Chapter 10
Distributed Fault Diagnosis and Fault-Tolerant Control

Abstract Distributed systems are formed by the interconnection of several subsystems or autonomous agents. Each entity is equipped with a local computing device that runs the whole or a part of the diagnosis and fault-tolerant control algorithms. This chapter explains the specific features of such systems and provides tools for the design and the coordination of distributed algorithms that achieve the overall diagnosis and control specifications, under given communication structures and local computing power limitations.

10.1 Introduction

The need for distributed control directly follows from the growing dimensions of complex, large-scale, multi-agent systems. Star architectures that connect all field devices (sensors and actuators) to one single computer running all the control laws are unpractical for large-scale applications. Using several computers and hubs to implement the control laws and connect the field devices is possible, thanks to local area networks that transfer the needed measurements to the computing devices and the generated control signals to the system actuators. The development of multi-agent systems (teams of robots, fleets of unmanned vehicles) also heavily rests on data transmissions between the individual entities and on local decision making.

Assuming a distributed control architecture, the implementation of a global diagnoser may be an unpractical option because of the amount of needed communication that sometimes makes it technically impossible. The diagnosis algorithms must then also be distributed, by assigning a part of the global fault detection and isolation task to each subsystem.

Fault-tolerant distributed systems have been considered for long in the software community to cope with hardware, software and communication faults. More recently, specific problems have been considered in the control community for fault-tolerant estimation, diagnosis and control of large-scale systems. As far as control is concerned, distributed systems introduce an *information pattern*, meaning that different data sets are available to different controllers, as opposed to the conventional design where all controllers share the same information. The information pattern

M. Blanke et al., *Diagnosis and Fault-Tolerant Control*,
DOI 10.1007/978-3-662-47943-8_10

plays also a very important role in distributed diagnosis, since the amount of known data available to each subsystem is a key parameter for its detection and isolation capabilities.

In this chapter, distributed systems are first presented. Distributed diagnosis is then addressed in reference with the structural fault detection and isolation capability of the overall system. According to the locally available model and data, the local diagnosers provide more or less powerful conclusions that must be coordinated (or aggregated) into a system-level overall diagnosis. Distribution algorithms are then considered, based on information patterns that take into account the specificities of the communication architecture. The constraints associated with possible local computing power limitations are also considered. The second part of the chapter addresses fault-tolerant distributed control. Since the solvability of the control design problem depends on the information pattern that is implemented, it follows that a fault that is not recoverable under a given information pattern might be recoverable under another one. The reconfiguration of the information patterns appears therefore as a powerful tool to achieve fault tolerance.

10.2 Distributed Systems

10.2.1 System Decomposition

Consider a system Σ equipped with a set I of m actuators, and a set J of p sensors. Its behaviour is described by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t) \tag{10.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}\,(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t), \tag{10.2}$$

where $\boldsymbol{x} \in |\mathcal{R}^n$ is the state, $\boldsymbol{u} \in |\mathcal{R}^m$ is the control vector, $\boldsymbol{y} \in |\mathcal{R}^p$ is the measurement vector and $\boldsymbol{d} \in |\mathcal{R}^q$ is some disturbance vector.

Let $\{\boldsymbol{u}_k,\ k = 1, \ldots, s\}$, $\left\{\boldsymbol{y}_k,\ k = 1, \ldots, s\right\}$ and $\{\boldsymbol{x}_k,\ k = 1, \ldots, s\}$ be partitions of $\boldsymbol{u}$, $\boldsymbol{y}$ and $\boldsymbol{x}$ into $s \geq 1$ subvectors, and let

$$\dot{\boldsymbol{x}}_k(t) = \boldsymbol{f}_k(\boldsymbol{x}_k(t), \overline{\boldsymbol{x}}_k(t), \boldsymbol{u}_k(t), \overline{\boldsymbol{u}}_k(t), \boldsymbol{d}(t), t) \tag{10.3}$$

$$\boldsymbol{y}_k = \boldsymbol{g}_k\,(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t) \tag{10.4}$$

$(k = 1, \ldots, s)$ be the resulting decomposition of Eqs. (10.1) and (10.2), where $\overline{\boldsymbol{x}}_k$ gathers all the components of $\boldsymbol{x}$ except $\boldsymbol{x}_k$.

Each equation in (10.3), (10.4) can be interpreted as describing the behaviour of a subsystem Σ_k with $\boldsymbol{u}_k \in |\mathcal{R}^{m_k}$ the local control vector associated with a subset I_k of the actuators, $\boldsymbol{y}_k \in |\mathcal{R}^{p_k}$ the local measurement vector associated with a subset J_k of the sensors and $\boldsymbol{x}_k \in |\mathcal{R}^{n_k}$ the local state. Note that $\mathcal{I} = \{I_k,\ k = 1, \ldots, s\}$ and $\mathcal{J} = \{J_k,\ k = 1, \ldots, s\}$ are partitions of I and J.

The functions $\boldsymbol{f}_k(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k, \boldsymbol{u}_k, \overline{\boldsymbol{u}}_k, \boldsymbol{d}, t)$ can take different forms. A specific case occurs when $\boldsymbol{f}_k(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k, \boldsymbol{u}_k, \overline{\boldsymbol{u}}_k, \boldsymbol{d}, t)$ is decomposable, namely it is the sum of two functions

$$\boldsymbol{f}_k(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k, \boldsymbol{u}_k, \overline{\boldsymbol{u}}_k, \boldsymbol{d}, t) = \boldsymbol{f}_k^{\text{self}}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{d}, t) + \boldsymbol{f}_k^{\text{coupled}}(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{u}}_k, \boldsymbol{d}, t),$$

where

- $\boldsymbol{f}_k^{\text{self}}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{d}, t)$ describes the self-dynamics of subsystem Σ_k and
- $\boldsymbol{f}_k^{\text{coupled}}(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{u}}_k, \boldsymbol{d}, t)$ describes the coupled dynamics with respect to the other subsystems (meaning the influence of the other subsystems on subsystem Σ_k).

Note that other decompositions of Σ could be defined, from the system global model (10.1), (10.2) by changing the value of s and the partitions of $\boldsymbol{u}$, $\boldsymbol{y}$ and $\boldsymbol{x}$. In practice, however, there is usually a natural decomposition into subprocesses associated with the global process to be controlled (then, each subsystem describes a given subprocess) or with the control system. Indeed, in large-scale processes, the control system is composed of several computing devices, each of them running some part of the real-time control algorithms (diagnosis, supervision, management, etc.), and the sensors and actuators are connected to the distributed control system through hubs and communication networks. In order to address distributed systems, the simple network architecture in which each subsystem Σ_k performs a part of the overall control and a part of the overall diagnosis is considered, as illustrated in Fig. 10.1.

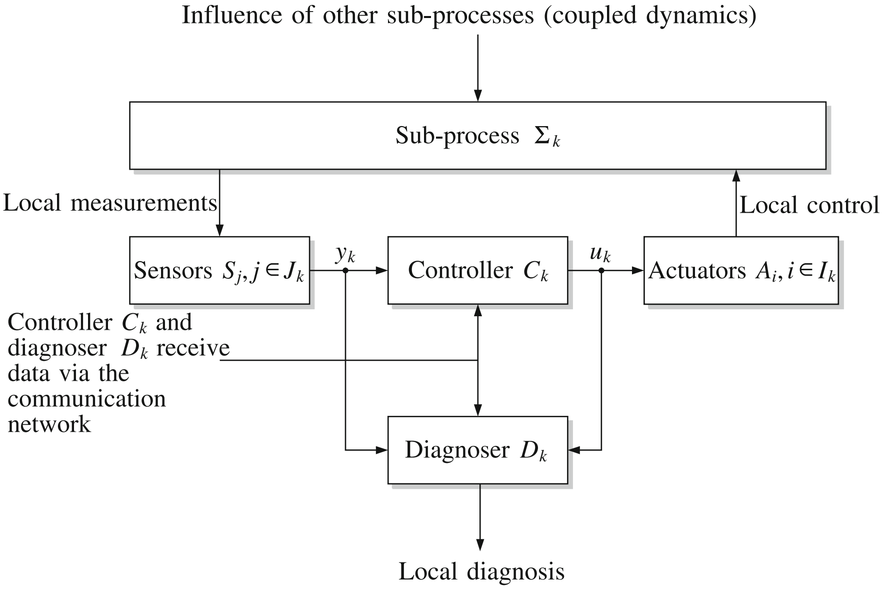
Influence of other sub-processes (coupled dynamics)
Sub-process Σ_k
Local measurements
Local control
Sensors $S_j, j \in J_k$
y_k
Controller C_k
u_k
Actuators $A_i, i \in I_k$
Controller C_k and diagnoser D_k receive data via the communication network
Diagnoser D_k
Local diagnosis

Fig. 10.1 Local controller and diagnoser

Example 10.1 System decomposition
Consider the sixth-order linear time invariant system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$

with

$$\boldsymbol{x}^{\mathrm{T}} = (x_1, x_2, x_3, x_4, x_5, x_6),$$

controlled by a set of 5 actuators ($I = \{1, 2, 3, 4, 5\}$) whose control signals are components of the vector

$$\boldsymbol{u}^{\mathrm{T}} = (u_1, u_2, u_3, u_4, u_5).$$

There are $2^6 - 2$ different ways to decompose this system into two subsystems. Indeed, each decomposition is obtained by considering a non-empty subset of the six states as the local state of the first subsystem and the remaining subset (provided it is non-empty) as the local state of the second subsystem.

More generally, the number of possible decompositions into s subsystems is the number of partitions of the state variables into s non-empty classes. If covers are considered instead of partitions, the result is a decomposition into overlapping subsystems, a case we shall not consider here. For example, with the matrices

$$\boldsymbol{A} = \begin{pmatrix} -1 & 2 & 1 & -1 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0.5 & -0.5 & -2 & 0 & 1 & 1 \\ 1 & -1 & -1 & -2 & 0.4 & 0 \\ 0 & 0 & 0 & 1 & -3 & 1 \\ 2 & 0 & -1 & -2 & 0 & -4 \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} \tag{10.5}$$

and the decomposition (x_1, x_2), (x_3, x_4, x_5, x_6), the two subsystems are, respectively,

$$\Sigma_1 : \quad \begin{pmatrix} \dot{x}_1 \\ \dot{x_2} \end{pmatrix} = \begin{pmatrix} -1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 1 & -1 & 0 & 1 \\ 2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

and

$$\Sigma_2 : \left\{ \begin{array}{l} \begin{pmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} -2 & 0 & 1 & 1 \\ -1 & -2 & 0.4 & 0 \\ 0 & 1 & -3 & 1 \\ -1 & -2 & 0 & -4 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} u_3 \\ u_4 \\ u_5 \end{pmatrix} \\ \qquad\qquad + \begin{pmatrix} 0.5 & -0.5 \\ 1 & -1 \\ 0 & 0 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \end{array} \right.$$

Note that for linear systems, the state equations are decomposable whatever the partition of the state that is considered, provided that the control signals do not simultaneously act on different subsystems. In this example, actuator 4 directly influences the state variables x_4 and x_5 and, therefore, decompositions in which these variables would belong to different subsystems would not enjoy the property that their state equations are decomposable. Associating a controller with each subsystem results in the determination of the control signals $u_1,\ u_2$ by Σ_1 and the determination of $u_3,\ u_4,\ u_5$ by Σ_2. Note that, unlike the control decomposition, the diagnosis decomposition is not implied by the system decomposition. Indeed, in addition to the computation of $u_1,\ u_2$, the computing device of subsystem Σ_1 could be assigned any part of some overall diagnosis algorithm (provided it is fed with the appropriate data and has enough computing power), and the same applies of course to subsystem Σ_2.

In the sequel, this example will be continued under the assumption that there is some physical reason to distinguish four subsystems based on the partition of the state (x_1, x_2), (x_3), (x_4, x_5), (x_6) and the partition of the actuators $I_1 = \{1,\ 2\}, I_2 = \{3\}, I_3 = \{4\}, I_4 = \{5\}$. The considered system decomposition is

$$\Sigma_1 : \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \cdots$$

$$+ \begin{pmatrix} 1 & -1 & 0 & 1 \\ 2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

$$\Sigma_2 : \quad \dot{x}_3 = -2x_3 + u_3 + \begin{pmatrix} 0.5 & -0.5 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

$$\Sigma_3 : \quad \begin{pmatrix} \dot{x}_4 \\ \dot{x}_5 \end{pmatrix} = \begin{pmatrix} -2 & 0.4 \\ 1 & -3 \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix} u_4 + \cdots$$

$$+ \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_6 \end{pmatrix}$$

$$\Sigma_4 : \quad \dot{x}_6 = -4x_6 + 2u_5 + \begin{pmatrix} 2 & 0 & -1 & -2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} . \square$$

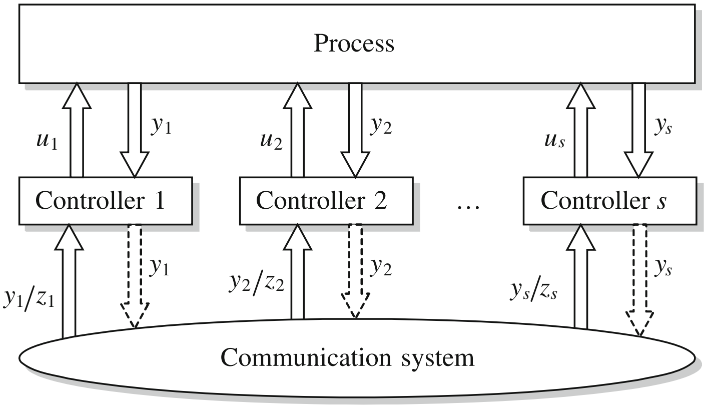

Fig. 10.2 Distributed system architecture

10.2.2 Distributed Control

Local controllers. Assuming that each subsystem Σ_k to be equipped with its own controller means that the overall control (i.e. the determination of vector $\boldsymbol{u}$) is distributed among the s controllers and each of them is in charge of computing the subvector $\boldsymbol{u}_k$. The design of efficient control algorithms might need some controllers to use more measurements than the locally available ones. This is possible, thanks to the existence of a communication network such that local controllers can use the measurements $z_k \in |\mathcal{R}^{\pi_k}$ provided by a subset of sensors Z_k. Since the local measurements are always available to subsystem Σ_k, the relation $J_k \subseteq Z_k \subseteq J$ holds and $Z_k \setminus J_k$ is the set of remote sensors whose measurements are made available to Σ_k over the communication network. Since $\mathcal{J} = \{J_k,\ k = 1, \dots, s\}$ is a partition of J, it follows that $\{Z_k,\ k = 1, \dots, s\}$ is a cover of J, i.e. one has $Z_k \neq \emptyset$, $(k = 1, \dots, s)$ and $\cup_{k=1,\dots,s} Z_k = J$. Figure 10.2 displays the corresponding architecture (dotted arrows mean that the variables may, or may not, be communicated). In the sequel, for the sake of conciseness, we use the same notation for the sensors and the signals they deliver (should they be ordered as vectors or not), for example $\boldsymbol{y}_k \subseteq z_k \subseteq \boldsymbol{y}$, $z_k \setminus \boldsymbol{y}_k$, etc.

Information pattern. Given a system decomposition, the s-tuple

$$\mathcal{Z} = \{z_k,\ k = 1, \dots, s\}$$

is an *information pattern*. The *full information pattern* is

$$\mathcal{Z}_{\max} = \{z_k = \boldsymbol{y},\ k = 1, \dots, s\},$$

meaning that all the measurement signals $\boldsymbol{y}$ are available to each local controller. Note that this is nothing but the centralised control architecture, when $s = 1$, and a distributed implementation of the centralised control when $s > 1$. On the contrary, under the *local information pattern*

$$\mathcal{Z}_{\min} = \left\{ z_k = \boldsymbol{y}_k,\ k = 1, \ldots, s \right\},$$

only locally produced measurements are used by each local controller, which characterises the decentralised control scheme.

Example 10.2 Local controllers
Assume that the system of Example 10.1 is equipped with 4 sensors $J = \{1,\ 2,\ 3,\ 4\}$ providing the measurement signals

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 0\ 1\ 0\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

and each controller is interfaced with one of them as follows:

$$J_1 = \{1\},\ J_2 = \{2\},\ J_3 = \{3\},\ J_4 = \{4\}.$$

Assume that output feedback is investigated, for the sake of simplicity. Consider for example the information pattern $\mathcal{Z} = \{(y_1, y_2), y_2, (y_1, y_3), (y_2, y_4)\}$. It needs the signal y_1 to be communicated from Σ_1 to Σ_3 and the signal y_2 to be communicated from Σ_2 to Σ_1 and Σ_4, and it allows to use the controllers

$$\begin{aligned} u_1(t) &= k_{11} y_1(t) + k_{12} y_2(t) \\ u_2(t) &= k_{22} y_2(t) \\ u_3(t) &= k_{31} y_1(t) + k_{33} y_3(t) \\ u_4(t) &= k_{42} y_2(t) + k_{44} y_4(t), \end{aligned}$$

where the k_{ij} are the real output feedback gains. By comparison, the full information pattern $\mathcal{Z}_{\max} = \{\boldsymbol{y},\ \boldsymbol{y},\ \boldsymbol{y},\ \boldsymbol{y}\}$ allows the design $\boldsymbol{u}(t) = \boldsymbol{K y}(t)$ with $K \in |\mathcal{R}^{5\times 4}$, but needs all the measurements to be communicated, while in the local information pattern

$$\mathcal{Z}_{\min} = \{y_1,\ y_2,\ y_3,\ y_4\}$$

associated with decentralised control, no variable at all is communicated, but the output feedback must satisfy the constraints $u_i(t) = k_{ii} y_i(t)$ where the k_{ii} are real numbers. □

10.2.3 Distributed Diagnosis

Whatever the way they have been designed (analytical redundancy relations, observers, identification-based designs), a centralised diagnoser evaluates all the residuals using the data available to it through its connection with the system sensors and controllers. In a distributed architecture, each subsystem Σ_k, $(k = 1, \ldots, s)$ runs its own local diagnoser, defined by a pair $(\boldsymbol{r}_k, \delta_k)$ where $\boldsymbol{r}_k$ are the residuals it has been assigned and δ_k is a decision procedure on the residuals $\boldsymbol{r}_k$. Let $z_k^a \subseteq \boldsymbol{u}$ and $z_k^s \subseteq \boldsymbol{y}$ be the control and measurement signals whose knowledge is needed to run the residuals $\boldsymbol{r}_k$ that have been assigned to subsystem Σ_k (z_k^a and z_k^s are determined by the computation form of the residuals in $\boldsymbol{r}_k$). Then, the information pattern that allows the local diagnosers to perform their task is

$$\mathcal{Z} = \left\{ \left(z_k^a, z_k^s \right), k = 1, \ldots, s \right\}.$$

The full information pattern is $\mathcal{Z}_{\max} = \{(\boldsymbol{u}, \boldsymbol{y}), k = 1, \ldots, s\}$, while the local information pattern is $\mathcal{Z}_{\min} = \left\{ \left(\boldsymbol{u}_k, \boldsymbol{y}_k \right), k = 1, \ldots, s \right\}$.

10.2.4 Communication Cost

Given an information pattern $\mathcal{Z}$, the sets $z_k \setminus \boldsymbol{y}_k$, $(k = 1, \ldots, s)$ contain those measurement signals that are needed by, but are not locally available to, the controller of subsystem Σ_k. Similarly, the sets $z_k^a \setminus \boldsymbol{u}_k$ and $z_k^s \setminus \boldsymbol{y}_k$ contain the control signals (resp. the measurement signals) that are needed by, but are not locally available to, the diagnoser of subsystem Σ_k. Those data are received through the communication network, that involves some communication cost. Whatever the network and the communication protocol, the communication cost would clearly depend on the variables coding, transmission rate, checking procedures, management strategy, etc., and it would be growing with the number of communicated variables. It is assumed that the communication cost is expressed by a function $com\,(K)$ where K is the set of communicated variables, such that $com\,(\emptyset) = 0$ and $K_1 \subseteq K_2 \Rightarrow com\,(K_1) \leq com\,(K_2)$.

