, and their interpretation is given, introducing the idea of causality which provides the bipartite graph with an orientation. Then the canonical decomposition of the system structural graph is presented, and structural observability and controllability issues are discussed. The design of FDI systems is addressed by the determination of robust and structured residuals, which can be designed for those subsystems in which some redundancy is present. Finally, fault tolerance issues consider the possibility to reconfigure the system in case of component failures, which rests on the permanence of the observability and controllability properties of the part of the system that has not failed.

## 2. Structural Model

### 2.1. Structure as a Bipartite Graph

The behavior model of a system is a pair  $[C, Z]$  where  $Z = \{z_1, z_2, \dots, z_N\}$  is a set of variables and parameters, and  $C = \{c_1, c_2, \dots, c_M\}$  is a set of constraints. Consider, for example, state space models like

$$\dot{x}(t) = f(x(t), u(t), \theta), \quad (1)$$

$$y(t) = g(x(t), u(t), \theta), \quad (2)$$

where  $x(t) \in R^n$  is the system state,  $u(t) \in R^m$  and  $y(t) \in R^p$  are respectively the system inputs and outputs, and  $\theta \in R^q$  is some parameter vector. Since the distinction between vectors and sets of components is clear from the context, no special notation will be introduced to distinguish them. Thus, in (1) (2), the sets of variables and constraints are

$$\begin{aligned} Z &= x \cup u \cup y \cup \theta \\ C &= f \cup g, \end{aligned} \quad (3)$$

where  $f$  stands for the set of differential constraints

$$\dot{x}_i(t) - f_i(x(t), u(t), \theta) = 0, \quad i = 1, \dots, n$$

and  $g$  stands for the measurement constraints

$$y_j(t) - g_j(x(t), u(t), \theta) = 0, \quad j = 1, \dots, p.$$

A popular structural representation of the behavior model (1), (2) uses a directed graph (digraph), whose set of vertices is the set of the input, output and state variables and

whose edges are defined by the following rules:

- an edge exists from vertex  $x_k$  (resp. from vertex  $u_l$ ) to vertex  $x_i$  if and only if the state variable  $x_k$  (resp. the input variable  $u_l$ ) really occurs in function  $f_i$  (i.e.  $\frac{\partial f_i}{\partial x_k}$  - resp.  $\frac{\partial f_i}{\partial u_l}$  - is not identically zero)
- an edge exists from vertex  $x_k$  to vertex  $y_j$  if and only if the state variable  $x_k$  really occurs in the function  $g_j$ .

The digraph representation is an abstraction of the behavior model since edges can be interpreted as “mutual influences” between variables. Indeed, an edge from  $x_k$  (resp. from  $u_l$ ) to  $x_i$  means that the time evolution of the derivative  $\dot{x}_i(t)$  depends on the time evolution of  $x_k(t)$  (resp.  $u_l(t)$ ). Similarly, an edge from  $x_k$  to  $y_j$  means that the time evolution of the output  $y_j(t)$  depends on the time evolution of the state variable  $x_k(t)$ .

For illustration, consider the following simple example

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, u) = x_2 \\ \dot{x}_2 &= f_2(x_1, x_2, u) = ax_2 + bu \\ y &= g(x_1, x_2, u) = x_1.\end{aligned}\tag{4}$$

The associated digraph is given by Figure 1.

