Int. J. Appl. Math. Comput. Sci., 2011, Vol. 21, No. 1, 109–125
DOI: 10.2478/v10006-011-0008-z

ALGEBRAIC APPROACH FOR MODEL DECOMPOSITION: APPLICATION TO FAULT DETECTION AND ISOLATION IN DISCRETE–EVENT SYSTEMS

DENIS BERDJAG [*], VINCENT COCQUEMPOT [**], CYRILLE CHRISTOPHE [**], ALEXEY SHUMSKY [***], ALEXEY ZHIRABOK [***]

[*] Automatic Control Laboratory, LAMIH
University of Valenciennes (UVHC), Bât. Malvache, Le Mont Houy, 59313 Valenciennes cedex 9, France
e-mail: denis.berdjag@univ-valenciennes.fr

[**] Automatic Control Laboratory, LAGIS
Polytech'Lille, Lille 1 University, 59655 Villeneuve d'Ascq, France
e-mail: {vincent.cocquempot,cyrille.christophe}@univ-lille1.fr

[***] Institute for Marine Technology Problems
Far Eastern Branch of the Russian Academy of Sciences, 5a Sukhanova St, Vladivostok, 690600, Russia
e-mail:shumsky@mail.primorye.ru,zhirabok@mail.ru

This paper presents a constrained decomposition methodology with output injection to obtain decoupled partial models. Measured process outputs and decoupled partial model outputs are used to generate structured residuals for Fault Detection and Isolation (FDI). An algebraic framework is chosen to describe the decomposition method. The constraints of the decomposition ensure that the resulting partial model is decoupled from a given subset of inputs. Set theoretical notions are used to describe the decomposition methodology in the general case. The methodology is then detailed for discrete-event model decomposition using pair algebra concepts, and an extension of the output injection technique is used to relax the conservatism of the decomposition.

Keywords: algebraic approaches, decomposition methods, decoupling, discrete-event systems.

1. Introduction

The increasing demand for secure and reliable systems boosts up the research on accurate methods for Fault Detection and Isolation (FDI). The FDI problem has been studied extensively, using two major approaches: model-based and model-free. In this paper we are concerned with model-based approaches, which are classified in two major groups with respect to the type of model used: the continuous/discrete time-driven model or the discrete-event model.

Time-driven model-based FDI is achieved by analysing fault indicators or residuals, which are signals obtained by comparing measured outputs of the monitored process with the corresponding simulated outputs. Three major techniques are used for residual generation (Patton, 1994; Isermann, 2005): the parameter estimation approach (Isermann, 1984; Isermann and Freyermuth, 1991) or, more recently (Fliess and Join, 2003; Fliess *et al.*, 2004), the parity space approach (Gertler, 1991; Staroswiecki and Comtet-Varga, 2001) based on elimination (Diop, 1991; Cox *et al.*, 1991; Maquin *et al.*, 1997; Staroswiecki and Comtet-Varga, 2001) or on projection (Chow and Willsky, 1984; Isidori, 1995; Leuschen *et al.*, 2005), and the observer-based approach (Patton, 1994; Hammouri *et al.*, 2001; Jiang *et al.*, 2004; 2006; Lootsma, 2001).

When the process to be monitored is subject to multiple failures, in order to isolate each failure, multiple residuals are required, leading to the synthesis of a residual generator bank. When an isolable fault occurs, the residuals react in a specific pattern, called the fault signature, characterized by robustness/sensitivity properties of each signal. When every residual is sensitive (or robust) to only one unique fault and robust (or sensitive) to all the remaining failures, then the residuals are said to be structured.

Many FDI techniques are described in the literature for timed/untimed discrete-event model of centralized or distributed systems. We will focus on untimed (Sampath *et al.*, 1995;1996; Zad *et al.*, 2003) centralized systems (Lafortune *et al.*, 2001; Lin, 1994; Bavishi and Chong, 1994). The monitored system is abstracted using a Discrete-Event System (DES) model, characterized by a discrete state space and event-based dynamics. DES models are used because of the simplicity of the associated algorithms to test diagnosability (Sampath *et al.*, 1995). In the last few years a number of DES modeling frameworks have been developed for diagnosability analysis. One frequently used model is the Finite State Machine (FSM) (Sampath *et al.*, 1995; Zad *et al.*, 2003; Lin, 1994; Lafortune *et al.*, 2001; Bavishi and Chong, 1994). It is based on the assumption that the system consists of several distinct physical components, modelled as FSMs, which may share certain events.

The states of the FSM correspond to the internal states of a component and the transitions refer to its events. The events are considered to be observable or unobservable, the latter being further classified into internal and failure events. Transitions and states following failure events model the behaviour of the system after a failure. Using standard synchronous composition operations, the individual components are composed to form a global model that describes the behaviour of the complete system. FDI is achieved by using the complete system model to synthesize another FSM called the diagnoser, which is basically a state estimator that determines the condition (normal or failure) of the system. Another possible approach is to produce diagnosers corresponding to each individual component. Others approaches to DES diagnosability are process algebra-based approaches (Hamscher *et al.*, 1992; Hillston, 1996) and Petri net models (Boel and Jiroveanu, 2004; Benveniste *et al.*, 2003; Boubour *et al.*, 1997; Hadjicostis and Verghese, 1999; Giua, 1997).

Time-driven and event-based FDI approaches share a lot of common points, but mostly use different mathematical techniques. The multiplication of these mathematical tools limits the application of such approaches to particular domains and imposes, for each approach, a specific model of the monitored process. For instance, observer-based geometric methods require linear or nonlinear time-driven modelling of the process. A useful advancement would be to develop mathematical FDI tools that could be applied in a similar way to monitor systems described by time-driven or event-based models.

An in-depth study of the existing FDI approaches shows that similar decomposition-based methods were developed for the two types of models: a bank of residual generators based on linear or nonlinear observers, parity space approaches, individual component-based FSMs.

In this paper, we present a decomposition methodology for deterministic behavioural models. The decomposition method is said to be model-type-free because it does not depend in its principle on the model type (time-based or discrete-event-based). The objective is to obtain a partial model which is decoupled from a given subset of inputs (that may be failures) while remaining coupled with respect to another subset of selected inputs (that may also be failures). These partial models may be used for FDI as, e.g., in the works of Patton (1994), Gertler (1998), Blanke *et al.* (2003), Kinnaert (1999) or Maquin *et al.* (1997) for continuous-time systems and by Sampath *et al.* (1996), Zad (1999) or Lefebvre (1999) for discrete-event methods. FDI is achieved by measuring the consistency between measured outputs of the monitored process and the corresponding simulated outputs of each partial model. The coupling and decoupling properties of each partial model allow detecting the occurrence of selected faults while ignoring the rest of them. Using several partial models with different coupling/decoupling properties leads to structured residual vectors achieving the isolation of all faults considered.

The methodology is described using a particular algebraic formulation, which allows considering the decomposition of continuous-time models and discrete-event models using the same algorithm. Of course, even if the general methodology is the same, some computations at given steps of the algorithm are specific to the domains considered. In this paper we use a particular algebraic formalism, inspired by the *algebra of functions* (Shumsky, 1991; Zhirabok and Shumsky, 1993; Shumsky and Zhirabok, 2006). Decomposition based on the algebra of functions was the topic of our previous publications (Berdjag *et al.*, 2006a; 2006c), where an iterative decomposition algorithm using output injection was presented for nonlinear continuous-time models. This paper emphasizes the extension of the decomposition algorithm to major types of deterministic behavioural models, using *set-theory* notions (Vereshchagin and Shen, 2002). The main algorithm is then used to propose a constrained decomposition of FSMs using *pair algebra* (Hartmanis and Stearns, 1966). There is a straightforward relation between set-theoretical and pair algebra formalisms, and we extensively use this relationship to propose an adapted formulation of the decomposition constraints for both cases. In particular, output injection is used to loosen decomposition constraints in the discrete-event case. The output injection technique is a well-known method of continuous-time model decoupling, and to the best of our knowledge it has not been yet employed for DES model decoupling.

The paper is organized as follows. In Section 2, a constrained decomposition problem is formulated using the set-theoretical framework. In Section 3, the decomposition constraints and conditions are detailed and organized to build a general decomposition algorithm with output injection. Section 4 provides basic reminders about *partitions* and *pair algebra* operators. In Section 5, constrained decomposition of discrete-event models with output injection is presented. An example is given in Section 6 to illustrate the decomposition and the benefits of output injection. Afterwards, conclusions and perspectives on future work are presented, and finally an appendix with illustrative examples on partition operations closes the paper.

2. Problem formulation

2.1. Preliminaries. As a general principle, it is possible to represent deterministic behavioural models, denoted by Σ, using the following quintuple:

$$(\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{F}, \mathcal{H}), \tag{1}$$

where $\mathcal{X}$ is the state set, $\mathcal{U}$ is the input set and $\mathcal{Y}$ is the output set. $\mathcal{F}$ and $\mathcal{H}$ are functions defined by

$$\mathcal{F}: \mathcal{X} \times \mathcal{U} \longrightarrow \mathcal{X} \quad \text{and} \quad \mathcal{H}: \mathcal{X} \times \mathcal{U} \longrightarrow \mathcal{Y}. \tag{2}$$

The function $\mathcal{F}$ is the state function and the function $\mathcal{H}$ is the output function. It is well known that the state function $\mathcal{F}$ is invariant (involutive) by definition, i.e., $\mathcal{F}(\mathcal{X},\mathcal{U}) \subseteq \mathcal{X}$ (see Isidori, 1995) for the definition of invariance. We make the choice of omitting the initial state $\mathcal{X}_0$ specification for the sake of simplicity, since it has no influence on the decomposition process.

The representation (1) allows describing continuous-time and discrete-event deterministic models using the same formalism. Indeed, if Σ is a continuous-time model, then the sets $\mathcal{X}, \mathcal{U}, \mathcal{Y}$ are infinite sets of dimensions n, l, m, respectively, i.e., $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^l$, $\mathcal{Y} \subseteq \mathbb{R}^m$. The state and output functions are defined by

$$\mathcal{F}: \mathbb{R}^n \times \mathbb{R}^l \longrightarrow \mathbb{R}^n \quad \text{and} \quad \mathcal{H}: \mathbb{R}^n \times \mathbb{R}^l \longrightarrow \mathbb{R}^m. \tag{3}$$

If the model Σ is a discrete-event model, then the sets $\mathcal{X}, \mathcal{U}, \mathcal{Y}$ are finite sets of respective cardinalities n', l', m':

$$\mathcal{X} = \{x_1, \dots, x_{n'}\}, \qquad \mathcal{U} = \{u_1, \dots, u_{l'}\}, \qquad \mathcal{Y} = \{y_1, \dots, y_{m'}\}.$$

We assume that the model Σ contains multiple dynamics. Every single dynamic is affected by a particular subset of inputs or input events and ignores the rest of the inputs. These dynamics can be represented by *partial models*. A partial model Σ_* is a model which replicates the behaviour of a part of the "complete" model Σ. The models Σ and Σ_* are said to be equivariant, i.e., for the same sequence of inputs (or input events), the states and outputs of Σ_* and Σ are *bisimilar*. Two states (or outputs) are considered bisimilar if the states (outputs) remain consistent as long as the two models are excited by the same input sequence with consistent initial states.