10.2.5 Communication Schemes

Among many available communication schemes, this chapter builds on the publisher/subscriber and the bilateral agreements based ones. The publisher/subscriber scheme is associated with diffusion-based networks and is well suited to factory communication protocols, like communication between intelligent sensors, actuators and subsystems, while the bilateral agreements scheme is well suited to describe the communication between autonomous agents.

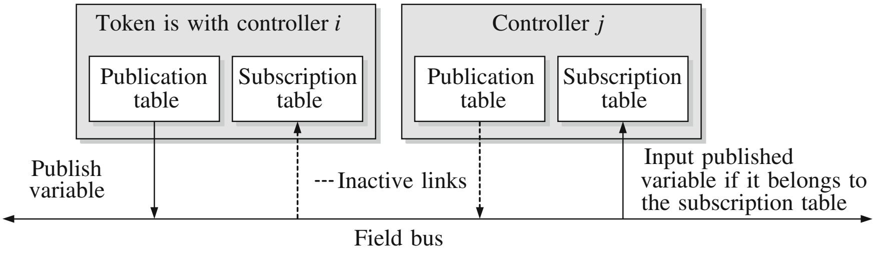

Fig. 10.3 The publisher/subscriber scheme

Diffusion-based networks. In diffusion-based networks,[1] a variable that is published in the communication system is available to all the subsystems that subscribe to it. The process globally works as follows:

- When a controller gets the token, it takes control of the communication bus and publishes the identifier and the value of the variables it is in charge of publishing (they are in its publication table);
- The other controllers recognise the identifier of a variable they have subscribed to (the list is in their subscription table). If recognised, they input its value; and
- The token passes to the next controller.

Figure 10.3 illustrates the publisher/subscriber scheme.

Bilateral agreements. In this scheme, subsystems establish bilateral agreements by which they share their data. Let a be the binary relation such that $a\left(\Sigma_i,\ \Sigma_j\right) = 1$ if Σ_i and Σ_j share their data, $a\left(\Sigma_i,\ \Sigma_j\right) = 0$ otherwise. Note that a being reflexive, symmetric and transitive, its graph $\mathcal{A}$ (which represents the set of agreements) involves a partition of all subsystems $\{\Sigma_k,\ k = 1, \ldots, s\}$ into equivalence classes $\mathcal{E}\,(\mathcal{A}) = \{E_l,\ l = 1, \ldots, \sigma\}$, with $\sigma = s$ when $\mathcal{A}$ is empty and $\sigma < s$ otherwise. It follows that the same data $z\,(E_l)$ are available to all the subsystems that belong to the same class E_l, as illustrated by Fig. 10.4 for a system with five distributed controllers and two equivalence classes.

Other schemes. Teams of autonomous agents most often use wireless communications, which restricts the communication possibilities of each agent to a subset of the other agents in its neighbourhood. In such applications, the network is described by a graph whose nodes $\mathcal{N}_i$ are the individual agents, and an arc between agents $\mathcal{N}_i$ and $\mathcal{N}_j$ indicates that the first can send data to the second. Such communication schemes are not considered in this chapter.

[1] Examples of diffusion-based networks are the Factory Instrumentation Protocol defined by the European Standards EN50170 and the IEC 61158/IEC 61784 Communication Profile Family 5.

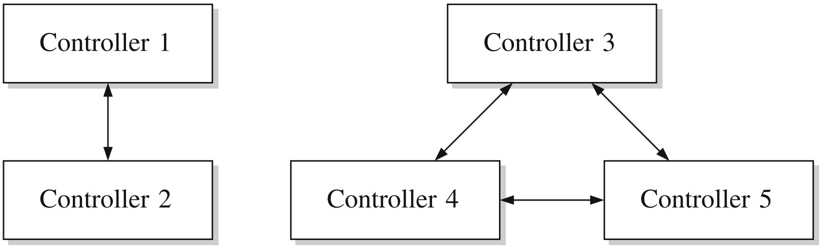

Fig. 10.4 Two agreement classes between 5 controllers

10.3 Distributed Diagnosis Design

In order to design a local diagnoser for each subsystem, two problems are to be solved:

1. characterise the system-level diagnosis that follows from the subsystem-level diagnosis and
2. design the local diagnosers so as to obtain specified results at the global system level.

These problems are addressed in this section and in the next one.

A direct means to evaluate the system-level diagnosis achieved by a set of distributed diagnosers is to compare it with the results that would be obtained with the overall (centralised) diagnoser. In order to develop this comparison, we first highlight the parameters that shape the design of a global diagnoser, namely its structural and its quantitative properties, which were, respectively, presented in Chaps. 5, 6 and 7.

10.3.1 Structural Diagnoser

Remember that from a structural point of view, the dynamical behaviour of a system Σ is described by a set of variables $\mathcal{V}$ and a set of constraints $\mathcal{C}$ that are satisfied when it is healthy. For continuous systems, the constraints $\mathcal{C}$ are algebraic and differential equations, the classical formulation of which is recalled here:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\,(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t) \tag{10.6}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}\,(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t), \tag{10.7}$$

where $\boldsymbol{x} \in |\mathcal{R}^n$ is the state, $\boldsymbol{u} \in |\mathcal{R}^m$ and $\boldsymbol{d} \in |\mathcal{R}^q$ are, respectively, the known and unknown inputs, and $\boldsymbol{y} \in |\mathcal{R}^p$ are the known measured outputs. In the sequel, we still use the same notation for sets and vectors of variables, as well as for sets and vectors of constraints, because no confusion is possible. Note that algebraic constraints on the state can easily be introduced via Eq. (10.7) as a subset of sensors whose output is constant and equal to zero. The variables $\mathcal{V}$ are partitioned into known $\mathcal{K} = \boldsymbol{u} \cup \boldsymbol{y}$ and unknown $\mathcal{X} = \boldsymbol{x} \cup \boldsymbol{d}$ variables.

In order to characterise the global diagnoser's properties, we first recall how analytical redundancy relations (ARR) are exhibited from the system canonical decomposition and we present a basic result from the logical theory of diagnosis.

Canonical decomposition. The structural analysis of Σ is the analysis of the bipartite graph $\mathcal{G} = (\mathcal{C}, \mathcal{Z}, \mathcal{E})$ introduced in Sect. 5.2, where $\mathcal{Z}$ is the set of variables, $\mathcal{C}$ is the set of constraints and $\mathcal{E}$ is the set of edges in which each pair $(c_i, z_j) \in \mathcal{E}$ means that the variable z_j appears in the constraint c_i. The DM decomposition of the graph $\mathcal{G}$ is explained in Sect. 5.4.1 that provides three canonical subsystems of Σ, namely an over-constrained, a just-constrained and an under-constrained subsystem. The over-constrained subsystem exhibits more than one solution to the complete matching problem of its unknown variables, while in the just-constrained subsystem, the complete matching is unique, and there is no complete matching in the under-constrained subsystem. Remember that the set $\mathcal{Z}$ of variables is decomposed into the set $\mathcal{K}$ of known variables and the set $\mathcal{X}$ of unknown variables and that a complete matching of the unknown variables allows to express the unknown variables as functions of the known variables, which means that it is possible to eliminate them in any constraint where they appear, simply by replacing them by their expression.

Analytical redundancy relations. The over-constrained subsystem is the monitorable part of Σ. Indeed, the existence of more than one complete matching of its unknown variables implies that a set of compatibility condition must be satisfied by the variables in $\mathcal{K}$ for Eqs. (10.6) and (10.7) to be consistent. These conditions are the analytical redundancy relations (ARR).

The essence of ARR-based diagnosis is to check whether the ARR are satisfied or not by the known data. This is done via a set of *residuals* whose computation involves only known variables, and whose value should be zero in normal operation. Let $\boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$ be the set of residuals associated with the set of constraints $\mathcal{C}$ and the known variables $\mathcal{K}$.

The computation of each residual $\rho \in \boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$ involves a subset $\mathcal{K}\,(\rho) \subseteq \mathcal{K}$ of known variables and a subset $\mathcal{C}\,(\rho) \subseteq \mathcal{C}$ of constraints. $\mathcal{K}\,(\rho)$ is known from its *computation form*, and $\mathcal{C}\,(\rho) \subseteq \mathcal{C}$ defines its *structure*.

Remark 10.1 ARR-based residuals generally call for derivatives of the known variables, which is often argued against them in real-time applications. However, the derivation order can be limited (at the cost of reducing the number of found ARR) and moreover, it is in general possible to design observers whose outputs are equivalent and do not suffer the noise sensitivity issues. □

10.3.2 Logical Theory of Diagnosis

The logical theory of diagnosis is a tool that will be used to explain the coordination of several local diagnoses. It rests on the residuals signatures presented in Chap. 5 that are further analysed here.

Structural detectability. Since a fault changes one (or several) system constraint(s), it follows that there is a contradiction between the two statements:

1. a residual $\rho(t)$ is *falsified* by the data ($\rho(t) \neq 0$) and
2. all the constraints in its structure $\mathcal{C}(\rho)$ hold true.

The structure of a falsified residual is named as *conflict*, meaning it contains at least one faulty (untrue) constraint. It follows that for a faulty constraint to be detectable, it must belong at least to one residual's structure. The set of faults structurally detectable by the residuals $\boldsymbol{r}$ is therefore $\mathcal{D} = \cup_{\rho \in r} \mathcal{C}(\rho)$, while the non-detectable faults are $\overline{\mathcal{D}} = \mathcal{C} \setminus \mathcal{D}$. Remark that this explains the term *monitorable* that applies to the over-constrained subsystem, because it is the only one that produces residuals.

Remark 10.2 It is important to remark that the system canonical decomposition is unique. It follows that the set of detectable faults is also unique. In particular, it cannot be extended by ARR combinations. □

Structural isolability. A constraint $c \in \mathcal{D}$ partitions the residuals $\boldsymbol{r}$ into $\boldsymbol{r}_1(c)$ whose structure contains c and $\boldsymbol{r}_0(c)$ whose structure does not contain c. Let us first consider single faults: when c is faulty, the residuals $\boldsymbol{r}_0(c)$ are satisfied while the residuals $\boldsymbol{r}_1(c)$ are falsified. The *signature* of fault c is the vector $\boldsymbol{s}(c)$ whose jth component gives the status of residual r_j (0 when satisfied, 1 when falsified). The diagnoser is characterised by its *distinguishability* partition $\left\{\mathcal{D}^i, i = 0, 1, 2, \ldots\right\}$ where $\mathcal{D}^0 = \mathrm{OK} \cup \overline{\mathcal{D}}$ are the situations that have the same signature as the healthy system, namely $\boldsymbol{s}\left(\mathcal{D}^0\right)$ such that $\boldsymbol{r}_0(\mathcal{D}^0) = \boldsymbol{r}$ and $\boldsymbol{r}_1\left(\mathcal{D}^0\right) = \emptyset$, and $\mathcal{D}^i$, $(i \neq 0)$ are the faulty situations that have the same signature $\boldsymbol{s}\left(\mathcal{D}^i\right)$.

Assuming that fault cancellations do not occur, a multiple fault

$$C = \{c_i, i = 1, 2, \ldots\}$$

has the signature $\boldsymbol{s}(C)$ such that

$$\boldsymbol{r}_1(C) = \cup_{c \in C} \boldsymbol{r}_1(c)$$

and

$$\boldsymbol{r}_0(C) = \boldsymbol{r} \setminus \boldsymbol{r}_1(C).$$

Note that a special case of multiple faults is addressed in the partition

$$\left\{\mathcal{D}^i, i = 0, 1, 2, \ldots\right\}$$

because the signature $\boldsymbol{s}\left(\mathcal{D}^i\right)$ characterises any subset of faults that belong to the same class $\mathcal{D}^i$, $(i \neq 0)$. The set of fault signatures can be studied by considering every single and multiple faults. However, it is simpler to rely on the following result from the logical theory of diagnosis.

Minimal hitting sets and diagnosis. Let $\boldsymbol{r} = \boldsymbol{r}_{\mathrm{s}} \cup \boldsymbol{r}_{\mathrm{f}}$ be a partition of the residuals $\boldsymbol{r}$ into satisfied residuals $\boldsymbol{r}_{\mathrm{s}}$ and falsified residuals $\boldsymbol{r}_{\mathrm{f}}$. A minimal subset of constraints Δ^i whose faults result in this very partition is a *possible diagnosis*. Since more than one such subset may exist, the *overall diagnosis* is the set of possibilities $\Delta = \left\{\Delta^i, i = 1, 2, \ldots\right\}$. Note that this definition automatically includes simple and multiple faults.

Theorem 10.1 (Minimal hitting set) *Let* $\{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_{\mathrm{f}}\}$ *be the set of conflicts associated with the partition of the residuals into* $\boldsymbol{r} = \boldsymbol{r}_{\mathrm{s}} \cup \boldsymbol{r}_{\mathrm{f}}$. *A possible diagnosis is a minimal hitting set of* $\{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_{\mathrm{f}}\}$.

A subset of constraints H is a hitting set of $\{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_{\mathrm{f}}\}$ if the two relations

- $H \subseteq \cup_{\rho \in \boldsymbol{r}_{\mathrm{f}}} \mathcal{C}(\rho)$ and
- $H \cap \mathcal{C}(\rho) \neq \emptyset, \forall \rho \in \boldsymbol{r}_{\mathrm{f}}$

hold. H is minimal if no proper subset of H satisfies these two conditions. In words, H is a minimal subset of constraints such that

- each of them belongs to at least one conflict and
- a corresponding multiple fault falsifies every residual in $\boldsymbol{r}_{\mathrm{f}}$.

Example 10.3 Ship with dual measurements

The simplified non-linear model of a ship steering system with dual measurements was considered in Chap. 5. The unknown variables are the heading angle ψ, the turn rate ω and the rudder angle δ. There are four known variables $\{y_1, y_2, y_3, y_4\}$. From the state and measurement equations

$$\begin{array}{l} c_1: \\ c_2: \end{array} \begin{pmatrix} \dot{\omega} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} \eta_1 \omega + \eta_3 \omega^3 + \delta \\ \omega \end{pmatrix}$$

$$\begin{array}{l} m_1: \\ m_2: \\ m_3: \\ m_4: \end{array} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} \psi \\ \psi \\ \dot{\psi} \\ \delta \end{pmatrix},$$

one finds three residuals whose computation forms and structures are, respectively, given by Eq. (10.8) and Table 10.1:

$$\begin{aligned} \rho_1 &= y_2 - y_1 \\ \rho_2 &= \dot{y}_1 - y_3 \\ \rho_3 &= \dot{y}_3 - \eta_1 y_3 - \eta_3 y_3^3 - y_4. \end{aligned} \tag{10.8}$$

A fault in any constraint is detectable since there is at least a “1” in each column of the signature table. Some faults are not isolable, since they have identical signatures. The resulting equivalence classes are $\mathcal{D}^1 = \{m_1\}$, $\mathcal{D}^2 = \{m_2\}$, $\mathcal{D}^3 = \{m_3\}$ and $\mathcal{D}^4 = \{m_4, c_1, c_2\}$, which gives the distinguishability Table 10.2 (*OK* has been re-labelled as $\mathcal{D}^0$):

Table 10.1 Structures of the ship example residuals

	OK	m_1	m_2	m_3	m_4	c_1	c_2
ρ_1	0	1	1	0	0	0	0
ρ_2	0	1	0	1	0	0	0
ρ_3	0	0	0	1	1	1	1

Table 10.2 Distinguishability table of the ship example

	$\mathcal{D}^0$	$\mathcal{D}^1$	$\mathcal{D}^2$	$\mathcal{D}^3$	$\mathcal{D}^4$
ρ_1	0	1	1	0	0
ρ_2	0	1	0	1	0
ρ_3	0	0	0	1	1

Assume residual ρ_1 is satisfied by the real-time measurements, residuals ρ_2, ρ_3 are falsified and the signature 011 directly leads to the diagnosis $\mathcal{D}^3$. Now, suppose that although there is no signature 111 in the table, all three residuals are falsified by the real-time measurements. Using the notation $\mathcal{D}^i \times \mathcal{D}^j$ for a double fault diagnosis in which the first fault is a constraint in $\mathcal{D}^i$ and the second fault a constraint in $\mathcal{D}^j$, it is seen that this is indeed possible as the result of the multiple faults $\mathcal{D}^1 \times \mathcal{D}^3 \cup \mathcal{D}^1 \times \mathcal{D}^4 \cup \mathcal{D}^2 \times \mathcal{D}^3$, as it can be visually checked on Fig. 10.5 where $\mathcal{D}^1 \times \mathcal{D}^3$, $\mathcal{D}^1 \times \mathcal{D}^4$ and $\mathcal{D}^2 \times \mathcal{D}^3$ are minimal hitting sets of $\{\mathcal{C}(\rho_1), \mathcal{C}(\rho_2), \mathcal{C}(\rho_3)\}$.

Considering all possible signatures (note that the signature 010 can never be obtained) gives the diagnosis Table 10.3.

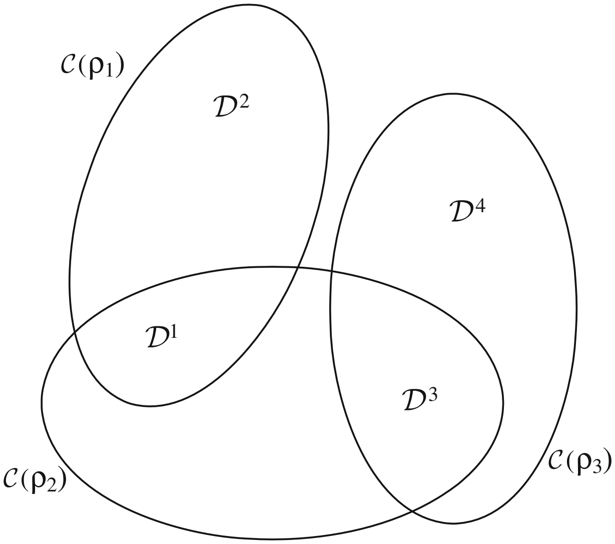

Fig. 10.5 The three conflicts associated with the signature 111

Table 10.3 Diagnosis table of the ship example

$\rho_1\rho_2\rho_3$	Diagnosis
000	$\mathcal{D}^0$
001	$\mathcal{D}^4$
010	cannot happen
011	$\mathcal{D}^3$
100	$\mathcal{D}^2$
101	$\mathcal{D}^2 \times \mathcal{D}^4$
110	$\mathcal{D}^1$
111	$\mathcal{D}^1 \times \left(\mathcal{D}^3 \cup \mathcal{D}^4\right) \cup \mathcal{D}^2 \times \mathcal{D}^3$

Remark 10.3 The conclusion obtained via the logical theory of diagnosis may contain many possible diagnosis, as it can be seen from the previous example, where there are three sets of possible double faults associated with the signature 111. All conclusions are indeed consistent from a logical point of view. However, in practical applications, one may have to select only one of them. Under the assumption that the joint probabilities of faults occurring in different constraints are known, it seems of course appropriate to select the most probable one. □

10.3.3 Practical Diagnoser and Real-Time Operation

Structural versus actual properties. Structural properties are necessary but not sufficient for actual properties to be true. A residual whose structure does not contain a given fault can by no means allow its detection, but a structurally detectable fault might never be detected in practice because the sensitivity of the residuals is too small, or because the signal/noise ratio does not allow its detection. Similarly, two isolable faults might never be isolated from each other if only a common subset of residuals is sensitive enough to them.

A practical diagnoser is a pair $(\boldsymbol{r}, \delta)$ where $\boldsymbol{r}$ is a set of residuals and δ is a decision procedure that checks the residuals status (satisfied/falsified), using the available knowledge on modelling errors, unknown inputs and measurement noises, in order to reduce false alarms, missed detections, detection delays and mis-isolations, as analysed in Chap. 7.

Real-time operation. The real-time operation of a practical diagnoser follows 4 steps (steps 3 and 4 are optional depending on the application):

1. Compute the value of the residuals $\boldsymbol{r}$ from the values of the known variables $\mathcal{K}$;
2. Fault detection: evaluate the residuals using the decision procedure δ and conclude whether a fault has occurred ($\boldsymbol{r}_{\mathrm{f}} \neq \emptyset$) or if the system can possibly be healthy ($\boldsymbol{r}_{\mathrm{s}} = \boldsymbol{r}$);

3. Fault isolation: find the minimal hitting sets consistent with the observed signature, if a detection has been fired; and
4. Fault estimation: estimate the model of the faulty system.