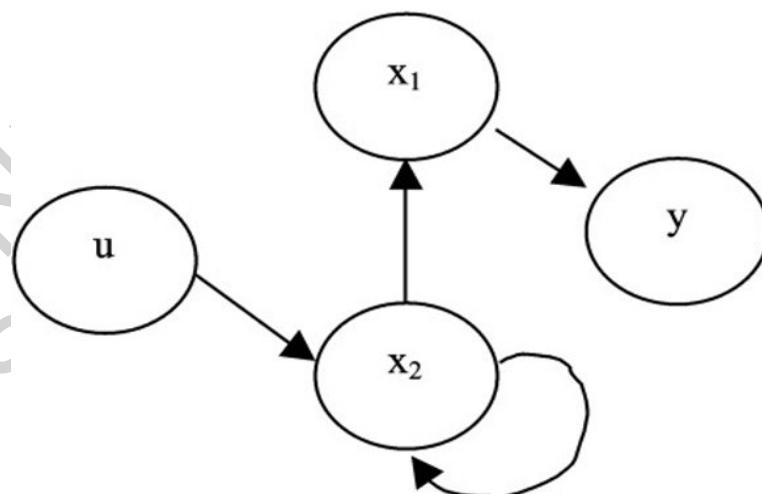


Figure 1: The digraph of system (4)

Alternatively, the structure of (1), (2) can be represented by a bipartite graph with the two sets of vertices  $C$  and  $Z$ , and edges defined by the following rule.

- an edge exists between vertex  $c_i \in C$  and vertex  $z_j \in Z$  if and only if the variable

$z_j$  really appears in the constraint  $c_i$  (whatever it is a differential or a measurement constraint).

Figure 2 gives the bipartite graph representation of the example (4), where bars represent constraints and circles represent variables.

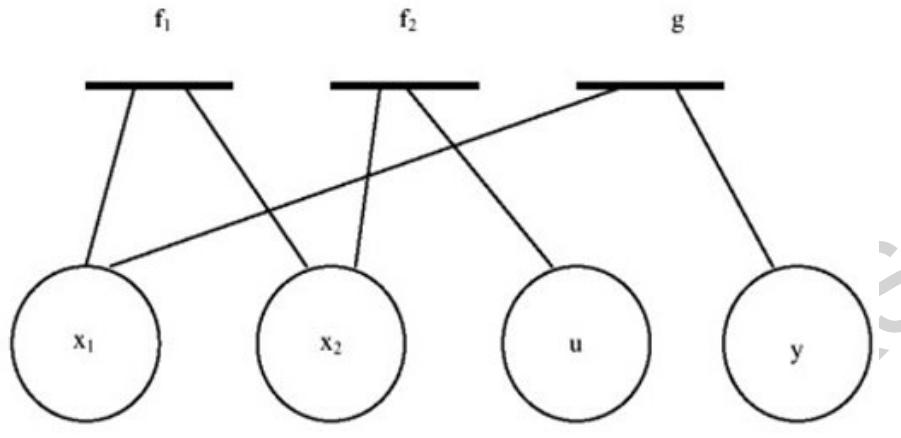


Figure 2: The bipartite graph of system (4)

Note that parameters can be taken into account, by considering them as variables which have constant (known or unknown) values. In the following this will be done implicitly (thus the set  $\Theta$  will no longer appear). Note also that the bipartite graph is a non oriented graph, which can be interpreted as: all the variables and parameters connected with a given constraint vertex have to satisfy the equation this vertex represents, namely differential equations for the  $f$  vertices and measurement equations for the  $g$  vertices. This graph allows to represent the structure of models more general than (1), (2) since algebraic constraints (different from the measurement constraints) might also exist in the system model. Let

$$\begin{aligned} Z &= x_a \cup x_d \cup u \cup y \\ C &= f \cup g \cup h, \end{aligned} \tag{5}$$

where  $x_a$  is the set of variables which appear only in some algebraic constraints  $h$ , and  $x_d$  are variables whose derivative obeys some differential constraints  $f$ . The system model is

$$\dot{x}_d = f(x_d, x_a, u) \tag{6}$$

$$0 = h(x_d, x_a, u) \tag{7}$$

$$y = g(x_d, x_a, u). \tag{8}$$

Note that it is possible to define an extra set of variables  $\dot{x}_d$  and an extra set of

constraints  $d = \{d_i, i = 1, \dots, n\}$ , where constraint  $d_i$  is defined by

$$\dot{x}_i(t) - \frac{d}{dt} x_i(t) = 0 \quad (9)$$

so that the system is

$$\begin{aligned} Z &= x_a \cup x_d \cup \dot{x}_d \cup x \cup y \\ C &= f \cup g \cup h \cup d. \end{aligned} \quad (10)$$

Therefore, all the constraints (6), (7) and (8) are algebraic, and the differential constraints (9) are all gathered in  $d$ . For system (4) such as representation is given in Figure 3.