Definition 1. (Bisimilarity) Consider $\mathcal{X}$ and $\mathcal{X}'$ two sets of equal cardinalities. Let $\rightarrow$ and $\mapsto$ be two relations defined on $\mathcal{X}^2$ and $\mathcal{X}'^2$, respectively. We say that the elements of $\mathcal{X}$ and $\mathcal{X}'$ are *bisimilar* if and only if there is a mapping $\theta : \mathcal{X} \rightarrow \mathcal{X}'$ such that

$$\forall x \in \mathcal{X},\ \exists \theta : (x \rightarrow \tilde{x}) \Leftrightarrow (\theta(x) \mapsto \theta(\tilde{x})).$$

Let $\Psi_{\mathcal{X}}$, $\Psi_{\mathcal{U}}$ and $\Psi_{\mathcal{Y}}$ be the power-sets of $\mathcal{X}, \mathcal{U}$ and $\mathcal{Y}$, respectively, i.e., $\Psi_{\mathcal{X}} = 2^{\mathcal{X}}$, $\Psi_{\mathcal{U}} = 2^{\mathcal{U}}$ and $\Psi_{\mathcal{Y}} = 2^{\mathcal{Y}}$. Consider another model Σ_* defined by

$$(\mathcal{X}_*, \mathcal{U}_*, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*), \tag{4}$$

where $\mathcal{X}_*, \mathcal{U}_*$ and $\mathcal{Y}_*$ are subsets of $\Psi_{\mathcal{X}}$, $\Psi_{\mathcal{U}}$ and $\Psi_{\mathcal{Y}}$, and $\mathcal{F}_*$ and $\mathcal{H}_*$ are restrictions of $\mathcal{F}$, $\mathcal{H}$ on $\Psi_{\mathcal{X}} \times \Psi_{\mathcal{U}} \mapsto \Psi_{\mathcal{X}}$ and $\Psi_{\mathcal{X}} \times \Psi_{\mathcal{U}} \mapsto \Psi_{\mathcal{Y}}$, respectively.

Definition 2. (Partial model) We say that the model $\Sigma_*(\mathcal{X}_*, \mathcal{U}_*, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*)$ constitutes a *partial model* of $\Sigma(\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$ if and only if the functions $\mathcal{F}_*$ and $\mathcal{H}_*$ are restrictions of $\mathcal{F}$, $\mathcal{H}$ on $\Psi_{\mathcal{X}} \times \Psi_{\mathcal{U}} \mapsto \Psi_{\mathcal{X}}$ and $\Psi_{\mathcal{X}} \times \Psi_{\mathcal{U}} \mapsto \Psi_{\mathcal{Y}}$, and the sets $\mathcal{X}_*$, $\mathcal{U}_*$ and $\mathcal{Y}_*$ are given by

$$\mathcal{X} \xrightarrow{\Theta_{\mathcal{X}}} \mathcal{X}_*, \quad \mathcal{U} \xrightarrow{\Theta_{\mathcal{U}}} \mathcal{U}_*, \quad \mathcal{Y} \xrightarrow{\Theta_{\mathcal{Y}}} \mathcal{Y}_*, \tag{5}$$

where $\Theta_{\mathcal{X}}$, $\Theta_{\mathcal{U}}$ and $\Theta_{\mathcal{Y}}$ are functions on $\Psi_{\mathcal{X}}$, $\Psi_{\mathcal{U}}$ and $\Psi_{\mathcal{Y}}$ ensuring that the outputs of Σ and Σ_* are bisimilar for the same sequence of inputs $u \in \mathcal{U}$ and $\Theta(u)_{\mathcal{U}}(u) \in \mathcal{U}_*$.

The homomorphism is a well-suited mathematical concept to express the link between the model of the system and its partial models. The homomorphism is a structure preserving map from an algebraic construct to another algebraic construct.

Definition 3. (Homomorphism) Consider a set $\mathcal{X}$ and a function $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{X}$. The function $\Theta : \mathcal{X} \rightarrow \mathcal{X}$ is a *homomorphism* if the following relation holds:

$$\forall x \in \mathcal{X} : \Theta(\mathcal{F}(x)) = \mathcal{F}(\Theta(x)).$$

This notion was extended separately to event-driven and time-driven dynamical systems in the literature, (Hartmanis and Stearns, 1966). We propose here a definition suited for the problem considered.

Definition 4. (Model homomorphism) The triple $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}}, \Theta_{\mathcal{Y}})$ is a structure preserving map of the model $\Sigma(\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{F}, \mathcal{H})$ into $\Sigma_*(\mathcal{X}_*, \mathcal{U}_*, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*)$ if the following relation holds:

$$\begin{aligned} &\forall x \in \mathcal{X}, u \in \mathcal{U}, y \in \mathcal{Y} : \\ &\qquad \Theta_{\mathcal{X}}(\mathcal{F}(x,u)) = \mathcal{F}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)), \\ &\qquad \Theta_{\mathcal{Y}}(\mathcal{H}(x,u)) = \mathcal{H}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)), \end{aligned}$$

where $\Theta_{\mathcal{X}} : \mathcal{X} \rightarrow \mathcal{X}_*$, $\Theta_{\mathcal{U}} : \mathcal{U} \rightarrow \mathcal{U}_*$ and $\Theta_{\mathcal{Y}} : \mathcal{Y} \rightarrow \mathcal{Y}_*$. The triple $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}}, \Theta_{\mathcal{Y}})$ is said to be a *model homomorphism*.

Proposition 1. *Let Σ and Σ_* be two models with Σ_* obtained from Σ. Σ_* is a partial model of Σ if and only if the triple $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}}, \Theta_{\mathcal{Y}})$ is a model homomorphism.*

Proof. Necessity and sufficiency are obvious: if $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}}, \Theta_{\mathcal{Y}})$ is a homomorphism, the following relations hold:

$$\begin{aligned}&\forall x \in \mathcal{X}, \forall u \in \mathcal{U}:\\ &\qquad \Theta_{\mathcal{X}}(\mathcal{F}(x,u)) = \mathcal{F}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)),\\ &\qquad \Theta_{\mathcal{Y}}(\mathcal{H}(x,u)) = \mathcal{H}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)).\end{aligned} \tag{6}$$

If Σ_* replicates partially Σ, then states and outputs are bisimilar, i.e.,

$$\begin{aligned}&\forall x \in \mathcal{X}, \forall u \in \mathcal{U}:\\ &\qquad \Theta_{\mathcal{X}}(\mathcal{F}(x,u)) = \mathcal{F}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)),\\ &\qquad \Theta_{\mathcal{Y}}(\mathcal{H}(x,u)) = \mathcal{H}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)).\end{aligned} \tag{7}$$

Obviously, (6) is identical to (7). ■

Remark 1.

- Proposition 1 is an extension of the automata homomorphism (Hartmanis and Stearns, 1966) to the general case.
- If only state bisimilarity is required, then only $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}})$ needs to be homomorphic.
- The dynamics of the partial model Σ_* are defined by $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}})$ only.

2.2. Practical application. The model Σ is supposed to represent the real behaviour of a physical system. This means that the input set $\mathcal{U}$ represents the occurrence of real events. The objective behind decomposition is to obtain a decoupled partial model Σ_* which allows detecting selected faults and ignoring others. We assume that the faults are unknown (unobservable) inputs or events (Patton, 1994; Sampath *et al.*, 1995). We also assume that perturbations and noise are unknown inputs. As a result, the input set is divided in three disjoint subsets:

$$\mathcal{U} = \mathcal{U}_c \cup \mathcal{U}_\rho \cup \mathcal{U}_\gamma, \tag{8}$$

where $\mathcal{U}_\rho$ contains inputs (subset of faults) to be detected, $\mathcal{U}_\gamma$ contains inputs to be ignored (perturbations or supplementary subset of faults) and $\mathcal{U}_c$ regroups the known control inputs. $\mathcal{U}_\rho$ and $\mathcal{U}_\gamma$ form unknown (unobservable) input sets.

Consider the function $\Theta_{\mathcal{U}}$ defined on $\Psi_{\mathcal{U}}$ such that

$$\mathcal{U}_\gamma \subseteq \ker(\Theta_{\mathcal{U}}), \quad \mathcal{U}_c \cup \mathcal{U}_\rho \nsubseteq \ker(\Theta_{\mathcal{U}}), \tag{9}$$

where $\ker(\Theta_{\mathcal{U}})$ denotes the kernel of the function $\Theta_{\mathcal{U}}$.

A partial model Σ_* is obtained using the homomorphism $(\Theta_{\mathcal{X}}, \Theta_{\mathcal{U}}, \Theta_{\mathcal{Y}})$ with $\Theta_{\mathcal{U}}$ from (9). Σ_* will replicate the behaviour of Σ, but it will totally ignore the inputs from $\mathcal{U}_\gamma$. However, if Σ_* and Σ are excited by different inputs from $\mathcal{U}_c \cup \mathcal{U}_\rho$, state and output discrepancies will appear.

Therefore, discrepancies can be used to detect unexpected events (represented by inputs from $\mathcal{U}_c \cup \mathcal{U}_\rho$) in the input sequence. Moreover, if an occurring unexpected event belongs to $\mathcal{U}_\gamma$, no output discrepancy is observed, since Σ_* is decoupled from $\mathcal{U}_\gamma$.

Application of these concepts to fault detection and isolation is straightforward: the effects of process failures may be modelled as unknown inputs (see Patton, 1994). Fault detection and isolation is performed by comparing real process outputs with simulated outputs of the partial model Σ_*. This comparison allows computing residuals and the analysis of these residuals will grant us information about failure occurrences in the real process. In order to produce a *structured* residual that allows detecting a subset of failures and ignoring the other subset, the unknown inputs representing the failures to detecting are grouped in $\mathcal{U}_\rho$.

The following section details the method for obtaining Σ_*, i.e., determining $\mathcal{X}_*, \mathcal{Y}_*, \mathcal{F}_*$ and $\mathcal{H}_*$ for a given $\mathcal{U}_*$.

3. Decomposition of generic behavioural models

3.1. Decomposition procedure. The decomposition method proposed in this section is presented as an iterative procedure, with $\Sigma(\mathcal{X}, \mathcal{U}_c \cup \mathcal{U}_\rho \cup \mathcal{U}_\gamma, \mathcal{Y}, \mathcal{F}, \mathcal{H})$ as the input and $\Sigma_*(\mathcal{X}_*, \mathcal{U}_*, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*)$ as the result. In order to ensure that Σ_* is decoupled from $\mathcal{U}_\gamma$ and coupled with respect to $\mathcal{U}_\rho$, the decomposition procedure is constrained to *coupling with respect to* $\mathcal{U}_\rho$ and *decoupling from* $\mathcal{U}_\gamma$ properties of the state set $\mathcal{X}_*$.

