5. Structural analysis

This chapter uses the structure graph to describe the direct interactions among the signals. This graph is used to analyse the redundancies which can be exploited for fault diagnosis and control reconfiguration.

5.1 Introduction

This chapter investigates the structural properties of dynamical system by analysing its structural model. Remember that the structural model of a system is an abstraction of its behaviour 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 *bi-partite 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 behaviour.

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

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 – or cannot – be tolerated through reconfiguration.

In this chapter, structural properties of interest are

- 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 fault diagnosis 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 (respectively 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 bi-partite graph are first presented, and their interpretation is given, introducing the idea of causality which provides the bi-partite 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 fault diagnosis 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 controllabilty properties of the non failed part of the system.

5.2 Structural model

5.2.1 Structure as a bi-partite graph

This section introduces the structural model of a system as a bi-partite graph which represents the links between a set of variables and a set of constraints. It is an abstraction of the behaviour model, because it merely describes which variables are connected by which constraints, but it does not say how these constraints look like. Hence, the structural model presents the basic features and properties of a system that are independent of its paramters.

The behaviour model of a system is defined by a pair $(\mathcal{C}, \mathcal{Z})$ where $\mathcal{Z} = \{z_1, z_2, ..., z_N\}$ is a set of variables and parameters, and $\mathcal{C} = \{c_1, c_2, ..., c_M\}$ is a set of constraints. According to the granularity of the variables (quantitative, qualitative, fuzzy) and of time (continuous, discrete), the constraints may be expressed in several different forms as algebraic and differential equations, difference equations, rules, etc.

Consider, for example, state-space models like

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta) \tag{5.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta), \tag{5.2}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the system state, $\boldsymbol{u}(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are respectively the system input and output, and $\theta \in \mathbb{R}^q$ 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 (5.1), (5.2), the sets of variables and constraints are

$$\mathcal{Z} = \boldsymbol{x} \cup \boldsymbol{u} \cup \boldsymbol{y}$$
$$\mathcal{C} = \boldsymbol{g} \cup \boldsymbol{h},$$

where $\boldsymbol{g}$ stands for the set of differential constraints

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

and $\boldsymbol{h}$ stands for the measurement constraints

$$y_j(t) - h_j(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta) = 0, \quad j = 1, ..., p\,.$$

Note that parameters can be considered as variables which have constant (known or unknown) values, when such a representation is needed. The set of variables is then $\mathcal{Z} = \boldsymbol{x} \cup \boldsymbol{u} \cup \boldsymbol{y} \cup \theta$, and the set of the constraints contains the differential equations $\dot{\theta}(t) = \mathbf{0}$. In the following this is implicit (thus vector θ does not appear, since it is part of the system state vector $\boldsymbol{x}$ and therefore, defined by the initial conditions).

A popular structural representation of the behaviour model (5.1), (5.2) uses a directed graph (digraph).

Definition 5.1 (Digraph)
The digraph associated with system (5.1), (5.2) is a graph whose set of vertices corresponds to the set of the inputs u_l, outputs y_j and state variables x_i and whose edges are defined by the following rules:

- *An edge exists from vertex x_k (respectively from vertex u_l) to vertex x_i if and only if the state variable x_k (respectively the input variable u_l) really occurs in the function g_i (i.e. $\frac{\partial g_i}{\partial x_k}$ – respectively $\frac{\partial g_i}{\partial u_l}$ – 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 h_j.*

The digraph representation is a structural abstraction of the behaviour model where edges can be interpreted as "mutual influences" between variables. Indeed, an edge from x_k (respectively from u_l) to x_i means that the time evolution of the derivative $\dot{x}_i(t)$ depends on the time evolution of $x_k(t)$ (respectively $u_l(t)$). 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 $x_k(t)$.

Example 5.1 *Digraph of a linear system*
Consider the linear system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ where the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are given by:

$$\boldsymbol{A} = \begin{pmatrix} 0 & a \\ b & c \end{pmatrix}, \ \boldsymbol{B} = \begin{pmatrix} 0 \\ d \end{pmatrix}.$$

Their structures are respectively

$$S_A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad S_B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and the digraph associated with the system is given by Fig. 5.1 where $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.
□

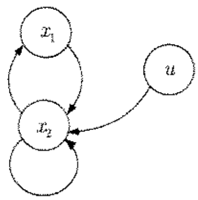


Fig. 5.1. Structure of the linear system

Alternatively, the structure of (5.1), (5.2) can be represented by a bi-partite graph. A graph is bi-partite if its vertices can be separated into two disjoint sets $\mathcal{C}$ and $\mathcal{Z}$ in such a way that every edge has one endpoint in $\mathcal{C}$ and the other one in $\mathcal{Z}$.

Definition 5.2 (Bi-partite graph)
The bi-partite graph associated with system (5.1), (5.2) is a graph with the two sets of vertices $\mathcal{C}$ and $\mathcal{Z}$, and edges defined by the following rule:

- *An edge exists between vertex $c_i \in \mathcal{C}$ and vertex $z_j \in \mathcal{Z}$ if and only if the variable z_j really appears in the constraint c_i (irrespectively of wheter it is a differential or a measurement constraint).*

Note that the bi-partite graph is an undirected graph, which can be interpreted as follows: All the variables and parameters connected with a given constraint vertex have to satisfy the equation this vertex represents, namely differential equations for the $\boldsymbol{g}$-vertices and measurement equations for the $\boldsymbol{h}$-vertices. This graph allows to represent the structure of models more general than (5.1), (5.2) since algebraic constraints (different from the measurement constraints) might also exist in the system model. Let

$$\begin{aligned} \mathcal{Z} &= \boldsymbol{x}_a \cup \boldsymbol{x}_d \cup \boldsymbol{u} \cup \boldsymbol{y} \\ \mathcal{C} &= \boldsymbol{g} \cup \boldsymbol{h} \cup \boldsymbol{m}, \end{aligned}$$

where $\boldsymbol{x}_a$ is the set of variables which appear only algebraically, and $\boldsymbol{x}_d$ are variables whose derivative obeys some differential constraints $\boldsymbol{g}$. The system model is

$$\dot{\boldsymbol{x}}_d = \boldsymbol{g}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}) \tag{5.3}$$

$$\mathbf{0} = \boldsymbol{m}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}) \quad (5.4)$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}). \quad (5.5)$$

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

$$i = 1, ..., n \quad \dot{x}_i(t) - \frac{d}{dt}x_i(t) = 0, \quad (5.6)$$

so that the system is

$$\begin{aligned} \mathcal{Z} &= \boldsymbol{x}_a \cup \boldsymbol{x}_d \cup \dot{\boldsymbol{x}}_d \cup \boldsymbol{u} \cup \boldsymbol{y} \\ C &= \boldsymbol{g} \cup \boldsymbol{h} \cup \boldsymbol{m} \cup \frac{d}{dt}, \end{aligned}$$

where $\frac{d}{dt}$ stands for the differential constraints (5.6) and all the constraints (5.3), (5.4) and (5.5) are algebraic. Indeed the behaviour model of a dynamical system links present and past values of its variables (for discrete time systems) or variables and their time derivatives up to a certain order (for continuous-time systems). Giving two variables the names $\boldsymbol{x}(t)$ and $\dot{\boldsymbol{x}}(t)$ does not guarantee that the second one is the time derivative of the first one. This is only true thanks to the analyst's interpretation, and this fact has to be represented, for automatic treatment, by separate constraints like (5.6).

In the sequel, bi-partite graphs will be used for the representation of the system structure.

Definition 5.3 (Structural model)
The structural model (or the structure) of the system $(\mathcal{C}, \mathcal{Z})$ is a bi-partite graph $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ where $\mathcal{E} \subset \mathcal{C} \times \mathcal{Z}$ is the set of edges defined by:

$$(c_i, z_j) \in \mathcal{E} \text{ if the variable } z_j \text{ appears in the constraint } c_i.$$

In the following figures, the vertices of $\mathcal{Z}$ will be represented by circles while the vertices of C will be represented by bars. Note that the edges are not oriented. The incidence matrix of the bi-partite graph is the matrix whose rows and columns represent the set of constraints or variables, respectively. Every edge $(c_i, z_j) \in \mathcal{E}$ is represented by a "1" in the intersection of row c_i and column z_j.

Example 5.2 *Bi-partite graph of a linear system*
The behaviour model of the previous linear system is described by four constraints $\{c_1, c_2, c_3, c_4\}$ which apply to five variables $\{x_1, x_2, \dot{x}_1, \dot{x}_2, u\}$

$$\begin{aligned} c_1 &: \quad \dot{x}_1 = \frac{dx_1}{dt} \\ c_2 &: \quad \dot{x}_1 = ax_2 \\ c_3 &: \quad \dot{x}_2 = \frac{dx_2}{dt} \\ c_4 &: \quad \dot{x}_2 = bx_1 + cx_2 + du \end{aligned}$$

and the structure graph is given by the incidence matrix

↗	u	x_1	x_2	$\dot{x}_1$	$\dot{x}_2$
c_1		1		1	
c_2			1	1	
c_3			1		1
c_4	1	1	1		1

leading to the bi-partite graph depicted in Fig. 5.2. □

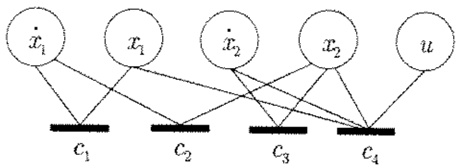


Fig. 5.2. Bi-partite graph of the linear system

Example 5.3 *The tank system*
Consider a tank system where the inflow is controlled via a level sensor and an electric pump and the outflow is realised through an output pipe (Fig.5.3).

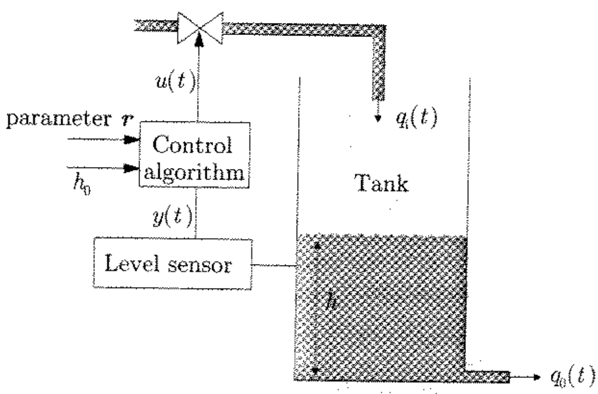


Fig. 5.3. Single tank system

The system consists of the components {tank, input valve, output pipe, level sensor, level control algorithm}. A continuous-variable continuous-time model is given by the following constraints:

$$\begin{array}{ll} \text{Tank } c_1: & \dot{h}(t) = q_i(t) - q_o(t) \\ \text{Input valve } c_2: & q_i(t) = \alpha u(t) \\ \text{Output pipe } c_3: & q_0(t) = k\sqrt{h(t)} \\ \text{Level sensor } c_4: & y(t) = h(t) \\ \text{Control algorithm } c_5: & u(t) = \begin{cases} 1 \text{ if } y(t) \leq h_0 - r \\ 0 \text{ else} \end{cases} \end{array} \tag{5.7}$$

u is the control variable, y is the sensor output, h_0 is the given set-point, and r and k are given parameters. h denotes the liquid level, q_i and q_o the flow into or out of the tank. α is a valve constant. Each component introduces one constraint. The extra constraint

$$c_6: \dot{h}(t) = \frac{dh(t)}{dt}$$

expresses the fact that $\dot{h}(t)$ is the derivative of the level $h(t)$. Applying the definition to the behaviour model (5.7) of the tank system (without controller) leads to the following incidence matrix:

	Input/Output		Internal variables			
↗	$u(t)$	$y(t)$	$h(t)$	$\dot{h}(t)$	$q_i(t)$	$q_o(t)$
c_1				1	1	1
c_2	1				1	
c_3			1			1
c_4		1	1			
c_6			1	1		

The structure graph corresponding to this incidence matrix is shown in Fig. 5.4. Every column of the matrix correspond to a circle-vertex and every row to a bar-vertex.

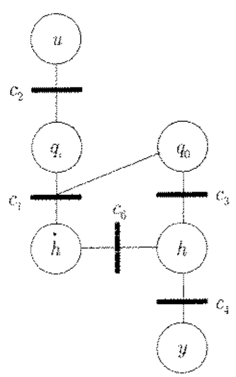


Fig. 5.4. Structure graph of the single tank system

If the controller is introduced, the graph is extended by a new bar-vertex c_5 and two new circle-vertices for h_0 and r. Furthermore, if the parameter k appearing in constraint c_3 is considered now as an important variable (rather than a fixed given parameter) a circle-vertex is introduced for k and linked with c_3. The incidence matrix becomes

	Input/Output		Parameters			Internal variables			
$\nearrow$	$u(t)$	$y(t)$	h_0	r	k	$h(t)$	$\dot{h}(t)$	$q_i(t)$	$q_o(t)$
c_1							1	1	1
c_2	1							1	
c_3					1	1			1
c_4		1				1			
c_5	1	1	1	1					
c_6						1	1		

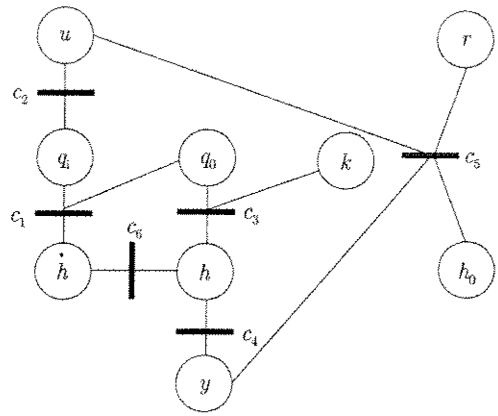


Fig. 5.5. Structure graph of the controlled tank

For simplicity, only the "ones" appear, empty boxes are "zero". Figure 5.5 shows the extended graph. □

5.2.2 Subsystems

Instead of considering the whole set of the constraints which describe the behaviour model of a system, it may sometimes be convenient to consider only subsets of them. A subsystem is defined by the set of contraints that it includes and the set of variables that occur in these contraints. This subsection introduces the vocabulary connected with subsets of the constraints. Let $2^{\mathcal{C}}$ (respectively $2^{\mathcal{Z}}$) be the collection of all the subsets of $\mathcal{C}$ (respectively of all the subsets of $\mathcal{Z}$), and let $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ be the structure of the system $(\mathcal{C}, \mathcal{Z})$.

Let Q be a mapping between a set of contraints and the set of variables used in these constraints:

$$\begin{aligned} Q: \quad & 2^{\mathcal{C}} \to 2^{\mathcal{Z}} \\ & \phi \mapsto Q(\phi) = \{z \in \mathcal{Z};\ \exists c \in \phi \quad \text{s.t.} \quad (c, z) \in \mathcal{E}\} \end{aligned} \tag{5.8}$$

Q associates with any subset of constraints ϕ, the subset of those variables which intervene in at least one of them. Correspondingly, R associates a set of variables with a set of constraints where these variables appear:

$$\begin{aligned} R: \quad & 2^{\mathcal{Z}} \to 2^{\mathcal{C}} \\ & \xi \mapsto R(\xi) = \{c \in \mathcal{C};\ \exists z \in \xi \quad \text{s.t.} \quad (c, z) \in \mathcal{E}\}. \end{aligned}$$

Example 5.4 *Q and R mappings*
Consider again the incidence matrix associated with the simple single tank.

$\nearrow$	h	$\dot{h}$	q_i	q_o	u	y
c_1		1	1	1		
c_2			1		1	
c_3	1			1		
c_4	1					1
c_5					1	1
c_6	1	1				

Examples for the Q and R mappings are:

$$\begin{aligned} Q(\{c_1, c_3\}) &= \left\{h, \dot{h}, q_i, q_o\right\} \\ Q(\{c_5\}) &= \{u, y\} \\ R(\{q_i, q_o\} &= \{c_1, c_2, c_3\}. \quad \square \end{aligned}$$

Definition 5.4 (Subsystem)
A subsystem is a pair $(\phi, Q(\phi))$, where $\phi \in 2^{\mathcal{C}}$. The sub-graph that is related with subsystem $(\phi, Q(\phi))$ is its structure.

In this definition, a subsystem is any subset of the system constraints ϕ along with the related variables $Q(\phi) \in \mathcal{Z}$. For example, in the system above, $\left(\{c_1, c_3\}, \left\{h, \dot{h}, q_i, q_o\right\}\right)$ is a subsystem, and its structure is the subgraph

$\nearrow$	h	$\dot{h}$	q_i	q_o
c_1		1	1	1
c_3	1			1

There are no specific requirements on the choice of the elements in ϕ, and 2^C contains all possible subsystems. Of course, only some of them are of interest in applications:

- First, subsystems can be associated with some physical interpretation. Complex systems are often decomposed into subsystems which have a physical or a functional meaning, e.g. the boiler in a steam generator, the instrumentation scheme in a closed-loop control system, etc. These subsystems are associated with subsets of constraints in the system model, so that the fault of one or several subsystem component(s) results in some of these constraints being changed.
- Second, subsystems can be associated with special properties. For example, fault diagnosis is possible only for subsystems which exhibit redundancy properties.

5.2.3 Structural properties

The structural model of a system is an abstraction of its behaviour 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 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.

From structural equivalence considerations, it can be seen that the structure of a system is indeed independent of the form under which the constraints are expressed. For example, suppose that the level sensor in the single tank system does not provide an analog output but a quantised one, its operation being described by the following table, where α, β, γ are given constants associated with the sensor.

h	$\in [0, \alpha[$	$\in [\alpha, \beta[$	$\in [\beta, \gamma[$	$\geq \gamma$
y	empty	low	medium	high

It can easily be understood that the structure of the system is exactly the same using the analog or the symbolic sensor.