Whatever the complexity of steps 2, 3 and 4, they apply to the residuals $\boldsymbol{r}$ issued from the structural analysis. Implementing a centralised or a distributed diagnoser is therefore based on implementing a centralised or a distributed computation scheme for the residuals $\boldsymbol{r}$. This is why only the fault detection and isolation properties of structural diagnosers are considered in the sequel.

Centralised or distributed implementation. In a centralised system, the diagnoser is run by a single computing device that is connected with all the sensors and controllers. A centralised diagnosis implementation is based on the assumptions that

- the data involved in the computation form of any residual are available to the central computing device;
- there is no data transmission delay, or if some delay is unavoidable, all the data involved in the computation form of a given residual are available under compatible time stamps.

In a distributed diagnosis scheme, each subsystem Σ_k, $(k = 1, \ldots, s)$ runs its own diagnosis algorithm, and the above assumptions may be no longer satisfied:

- the data available to each subsystem depend on the information pattern that is implemented;
- the communication network may introduce transmission errors, data losses and unacceptable delays; and
- even when not faulty, the communication network may introduce different transmission delays for different variables, due to the network scheduling procedures.

The design of a distributed diagnosis scheme rises two interrelated problems:

- **Problem 1**. Given a residual vector $\boldsymbol{r}$ and a set of subsystems Σ_k, $(k = 1, \ldots, s)$ how to design an information pattern and how to distribute the residual computations between the different subsystems? and
- **Problem 2**. Given a set of local diagnosers how to achieve an overall decision that is consistent with the locally achieved ones?

Because it is needed to understand the coordination procedure in order to design the residuals distribution, we start with the solution of Problem 2.

10.3.4 Local Diagnosers and Their Coordination

As the data available to local diagnosers depend on the information pattern, we will need to manipulate these entities in order to understand the *subsystem-level diagnosis capabilities*. The *system-level coordination* of all the subsystem-level diagnosis will be addressed after.

Information pattern. We first give a formal definition of the set of all information patterns and define an order on this set.

Definition 10.1 (*Information pattern*) An information pattern is a set $\mathcal{Z} = \{z_k, k = 1, \ldots, s\}$, where z_k is a pair $\left(z_k^a, z_k^s\right)$ such that $\boldsymbol{u}_k \subseteq z_k^a \subseteq \boldsymbol{u}$ and $\boldsymbol{y}_k \subseteq z_k^s \subseteq \boldsymbol{y}$.

In other words, an information pattern is a s-tuple whose ith element is the subset of input/output data available to the ith subsystem. The set of information patterns is easily provided with a partial order relation defined as

$$\mathcal{Z}_1 \preceq \mathcal{Z}_2 \Leftrightarrow \forall k = 1, \ldots, s : z_{1,k}^a \subseteq z_{2,k}^a \wedge z_{1,k}^s \subseteq z_{2,k}^s.$$

In this case, $\mathcal{Z}_2$ is said to be *wider* than $\mathcal{Z}_1$—or $\mathcal{Z}_1$ is *narrower* than $\mathcal{Z}_2$. The minimal information pattern

$$\mathcal{Z}_{\min} = \left\{\left(\boldsymbol{u}_k, \boldsymbol{y}_k\right), k = 1, \ldots, s\right\}$$

is narrower than any other information pattern, and the maximal information pattern

$$\mathcal{Z}_{\max} = \{(\boldsymbol{u}, \boldsymbol{y}), k = 1, \ldots, s\}$$

is wider than any other one.

It follows from the definition that any information pattern

$$\mathcal{Z} = \left\{\left(z_k^a, z_k^s\right), k = 1, \ldots, s\right\}$$

is such that $\cup_{k=1,\ldots,s} z_k^a = \boldsymbol{u}$ and $\cup_{k=1,\ldots,s} z_k^s = \boldsymbol{y}$, in other words, the z_k^a, respectively, and the z_k^s are a cover of $\boldsymbol{u}$, respectively, of $\boldsymbol{y}$.

Local diagnosers. The structural analysis of the global system model (10.6), (10.7)

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t\right) \\ \boldsymbol{y}(t) &= \boldsymbol{g}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t\right) \end{aligned}$$

results in the set of residuals $\boldsymbol{r}(\mathcal{C}, \mathcal{K})$ where $\mathcal{C} = \boldsymbol{f} \cup \boldsymbol{g}$ and $\mathcal{K} = \boldsymbol{u} \cup \boldsymbol{y}$. Assume that for some reason, we are interested in the structural analysis of the constraints $\mathcal{C}$ when there are less known variables than $\mathcal{K}$, namely $\mathcal{K}^- \subseteq \mathcal{K}$. The monotonicity property

$$\mathcal{K}^- \subseteq \mathcal{K} \Rightarrow \boldsymbol{r}\left(\mathcal{C}, \mathcal{K}^-\right) \subseteq \boldsymbol{r}(\mathcal{C}, \mathcal{K})$$

holds true, with the conclusion that if $\mathcal{Z} = \left\{\left(z_k^a, z_k^s\right), k = 1, \ldots, s\right\}$ is an information pattern by which the known variables available to subsystem Σ_k are $\mathcal{K}_k = \left(z_k^a, z_k^s\right)$, the subset of residuals that can be computed by subsystem Σ_k is $\boldsymbol{r}(\mathcal{C}, \mathcal{K}_k) \subseteq \boldsymbol{r}(\mathcal{C}, \mathcal{K})$.

Similarly, let $\mathcal{C}^- \subseteq \mathcal{C}$ be a subset of constraints, then whatever the known variables $\mathcal{K}$ one has $\boldsymbol{r}\left(\mathcal{C}^-, \mathcal{K}\right) \subseteq \boldsymbol{r}(\mathcal{C}, \mathcal{K})$, with the consequence that another monotonicity property holds true under the information pattern $\mathcal{Z}$, namely

$$\mathcal{C}_k \subseteq \mathcal{C} \Rightarrow \boldsymbol{r}\,(\mathcal{C}_k, \mathcal{K}_k) \subseteq \boldsymbol{r}\,(\mathcal{C}, \mathcal{K}_k) \subseteq \boldsymbol{r}\,(\mathcal{C}, \mathcal{K}).$$

Definition 10.2 (*Distributed diagnosis scheme*) Let $\mathcal{C} = \{\mathcal{C}_k, k = 1, \ldots, s\}$ be a collection of constraint subsets and consider an information pattern $\mathcal{Z} = (z_k, k = 1, \ldots, s)$. A distributed diagnosis scheme is a set of local diagnosers $(\boldsymbol{r}\,(\mathcal{C}_k, z_k), \delta_k)$ where δ_k is the decision procedure associated with the evaluation of the residuals $\boldsymbol{r}\,(\mathcal{C}_k, z_k)$.

The subset of constraints $\mathcal{C}_k$ associated with a local diagnoser need not be the constraints $\boldsymbol{f}_k \cup \boldsymbol{g}_k$ that describe the behavioural model of subsystem Σ_k in Eqs. (10.3), (10.4).

Before we describe different diagnosis schemes associated with different choices of $\mathcal{C}$, let us first investigate the relation between local and global diagnosis.

Local versus global detection. From a structural point of view, each local diagnoser in a distributed diagnosis scheme is characterised by the partition

$$\left\{\mathcal{D}_k^i, i = 0, 1, 2, \ldots\right\}$$

that defines the system situations and it is able to distinguish from its different residual signatures. Note that such a local partition is necessarily wider than the partition $\left\{\mathcal{D}^i, i = 0, 1, 2, \ldots\right\}$ associated with the global diagnoser, since $\mathcal{D}_k^i$ is the union of several subsets $\mathcal{D}^j$ in the global distinguishability table.

In order to be detected, a fault must be detectable by at least one local diagnoser. The set of detectable faults in a distributed diagnosis scheme is therefore $\cup_{k=1,\ldots,s}\mathcal{D}_k$ where $\mathcal{D}_k = \cup_{i\neq 0}\mathcal{D}_k^i$ is the set of faults detectable by the local diagnoser Σ_k. As the global scheme can detect the faults in $\mathcal{D}$, the difference $\mathcal{D} \setminus \cup_{k=1,\ldots,s}\mathcal{D}_k$ characterises the loss of detectability caused by the distributed diagnosis with respect to the global diagnosis.

Example 10.3 (**cont.**) **Ship with dual measurements**
Using the three residuals ρ_1, ρ_2, ρ_3, the distinguishability partition associated with the ship example was $\mathcal{D}^0 = \{OK\}$, $\mathcal{D}^1 = \{m_1\}$, $\mathcal{D}^2 = \{m_2\}$, $\mathcal{D}^3 = \{m_3\}$ and $\mathcal{D}^4 = \{m_4, c_1, c_2\}$. Assume a distributed diagnosis where the local diagnoser 1 runs only residual ρ_1. Considering only the first row of Table 10.2, its local distinguishability partition is $\left\{\mathcal{D}_1^0, \mathcal{D}_1^1\right\}$, with $\mathcal{D}_1^0 = \mathcal{D}^0 \cup \mathcal{D}^3 \cup \mathcal{D}^4$ and $\mathcal{D}_1^1 = \mathcal{D}^1 \cup \mathcal{D}^2$. □

Local versus global isolation. In order to evaluate the combined performance of the local diagnosers, one needs a *coordination or aggregation* procedure that provides a global diagnosis from the set of local diagnosis. Without loss of generality, the coordination procedure can be analysed for the case of two local diagnosers.

Theorem 10.2 *Let* $\Delta_k = \left\{\Delta_k^i, i \in i_k\right\}$ *be the local diagnosis delivered by two local diagnosers* $(k = 1, 2)$*, where* $\Delta_k^0 = OK \cup \overline{\mathcal{D}}_k$*,* $\overline{\mathcal{D}}_k$ *are the faults non-detectable* *by diagnoser k and each Δ_k^i, $(i \neq 0)$ is a minimal hitting set of the conflicts $\left\{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_f(k)\right\}$. Consistent diagnosis are obtained as*

$$\Delta_{12} = \left\{\Delta_{12}^{00}, \Delta_{12}^{0i}, \Delta_{12}^{j0}, \Delta_{12}^{ij}, i, j \neq 0\right\}, \tag{10.9}$$

where

$$\Delta_{12}^{00} = OK \cup \left(\overline{\mathcal{D}}_1 \times \overline{\mathcal{D}}_2\right) \tag{10.10}$$

$$i \neq 0 : \begin{cases} \Delta_{12}^{0i} = \overline{\mathcal{D}}_1 \times \left(\mathcal{D}_2^i \cap \overline{\mathcal{D}}_1\right) \\ \Delta_{12}^{i0} = \overline{\mathcal{D}}_2 \times \left(\mathcal{D}_1^i \cap \overline{\mathcal{D}}_2\right) \end{cases} \tag{10.11}$$

$$i, j \neq 0 : \Delta_{12}^{ij} = \mathcal{D}_1^i \times \mathcal{D}_2^j \tag{10.12}$$

under a simplification and a deletion rule:

1. *Simplification rule: a double fault that consists of a pair of identical faults is simplified into a single fault.*
2. *Deletion rule: non-minimal hitting sets are deleted.*

Understanding the coordination procedure is quite simple: let $\boldsymbol{r}(1)$ and $\boldsymbol{r}(2)$ be the residuals run by the local diagnosers. Associated with the signatures $\boldsymbol{r}(1) = \boldsymbol{r}_\text{s}(1) \cup \boldsymbol{r}_\text{f}(1)$ and $\boldsymbol{r}(2) = \boldsymbol{r}_\text{s}(2) \cup \boldsymbol{r}_\text{f}(2)$ are the conflicts $\mathcal{C}(1) = \{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_\text{f}(1)\}$ and $\mathcal{C}(2) = \{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_\text{f}(2)\}$. Four cases can be distinguished, according to the fact that $\mathcal{C}(1)$ and $\mathcal{C}(2)$ are empty or not.

- **Case 1**: $\boldsymbol{r}_\text{s}(1) = \boldsymbol{r}(1)$ and $\boldsymbol{r}_\text{s}(2) = \boldsymbol{r}(2)$. In this case, there is no conflict, and the two local diagnosis are $\Delta_1^0 = OK \cup \overline{\mathcal{D}}_1$ and $\Delta_2^0 = OK \cup \overline{\mathcal{D}}_2$, where $\overline{\mathcal{D}}_1$ (resp. $\overline{\mathcal{D}}_2$) are the faults non-detectable by Σ_1 (resp. by Σ_2). The global diagnosis consistent with the local ones is $OK \cup \left(\overline{\mathcal{D}}_1 \times \overline{\mathcal{D}}_2\right)$.
- **Case 2**: $\boldsymbol{r}_\text{s}(1) = \boldsymbol{r}(1)$ and $\boldsymbol{r}_\text{f}(2) \neq \emptyset$. In this case, the first diagnosis is $\Delta_1^0 = OK \cup \overline{\mathcal{D}}_1$, while the second is $\Delta_2^1 = \cup_{i \in i_2} \mathcal{D}^i$ where $\mathcal{D}^i$, $i \in i_2$ are the faults that have the signature $\boldsymbol{r}_\text{s}(2) \cup \boldsymbol{r}_\text{f}(2)$. The global diagnosis consistent with the local ones is $\overline{\mathcal{D}}_1 \times \left(\mathcal{D}_2^1 \cap \overline{\mathcal{D}}_1\right)$. Indeed, OK is inconsistent, since the residuals $\boldsymbol{r}_\text{f}(2)$ are falsified. Any fault in $\overline{\mathcal{D}}_1$ satisfies $\boldsymbol{r}(1)$ and any fault in $\mathcal{D}_2^1$ falsifies $\boldsymbol{r}_\text{f}(2)$. The reason why only faults in $\mathcal{D}_2^1 \cap \overline{\mathcal{D}}_1$ are considered is that faults that falsify $\boldsymbol{r}_\text{f}(2)$ must also satisfy $\boldsymbol{r}_\text{s}(1)$.
- **Case 3**: $\boldsymbol{r}_\text{f}(1) \neq \emptyset$, $\boldsymbol{r}_\text{s}(2) = \boldsymbol{r}(2)$ is similar to case 2.
- **Case 4**: $\boldsymbol{r}_\text{f}(1) \neq \emptyset$, $\boldsymbol{r}_\text{f}(2) \neq \emptyset$. In this case, the first diagnosis is $\Delta_1^1 = \cup_{i \in i_1} \mathcal{D}^i$ and the second is $\Delta_2^1 = \cup_{i \in i_2} \mathcal{D}^i$. Any double fault in $\mathcal{D}_1^1 \times \mathcal{D}_2^1$ is indeed possible.

The simplification rule follows from the fact that when the same fault is concluded to be present by each local diagnoser, the "double fault" is in fact a simple one. Finally, each Δ_k^i, $i \neq 0$ being a minimal hitting set of the conflicts $\{\mathcal{C}(\rho), \rho \in \boldsymbol{r}_\text{f}(k)\}$, a pair $\mathcal{D}_1^i \times \mathcal{D}_2^j$, $i, j \neq 0$ is a hitting set of $\left\{\mathcal{C}(\rho), \rho \in \cup_{k=1,2} \boldsymbol{r}_f(k)\right\}$ and it provides a possible conclusion as seen above. However, this hitting set may be non-minimal, and in this case it cannot be a possible diagnosis.

Remark 10.4 The coordination unit provides the overall diagnosis consistent with all the local diagnosers' conclusions. It may be implemented in the computing device of any subsystem (we are not discussing here its possible distribution). Technically, it receives the local subsystems' decisions and coordinates them according to the procedure of Theorem 10.2. Note that alternatively, the coordination could also be done by a direct combination of all the locally obtained signatures according to the global diagnoser's distinguishability table. □

Example 10.3 (**cont.**) **Ship with dual measurements**

Let us exemplify the coordination procedure in the ship with dual measurements assuming there are two computing devices Σ_1 and Σ_2, which are, respectively, interfaced with the measurement signals y_1, y_2 and y_3, y_4. Under the minimal information pattern, noted $\{(y_1, y_2), (y_3, y_4)\}$, they respectively run the residuals ρ_1 and ρ_3, since the computation form of ρ_2 is available to none of them. The local distinguishability tables are

	$\mathcal{D}^0 \cup \mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^1 \cup \mathcal{D}^2$
ρ_1	0	1

	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^3 \cup \mathcal{D}^4$
ρ_3	0	1

and the application of Theorem 10.2 gives

$\rho_1 \rho_3$	Local diagnosis Δ_1	Local diagnosis Δ_2	Coordinated diagnosis Δ_{12}
00	$\mathcal{D}^0 \cup \mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^0$
01	$\mathcal{D}^0 \cup \mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^3 \cup \mathcal{D}^4$
10	$\mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^1 \cup \mathcal{D}^2$
11	$\mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^3 \cup \mathcal{D}^4$	$\left(\mathcal{D}^1 \cup \mathcal{D}^2\right) \times \left(\mathcal{D}^3 \cup \mathcal{D}^4\right)$

It can be checked that the coordinated diagnosis is the same as the centralised diagnosis based on the two residuals ρ_1 and ρ_3. Indeed, the centralised distinguishability table would be

	$\mathcal{D}^0$	$\mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^3 \cup \mathcal{D}^4$
ρ_1	0	1	0
ρ_3	0	0	1

with the diagnosis

$\rho_1\,\rho_3$	Global diagnosis
00	$\mathcal{D}^0$
01	$\mathcal{D}^3 \cup \mathcal{D}^4$
10	$\mathcal{D}^1 \cup \mathcal{D}^2$
11	$\left(\mathcal{D}^1 \cup \mathcal{D}^2\right) \times \left(\mathcal{D}^3 \cup \mathcal{D}^4\right)$

Let us examine other information patterns. The publication of y_3 by Σ_2 and its subscription by Σ_1 allows the distributed diagnosis scheme ρ_1, ρ_2 by Σ_1 and ρ_3 by Σ_2. The local distinguishability tables and the coordination result are given in Tables 10.4, 10.5 and 10.6. □

Table 10.4 Local distinguishability table of Σ_1

	$\mathcal{D}^0 \cup \mathcal{D}^4$	$\mathcal{D}^1$	$\mathcal{D}^2$	$\mathcal{D}^3$
ρ_1	0	1	1	0
ρ_2	0	1	0	1

Table 10.5 Local distinguishability table of Σ_2

	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^3 \cup \mathcal{D}^4$
ρ_3	0	1

Table 10.6 Coordination table for Σ_1 and Σ_2

$\rho_1\,\rho_2\,\rho_3$	Δ_1	Δ_2	Δ_{12}
000	$\mathcal{D}^0 \cup \mathcal{D}^4$	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^0$
001	$\mathcal{D}^0 \cup \mathcal{D}^4$	$\mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^4$
010	$\mathcal{D}^3$	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	cannot happen
011	$\mathcal{D}^3$	$\mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^3 \cup \mathcal{D}^4$
100	$\mathcal{D}^2$	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^2$
101	$\mathcal{D}^2$	$\mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^2 \times \mathcal{D}^4$
110	$\mathcal{D}^1 \cup \left(\mathcal{D}^2 \times \mathcal{D}^3\right)$	$\mathcal{D}^0 \cup \mathcal{D}^1 \cup \mathcal{D}^2$	$\mathcal{D}^1 \cup \mathcal{D}^2$
111	$\mathcal{D}^1 \cup \left(\mathcal{D}^2 \times \mathcal{D}^3\right)$	$\mathcal{D}^3 \cup \mathcal{D}^4$	$\mathcal{D}^1 \times \left(\mathcal{D}^3 \cup \mathcal{D}^4\right) \cup \mathcal{D}^2 \times \mathcal{D}^3$

10.3.5 Distribution Schemes

We now describe different distributed diagnosis schemes, associated with different choices of the collection $\mathcal{C} = \{\mathcal{C}_k, k = 1, \ldots, s\}$.

Global diagnoser in one subsystem. Let $\mathcal{C}_{\max,k}$ be defined by $\mathcal{C}_i = \emptyset, i \neq k$ and $\mathcal{C}_k = \mathcal{C}$, and the information pattern $\mathcal{Z}_{\max,k}$ be such that $z_k = \boldsymbol{u} \cup \boldsymbol{y}$. Then subsystem Σ_k runs the global diagnosis algorithm, while the other subsystems do not perform any diagnosis at all.