However, the success of the constrained decomposition procedure is essentially based on the fulfilment of the existence condition given in Proposition 1. This somewhat complex condition is simplified in the following. Moreover, in order to improve the procedure, an extension is proposed based on a technique called *output injection*.

In the following, both constraints, coupling to $\mathcal{U}_\rho$ and decoupling from $\mathcal{U}_\gamma$, are detailed. Necessary conditions to obtain a partial model Σ_* and to guarantee bisimilar states and outputs are also given. An extension of the decomposition procedure using output injection is proposed. Finally, an iterative decomposition algorithm is synthesized.

3.2. Decomposition constraints.

3.2.1. Decoupling constraint. Consider some partial model Σ_* such that Σ and Σ_* are equivariant. Σ_* is decoupled from $\mathcal{U}_\gamma$ if $\Theta_{\mathcal{U}}(\mathcal{U}_\gamma) \cap \mathcal{U}_* = \emptyset$. This means that

$\mathcal{X}_*$ does not intersect the state set coupled with respect to the subset $\mathcal{U}_\gamma$, i.e.

$$\Theta_{\mathcal{X}}^{-1}(\mathcal{X}_*) \cap \mathcal{X}_\gamma = \emptyset, \tag{10}$$

where $\Theta_{\mathcal{X}}^{-1}$ denotes the inverse of $\Theta_{\mathcal{X}}$. The set $\mathcal{X}_\gamma$ is given by

$$\mathcal{X}_\gamma = \mathcal{F}(\mathcal{X},\mathcal{U}_\gamma). \tag{11}$$

3.2.2. Coupling constraint. In the same way, the coupling condition is expressed: Σ_* is coupled with respect to $\mathcal{U}_\rho$ if $\Theta_{\mathcal{U}}(\mathcal{U}_\rho) \cap \mathcal{U}_* \neq \emptyset$. This means that the state subset coupled with respect to the subset $\mathcal{U}_\rho$ is not included in the kernel of $\Theta_{\mathcal{X}}$, i.e.,

$$\mathcal{X}_\rho \not\subseteq \ker(\Theta_{\mathcal{X}}). \tag{12}$$

By analogy with (11), the set $\mathcal{X}_\rho$ is given by

$$\mathcal{X}_\rho = \mathcal{F}(\mathcal{X},\mathcal{U}_\rho). \tag{13}$$

3.3. Decomposition conditions.

Two conditions are required: the invariance condition is needed to show that Σ_* is a partial model of Σ and can be used to mirror a partial evolution of Σ, while the output condition is necessary to ensure bisimilar outputs of Σ_* and Σ.

3.3.1. Invariance condition. The existence condition given in Proposition 1 is developed.

Lemma 1. *Let Σ and Σ_* be two models with Σ_* obtained from Σ. The restriction (cf. Vereshchagin and Shen, 2002) of the function $\mathcal{F}_*$ on $\Theta_{\mathcal{X}}(\mathcal{X})$ is invariant if and only if Σ_* is a partial model of Σ .*

Proof. (Necessity) If the restriction of $\mathcal{F}_*$ on $\Theta_{\mathcal{X}}(\mathcal{X})$ is not invariant, then

$$\mathcal{F}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \not\subseteq \Theta_{\mathcal{X}}(\mathcal{X}),$$

which means that

$$\exists x \in \mathcal{X}\ \exists u \in \mathcal{U}\ :\ \mathcal{F}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)) \neq \Theta_{\mathcal{X}}(\mathcal{F}(x,u))$$

and implies that Σ_* is not a partial model of Σ, since it does not replicate Σ for all x and u.

(Sufficiency) If Σ_* is a partial model of Σ, then

$$\forall x \in \mathcal{X}\ \forall u \in \mathcal{U} : \mathcal{F}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)) = \Theta_{\mathcal{X}}(\mathcal{F}(x,u))$$

and

$$\mathcal{F}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \subseteq \Theta_{\mathcal{X}}(\mathcal{F}(\mathcal{X},\mathcal{U})). \tag{14}$$

We know that $\mathcal{F}$ is invariant by definition, so the relation $\Theta_{\mathcal{X}}(\mathcal{F}(\mathcal{X},\mathcal{U})) = \Theta_{\mathcal{X}}(\mathcal{X})$ is true. Thus (14) becomes

$$\mathcal{F}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \subseteq \Theta_{\mathcal{X}}(\mathcal{X}),$$

which means that the restriction of $\mathcal{F}_*$ on $\Theta_{\mathcal{X}}(\mathcal{X})$ is invariant. ■

Finally, the new existence condition is

$$\mathcal{F}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \subseteq \Theta_{\mathcal{X}}(\mathcal{X}). \tag{15}$$

3.3.2. Output condition. If outputs of Σ_* and Σ are bisimilar, then it is possible to check the discrepancy between the evolutions of Σ_* and Σ and the *output condition* is fulfilled. This condition makes sense only if the invariance condition is satisfied.

Output bisimilarity is ensured if Proposition 1 is fulfilled,

$$\forall x \in \mathcal{X}, \forall u \in \mathcal{U} : \\ \mathcal{H}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)) = \Theta_{\mathcal{Y}}(\mathcal{H}(x,u)). \tag{16}$$

However, this is the perfect case. Practically, only some bisimilar outputs are required to check the consistency of Σ and Σ_*. Let us replace $\mathcal{Y}$ with a subset $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$. The relation (16) becomes

$$\forall u \in \mathcal{U},\ \forall x \in \mathcal{X},\ \exists \tilde{\mathcal{Y}} \subseteq \mathcal{Y}\ : \\ \forall y \in \tilde{\mathcal{Y}} \Rightarrow \mathcal{H}_*(\Theta_{\mathcal{X}}(x), \Theta_{\mathcal{U}}(u)) = \Theta_{\mathcal{Y}}(\mathcal{H}(x,u)) \tag{17}$$

or

$$\exists \tilde{\mathcal{Y}} \subseteq \mathcal{Y} :\ \mathcal{Y}_* \cap \Theta_{\mathcal{Y}}(\tilde{\mathcal{Y}}) \neq \emptyset. \tag{18}$$

The relation (18) is the final form of the output condition.

3.4. Output injection.

In some cases, the constraints of the decomposition are too strong, resulting in an impossible decomposition, i.e., there is no restriction of $\mathcal{F}_*$ on a given $\mathcal{X}_* = \Theta_{\mathcal{X}}(\mathcal{X})$ satisfying (15):

$$\mathcal{F}_*(\mathcal{X}_*,\mathcal{U}) \not\subseteq \mathcal{X}_*. \tag{19}$$

A special technique called *output injection* may be then used in order to relax the invariance condition. Output injection is a well-known technique for continuous-time model decoupling. The main idea is to replace the information loss due to the truncated state set $\mathcal{X}_*$ by extending the input set of Σ_* with selected outputs of Σ.

Consider a set $\tilde{\mathcal{X}} \subseteq \mathcal{X}$ such that $\mathcal{X}_* \subseteq \tilde{\mathcal{X}}$ and

$$\mathcal{F}_*(\tilde{\mathcal{X}},\mathcal{U}) \subseteq \tilde{\mathcal{X}}. \tag{20}$$

The relation (20) is always fulfilled, since we can take $\tilde{\mathcal{X}} = \mathcal{X}$.

Let ξ be a function on $\mathcal{Y} \mapsto \mathcal{X}_*$ such as $\tilde{\mathcal{X}} = \Theta_{\mathcal{X}}(\mathcal{X}) \cup \xi(\mathcal{Y})$ and $\xi(\mathcal{Y}) = \mathcal{X}_{inj}$. The relation (20) is rewritten using ξ,

$$\mathcal{F}_*(\Theta_{\mathcal{X}}(\mathcal{X}) \cup \xi(\mathcal{Y}),\mathcal{U}) \subseteq \Theta_{\mathcal{X}}(\mathcal{X}) \cup \xi(\mathcal{Y}). \tag{21}$$

The relation (21) ensures the existence of a state function for Σ_* denoted by $\tilde{\mathcal{F}}_* : \Psi_{\mathcal{X}} \times \mathcal{Y} \times \Psi_{\mathcal{U}} \to \mathcal{X}_*$, based on the functions $\mathcal{F}_*$ and ξ for Σ_* such that

$$\tilde{\mathcal{F}}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \mathcal{Y}, \Theta_{\mathcal{U}}(\mathcal{U})) \subseteq \Theta_{\mathcal{X}}(\mathcal{X}) \cup \mathcal{X}_{inj}. \tag{22}$$

Therefore, the appropriate use of the output injection $\xi(\mathcal{Y})$ ensures the fulfilment of the invariance condition.

The relation (22) is referred to as the *extended invariance condition*. Notice that we use a different notation for the state function to emphasize that $\tilde{\mathcal{F}}_*$ is not a proper restriction of the function $\mathcal{F}$, mathematically speaking, but rather a modification based on the restriction $\mathcal{F}_*$.

For the sake of simplicity, in the following, we will refer to the state function of Σ_* as $\mathcal{F}_*$ with or without output injection.

Remark 2. In order to satisfy the decoupling condition (11) and to keep Σ_* decoupled, $\mathcal{X}_{inj}$ must be independent from $\mathcal{U}_\gamma$, i.e.,

$$\Theta_{\mathcal{X}}^{-1}(\mathcal{X}_{inj}) \cap \mathcal{X}_\gamma = \emptyset, \tag{23}$$

where $\mathcal{X}_\gamma$ is the same as in (11). This means that an appropriate selection of outputs to be injected must be performed. In the following, the injected output is denoted by $\mathcal{Y}_{inj}$.

3.5. Decomposition algorithm. The decomposition procedure is formed as solving a constrained optimization problem. An iterative pseudo-algorithm is designed (Algorithm 1) to represent the three steps needed to obtain the decomposition function: The first step consists in the determination of the largest decoupled state set $\mathcal{X}^0$ using relation (10), and to do so, we also need to determine $\mathcal{X}_\gamma$ using the relation (11). Some other key elements are determined: the set $\mathcal{X}_\rho$ using (13) to check the coupling constraint, the functions $\mathcal{F}_*$ and $\mathcal{H}_*$ since they are necessary to describe the partial model, and finally, the output injection $\mathcal{Y}_{inj}$. This is achieved picking the observable part of the set $\mathcal{X}^0$, i.e., $\mathcal{H}(\mathcal{X}^0, \mathcal{U} - \mathcal{U}_\gamma) = \mathcal{Y}_{inj}$ and $\mathcal{X}_{inj} = \Theta_{\mathcal{X}}^0(\mathcal{H}^{-1}(\mathcal{Y}_{inj}))$. The second step is an iterative procedure in order to determine the largest invariant subset with respect to $\mathcal{F}$ in $\mathcal{X}^0$ along with the map $\Theta_{\mathcal{X}}$. The principle is to determine an initial set of decomposition candidates satisfying the decoupling constraint represented by the function $\Theta_{\mathcal{X}}^0$, and to determine $\mathcal{X}_*$ and $\Theta_{\mathcal{X}}$ using an iterative loop, based on a scheme proposed by Shumsky (1991). When the extended invariance condition is fulfilled, the loop ends and the result is saved for the next step.