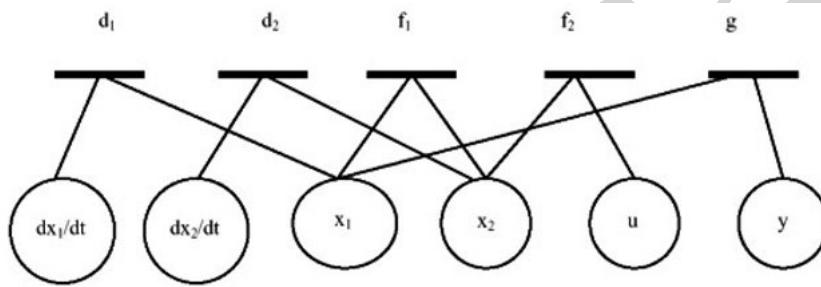


Figure 3: Extended bipartite graph of system (4)

In the following, bipartite graphs will be used for the representation of the system structure.

**Definition 1 (structural model):** The structural model (or the structure) of the system  $[C, Z]$  is a bipartite graph  $\langle C, Z, E \rangle$  where  $E \subset C \times Z$  is the set of edges defined by:

$(c_i, z_j) \in E$  iff the variable  $z_j$  appears in the constraint  $c_i$

## 2.2. Subsystems

In this subsection, subsets of the constraints are considered. Let  $2^C$  (resp.  $2^Z$ ) be the collection of all subsets of  $C$  (resp. of all the subsets of  $Z$ ), and let  $\langle C, Z, E \rangle$  be the structure of the system  $[C, Z]$ . Let  $Q$  be the mapping which associates with any subset of constraints  $\phi$ , the variables which intervene in at least one of them:

$$\begin{aligned} Q: 2^C &\rightarrow 2^Z \\ \phi \mapsto Q(\phi) &= \{z \in Z; \exists c \in \phi \text{ s.t. } (c, z) \in E\}. \end{aligned} \quad (11)$$

**Definition 2 (subsystem):** A subsystem is a pair  $[\phi, Q(\phi)]$ , where  $\phi \in 2^C$ . The subgraph that is related with subsystem  $[\phi, Q(\phi)]$  is its structure.

In this definition, a subsystem is any subset of the system constraints  $\phi$  along with the related variables  $Q(\phi) \in Z$ . There are no specific requirements to the choice of the elements in  $\phi$ , and  $2^C$  contains all possible subsystems. Of course, only some of them are of interest in applications (e.g. the monitorable subsystem).

### 2.3. Structural Properties

The structural model of a system is an abstraction of its behavior model. Two systems which have the same structure are said to be structurally equivalent. Since structural properties are properties of the structural graph, they are obviously shared by all the systems which have the same structure. In particular, systems which only differ by the value of their parameters are structurally equivalent, thus making structural properties independent of the values of the system parameters.

Of course, actual system properties may differ from structural ones, as can be seen from the following simple example. Let

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} a(\theta) & b(\theta) \\ c(\theta) & d(\theta) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

be the model of a system where  $y_1$  and  $y_2$  are known,  $\theta \in R^q$  is some parameter vector, and the observability of the unknowns  $x_1, x_2$  is investigated. The observability condition is that the matrix is invertible. The structural condition is that no row (no column) of this matrix contains only zeros. This is of course necessary, but not sufficient, since the determinant  $\Delta(\theta) = a(\theta)d(\theta) - b(\theta)c(\theta)$  might be zero, so that the property would not hold for the actual system although the structural property holds. Two cases can be distinguished:

- 1) in the first case, parameters  $\theta$  **always** satisfy the relation  $\Delta(\theta) = 0$  and thus the structural property is **never** translated into an actual property. This is excluded in structural analysis. First, an algebraic relation like  $\Delta(\theta) = 0$  is always supposed to define a manifold of dimension  $q-1$ , which means that it cannot be satisfied by any  $\theta \in R^q$  or, in other words, that it does not boil down to  $0=0$ . Second, the parameters are always supposed to be independent, which means that they live in the whole space  $R^q$ . Indeed, if this were not the case, equation  $\Delta(\theta) = 0$  should have been included in the system model.
- 2) in the second case, the parameters  $\theta$  of the system under investigation satisfy the relation  $\Delta(\theta) = 0$ , and thus the structural property is not translated into an actual property **for that particular system**. Structural analysis however provides interesting conclusions, since under mild assumptions about functions  $a, b, c, d$  there always exists a parameter vector  $\theta'$  in the neighborhood of  $\theta$  for which the actual property coincides with the structural one.

In conclusion, actual properties are only potential when structural properties are satisfied. They can certainly not be true when structural properties are not satisfied. In other words, structural properties are properties which hold for actual systems almost everywhere in the space of their independent parameters.

### 3. Matching on a Bipartite Graph

The basic tool for structural analysis is the concept of matching on a bipartite graph, which is introduced in this section. In loose terms, a matching is a causal assignment which associates some system variables with the system constraints from which they can be calculated. Variables which cannot be matched cannot be calculated. Variables which can be matched in several ways can be calculated by different (redundant) means, thus providing a means for fault detection and a possibility for reconfiguration.

- 
- 
- 

TO ACCESS ALL THE 27 PAGES OF THIS CHAPTER,  
[Click here](#)

### Bibliography

- Carpentier, T., Litwak, R. and Cassar, J.-Ph. (1997). Criteria for the evaluation of FDI systems. Application to sensors location, in Proc. of IFAC Safeprocess' 97, 1083-1088. [This paper shows how structural analysis can be used for selecting a set of sensors such that the system is monitorable]
- Declerck Ph. and Staroswiecki, M. (1991). Characterization of the canonical components of a structural graph for fault detection in large scale industrial plants, in Proc. of European Control Conference, ECC'91, 298-303. [This paper introduces the relation between the canonical decomposition and the monitorable part of the system].
- Dulmage, A.L. and Mendelsohn, N.S. (1958). Covering of bi-partite graphs, Canada J.Math., 10:517-534. [This paper gives the basic concepts for the canonical decomposition of bi-partite graphs].
- Dulmage, A.L. and Mendelsohn, N.S.(1959). A structure theory of bi-partite graphs of finite exterior dimension. Trans. of Royal Soc. Canada, Section III, 53:1-13. [This paper introduces the canonical decomposition of bi-partite graphs].
- Gertler, J.J. and Singer D. (1990). A new structural framework for parity space equation based failure detection and isolation, Automatica, 26:381-388. [This paper emphasizes analytical redundancy based approaches and structured residuals for FDI].
- Harary, F.(1962). A graph theoretic approach to matrix inversion by partitioning, Numer. Math., 4:128-135. [This paper introduces graph theoretic approaches for the solution of large scale numerical problems].
- Izadi-Zamanabadi, R., Amann, P., Blanke, M., Cocquempot, V., Gissinger, G.L., Kerrigan E.C., Lootsma, T.F., Perronne, J.M. and Schreier, G. (2000). Fault Detection and Reconfigurable Control for a Ship, , in control of Complex Systems, K.J. Aström, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger, T. Sanz ed., Springer 2000. [This chapter presents different approaches, including the structural analysis approach, to FDI and FTC of a ship propulsion benchmark].
- Iwasaki, Y. and Simon, H.A. (1986). Causality in device behvaiour, Artificial Intelligence, 29:3-32. [This

paper introduces causality in systems of equations, which are very close to the matchings in bi-partite graphs].

Meyer, M., Le Lann, J.-M., Koehret, B. and Enjalbert M. (1994). Optimal selection of sensor location on a complex plant using a graph oriented approach, Computer Chemical Eng., 18: S535-S540. [This paper introduces a graph approach for the selection of sensors in large plants].