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

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

be the model of a system where y_1 and y_2 are known, $\boldsymbol{\theta} \in \mathbb{R}^q$ 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 necessary, but not sufficient, since the determinant $\Delta(\boldsymbol{\theta}) = a(\boldsymbol{\theta})\, d(\boldsymbol{\theta}) - b(\boldsymbol{\theta})\, c(\boldsymbol{\theta})$ 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 $\boldsymbol{\theta}$ *always* satisfy the relation $\Delta(\boldsymbol{\theta}) = 0$ and thus the structural property is *never* translated into an actual property. This is excluded in structural analysis. First, an algebraic relation like $\Delta(\boldsymbol{\theta}) = 0$ is always supposed to define a manifold of dimension at most $q - 1$, which means that it cannot be satisfied by any $\boldsymbol{\theta} \in \mathbb{R}^q$ 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 $\mathbb{R}^q$. Indeed, if this were not the case, equation $\Delta(\boldsymbol{\theta}) = 0$ should have been included in the system model.
2. In the second case, the parameters $\boldsymbol{\theta}$ *of the system under investigation* satisfy the relation $\Delta(\boldsymbol{\theta}) = 0$, 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 $\boldsymbol{\theta}'$ in the neighbourhood of $\boldsymbol{\theta}$ 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. It is extremly unlikely that the system under consideration has a parameter vector for which the structural properties do not hold.

5.2.4 Known and unknown variables

The system variables and parameters can be decomposed into known and unknown ones. System input and output are examples of known variables. Similarly, parameters which enter the model and have been previously identified are known. Known variables are available in real time and they can directly be used in fault diagnosis or fault-tolerant control algorithms. Unknown variables are not directly measured, though there might exist some way to compute their value from the values of known ones. In the tank example, the last four columns of the incidence matrix $\left\{h,\, \dot{h},\, q_i,\, q_o\right\}$ correspond to unknown variables, while the first five ones correspond to known variables and parameters $\{u,\, y,\, h_0,\, r,\, k\}$.

Following that decomposition, the set of the variables is partitioned into $\mathcal{Z} = \mathcal{K} \cup \mathcal{X}$, where $\mathcal{K}$ is the subset of the known variables and parameters and $\mathcal{X}$ is the subset of the unknown ones. Similarly, the set of constraints is partitioned into $\mathcal{C} = \mathcal{C}_{\mathcal{K}} \cup \mathcal{C}_{\mathcal{X}}$, where $\mathcal{C}_{\mathcal{K}}$ is the subset of those constraints which link only known variables and $\mathcal{C}_{\mathcal{X}}$ includes those constraints in which at least one unknown variables appears. $\mathcal{C}_{\mathcal{K}}$ is the largest subset of constraints such that $Q(\mathcal{C}_{\mathcal{K}}) \subseteq \mathcal{K}$. It can be noticed that the relations which define control algorithms belong to $\mathcal{C}_{\mathcal{K}}$ since they introduce constraints between the sensor output, the control objectives (set-points, tracking references, final states) and the control output, which are all known variables.

By following the decomposition of $\mathcal{Z}$ and $\mathcal{C}$, the graph $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ can be decomposed into two sub-graphs which correspond to the two subsystems $(\mathcal{C}_{\mathcal{K}}, Q(\mathcal{C}_{\mathcal{K}}))$ and $(\mathcal{C}_{\mathcal{X}}, \mathcal{Z})$. The behaviour model of the subsystem $(\mathcal{C}_{\mathcal{K}}, Q(\mathcal{C}_{\mathcal{K}}))$ involves only known variables. In some further developments, it will be of interest to focus on the subsystem $(\mathcal{C}_{\mathcal{X}}, \mathcal{Z})$ which is also called the *reduced structure graph.* This graph includes only those constraints that refer to at least one unknown variable $z_i \in \mathcal{X}$. Indeed, a fundamental question of fault diagnosis concerns the determination of unknown variables from known variables by means of constraints. The question whether this is possible or not does only depend on the structure of the subgraph $(\mathcal{C}_{\mathcal{X}}, \mathcal{X}, \mathcal{E}_{\mathcal{X}})$ that results from the reduced graph by deleting all known variables $z_i \in \mathcal{K}$ together with the corresponding edges.

Example 5.5 *Reduction of the structural graph of the tank system*
Consider the tank, whose structure graph is given in Fig. 5.5. Assume that only the input u and the output y are known signals and, furthermore, h_0, r and k are known parameters. Then the decomposition of the variable set

$$\mathcal{Z} = \{h, \dot{h}, q_i, q_o, u, y, h_0, r, k\}$$

into known and unknown variables yields the sets

$$\mathcal{K} = \{u, y, h_0, r, k\}$$

and

$$\mathcal{X} = \{h, \dot{h}, q_i, q_o\}.$$

By selecting all constraints whose variables are all in the set $\mathcal{K}$, the set $\mathcal{C}_{\mathcal{K}} = \{c_5\}$ is obtained. All other constraints comprise the set

$$\mathcal{C}_{\mathcal{X}} = \{c_1, c_2, c_3, c_4, c_6\}.$$

Obviously, $Q(\mathcal{C}_{\mathcal{K}}) = \mathcal{K}$ and

$$Q(\mathcal{C}_{\mathcal{X}}) = \{u, y, q_i, q_o, h, \dot{h}\}$$

holds. The bi-partite graph can be re-organised as follows:

	known					unknown			
$\nearrow$	u	y	h_0	r	k	h	$\dot{h}$	q_i	q_0
c_5	1	1	1	1					
c_1						1		1	1
c_2	1							1	
c_3					1	1			1
c_4		1				1			
c_6						1	1		

The known variables are in the left columns and the constraint that refers merely to known variables in the first row. The reduced structure graph which corresponds to the subsystem $\mathcal{C}_{\mathcal{X}}$, $\mathcal{Z}$ is given by the lower part of the incidence matrix. As the variables h_0 and r do not appear in this part of the matrix, their columns are deleted:

	known			unknown			
$\nearrow$	u	y	k	h	$\dot{h}$	q_i	q_0
c_1				1		1	1
c_2	1					1	
c_3			1	1			1
c_4		1		1			
c_6				1	1		

This reduced graph is shown in Fig. 5.6. After deleting the known variables the incidence matrix

$\nearrow$	h	$\dot{h}$	q_i	q_0
c_1	1		1	1
c_2			1	
c_3	1			1
c_4	1			
c_6	1	1		

and the graph depicted in Fig. 5.7 result. □

5.3 Matching on a bi-partite graph

The basic tool for structural analysis is the concept of matching on a bi-partite 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

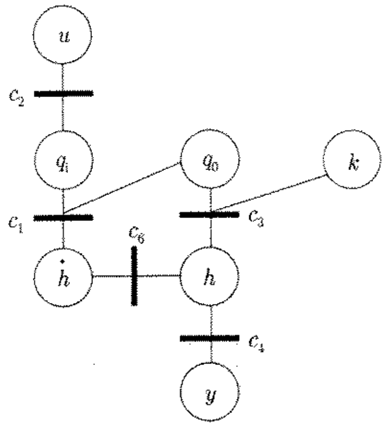


Fig. 5.6. Reduced structure graph of the tank system

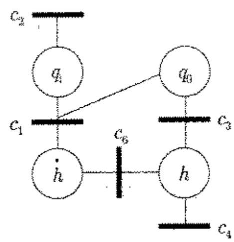


Fig. 5.7. Subgraph concerning the unknown variables of the tank system

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.

5.3.1 Definitions

Let $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ be a bi-partite graph, let $e \in \mathcal{E}$, $e = (\alpha, \beta)$ be an edge which links the constraint α and the variable β, and let $p_\mathcal{C}$ and $p_\mathcal{Z}$ be the two projections

$$\begin{aligned} p_\mathcal{C} \quad &: \quad \mathcal{E} \rightarrow \mathcal{C} \\ &\quad\;\; e \mapsto p_\mathcal{C}(e) = \alpha \\ p_\mathcal{Z} \quad &: \quad \mathcal{E} \rightarrow \mathcal{Z} \\ &\quad\;\; e \mapsto p_\mathcal{Z}(e) = \beta. \end{aligned}$$

In other words, the projection of the edge on the constraint set is $p_\mathcal{C}(e) = \alpha$ (the constraint node of the edge e) and the projection of the edge on the variable set is $p_\mathcal{Z}(e) = \beta$ (the variable node of the edge e).

Definition 5.5 (Matching)
A matching $\mathcal{M}$ is a subset of $\mathcal{E}$ such that the restrictions of p_C and $p_{\mathcal{Z}}$ to $\mathcal{M}$ are injective, i.e.

$$\forall e_1, e_2 \in \mathcal{M}: \; e_1 \neq e_2 \Rightarrow p_C(e_1) \neq p_C(e_2) \wedge p_{\mathcal{Z}}(e_1) \neq p_{\mathcal{Z}}(e_2).$$

This means that a matching is a subset of edges such that any two edges have no common node (neither in C nor in $\mathcal{Z}$). In general, different matchings can be defined on a given bi-partite graph, as illustrated by Fig. 5.8.

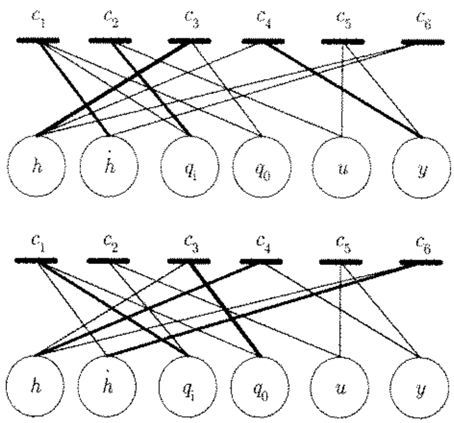


Fig. 5.8. Two possible matchings for the tank system: The edges $e \in \mathcal{M}$ are drawn by thick lines.

The set $\boldsymbol{M}$ of all matchings is a subset of $2^{\mathcal{E}}$, which is partially ordered by the set-inclusion order relation. Thus, maximal elements can be defined.

Definition 5.6 (Maximal matching)
A maximal matching is a matching $\mathcal{M}$ such that $\forall \mathcal{N} \in 2^{\mathcal{E}}$ with $\mathcal{M} \subset \mathcal{N}$, $\mathcal{N}$ is not a matching.

Thus, a maximal matching is a matching such that no edge can be added without violating the *no common node* property. Since the set of matchings $\boldsymbol{M}$ is only partially ordered, it follows that there is in general more than one maximal matching. Let $\boldsymbol{M}^* \subseteq \boldsymbol{M}$ be the set of maximal matchings. Extending the definition of the projections p_C and $p_{\mathcal{Z}}$ to sets of edges (instead of one single edge),

$$\begin{aligned}
\pi_C \quad &: \quad \boldsymbol{M} \to 2^C \\
&\quad \mathcal{M} \mapsto \pi_C(\mathcal{M}) = \{c \in C;\; \exists e \in \mathcal{M} \text{ such that } \; c = p_C(e)\} \\
\pi_{\mathcal{Z}} \quad &: \quad \boldsymbol{M} \to 2^{\mathcal{Z}} \\
&\quad \mathcal{M} \mapsto \pi_{\mathcal{Z}}(\mathcal{M}) = \{z \in \mathcal{Z};\; \exists e \in \mathcal{M} \text{ such that } \; z = p_{\mathcal{Z}}(e)\},
\end{aligned}$$

each matching $\mathcal{M}$ is associated with the subset $\pi_{\mathcal{C}}(\mathcal{M})$ of its matched constraints, and with the subset $\pi_{\mathcal{Z}}(\mathcal{M})$ of its matched variables. Since no more than one constraint (respectively no more than one variable) can be associated with each edge of a matching, it follows that any matching (and therefore also maximal matchings) satisfies the following property

$$\forall \mathcal{M} \in \boldsymbol{M}^* \qquad \begin{aligned} &\pi_{\mathcal{C}}(\mathcal{M}) \subseteq \mathcal{C} \\ &\pi_{\mathcal{Z}}(\mathcal{M}) \subseteq \mathcal{X}, \end{aligned}$$

from which it follows that the relation

$$|\mathcal{M}| \leq \min\{|\mathcal{C}|\,,\, |\mathcal{Z}|\}$$

holds, where the bars $|.|$ denote the cardinality of the sets (which means the number of elements). Matchings for which the equality sign holds are called complete matchings.

Definition 5.7 (Complete matching)
A matching is called complete with respect to $\mathcal{C}$ if $|\mathcal{M}| = |\mathcal{C}|$ holds. A matching is called complete with respect to $\mathcal{Z}$ if $|\mathcal{M}| = |\mathcal{Z}|$ holds.

Definition 5.7 means that for a complete matching $\mathcal{M}$ on $\mathcal{C}$ (respectively on $\mathcal{Z}$), each constraint (respectively each variable) belongs to exactly one edge of the matching:

$$\begin{aligned} &\forall c \in \mathcal{C}: \qquad &&\exists z \in \mathcal{Z} \text{ such that } (c, z) \in \mathcal{M} \\ &\forall z \in \mathcal{Z}: \qquad &&\exists c \in \mathcal{C} \text{ such that } (c, z) \in \mathcal{M}. \end{aligned}$$

A matching can be represented by selecting at most one "1" in each row and in each column in the incidence matrix of the bi-partite graph, and representing them by the "①" in the examples. Each selected "1" represents an edge of the matching. No other edge should contain the same variable (thus it is the only one in the row) or the same constraint (this it is the only one in the column).

It is obviously possible to define matchings, maximal matchings, and complete matchings by considering either the whole structure of the system or only subgraphs of its structural graph, i.e. subsets of the constraints and variables instead of the whole sets.

Example 5.6 *Matchings on the reduced structure graph of a tank*
In order to illustrate the notion of maximal and complete matchings, consider the reduced structure graph of the single tank example. The edges of a matching are identified by a thick line in the graph representation and by "①" in the following incidence matrices.

$\nearrow$	h	$\dot{h}$	q_i	q_o
c_1		1	①	1
c_2			1	
c_3	①			1
c_4	1			
c_6	1	①		

Matching a)

$\nearrow$	h	$\dot{h}$	q_i	q_o
c_1		1	1	1
c_2			①	
c_3	1			①
c_4	①			
c_6	1	①		

Matching b)

$\nearrow$	h	$\dot{h}$	q_i	q_o
c_1		①	1	1
c_2			①	
c_3	1			①
c_4	1			
c_6	①	1		

Matching c)

As in a matching, unknown variables are associated with a constraint by means of which they can be determined, an intuitive graphical representation is given in Fig. 5.9 where the constraints are drawn on the left-hand side and the variables on the right-hand side. The thick edges connect the variables with the constraints by which the variable can be calculated. The graphs are the same as in Fig. 5.7.

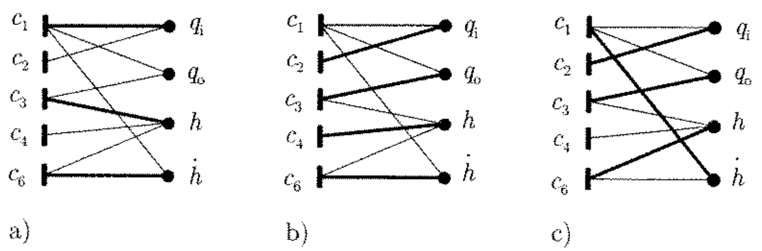


Fig. 5.9. An incomplete (a) and two complete (b, c) matchings

Figure 5.9a shows an incomplete matching. It is not complete with respect to the constraints since constraints c_2 and c_4 are not matched, nor is it complete with respect to the variables since q_o is not matched. However, no edge can be added to the matching without violating Definition 5.5.

Two complete matchings with respect to the unknown variables are shown in Fig. 5.9b and Fig. 5.9c. There is no matching that is complete with respect to $\mathcal{C}_\mathcal{X}$, because the number of constraints is larger than the number of variables. Note that it is not guaranteed that a complete matching exists, either with respect to $\mathcal{C}_\mathcal{X}$ or to $\mathcal{X}$. □

5.3.2 Oriented graph associated with a matching

Defining a matching on a structure graph introduces some orientations of the edges which, until now, were undirected. Constraints which appear in the system description have no direction, because all variables have the same status. For example, the tank constraint

$$c_1 : q_i(t) - q_o(t) - \dot{h}(t) = 0 \tag{5.9}$$

can be used to compute any of the three variables whenever the two other variables are known. It is written in the non-oriented form to stress that the constraint itself has no preference for any of the three variables.

Once a matching is chosen, this symmetry is broken, since each matched constraint is now associated with one matched variable and some non-matched ones. For a given constraint, matched and non-matched variables are identified in the graph incidence matrix by ① or 1, respectively. For example, according to the matching in Fig. 5.9a, the above constraint is used to compute $q_i(t)$.

In the graphical representation, the unsymmetries associated with a matching are represented by transforming the originally non-oriented edges into oriented ones. Since some constraints might not be matched, the following rules are applied:

- **Matched constraints:** The edges adjacent to a matched constraint are provided with an orientation
  - from the non-matched (input) variables, to the constraint,
  - from the constraint to the matched (output) variables.
- **Non-matched constraints:** All the variables are considered as input and, hence, all edges are oriented from the variables to the constraint.