The final step consists in checking the coupling constraint and the output condition using (17) and (18), and if the two conditions are satisfied, in building the quintuple describing the partial model Σ_* using the decomposition function $\Theta_{\mathcal{X}}$.

Algorithm 1 shows the steps required to determine the decoupling function $\Theta_{\mathcal{X}}$, if a decomposition is possible. However, this *set-theoretical* formalism is difficult to implement directly. Special mathematical techniques are proposed to simplify the implementation.

A possible approach is to define mathematical *delimiters* used to regroup all set elements into one mathematical entity. It is then possible to manipulate the delimiters rather than deal with the corresponding elements individually. In the case of infinite sets, set delimiters are defined using functions (Shumsky, 1991). In the case of finite sets, set delimiters are defined using partitions of el-

Algorithm 1 Decoupling algorithm

Require: $\Sigma(\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{F}, \mathcal{H}), \mathcal{U}_\rho, \mathcal{U}_\gamma$;
  Determine $\mathcal{X}^0$ and $\Theta_{\mathcal{X}}^0$ such that
    $\mathcal{X}^0 = \mathcal{X} - \mathcal{X}_\gamma$ and $\Theta_{\mathcal{X}}^0(\mathcal{X}) = \mathcal{X}^0$
    with $\mathcal{X}_\gamma = \mathcal{F}(\mathcal{X}, \mathcal{U}_\gamma)$;
  Determine $\mathcal{X}_\rho = \mathcal{F}(\mathcal{X}, \mathcal{U}_\rho)$;
  Select an appropriate $\Theta_{\mathcal{U}}$ such that $\mathcal{U}_\gamma \subseteq \ker(\Theta_{\mathcal{U}})$;
  Determine $\mathcal{Y}_{inj} \subseteq \mathcal{H}(\mathcal{X}^0, \mathcal{U} - \mathcal{U}_\gamma)$ and $\mathcal{X}_{inj}$;
  Determine $\mathcal{F}_*, \mathcal{H}_*$ restrictions of $\mathcal{F}, \mathcal{H}$;
  {Initialization}: Set i=1;
  Choose $\mathcal{X}^1$ and $\Theta_{\mathcal{X}}^1$ such that
    $\mathcal{X}^1 \subseteq \mathcal{X}^0$ and $\Theta_{\mathcal{X}}^1(\mathcal{X}) = \mathcal{X}^1$;
  { For the first loop any singleton $\mathcal{X}^1$ can be taken.}
  while $\mathcal{F}_*(\Theta_{\mathcal{X}}^i(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \nsubseteq \Theta_{\mathcal{X}}^i(\mathcal{X})$ **do**
    Determine the subset $\mathcal{X}^{i+1} \subseteq \mathcal{X}^0$ such that
      $\mathcal{F}_*(\Theta_{\mathcal{X}}^i(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \subseteq \mathcal{X}_{inj} \cup \mathcal{X}^1 \cup \ldots \cup \mathcal{X}^{i+1}$;
    Determine the function $\Theta_{\mathcal{X}}^{i+1}$ such that
      $\Theta_{\mathcal{X}}^{i+1}(\mathcal{X}) = \mathcal{X}_{inj} \cup \mathcal{X}^1 \cup \ldots \cup \mathcal{X}^{i+1}$;
    Increment i;
  end while
  $\Theta_{\mathcal{X}} = \Theta_{\mathcal{X}}^i$;
  if $\Theta_{\mathcal{X}} = \emptyset$ **then**
    return Decoupling impossible;
  else
    Determine $\Theta_{\mathcal{Y}}$ and $\tilde{\mathcal{Y}} \subseteq \mathcal{Y}$ such that
      $\mathcal{H}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U})) \cap \Theta_{\mathcal{Y}}(\tilde{\mathcal{Y}}) \neq \emptyset$;
    if $\exists \Theta_{\mathcal{Y}}, \tilde{\mathcal{Y}}$ **then**
      Output condition satisfied by $\Theta_{\mathcal{X}}$;
    else
      Output condition not satisfied by $\Theta_{\mathcal{X}}$;
      Take $\mathcal{X}^1 \nsubseteq \mathcal{F}(\Theta_{\mathcal{X}}(\mathcal{X}), \mathcal{U})$;
      if Impossible **then**
        Go to END
      else
        Go to {Initialization}
      end if
    end if
    if $\mathcal{X}_\rho \nsubseteq \ker(\Theta_{\mathcal{X}})$ **then**
      Coupling constraint not satisfied by $\Theta_{\mathcal{X}}$;
      Take $\mathcal{X}^1 \subseteq \mathcal{F}(\Theta_{\mathcal{X}}(\mathcal{X}), \mathcal{U})$;
      if Impossible **then**
        Go to END
      else
        Go to {Initialization}
      end if
    else
      Coupling constraint satisfied by $\Theta_{\mathcal{X}}$;
    end if
    $\mathcal{U}_* = \mathcal{U} \cup \mathcal{Y}_{inj}$,
    $\mathcal{X}_* = \Theta_{\mathcal{X}}(\mathcal{X})$,
    $\mathcal{Y}_* = \mathcal{H}_*(\Theta_{\mathcal{X}}(\mathcal{X}), \Theta_{\mathcal{U}}(\mathcal{U}))$
  end if
  return $\Sigma_*(\mathcal{X}_*, \mathcal{U}_*, \mathcal{Y}_*, \mathcal{F}_*, \mathcal{H}_*)$

ements (Hartmanis and Stearns, 1966).

The set of all delimiters with the corresponding mathematical relations forms an algebraic structure. If the definition sets of a model are finite, *pair algebra* is involved. Pair algebra was introduced by Hartmanis and Stearns (1966) to manipulate partitions of finite elements. An extension to infinite sets of elements was proposed by Shumsky (1991) as well as Zhirabok and Shumsky (1993), using functions to define partitions of the different sets. The algebraic structure used is known under the name of the *algebra of functions*. Recently, the algebra of functions was used in several topics of model-based monitoring for deterministic systems (Berdjag, 2006b; 2006c), uncertain systems (Shumsky, 2007) and canonical decomposition (Zhirabok, 2006). An analysis of the algebra of functions was presented by Zhirabok and Shumsky (1993) and Berdjag *et al.* (2006b).

An implementation of Algorithm 1 is proposed in Section 5 using pair algebra, for the *finite-state set* case. The following section will recall and introduce the notions that will be manipulated for such implementation.

4. Partitions and pair algebra

Reminders on partitions and partition operations are provided in this section, along with definitions of pair algebra operators. Examples provided for each case are regrouped in Appendix.

4.1. Mathematical background.

4.1.1. Partition. Consider some finite set S. A partition π on S is a collection of disjoint subsets of S whose set union is S. These subsets are called blocks and denoted by B^{π}_{α}, where α is an element of S, which literally means "the partition block containing the element α". For example, if a block is composed of two elements $\{\alpha, \beta\}$, then it can be referred to using the notation B_α or B_β,

$$\pi = \{B_\alpha\} \text{ such that } \begin{cases} B_\alpha \cap B_\beta = \emptyset \text{ for } \alpha \neq \beta, \\ \bigcup\{B_\alpha\} = S. \end{cases} \tag{24}$$

Consider a block B from π, and two elements s and t from S. If s and t are contained in the same block B of π, then we have $s \equiv t(\pi)$.

Remark 3. If a confusion between blocks of two different partitions appears in the following, for example, π_1 and π_2, then the following notation is used for the blocks: $B^{\pi_1}_{\alpha}$ and $B^{\pi_2}_{\alpha}$

4.1.2. Operations on partitions. Let S be a set and π_1 and π_2 two partitions on S. s and t are two elements from S. The operations "$\cdot$" and "$+$" along with the relations "$\leq$" and "$=$" are defined:

- $\pi_1 \cdot \pi_2$ is the partition on S such that

  $$s \equiv t(\pi_1 \cdot \pi_2) \text{ iff } s \equiv t(\pi_1) \text{ and } s \equiv t(\pi_2).$$

- $\pi_1 + \pi_2$ is the partition on S such that

  $$s \equiv t(\pi_1 + \pi_2) \text{ iff there is a sequence in } S$$

  $$s = s_0, s_1, \ldots, s_n = t,$$

  for which either

  $$s_i \equiv s_{i+1}(\pi_1) \text{ or } s_i \equiv s_{i+1}(\pi_2)\ ,\ 0 \leq i \leq n-1.$$

- $\pi_1 \leq \pi_2$ if and only if $\pi_1 \cdot \pi_2 = \pi_1$ and $\pi_1 + \pi_2 = \pi_2$. Partition π_2 is said larger than or equal to π_1.

- $\pi_1 = \pi_2$ if and only if $\pi_1 \leq \pi_2$ and $\pi_2 \leq \pi_1$. Partitions π_1 and π_2 are equal.

It can be shown that the relation $\leq$ is a partial order on the set of all the possible partitions on S, denoted by Π_S (see Hartmanis and Stearns, 1966). The set Π_S is said to be ordered by the partial order relation $\leq$ with the smallest partition denoted by $\mathbb{O}$ and the largest partition denoted by $\mathbb{I}$. For example, let $S = \{1, 2, 3\}$. The smallest partition is given by $\mathbb{O} = \{\{1\}, \{2\}, \{3\}\}$ and the largest by $\mathbb{I} = \{\{1, 2, 3\}\}$

For a more detailed overview on partition operations with examples, check the Appendix.

4.2. Substitution property.

Let S and I be two sets and δ a function defined by

$$\delta : S \times I \longrightarrow S.$$

Let π be a partition on S. The partition π is said to have the *substitution property* with respect to the function δ if and only if

$$s \equiv t(\pi) \Rightarrow \delta(s, i) \equiv \delta(t, i)(\pi)\, \forall i \in I. \tag{25}$$

If $\pi = \{B_\alpha\}$, for all $\alpha \in S$, has the substitution property, then consider a function $\delta_\pi : \pi \times I \longrightarrow \pi$ such that

$$\delta_\pi(B_\alpha, i) = B_{\delta(\alpha)} | \forall i \in I, \forall \alpha \in B_\alpha : \delta(\alpha, i) \subseteq B_{\delta(\alpha)}.$$

The function δ_π is the image of δ by π and results from a restriction of δ on π. Notice that we consider here the partition π as a set of block elements B_α.