Murota, K. (1987). Systems analysis by graphs and matroids. Structural solvability and controllability, Springer Verlag. [This book gives a detailed treatment of graph based approaches to system analysis].

Staroswiecki, M. and Gehin, A.-L. (2001). From Control to Supervision, Annual Reviews in Control, 25: 1-11. [This paper sets the basic problems in fault tolerant control].

Staroswiecki, M., Cassar, J.Ph. and Declerck, Ph. (2000). A structural framework for the design of FDI in large scale industrial plants, in Issues of Fault Diagnosis for Dynamic Systems. ed. R. Patton, P. Frank and R. Clark, Springer Verlag. [This chapter presents the structural analysis approach for the design of FDI systems].

Cordier, M-O., Dague, Ph., Lévy, F., Dumas, M., Montmain, J., Staroswiecki, M. and Travé-Massuyès, L.(2000). AI and automatic control approaches of model-based diagnosis: links and underlying hypotheses, In Proc. of IFAC Safeprocess' 2000, Vol.1.: 274-279. [This paper discusses the basic assumptions used in AI and Control for the design and exploitation of structured residuals].

Gehin, A.-L., Assas, M. and Staroswiecki, M. (2000). Structural Analysis of System Reconfigurability, In Proc. of IFAC Safeprocess' 2000, Vol.1: 292-297. [This paper presents the graph based analysis of reconfiguration possibilities when faults occur].

Steward, D.V. (1962). On an approach to techniques for the analysis of the structure of large systems of equations, SIAM Review, 4: 321-342. [One of the earliest approaches to structural analysis of large sets of equations].

Tagina, M., Cassar, J.-Ph., Dauphin-Tanguy, G. and Staroswiecki, M. (1996). Bond-Graph models for direct generation of formal fault detection systems, Systems Analysis Modelling and Simulation J., 23: 1-17. [In this paper, structural analysis is performed on the basis of the system Bond-Graph model].

### Biographical Sketch

**Marcel Staroswiecki** was born in Melitopol (Ukraina) in 1945. He obtained the Engineering Degree from the Ecole Nationale Supérieure d'Ingénieurs des Arts et Métiers (with a silver medal distinction), in 1968. He then obtained a PhD in Automatic Control in 1970, and the french « Doctor es Sciences » degree in Physical Sciences in 1979.

He joined the University of Lille I as an Assistant Professor in 1969 and he became full Professor in 1983. He currently teaches at the University's Engineering School (Ecole Polytechnique Universitaire de Lille), and he is the Director of the Laboratoire d'Automatique et d'Informatique Industrielle de Lille (a shared laboratory between Ecole Centrale de Lille, Université des Sciences et Technologies de Lille and CNRS).

His research interests are on Fault Detection, Isolation and Recovery (FDIR) algorithms for complex systems. He heads a research group in which both the model based and the pattern recognition based approaches are developed. A special attention is paid to the implementation of FDIR procedures, especially within the frame of intelligent instruments and components. Prof. Staroswiecki is a member of two IFAC Technical Committees : Safeprocess and Intelligent Components and Instruments.

His activity has also been intensively dedicated to technology transfer, especially in direction of SME's. He created and was the director of the Lille University Technology Transfer Center in the field of Automatic Control from 1982 to 1994, which performed more than 100 joint research contracts between Lille University and industrial companies. For this activity, he was honored by the 'Grand Prix Edmond Faucheur de la Société des Industriels du Nord de la France'.

He was a 'chargé de mission' at the French Ministry of Research in the field of Automatic Control and Robotics from december 1993 to december 1996. He there impulsed different policies for the development of the research and of the doctoral activity in french laboratories.

Prof. Staroswiecki obtained the french distinction of « Chevalier de l'Ordre des Palmes Académiques » in 1990 and became « Officier dans l'ordre des Palmes Académiques » in 1995.

UNESCO – EOLSS  
SAMPLE CHAPTERS