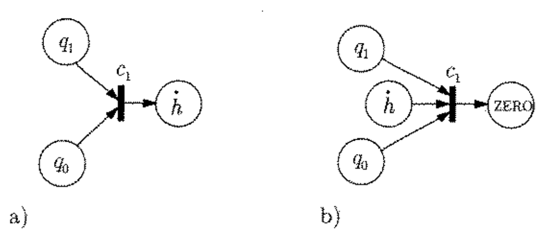


Fig. 5.10. a) A matched and b) a non-matched constraint (5.9)

To understand the reason for these rules, consider a matching $\mathcal{M}$ and choose an edge $(c, x) \in \mathcal{M}$. Then the variable x can be considered as the output of the constraint c while the other variables $Q(c)\backslash\{x\}$ are the input[1]. The

[1] In Eq.(5.8), Q has been defined as a mapping from a set of constraints towards the sets of variables. It is used here also for a single constraint c where for notational convenience $Q(\{c\})$ is written as $Q(c)$.

interpretation is that the matching represents some causality assignment by which the constraint c is used to compute the variable x assuming the other variables to be known. An explicit representation of the constraint c that can be used to determine x is denoted by

$$x = \gamma\left(Q\left(c\right)\backslash\{x\}\right).$$

For non-matched constraints all variables are considered as input and no variable of $Q(c)$ can be considered as an output. Hence, the constraint can be written in the form

$$c\left(Q(c)\right) = 0$$

like Eq. (5.9). If the zero on the right-hand side is considered as output, the constraint can be associated with a ZERO vertex like in Fig. 5.10b. Using no label at all is considered as an implicit ZERO label.

Example 5.7 *Computation of unknown variables of the tank*
For the single tank, the reduced graph shown in Fig. 5.7 and the three matchings shown in Fig. 5.9 yield the oriented graphs shown in Fig. 5.11. Remember that the reduced structure graph includes only the unknown variable. The measured input u and output y are introduced into the graph to illustrate how the constraints can be used to determine the internal variables q_i, q_o, h and $\dot{h}$ for known values of u and y.

As Matching 1 is incomplete, the unknown variable q_o cannot be computed as shown in the graph. Matchings 2 and 3 are complete in $\mathcal{X}$ but incomplete in $\mathcal{C}_{\mathcal{X}}$. The non-matched constraint c_1 or c_4, respectively, leads to a ZERO output, that is, they have to hold for the variables q_i and $\dot{h}$ or h and y, which have been determined by other constraints or have been measured, respectively. □

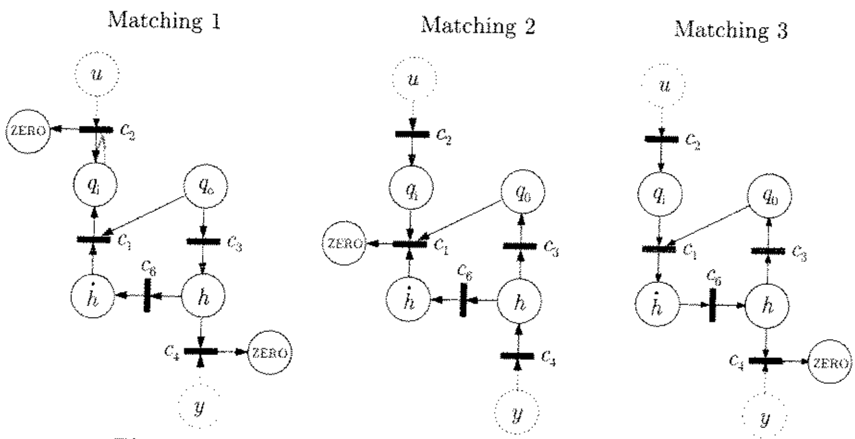


Fig. 5.11. Graphs corresponding to the three matchings

Note that subgraphs whose input and output nodes are all known or ZERO variables provide the system input-output relations. By using Matching 2 in Fig. 5.11 the two following input-output relations are found. The first one is provided by constraint c_5 which links only known variables and are, therefore, deleted when drawing the reduced graph, and the second one is provided by the non-matched constraint c_2

$$c_5\left(u,\,y\right) = 0$$
$$c_2\left(u,\,\gamma_1\,\left(\gamma_3\,\left(k,\,\gamma_4(y)\right),\,\gamma_6\left(\gamma_4(y)\right)\right)\right) = 0,$$

where $\gamma_i(z)$ denotes the output of constraint c_i whose input is z.

5.3.3 Alternated chains and reachability

The oriented graph associated with a matching obviously enjoys the following property: Any existing path between two nodes (variables or constraints) alternates successively variables and constraints nodes. Such a path is called an *alternated chain*. Its length is the number of constraints that are crossed along the path. (Note that if a non-matched constraint belongs to an alternated chain, the chain ends on the ZERO variable associated with the non-matched constraint).

Associated with alternated chains is the notion of reachability.

Definition 5.8 (Reachability)
A variable z_2 is reachable from a variable z_1 if there exists an alternated chain from z_1 to z_2. A variable z_2 is reachable from a subset $\chi \subseteq \mathcal{Z} \backslash \{z_2\}$ if there exists $z_1 \in \chi$ such that z_2 is reachable from z_1. A subset of variables $\mathcal{Z}_2$ is reachable from a subset of variables $\mathcal{Z}_1$ if any variable of $\mathcal{Z}_2$ is reachable from $\mathcal{Z}_1$.

Example 5.8 *Alternated chains in the tank example*
Some alternated chains associated with the oriented graph of the tank example are as follows:

$$y - c_4 - h - c_3 - q_0 - c_1 - q_i$$
$$h - c_6 - \dot{h} - c_1 - q_i$$

and it can be checked that any variable of the set $\left\{q_i,\, q_o,\, h,\, \dot{h}\right\}$ is reachable from y. □

5.3.4 Causal interpretation

The aim of this subsection is to discuss the causal interpretation of the oriented bi-partite graph associated with a matching.

Indeed, selecting a pair (c, z) to belong to a matching implies a causality assignment, by which the constraint c is used to compute the variable z, assuming the other variables to be known. The oriented bi-partite graph which results from a causality assignment is named a *causal graph*. Causal graphs are used in qualitative reasoning, alarm filtering or in providing the computation chain needed for the numerical or formal determination of some variables of interest, as shown by the above interpretation. Although this interpretation is straightforward for simple algebraic constraints, it has to be considered more carefully when loops and differential constraints are present.

Algebraic constraints. Let $c \in C$ be an algebraic constraint, $Q(c)$ the set of the variables constrained by c and let $n_c = |Q(c)|$. In structural analysis, the following assumption is made:

Assumption 5.1 *Any algebraic constraint $c \in C$ defines a manifold of dimension $n_c - 1$ in the space of the variables $Q(c)$.*

Indeed, since the constraint has to be satisfied at any time t, the variables $Q(c)$ cannot behave independently of each other. Assumption 5.1 means that only $n_c - 1$ unknowns can be chosen arbitrarily (or imposed) in constraint c or, in other words, that there is at least one variable $z \in Q(c)$ such that $\frac{\partial c}{\partial z} \neq 0$ (almost everywhere in the space of the variables $Q(c)$). Therefore, from the inverse function theorem, its trajectory can be deduced (at least locally) from the constraint c and the trajectories of the $n_c - 1$ others. This is exactly the causal interpretation of matching this variable with constraint c, and it can be interpreted as: constraint c decreases by one the degrees of freedom associated with the variables $Q(c)$.

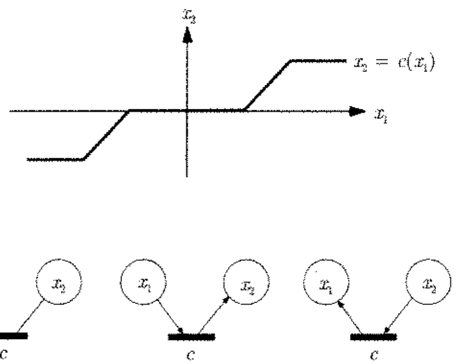
Example 5.9 *Non-invertible algebraic constraints*
Consider the constraint

$$c_1 : a_1 x_1 + b_1 x_2 - y_1 = 0, \tag{5.10}$$

where x_1 and x_2 are two unknowns (2 degrees of freedom), a_1 and b_1 are parameters, and y_1 is known. This constraint obviously defines a one dimensional space to which any vector $(x_1, x_2)'$ should belong when it is satisfied. Thus only one degree of freedom is left since only one of the unknowns can be chosen arbitrarily, the possible value(s) of the other one being deduced from (5.10).

Note that the structural point of view considers the most general case of any pair of parameters a_1 and b_1. Particular cases would be a_1 or b_1 equal to zero ((5.10) would still define a one dimensional manifold) and a_1 and b_1 both equal to zero (in this case c_1 would not define a one dimensional manifold when $y_1 = 0$, since any point $(x_1, x_2)'$ in the two dimensional space would satisfy the constraint, and there would be no solution when $y_1 \neq 0$ i.e. the system model would not be sound since constraint (5.10) would allow for no solution). □

The fact that *at least one* variable can be matched in a given constraint under the causal interpretation does not mean that *any* variable enjoys this property. An obvious situation in which (c, x) cannot be matched is when c is not invertible with respect to x. The constraint shown in Fig. 5.12 defines a manifold of dimension 1 in $\mathbb{R}^2$, and it is always possible to compute x_2 once x_1 is given. Matching x_2 with this constraint can obviously be interpreted as explained above. However, the interpretation does not apply to the matching of x_1, since $\frac{\partial c}{\partial x_1}$ is not different from zero almost everywhere, thus the constraint c cannot be used to compute x_1 whatever the value of x_2.



a) b) c)

Fig. 5.12. a) Structure graph, b) possible and c)impossible matching

Differential constraints. The case of differential constraints has to be considered more carefully. Remember that differential constraints can always be represented under the form (see (5.6))

$$d : x_2(t) - \frac{d}{dt}x_1(t) = 0 \tag{5.11}$$

and indeed functions $x_1(t)$ and $x_2(t)$ cannot be chosen independently of each other. Obviously, when the trajectory $x_1(t)$ is known, its derivative can always be computed (from an analytical point of view, derivatives are here supposed to exist, and from a numerical point of view, there might be problems rised by the presence of noise, which are not considered here). It follows that the constraint can always be matched for x_2 which is then uniquely defined. This is called *derivative causality*. When $x_2(t)$ is known, matching this constraint for x_1, (which is called *integral causality*), leads to the computation

$$x_1(t) = x_1(0) + \int_0^t x_2(\sigma)d\sigma, \tag{5.12}$$

which does not determine $x_1(t)$ uniquely, unless the initial condition $x_1(0)$ is known. Indeed, let $(x_1(t), x_2(t))'$ be two functions which satisfy constraint d. Then, $(x_1(t) + \alpha, x_2(t))'$, where α is any constant function, also satisfies constraint d. Thus, computing x_1 from constraint d may be possible or impossible, depending on the context. Initial values are known in a simulation context, since they are under the control of the user, but this is generally not true in a fault diagnosis context, thus forbidding integral causality in differential constraints.

Remark 5.1 *Consequences for residual generation*

Parity space or identification-based residual generation approaches aim at eliminating the unknown initial values by using the system input-output relations which are obtained through derivative causality. It can be noticed that observer-based approaches use integral causality by implementing an auxiliary system – the observer – which provides results that are (asymptotically) independent of the estimate of the initial state. □

In summary, different cases have to be considered as far as the satisfaction of Assumption 5.1 is concerned with respect to the differential constraint (5.11):

- If $x_1(t)$ is known, $x_2(t)$ can be matched in constraint d which brings about using differential causality. This provides a unique result for x_2. Assumption 5.1 is satisfied since constraint d leaves only one degree of freedom in the determination of $(x_1(t), x_2(t))$.
- If $x_2(t)$ and the initial value $x_1(0)$ are known, $x_1(t)$ can be matched in constraint d using integral causality. This provides a unique result obtained from Eq. (5.12). Assumption 5.1 is satisfied since constraint d leaves only one degree of freedom in the determination of $(x_1(t), x_2(t))$.
- If only $x_2(t)$ is known, Assumption 5.1 is not satisfied, because whatever the matching, two degrees of freedom (the constant function α, and the input function $x_2(t)$) still exist in the determination of $(x_1(t), x_2(t))$.

Example 5.10 *Differential system*

A model whose solution exists but is not unique, as the result of Assumption 5.1 being not satisfied, is given by the following simple example

$$\begin{aligned} c_1 &: \quad x_2 - ax_1 - bu = 0 \\ c_2 &: \quad x_2 - \frac{d}{dt}x_1 = 0, \end{aligned}$$

which is a single input first order system. Constraint c_1 is an algebraic one which expresses that the vector $(x_1, x_2)'$ lives in a linear manifold of dimension one (since u is known). Constraint c_2 does not allow to decrease the dimension of the unknown vector: indeed, if x_1 were known (which is not the case), one could compute its derivative x_2, but the knowledge of x_2 (which could be obtained as a function of x_1 and u in constraint c_1) is of no help to compute x_1 since one should proceed by integration and the initial value $x_1(0)$ is unknown. □

Example 5.11 *Derivative causality in the tank system*

As a second example, consider the following matching in the tank structural model. Although it is complete with respect to the variables $\{h, \dot{h}, q_i, q_o\}$, it cannot be used for their computation since it introduces an integral causality (h should be computed from $\dot{h}$ by constraint c_6, while its initial value is not known because constraint c_4 is not matched).

↗	h	$\dot{h}$	q_i	q_o	u	y
c_1		①	1	1		
c_2			①		1	
c_3	1			①		
c_4	1					1
c_5					1	1
c_6	①	1				

Derivative causality can be forced, when necessary. To represent this situation, a special notation can be used, namely **x**, which forbids integral matchings. The previous matching will not be obtained if the tank structural model is written as

↗	h	$\dot{h}$	q_i	q_o	u	y
c_1		1	1	1		
c_2			1		1	
c_3	1			1		
c_4	1					1
c_5					1	1
c_6	x	1				

where **x** means that although there is an edge between c_6 and h, h cannot be matched with c_6. □

Subsets of constraints. When a set $\mathcal{C}$ of constraints (or a subset) are simultaneously considered, two further assumptions are made in structural analysis. They express that the model defined by that set of constraints is well formed, which means that it indeed has solutions (Assumption 5.2), and that the same set of solutions could not have been represented using a model with less constraints (Assumption 5.3).

Assumption 5.2 *All the constraints in* $\mathcal{C}$ *are compatible.*

This is a very obvious assumption, which means that the set of the constraints is associated with a sound model, namely a model whose solutions exist. In other words, the constraints do not carry any contradiction. Let $\mathcal{C}_1 \subseteq \mathcal{C}$ be the subset of all the constraints which satisfy Assumption 5.1. Let

$V(c)$ be the $n_c - 1$ dimensional manifold associated with constraint $c \in \mathcal{C}_1$, and let

$$V(\mathcal{C}) = \cap_{c \in \mathcal{C}_1} V(c).$$

Assumption 5.2 means that

$$V(\mathcal{C}) \neq \emptyset,$$

which can be interpreted as follows: The system behaviour model has at least one solution, i.e. the set of the values of the variables which satisfy all the system constraints is not empty.

Assumption 5.3 *All the constraints in $\mathcal{C}$ are independent.*

Assumption 5.3 states that the model is minimal in the sense that no constraint defines (at least locally) the same manifold as another one, or more generally that there does not exist in $\mathcal{C}$ two different subsets of constraints $\mathcal{C}'$ and $\mathcal{C}''$ such that

$$V(\mathcal{C}') \subseteq V(\mathcal{C}'').$$

When Assumption 5.3 is satisfied by the constraints of a given set $\mathcal{C}$, the consequence is that the dimension of the manifold $V(\mathcal{C})$ is

$$d\left[V(\mathcal{C})\right] = |Q(\mathcal{C})| - |\mathcal{C}_1|, \tag{5.13}$$

where $\mathcal{C}_1$ is the subset of the constraints of $\mathcal{C}$ which satisfies Assumption 5.1.

Example 5.12 *Dependent constraints*
Consider the two constraints

$$\begin{aligned} c_1 : &\quad z_1 - 1 = 0 \\ c_2 : &\quad (z_1 - 1)(z_2 - 1) = 0. \end{aligned}$$

They are obviously not independent, since one has $V(c_1) \cap V(c_2) = V(c_1)$. In fact, constraint c_1 is enough to describe the set of the system solutions. □

Loops. In the oriented graph associated with a matching, loops are special subsets of constraints, which have to be solved simultaneously, because the output signals of some constraints in the loop are the inputs of some others in the same loop. Since only one variable is matched with one constraint, the number of matched variables in a loop is equal to the length of the loop, i.e. the number of the constraints which appear in it.

The causal interpretation of a loop is that of a subset of constraints whose solution is the set of the matched variables, when all the other variables (not matched in the loop) are known.

Suppose n_v variables are constrained by a subsystem of n_l constraints, and there is a matching such that they form a loop. Then, n_l variables are internal (matched in the loop), and $n_v - n_l$ variables are external (not matched in the loop).

In structural analysis, an algebraic loop is always supposed to have a unique solution (more precisely: a finite number of solutions), which is the intersection of n_l manifolds of dimension $n_l - 1$, if the external variables are known (by Assumptions 5.2 and 5.3). Indeed, the loop is associated with a subset of n_l constraints

$$h_l(x_l, x_e) = 0,$$

where x_l (respectively x_e) are the internal (respectively the external) variables, and each component of x_l is matched with one constraint in h_l. It is worth noticing that the interpretation associated with causality in single constraints is not directly extendable to loops.

Example 5.13 *A system of two non-invertible constraints*
Consider the non-invertible constraint from Example 5.9 and suppose now there are two constraints $\{c_1, c_2\}$ of the same form, but with different parameters. The structural graph of this system is

$\nearrow$	x_1	x_2
c_1	1	1
c_2	1	1

A complete matching is given by

$\nearrow$	x_1	x_2
c_1	①	1
c_2	1	①

which shows the existence of a loop, and indeed the system of two constraints with two unknowns has a (finite number of) solution(s), as illustrated by Fig. 5.13, although the matching of x_1 with c_1 has certainly not the nice interpretation of computing the variable from the constraint.