The partition pair is an extension of the substitution property to two partitions. A partition pair (π, π') is an ordered pair of partitions on S such that

$$s \equiv t(\pi) \Rightarrow \delta(s, i) \equiv \delta(t, i)(\pi')\, \forall i \in I. \tag{26}$$

The set of all partition pairs is not anti-symmetric. Also, if π has the substitution property, then π satisfies the relation (26), and (π, π) is a partition pair.

4.3. Pair algebra. Consider some set of partitions L ordered by the ordering relation $\leq$, and a function δ. The subset $\Delta_\delta \subseteq L \times L$ of all the partitions pairs with respect to δ, along with the partition operations "$\cdot$" and "$+$", forms an algebra called pair algebra. If the pair (π_1, π_2) is a partition pair, then we have $(\pi_1, \pi_2) \in \Delta_\delta$.

Now, let μ and π be partitions on I and S, respectively,

$$(\mu, \pi) \in \Delta_\delta \text{ iff } i \equiv j(\mu) \Rightarrow \delta(s,i) \equiv \delta(s,j)(\pi)\, \forall s \in S, \tag{27}$$

with $i, j \in I$.

In the partition pair framework, for a given partition π the minimal operator m and the maximal operator M define respectively the smallest partition and the largest partition pairing with π.

Definition 5. Let μ be a partition on I. Then $m_\delta(\mu)$ is the minimal partition that forms a partition pair with μ, i.e., $(\mu, m_\delta(\mu)) \in \Delta_\delta$, and if $(\mu, \pi) \in \Delta_\delta$, then $m_\delta(\mu) \leq \pi$. The result $m_\delta(\mu)$ is also given by the following relation:

$$m_\delta(\mu) = \prod \{\pi_i | (\mu, \pi_i) \text{ is a partition pair}\}. \tag{28}$$

Definition 6. Let π be a partition on S. $M_\delta(\pi)$ is the maximal partition that forms a partition pair with π, i.e., $(M_\delta(\pi), \mu) \in \Delta_\delta$, and if $(\mu, \pi) \in \Delta_\delta$, then $\mu \leq M_\delta(\pi)$. The result $M_\delta(\pi)$ is also given by the following relation:

$$M_\delta(\pi) = \sum \{\mu_i | (\mu_i, \pi) \text{ is a partition pair}\}. \tag{29}$$

5. Decomposition of finite state machines

The decomposition of discrete-event models in order to determine reduced equivalent models is a popular topic. However, model decomposition with a *decoupling constraint* is not common for this type of model. In this section, a constrained decomposition methodology based on Algorithm 1 is proposed for FSMs, which are a common type of deterministic discrete-event models. FSMs are denoted by $(S, I, O, \delta, \lambda)$ for distinction from the general case. S,I,O are respectively the state set, the input set and the output set of the model. Furthermore, δ is the state function and λ is the output function.

The decomposition problem is formulated as follows: Consider an FSM $\Sigma(S, I, O, \delta, \lambda)$ with $I = I_c \cup I_\gamma \cup I_\rho$. A partial FSM Σ_* decoupled from I_γ and coupled with respect to I_ρ is investigated. The machine Σ_* is defined by the quintuple $(S_*, I_*, O_*, \delta_*, \lambda_*)$ with

- $S_* = \pi$, where π is a partition of S;
- $O_* = \pi_O$, where π_O is a partition of O;
- I_* is the input set;
- $\delta_* : \pi \times I_* \to \pi$, where δ_* is a restriction of δ;
- $\lambda_* : \pi \times I_* \to \pi_O$, where λ_* is a restriction of λ.

5.1. Decomposition constraints. In order to express coupling and decoupling constraints using partitions, a neutral element i_0 is added to I,

$$\forall s \in S : \delta(s, i_0) = s. \tag{30}$$

Hence, Σ is decoupled from the element i_0 by definition. If a block of a partition of I contains i_0, then all the elements of this block are also decoupled from Σ. Let $I_\gamma = \{a_1, a_2, \ldots\}$ and $I_\rho = \{b_1, b_2, \ldots\}$.

5.1.1. Decoupling constraint. Let us recall that in order for an FSM to be decoupled from a particular input $a \in I$, the kernel of state function δ must include this input, i.e., $a \in \ker(\delta)$. To obtain an FSM decoupled from a, the partition π of the state set S must be determined such that

$$a \in \ker(\delta_\pi), \tag{31}$$

with $\delta_p i$ being a restriction of δ on $\pi \times I$.

Consider the following partition:

$$\pi_\gamma = \{\{i_0, a_1, a_2, \ldots\}, \{i_1\}, \ldots, \{i_l\}, \{b_1\}, \{b_2\}, \ldots\}, \tag{32}$$

where i_j, with $j = 1, \ldots, l$, are elements of I_c. The partition π_γ is composed by a block regrouping all the elements of I_γ with the neutral element i_0, and singleton blocks formed from the elements of $I_c \cup I_\rho$. Using the operator m_δ and π_γ, the state set partition π^0 that is decoupled from I_γ is determined,

$$\pi^0 = m_\delta(\pi_\gamma). \tag{33}$$

To find a relationship between the partitions π^0 and π, consider a state s_1 such that $\delta(s_1, a) = s_2 \neq s_1$, $a \in I_\gamma$. By the definition of the partition π^0, $s_1 \equiv s_2(\pi^0)$. It is shown by analogy that $s_{i+1} \equiv s_i(\pi)$, where $s_{i+1} = \delta(s_i, a)$ and $i = 1, 2, \ldots, k-1$, which means that for all inputs $a \in I_\gamma$ the FSM state s remains in the same block of π^0. In other terms, we have $\delta(B^{\pi^0}, a) \subseteq B^{\pi^0}$ for some block B^{π^0} from π^0 or $I_\gamma \in \ker(\delta_{\pi^0})$, where δ_{π^0} is the restriction of δ on $\pi^0 \times I$.

By analogy, to ignore the input a, the following relationship for each block B^π from π must hold:

$$\delta(B^\pi, a) \subseteq B^\pi. \tag{34}$$

Since the operator m_δ gives the smallest partition (33), each block B^{π^0} from π^0 is included into the appropriate block B^π from π, i.e., $B^{\pi^0} \subseteq B^\pi$. Therefore

$$\pi^0 \leq \pi. \tag{35}$$

It can be shown by analogy that if $\pi^0 \leq \pi$, then each input $a \in I_\gamma$ is ignored by the partial FSM obtained using the partition π.

5.1.2. Coupling constraint. By analogy, an FSM is a coupled to the particular input $b \in I$ if the kernel of state function δ does not include this input, i.e., $b \notin \ker(\delta)$. To obtain an FSM coupled to a, the partition π of the state set S must be determined such that

$$b \notin \ker(\delta_\pi), \tag{36}$$

with $\delta_p i$ being a restriction of δ on $\pi \times I$.

Consider the partition π_ρ that decouples I_ρ,

$$\pi_\rho = \{\{i_0, b_1, b_2, \dots\}, \{i_1\}, \dots, \{i_l\}, \{a_1\}, \{a_2\}, \dots\}, \tag{37}$$

and the corresponding state set partition,

$$\bar{\pi}^0 = m_\delta(\pi_\rho). \tag{38}$$

We have previously seen that, if the machine Σ_* is decoupled from I_γ, then its state set is a partition of π^0. Accordingly, if Σ_* is coupled to I_ρ, then

$$\bar{\pi}^0 \nleq \pi. \tag{39}$$

5.2. Decomposition conditions.

5.2.1. Invariance condition. Consider the FSM Σ and a partition π which has the substitution property with respect to δ. This means that if π has the substitution property, i.e., $(\pi, \pi) \in \Delta_\delta$, then the discrete-event model described by $(\pi, I, \pi_O, \delta_*, \lambda_*)$ is a partial model of Σ and the restriction of δ on $\pi \times I$ exists (see Definition 2).

From Definitions 5 and 6, if $(\pi, \pi) \in \Delta_\delta$, then the following relations are satisfied:

$$\pi \le M_\delta(\pi) \text{ and } \pi \ge m_\delta(\pi). \tag{40}$$

For the FSM case, the relation $\pi \le M_\delta(\pi)$ implies $\pi \ge m_\delta(\pi)$ and vice versa. Thus, only one relation of (40) is required to test the invariance condition.

5.2.2. Output condition. Consider a partition π of the state set S. On the analogy of the invariance condition, if π has the substitution property, then there is a restriction of λ on $\pi \times = I \to \pi_O$. Here π_O is determined as $\pi_O = m_\lambda(\pi)$. Let $\pi_\lambda = M_\lambda(\mathbb{O})$ be a partition induced by the output function λ and the output set O on the state set. Each block of π_λ is associated with a single element of O, since $\mathbb{O}$ is a partition of singleton blocks. If $\pi \ge \pi_\lambda$ is satisfied and (π, π) is a partition pair, then all the outputs of Σ and the outputs of the partial model determined by π are bisimilar. This is obviously is the best case, but this condition is conservative.

Fortunately, to fulfil the output condition, it is sufficient to have one single bisimilar output which is not $\mathbb{I}$. This means that partitions π and π_λ must share at least one block and $\pi_\lambda + \pi \neq \mathbb{I}$. The output condition is given by

$$\pi + M_\lambda(\mathbb{O}) \neq \mathbb{I}. \tag{41}$$

5.3. Output injection for discrete-event models. If there are no partitions π satisfying the invariance condition, i.e., $(\pi, \pi) \notin \Delta_\delta$, then the loss in state information induced by the decomposition constraints is too significant. However, it is possible to compensate the information loss using the information provided by outputs. The information added is represented by a state set partition π_y such that

$$(\pi \cdot \pi_y, \pi) \in \Delta_\delta. \tag{42}$$

The relation (42) is satisfied if the following statements are true:

$$M_\delta(\pi) \ge (\pi \cdot \pi_y) \quad \text{and} \quad \pi \ge m_\delta(\pi \cdot \pi_y). \tag{43}$$

The injection mechanism is now explained. A partition π_{inj} of the output set O is determined. Here π_{inj} represents the injected outputs. Each block of π_{inj} is related via the function λ to a block of the state set partition π_y, and this relation is given by $\pi_y = M_\lambda(\pi_{inj})$. Since $\pi^0 \le \pi$, the best possible output injection π_{inj} will satisfy the relation

$$\pi^0 \cdot M_\lambda(\pi_{inj}) = \mathbb{O}. \tag{44}$$

In this case, the output injection π_{inj} completely compensates the loss in state information induced by the partitioning π^0, since the blocks of $\mathbb{O}$ are singletons and correspond on a one-on-one basis to elements of S. If a partition π_{inj} exists such that the relation (44) is satisfied, then we can use the partition π^0 as the decomposition partition since

$$\forall \pi : M(\mathbb{O}) \le \pi \wedge \pi \ge m(\mathbb{O})$$

is always satisfied by the definition of the operators m and M (see Hartmanis and Stearns, 1966).