Global diagnoser with replicas. Let K be a subset of subsystems, let $\mathcal{C}_{\max,K}$ be defined by $\mathcal{C}_i = \emptyset$, $(i \notin K)$ and $\mathcal{C}_i = \mathcal{C}$, $(i \in K)$ and let $\mathcal{Z}_{\max,K}$ be an information pattern such that $\forall i \in K : z_i = \boldsymbol{u} \cup \boldsymbol{y}$, then each subsystem in K runs a *replica* of the global diagnoser, while the other ones do not perform any diagnosis at all.

Decentralised diagnosers. Under the collection $\mathcal{C}_{\max} = \{\mathcal{C}_k = \mathcal{C}, k = 1, \ldots, s\}$ and the local information pattern $\mathcal{Z}_{\min} = \left\{\mathcal{K}_k = \boldsymbol{u}_k \cup \boldsymbol{y}_k, k = 1, \ldots, s\right\}$, each subsystem runs the residuals whose computation form uses the local variables $\boldsymbol{u}_k \cup \boldsymbol{y}_k$. This scheme needs no data transmission for the computation of the local residuals (but communication is still needed for the coordination task). It may yield weak results, because only a subset of the global residuals is run (for example, it is easy to see that a residual whose computation form needs measurements generated by sensors from different subsystems will not be run at all). It is of course possible to consider an even more reduced scheme with $\mathcal{C}_k \subseteq \mathcal{C}, k = 1, \ldots, s$ (at least one inclusion being strict) under the local information pattern.

Distributed diagnosers. The collection $\mathcal{C} = \{\mathcal{C}_k = \boldsymbol{\varphi_k} \cup \boldsymbol{\gamma_k}, k = 1, \ldots, s\}$ where $\boldsymbol{\varphi_k} \subseteq \boldsymbol{f}$ and $\boldsymbol{\gamma_k} \subseteq \boldsymbol{g}$ is the most general one. Associated with the information pattern $\mathcal{Z} = \left\{\left(z_k^a, z_k^s\right), k = 1, \ldots, s\right\}$, each subsystem Σ_k sees the global state $\boldsymbol{x}$ as $\left(\boldsymbol{\xi_k}, \overline{\boldsymbol{\xi}_k}\right)$, the global control $\boldsymbol{u}$ as $\left(z_k^a, \overline{z}_k^a\right)$ and the global measurements $\boldsymbol{y}$ as $\left(z_k^s, \overline{z}_k^s\right)$:

$$\begin{pmatrix} \dot{\xi}_k \\ \dot{\overline{\xi}}_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{\varphi_k}\left(\boldsymbol{\xi_k}, \overline{\boldsymbol{\xi}_k}, z_k^a, \overline{z}_k^a, \boldsymbol{d}, t\right) \\ \overline{\varphi_k}\left(\boldsymbol{\xi_k}, \overline{\boldsymbol{\xi}_k}, z_k^a, \overline{z}_k^a, \boldsymbol{d}, t\right) \end{pmatrix} \tag{10.13}$$

$$\begin{pmatrix} z_k^s \\ \overline{z}_k^s \end{pmatrix} = \begin{pmatrix} \gamma_k\left(\boldsymbol{x}, z_k^a, \overline{z}_k^a, \boldsymbol{d}, t\right) \\ \overline{\gamma_k}\left(\boldsymbol{x}, z_k^a, \overline{z}_k^a, \boldsymbol{d}, t\right) \end{pmatrix} \tag{10.14}$$

which results in the local residuals $\boldsymbol{r}_k\left(\boldsymbol{\varphi_k} \cup \boldsymbol{\gamma_k}, z_k^a \cup z_k^s\right)$.

Remark 10.5 The diagnosis decomposition needs by no means be identical to the control decomposition (10.3) and (10.4). Taking $\boldsymbol{\varphi_k} = \boldsymbol{f}_k, k = 1, \ldots, s$ is sometimes justified in the literature by the argument that under the decentralised information pattern, the local residuals $\boldsymbol{r}_k\left(\boldsymbol{\varphi_k} \cup \boldsymbol{g}_k, \boldsymbol{u}_k \cup \boldsymbol{y}_k\right)$ are sensitive only to faults in Σ_k, but this is true only if an over-constrained subsystem exists in the decomposition of Σ_k. In general, both the interconnection variables and the consideration of wider information patterns introduce constraints from other subsystems whose elimination (when possible) results in residual structures that do not contain only local constraints.

Remark 10.6 z_k^a and z_k^s being defined by the given information pattern, the largest set of local residuals is obtained with $\mathcal{C}_k = \boldsymbol{f} \cup \gamma_k$, $k = 1, \ldots, s$, where γ_k is determined by z_k^s. This is nothing but the subset of global residuals whose computation form is available to Σ_k. □

Replicas. Local diagnoser residuals may have non-empty intersections. For example, two subsystems that share data both see a common subset of constraints by means of the same known variables, which results in identical local residuals. The same conclusion holds when two subsystems publish their data, and each of them subscribes to the data published by the other one. It is a design decision to implement several replicas of the same residuals in several local diagnosers. The decision has a cost associated with multiple calculations of the same residuals, but it allows to detect faults that might occur in the computing devices, using a voting scheme. Moreover, the diagnosis remains available under such faults, if the number of replicas is large enough. Note that the local to global coordination rules remain unchanged when replicas are used, as it can be checked from the following example.

Example 10.3 (**cont.**) **Ship with dual measurements**

Assume that Σ_1 runs ρ_1, ρ_2 and Σ_2 runs ρ_2, ρ_3. The local distinguishability partitions become

	$\mathcal{D}^0$	$\mathcal{D}^1$	$\mathcal{D}^2$	$\mathcal{D}^3$	$\mathcal{D}^4$
ρ_1	0	1	1	0	0
ρ_2	0	1	0	1	0

and

	$\mathcal{D}^0$	$\mathcal{D}^1$	$\mathcal{D}^2$	$\mathcal{D}^3$	$\mathcal{D}^4$
ρ_2	0	1	0	1	0
ρ_3	0	0	0	1	1

and they still provide the coordinated diagnosis:

$\rho_1\rho_2\rho_3$	Δ_1	Δ_2	Δ_{12}
000	$\mathcal{D}^0 \cup \mathcal{D}^4$	$\mathcal{D}^0 \cup \mathcal{D}^2$	$\mathcal{D}^0$
001	$\mathcal{D}^0 \cup \mathcal{D}^4$	$\mathcal{D}^4$	$\mathcal{D}^4$
010	$\mathcal{D}^3$	$\mathcal{D}^1$	cannot happen
011	$\mathcal{D}^3$	$\mathcal{D}^3 \cup \mathcal{D}^4 \times \mathcal{D}^1$	$\mathcal{D}^3 \cup \mathcal{D}^4$
100	$\mathcal{D}^2$	$\mathcal{D}^0 \cup \mathcal{D}^2$	$\mathcal{D}^2$
101	$\mathcal{D}^2$	$\mathcal{D}^4$	$\mathcal{D}^2 \times \mathcal{D}^4$
110	$\mathcal{D}^1 \cup \mathcal{D}^2 \times \mathcal{D}^3$	$\mathcal{D}^1$	$\mathcal{D}^1 \cup \mathcal{D}^2$
111	$\mathcal{D}^1 \cup \mathcal{D}^2 \times \mathcal{D}^3$	$\mathcal{D}^3 \cup \mathcal{D}^4 \times \mathcal{D}^1$	$\mathcal{D}^1 \times \left(\mathcal{D}^3 \cup \mathcal{D}^4\right) \cup \mathcal{D}^2 \times \mathcal{D}^3$

10.4 Design of the Local Diagnosers

10.4.1 Specifications

The design of a distributed diagnosis scheme aims at satisfying functional and fault-tolerance specifications, under local computing capacity constraints, at a minimal communication cost.

Functional specifications. The functional specifications encompass the following points:

- The detectability and isolability performances of the global system $(\mathcal{C}, \mathcal{K})$ are entirely defined by the set of residuals $\boldsymbol{r}\ (\mathcal{C}, \mathcal{K})$. In what follows, it is supposed that the diagnosis performances of the distributed system are wished to be the same as those of the centralised system. However, the approach can be applied whatever the subset of residuals $\boldsymbol{r} \subseteq \boldsymbol{r}\ (\mathcal{C}, \mathcal{K})$ that are wished to be implemented.
- The computing cost of a subsystem Σ_k which has been assigned the set of residuals $\boldsymbol{r}_k$ is a function $h\ (\boldsymbol{r}_k)$ assumed to be known. The capacity constraint is expressed as $h\ (\boldsymbol{r}_k) \leq h_k, k = 1, \ldots, s$.
- The communication cost depends on the information pattern that is implemented, and it is an increasing function of the set of communicated variables. For the sake of simplicity, information patterns are considered first under the publisher/subscriber scheme. The extension to the bilateral agreement scheme is considered next.

Fault-tolerance specifications. Fault-tolerance specifications may be added to the functional specifications. They specify the diagnosis performances that are still to be achieved should faults occur in

- the sensors or in the communication system (they decrease the set of known inputs that can be used by each local diagnoser),
- the process components (they decrease the set of healthy constraints upon which the set of residuals to be used depends), and
- the local computing devices (the local diagnosis from faulty devices cannot be used in the coordination procedure).

10.4.2 Simple Distribution Problem

Let us start with the following simple problem associated with the functional specifications: assuming there is no capacity constraint associated with the subsystems Σ_k, $(k = 1, \ldots, s)$ distribute the residual computations among them so as to obtain the same diagnosis performances as in the centralised scheme, at a minimal communication cost.

Let $\boldsymbol{r}\left(\mathcal{C},\mathcal{K}\right)$ be the residuals of the global system, and $\boldsymbol{r}_k\left(\mathcal{C},\mathcal{Z}\right)$ be the residuals whose computation form is available to subsystem Σ_k under the information pattern $\mathcal{Z}$. Then, the distributed scheme achieves the same performances as the centralised scheme if and only if

$$\cup_{k=1,\dots,s}\,\boldsymbol{r}_k\left(\mathcal{C},\mathcal{Z}\right)=\boldsymbol{r}\left(\mathcal{C},\mathcal{K}\right) \tag{10.15}$$

i.e. the residuals $\boldsymbol{r}_k\left(\mathcal{C},\mathcal{Z}\right)$, $k=1,\dots,s$ cover the residuals $\boldsymbol{r}\left(\mathcal{C},K\right)$.

From the monotonicity property $\mathcal{Z}^+\succeq\mathcal{Z}\Rightarrow\boldsymbol{r}_k\left(\mathcal{C},\mathcal{Z}^+\right)\supseteq\boldsymbol{r}_k\left(\mathcal{C},\mathcal{Z}\right)$, it follows that if Eq. (10.15) is not satisfied under an information pattern $\mathcal{Z}$ it may be satisfied under a wider one $\mathcal{Z}^+$. In the publisher/subscriber scheme, wider information patterns are obtained by publishing more variables. The set of all possible information patterns is therefore the lattice of all publishable variables, namely $L=2^{\boldsymbol{u}\cup\boldsymbol{y}}$, which is organised into levels L_i that contain subsets of i variables. The algorithm that solves the simple distribution problem is therefore

Algorithm 10.1 *Simple distribution*

```
         Given: a set $\boldsymbol{r}(\mathcal{C},\mathcal{K})$ of residuals to be covered
                a system decomposition into subsystems $\Sigma_k$ with local known
                variables $\mathcal{K}_k = \boldsymbol{u}_k \cup \boldsymbol{y}_k$
Initialisation: $E_i = L_i,\quad i = 0, 1, \dots |\boldsymbol{u} \cup \boldsymbol{y}|$.
          Loop: While $E_t \neq \emptyset$
            1.  for each subset of published variables $\boldsymbol{z} \in E_t$, identify the
                subsets of residuals $\boldsymbol{r}_k(\mathcal{C},\mathcal{Z}) \subseteq \boldsymbol{r}(\mathcal{C},\mathcal{K})$, $(k = 1, \dots, s)$
                whose computation form is available, and update $E_t$ as $E_t \setminus \{\boldsymbol{z}\}$
            2.  If Eq. (10.15) is satisfied, $\boldsymbol{z}$ solves the problem. List $\boldsymbol{z}$ in the
                set of solutions $Z^*$ and update $E_{t+1}$ as $E_{t+1} \setminus \mathcal{P}(Z^*)$ where
                $\mathcal{P}(Z^*) = \cup_{\boldsymbol{z} \in Z^*} \mathcal{P}(\boldsymbol{z})$ and $\mathcal{P}(\boldsymbol{z})$ are the predecessors of $\boldsymbol{z}$ in
                the lattice $L$.
        Result: List $Z^*$ of minimal subsets of variables to be published in
                the publisher/subscriber scheme in order for the distributed
                diagnosis to achieve the same performance as the centralised
                diagnosis.
```

Comments.

1. Since any solution $\boldsymbol{z}\in Z^*$ results in the running of all the residuals $\boldsymbol{r}\left(\mathcal{C},\mathcal{K}\right)$, any predecessor of $\boldsymbol{z}$ also results in running all the residuals. Because there are more available data to the subsystems, some residuals may be replicated in several subsystems.

2. The process considers wider and wider information patterns, so it must eventually terminate, with a non-empty set of solutions. Indeed, the worst case in which all the publishable variables are published is associated with the information pattern $\mathcal{Z}_{\max}$ that implements the whole set of residuals $\boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$ in each subsystem.
3. The presentation has been aimed at distributing all the residuals in $\boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$ but the approach can be applied whatever the subset of residuals $\boldsymbol{r} \subseteq \boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$ that are wished to be implemented.

Example 10.3 (**cont.**) **Ship with dual measurements**
Let us illustrate the simple distribution procedure with the ship example whose results have been presented earlier. Let the two subsystems be Σ_1 with local sensors $\{y_1, y_2\}$ and Σ_2 with local sensors $\{y_3, y_4\}$. The specification is that the distributed diagnosis should be as powerful as the centralised diagnosis.

The procedure starts with the minimal information pattern

$$\mathcal{Z}_{\min} = \{(y_1, y_2), (y_3, y_4)\}$$

associated with the decentralised system.

The first iteration provides the distribution ρ_1 assigned to Σ_1 and ρ_3 assigned to Σ_2 as already seen, which is not admissible because there is a loss of detectability and isolability with respect to the centralised scheme. In the second iteration, the constraints are considered under wider information patterns. Under the publisher/subscriber scheme, four wider information patterns can be obtained by publishing one single variable, namely

$$\begin{aligned}
\mathcal{Z}_1 &= \{(y_1, y_2), (y_1, y_3, y_4)\} \\
\mathcal{Z}_2 &= \{(y_1, y_2), (y_2, y_3, y_4)\} \\
\mathcal{Z}_3 &= \{(y_1, y_2, y_3), (y_3, y_4)\} \\
\mathcal{Z}_4 &= \{(y_1, y_2, y_4), (y_3, y_4)\}\,.
\end{aligned}$$

From the residuals (10.8), the associated decompositions are

Information pattern	Σ_1	Σ_2
$\mathcal{Z}_1$	ρ_1	ρ_2, ρ_3
$\mathcal{Z}_2$	ρ_1	ρ_3
$\mathcal{Z}_3$	ρ_1, ρ_2	ρ_3
$\mathcal{Z}_4$	ρ_1	ρ_3

Only $\mathcal{Z}_1$ and $\mathcal{Z}_3$ improve the diagnosis capability. The diagnosis performances under $\mathcal{Z}_3$ are given in Tables 10.4 and 10.5. They show that there is no loss of detectability/isolability with respect to the global diagnosis scheme. It can be checked from the local tables

	OK	m_1	m_2	m_3	m_4	c_1	c_2
ρ_1	0	1	1	0	0	0	0

and

	OK	m_1	m_2	m_3	m_4	c_1	c_2
ρ_2	0	1	0	1	0	0	0
ρ_3	0	0	0	1	1	1	1

that the same conclusion holds under $\mathcal{Z}_1$, so the two schemes satisfy the diagnosis specifications.

Finally, the information patterns $\mathcal{Z}_1$ and $\mathcal{Z}_3$ are the minimal ones for which the diagnosis specifications can be obtained. Considering wider information patterns is not necessary (unless replications are wished), since the diagnosis performances would not be increased, but the local computing costs could only be increased (because more residuals would be computed in each subsystem). In order to illustrate this point, let us investigate the case where two variables are published. There are six possible information patterns, which lead to six possible distributed schemes, according to the table:

Information pattern	Σ_1	Σ_2
$\mathcal{Z}_{12} = \{(y_1, y_2), (y_1, y_2, y_3, y_4)\}$	ρ_1	ρ_1, ρ_2, ρ_3
$\mathcal{Z}_{13} = \{(y_1, y_2, y_3), (y_1, y_3, y_4)\}$	ρ_1, ρ_2	ρ_2, ρ_3
$\mathcal{Z}_{14} = \{(y_1, y_2, y_4), (y_1, y_3, y_4)\}$	ρ_1	ρ_2, ρ_3
$\mathcal{Z}_{23} = \{(y_1, y_2, y_3), (y_2, y_3, y_4)\}$	ρ_1, ρ_2	ρ_3
$\mathcal{Z}_{24} = \{(y_1, y_2, y_4), (y_2, y_3, y_4)\}$	ρ_1	ρ_3
$\mathcal{Z}_{34} = \{(y_1, y_2, y_3, y_4), (y_3, y_4)\}$	ρ_1, ρ_2	ρ_3

All schemes (except $\mathcal{Z}_{24}$) satisfy the diagnosis specifications since they allow to distribute the computation of all residuals. Note that schemes $\mathcal{Z}_{12}$ and $\mathcal{Z}_{13}$ implement residuals replications. Note also that any information pattern wider than the two minimal patterns $\mathcal{Z}_1$ and $\mathcal{Z}_3$ under which the specifications are satisfied also satisfies the specifications, as shown in Fig. 10.6, where the different information patterns are displayed along with the associated residual distribution. Information patterns under which the distributed diagnosis specifications are satisfied are in white, and the minimal ones have a bold contour. □

10.4.3 Distribution Under Computing Cost Constraints

The simple distribution problem does not take into account the possible limitations in the local computing power of the different subsystems. Assume there is a known function $h_k(\rho)$ associated with each pair (ρ, k), $\rho \in \boldsymbol{r}(\mathcal{C}, \mathcal{K})$, $(k = 1, \ldots, s)$ that evaluates the computing cost of a residual ρ by the computing device of subsystem Σ_k and that subsystem Σ_k can devote only an amount h_k of computing effort to the distributed diagnosis task. Then, assuming that computing costs are additive, the previous distribution problem must take into account the computing cost constraints:

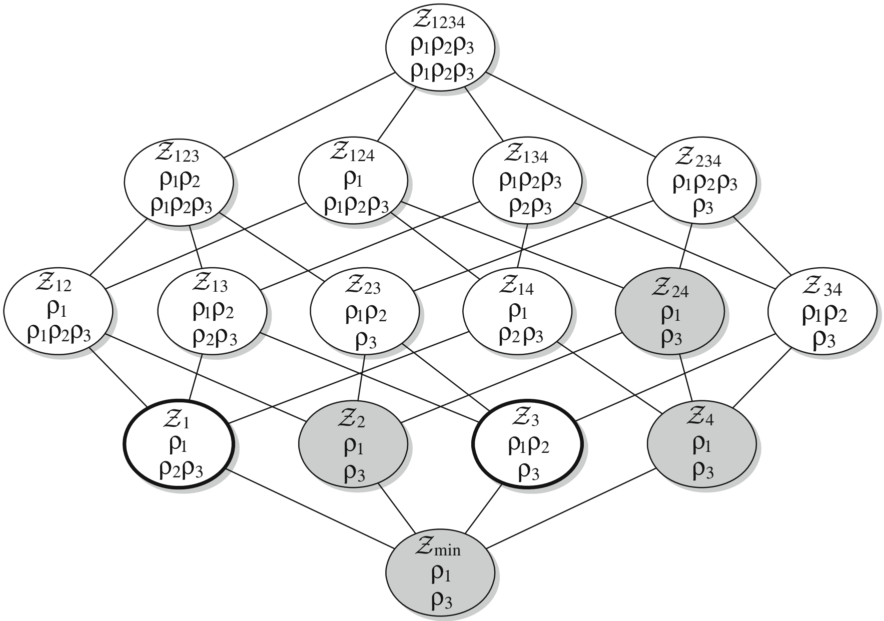
$\mathcal{Z}_{1234}$ $\rho_1\rho_2\rho_3$ $\rho_1\rho_2\rho_3$
$\mathcal{Z}_{123}$ $\rho_1\rho_2$ $\rho_1\rho_2\rho_3$
$\mathcal{Z}_{124}$ ρ_1 $\rho_1\rho_2\rho_3$
$\mathcal{Z}_{134}$ $\rho_1\rho_2\rho_3$ $\rho_2\rho_3$
$\mathcal{Z}_{234}$ $\rho_1\rho_2\rho_3$ ρ_3
$\mathcal{Z}_{12}$ ρ_1 $\rho_1\rho_2\rho_3$
$\mathcal{Z}_{13}$ $\rho_1\rho_2$ $\rho_2\rho_3$
$\mathcal{Z}_{23}$ $\rho_1\rho_2$ ρ_3
$\mathcal{Z}_{14}$ ρ_1 $\rho_2\rho_3$
$\mathcal{Z}_{24}$ ρ_1 ρ_3
$\mathcal{Z}_{34}$ $\rho_1\rho_2$ ρ_3
$\mathcal{Z}_{1}$ ρ_1 $\rho_2\rho_3$
$\mathcal{Z}_{2}$ ρ_1 ρ_3
$\mathcal{Z}_{3}$ $\rho_1\rho_2$ ρ_3
$\mathcal{Z}_{4}$ ρ_1 ρ_3
$\mathcal{Z}_{\min}$ ρ_1 ρ_3

Fig. 10.6 Information patterns and diagnosis distribution in the ship example

$$\sum_{\rho \in \boldsymbol{r}_k} h_k\left(\rho\right) \leq h_k. \tag{10.16}$$

Starting with the results of the simple distribution algorithm, it is easily seen that if the set Z^* contains at least one solution that satisfies the computing cost constraints, then the constrained problem is solved, by discarding those solutions that are inadmissible.