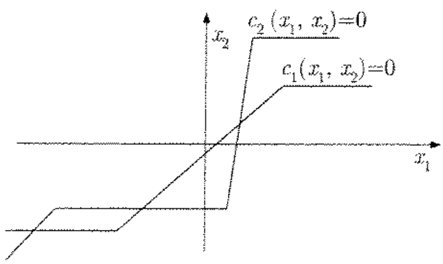


Fig. 5.13. Two algebraic constraints with two unknowns

The correct interpretation, in this case, comes from the fact that each constraint defines a (different) manifold of dimension one in $\mathbb{R}^2$, and that, in general, such two manifolds will intersect in a finite number of points. Indeed, no solution at all would be a particular case (which would not satisfy Assumption 5.2), and an infinite number of solutions would be the result of the two manifolds being the same one, at least on some sub-region of the space - another particular case (which would not satisfy Assumption 5.3). □

The uniqueness of the solution associated with a loop which contains differential constraints will depend on the context of the problem. Indeed, consider a set of $n_l + n_e$ variables which are constrained by n_l differential equations, and consider matchings such that this system forms a loop

$$\begin{aligned} \boldsymbol{z}_l &= \boldsymbol{g}_l(\boldsymbol{x}_l, \boldsymbol{x}_e, \boldsymbol{u}) \\ \boldsymbol{z}_l &= \frac{d}{dt}\boldsymbol{x}_l, \end{aligned} \tag{5.14}$$

where $\boldsymbol{x}_l$ is the vector of the variables in the loop, $\boldsymbol{g}_l$ are the constraints in the loop, and $\boldsymbol{x}_e$ are the external variables, supposed to be known. The system (5.14) has a unique solution only if the initial value $\boldsymbol{x}_l(0)$ is known. When this is not the case, the solution will depend on the n_l unknowns $\boldsymbol{x}_l(0)$, and thus it will belong to a manifold of dimension n_l. Such a differential loop is called *non-causal.*

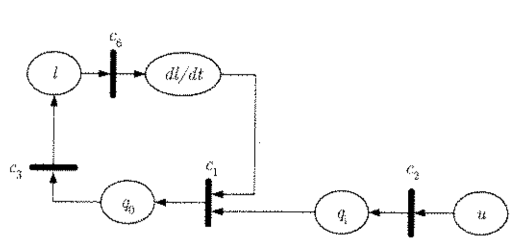


Fig. 5.14. A matching with a differential loop

Example 5.14 *Differential loop in the tank example*
Consider again the tank example, with the following matching, which is complete with respect to $\{h, \dot{h}, q_i, q_o\}$, and in which differential causality is now used in constraint c_6. This is obviously not enough for being able to compute h, since there is a differential loop: $h - c_6 - \dot{h} - c_1 - q_0 - c_3 - h$, which is shown of Fig. 5.14. □

↗	h	$\dot{h}$	q_i	q_o	u	y
c_1		1	1	①		
c_2			①		1	
c_3	①			1		
c_4	1					1
c_5					1	1
c_6	1	①				

Following a classical graph theory approach, a loop can be condensed into one single node (which thus represents a subsystem of constraints which are to be solved simultaneously). Another approach is to avoid loops (when possible) by some transformation of the constraints, leading to diagonal or triangular system structures.

Example 5.15 *Treatment of loops*
Let a subsystem be defined by $\mathcal{Z} = \{x_1, x_2, y_1, y_2\}$, $\mathcal{C} = \{c_1, c_2\}$. The variables are real numbers, the constraints are linear, y_1, y_2 are supposed to be known, and we are interested in the computation of x_1, x_2.

$$\begin{aligned} c_1 &: \quad a\,y_1 + b\,x_1 + c\,x_2 = 0 \\ c_2 &: \quad \alpha\,y_2 + \beta\,x_1 + \gamma\,x_2 = 0. \end{aligned} \tag{5.15}$$

The incidence matrix of the structure graph, and a complete matching w.r.t. $\{x_1, x_2\}$ are

↗	x_1	x_2	y_1	y_2
c_1	①	1	1	
c_2	1	①		1

and Fig. 5.15 shows the resulting loop in the associated oriented graph. Note that this matching is valid almost for every value of the coefficients since it obviously supposes that b and γ are non-zero, note also that the solvability condition $b\,\gamma - c\,\beta \neq 0$ cannot be seen from structural considerations.

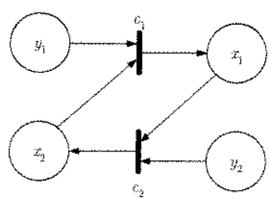


Fig. 5.15. An algebraic loop

Figure 5.16 illustrates the condensation in which the loop is "condensed" into one single node, which means that the system of two equations with two unknowns is solved, but no detail is given about how this is done.

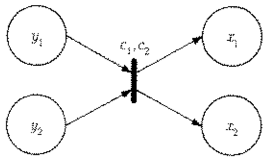


Fig. 5.16. Condensed representation of the loop

Transforming the constraints may also lead to a loop-free oriented graph, since it ay give the system a diagonal or a triangular structure (note that this results from anipulations which are not purely structural, but which are done on the system ehaviour model). Indeed, for example, the two following systems are equivalent to .15) (other equivalent systems can also be constructed):

$$\begin{aligned} c_1' &: \quad a\,\gamma\,y_1 - \alpha\,c\,y_2 + (b\,\gamma - \beta\,c)\,x_1 = 0 \\ c_2' &: \quad a\,\beta\,y_1 - \alpha\,b\,y_2 + (c\,\beta - b\,\gamma)\,x_2 = 0 \end{aligned} \tag{5.16}$$

$$\begin{aligned} c_1' &: \quad a\,\gamma\,y_1 - \alpha\,c\,y_2 + (b\,\gamma - \beta\,c)\,x_1 = 0 \\ c_2'' &: \quad ay_1 + bx_1 + cx_2 = 0. \end{aligned} \tag{5.17}$$

gure 5.17 illustrates the loop-free oriented graphs associated with systems (5.16) id (5.17). □

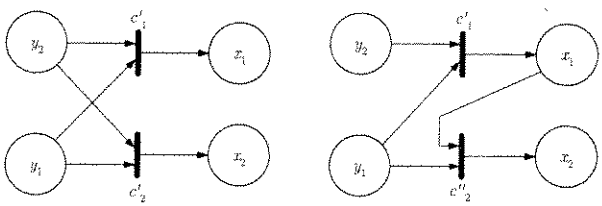


Fig. 5.17. Two equivalent loop-free oriented graphs

3.5 Matching algorithm

om the definition, a matching can be represented in the incidence matrix the bi-partite graph by selecting one "1" at most in each row and in each lumn. Each selected "1" represents an edge of the matching, and no other ge of the matching should contain the same variable (thus it is the only one the row) or the same constraint (thus it is the only one in the column).

Finding maximal matchings thus calls for "affectation algorithms" which e well known in operations research.

The following *constraint propagation* (or *ranking*) algorithm can be used to d a matching. It does only generate oriented graphs without loops, so it iy not find any complete matching, even if it exists. The idea is to start with some variable (in applications the starting nodes are the known variables), and to "propagate" the knowledge, step by step, by matching, at each step, the variables which intervene in constraints where all other involved variables are matched or known.

Algorithm 5.1 *Ranking of the constraints*

Given: Incidence matrix or structure graph

1. Mark all known variables.
   $i = 0$
2. Find all constraints with exactly one unmarked variable. Associate rank i with these constraints and mark these constraints as well as the corresponding variable.
3. If there are unmarked constraints whose variables are all marked, associate them with rank i, mark them and connect them with the pseudo-variable ZERO.
4. Set $i := i + 1$.
5. If there are unmarked variables or constraints, continue with step 2.

Result: Ranking of the constraints.

In the first step, all known variables $\mathcal{K}$ are marked and all unknown variables remain unmarked. Then every constraint that contains at most one unmarked variable is assigned rank 0. The constraint is matched for the unmarked variables (or zero, if there is none), and the variable is marked. This step is repeated with an increasing rank number, until no new variables can be matched.

As every matched variable is also given a number, this approach is called ranking algorithm. The rank can be interpreted as the number of steps needed to calculate an unknown variable from the known ones.

Example 5.16 *Constraint propagation in the single tank*
The constraint propagation algorithm applied to the tank example works as follows (remember that u, y being known, only $\{q_i, q_o, h, \dot{h}\}$ have to be matched):

Starting set (rank 0): $\{u, y\}$
First step (rank 1): match q_i with c_2, match h with c_4
Second step (rank 2): match q_0 with c_3, match $\dot{h}$ with c_6
End (every variable is matched)

The obtained matching is:

↗	h	$\dot{h}$	q_i	q_o	u	y
c_1		1	1	1		
c_2			①		1	
c_3	1			①		
c_4	①					1
c_5					1	1
c_6	1	①				

□

In more complex situations, maximum matchings can be constructed by selecting any initial matching and trying to increase the length of the associated alternated chains by exchanging the role of matched and non-matched variables. In order to give an idea of the procedure, let us first introduce some terminology.

Let $\mathcal{M}$ be a matching on a graph $\mathcal{G}$. An edge is defined to be *weak with respect to* $\mathcal{M}$ if it does not belong to the matching. A vertex is defined to be *weak with respect to* $\mathcal{M}$ if it is only incident to weak edges. An $\mathcal{M}$*-alternating path* is a path whose edges are alternately in $\mathcal{M}$ and not in $\mathcal{M}$ (or conversely). An $\mathcal{M}$*-augmenting path* is an alternating path whose end vertices are both weak with respect to $\mathcal{M}$. Then, BERGE (1957) has the following theorem.

Theorem 5.1 *A matching $\mathcal{M}$ in a graph is a maximum matching if and only if there exists no $\mathcal{M}$-augmenting path in $\mathcal{G}$.*

The proof is not given here but roughly, the idea is that if such an augmenting path exists, a new matching of size $|\mathcal{M} + 1|$ will be obtained by exchanging the roles of matched and non-matched constraints.

Example 5.17 *Augmenting path in the tank example*
Let us start with the maximal incomplete matching, whose cardinal is three: ?

↗	h	$\dot{h}$	q_i	q_o	u	y
c_1		1	①	1		
c_2			1		1	
c_3	①			1		
c_4	1					1
c_5					1	1
c_6	1	①				

It is possible to increase the size of the matching by exchanging the roles of q_i and q_0 in constraint c_1, since q_i can also be matched in c_2, or by exchanging the roles of h and q_0 in c_3, since h can also be matched in c_4. The result is a maximal matching whose cardinal is larger. In that case the matching is not only maximum but it is also complete with respect to $\{q_i, q_o, h, \dot{h}\}$. □

Other matching algorithms are connected with the search of a maximal flow or a maximal cover (see the bibliographical notes).

5.4 System canonical decomposition

This section recalls a classical result from bi-partite graph theory, which states that any finite-dimensional graph can be decomposed into three subgraphs 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, monitorability, reconfigurability.

5.4.1 Definitions

Definition 5.9 (Over-constrained graph)
A graph $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ is called over-constrained if there is a complete matching on the variables $\mathcal{Z}$ but not on the constraints $\mathcal{C}$.

There remains a complete matching on $\mathcal{Z}$ after any single contraint has been removed from the set $\mathcal{C}$.

Definition 5.10 (Just-constrained graph)
A graph $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ is called just-constrained if there is a complete matching on the variables $\mathcal{Z}$ and on the constraints $\mathcal{C}$.

Definition 5.11 (Under-constrained graph)
A graph $(\mathcal{C}, \mathcal{Z}, \mathcal{E})$ is called under-constrained if there is a complete matching on the constraints $\mathcal{C}$ but not on the variables $\mathcal{Z}$.

Example 5.18 *Property of the reduced graph of the tank system.*
Matching 2 of the tank graph is complete with respect to the variables, but there is still one non-matched constraint: the reduced graph of the tank system is over-constrained.

↗	h	$\dot{h}$	q_i	q_o
c_1		1	①	1
c_2			1	
c_3	1			①
c_4	①			
c_6	1	①		

(It can be furthermore noticed that any of the 5 constraints can be removed, and there still is a complete matching on the resulting graph). □

5.4.2 Canonical subsystems

The graph of a given system $\mathcal{S} = (\mathcal{C}, \mathcal{Z})$ may of course fail to conform to any of the three above properties. In this case, it can be proved that there exists a unique decomposition of $\mathcal{S}$ into three subsystems

$$\begin{aligned} \mathcal{S}^+ &= (\mathcal{C}^+, \mathcal{Z}^+) \\ \mathcal{S}^0 &= (\mathcal{C}^0, \mathcal{Z}^+ \cup \mathcal{Z}^0) \\ \mathcal{S}^- &= (\mathcal{C}^-, \mathcal{Z}^+ \cup \mathcal{Z}^0 \cup \mathcal{Z}^-) \end{aligned}$$

such that

(a) $(\mathcal{C}^-, \mathcal{C}^0, \mathcal{C}^+)$ form a partition of $\mathcal{C}$,
(b) $(\mathcal{Z}^-, \mathcal{Z}^0, \mathcal{Z}^+)$ form a partition of $\mathcal{Z}$,
(c) $(\mathcal{C}^+, \mathcal{Z}^+)$ is over-constrained,
(d) $(\mathcal{C}^0, \mathcal{Z}^0)$ is just-constrained,
(e) $(\mathcal{C}^-, \mathcal{Z}^-)$ is under-constrained.

- $\mathcal{S}^+$ is called the over-constrained subsystem because it follows from the definition that $|\mathcal{C}^+| > |\mathcal{Z}^+|$, which means that the variables $\mathcal{Z}^+$ (let n^+ be their number) have to satisfy more than n^+ constraints.
- $\mathcal{S}^0$ is called the just-constrained subsystem, because it introduces as many new variables as constraints ($|\mathcal{C}^0| = |\mathcal{Z}^0|$).
- Similarly, $\mathcal{S}^-$ is called the under-constrained subsystem. It is characterised by $|\mathcal{Z}^-| > |\mathcal{C}^-|$ (it introduces more new variables than constraints).

Figure 5.18 illustrates the canonical decomposition of the structure graph, showing the partition of $\mathcal{C}$ into $\{\mathcal{C}^+, \mathcal{C}^0, \mathcal{C}^-\}$ and the partition of $\mathcal{Z}$ into $\{\mathcal{Z}^+, \mathcal{Z}^0, \mathcal{Z}^-\}$ which define the three canonical components on its incidence matrix. White areas are zeros, grey areas contain zeros and ones, and the thick line represents a matching (after the rows and columns have been rearranged so that the matched variables and constraints appear on the diagonal).

It can be noticed that the over-constrained subsystem may contain several smaller over-constrained or just-constrained ones (at least one smaller over-constrained subsystem has to exist, otherwise the overall graph would be just-constrained, and of course no under-constrained subsystem can be found in an over-constrained one). Similarly, the just-constrained subsystem can contain several smaller just-constrained ones (no over-constrained, no under-constrained subsystem can exist otherwise it would not be just-constrained), and in an under-constrained subsystem, several under-constrained ones (no over-constrained, no just-constrained) may exist.

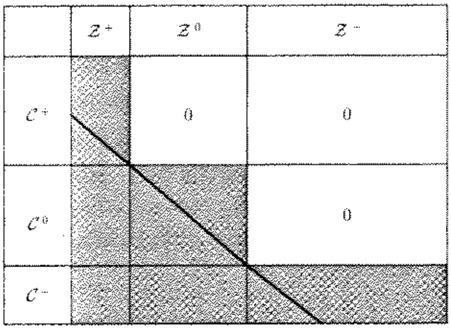


Fig. 5.18. Canonical decomposition of the structure graph

Subsystems which cannot be decomposed into smaller ones are minimal subsystems.

Example 5.19 *Minimal subsystems*
The following over-constrained system

↗	x_1	x_2	x_3	x_4	x_5	x_6
c_1	1					1
c_2		1				1
c_3			1	1	1	1
c_4	1			1		
c_5		1			1	
c_6			1			1
c_7						1

contains six minimal subsystems, five being just-constrained

$$(\{c_7\}, \{x_6\}),$$
$$(\{c_1\}, \{x_1, x_6\}),$$
$$(\{c_2\}, \{x_2, x_6\}),$$
$$(\{c_6\}, \{x_3, x_6\}),$$
$$(\{c_4\}, \{x_1, x_4\}),$$

and the last one

$$\{c_5, c_3\}, \qquad \{x_2, x_3, x_4, x_5, x_6\}$$

being over-constrained. This can be seen from the following rearrangement of the rows and columns of the incidence matrix.

↗	x_6	x_1	x_2	x_3	x_4	x_5
c_7	1					
c_1	1	1				
c_2	1		1			
c_6	1			1		
c_4		1			1	
c_5			1			1
c_3	1			1	1	1

□

Example 5.20 *The reduced graph of the tank*
Re-arranging the rows and columns of the reduced graph of the tank, the incidence matrix becomes:

↗	h	$\dot{h}$	q_o	q_i
c_4	1			
c_6	1	1		
c_3	1		1	
c_2				1
c_1		1	1	1

Three minimal just-constrained subsystems, namely $(\{c_4\}, \{h\})$, $\big(\{c_6\}, \{h, \dot{h}\}\big)$, $(\{c_3\}, \{h, q_o\})$ and one over-constrained subsystem, namely $\big(\{c_1, c_2\}, \{\dot{h}, q_o, q_i\}\big)$ can be distinguished. □

5.4.3 Interpretation of the canonical decomposition

In this section, the canonical subsystems are analysed from the point of view of the above assumptions, i.e. from the point of view of the existence of solutions, thus providing a key for the analysis of the system observability and controllability.