However, if this is not possible, all π_{inj} satisfying the relation (45) are candidates to satisfy (43),

$$\pi^0 \cdot M_\lambda(\pi_{inj}) \neq \pi^0. \tag{45}$$

If multiple partitions π_{inj} are acceptable, the largest partition will guarantee the simpler partial FSM Σ_* and the smallest partition π_{inj} minimizes the information loss in the decomposition.

Finally, if the appropriate output injection is determined and the relations (43) are satisfied, then the decomposition partition π determines a partial FSM with an extended input set,

$$\Sigma_*(\pi, I_* \times \pi_{inj}, \pi_O, \delta_*, \lambda_*),$$

from some partition π_O and function λ_*.

5.4. Decomposition algorithm. Similarly to Algorithm 1, Algorithm 2 consists of three steps: The first step consists in the determination of the different elements

Algorithm 2 Decomposition algorithm for discrete-event models

Require: $\Sigma(S, I, O, \delta, \lambda)$ {Complete system}
Require: π_γ, π_λ { Decomposition constraints}
$\pi^0 = m_\delta(\pi_\gamma)$ { Decoupled state set partition }
$\bar{\pi}^0 = m_\delta(\pi_\rho)$ { Coupled state set partition }
$\pi_\lambda = M_\lambda(\mathbb{O})$ { State set partition induced by O }
{Injected outputs }
$\pi_y = M_\lambda(\pi_{inj})$
Determine π_{inj} such that $\pi^0 \cdot \pi_y = \mathbb{O}$;
{Initialization of the iterative loop}
$\xi^0 = \pi^0$, $\xi^1 = m_\delta(\xi^0 \cdot \pi_y) + \xi^0$, $i = 1$;
while $\xi^i \neq \xi^{i-1}$ **do**
$\xi^{i+1} = m_\delta(\xi^i \cdot \pi_y) + \xi^i$;
Increment i;
end while
$\pi = \xi^i$
if $\pi = \mathbb{I}$ **then**
return Invariance condition not satisfied by π
else
if $\pi + \pi_\lambda = \mathbb{I}$ **then**
Output condition not satisfied by π
else
Output condition satisfied by π
end if
if $\pi \geq \bar{\pi}^0$ **then**
Coupling constraint not satisfied by π
else
Coupling constraint satisfied by π
end if
$S_* = \pi$;
$I_* = (I_c \cup I_\rho \times \pi_{inj})$;
$O_* = \pi_O = m_\lambda(\pi)$;
Determine δ_* restriction of δ on $\pi \times I_* \to \pi$
Determine λ_* restriction of λ on $\pi \times I_* \to \pi_O$
{Decoupled partial FSM}
return $\Sigma_*(S_*, I_*, O_*, \delta_*, \lambda_*)$
end if

needed in the calculus, i.e., the decoupled state set partition π^0 for the initialisation of the iterative loop, the coupled state set partition π_λ to check the coupling constraint and the state set partition induced by the output set π_λ to check the output condition. Also, the partition π_{inj} is computed to obtain the outputs to be injected. The second step of the algorithm is the iterative loop to obtain the invariant decoupled partition π, which is the basis of the partial model to be obtained. The loop is initialized in its first step by taking $\xi^0 = \pi^0$ and is based on the two main conditions for the partition π: $\pi^0 \leq \pi$ and $m_\delta(\pi \cdot \pi_y) \leq \pi$.

Finally, the resulting partition π is checked for the coupling constraint and the output condition, and if both conditions are satisfied, the decoupled partial FSM is built.

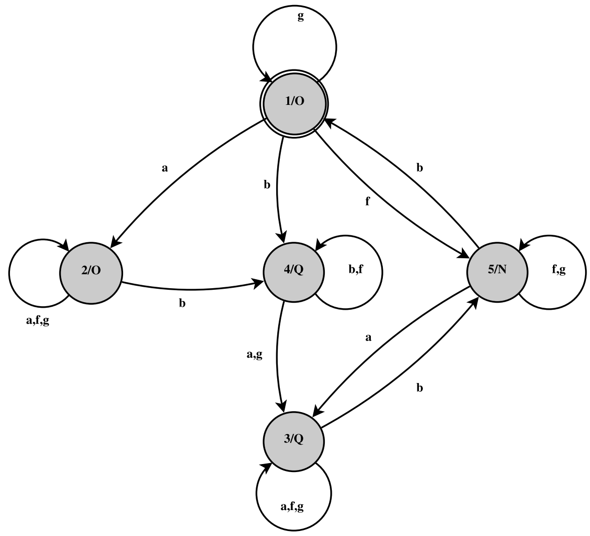


Fig. 1. FSM Σ.

Remark 4. If $\pi^0 \cdot \pi_y = \mathbb{O}$, then the iterative loop in Algorithm 2 is skipped and $\pi = \pi^0$. This is possible because $m_\delta(\pi \cdot \pi_y) = m_\delta(\mathbb{O}) = \mathbb{O}$ and $\pi^0 + \mathbb{O} = \pi^0$.

6. Illustration

Consider the FSM Σ, assumed to represent some real-world process and described by Fig. 1 and Table 1.

Table 1. Transition table of the model Σ.

	a	b	f	g	o
1	2	4	5	1	O
2	2	4	2	2	O
3	3	5	3	3	Q
4	3	4	4	3	Q
5	3	1	5	5	N

Σ is a five-state model, with two known inputs, two unknown inputs f and g and three outputs $\{O, Q, N\}$. The initial state is 1. Here g represents the fault to be detected and f the event to be ignored. Therefore, Σ is going to be decomposed in order to obtain the partial model decoupled from $I_\gamma = \{f\}$ and coupled to $I_\rho = \{g\}$.

The first step requires computation of the decoupling partition π^0. The input set partition decoupled from I_γ is given by

$$\pi_\gamma = \{\{i_0, f\}, \{a\}, \{b\}, \{g\}\}.$$

The corresponding state set partition is given by

$$\pi^0 = m_\delta(\pi_\gamma) = \{\{1, 5\}, \{2\}, \{3\}, \{4\}\}.$$

The smallest partition π which fulfills the invariance condition is obtained by iteration:

$$\begin{aligned} \xi^0 &= \pi^0, \\ \xi^1 &= \xi^0 + m_\delta(\xi^0) = \{\{1,4,5\},\{2,3\}\} \neq \xi^0, \\ \xi^2 &= \xi^1 + m_\delta(\xi^1) = \{\{1,2,3,4,5\}\} \neq \xi^1, \\ \xi^3 &= \xi^2 + m_\delta(\xi^2) = \{\{1,2,3,4,5\}\} = \xi^2. \end{aligned}$$

Since $\xi^3 = \xi^2$, we have $\pi = \{\{1,2,3,4,5\}\} = \mathbb{I}$. In this case, the decomposition is impossible with the decoupling constraints from π_γ.

A solution may be obtained using output injection. There are three outputs generated by Σ: $\{O,Q,N\}$. We do not need to inject all the outputs. The outputs to be injected are taken from the partition π_{inj} of $O = \{O,Q,N\}$ such that

$$\pi^0 \cdot \pi_y = \mathbb{O}$$

with $\pi_y = M_\lambda(\pi_{inj})$. Two partitions of outputs are possible: $\{\{O,Q\},\{N\}\}$ and $\{\{O\},\{Q,N\}\}$. We choose the first one, $\pi_{inj} = \{\{O,Q\},\{N\}\}$, since the two possible partitions have two blocks. In the general case, the output partition with fewer blocks should be preferred in order to obtain a simpler partial model. The decomposition algorithm is resumed in the π determination step. The smallest partition π which fulfils the invariance condition with output injection is obtained by the following iteration:

$$\begin{aligned} \xi^0 &= \pi^0, \\ \xi^1 &= \xi^0 + m_\delta(\xi^0 \cdot M_\lambda(\pi_{inj})) \\ &= \{\{1,5\},\{2\},\{3\},\{4\}\} = \xi^0. \end{aligned}$$

Since $\xi^1 = \xi^0$, we get $\pi = \xi^1$. The decomposition with a decoupling constraint is possible using π.

The verification step consists in testing the output condition and the coupling to $I_\rho = \{g\}$. The state set partition induced by the output is obtained by

$$\pi_\lambda = \{\{1,2\},\{3,4\},\{5\}\}.$$

The output condition is fulfilled by π since

$$\pi + \pi_\lambda = \{\{1,2,5\},\{3,4\}\} \neq \mathbb{I}.$$

To test the coupling constraint, the partition π_ρ which decouples I_ρ is calculated:

$$\pi_\rho = \{\{i_0, g\},\{a\},\{b\},\{f\}\}.$$

The corresponding state set partition is given by

$$\bar{\pi}^0 = m_\delta(\pi_\rho) = \{\{1\},\{2\},\{3,4\},\{5\}\}.$$

Coupling constraint is fulfilled because

$$\bar{\pi}^0 \nleq \pi.$$

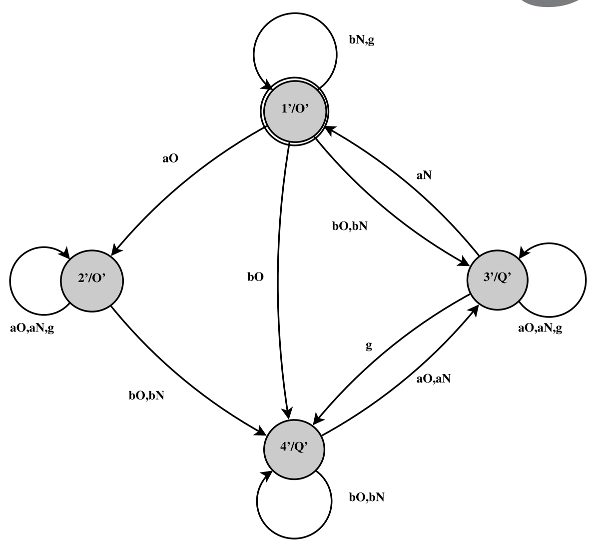


Fig. 2. FSM Σ_g.

Finally, the partial FSM Σ_g is determined using the decomposition partition π. The input set of the partial machine is given by

$$\begin{aligned} I_* = &\{aO = \{a,\{O,Q\}\}, bO = \{b,\{O,Q\}\}, \\ &\ldots, aN = \{a,\{N\}\}, bN = \{b,\{N\}\}\}. \end{aligned}$$

The state set is given by

$$S_* = \{1' = \{1,5\}, 2' = \{2\}, 3' = \{3\}, 4' = \{4\}\},$$

and the output set is

$$O_* = \{O' = \{O,N\}, Q' = \{Q\}\}.$$

The state function δ_* and the output function λ_* are shown in Table 2.

Table 2. Transition table of Σ_g.

	aO	aN	bO	bN	g	o_*
1'	2'	3'	4'	1'	1'	O'
2'	2'	2'	4'	4'	2'	O'
3'	3'	3'	1'	1'	3'	Q'
4'	3'	3'	4'	4'	3'	Q'

Figure 2 shows the transition graph of the resulting partial model.