Two situations must be distinguished when all solutions in Z^* are inadmissible:

- First, an inadmissible solution can be transformed into an admissible one, if there exists subsets of residuals in the overloaded subsystems whose deletion leads to an admissible computing cost, but does not degrade the diagnosis performance because they are replicas of residuals computed in other—non-overloaded—subsystems.
- If no such possibility exists, non-minimal subsets of publishable data must be considered, in order to provide non-overloaded subsystem with replicas of residuals that could be deleted from overloaded subsystems.

In order to implement this procedure, the previous algorithm is modified as follows.

Algorithm 10.2 *Distribution under cost constraints*

Given: A set $\boldsymbol{r}\,(\mathcal{C},\mathcal{K})$ of residuals to be covered
a system decomposition into subsystems Σ_k with local known variables $\mathcal{K}_k = \boldsymbol{u}_k \cup \boldsymbol{y}_k$, known computing costs $h_k\,(\rho)$, $\rho \in \boldsymbol{r}\,(\mathcal{C},\mathcal{K})$, and known computing power limitations h_k

Initialisation: $E_i = L_i, \quad i = 0, 1, \ldots |\boldsymbol{u} \cup \boldsymbol{y}|$.

Loop: While $E_t \neq \emptyset$:

1. For each subset of published variables $z \in E_t$, identify the subsets of residuals $\boldsymbol{r}_k\,(\mathcal{C},\mathcal{Z}) \subseteq \boldsymbol{r}\,(\mathcal{C},\mathcal{K})$, $(k = 1, \ldots, s)$ whose computation form is available, and update E_t as $E_t \setminus \{z\}$,

2. If Eqs. (10.15) and (10.16) are satisfied, or if Eq. (10.15) is satisfied and Eq. (10.16) is not satisfied but becomes satisfied by deleting the replicated residuals in the overloaded subsystems, z solves the problem. List z in the set of solutions Z^* and update E_{t+1} as $E_{t+1} \setminus \mathcal{P}\,(Z^*)$ where $\mathcal{P}\,(Z^*) = \cup_{z \in Z^*}\mathcal{P}\,(z)$ and $\mathcal{P}\,(z)$ are the predecessors of z in the lattice L.

Result: List Z^* of minimal subsets of variables to be published in the publisher/subscriber scheme in order for the distributed diagnosis to achieve the same performance as the centralised diagnosis while satisfying the computation cost constraints.

Comment. Since it explores wider and wider information patterns, the algorithm must eventually terminate. However, a solution is not guaranteed to exist. A necessary and sufficient condition for a solution to exist is that it exists under the maximal information pattern. In that case, all subsystems are able to run all residuals $\boldsymbol{r}\,(\mathcal{C},\mathcal{K})$, and the deletion of replicated residuals problem boils down to finding a partition of the set $\boldsymbol{r}\,(\mathcal{C},\mathcal{K})$ into s classes such that the computing cost constraints are satisfied. Let $\sigma_k\,(\rho)$ be the binary variables such that $\sigma_k\,(\rho) = 1$ when residual ρ is assigned to subsystem Σ_k and $\sigma_k\,(\rho) = 0$ when residual ρ is not assigned to subsystem Σ_k. Then the residual distribution problem under computation cost constraints has a solution if and only if the constraint satisfaction problem,

$$\forall \rho \in \boldsymbol{r}\,(\mathcal{C},\mathcal{K}) : \sum_{k=1,\ldots,s} \sigma_k\,(\rho) = 1 \tag{10.17}$$

$$k = 1, \ldots, s : \sum_{\rho \in \boldsymbol{r}(\mathcal{C},\mathcal{K})} \sigma_k\,(\rho)\, h_k\,(\rho) \leq h_k, \tag{10.18}$$

has a solution, which can easily be checked since it is a classical task allocation problem (algorithms to solve a version of this problem—namely finding maximal matchings in a bipartite graph—were given in Chap. 5).

10.4.4 The Bilateral Agreements Scheme

In the bilateral agreements scheme, the set of all subsystems is partitioned into equivalence classes such that subsystems in the same class share all their data. Denoting by $\mathcal{K}_{\mathrm{K}}$ the known data available to all subsystems in a class $\{\Sigma_k, k \in K\}$, these bilateral agreements result in the residual assignments $\boldsymbol{r}_k\,(\mathcal{C}, \mathcal{K}_{\mathrm{K}}) = \boldsymbol{r}_{\mathrm{K}}, k \in K$. It follows that for each residual possibly run by subsystem Σ_k, there are $|K| - 1$ replicas possibly run by the other subsystems Σ_j, $(j \neq k)$ in the same class.

The following algorithm explores the increasing levels of a hierarchy built on the atomic decomposition Σ_k, $(k = 1, \ldots, s)$. At each level of the hierarchy, two subsystems are merged according to some merging policy, for example, the two subsystems whose merger implies the smallest communication cost, the two subsystems whose merger implies the largest set of computable residuals, the two subsystems with the best efficiency ratio computed from the increase in the communication cost versus the increase in the number of computable residual, etc. The satisfaction of the computing cost constraints is achieved by deleting from the overloaded subsystems those residuals whose replica is present in some underloaded subsystem.

Algorithm 10.3 *Bilateral agreements*

Given: A set $\boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$ of residuals to be covered
a system decomposition into subsystems Σ_k with local known variables $\mathcal{K}_k = \boldsymbol{u}_k \cup \boldsymbol{y}_k$, known computing costs $h_k\,(\rho)$, $\rho \in \boldsymbol{r}\,(\mathcal{C}, \mathcal{K})$, and known computing power limitations h_k.

Initialisation: $E_0 = \left\{E_{0,k} = \Sigma_k, \;\; k = 1, \ldots, s\right\}$

Loop: While E_t is not a singleton

1. For each pair of classes $E_{t,i}$ and $E_{t,j}$, evaluate their possible merger in terms of the induced communication cost and of the residuals that become computable, and select the pair whose merger is preferred according to the selected merging policy. Update E_{t+1} by replacing $\left\{E_{t,i}, E_{t,j}\right\}$ in E_t by $E_{t,i} \cup E_{t,j}$.

2. If all residuals are covered and the computing cost constraints are satisfied, or if they become satisfied by deleting the replicated residuals in the overloaded subsystems, the decomposition E_{t+1} solves the problem.

Result: a decomposition of the system into equivalence classes associated with bilateral communication agreements that achieves the same performance as the centralised diagnosis while satisfying the computation cost constraints.

Comments.

1. Since it explores increasing levels of the system hierarchy, the algorithm must eventually terminate. The necessary and sufficient condition for a solution to exist (and therefore to be found) is the same as in the publisher/subscriber scheme, that has been given in Eqs. (10.17) and (10.18).
2. It is well known that hierarchical procedures are by no way optimal, since mergers are performed at each level following a greedy approach (the best merger at a given level is not necessarily the best one from a global point of view), and are never un-merged. However, they are very popular because of their simplicity. They are in general run several times, under several merging policies, which allows for comparison between the results and get a good idea of the main features of the solutions.

Example 10.4 Hierarchical distribution algorithm
In this example, we consider the distribution of a set of nine residuals among four subsystems, under the bilateral communication scheme. The structures of the residuals computation form are

	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}
ρ_1						1		1			
ρ_2		1	1		1						
ρ_3						1	1				
ρ_4		1		1							
ρ_5								1		1	1
ρ_6	1	1									
ρ_7							1	1	1		
ρ_8						1	1			1	
ρ_9	1	1	1	1							

and the local data are given by

Subsystem	Σ_1	Σ_2	Σ_3	Σ_4
Local data	z_1, z_2	z_3, z_4, z_5	z_6, z_7	z_8, z_9, z_{10}, z_{11}

The atomic decomposition associated with the decentralised scheme leads to the computable residuals

E_0	Σ_1	Σ_2	Σ_3	Σ_4
Computable residuals	ρ_6	$\emptyset$	ρ_3	ρ_5

that do not cover the whole set $\rho_1, \ldots, \rho_9$, and therefore the information exchange must be increased. At the first level of the hierarchy, there are six possible bilateral agreements, which give the following results:

Mergers	$\Sigma_1 \cup \Sigma_2$	$\Sigma_1 \cup \Sigma_3$	$\Sigma_1 \cup \Sigma_4$
Computable residuals	$\rho_2\,\rho_4\,\rho_6\,\rho_9$	$\rho_3\,\rho_6$	$\rho_5\,\rho_6$
Communication load	$z_1\,z_2\,z_3\,z_4\,z_5$	$z_1\,z_2\,z_6\,z_7$	$z_1\,z_2\,z_8\,z_9\,z_{10}\,z_{11}$

Mergers	$\Sigma_2 \cup \Sigma_3$	$\Sigma_2 \cup \Sigma_4$	$\Sigma_3 \cup \Sigma_4$
Computable residuals	$\rho_3\,\rho_8$	ρ_5	$\rho_1\,\rho_3\,\rho_5\,\rho_7\,\rho_8$
Communication load	$z_3\,z_4\,z_5\,z_6\,z_7$	$z_3\,z_4\,z_5\,z_8\,z_9\,z_{10}\,z_{11}$	$z_6\,z_7\,z_8\,z_9\,z_{10}\,z_{11}$

Assume that the merging policy that gives the largest number of computable residuals is chosen. There are two decompositions at level 1 that both allow to compute six residuals, namely

E_{11}	Σ_1	Σ_2	$\Sigma_3 \cup \Sigma_4$
Computable residuals	ρ_6	$\emptyset$	$\rho_1\,\rho_3\,\rho_5\,\rho_7\,\rho_8$

and

E_{12}	$\Sigma_1 \cup \Sigma_2$	Σ_3	Σ_4
Computable residuals	$\rho_2\,\rho_4\,\rho_6\,\rho_9$	ρ_3	ρ_5

Note that E_{12} needs less communication than E_{11}, but neither E_{11} nor E_{12} covers the whole set of wished residuals, so more communication has to be introduced by considering the second level of the hierarchy.

From aggregating E_{11} and E_{12}, one gets

E_{21}	$\Sigma_1 \cup \Sigma_2$	$\Sigma_3 \cup \Sigma_4$
Computable residuals	$\rho_2\,\rho_4\,\rho_6\,\rho_9$	$\rho_1\,\rho_3\,\rho_5\,\rho_7\,\rho_8$

that covers all the residuals, and from E_{12} one gets three possibilities, associated with the mergers $\Sigma_3 \cup \Sigma_4$, $\Sigma_1 \cup \Sigma_2 \cup \Sigma_3$ and $\Sigma_1 \cup \Sigma_2 \cup \Sigma_4$ but none of them covers the whole set of residuals. Note that in the solution E_{21}, both Σ_1 and Σ_2 are able to run the residuals $\rho_2\rho_4\rho_6\rho_9$ and both Σ_3 and Σ_4 are able to run the residuals $\rho_1\rho_3\rho_5\rho_7\rho_8$. Assuming each subsystem has a sufficient computing power, the existence of the replicas makes the distributed scheme tolerant to faults in the individual computing devices: for example, in the presence of a complete failure of Σ_1, the residuals $\rho_2\rho_4\rho_6\rho_9$ could still be run by Σ_2 (of course, one should also consider in this case the effects of such a failure on the control functionalities, but this is not the topic of this section).

Assuming limited computing powers that do not allow full duplication, the set of residuals $\rho_2\rho_4\rho_6\rho_9$ should be split into two subsets, respectively, run in Σ_1 and Σ_2, according to the classical task allocation problem under constraints, a version of which has been used in the comment on p. 499 to evaluate the existence of a solution to the distribution problem under constraints. □

10.4.5 Fault-Tolerant Distributed Diagnosis

Fault-tolerant distributed diagnosis considers the effect of faults on the diagnosis capability of a distributed diagnosis system. It will not be developed in detail, since most of the tools that are useful for the analysis and the design of fault-tolerant diagnosis have been presented in this chapter and in previous chapters, as it appears from the following analysis of the different fault consequences.

Faults in the process components. Faults in the process components decrease the set of constraints that can be used to build the residuals. Less constraints means less residuals, which means less detectability and distinguishability. The analysis of the fault tolerance of a given diagnosis system is therefore nothing but the analysis of the subsets of residuals that still allow to perform the desired detection and isolation specifications. Considering subsets of residuals (i.e. residual configurations) is quite similar to considering actuator or sensor configurations as in Chap. 8.

Faults in the sensors or in the communication network. Faults in the sensors or in the communication network decrease the set of known variables that are available to the local computing devices of the distributed system. Less known variables means less residuals, which brings back to the above problem.

Faults in the local computing devices. Faults in the local computing devices result in erroneous local diagnosis. Using replicas of the same residuals in different computing devices allows to detect inconsistencies by means of appropriate voting schemes. The general problem has been thoroughly studied in the computer science community, and the reader is referred to the bibliographical notes for an overview of the main results.

10.5 Fault-Tolerant Control by Information Pattern Reconfiguration

The previous sections have shown the prominent role of the information pattern in the design of a distributed diagnosis algorithm. Similarly, the role of the information pattern is the main feature that distinguishes fault-tolerant control in distributed systems from fault-tolerant control in embedded systems. Other features are that

- due to the interactions between subsystems, the specifications to be satisfied in normal and in faulty operations must be considered at the system level, while the control is designed at the subsystem level; and
- the fault recovery process is desired to be limited to the smallest possible number of subsystems.

These features are now addressed in the frame of reconfiguration-based fault tolerance.

10.5.1 Admissibility and Reconfigurability

Remember that fault tolerance is the property that some specification $\mathcal{P}$ satisfied by the nominal system is also satisfied in the presence of faults (performance degradation may be allowed by introducing a less-demanding specification when faults occur). When distributed systems are considered, an important question to decide is whether each subsystem is responsible for finding an admissible control that achieves the part of the global specification it has been assigned to fulfil (in this case, the specification is said to be *decomposable*), or whether system-level admissibility is to be considered (this is the *non-decomposable* specifications case).

Definition 10.3 (*Decomposable specification*) A specification $\mathcal{P}$ is *decomposable* if it is equivalent to some set $\{\mathcal{P}_k,\ k = 1, \ldots, s\}$ where $\mathcal{P}_k$ is a specification of subsystem Σ_k.

Due to the coupling variables $\overline{\boldsymbol{x}}_k$ in Eq. (10.3), it appears that not all specifications are decomposable. In the sequel of this chapter, we consider non-decomposable specifications. Indeed, there is no interest, when addressing fault-tolerant distributed systems, in considering decomposable specifications: should the specification be decomposable, one could simply address fault tolerance within each subsystem, using the local subsystem model, and treating the interconnection variables as unknown inputs. This would of course result in a distributed recovery algorithm (each subsystem recovers its own faults, independently of the others), but would not bring much new insight to the fault-tolerance problem of the distributed system, since each subsystem would be treated as a system of its own, using the methods presented in the previous chapters.

Example 10.5 Decomposable specifications

- The controllability of the system in Example 10.1 is a structural property that depends on the pair $(\boldsymbol{A}, \boldsymbol{B})$ given in Eq. (10.5). It is in general not equivalent to the controllability of the four subsystems associated with the respective pairs of matrices:

Subsystem	Matrix $\boldsymbol{A}$	Matrix $\boldsymbol{B}$
Σ_1	$\begin{pmatrix} -1 & 2 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$
Σ_2	-2	1
Σ_3	$\begin{pmatrix} -2 & 0.4 \\ 1 & -3 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$
Σ_4	-4	2

- Given an output feedback $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}$ and a positive value of α, the system Σ is α-stable if there exists a Lyapunov function $V = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$ such that $\dot{V} \leq -\alpha V$ along its trajectories, i.e.

$$\boldsymbol{Q}\,(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\boldsymbol{C}) + (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{\mathrm{T}}\,\boldsymbol{Q} + \alpha\boldsymbol{Q} \leq 0.$$

  This is a non-structural property that depends on the control law (via the output feedback matrix $\boldsymbol{K}$), and again it is not equivalent to the α-stability of each subsystem $\Sigma_k, k =$

$1, \ldots, 4$ because the subsystems are coupled. As a matter of fact, it is well known that the interconnection of several stable subsystems might well result in an unstable system, as illustrated by the very simple example

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -1 & \theta_{12} \\ \theta_{21} & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

which is unstable for any values of the interconnection parameters such that $\theta_{12}\theta_{21} > 2$ although the self-dynamics of the two subsystems, respectively, $\dot{x}_1 = -x_1$ and $\dot{x}_2 = -2x_2$ are stable.

- Examples of degraded specifications would be to accept stabilisability instead of controllability, or a smaller decay rate in the α-stability specification. □

Recoverable faults. Let us first consider actuator faults under the reconfiguration strategy (sensor or system component faults are treated the same way). Let $I_\mathrm{N} \subseteq I$ be an actuator configuration, i.e. the actuators that are available to achieve specification $\mathcal{P}$, if possible, when actuators in $I_\mathrm{F} = I \backslash I_\mathrm{N}$ are faulty and have been switched-off. The set of all possible configurations (including the nominal one) is 2^I the power set of I, i.e. the set of all its subsets. Remember that 2^I is a lattice, a mathematical structure whose properties have already been used in previous chapters to address implementation issues and evaluation measures. We will now use the lattice tool for distributed control systems, by considering the set of all possible information patterns, and analysing specific monotonicity properties of interest for the reconfiguration problem.

Remark that although configuration I_N is decomposed into $\left\{I_{\mathrm{N},k},\ k = 1, \ldots, s\right\}$ where $I_{\mathrm{N},k} \subseteq I_k$ is the subset of actuators available in subsystem Σ_k, recoverability must be analysed with respect to the global system, because non-decomposable specifications are considered.

Let $(\boldsymbol{u},\ \mathcal{Z})$ be a pair where $\mathcal{Z}$ is an information pattern and $\boldsymbol{u}$ is a control law under $\mathcal{Z}$. The notation $\mathcal{P}\left(I_\mathrm{N},\ \boldsymbol{u},\ \mathcal{Z}\right)$ means that the pair $(\boldsymbol{u},\ \mathcal{Z})$ achieves the specification $\mathcal{P}$ when applied to the subset of actuators $I_\mathrm{N} \subseteq 2^I$.