First, it is clear that Assumption 5.2 must be satisfied by each of the subsets of constraints $\mathcal{C}^+$, $\mathcal{C}^0$ and $\mathcal{C}^-$. Indeed, if this was not true, the system model would have no solution, which contradicts the fact that it describes the behaviour of a physical system (which indeed has a solution).

Second, in the structural point of view, any algebraic constraint is assumed to satisfy Assumption 5.1, thus a subset of n variables completely matched within a subset of n constraint, is uniquely defined, while the result depends on the causality and on the existence of differential loops when constraints of the form

$$d: \quad z_2(t) - \tfrac{d}{dt} z_1(t) = 0$$

are considered.

Finally, it will be seen that there are cases in which Assumption 5.3 cannot hold true.

Static systems. For clarity, let us start the analysis with static systems, whose behaviour model contains only algebraic constraints.

In the **over-constrained subsystem**, $(\mathcal{C}^+, Q(\mathcal{C}^+))$, the variables $\mathcal{Z}^+$ (let n^+ be their number) have to satisfy more than n^+ constraints, which satisfy Assumption 5.1 and 5.2. Therefore, they belong to the intersection of the manifolds associated with the constraints $\mathcal{C}^+$. Since there are more manifolds than variables, by (5.13), no solution can exist if they also satisfy Assumption 5.3. Because the model should exhibit at least one solution (indeed, the system is a physical one), one concludes that the constraints in $\mathcal{C}^+$ are not independent, i.e. the system description is redundant. In other words, for the system to have a solution, some compatibility conditions must hold. Structural analysis always assumes the most general case, i.e. the minimum number of relations between the system parameters. This means that the number of independent constraints is maximal, thus leading to the following equivalent conclusions:

- The over-constrained subsystem has a unique solution (more generally it has a finite number of solutions).
- The number of independent constraints in $\mathcal{C}^+$ is n^+.
- The number of compatibility conditions is $|\mathcal{C}^+| - n^+$.

In the **just-constrained subsystem**, $(\mathcal{C}^0, Q(\mathcal{C}^0))$, the variables $\mathcal{Z}^0$ (let n^0 be their number) have to satisfy exactly n^0 constraints, which satisfy Assumption 5.1 and 5.2. By (5.13), a unique solution exists, which is the intersection of the manifolds associated with the constraints $\mathcal{C}^0$, which are assumed to satisfy Assumption 5.3. This being the most general case, structural analysis proposes the following conclusions:

- The just-constrained subsystem has a unique solution.
- The number of independent constraints in $\mathcal{C}^0$ is n^0.
- There is no compatibility condition.

In the **under-constrained subsystem**, $(\mathcal{C}^-, Q(\mathcal{C}^-))$, the variables $\mathcal{Z}^-$ (let n^- be their number) have to satisfy less than n^- constraints, which satisfy Assumption 5.1 and 5.2. By (5.13), all the model can tell is that the unique solution of the physical system belongs to the intersection of less than n^- manifolds, and thus it is not uniquely defined by the model (indeed, it belongs to a manifold of dimension $n^- - |\mathcal{C}^-|$ if the constraints also satisfy Assumption 5.3). This being the most general case, structural analysis proposes the following conclusions:

- The under-constrained subsystem has no unique solution.

- The constraints in C^- are independent.
- There is no compatibility condition.

Example 5.21 *Compatibility conditions in an over-constrained subsystem*
Consider the set of linear constraints

$$\begin{aligned} c_1: &\quad a_1 x_1 + b_1 x_2 - y_1 = 0 \\ c_2: &\quad a_2 x_1 + b_2 x_2 - y_2 = 0 \\ c_3: &\quad a_3 x_1 + b_3 x_2 - y_3 = 0, \end{aligned} \tag{5.18}$$

where $a = (a_1, a_2, a_3)'$, $b = (b_1, b_2, b_3)'$ are known parameters and $y = (y_1, y_2, y_3)'$ are known variables. This system is clearly over-constrained with respect to the unknown variables (x_1, x_2). Let us consider the following cases.

1. rank$[a, b, y] = 3$, i.e. the parameters a, b and the known variables y are independent vectors, i.e. their nine components can be chosen arbitrarily. The system has in general no solution, since the three constraints are incompatible (Assumptions 5.1, 5.2 and 5.3 cannot hold simultaneously).
2. If rank$[a, b, y] = 2$, one solution exists. Note that the parameters and the known variables are no longer independent (namely $[a, b, y]$ has one null eigenvalue, thus $\exists\ \lambda, \mu \in R\backslash\{0\}$ such that $y = \lambda a + \mu b$, which is the compatibility condition). The unique solution is $x_1 = \lambda$ and $x_2 = \mu$. Assumptions 5.1 and 5.2 hold, and Assumption 5.3 does not.
3. If rank$[a, b, y] = 1$, $[a, b, y]$ has two null eigenvalues, thus $\exists\ \lambda, \mu \in R\backslash\{0\}$ such that $y = \lambda a = \mu b$, and more than one solution exists. Indeed, any pair (x_1, x_2) such that $x_1 + x_2 - \lambda\mu = 0$ satisfies the system of equations. Note that in that case, two compatibility conditions exist, and Assumption 5.3 does not hold.
4. The last case is rank$[a, b, y] = 0$, i.e. $a = b = y = 0$. In this case, all parameters are specified and any pair $(x, y)' \in \mathbb{R}^2$ satisfies the system of equations. Assumption 5.1 does not hold.

Since (5.18) is the behaviour model of a physical system, it should exhibit at least one solution. Then obviously the most general situation is case number 2 in which only one relation holds between the parameters. This is what is assumed by the structural point of view. □

Dynamic systems. Remember that, when differential constraints are considered, matching all the variables in a subsystem guarantees that there is a unique solution under integral causality, i.e. when the initial conditions are known. Under derivative causality, the solution is unique if and only if there is a matching which avoids differential loops.

Let n_1^+ (respectively n_1^0, n_1^-) be the maximal number of variables which can be matched in the over-constrained subsystem (respectively in the just-constrained, the under-constrained subsystems) without introducing any differential loop. One obviously has $n_1^+ \leq n^+$, $n_1^0 \leq n^0$, $n_1^- \leq n^-$.

The over-constrained (respectively the just-constrained) subsystem is called *causal* if there exists a complete matching with respect to the variables $\mathcal{Z}^+$ (respectively $\mathcal{Z}^0$) which does not contain any differential loop, i.e. if one has $n_1^+ = n^+$(respectively $n_1^0 = n^0$). Note that the under-constrained subsystem can obviously not be causal, since there does not even exist any complete matching with respect to $\mathcal{Z}^-$.

When the over-constrained (respectively the just-constrained) subsystem is causal, the same conclusions as above obviously apply for the interpretation of the three canonical subsystems. A non-causal subsystem does not exhibit all unique solution, since there are $n^+ - n_1^+$ variables in the over-constrained subsystem (respectively $n^0 - n_1^0$ variables in the just-constrained one) which are matched in differential loops, i.e. which are defined up to an unknown constant.

Example 5.22 *Causal over-constrained system*
As an example, consider the following system

$$\begin{aligned} c_1: &\quad x_2 - a\,x_1 - bu = 0 \\ c_2: &\quad x_2 - \alpha\,x_1 - \beta u = 0 \\ c_3: &\quad x_2 - \tfrac{d}{dt}x_1 = 0, \end{aligned}$$

which is over-constrained with respect to the variables $(x_1,\, x_2)$ and where u is supposed to be known. The system is causal since $(x_1,\, x_2)$ can be matched with $(c_1,\, c_2)$ and this introduces no differential loop. Thus, there is a unique solution, which is obtained from the intersection of the two manifolds associated with $(c_1,\, c_2)$, and which can be checked to be

$$\begin{aligned} x_1 &= \frac{\beta - b}{a - \alpha}u \\ x_2 &= \left(\frac{a\,\beta - \alpha\,b}{a - \alpha}\right)u \end{aligned}$$

($a - \alpha$ is assumed not to be zero). Moreover, constraint c_3 is redundant, and acts as a compatibility condition which has to be satisfied for the system solution to exist, namely

$$\left(\frac{a\,\beta - \alpha\,b}{a - \alpha}\right)u - \frac{\beta - b}{a - \alpha}\dot{u} = 0.$$

Suppose now that constraint c_2 does not exist, then the system is just-constrained but it is not causal, and its solution is defined up to the constant $x_1(0)$, which is unknown under differential causality. □

5.5 Observability

5.5.1 Observability and computability

Known and unknown variables. The system variables $\mathcal{Z}$ are decomposed into known (the set $\mathcal{K}$) and unknown ones (the set $\mathcal{X}$). Known variables are available in real time, while unknown variables are not directly measured. However, there might exist some way to compute their value from the values of known ones (past and present values are considered in discrete time models, while variables and their derivatives are considered in continuous time models). Analysing the system observability coincides with identifying those unknown variables for which such possibility exists.

Considering the general system described by the Eqs. (5.3), (5.4) (5.5) and (5.6)

$$\begin{aligned} \dot{\boldsymbol{x}}_d &= \boldsymbol{g}\left(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}\right) && (5.19) \\ \mathbf{0} &= \boldsymbol{m}\left(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}\right) && (5.20) \\ \boldsymbol{y} &= \boldsymbol{h}\left(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}\right) && (5.21) \\ \dot{\boldsymbol{x}}_d &= \frac{d}{dt}\boldsymbol{x}_d, && (5.22) \end{aligned}$$

the set of known variables is $\mathcal{K} = \boldsymbol{u} \cup \boldsymbol{y}$, the set of unknown variables $\mathcal{X} = \boldsymbol{x}_a \cup \boldsymbol{x}_d \cup \dot{\boldsymbol{x}}_d$ and the set of constraints $\mathcal{C} = \boldsymbol{g} \cup \boldsymbol{m} \cup \boldsymbol{h} \cup \frac{d}{dt}$. Following the decomposition of $\mathcal{Z}$ into $\mathcal{K} \cup \mathcal{X}$, $\mathcal{C}$ is decomposed into $\mathcal{C}_{\mathcal{K}} \cup \mathcal{C}_{\mathcal{X}}$, where

$$\begin{aligned} \mathcal{C}_{\mathcal{K}} &= \{c \in \mathcal{C};\ Q(c) \cap \mathcal{X} = \emptyset\} \\ \mathcal{C}_{\mathcal{X}} &= \{c \in \mathcal{C};\ Q(c) \cap \mathcal{X} \neq \emptyset\}. \end{aligned}$$

$\mathcal{C}_{\mathcal{K}}$ is obviously the largest subset of constraints such that $Q(\mathcal{C}_{\mathcal{K}}) \subseteq \mathcal{K}$. It can be noticed that the relations which define control algorithms belong to $\mathcal{C}_{\mathcal{K}}$ since they introduce constraints between the sensor output, the control objectives (set-points, tracked trajectories, final states) and the control output, which are all known variables. The aim being to analyse the possibility of computing the unknowns $\mathcal{X}$, only the subgraph $(\mathcal{C}_{\mathcal{X}}, \mathcal{X}, \mathcal{E}_{\mathcal{X}})$ needs to be decomposed.

Remark 5.2 *Observability and computability*
Consider the set $\mathcal{X} = \boldsymbol{x}_a \cup \boldsymbol{x}_d \cup \dot{\boldsymbol{x}}_d$ of the unknown variables. For the static variables $\boldsymbol{x}_a$, the term "observable" has obviously the same meaning as the term "computable". Namely, it means that their trajectories can be determined from the knowledge of the trajectories of the known variables. Consider a dynamical variable $x_d^i \in \boldsymbol{x}_d$. It appears along with its derivative in the system behaviour model. Therefore, several cases have to be considered:

Case 1:	x_d^i and $\dot{x}_d^i$ can be computed.
Case 2:	$\dot{x}_d^i$ can be computed, but not x_d^i.
Case 3:	x_d^i can be computed, but not $\dot{x}_d^i$.
Case 4:	nor x_d^i neither $\dot{x}_d^i$ can be computed.

Case 3 obviously cannot exist, since the knowledge of $x_d^i(t)$ implies the knowledge of $\dot{x}_d^i(t)$. In case 1, x_d^i and $\dot{x}_d^i$ are both computable and x_d^i is observable. In case 2, x_d^i is not observable, although $\dot{x}_d^i$ is computable, and in case 4 no one is computable, and x_d^i is non observable. Therefore, the set of computable variables includes (but is not restricted to) the set of observable variables, if "variable" means "component of the state vector" as usual in the control literature. In the framework of this chapter, since "variable" means any component in $\boldsymbol{x}_a \cup \boldsymbol{x}_d \cup \dot{\boldsymbol{x}}_d$, observability and computability do indeed coincide. For example in case 1, we say that variable $\dot{x}_d^i$ is observable and so is also variable x_d^i, in case 2, $\dot{x}_d^i$ is observable while x_d^i is not, and in case 4, no one of them is observable. □

5.5.2 Structural observability conditions

Let

$$\begin{aligned} \mathcal{S}^+ &= (\mathcal{C}^+_{\mathcal{X}}, \mathcal{X}^+) \\ \mathcal{S}^0 &= (\mathcal{C}^0_{\mathcal{X}}, \mathcal{X}^+ \cup \mathcal{X}^0) \\ \mathcal{S}^- &= (\mathcal{C}^-_{\mathcal{X}}, \mathcal{X}^+ \cup \mathcal{X}^0 \cup \mathcal{X}^-) \end{aligned}$$

be the canonical decomposition of the subgraph $(\mathcal{C}_{\mathcal{X}}, \mathcal{X}, \mathcal{E}_{\mathcal{X}})$ associated with system (5.19) – (5.22).

Theorem 5.2 (Structural observability)
A necessary and sufficient condition for system (5.19) – (5.22) to be structurally observable is that, under derivative causality

1. *all the unknown variables are reachable from the known ones,*
2. *the over-constrained and the just-constraint subsystems are causal,*
3. *the under-constrained subsystem is empty.*

Condition 1 expresses that there does not exist any subsystem whose behaviour is not reflected in the behaviour of the known variables, while conditions 2 and 3 express that all the variables can be matched using causal matchings, i.e. they belong to the intersection of at least as many manifolds as unknowns, and thus they are uniquely defined once the known variables are given.

Note that since derivative causality is invoked, condition 1 could as well have been stated as: all the variables $\boldsymbol{x}_a \cup \boldsymbol{x}_d$ are reachable from the known ones (since under derivative causality, $\dot{\boldsymbol{x}}_d$ can always be reached from $\boldsymbol{x}_d$).

Example 5.23 *Non-reachability*
Consider the following incidence matrix, in which the variable x_3 is not reachable from the output.

$\nearrow$	x_1	x_2	x_3	$\dot{x}_1$	$\dot{x}_2$	$\dot{x}_3$	u	y
c_1	1	1		1			1	
d_1	x			1				
c_2	1	1			1			
d_2		x			1			
c_3			1			1		
d_3			x			1		
m	1							1

The system equations associated with such a structure are obviously of the form

$$\begin{array}{llcl} \text{Subsystem 1} & \dot{x}_1 & = & g_1(x_1, x_2, u) \\ & \dot{x}_2 & = & g_2(x_1, x_2) \\ & y & = & h(x_1) \\ \text{Subsystem 2} & \dot{x}_3 & = & g_3(x_3) \end{array} \tag{5.23}$$

and it is seen that subsystem 2 can by no means be observable. □

Example 5.24 *Observability of a nonlinear system*
Consider the following non-linear dynamical system with two states, two input signals, one parameter and one sensor:

$$\begin{array}{ll} c_1: & \dot{x}_1 = (\theta - 1)x_2 u_1 \\ c_2: & \dot{x}_2 = u_2 \\ m: & y = x_1 \end{array}$$

This system is over-constrained, and it satisfies the three conditions of the above theorem. The following matching allows to compute the state.

↗	$\dot{x}_1$	$\dot{x}_2$	x_1	x_2	u_1	u_2	y
c_1	1			①	1		
c_2		1				1	
d_1	①		x				
d_2		①		x			
m			①				1

It can be noticed that x_2 can be reached from the known variables if and only if the matching (c_1, x_2) can be used, which means that the two conditions

$$u_1 \neq 0 \quad \text{and} \quad \theta \neq 1$$

simultaneously hold. If not, the system is not observable, since there is no matching by means of which x_2 could be computed under derivative causality. This example emphasises the fact that structural properties provide results which are valid for almost every value of the system parameters *and variables*. It can be noticed that the system being over-constrained, constraint c_2 is not matched and provides an input-output relation whose expression is

$$\frac{d}{dt}\frac{\dot{y}(t)}{u_1(t)} = u_2(t). \quad \square$$

5.5.3 Observability of linear systems

Let us consider the linear time-invariant deterministic system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \tag{5.24}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}, \tag{5.25}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are respectively of dimensions n and p. The state is observable if and only if the following condition holds

$$\exists s \in \mathcal{N} \quad \text{such that rank} \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \dots \\ \boldsymbol{CA}^s \end{pmatrix} = n, \tag{5.26}$$

for which a necessary condition is

$$\begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{C} \end{pmatrix} = n. \tag{5.27}$$

Note that (5.27) means, in structural terms, that the unknown variables $\boldsymbol{x}$ belong to a causal just-constrained or over-constrained subsystem, when derivative causality is imposed. Indeed, the structural graph is

$\nearrow$	$\dot{\boldsymbol{x}}$	$\boldsymbol{y}$	$\boldsymbol{x}$
$\boldsymbol{d}$	$\boldsymbol{I}$		x
$\boldsymbol{m}$		$\boldsymbol{I}$	$\boldsymbol{S}_C$
$\boldsymbol{c}$	$\boldsymbol{I}$		$\boldsymbol{S}_A$

where $\boldsymbol{d}$ are the mathematical constraints (which express that dots mean derivatives), $\boldsymbol{m}$ are the constraints from the measurement Eq. (5.25), and $\boldsymbol{c}$ are the system constraints (5.24). $\boldsymbol{S}_C$ and $\boldsymbol{S}_A$ are the structures associated with matrices $\boldsymbol{C}$ and $\boldsymbol{A}$. Since no variable in $\boldsymbol{x}$ can be matched from any constraint in $\boldsymbol{d}$, the system $(\boldsymbol{c} \cup \boldsymbol{m},\ \dot{\boldsymbol{x}} \cup \boldsymbol{x} \cup \boldsymbol{y})$ must indeed be over-constrained with respect to $\boldsymbol{x}$. It can be noted that this does not constitute a sufficient condition, because the system parameters might have values such that (5.26) – or (5.27) – is not satisfied.