The output value O' of Σ_* is equivalent to both output values O or N for Σ, and the output value Q' is equivalent to Q. For example, if the current output of Σ is O or Q and the output of Σ_* is O', then the outputs are consistent.

6.1. Simulations.

Simulation results are provided here. The model Σ is excited by two sequences of known and unknown inputs; the first one contains several occurrences of the unknown input f and the second one contains occurrences of the unknown input g. In this example, the outputs of Σ represent the measured outputs of the process to be monitored. Sequences composed of known inputs (a, b) of seq_1 and seq_2 combined with outputs from Σ are injected into the decoupled partial model Σ_*. Outputs are compared and a discrepancy indicator sequence is computed. The analysis of the discrepancy indicator sequence permits detecting the event g.

6.1.1. Input sequence containing *f*. The first injected sequence is given by

$$seq_1 = [a, b, a, b, b, f, a, b, a, b, b, f].$$

Outputs of Σ and Σ_* are shown in Fig. 3.

Outputs of Σ and Σ_* remain consistent even if event f occurs. Simulations confirm that Σ_* is decoupled from event f.

6.1.2. Input sequence containing *g*. The second injected sequence is given by

$$seq_2 = [a, b, g, a, b, b, a, b, g, a, b, b].$$

Outputs of Σ and Σ_* are shown in Fig. 4.

Outputs of Σ and Σ_* remain consistent until the first occurrence of event g, after which they become inconsistent after a slight delay. The delay occurs because the event g is weakly detectable (Sampath *et al.*, 1995). Simulations confirm that Σ_* is coupled to event g, and will react to every occurrence of this event.

7. Conclusion

In this paper, the decoupling of deterministic behavioural models was addressed. An algebraic formulation of the problem and of the solution was presented, based on previous work on continuous-time model decoupling (Berdjag *et al.*, 2006c). This general formulation permits addressing all types of deterministic models. The decomposition algorithm is then applied to a particular problem: the constrained decomposition of FSMs. It is important to notice that the algebraic formalism used to implement the decomposition (Algorithm 1) remains the same in the case of continuous-time models (Berdjag *et al.*, 2006b).

The first contribution is the introduction of decoupling constraints in the FSM decomposition. The resulting decoupled partial model can be used to detect unexpected events in a process using a discrete-event model. Another contribution is the use of the *output injection* technique to extend the invariance condition in the decomposition methodology. The authors' future work addresses the decomposition of mixed dynamic models known as hybrid models.

Acknowledgment

This work was supported in part by the Russian Foundation of Basic Research (Grants 07-08-00102 and 07-08-92101).

References

Bavishi, S. and Chong, E. (1994). Automated fault diagnosis using a discrete event systems framework, *IEEE Symposium on Intelligent Control, Columbus, OH, USA*, pp. 213–218.

Benveniste, A., Fabre, E., Haar, S. and Jard, C. (2003). Diagnosis of asynchronous discrete event systems: A net unfolding approach, *IEEE Transactions of Automatic Control* **48**(5): 714–727.

Berdjag, D., Christophe, C. and Cocquempot, V. (2006a). An algebraic method for nonlinear system decomposition, *6th IFAC Symposium on Fault Detection Supervision ans Safety for Technical Processes, SAFEPROCESS'2006, Beijing, China*, pp. 42–53.

Berdjag, D., Christophe, C. and Cocquempot, V. (2006b). Nonlinear model decomposition for fault detection and isolation system design, *45th IEEE Conference on Decision and Control, San Diego, CA, USA*, pp. 3321–3326.

Berdjag, D., Christophe, C., Cocquempot, V. and Jiang, B. (2006c). Nonlinear model decomposition for robust fault detection and isolation using algebraic tools, *International Journal of Innovative Computing, Information and Control* **2**(6): 1337–1353.

Blanke, M., Kinnaert, M., Lunze, J. and Staroswiecki, M. (2003). *Diagnosis and Fault-Tolerant Control*, Springer, Berlin.

Boel, R. and Jiroveanu, G. (2004). Distributed contextual diagnosis for very large systems, *International Workshop on Discrete Event Systems, Reims, France*, pp. 343–348.

Boubour, R., Jard, C., Aghasaryan, A., Fabre, E. and Benveniste, A. (1997). A Petri net approach to fault detection and diagnosis in distributed systems (Parts 1 and 2), *IEEE 36th International Conference on Decision and Control, San Diego, CA, USA*, pp. 720–731.

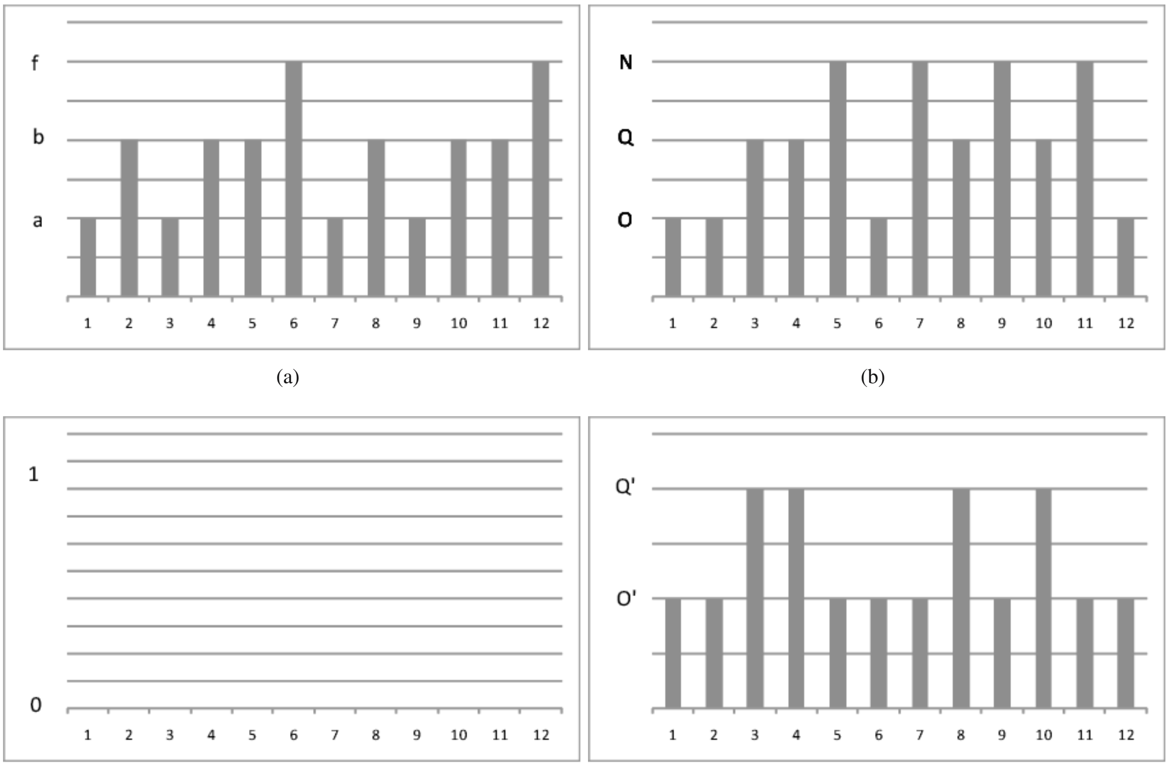
Chow, E. and Willsky, A. (1984). Analytical redundancy and the design of robust failure detection systems, *IEEE Transactions on Automatic Control* **29**(7): 603–614.

Cox, D., Little, J. and O'Shea, D. (1991). *Ideals, Varieties, and Algorithms*, Springer-Verlag, New York, NY.

Diop, S. (1991). Elimination in control theory, *Mathematics of Control, Signals, and Systems* **4**: 17–32.

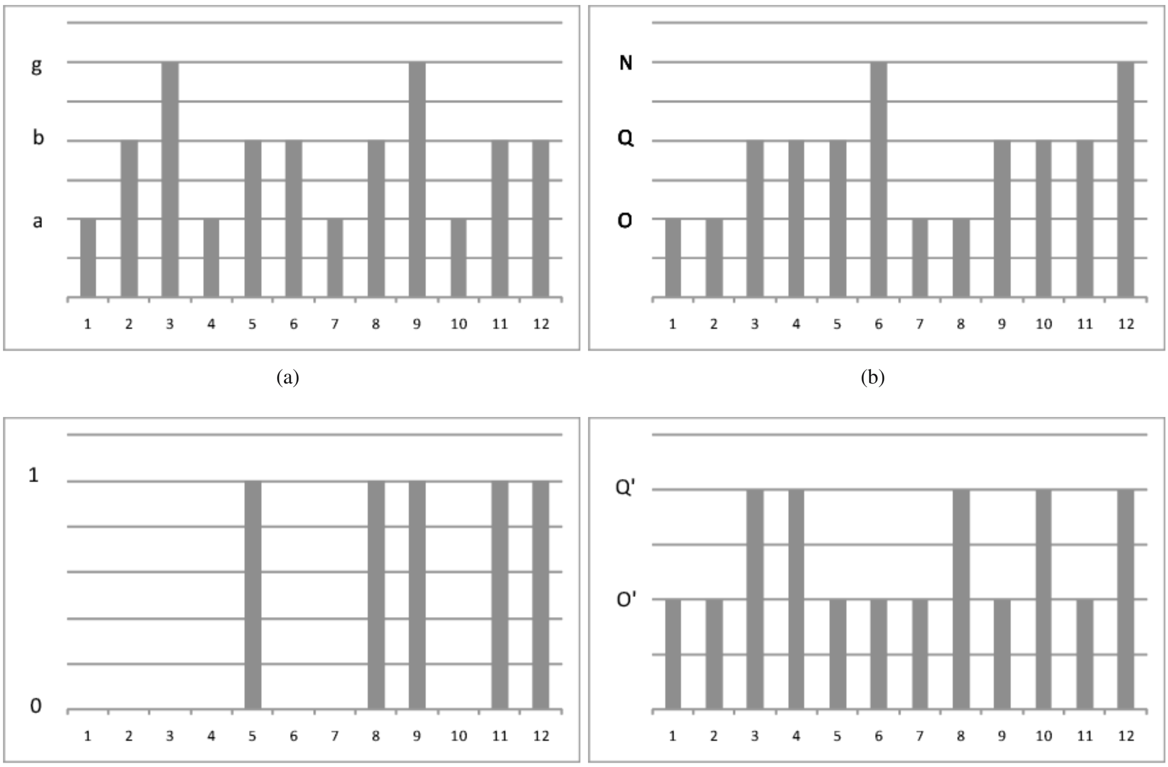
Fliess, M. and Join, C. (2003). An algebraic approach to fault diagnosis for linear systems, *Proceedings of the International Conference on Computational Engineering in System Applications, CESA, Lille, France*, pp. 1–9.