Definition 10.4 (*Admissibility, admissibility span*) A pair $(\boldsymbol{u},\ \mathcal{Z})$ is admissible for configuration I_N, if it satisfies the specification $\mathcal{P}$, i.e. if $\mathcal{P}\left(I_\mathrm{N},\ \boldsymbol{u},\ \mathcal{Z}\right)$. The admissibility span of a pair $(\boldsymbol{u},\ \mathcal{Z})$ is the set $\mathcal{R}\left(\boldsymbol{u},\ \mathcal{Z}\right)$ of all configurations I_N for which the control law $\boldsymbol{u}$ is admissible:

$$\mathcal{R}\left(\boldsymbol{u},\ \mathcal{Z}\right) = \left\{I_\mathrm{N} \in 2^I : \mathcal{P}\left(I_\mathrm{N},\ \boldsymbol{u},\ \mathcal{Z}\right)\right\}.$$

Definition 10.5 (*Recoverability, recoverability span*) The fault I_F - equivalently the configuration I_N - is recoverable under the information pattern $\mathcal{Z}$ if there exists a control law $\boldsymbol{u}$ such that the pair $(\boldsymbol{u},\ \mathcal{Z})$ is admissible for configuration I_N. The recoverability span of the information pattern $\mathcal{Z}$ is the set $\mathcal{R}\left(\mathcal{Z}\right)$ of all configurations I_N that are recoverable under $\mathcal{Z}$:

$$\mathcal{R}\left(\mathcal{Z}\right) = \left\{I_\mathrm{N} \in 2^I : \exists\left(\boldsymbol{u},\ \mathcal{Z}\right) : \mathcal{P}\left(I_\mathrm{N},\ \boldsymbol{u},\ \mathcal{Z}\right)\right\}.$$

Note that recoverability is a structural property, since it depends only on the pair $(I_{\mathrm{N}},\ \mathcal{Z})$.

Example 10.6 Recoverability span
In Example 10.1, assume that it is desired to α-stabilise the system by decentralised control via output feedback. Under the information pattern $\mathcal{Z}_{\min} = \{y_1,\ y_2,\ y_3,\ y_4\}$, the design problem is to find the parameters k_{ij} such that the control laws

$$\begin{aligned}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} &= \begin{pmatrix} k_{11} \\ k_{21} \end{pmatrix} y_1 \\ u_3 &= k_{32} y_2 \\ u_4 &= k_{43} y_3 \\ u_5 &= k_{54} y_4\end{aligned}$$

α-stabilise the system. Remember that a system is α-stable if there exists a Lyapunov function $V = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x}$ such that $\dot{V} \leq -\alpha V$ along its trajectories, i.e.

$$\boldsymbol{Q}\,(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\boldsymbol{C}) + (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{\mathrm{T}}\,\boldsymbol{Q} + \alpha \boldsymbol{Q} \leq 0.$$

It is easy to verify that under the nominal actuator configuration $I = \{1,\ 2,\ 3,\ 4,\ 5\}$, the specification associated with $\alpha = 1$ is achieved using the control laws:

$$\begin{aligned}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} &= \begin{pmatrix} 1.2991 \\ 5.6882 \end{pmatrix} y_1 \\ u_3 &= 4.6939 y_2 \\ u_4 &= 0.1190 y_3 \\ u_5 &= 3.3712 y_4.\end{aligned} \tag{10.19}$$

Using the short notations 123 for configuration $\{1,\ 2,\ 3\}$, 2345 for configuration $\{2,\ 3,\ 4,\ 5\}$, etc., it can be easily checked that in the presence of actuator faults, configurations 2345, 1235, 1234, 1245, 245, 235, 234, 125 and 123 can still be α-stabilised using the local information pattern, but this is true neither for configurations 1345, 345, 145, 135, 134 and 124 nor for their subsets. The white nodes in Fig. 10.7 show the recoverability span associated with the local information pattern $\mathcal{Z}_{\min} = \{y_1,\ y_2,\ y_3,\ y_4\}$. For example, it can be checked that the α-stabilisation problem has a solution for configuration 245 which is

$$\begin{aligned} u_2 &= 15.6231 y_1 \\ u_4 &= 16.0533 y_3 \\ u_5 &= 5.4455 y_4,\end{aligned}$$

while it has no solution for the grey configurations.

Note that the algorithmic complexity of the determination of the set of recoverable configurations is limited by the fact that it is enough to find the minimal ones. Indeed, if a configuration I_{N} is recoverable under an information pattern $\mathcal{Z}$, then any configuration that includes I_{N} is also recoverable under $\mathcal{Z}$. In this example, there are two minimal recoverable configurations, namely 23 and 25, that are shown with a bold contour on the figure. □

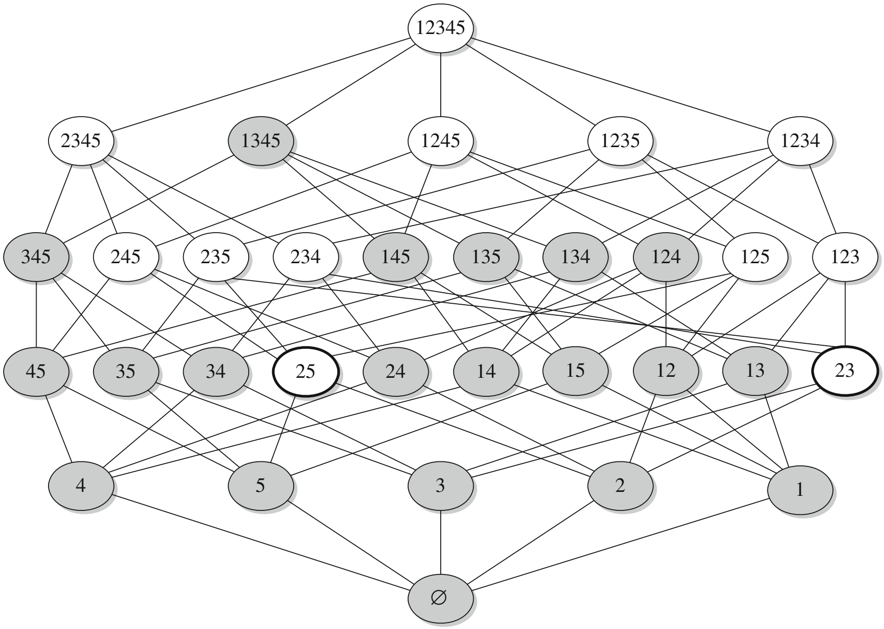

Fig. 10.7 Recoverability span under $\mathcal{Z}_{\min}$

10.5.2 Information Pattern Reconfiguration

As the recoverability of a configuration depends on the information pattern that is used, adapting the information pattern to the system situation is a means to achieve fault tolerance. Let I_{C} be the current system configuration (either nominal or the result of previous faults), and $\mathcal{Z}_{\mathrm{C}}$ be the current information pattern, such that $I_{\mathrm{C}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$, and assume a fault occurs, leading to configuration $I_{\mathrm{N}} \subset I_{\mathrm{C}}$. Then, either $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$ or $I_{\mathrm{N}} \notin \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$.

In the first case (Problem 1), one has to find a control law under $\mathcal{Z}_{\mathrm{C}}$ that is admissible for I_{N}. Such a control law indeed exists, since $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$. In the second case, no control law under $\mathcal{Z}_{\mathrm{C}}$ can achieve the specification $\mathcal{P}$. A possibility is therefore to relax the information pattern constraint by finding, if possible, an information pattern $\mathcal{Z}_{\mathrm{N}}$ such that $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{N}})$ (Problem 2). Once $\mathcal{Z}_{\mathrm{N}}$ is determined, Problem 1 is solved to find a control law that achieves the specification $\mathcal{P}$ under $\mathcal{Z}_{\mathrm{N}}$. If an information pattern $\mathcal{Z}_{\mathrm{N}}$ such that $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{N}})$ does not exist, the fault is not recoverable at all and the objective reconfiguration level must be triggered.

In the sequel, we focus on the information pattern reconfiguration problem (Problem 2), since Problem 1 is nothing but the classical fault-tolerance problem.

Ordering the set of information patterns. Let **Z** be the set of information patterns that are considered. Note that while for distributed diagnosis, local diagnosers were provided with local and possibly remote control and measurement signals, and in distributed control, each local controller has to be provided only with local and possibly remote measurement signals. The information patterns considered here are therefore simpler than in the distributed diagnosis case, being only associated with covers of J. It follows that $\mathbf{Z}$ is associated with the set of covers of J, and that the partial order relation on $\mathbf{Z}$ is also simplified as follows:

Definition 10.6 (*Order on the set of information patterns*) Let $\mathcal{Z}^+ = \{z_k^+, k = 1, \ldots, s\}$ and $\mathcal{Z}^- = \left\{z_k^-, k = 1, \ldots, s\right\}$ be two information patterns in $\mathbf{Z}$. $\mathcal{Z}^+$ is *wider* than $\mathcal{Z}^-$ ($\mathcal{Z}^+ \succeq \mathcal{Z}^-$) if

$$\forall k \in \{1, \ldots, s\}, \ z_k^- \subseteq z_k^+ .$$

Remark 10.7 Similar to the distributed diagnosis case, the full information pattern $\mathcal{Z}_{\max}$ is wider than any other, making $\mathcal{Z}_{\max}$ the maximal element of $\mathbf{Z}$. Also, there exists no information pattern in $\mathbf{Z}$ that is *narrower* than $\mathcal{Z}_{\min}$ (meaning that $\mathcal{Z}_{\min}$ would be wider than it); therefore, $\mathcal{Z}_{\min}$ is the minimal element of $\mathbf{Z}$ (to see this, consider any $\mathcal{Z}^* \preceq Z_{\min}$ and conclude that $\mathcal{Z}^* = \mathcal{Z}_{\min}$ if condition $\cup_{k=1,\ldots,s} Z_k = J$ is to be satisfied). □

A monotonicity property. The main result here is that the information pattern reconfiguration problem can be solved only for those configurations that are recoverable under the full information pattern.

Theorem 10.3 *Let $(I_{\mathrm{C}}, \mathcal{Z}_{\mathrm{C}})$ be the current system situation and assume that a fault occurs such that $I_{\mathrm{N}} \subset I_{\mathrm{C}}$. A necessary and sufficient condition for the existence of an information pattern $\mathcal{Z}_{\mathrm{N}}$ such that $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{N}})$ is that $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\max})$.*

This result is easy to understand from the fact that recoverability spans are monotonous with respect to the order $\succeq$ on $\mathcal{Z}$, i.e. one has

$$\mathcal{Z}^+ \succeq \mathcal{Z}^- \Rightarrow \mathcal{R}\left(\mathcal{Z}^-\right) \subseteq \mathcal{R}\left(\mathcal{Z}^+\right) . \tag{10.20}$$

Indeed, under the information pattern $\mathcal{Z}^+$, each local controller can use a super-set of the measurement signals available under the pattern $\mathcal{Z}^-$. Therefore, if there exists an admissible control under $\mathcal{Z}^-$, there is one under $\mathcal{Z}^+$. Now, assume $I_{\mathrm{N}} \notin \mathcal{R}(\mathcal{Z}_{\max})$, and then from Eq. (10.20), there is no $\mathcal{Z}_{\mathrm{N}} \preceq \mathcal{Z}_{\max}$ such that $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{N}})$, and the fault is therefore non-recoverable. Assume now $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\max})$, and then $\mathcal{Z}_{\mathrm{N}} = \mathcal{Z}_{\max}$ solves the problem.

Note that the result in Theorem 10.3 is true whatever the status $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$ or $I_{\mathrm{N}} \notin \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$. If $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$, one indeed has $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\max})$ but there is no need to reconfigure the information pattern $\mathcal{Z}_{\mathrm{C}}$ (note that this does not mean that the control laws should not be reconfigured, but only that the data they use do not need to be changed!). If $I_{\mathrm{N}} \notin \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$ and $I_{\mathrm{N}} \notin \mathcal{R}(\mathcal{Z}_{\max})$, the fault is not recoverable, and objective reconfiguration has to take place. We now consider the case $I_{\mathrm{N}} \notin \mathcal{R}(\mathcal{Z}_{\mathrm{C}})$ (the fault is not recoverable under the current information pattern) but $I_{\mathrm{N}} \in \mathcal{R}(\mathcal{Z}_{\max})$ (the fault is recoverable under the full information pattern).

Application to fault-tolerant distributed control. Let $(I_\mathrm{C},\ \mathcal{Z}_\mathrm{C})$ be the current system situation and consider a fault that is not recoverable under the current information pattern, but is recoverable under the full information pattern. Solving the information pattern reconfiguration problem is equivalent to determining the set:

$$\mathcal{Z}\,(I_\mathrm{N}) = \{\mathcal{Z} \in \mathbf{Z} :\ I_\mathrm{N} \in \mathcal{R}\,(\mathcal{Z})\}\,.$$

From an algorithmic point of view, testing every $\mathcal{Z} \in \mathbf{Z}$ for the possibility to find a control $\boldsymbol{u}$ such that $(\boldsymbol{u},\ \mathcal{Z})$ is admissible for I_N is a huge problem. Indeed, from Remark 10.7, one has

$$\mathcal{Z} \in \mathbf{Z} \Rightarrow \mathcal{Z}_\mathrm{min} \preceq \mathcal{Z} \preceq \mathcal{Z}_\mathrm{max}$$

and since $\mathcal{Z}_\mathrm{min} = \left\{\boldsymbol{y}_k,\ k = 1, \ldots, s\right\}$ and $\mathcal{Z}_\mathrm{max} = \{\boldsymbol{y},\ k = 1, \ldots, s\}$, it follows that for any information pattern $\mathcal{Z} = \{\boldsymbol{z}_k,\ k = 1, \ldots, s\}$ one has $\boldsymbol{z}_k = \boldsymbol{y}_k \cup \gamma_k$, where $\gamma_k \subseteq \boldsymbol{y}\backslash\boldsymbol{y}_k$ is the subset of measurements that are "added" to the ones already available to subsystem Σ_k in the local information pattern $\mathcal{Z}_\mathrm{min}$. It follows that

$$\mathbf{Z} = \prod_{k=1,\ldots,s} 2^{\boldsymbol{y}\backslash\boldsymbol{y}_k}.$$

Example 10.7 The number of candidate information patterns
Assume that four subsystems have the local measurement vectors $y_1 \in |\mathcal{R}$, $\boldsymbol{y}_2 \in |\mathcal{R}^2$, $y_3 \in |\mathcal{R}$ and $\boldsymbol{y}_4 \in |\mathcal{R}^2$. Then, in addition to y_1, Σ_1 could receive any subset of the other five measurements, Σ_2 could receive any subset of the other four in addition to $\boldsymbol{y}_2$, etc. This gives a total of $2^5 \times 2^4 \times 2^5 \times 2^4$ information patterns that are wider than $\mathcal{Z}_\mathrm{min}$. □

Reducing the set of candidate information patterns. In this section, we consider simple arguments that allow to reduce the number of information patterns to be explored in order to determine $\mathcal{Z}\,(I_\mathrm{N})$.

Theorem 10.4 *Let $(I_\mathrm{C},\ \mathcal{Z}_\mathrm{C})$ be the current system situation and assume that a fault occurs such that $I_\mathrm{N} \notin \mathcal{R}\,(\mathcal{Z}_\mathrm{C})$ but $I_\mathrm{N} \in \mathcal{R}\,(\mathcal{Z}_\mathrm{max})$. Then, it holds that*

$$\mathcal{Z}\,(I_\mathrm{N}) \subseteq \mathcal{Z}\backslash\mathcal{N}\,(\mathcal{Z}_\mathrm{C}),$$

where $\mathcal{N}\,(\mathcal{Z}_\mathrm{C}) = \left\{\mathcal{Z}_\mathrm{C}^- : \mathcal{Z}_\mathrm{C}^- \preceq \mathcal{Z}_\mathrm{C}\right\}$ is the set of information patterns that are narrower than $\mathcal{Z}_\mathrm{C}$ with respect to the order relation $\preceq$. □

Indeed, noting that $I_\mathrm{N} \notin \mathcal{R}(\mathcal{Z}_\mathrm{C}) \Rightarrow \mathcal{Z}_\mathrm{C} \notin \mathcal{Z}\,(I_\mathrm{N})$, the result follows from Eq. (10.20) which implies

$$\mathcal{Z}_\mathrm{C}^- \preceq \mathcal{Z}_\mathrm{C} \iff \mathcal{Z}_\mathrm{C}^- \notin \mathcal{Z}\,(I_\mathrm{N}).$$

Furthermore, in order to determine all the elements of $\mathcal{Z}\,(I_\mathrm{N})$, it is enough to find its minimal ones. Remember that a minimal information pattern $\mathcal{Z}_\mathrm{min}$ in $\mathcal{Z}\,(I_\mathrm{N})$ is such that

$$\mathcal{Z}_{\min} \in \mathcal{Z}(I_\mathrm{N})$$
$$\mathcal{Z} \preceq \mathcal{Z}_{\min} \Rightarrow \mathcal{Z} \notin \mathcal{Z}(I_\mathrm{N}).$$

Theorem 10.5 *Let $\mathcal{Z}_\mathrm{M}(I_\mathrm{N})$ be the set of minimal elements of $\mathcal{Z}(I_\mathrm{N})$. Then one has*

$$\mathcal{Z}(I_\mathrm{N}) = \bigcup_{\mathcal{Z} \in \mathcal{Z}_\mathrm{M}(I_\mathrm{N})} \mathcal{W}(\mathcal{Z}),$$

where $\mathcal{W}(\mathcal{Z}) = \left\{\mathcal{Z}^+ : \mathcal{Z} \preceq \mathcal{Z}^+\right\}$ is the set of information patterns that are wider than $\mathcal{Z}$ with respect to the order relation $\preceq$.

Indeed, it is clear that $\mathcal{Z}_{\min} \in \mathcal{Z}_\mathrm{M}(I_\mathrm{N}) \Rightarrow \mathcal{Z}_{\min} \in \mathcal{Z}(I_\mathrm{N})$. Then, the implication

$$\mathcal{Z} \in \mathcal{Z}(I_\mathrm{N}) \backslash \mathcal{Z}_\mathrm{M}(I_\mathrm{N}) \Rightarrow \exists \mathcal{Z}_{\min} \in \mathcal{Z}_\mathrm{M}(I_\mathrm{N}) : \mathcal{Z} \in \mathcal{W}(\mathcal{Z}_{\min})$$

is true, because $\mathcal{Z}$ being not minimal, and there exists an information pattern $\mathcal{Z}_1$ such that $\mathcal{Z}_1 \in \mathcal{Z}(I_\mathrm{N})$ and $\mathcal{Z} \in \mathcal{W}(\mathcal{Z}_1)$. If $\mathcal{Z}_1 \in \mathcal{Z}_\mathrm{M}(I_\mathrm{N})$, the conclusion of the theorem is obtained. If not, which means $\mathcal{Z}_1$ is not minimal, there exists an information pattern $\mathcal{Z}_2$ such that $\mathcal{Z}_2 \in \mathcal{Z}(I_\mathrm{N})$ and $\mathcal{Z}_1 \in \mathcal{W}(\mathcal{Z}_2)$, which implies $\mathcal{Z} \in \mathcal{W}(\mathcal{Z}_2)$ by transitivity. If $\mathcal{Z}_2 \in \mathcal{Z}_\mathrm{M}(I_\mathrm{N})$, the conclusion is obtained; otherwise, the process is repeated until the conclusion holds, which must eventually occur because $\mathcal{Z}$ contains a finite number of information patterns. Finally, the monotonicity property (10.20) implies that

$$\mathcal{Z} \in \bigcup_{\mathcal{Z} \in \mathcal{Z}_\mathrm{M}(I_\mathrm{N})} \mathcal{W}(\mathcal{Z}) \Rightarrow \mathcal{Z} \in \mathcal{Z}(I_\mathrm{N}).$$

Remark 10.8 A subset of $\mathcal{Z}(I_\mathrm{N})$ is found by exploring only a subset of $\mathcal{Z} \backslash \mathcal{N}(\mathcal{Z}_\mathrm{C})$, provided it contains at least one admissible information pattern. This obviously happens with $\mathcal{W}(\mathcal{Z}_\mathrm{C})$ since one has $I_\mathrm{N} \in \mathcal{R}(\mathcal{Z}_{\max})$. In the sequel, we look for solutions within $\mathcal{W}(\mathcal{Z}_\mathrm{C})$ because information patterns that are wider than $\mathcal{Z}_\mathrm{C}$ are easy to construct, especially if technological constraints associated with the communication system are taken into account. The next sections, respectively, consider the publisher/subscriber and the bilateral agreements schemes. □

10.5.3 Publisher/Subscriber Scheme

Optimal subscriptions. Before we address the construction of wider information patterns in the publisher/subscriber scheme, let us first remark that since the communication cost is associated with the published variables, the best use of the published variables is achieved when the local controllers subscribe to all of them. Indeed, let $\mathcal{Z}_\mathrm{C} = \left\{z_{\mathrm{C,k}},\ k = 1, \ldots, s\right\}$ be the current information pattern, and let γ_C and $\gamma_{\mathrm{C,k}}$ be, respectively, the current set of published variables, and the current set of variables subscribed by subsystem Σ_k (meaning that $z_{\mathrm{C,k}} = \boldsymbol{y}_k \cup \gamma_{\mathrm{C,k}}$). One has $\gamma_{\mathrm{C,k}} \subseteq \gamma_{\mathrm{C}}$, $k = 1, \ldots, s$ but from Eq. (10.20) it follows that for the cost of publishing the variables γ_{C}, the largest sets of recoverable configurations are obtained with the subscriptions $\gamma_{\mathrm{C,k}} = \gamma_{\mathrm{C}}, k = 1, \ldots, s$. However, it is worth to remark that in order to recover a given configuration, there is no obligation for all subsystems to subscribe to all the published variables γ_{C}.