Example 5.25 *Observability of linear systems*
Consider the non observable linear time-invariant system

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & d \\ a & b & e \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{5.28}$$

$$y = \begin{pmatrix} 0 & 0 & f \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{5.29}$$

where the parameters a, b, c, d, e, f can take any real value. Its structure is given by the incidence matrix

↗	$\dot{x}_1$	$\dot{x}_2$	$\dot{x}_3$	x_1	x_2	x_3	y
c_1	1					1	
c_2		1				1	
c_3			1	1	1	1	
d_1	1			x			
d_2		1			x		
d_3			1			x	
m						1	1

where the constraints c_1, c_2, c_3 are the system (5.28), the constraints d_1, d_2, d_3 express the derivative link between the x_1, x_2, x_3 and the $\dot{x}_1, \dot{x}_2, \dot{x}_3$ (remember that only derivative causality is allowed) and m is the measurement Eq. (5.29). This system can be decomposed into a just-constrained part $\mathcal{C}^0_{\mathcal{X}} = \{c_1, c_2, d_3, m\}$, $\mathcal{X}^0 = \{\dot{x}_1, \dot{x}_2, \dot{x}_3, x_3\}$ from which $\dot{x}_1$, $\dot{x}_2$, $\dot{x}_3$ and x_3 can be computed as functions of y, (for almost all values of the parameters), and an under-constrained one $\mathcal{C}^-_{\mathcal{X}} = \{c_3, d_1, d_2\}$, $\mathcal{X}^- = \{x_1, x_2\}$ in which x_1 and x_2 should both be computed from constraint c_3 (since they can be matched neither with d_1 nor with d_2). It can be checked that adding $\dot{y}$ and the associated constraints, the subsystem $(\{c_3, d_1, d_2\}, \{x_1, x_2\})$ remains under-constrained and that this will always be the case when more signals $y^{(i)}$ will be considered. This means that the information available from the sensor is enough to place (x_1, x_2) in a subspace of dimension one (since they are linked by one constraint which is known to be linear), but is not enough to compute the actual value of this vector. Indeed, the observability matrix

$$\begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & f \\ af & bf & ef \\ aef & bef & (ac+bd+e^2)f \end{pmatrix}$$

is not full rank, whatever the coefficients a, b, c, d, e, f, and it can be checked that no more than the linear form $ax_1 + bx_2$ can be determined from the observation $(y, \dot{y}, ...y^{(s)})$ for any $s \geq 1$.

Consider the same linear time-invariant system, in which the second state is now measured (the system is now observable)

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & d \\ a & b & e \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{5.30}$$

$$y = \begin{pmatrix} 0 & f & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \tag{5.31}$$

The structural graph is

↗	$\dot{x}_1$	$\dot{x}_2$	$\dot{x}_3$	x_1	x_2	x_3	y
c_1	1					1	
c_2		1				1	
c_3			1	1	1	1	
d_1	1			x			
d_2		1			x		
d_3			1			x	
m					1		1

and the following causal matching shows that all the components of the state can be computed from y and its derivatives

↗	$\dot{x}_1$	$\dot{x}_2$	$\dot{x}_3$	x_1	x_2	x_3	y
c_1	1					1	
c_2		1				①	
c_3			1	①	1	1	
d_1	①			x			
d_2		①			x		
d_3			①			x	
m					①		1

□

5.5.4 Graph-based interpretation and formal computation

Since an oriented graph can be associated with each matching, the observability property can be analysed from a graph-theoretical point of view. Let x be an observable variable. Then x can be matched with a constraint the input of which is either known or a set of observable variables. Repeating this argument, it follows that for x to be observable, it is necessary that there exists at least one subgraph (a set of alternated chains) which links this variable (call it the target) and the set of the known ones $\boldsymbol{u} \cup \boldsymbol{y}$, and that no non-observable variable acts as an input in any constraint of this subgraph. This means that the subgraph with the observable target x may contain algebraic loops, but it does not contain any differential loop.

The constraints along the alternated chains show the computations which are to be performed in order to compute x (they could be automatically analysed in order to provide the formal expression of x). Indeed, the crossing of a simple algebraic constraint means that the matched variable is computed as a function of the non-matched ones. An algebraic loop shows that a system of simultaneous constraints has to be solved, providing as a solution all the variables matched within the loop. The crossing of a derivative constraint means that the non-matched variable has to be derivated in order to obtain the matched variable (remember that only derivative causality is allowed).

The number of derivative constraints which are crossed between a given input and the target shows the maximum order of derivation needed on this input for computing this target.

Note that this interpretation expresses that x belongs to a just or an over-constrained causal subsystem. Indeed, if x were to belong to an under-constrained subsystem, the corresponding subgraph would have less constraints than variables, i.e. some unknown variables would be input signals to constraints while being output of no other constraint (i.e. being not matched, thus non-observable). Figure 5.19 shows the two graphs associated with the linear systems (5.28), (5.29) and (5.30), (5.31) which are respectively non-observable and observable. It can be seen that in the first case, either x_2 or x_1 stands as an unknown input of constraint c_3 while in the second case, both can be matched thus providing all the states with known predecessors at some level.

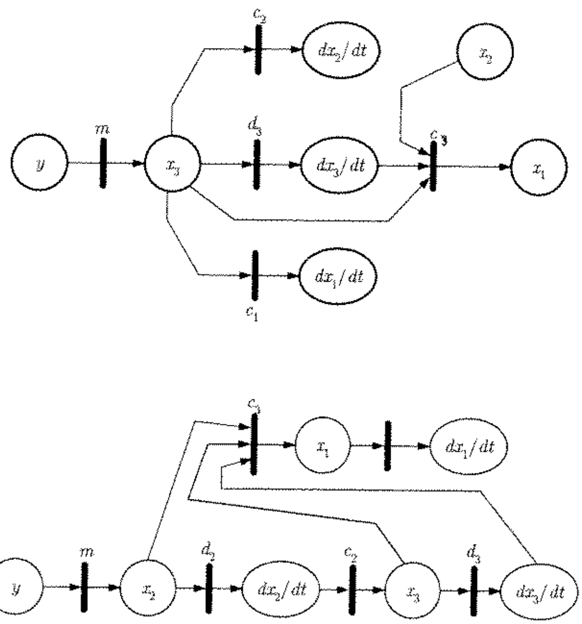


Fig. 5.19. Graph-based interpretation of the observability property

When different estimation subgraphs with the same target exist, they provide different computation schemes for the same variable. This feature is of interest when monitorability and reconfigurability are considered.

5.6 Monitorability

A system is said to be *monitorable* if it can be determined, using only (trajectories of) known variables, whether the system constraints are satisfied or not. This section is concerned with the analysis of system monitorability, and with the design of fault detection and isolation algorithms based on *Analytical Redundancy Relations* (ARRs). Analytical redundancy-based fault diagnosis tries to identify faults by comparing the actual behaviour of the system, which is observed through the time evolution of the known variables, with the theoretical behaviour described by the system constraints. This comparison can be performed only if some redundancy exists in the system. ARRs are the constraints that express this redundancy. In this section, the analytical redundancy relation-based approach to fault diagnosis is first briefly recalled and stated in the frame of structural analysis, leading to characterise the structurally monitorable part of the system. The problem of designing robust and structured residuals is then addressed.

5.6.1 Analytical redundancy-based fault detection and isolation

Analytical redundancy relations are static or dynamical constraints which link the time evolution of the known variables when the system operates according to its normal operation model. Once ARRs are designed, the fault detection procedure checks at each time whether they are satisfied or not, and when not, the fault isolation procedure identifies the system component(s) which is (are) to be suspected. The existence of ARR is thus a prerequisite to the design of fault diagnosis procedures. Moreover, in order for the fault diagnosis procedure to work properly, ARR should have the following properties:

- Robust, i.e. insensitive to unknown input and unknown parameters. This insures that they are satisfied when no fault is present, so that false alarms are not issued by the fault diagnosis algorithm.
- 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 a subset of the ARRs are not satisfied, thus allowing to recognise (from the subset of satisfied and the subset of not satisfied ARR), the fault which occurred.

Faults. Analysing the fault diagnosis possibilities of a system needs faults to be precisely defined. In structural analysis, a fault is defined as *a change in some constraint*. Indeed, a system is the interconnection of a number of components, each of them being described by its behaviour model in normal operation. Let $\{COMP_i, i \in \Sigma\}$ be the set of the system components. Each of them is a subsystem $(\phi_i, Q(\phi_i))$ which imposes the set of constraints ϕ_i to the system variables $Q(\phi_i)$, where $Q(\phi_i) \cap \mathcal{X}$ are unknown (non measured state variables, unknown input, unknown parameters) while $Q(\phi_i) \cap \mathcal{K}$ are known (input, output, known parameters). A fault in component $COMP_i$ is defined as a change in at least one of the constraints $\varphi \in \phi_i$ (the notation φ is used – as a mnemonic for "fault" – instead of c which was a mnemonic for "constraint"). Note that this allows to consider different fault modes associated with the same component (each subset of ϕ_i can in fact be considered as a fault mode of $COMP_i$). Note also that since only the structure is of interest, there is no need to define, nor to model the nature of the change (e.g. using additive or multiplicative fault models).

Example 5.26 *An insulated pipe*
Consider an insulated pipe, and suppose one is interested in modelling the mass and the heat transfers. A simple model of the pipe is given by the two constraints

$$\begin{aligned} \varphi_1 &: \quad q_i(t) - q_o(t) = 0 \\ \varphi_2 &: \quad q_i(t)\,\theta_i(t) - q_o(t)\,\theta_o(t) = 0, \end{aligned}$$

where q_i (respectively q_o) is the input (respectively the output) massic flow of the (incompressible) fluid, and θ_i (respectively θ_o) is the input (respectively the output) fluid temperature. A defect in the insulation would obviously result in φ_2 being violated, while a leak in the pipe would be modeled by φ_1 and φ_2 being violated. □

Considering the whole set of components, the overall system is $(\mathcal{C}, \mathcal{Z})$, where

$$\begin{aligned} \mathcal{C} &= \cup_{i \in \Sigma}\, \phi_i \\ \mathcal{Z} &= \cup_{i \in \Sigma}\, Q(\phi_i). \end{aligned}$$

Direct redundancy. Consider any constraint $\varphi \in \mathcal{C}_{\mathcal{K}}$ (remember that $\mathcal{C}_{\mathcal{K}}$ is the subset of constraints such that $Q(\mathcal{C}_{\mathcal{K}}) \subseteq \mathcal{K}$), and let $COMP$ be the component to which φ belongs. This constraint is an ARR since it links only known variables, and it can be checked in real time if it is satisfied or not, by taking the numerical values of the known variables, putting them into constraint φ, and testing whether the result is ZERO or not. Note that when the constraint is not satisfied, it can be concluded that the system is not in normal operation, while when the constraint is satisfied it can only be said that the normal operation hypothesis is not contradicted (or falsified) by the values of the observations.

In practical situations, variables are not very precisely known, measurements are corrupted by noise, and models only approximate the system actual behaviour, thus the obtained value will never be exactly zero, even in normal operation. Let $r_\varphi(\mathcal{K})$ be the obtained value. $r_\varphi(\mathcal{K})$ is called the *residual* associated with ARR φ, and fault detection boils down to decide whether it is small enough so that the ZERO hypothesis can be accepted. Fault isolation obviously follows fault detection since only a fault in component $COMP$ could cause constraint φ not to be satisfied.

In all systems, the control algorithms are direct ARR, since the subset $\mathcal{C}_\mathcal{K}$ includes the constraints which describe them. Hence, they can be used to check whether the controller is working properly. Although this might be of practical interest, such direct redundancy relations are of little interest as far as structural analysis is concerned, because the result is obvious. Therefore, the aim of the following is to find ARRs in the subsystem $(\mathcal{C}_\mathcal{X}, \mathcal{Z})$ which includes unknown variables.

Deduced redundancy. Consider some constraint $\varphi \in \mathcal{C}_\mathcal{X}$ and again let $COMP$ be the component to which φ belongs. Let $\mathcal{X}_\varphi = Q(\varphi) \cap \mathcal{X}$ be the subset of unknowns which appear in constraint φ, and suppose

$$\mathcal{X}_\varphi \subseteq \mathcal{X}_{obs}, \tag{5.32}$$

where $\mathcal{X}_{obs}$ is the subset of the observable variables. Then, any variable $x \in \mathcal{X}_\varphi$ can be expressed as a function of the known ones (including eventually their derivatives), using the normal operation system model (this results from the existence of one or several alternated chains between the known variables and the target x). Suppose that there exists at least one alternated chain with target x which does not cross constraint φ (this means that even when constraint φ is removed, x can still be matched and computed as a function of the known variables, which indicates that constraint φ belongs to an over-constrained subsystem, as it will be seen later). When this is true, this alternated chain can be used to compute x as a function of the known variables, and one can put the obtained expression into φ, which obviously produces an ARR. The associated residual $r_\varphi(\mathcal{K})$ should be ZERO when the system operates properly, which is used, as previously, for fault detection. However, fault isolation will be slightly different since the residual associated with φ will be non ZERO not only when $COMP$ is not performing well, but also when the actual values of the $\mathcal{X}_\varphi$ variables are different from those computed from the observations via the normal operation model. This may happen when the fault changes some constraint which belongs to an alternated chain whose target is in $\mathcal{X}_\varphi$. The conclusion is that when $r_\varphi(\mathcal{K})$ is non ZERO, there is an associated set of components to be suspected instead of a single one[2]. It can be easily determined from the graph-based interpretation.

Example 5.27 *Single tank*

Consider the single tank whose structure graph is shown in Fig. 5.4. Obviously, there are two redundancy relations. The first one is given by constraint c_5 and is of no interest since it is a direct redundancy relation which only duplicates the control algorithm. The second one is given by c_2 which should be satisfied when the system operates normally, and which will be false if one of the constraints $\{c_1, c_2, c_3, c_4\}$ is not satisfied (c_6 is a mathematical constraint which is not linked with any hardware or software component and thus it cannot be faulty). □

[2] This set is called the *structure of the residual* in the control community and it is called a *conflict* in the Artificial Intelligence community

5.6.2 Structurally monitorable subsystems

Unfortunately, not every fault can be detected. Indeed, it is easily seen that when (5.32) does not hold, it is impossible to obtain any ARRs using constraint φ. Thus, any fault which would change constraint φ could not be detected.

Definition 5.12 (Structurally monitorable subsystem)
The structurally monitorable part of the system is the subset of the constraints such that there exists ARRs which are structurally sensitive to their change.

Then, from above, it can be characterised by the following theorem:

Theorem 5.3 (Monitorability)
Two equivalent necessary conditions for a fault φ to be monitorable are:

(i) $\mathcal{X}_\varphi$ is structurally observable - according to (5.32) - in the system $(\mathcal{C}\backslash\{\varphi\}, \mathcal{Z})$,
(ii) φ belongs to the structurally observable over-constrained part of the system $(\mathcal{C}, \mathcal{Z})$.

Indeed, let $(\mathcal{C}_\mathcal{X}, \mathcal{X})$ be a structurally observable over-constrained subsystem, then there exists a subset $\mathcal{S}_\mathcal{X} \subset \mathcal{C}_\mathcal{X}$ of $n = |\mathcal{X}|$ constraints which (from a structural point of view) can be solved uniquely for the variables $\mathcal{X}$ (notation $\mathcal{S}$ is used as a mnemonic for Solve). These variables can thus be computed as functions of the known variables $\mathcal{K}$. Putting the obtained values into the remaining constraint set $\mathcal{R}_\mathcal{X} = \mathcal{C}_\mathcal{X}\backslash\mathcal{S}_\mathcal{X}$ (notation $\mathcal{R}$ is used as a mnemonic for Remaining, or Redundant), one obtains $|\mathcal{C}_\mathcal{X}| - |\mathcal{X}|$ relations which link only known variables and which are, therefore, redundancy relations. For a more convenient notation the function

$$\mathcal{X} = \Gamma_\mathcal{X}(\mathcal{K}) \tag{5.33}$$

is introduced for the computation of the unknown variables, leading to expressing the set of constraints $\mathcal{C}_\mathcal{X}$ under the equivalent form

$$\begin{aligned} \mathcal{S}_\mathcal{X} : &\quad \mathcal{X} - \Gamma_\mathcal{X}(\mathcal{K}) = 0 \\ \mathcal{R}_\mathcal{X} : &\quad [\mathcal{C}_\mathcal{X}\backslash\mathcal{S}_\mathcal{X}] \circ \Gamma_\mathcal{X}(\mathcal{K}) = 0, \end{aligned} \tag{5.34}$$

where $\circ$ means the substitution of $\mathcal{X}$ by $\Gamma_\mathcal{X}(\mathcal{K})$.

It can be noted that in general, several different complete matchings can be performed in a given causal over-constrained subsystem, thus providing different means of computing the unknown variables $\mathcal{X}$ from the known ones. This fact will be used when fault-tolerant observation schemes will be considered, but it can also provide another interpretation of redundancy, since obviously the unknown variables $\mathcal{X}$ have to be the same for all matchings. For example suppose two matchings exist such that $\mathcal{X}$ is associated with

$\mathcal{S}_{\mathcal{X}} \subset \mathcal{C}_{\mathcal{X}}$ in the first one, providing $\mathcal{X} = \Gamma_{\mathcal{X}}(\mathcal{K})$, and with $\mathcal{P}_{\mathcal{X}} \subset \mathcal{C}_{\mathcal{X}}$ in the second one, providing $\mathcal{X} = \Lambda_{\mathcal{X}}(\mathcal{K})$. The redundancy relations

$$\Gamma_{\mathcal{X}}(\mathcal{K}) - \Lambda_{\mathcal{X}}(\mathcal{K}) = 0$$

directly follows from the fact that the two results should obviously be the same.

Example 5.28 *Sensor redundancy*
The simplest illustration of this idea is provided by sensor hardware redundancy. Suppose that two sensors measure the same unknown variable x. The measurement equations are given by

$$\begin{array}{lll} \text{Sensor 1} & c_1: & y_1 - x - \varepsilon_1 = 0 \\ \text{Sensor 2} & c_2: & y_2 - x - \varepsilon_2 = 0, \end{array}$$

where ε_1 and ε_2 denote measurement noise with known distribution. The structure graph has the incidence matrix

	known				unknown
↗	y_1	y_2	ε_1	ε_2	x
c_1	1		1		1
c_2		1		1	1

ε_1 and ε_2 are here considered as known variables because their probability distribution is known. This system is over-constrained with $\mathcal{C}_{\mathcal{X}} = \{c_1, c_2\}$ and $\mathcal{X} = \{x\}$. The unknown x can be matched with each of the two constraints and, hence, be calculated by each of the sensor equations. This is not only true from the structural point of view because x can also be calculated numerically if $\frac{dc_1}{dx}$ and $\frac{dc_2}{dx}$ are both non-zero. Otherwise at least one of the sensors would be completely useless.

For the matching

	known				unknown
↗	y_1	y_2	ε_1	ε_2	x
c_1	1		1		①
c_2		1		1	1

the oriented graph is given by Fig. 5.20, in which the unknown x is computed by

$$x = \gamma_1(y_1, \varepsilon_1)$$

and c_2 is used as a redundancy relation which can be written as

$$c_2(\gamma_1(y_1, \varepsilon_1), y_2, \varepsilon_2) = 0.$$

Choosing the second possible matching

	known				unknown
↗	y_1	y_2	ε_1	ε_2	x
c_1	1		1		1
c_2		1		1	①

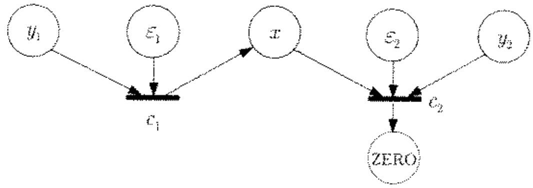


Fig. 5.20. Oriented structure graph for the sensor redundancy investigations

provides

$$x = \gamma_2(y_2, \varepsilon_2)$$

and the redundancy relation

$$c_1\left(y_1, \varepsilon_1, \gamma_2\left(y_2, \varepsilon_2\right)\right) = 0.$$

Since two matchings exist, the remark that the (same) value of x can be computed either from the first or from the second one leads to the redundancy relation, which takes the form

$$\gamma_1\left(y_1, \varepsilon_1\right) - \gamma_2\left(y_2, \varepsilon_2\right) = 0. \ \square$$

5.6.3 Design of analytic redundancy relations

Redundancy relations are subgraphs of the structure graph, which are associated with complete causal matchings of the unknown variables associated with the over-constrained subsystem of the reduced bi-partite graph. Redundancy relations are composed of alternated chains, which start with known variables and which end with non-matched constraints whose output is labelled ZERO. Designing a set of residuals calls for building maximal matchings 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. Algorithms which find maximal matchings have been previously presented (cf. Section 5.3.5), and some hints have been given, using the tank and the sensor hardware redundancy examples, on the residuals design procedure. Let us now give a complete illustration, first using the simple single tank system, and then giving a larger practical example with the two-tank system.

Example 5.29 *Ranking the constraints of the single tank*
Consider once more the single tank example, and recall the incidence matrix of its reduced structure graph from Example 5.6:

↗	h	$\dot{h}$	q_i	q_0
c_1		1	1	1
c_2			1	
c_3	1			1
c_4	1			
c_6	**x**	1		

The result of the ranking algorithm is shown in the following table and in Fig. 5.21. The matching is identical with the second matching on Example 5.6. Note that a new column has been introduced to mark constraints which have the output ZERO. Since ZERO is not a variable, it may be matched several times.

	unknown				Ranking	
↗	h	$\dot{h}$	q_i	q_0	ZERO	Rank
c_1		1	1	1	①	2
c_2			①			0
c_3	1			①		1
c_4	①					0
c_6	**x**	①				1

Sorted according to the rank, the following constraint set is obtained:

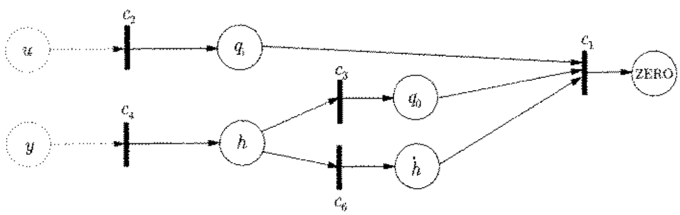


Fig. 5.21. Ranking for the single tank system

Rank	Constraint	Output
0	c_2 c_4	$q_i(t)$ $h(t)$
1	c_3 c_6	$q_o(t)$ $\dot{h}$
2	c_1	ZERO

If the reduced structure graph is redrawn according to the ranking of the constraints, Fig. 5.21 is obtained. The figure shows how the internal variables q_i, h, q_0 and $\dot{h}$ can be successively determined. The constraints are ordered according to their associated rank. Finally, the contraint c_1 is used to test wether the variables are consistent with the model.

As all constraints are ranked, the system is fully observable and monitorable. By solving the constraints for the matched variables, the following equations are obtained:

$$\begin{aligned}
c_2: &\quad q_i(t) = \alpha \cdot u(t) \\
c_4: &\quad h(t) = y(t) \\
c_3: &\quad q_0(t) = k\sqrt{h(t)} \\
c_6: &\quad \dot{h}(t) = \frac{d}{dt}h(t). \\
c_1: &\quad 0 = \dot{h}(t) + q_o(t) - q_i(t)
\end{aligned}$$

These equations can be simplified to obtain the redundancy relations:

$$c_1: \quad 0 \quad = \frac{d}{dt}y(t) + k\sqrt{y(t)} - \alpha u(t).$$

Note that all variables on the right-hand side of the two equations are known. Hence, these equations can be tested for given measurements u and y which are marked in Fig. 5.21 to illustrate this fact. □

Example 5.30 *The two-tank system*

As a larger practical example the two-tank system introduced in Section 2.1 will be considered. u is the known control input and q_m the measured outflow. The following equations lead to the structure graph of the system (Fig. 5.22):

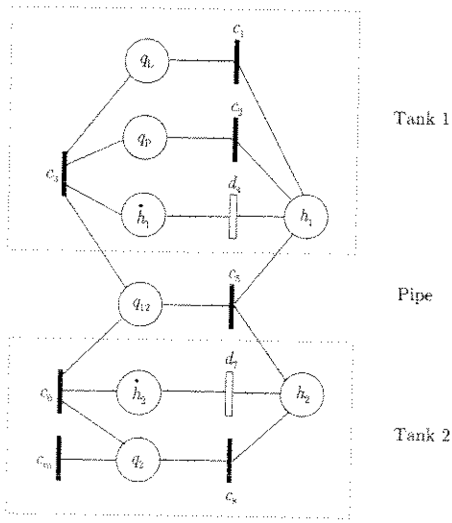


Fig. 5.22. Structure graph of the two-tank system

$$
\begin{aligned}
c_1 &: \quad q_L = 0 \\
c_2 &: \quad q_P = u \cdot f(h_1) \\
c_3 &: \quad \dot{h}_1 = \frac{1}{A}\,(q_P - q_L - q_{12}) \\
d_4 &: \quad \dot{h}_1 = \frac{d}{dt}\,h_1 \\
c_5 &: \quad q_{12} = k_1 \sqrt{h_1 - h_2} \\
d_6 &: \quad \dot{h}_2 = \frac{d}{dt}\,h_2 \\
c_7 &: \quad \dot{h}_2 = \frac{1}{A}\,(q_{12} - q_2) \\
c_8 &: \quad q_2 = k_2 \sqrt{h_2} \\
c_m &: \quad q_m = k_m\, q_2 .
\end{aligned}
$$

A, k_1, k_2 and k_m are known parameters. In the structure graph the constraints c_1, c_2, c_3 and d_4 representing the Tank 1 are separated from constraints c_6, d_7, c_8 and c_m describing the Tank 2.

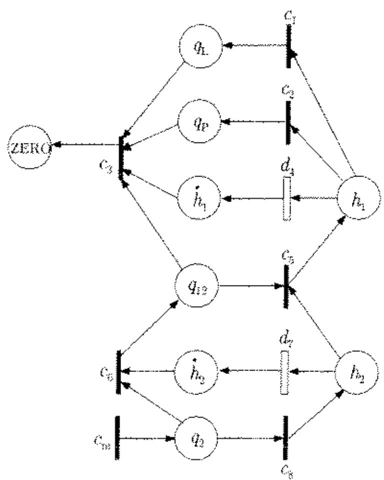


Fig. 5.23. Oriented graph of the two-tank system

The following matching is found using the ranking algorithm, where the last column shows the rank of the constraints obtained:

$$
\begin{aligned}
q_L &= 0\\
q_P &= u \cdot f(h_1)\\
0 &= -q_L + q_p - q_{12} - A\dot{h}_1\\
\dot{h}_1 &= \tfrac{d}{dt}\, h_1\\
h_1 &= h_2 + \left(\tfrac{q_{12}}{k_1}\right)^2\\
q_{12} &= A\dot{h}_2 + q_2\\
\dot{h}_2 &= \tfrac{d}{dt} h_2\\
h_2 &= \left(\tfrac{q_2}{k_2}\right)^2\\
q_2 &= \tfrac{q_m}{k_m}
\end{aligned}
$$

↗	q_L	q_P	$\dot{h}_1$	h_1	q_{12}	$\dot{h}_2$	h_2	q_2	R
c_1	①			1					5
c_2		①		1					5
c_3	1	1	1		1				6
d_4			①	1					5
c_5				①	1		1		4
c_6					①	1		1	3
d_7						①	1		2
c_8							①	1	1
c_m								①	0

The equations shown on the left are already solved for the matched variable. The corresponding oriented graph is shown in Fig. 5.23. Simplifying these equations results in the following redundancy relation:

$$0 = u(t) \cdot \boldsymbol{f}\left(h_1(t)\right) - A\dot{h}_2 + \frac{q_m(t)}{k_m} - A\dot{h}_1 \tag{5.35}$$

with

$$h_1(t) \quad = \quad h_2(t) - \left(\frac{A\dot{h}_2}{k_1} + \frac{q_m(t)}{k_m\, k_1}\right)^2 \tag{5.36}$$

$$h_2(t) \quad = \quad \left(\frac{q_m(t)}{k_m\, k_2}\right)^2 . \tag{5.37}$$

Equations (5.35) – (5.37) can be used to monitor the two-tank system. By using Eq. (5.37), $h_2(t)$ and, hence, $\dot{h}_2$ can be determined for given measurement $q_m(t)$. Then Eq. (5.36) yields $h_1(t)$ and $\dot{h}_1$. Finally, Eq. (5.35) is checked for known $u(t)$, $q_m(t)$ and for $h_1(t), \dot{h}_1$ and $\dot{h}_2$ just obtained.

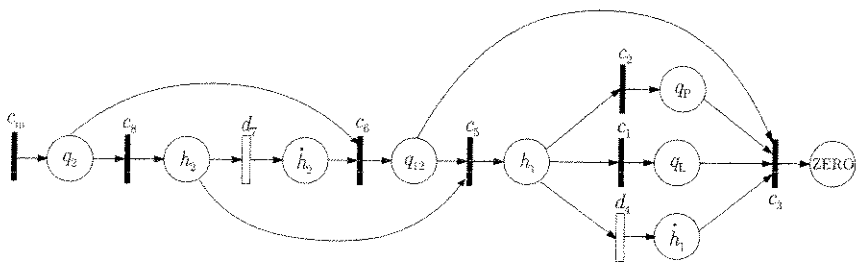


Fig. 5.24. Graph showing the order in which the unknown variables can be determined for given q_m

After redrawing the structure graph, Fig. 5.24 is obtained. This graph shows in which order the constraints can be used to determine all internal variables for given measurement q_m. Finally, constraint c_3 is used to test the consistency of the variables with the model.

A simulation result is shown in Fig. 5.25 which shows from top to bottom the signals $u(t)$, $x_1(t)$ and $x_2(t)$, the measurement $q_m(t)$ and the right-hand side of Eq. (5.35). Note that the states are reconstructed very nicely. The residual (which is in fact the loss of water through the leak) shows the occurrence of the fault very precisely and without any delay. The little spike at time 155 s is due to the reversal of the flow direction in the connection pipe, which represents a singular point in the linearised system.

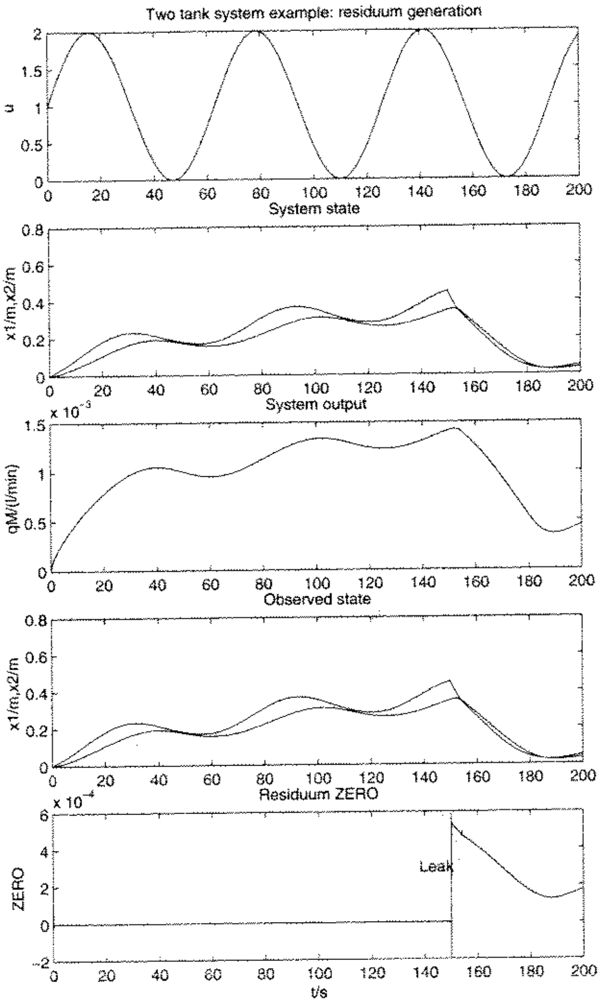


Fig. 5.25. Simulation results of the two-tank system. From top to bottom: input u; tank levels h_1, h_2; measured q_m; reconstructed levels h_1, h_2; right-hand side of Eq. (5.35).

Of course, the parity relation is very sensitive to measurement noise due to the two differentiations. Even a very small noise can disturb the fault detection scheme heavily as shown in Fig. 5.26. Although the noise is not visible in the measured signal q_m, its differentiation is large and calls for the use of statistical deci making algorithms in order to make fault detection possible.