The set of wider information patterns. As opposed to distributed diagnosis, only subsets of measurements are published for distributed fault-tolerant control. Therefore, in the publisher/subscriber scheme, any information pattern wider than $\mathcal{Z}_{\mathrm{C}}$ is associated with the publication of a subset of variables in $\boldsymbol{y}\backslash\gamma_{\mathrm{C}}$. It follows that $\mathcal{W}(\mathcal{Z}_{\mathrm{C}})$ is the lattice $2^{\boldsymbol{y}\backslash\gamma_{\mathrm{C}}}$. Note that the minimal elements in $\mathcal{W}(\mathcal{Z}_{\mathrm{C}})$ are the first ones found when exploring $2^{\boldsymbol{y}\backslash\gamma_{\mathrm{C}}}$ by increasing levels (indeed they are associated with the minimal sets of measurements to be published in addition to those already present in $\mathcal{Z}_{\mathrm{C}}$).

Example 10.8 Information pattern reconfiguration in the publisher/subscriber scheme

Let y_1, y_2, y_3, y_4 be the local measurements associated with a system composed of four subsystems. In the information pattern $\mathcal{Z}_{\mathrm{C}} = \{(y_1, y_2), y_2, (y_1, y_3), (y_2, y_4)\}$, the published variables are $\{y_1, y_2\}$. However, $\mathcal{Z} = \{(y_1, y_2), (y_1, y_2), (y_1, y_2, y_3), (y_1, y_2, y_4)\}$ is wider and has the same communication cost. Assume $\mathcal{Z}_{\mathrm{C}}$ is the current information pattern and a fault that is not recoverable under $\mathcal{Z}_{\mathrm{C}}$, but is recoverable under $\mathcal{Z}_{\max}$ occurs. Then, a subset of solutions to the information pattern reconfiguration problem is generated by considering

$$\mathcal{W}(\mathcal{Z}_{\mathrm{C}}) = \left\{\mathcal{Z}(\gamma) : k = 1, \ldots, 4,\ z_k = z_{\mathrm{C,k}} \cup \gamma, \gamma \subseteq y_3 \cup y_4\right\}.$$

Note that any information pattern in $\mathcal{W}(\mathcal{Z}_{\mathrm{C}})$ is obtained by publishing one subset of $\{y_3, y_4\}$ in addition to the variables already published in $\mathcal{Z}_{\mathrm{C}}$. Note also that $\mathcal{W}(\mathcal{Z}_{\mathrm{C}})$ is a subset of $\mathcal{W}(\mathcal{Z}_{\min})$ because any $\mathcal{Z}_{\mathrm{C}} \in \mathcal{Z}$ is wider than $\mathcal{Z}_{\min}$. Finally, note that $\mathcal{W}(\mathcal{Z}_{\min})$ can be determined off-line, since the set of publishable data is nothing but the lattice of the sensors subsets. This lattice is displayed in Fig. 10.8, where $\emptyset$ corresponds to $\mathcal{Z}_{\min}$ (no data are published), while 1234, which stands for $\{y_1, y_2, y_3, y_4\}$, is associated with $\mathcal{Z}_{\max}$ (all sensor outputs are published). Since $\{y_1, y_2\}$ are currently published in $\mathcal{Z}_{\mathrm{C}}$ (so $\mathcal{Z}_{\mathrm{C}}$ is represented by node 12), $\mathcal{W}(\mathcal{Z}_{\mathrm{C}})$ is the sub-lattice with grey nodes. □

10.5.4 Bilateral Communication Scheme

The set of wider information patterns. Let $\mathcal{A}_{\mathrm{C}}$ characterise the current set of agreements, leading to the equivalence classes $\mathcal{E}(\mathcal{A}_{\mathrm{C}}) = \left\{E_{\mathrm{C},l},\ l = 1, \ldots, \sigma\right\}$, and the current information pattern $\mathcal{Z}_{\mathrm{C}} = \left\{z_{\mathrm{C,k}},\ k = 1, \ldots, s\right\}$. A wider information pattern can only be obtained by establishing a set of agreements whose graph $\mathcal{A}_{\mathrm{W}}$ is such that $\mathcal{A}_{\mathrm{C}} \subset \mathcal{A}_{\mathrm{W}}$. This results in the equivalence classes $\mathcal{E}(\mathcal{A}_{\mathrm{W}}) = \left\{E_{\mathrm{W,k}},\ k = 1, \ldots, \rho\right\}$ where $\rho < \sigma$ such that each class of $\mathcal{E}(\mathcal{A}_{\mathrm{W}})$ is equal to one class of $\mathcal{E}(\mathcal{A}_{\mathrm{C}})$, or is the

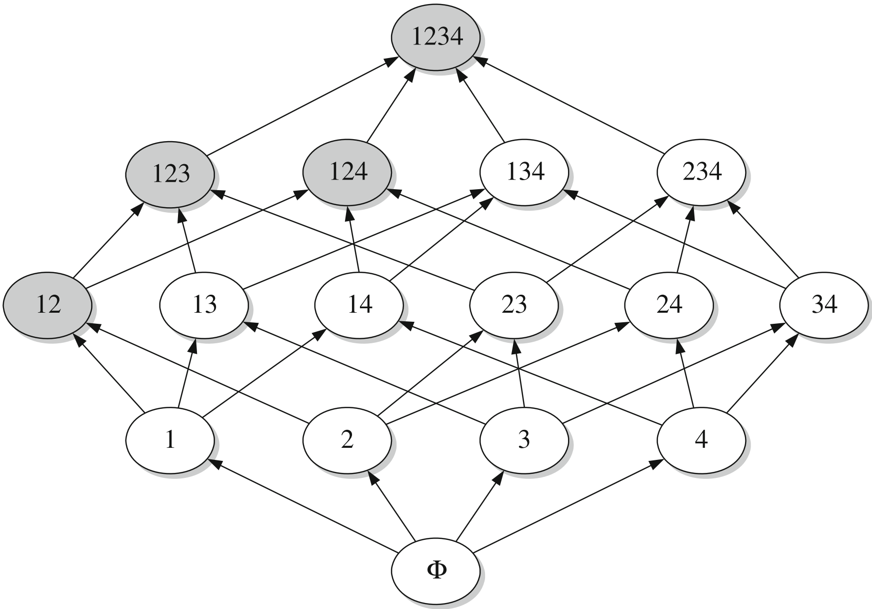

Fig. 10.8 Publishable sets of data

union of several classes of $\mathcal{E}(\mathcal{A}_\mathrm{C})$. The set $\mathcal{W}(\mathcal{Z}_\mathrm{C})$ of information patterns wider than $\mathcal{Z}_\mathrm{C}$ is then

$$\mathcal{W}(\mathcal{Z}_\mathrm{C}) = \\ \left\{\mathcal{Z}_\mathrm{W} = \left\{z_{\mathrm{W,i}},\ i = 1, \ldots, s : \Sigma_i \in E_{\mathrm{W,k}} \longrightarrow z_{\mathrm{W,i}} = z\left(E_{\mathrm{W,k}}\right), \forall \mathcal{A}_\mathrm{W} \supset \mathcal{A}_\mathrm{C}\right\}\right\}.$$

Hierarchical decomposition. In order to construct $\mathcal{W}(\mathcal{Z}_\mathrm{C})$, remark that sets of agreements and system decompositions are related. Indeed, given a set of agreements $\mathcal{A}$ and the equivalence classes $\mathcal{E}(\mathcal{A}) = \{E_\mathrm{l},\ l = 1, \ldots, \sigma\}$, all the subsystems in class E_l share the same information, and therefore they constitute one (high-level) subsystem $\Sigma(E_\mathrm{l}), l = 1, \ldots, \sigma$ whose state $\boldsymbol{x}(E_\mathrm{l})$ and control $\boldsymbol{u}(E_\mathrm{l})$ are the concatenation of the local states $\{\boldsymbol{x}_i,\ \Sigma_i \in E_\mathrm{l}\}$ and controls $\{\boldsymbol{u}_i,\ \Sigma_i \in E_\mathrm{l}\}$. Since $\mathcal{E}(\mathcal{A})$ is a partition of Σ into $\sigma \leq s$ classes, the decomposition of Σ defined by $\{\Sigma(E_\mathrm{l}), l = 1, \ldots, \sigma\}$ is *coarser* than $\{\Sigma_i,\ i = 1, \ldots, s\}$, meaning that every subsystem Σ_i is included in one and only one $\Sigma(E_\mathrm{l})$.

It follows that there is a one to one correspondence between the set of all bilateral agreements and a hierarchy $\mathcal{H}$ of decompositions of Σ.

Definition 10.7 (*Hierarchy of decompositions*) A hierarchy $\mathcal{H}$ is a set of decompositions of Σ organised into levels $\mathcal{H}_\sigma$ that contain decompositions into σ subsystems. Level $\mathcal{H}_1$ is the overall system, while level $\mathcal{H}_s$ is $\{\Sigma_i,\ i = 1, \ldots, s\}$, called the *atomic* decomposition. Two decompositions $\mathcal{E}_0$ and $\mathcal{E}_1$ that belong to two adjacent levels $\mathcal{H}_\sigma$

and $\mathcal{H}_{\sigma-1}$ contain the same subsystems, except for one subsystem in $\mathcal{E}_1$ that is the union of two subsystems in $\mathcal{E}_0$.

Figure 10.9 illustrates two possible hierarchies for the decomposition of a system with four subsystems. Each subsystem is represented by its index, for example, 1 stands for Σ_1 while 34 represents the union of the two subsystems $\Sigma_3 \cup \Sigma_4$ and 1234 stands for the overall system $\cup_{i=1,\ldots,4}\Sigma_i$.

Based on the correspondence between bilateral agreements and decomposition hierarchies, it follows that the minimal elements of $\mathcal{W}(\mathcal{Z}_C)$ are associated with the first decompositions found when exploring the hierarchy by decreasing levels (indeed, they concern the minimum sets of variables shared between subsystems in addition to the variables already shared in the previous step).

Example 10.9 Information pattern reconfiguration in the bilateral agreements scheme

Figure 10.10 displays the hierarchy associated with all possible information patterns under bilateral agreements, for the four subsystems' example. The atomic decomposition $\{\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4\}$ is abbreviated as 1, 2, 3, 4, while 1234 stands for the overall system $\cup_{i=1,\ldots,4}\Sigma_i$, and 14, 23 represent the decomposition into two subsystems $\Sigma_1 \cup \Sigma_4$ and $\Sigma_2 \cup \Sigma_3$. Assuming the current information pattern is $\mathcal{Z}_C = \{(y_1, y_3), y_2, (y_1, y_3), y_4\}$ (represented by the nodes $13,\ 2,\ 4$ with a bold contour in Fig. 10.10), the white sub-lattice shows the set $\mathcal{W}(\mathcal{Z}_C)$ under the bilateral agreements communication scheme. The minimal elements are (123, 4), (134, 2) and (13, 24). The figure also shows the three hierarchies associated with the three paths between the node (13, 2, 4) associated with the current information pattern and the node (1234) associated with the maximal information pattern. □

Example 10.10 Information pattern reconfiguration

Figure 10.7 displays the non-recoverable configurations under the local information pattern associated with the decentralised control of the system in Example 10.1. Applying Theorem 10.3, it can be checked that some configurations that are non-recoverable under the local information pattern become recoverable by an information pattern extension.

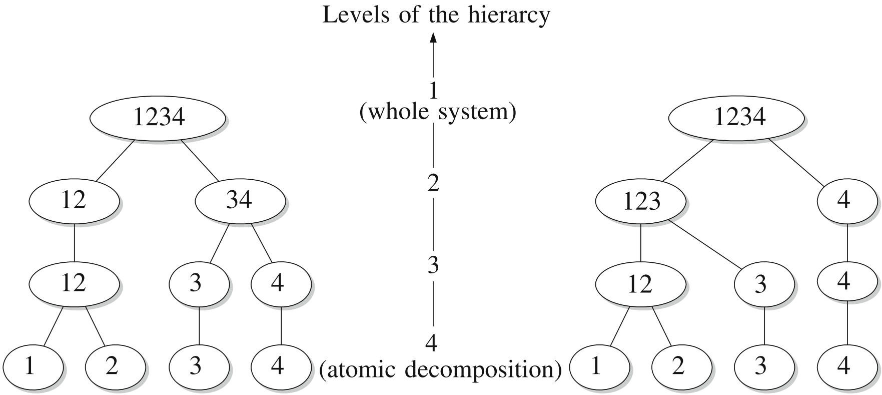

Fig. 10.9 Two possible decomposition hierarchies

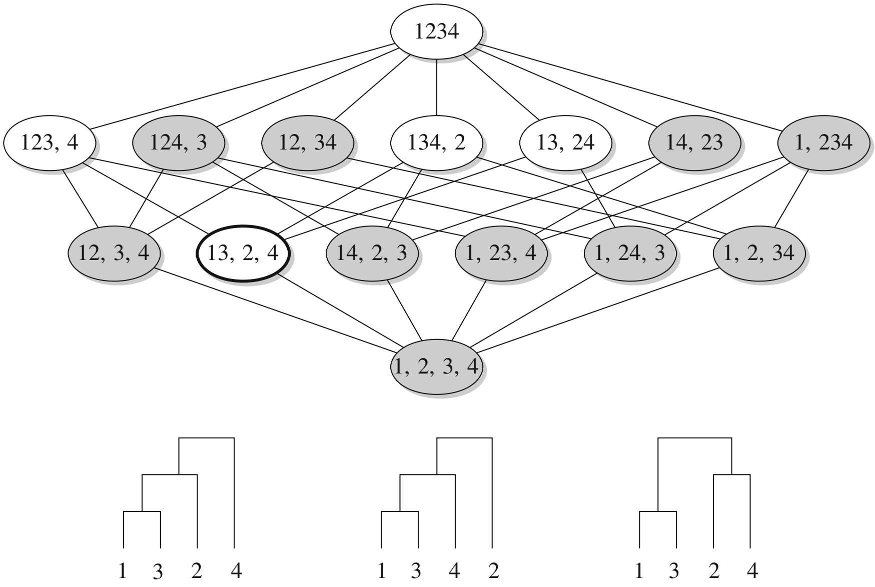

Fig. 10.10 Information patterns under bilateral agreements for four subsystems

Let us focus, for example, on configuration 1345 which is recoverable under a reconfigured information pattern. The publication of y_2 leads to the reconfigured information pattern $\mathcal{Z}_{1345} = \{(y_1, y_2), y_2, (y_2, y_3), (y_2, y_4)\}$ under which there exists a reconfigured distributed control of the form:

$$\begin{aligned} u_1 &= k_{11}y_1 + k_{12}y_2 \\ u_3 &= k_{32}y_2 \\ u_4 &= k_{42}y_2 + k_{43}y_3 \\ u_5 &= k_{52}y_2 + k_{54}y_4. \end{aligned} \tag{10.21}$$

It can indeed be checked that the reconfigured control laws,

$$\begin{aligned} u_1 &= 9.6660y_1 + 8.3264y_2 \\ u_3 &= 0.6995y_2 \\ u_4 &= 2.9872y_2 + 13.8406y_3 \\ u_5 &= 1.7713y_2 + 2.0056y_4, \end{aligned}$$

allows the system to be recovered after the fault. However, it is interesting to remark that the solution of Eq. (10.21) is not unique. Indeed, Eq. (10.22) exhibits another solution, such that k_{42} and k_{52} are both equal to zero, meaning that Σ_2, Σ_3 and Σ_4 still work in a decentralised way. This illustrates the fact that while $\mathcal{Z}_{1345}$ is the most efficient information pattern associated with the publication of y_2 as noted in the presentation of the publisher/subscriber scheme (Optimal subscriptions), solutions based on narrower information patterns might also exist. Clearly, the publication of y_2 is necessary for configuration 1345 to become recoverable, but that does not mean that all subsystems have to subscribe to the newly published variable y_2:

$$\begin{aligned} u_1 &= 9.8167y_1 + 8.800y_2 \\ u_3 &= 0.7470y_2 \\ u_4 &= 9.2974y_3 \\ u_5 &= 9.1418y_4. \end{aligned} \tag{10.22}$$

On another hand, should the bilateral communication scheme be of interest, the hierarchy of Fig. 10.10 suggests the reconfigured information pattern $\mathcal{Z}_{1345} = \{(y_1, y_2), (y_1, y_2), y_3, y_4\}$ which results in an admissible controller:

$$\begin{aligned} u_1 &= 9.5717y_1 + 8.0028y_2 \\ u_3 &= 4.8538y_1 + 1.4189y_2 \\ u_4 &= 8.4352y_3 \\ u_5 &= 18.3147y_4. \square \end{aligned} \tag{10.23}$$

10.5.5 Extensions

Several extensions of the information pattern reconfiguration frame can be considered. The simplest one addresses sensor (or general system component) faults, since only actuator faults (hence actuator configurations) have been considered up to now. Other extensions address optimality issues in the information pattern reconfiguration process, namely the selection of an optimal information pattern reconfiguration from the point of view of the communication cost and the more complex issue of minimising the reconfiguration effort.

Sensor faults. Only actuator faults have been considered up to now, for the sake of simplicity. It is easy to see that sensor faults (more generally system components faults) can easily be dealt with in the reconfiguration strategy. Indeed, assume there is a set J_{F} of faulty sensors, then the available sensors are $J_{\mathrm{N}} = J \backslash J_{\mathrm{F}}$ (note that faults in the communication system that prevent the measurements of some sensors to be transmitted to the controllers that need them are also represented by this model). The pre-fault information pattern $\mathcal{Z} = \{z_k,\ k = 1, \ldots, s\}$ becomes the post-fault information pattern $\mathcal{Z}_{\mathrm{N}} = \left\{z_k \cap \boldsymbol{y}_N,\ k = 1, \ldots, s\right\}$, and $\mathcal{Z}_{\max} = \{\boldsymbol{y},\ k = 1, \ldots, s\}$ becomes $\mathcal{Z}_{\mathrm{N}\max} = \left\{\boldsymbol{y}_{\mathrm{N}},\ k = 1, \ldots, s\right\}$. The fault is recoverable if and only if the current actuator configuration I_{N} and the current sensor configuration J_{N} are such that $I_{\mathrm{N}} \in \mathcal{R}\left(\mathcal{Z}_{\mathrm{N}\max}\right)$.

Minimal communication cost. Since each $\mathcal{Z} \in \mathcal{Z}\left(I_{\mathrm{N}}\right)$ is associated with the communication cost $com\left(\mathcal{Z}, \mathcal{J}\right)$, selecting the information pattern $\mathcal{Z}^*$ such that

$$\mathcal{Z}^* = \arg \min_{\mathcal{Z} \in \mathcal{Z}(I_{\mathrm{N}})} com\left(\mathcal{Z}, \mathcal{J}\right) \tag{10.24}$$

provides an optimally reconfigured information pattern with respect to the communication cost. It is easily proved, from the monotonicity of the cost function $com\left(\mathcal{Z}, \mathcal{J}\right)$, that the solutions of Eq. (10.24) belong to the set $\mathcal{Z}_{\mathrm{M}}\left(I_{\mathrm{N}}\right)$ of the minimal elements of $\mathcal{Z}\left(I_{\mathrm{N}}\right)$. Following Remark 10.8, sub-optimal solutions are easily obtained from

$$\mathcal{Z}^*_{\text{sub}} = \arg \min_{\mathcal{Z} \in \mathcal{W}(\mathcal{Z}_{\text{C}})} com\,(\mathcal{Z}, \mathcal{J}) \tag{10.25}$$

once the set $\mathcal{W}\,(\mathcal{Z}_{\text{C}})$ has been determined.

10.5.6 Minimal Reconfiguration Effort

Let us consider again the case where the system is operating with the current subset of actuators $I_{\text{C}} \subseteq I$ and the current information pattern $\mathcal{Z}_{\text{C}}$, and a fault occurs such that the post-fault configuration $I_{\text{N}} \subset I_{\text{C}}$ no longer belongs to $\mathcal{R}\,(\mathcal{Z}_{\text{C}})$. Theorem 10.3 gives a necessary and sufficient condition for the existence of a solution to the information pattern reconfiguration problem, namely "Is there an information pattern $\mathcal{Z}_{\text{N}}$ such that $I_{\text{N}} \in \mathcal{R}\,(\mathcal{Z}_{\text{N}})$?" When solutions exist, Theorem 10.5 and Remark 10.8 provide some practical tools to find such information patterns, whose communication cost can be minimised by solving the problem (10.24) (or (10.25)). However, these results do not provide any characterisation of the *number of subsystems* whose data or control laws have to be reconfigured, a number that clearly characterises the reconfiguration effort. In order to address this point, we will now consider constraints on the possible reconfigured information patterns or the possible reconfigured control laws.

Σ_{K}-recoverability. We first start with the notion of Σ_{K}-recoverability, which addresses the number of subsystems whose available data have to be reconfigured after the occurrence of a fault.

Definition 10.8 (Σ_{K}*-recoverability*) Let $\Sigma_{\text{K}} = \{\Sigma_k, k \in K \subseteq \{1, \ldots, s\}\}$ be a subset of subsystems. A configuration I_{N} is Σ_{K}-recoverable if it is recoverable by reconfiguring only the data available to the subsystems in Σ_{K}.

Theorem 10.6 *Let $(I_{\text{C}}, \mathcal{Z}_{\text{C}})$ be the current system configuration, and assume a fault occurs such that the resulting configuration is $I_{\text{N}} \subset I_{\text{C}}$. A necessary and sufficient condition for configuration I_{N} to be Σ_{K}-recoverable is that $I_{\text{N}} \in \mathcal{R}\left(\mathcal{Z}_{\text{K,max}}\right)$ where $\mathcal{Z}_{\text{K,max}} = \mathcal{Z}_{\text{C}}$ except for $\boldsymbol{z}_k = \boldsymbol{y}$, $k \in K$.*

The idea of this theorem is quite similar to the idea of Theorem 10.3.

Comments.

1. Taking $K = \{1, \ldots, s\}$, i.e. accepting the possibility for the data of all the subsystems to be reconfigured, is just the problem addressed by Theorem 10.3.
2. The set of all possible subsets Σ_{K} is the lattice $2^{\{1,\ldots,s\}}$, which implies the monotonicity property that if I_{N} is Σ_{L}-recoverable, and $L \subseteq K$, then I_{N} is Σ_{K}-recoverable. It follows that the minimal subsets Σ_{L} such that I_{N} is Σ_{L}-recoverable, defined by

$$I_{\mathrm{N}} \in \mathcal{R}\left(\mathcal{Z}_{\mathrm{L,max}}\right)$$
$$\forall K \subset L, I_{\mathrm{N}} \notin \mathcal{R}\left(\mathcal{Z}_{\mathrm{K,max}}\right)$$

can be found using a classical bottom-up algorithm on the lattice $2^{\{1,\ldots,s\}}$.

Example 10.11 ***Σ_{K}*-recoverability**
It has been seen that configuration 1345 is not recoverable under the decentralised information pattern $\mathcal{Z}_{\min} = \{y_1, y_2, y_3, y_4\}$ but becomes recoverable when it is reconfigured as $\mathcal{Z}_{1345} = \{(y_1, y_2), y_2, y_3, y_4\}$.

Figure 10.11 shows all the subsets Σ_{K} such that configuration 1345 is Σ_{K} recoverable.

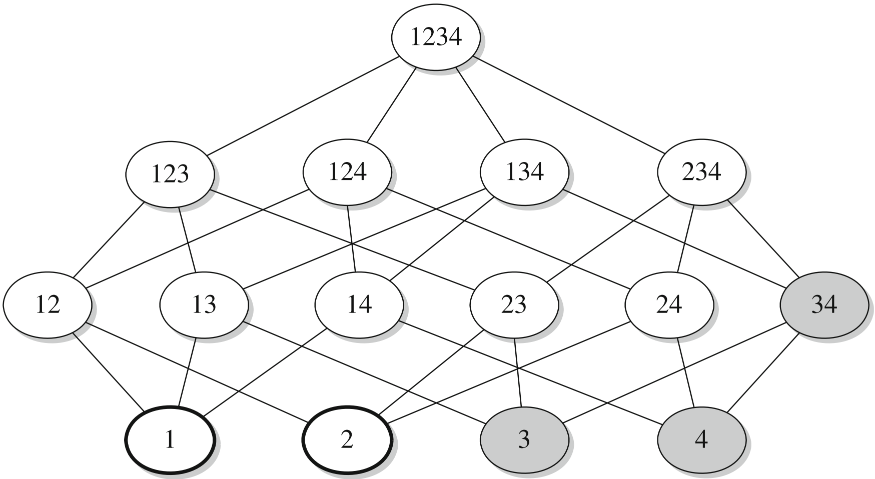

Fig. 10.11 Σ_{K}-recoverability of configuration 1345

Remark 10.9 The Σ_{K}-recoverability of a configuration I_{N} means that I_{N} becomes recoverable if the system information pattern is reconfigured so that only the data available to the subsystems in Σ_{K} are reconfigured (and so are the corresponding control laws). Although their available data remain unchanged, note that the control parameters of the subsystems that do not belong to Σ_{K} are allowed to change. A stronger version of the minimal reconfiguration effort problem is set by constraining the controls of those subsystems that do not belong to Σ_{K} to remain unchanged. In order to solve this problem, strong Σ_{K}-recoverability is now defined. □

Strong $\boldsymbol{\Sigma}_{\mathbf{K}}$-recoverability. The following notion is defined for a more detailed analysis:

Definition 10.9 (*Strong Σ_{K}-recoverability*) Let $\Sigma_{\mathrm{K}} = \{\Sigma_k, k \in K \subseteq \{1, \ldots, s\}\}$ be a subset of subsystems and let $\boldsymbol{u}$ be decomposed into $(\boldsymbol{u}_{\mathrm{K}}, \overline{\boldsymbol{u}}_{\mathrm{K}})$ where $\boldsymbol{u}_{\mathrm{K}}$ gathers the controls of those subsystems that belong to Σ_{K} while $\overline{\boldsymbol{u}}_{\mathrm{K}}$ gathers the controls of the other ones. A configuration I_N is strongly Σ_K-recoverable if it is recoverable by reconfiguring only the data available to the subsystems in Σ_K and the control laws in $\boldsymbol{u}_K$.

Theorem 10.7 *Let $\Sigma_K = \{\Sigma_k, k \in K \subseteq \{1, \dots, s\}\}$ be a subset of subsystems and let $\boldsymbol{u}$ be decomposed into $(\boldsymbol{u}_K, \overline{\boldsymbol{u}}_K)$. Let $(I_C, \mathcal{Z}_C)$ be the current system situation, and assume a fault occurs which results in configuration $I_N \subset I_C$. A necessary and sufficient condition for configuration I_N to be strongly Σ_K-recoverable is that there exists a control law $\boldsymbol{v}_K$ such that $(\boldsymbol{v}_K, \overline{\boldsymbol{u}}_K)$ is admissible under $\mathcal{Z}_{K,\max}$ where $\mathcal{Z}_{K,\max} = \mathcal{Z}_C$ except for $Z_k = J, \forall k \in K$.*

Example 10.12 Strong Σ_K-recoverability
Let us consider again configuration 1345, which is not recoverable under the decentralised information pattern $\mathcal{Z}_{\min} = \{y_1, y_2, y_3, y_4\}$ but becomes recoverable when it is reconfigured as $\mathcal{Z}_{1345} = \{(y_1, y_2), y_2, y_3, y_4\}$. By comparing Eqs. (10.19) and (10.22), it is seen that although the data they use were unchanged, subsystems Σ_2, Σ_3 and Σ_4 did reconfigure the parameters of their control laws from $u_3 = 4.6939y_2, u_4 = 0.1190y_3, u_5 = 3.3712y_4$ to $u_3 = 0.7470y_2, u_4 = 9.2974y_3, u_5 = 9.1418y_4$. Unfortunately, there exists no solution to the strong Σ_1-recoverability problem: it is indeed impossible to recover configuration 1345 by changing only the data and the control law of subsystem Σ_1. Figure 10.12 displays the result obtained when applying Theorem 10.7: the white nodes are those subsets of subsystems with respect to which actuator configuration 1345 is strongly recoverable. In other words, it is possible to recover configuration 1345 by reconfiguring only the data sets and control laws of those subsystems. Note that if a configuration is strongly Σ_K-recoverable, then it is also strongly Σ_L-recoverable for any subset of subsystems Σ_L that includes Σ_K. In the figure, the minimal subsets of subsystems such that 1345 is strongly recoverable are shown with a bold contour. These subsystems are associated with the minimal reconfiguration effort to recover the configuration of interest, namely the minimal number of data and control laws to be reconfigured for its recovery to be possible. In this example, it is seen that configuration 1345 is strongly Σ_2-recoverable. It can indeed be checked that the control laws,

$$\begin{aligned} u_1 &= 1.2991y_1 \\ u_3 &= 9.3399y_1 + 6.7874y_2 + 7.5774y_3 + 8.4913y_4 \\ u_4 &= 0.1190y_3 \\ u_5 &= 3.3712y_4 \end{aligned}$$

where u_1, u_4 and u_5 are unchanged from the nominal decentralised case, satisfy the α-stability specification.

10.6 Exercises

Exercise 10.1 Diagnosis of the two-tank system
In this exercise, we develop the complete diagnosis scheme of the two-tank system in Chap. 2, where two level sensors h_{1m} and h_{2m} were implemented in addition to the flow sensor q_m. The set of constraints and unknown variables are the following:

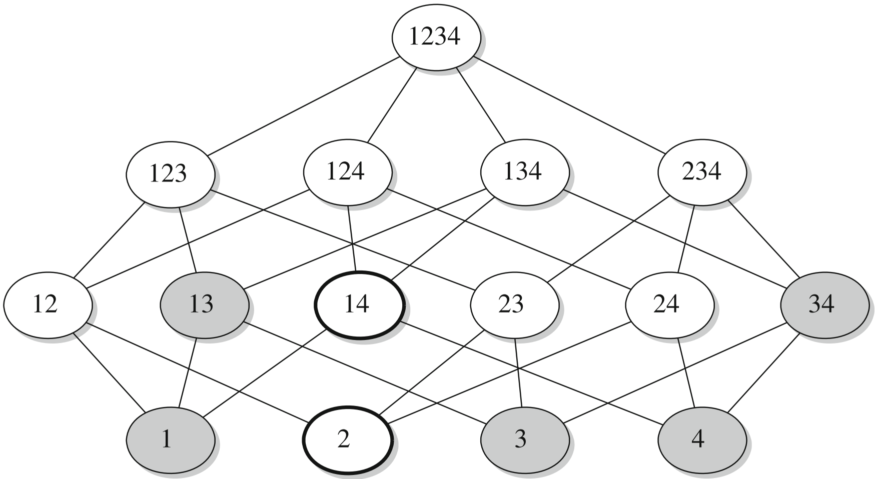

Fig. 10.12 Strong Σ_{K}-recoverability of configuration 1345

$$\begin{aligned} \boldsymbol{f} \cup \boldsymbol{g} &= \{c_1, c_2, c_3, d_4, c_5, c_6, d_7, c_8, c_{\mathrm{m}}, c_{h1}, c_{h2}\} \\ \mathcal{X} &= \left\{q_L, q_P, h_1, \dot{h}_1, h_2, \dot{h}_2, q_2, q_{12}\right\}. \end{aligned}$$

The correspondence with the model in Chap. 2 is as follows: c_1 is Eq. (2.7), c_2 is Eq. (2.6), c_3 is Eq. (2.1), c_5 is Eq. (2.4), c_6 is Eq. (2.2) and c_8 is Eq. (2.5). The measurement equations are c_{m} which is Eq. (2.3) and c_{h1}, c_{h2} which are, respectively, the added measurements of the two levels h_1 and h_2. The constraints d_4 and d_7, respectively, express that $\dot{h}_1$ and $\dot{h}_2$ are the time derivatives of h_1 and h_2. The incidence matrix with respect to $\mathcal{X}$ is:

Σ_1	q_L	q_P	$\dot{h}_1$	h_1	q_{12}	h_2	$\dot{h}_2$	q_2
c_1	①			1				
c_2		①		1				
c_3	1	1		1	①			
d_4			①	1				
c_5				1	1	1		
c_{h1}				①				
c_6					1		1	1
d_7						1	①	
c_8						1		1
c_{m}								①
c_{h2}						①		

Based on the complete matching shown by the entries ①, the over-constrained subsystem produces three residuals whose structures are

$$\begin{aligned}\mathcal{C}\left(\rho_1\right) &= \{c_1, c_2, c_3, c_5, c_{h1}, c_{h2}\} \\ \mathcal{C}\left(\rho_2\right) &= \{c_1, c_2, c_3, c_6, d_7, c_{\mathrm{m}}, c_{h1}, c_{h2}\} \\ \mathcal{C}\left(\rho_3\right) &= \{c_8, c_{\mathrm{m}}, c_{h2}\}\,.\end{aligned}$$

1. What is the residuals' signature table.
2. The mathematical constraints d_4 and d_7 specify that $\dot{h}_1$ and $\dot{h}_2$ are the derivatives of h_1 and h_2. Discarding them (since they cannot be faulty), determine the system's distinguishability classes and draw the distinguishability table.
3. For each of the eight possible residual configurations, find the minimal hitting sets and draw the diagnosis table. □

Exercise 10.2 Two-tank system decomposition
This exercise illustrates Remark 10.5 still with the two-tank system. Assume each tank is a subsystem with the structures:

$$\begin{aligned}\boldsymbol{f}_1 \cup \boldsymbol{g}_1 &= \{c_1, c_2, c_3, d_4, c_5, c_{h1}\} \\ \boldsymbol{x}_1 &= \left\{q_L, q_P, h_1, \dot{h}_1, q_{12}\right\} \\ \overline{\boldsymbol{x}}_1 &= \{h_2\} \\ \boldsymbol{f}_2 \cup \boldsymbol{g}_2 &= \{c_6, d_7, c_8, c_{\mathrm{m}}, c_{h2}\} \\ \boldsymbol{x}_2 &= \left\{h_2, \dot{h}_2, q_2\right\} \\ \overline{\boldsymbol{x}}_2 &= \{q_{12}\}\,.\end{aligned}$$

The global incidence matrix is decomposed as follows,

Σ_1	q_L	q_P	$\dot{h}_1$	h_1	q_{12}	h_2
c_1	①			1		
c_2		①		1		
c_3	1	1		1	①	
d_4			①	1		
c_5				1	1	①
c_{h1}				①		

Σ_2	$\dot{h}_2$	h_2	q_2	q_{12}
c_6	1		1	①
d_7	①	1		
c_8		1	1	
c_{m}			①	
c_{h2}		①		

and two complete matchings with respect to the unknown variables are shown by ①:

1. How many local residuals are, respectively, provided by Σ_1 and Σ_2 and what are their structures?
2. Can you explain why there are less local residuals than when considering the global structure? □

Exercise 10.3 Coordination of local diagnosis
Consider a system in which there are three different estimation versions of an unknown variable x from the known variables $u' \cup y'$ (remember that the notation u', y' means u, y and a number of their time derivatives):

$$\begin{aligned} x &= f_1\left(u', y'\right) && \text{using the subset of constraints } C_1 = \{a, b, c, d\} \\ x &= f_2\left(u', y'\right) && \text{using the subset of constraints } C_2 = \{e, f\} \\ x &= f_3\left(u', y'\right) && \text{using the subset of constraints } C_3 = \{b, f, g, h\}. \end{aligned}$$

Three residuals are obtained:

$$\begin{aligned} \rho_1 &= f_1\left(u', y'\right) - f_2\left(u', y'\right) \\ \rho_2 &= f_1\left(u', y'\right) - f_3\left(u', y'\right) \\ \rho_3 &= f_2\left(u', y'\right) - f_3\left(u', y'\right). \end{aligned}$$

1. What are the structures of the residuals?
2. What is the distinguishability table?
3. Assuming there are three subsystems that run one residual each, what are the local diagnosis tables?
4. What is the coordinated diagnosis table? □

10.7 Bibliographical Notes

Fault-tolerant computing. Due to the increasing complexity of software applications and the increasing size of data bases, distributed computing and related reliability issues have been an important research area in the Computer Science community. A conceptual taxonomy of the basic concepts in the dependability of computer systems (reliability, availability, safety, confidentiality, integrity, maintainability, etc.) is presented in [10]. Many solutions have been proposed, ranging from node-level to system-level approaches: redundant execution of critical programmes on several nodes, providing each node with a fail-aware ability, with the capacity of testing its neighbours, of estimating the state of all nodes, of entering a fail-silent state [2, 101].

New problems in networked and multi-agent systems. A huge research activity has also been triggered in the control community on large-scale control systems distributed over networks and multi-agent systems. New theoretical problems range from the role of the information pattern in the problem solvability [137] to the controllability and observability analysis of networked dynamical systems [413]. Technological problems are not only connected with the controlled process (sensors, actuators, process components faults) but also with channel limitations (packet rates, sampling, delays) or channel failures (packet dropouts). The impact of such faults on the system stability and performance is analysed in [144, 369], and a survey of recent results on estimation, analysis and controller synthesis can be found in [60, 146, 307]. Estimation by means of geographically distributed sensors has been thoroughly studied using linear estimation [53, 81, 315] and Kalman filtering [1].

Analytical redundancy-based diagnosers. Although it involves signal derivatives, the applicability of the Analytical Redundancy-based approach is well established by observers or specific integration schemes, examples of which can be found in [350]. The logical theory of model-based FDI was developed in the artificial intelligence community [77], and its connections with the structural analysis approach were further analysed in [70].

Distributed diagnosis schemes. When the implementation of a global diagnoser is not technically possible, distributed diagnosis schemes rest on assigning a part of the global task to each subsystem/agent [74, 229]. Under specific assumptions about the locally available models and data, the investigated problems range from distributed estimation [53, 81, 83] and the design of a coordination process [312, 315] to robustness with respect to network uncertainties [144, 262, 360, 378], or model uncertainties and non-linearities. Global system models are often assumed to be available [100, 411], or local models are used along with a real-time estimator of the interconnections [314], based on global or only locally sensed information [295].

Distributed control and fault-tolerant control. The main features underlying the control or fault-tolerant control of distributed dynamical systems or networks of dynamical agents are the (often unknown) interactions between subsystems/agents and the limited amount of information available to make their local decisions. Networks of dynamical agents are a wide application area: [137] addresses the synthesis of control laws via a sub-optimal algorithm, the agents coordination problem is studied in [404] and the problem of achieving a consensus under partial information is the subject of [312]. For control systems distributed over a network, a generic fault-tolerance strategy is proposed in [267]. Reference [260] analyses the fault accommodation problem under partially available information, while the reconfiguration of the information pattern was recently shown to allow fault tolerance under some conditions [338].

Operations research and mathematical tools. The basic tools of operations research (task allocation problem) and lattices that are used in this chapter can be found in [73, 382].