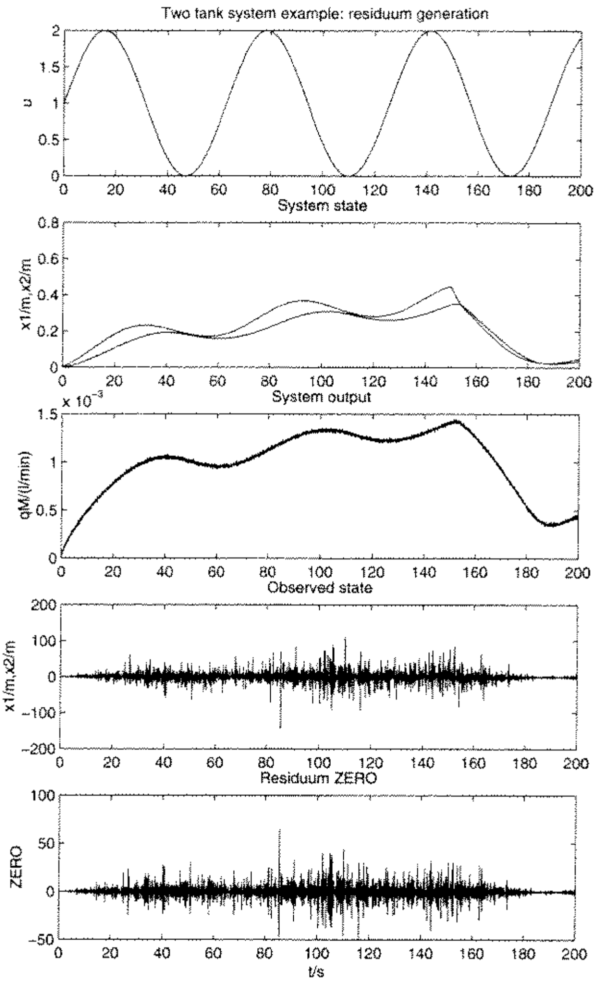


Fig. 5.26. Simulation result similar to Fig. 5.25 but with a small noise of the measured signal q_m

Alternatively, if a filter is included when determining $\dot{h}_1$ and $\dot{h}_2$ from h_1 or respectively, the residuals increase slightly delayed after the fault has occurr Due to the phase lag introduced by the filter, the residual given by Eq. (5.35 no longer zero for the faultless case. Using digital filters without phase lag m precise results are possible, but these filters delay the calculation of the resid further. □

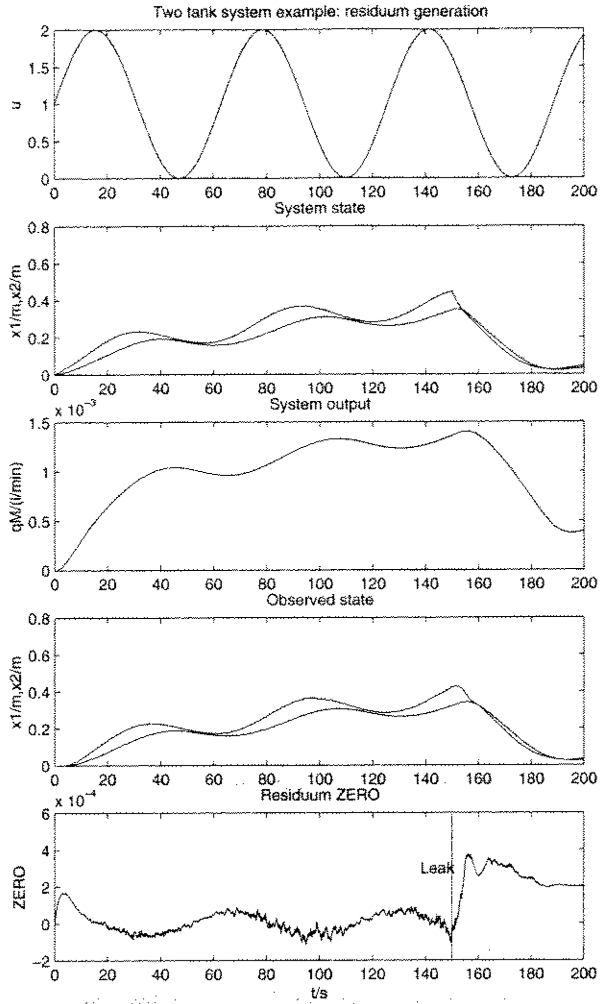


Fig. 5.27. Simulation result similar to Fig. 5.26 but with filtered measurement; q_M is the filtered system output.

5.6.4 Parity space and observer-based approaches

Fault diagnosis based on redundancy relations is connected with the parity space approach. Indeed, redundancy relations are the result of the elimination of the unknown variables from the original system of constraints. In the tank example this is performed by computing the unknown variables $\{h,\ \dot{h},\ q_0,\ q_i\}$ as functions of the known ones $\{u,\ y\}$. Note that this elimination is performed through a first order derivation of the known signal y as it can be seen from the presence of constraint c_6 in the subgraph. This is indeed a characteristic (and a drawback) of analytic redundancy, since in practice signal derivation is very sensitive to measurement noise and thus calls for sophisticated algorithms. The reason why derivative expressions are found is that derivative causality is imposed by the fact that the initial state is not known.

In observer based approaches, there is no need for eliminating the unknown initial state through derivation, at the price of approximating the system output by integration. The following example illustrates the different mechanisms of analytic redundancy-based and observer based-approaches.

Example 5.31 *Parity-space and observer-based fault detection of a first-order system*
Consider the first-order dynamical system

$$\begin{aligned} c_1 &: \dot{x} - ax - bu = 0 \\ c_2 &: \dot{x} - \frac{dx}{dt} = 0 \\ c_3 &: y - cx = 0. \end{aligned}$$

Its structure graph and a possible parity space redundancy relation are illustrated Fig. 5.28. The constraints c_2 and c_3 are used to determine x and then $\dot{x}$ from the measurement y. c_1 is used to test the consistency of the obtained results with the system dynamics for given u.

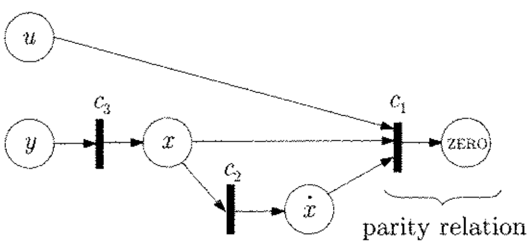


Fig. 5.28. Parity relation of a first-order system

Suppose now that the output y is estimated by a Luenberger observer in order to test the observed result y_{obs} with the measurement y. The following five constraints are added to $\{c_1, c_2, c_3\}$:

$$\begin{aligned} c_4 &: \dot{x}_{obs} - ax_{obs} - bu - ke_y = 0 \\ c_5 &: e_y - y + y_{obs} = 0 \\ c_6 &: \dot{x}_{obs} - \frac{dx_{obs}}{dt} = 0 \\ c_7 &: y_{obs} - cx_{obs} = 0 \\ c_8 &: \lim_{t \longrightarrow \infty} e_y = 0, \end{aligned}$$

where x_{obs} and y_{obs} are the observer state and output, and constraint c_8 results from the choice of the observer gain k such that the output estimation error tends to zero.

The structure graph of the system and the redundancy relations are given by Fig. 5.29. Note that x_{obs} belongs to the known variables and that integral causality can be used because the initial value of x_{obs} is known, which is symbolised on the figure by the INIT arrow. Note also that the subgraph which is enclosed in the dashed line is used in an implicit way. The reason why c_8 holds is that the output of the two dynamical systems $\{c_1, c_2, c_3\}$ and $\{c_4, c_5, c_6, c_7\}$ converge to each other provided that the observer gain k has been chosen appropriately.

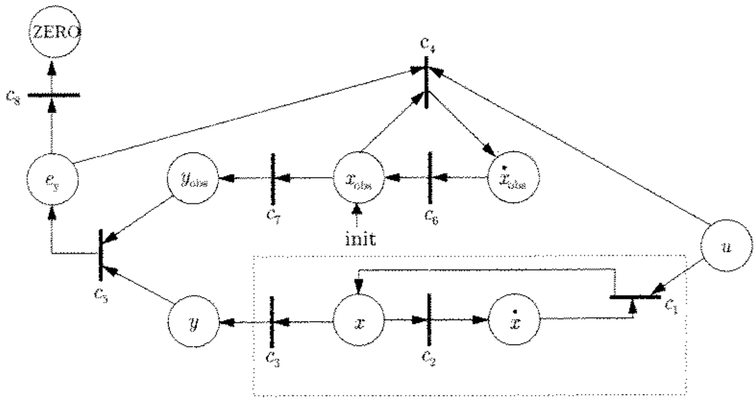


Fig. 5.29. Structure graph of a first-order system together with the Luenberger observer

The comparison of both approaches shows that the parity-space method is simpler but necessitates to determine the time derivative of x. The observer-based method does not use differential causality. □

5.6.5 Design of robust and structured residuals

It is worth to notice that the residual design algorithms can be performed on any subgraph of the system structure graph. The reason why one should consider only a subgraph is that this allows to find redundancy relations with specific properties, namely robustness and structuring.

Robust residuals. Robust residuals are insensitive to unknown input and unknown or uncertain parameters. Therefore, they are satisfied when no fault is present, whatever the value of the unknown input or uncertain parameters. Note that the robustness problem is automatically solved in structural analysis, using the decoupling approach presented in Chapter 6, since it exhibits ARRs which are, by definition, only dependent on known variables. Unknown variables which affect the structurally monitorable subsystem are eliminated so that no residual can depend on them; when unknown variables cannot be eliminated, the part of the system they affect is not monitorable. When uncertain parameters are present, it should not be used in ARRs, because such ARRs might generate false alarms or missed detections. The solution is simply to design the fault diagnosis system considering them as unknown variables (this boils down to use the subset of residuals in which no uncertain parameter intervenes). Of course, the number of ARRs will in that case be smaller.

Structured residuals. As defined above, the structure of a residual is the set of the constraints which can be suspected when this residual is not ZERO. Let $\mathcal{R}$ be a set of residuals, and let $\Phi(r) \in 2^{\mathcal{C}}$ be the structure of residual $r \in \mathcal{R}$. This means that r is expected to be non-ZERO when at least one of the constraints in $\Phi(r)$ is faulty (in fact r will be non-ZERO only for detectable faults). Similarly, when some constraint $\varphi \in \mathcal{C}$ is faulty, then all the residuals whose structure contains φ are expected to be non-ZERO. The pattern of ZERO and non-ZERO residuals associated with a given fault is called its signature. Faults which have different signatures are isolable from each other, while faults which share the same signature are non-isolable. Indeed, let $\mathcal{R} = \mathcal{R}_0(t) \cup \mathcal{R}_1(t)$ be the decomposition of the set of residuals provided at some given time t by the decision procedure, where $\mathcal{R}_0(t)$ is the subset of the ZERO residuals and $\mathcal{R}_1(t)$ is the subset of non-ZERO ones. The subset of *suspected* constraints (the constraints which might be non satisfied) at time t is given by

$$\mathcal{C}_{susp}(t) = \cap_{r \in \mathcal{R}_1(t)} \Phi(r).$$

Note that it is possible to define the subset of *exonerated* constraints (the constraints which are certainly satisfied) at time t by

$$\mathcal{C}_{exo}(t) = \cup_{r \in \mathcal{R}_0(t)} \Phi(r),$$

but one must be aware that this supposes all faults to be detectable. Indeed, exoneration is based on the assumption that if a constraint is not satisfied then it will necessarily show through the residuals whose structure it belongs to. The diagnosis at time t is

$$\mathcal{C}_{diag}(t) = \mathcal{C}_{susp}(t) \setminus \mathcal{C}_{exo}(t).$$

In order to obtain good isolability properties, it may be of interest to design residuals with given structure. Suppose one wishes to design residuals which are insensitive to the faults of a subset of constraints $\mathcal{C}'$ and are sensitive to the faults of the subset of constraints $\mathcal{C} \setminus \mathcal{C}'$. A direct approach to do this is to consider only the system $(\mathcal{C} \setminus \mathcal{C}', \mathcal{Z})$ in the design process. However, from the structural monitorability condition, it is seen that the residuals can be made sensitive only to the faults in the monitorable subsystem of $(\mathcal{C} \setminus \mathcal{C}', \mathcal{Z})$, which may be smaller than that of $(\mathcal{C}, \mathcal{Z})$, since the former contains less constraints.

Example 5.32 *Structured redundancy relations for the two-tank system*
Consider the two-tank system and assume that the flow q_{12} between the two tanks can be measured in addition to the input u and the outflow q_m, which leads to the additional measurement constraint

$$m_1: \quad q_{12} = q_{12,m}.$$

The following matching shows that the system is over-constrained. Therefore, the two remaining conditions can be used as parity conditions, as marked here:

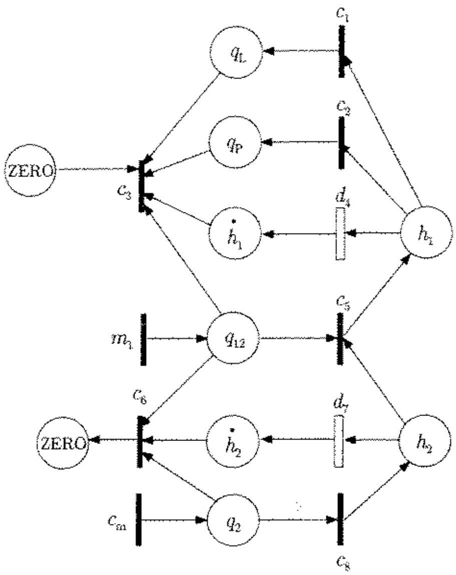


Fig. 5.30. Oriented graph, in which the arrows indicate the order of matching

Constraint	$\nearrow$	q_L	q_P	$\dot{h}_1$	h_1	q_{12}	$\dot{h}_2$	h_2	q_2	R
$q_L = 0$	c_1	①			1					3
$q_P = f(h_1)$	c_2		①		1					3
$0 = -q_L + q_p - q_{12} - A\dot{h}_1$	c_3	1	1	1		1				4
$\dot{h}_1 = \frac{d}{dt} h_1$	d_4			①	1					3
$h_1 = h_2 + \left(\frac{q_{12}}{k_1}\right)^2$	c_5				①	1		1		2
$0 = q_{12} - A\dot{h}_2 - q_2$	c_6					1	1		1	4
$\dot{h}_2 = \frac{d}{dt} h_2$	d_7						①	1		2
$h_2 = \left(\frac{q_2}{k_2}\right)^2$	c_8							①	1	1
$q_{12} = q_{12,m}$	m_1					①				0
$q_2 = \frac{q_m}{k_m}$	c_m								①	0

This matching results in the oriented graph shown in Fig. 5.30. Following the orientation of the edges, it is easy to see that the first parity relation depends on the variables $\{u, q_L, q_P, \dot{h}_1, h_1, q_{12}, q_{12,m}, h_2, q_2, q_m\}$ only, while the second depends on $\{q_{12}, q_2, h_2, \dot{h}_2, q_{12,m}, q_m\}$. So these two conditions can be used to selectively monitor different parts of the system (Tank 1 and Tank 2, to be precise). Only a fault in the connection flow q_{12} or its measurement would affect both constraints.

By signification the following two residuals are obtained:

$$0 = u \cdot f(h_1) - q_{12,m} - A\dot{h}_1$$

$$0 = q_{12,m} - A\dot{h}_2 - \frac{q_m}{k_m}$$

with

$$h_1 = h_2 - \left(\frac{q_{12,m}}{k_1}\right)^2$$

$$h_2 = \left(\frac{q_m}{k_m k_2}\right)^2. \quad \square$$

5.6.6 Fault propagation and alarm filtering

Using the matching procedure, the redundancy relations appear on the structural graph as subgraphs whose output is a ZERO variable and in which the whole input is known.

The analytical redundancy-based fault detection procedure consists of inserting the actual values of the known variables into those redundancy relations and checking in real time the truth value of the output node. The fault isolation procedure consists of identifying the subset of the fired (non ZERO) output nodes.

However, in many existing systems, fault diagnosis procedures have not been designed in such a consistent way, and most industrial applications use supervision systems which create alarms through limit checking procedures applied to measured variables. The problem is that those alarms are correlated, since a single fault in one component may cause many measured variables to trespass their respective thresholds. The consequence is that when some fault occurs, hundreds of alarms may fire almost simultaneously (this is called the "Christmas tree" syndrome), or alarms may be fired a long time after the fault occurrence, making it difficult for operators to isolate the origin of the problem.

Considering the structural graph may allow to analyse the consequences of certain faults, in terms of alarm firing. Indeed, when a fault occurs in a component, some among the constraints associated with it (according to the kind of the fault) will no longer be satisfied. The measured variables which are connected through an alternated chain with those constraints may get abnormal values and thus may trespass their associated thresholds. So, each fault can be associated, through the analysis of the structural graph, with a subset of alarms which could be fired when this fault occurs, thus providing a means to assign a limited number of possible root causes to situations in which numerous alarms are fired.

5.7 Controllability

Controllability is a property which only concerns that part of the model which describes the links between the unknown variables and the input variables, independently of the fact that some unknown variables might be measured or not. Thus, it can be analysed from the system bi-partite graph in which the measurement constraints have been removed. Roughly speaking, controllability is concerned with the possibility of finding controls (this explains why the input variables will be considered as unknowns to be computed) so as to achieve objectives, which are defined in terms of the values one wishes the system variables to be given.

The reachable set of a system (the meaning of reachability is here of course different from the meaning it has in graph theory) is the set of states such that there exists a control which drives the system state, from some given initial value, to one of them. Global controllability is a strong property, which states that the reachable set is the whole state space. Local controllability is a weaker property, which states that any point in the open ball around a reachable point is also reachable (which means that the states which are reachable from a given state are not restricted to a r-dimensional manifold, where r is less than n, the dimension of the state space). For linear systems, local and global properties obviously coincide.

Let us first consider static systems $(\mathcal{C}, \mathcal{Z})$ like

$$0 = \boldsymbol{h}(\boldsymbol{x}_a, \boldsymbol{u}), \tag{5.38}$$

where $\mathcal{C} = \boldsymbol{h}$, $\mathcal{Z} = \boldsymbol{x}_a \cup \boldsymbol{u}$. For such systems, global controllability means that Eq. (5.38) can be solved for the unknown variables $\boldsymbol{u}$ (to be determined) for any value of the known (wished) variables $\boldsymbol{x}_a$, thus justifying the decomposition of $\mathcal{Z}$ into $\mathcal{Z} = \mathcal{K} \cup \mathcal{X}$, where $\mathcal{K} = \boldsymbol{x}_a$, $\mathcal{X} = \boldsymbol{u}$.

Theorem 5.4 (Controllability of static systems)
A necessary and sufficient condition for system (5.38) to be structurally controllable is:

(i) $\mathcal{K}$ is reachable from the input,
(ii) the canonical decomposition of $(\mathcal{C}_\mathcal{X}, \mathcal{X}, \mathcal{E}_\mathcal{X})$ contains no over-constrained subsystem.

Indeed, if $\mathcal{K}$ were not reachable from the input, there would be a decomposition of $\boldsymbol{x}_a$ into $\boldsymbol{x}'_a$ (the reachable part), and $\boldsymbol{x}''_a$ (the non reachable part), such that