Fliess, M., Join, C. and Sira-Ramírez, H. (2004). Robust residual generation for linear fault diagnosis: An algebraic setting with examples, *International Journal of Control* **77**(20): 1223–1242.



(a) (b) (c) (d)

Fig. 3. Simulations for the first sequence: input sequence (a), Σ output (b), discrepancy indicator (c), Σ_* output (d).



(a) (b) (c) (d)

Fig. 4. Simulations for the second sequence: input sequence (a), Σ output (b), discrepancy indicator (c), Σ_* output (d).

Gertler, J. (1991). Analytical redundancy methods in fault detection and isolation—Survey and synthesis, *Proceedings of the 1st IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS'91, Baden Baden, Germany*, Vol. 1, pp. 9–21.

Gertler, J.J. (1998). *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, New York, NY.

Giua, A. (1997). Petri net state estimators based on event observation, *Proceedings of the 36th IEEE International Conference on Decision and Control, San Diego, CA, USA*, pp. 4086–4091.

Hadjicostis, C. and Verghese, G. (1999). Monitoring discrete event systems using Petri net embeddings, *in* S. Donatelli and J. Kleijn (Eds.), *Application and Theory of Petri Nets*, Lecture Notes in Computer Science, Vol. 1639, Springer Verlag, Berlin/Heidelberg, pp. 188–207, DOI: 10.1007/3-540-48745-X_12.

Hammouri, H., Kinnaert, M. and El Yaagoubi, E. (2001). A geometric approach to fault detection and isolation for bilinear systems, *IEEE Transactions on Automatic Control* **46**(9): 1451–1455.

Hamscher, W., Console, L. and Kleer, J.D. (1992). *Readings in Model-Based Diagnosis*, Morgan Kaufmann Publishers, San Mateo, CA.

Hartmanis, J. and Stearns, R. (1966). *The Algebraic Structure Theory of Sequential Machines*, Prentice-Hall, New York, NY.

Hillston, J. (1996). *A Compositional Approach to Performance Modelling*, Cambridge University Press, Cambridge.

Isermann, R. (1984). Process fault-detection based on modelling and estimation methods—A survey, *Automatica* **20**(4): 387–404.

Isermann, R. (2005). Model-based filt detection and analysis—Status and application, *Annual Reviews in Control* **29**(1): 71–85.

Isermann, R. and Freyermuth, B. (1991). Process fault diagnosis based on process model knowledge, Part 1: Principles for fault diagnosis with parameter estimation, *Transactions of the American Society of Mechanical Engineers* **113**(4): 620–626.

Isidori, A. (1995). *Nonlinear Control Systems,* 3rd Edn., Springer-Verlag, Berlin.

Jiang, B., Staroswiecki, M. and Cocquempot, V. (2004). Fault diagnosis based on adaptive observer for a class of nonlinear systems with unknown parameters, *International Journal of Control* **77**(4): 415–426.

Jiang, B., Staroswiecki, M. and Cocquempot, V. (2006). Fault accommodation for nonlinear dynamic systems, *IEEE Transactions on Automatic Control* **51**(9): 1578–1583.

Kinnaert, M. (1999). Robust fault detection based on observers for bilinear systems, *Automatica* **35**(11): 1829–1842.

Lafortune, S., Teneketzis, D., Sengupta, R., Sampath, M. and Sinnamohideen, K. (2001). Failure diagnosis of dynamic systems: An approach based on discrete event systems, *Proceedings of the American Control Conference, Arlington, VA, USA,* pp. 2058–2071.

Lefebvre, D. (1999). Failure detection and isolation for manufacturing systems, *Revue internationale d'ingenierie des systemes de production mecanique* **2**: V.33–V.44.

Leuschen, M., Walker, I. and Cavallaro, J. (2005). Fault residual generation via nonlinear analytical redundancy, *IEEE Transactions on Control Systems Technology* **13**(3): 452–458.

Lin, F. (1994). Diagnosability of discrete event systems and its applications, *Discrete Event Dynamic Systems* **4**(2): 197–212, DOI: 10.1007/BF01441211.

Lootsma, T. (2001). *Observer-based Fault Detection and Isolation for Nonlinear Systems*, Ph.D. thesis, Aalborg University, Aalborg.

Maquin, D., Cocquempot, V., Cassar, J., Staroswiecki, M. and Ragot, J. (1997). Generation of analytical redundancy relations for FDI purposes, *IEEE International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEMPED'97, Carry-le Rouet, France,* pp. 270–276.

Maquin, D., Luong, M. and Ragot, J. (1997). Fault detection and isolation and sensor network design, *Journal européen des systèmes automatisés* **31**(2): 393–406.

Patton, R. (1994). Robust model-based fault diagnosis: The state of the art, *Proceedings of the 2nd IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS'94, Espoo, Finland*, Vol. 1, pp. 1–24.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. and Teneketzis, D. (1995). Diagnosability of discrete-event systems, *IEEE Transactions on Automatic Control* **40**(9): 1555–1575.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. and Teneketzis, D. (1996). Failure diagnosis using discrete-event models, *IEEE Transactions on Control Systems Technology* **4**(2): 105–124.

Shumsky, A. (1991). Fault isolation in nonlinear dynamic systems by functional diagnosis, *Automation and Remote Control* **12**: 148–155.

Shumsky, A. (2007). Redundancy relations for fault diagnosis in nonlinear uncertain systems, *International Journal of Applied Mathematics and Computer Science* **17**(4): 477–489, DOI: 10.2478/v10006-007-0040-1.

Shumsky, A. and Zhirabok, A. (2006). Nonlinear diagnostic filter design: Algebraic and geometric points of view, *International Journal of Applied Mathematics and Computer Science* **16**(1): 115–127.

Staroswiecki, M. and Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems, *Automatica* **37**(5): 687–699.

Vereshchagin, N. and Shen, A. (2002). *Basic Set Theory*, Student Mathematical Library, Vol 17, American Mathematical Society, Providence, RI.

Zad, H., Kwong, R. and Wonham, W. (2003). Fault diagnosis in discrete-event systems: Framework and model reduction, *IEEE Transactions on Automatic Control* **48**(7): 1199–1212.

Zad, S.H. (1999). *Fault Diagnosis in Discrete-event and Hybrid Systems*, Ph.D. thesis, University of Toronto, Toronto.

Zhirabok, A. (2006). Nonlinear dynamic systems: Their canonical decomposition based on invariant functions, *Automation and Remote Control* **67**(4): 517–528.

Zhirabok, A. and Shumsky, A. (1993). A new mathematical techniques for nonlinear systems research, *Proceedings of the 12th IFAC World Congress, Sydney, Australia*, pp. 485–488.

Denis Berdjag was born in Donetsk (Ukraine) in 1976. He obtained the engineering degree from the University of Sétif (Algeria) in 1999, the Master's degree and the Ph.D. degree in automatic control, respectively, in 2003 and 2007 from the University of Science and Technology of Lille (France). He is currently an associate professor of automatic control at Valenciennes University (France) in the LAMIH (Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines) research group. His research addresses fault detection, isolation and diagnosis of systems (nonlinear, hybrid or human machine systems).

Vincent Cocquempot received the Ph.D. degree in automatic control from the Lille University of Sciences and Technologies in 1993. He is currently a full professor of automatic control and computer science at Institut Universitaire de Technologies de Lille, France. He is a researcher in LAGIS-CNRS FRE3303: Automatic Control, Computer Science and Signal Processing Laboratory of Lille 1 University and the head of the team on fault tolerant systems of this laboratory. His research interests include robust on-line fault detection and isolation for uncertain dynamical nonlinear systems and fault tolerant control for hybrid dynamical systems.

Cyrille Christophe received the Ph.D. degree in automatic control from the Lille University of Sciences and Technologies in 2001. He is currently an associate professor of automatic control and computer science at Institut Universitaire de Technologie de Lille, France. He is a researcher in LAGIS-CNRS FRE3303: Automatic Control, Computer Science and Signal Processing Laboratory of Lille. His main research interest is fault detection and isolation of uncertain dynamical nonlinear and hybrid systems.

Alexey Shumsky is a professor at Pacific State Economic University and a chief researcher at the Institute of Marine Technologies Problems, Russian Academy of Sciences (both in Vladivostok). He received his Candidate of Sciences (Ph.D.) degree in radiolocation and radionavigation from Leningrad (St. Petersburg) Electrotechnical Institute in 1985 and the D.Sc. degree in automatic control from the Institute of Control Problems, Russian Academy of Sciences (Moscow), in 1996. He serves as an associate editor for the *International Journal of Applied Mathematics and Computer Science*. His research interests include nonlinear control theory with application to fault diagnosis and fault tolerant control.

Alexey Zhirabok is a professor at the Far Eastern State Technical University (FESTU), Vladivostok. He received the Candidate of Science (Ph.D.) degree in electrical engineering from Leningrad (currently St. Petersburg) Electrotechnical Institute in 1978 and the D.Sc. degree in control engineering from the Russian Academy of Sciences, Vladivostok, in 1996. He is currently the head of the Department of Radio Design and Technology of the FESTU and the main scientific secretary of the Far Eastern Branch of the Russian Academy of Engineering. His current research interests are nonlinear system theory and fault detection and isolation in nonlinear systems. He has authored about 180 scientific publications (five books, seven booklets).

Appendix

In the following, S, I denotes sets, s, t are elements of S and π_i, $i \in \mathbb{N}$ are partitions of S. The set of all partitions of the set S is denoted by Π_S. Furthermore, δ is a function defined by $\delta : S \times I \longrightarrow S$ and Δ_δ is a pair algebra.

Partitions. A partition π of a set S is a collection of complementary disjoint subsets of S. Elements of π are called *blocks.*

Example 1. Consider the set $S = \{1, 2, 3, 4\}$, and the partitions π_1, π_2 and π_3 on the set S such that

$$\pi_1 = \{\{1, 2\}, \{3\}, \{4\}\}, \quad \pi_2 = \{\{1\}, \{2\}, \{3, 4\}\},$$
$$\pi_3 = \{\{1, 2\}, \{3, 4\}\}.$$

The blocks of these partitions are

$$\begin{aligned} \pi_1 :& B_1^{\pi_1} = \{1, 2\}, && B_2^{\pi_1} = \{1, 2\}, \\ & B_3^{\pi_1} = \{3\}, && B_4^{\pi_1} = \{4\}, \\ \pi_2 :& B_1^{\pi_2} = \{1\}, && B_2^{\pi_2} = \{2\}, \\ & B_3^{\pi_2} = \{3, 4\}, && B_4^{\pi_2} = \{3, 4\}, \\ \pi_3 :& B_1^{\pi_3} = \{1, 2\}, && B_2^{\pi_3} = \{1, 2\}, \\ & B_3^{\pi_3} = \{3, 4\}, && B_4^{\pi_3} = \{3, 4\}. \end{aligned}$$

Partition multiplication. Multiplication of partitions denoted with $\pi_1 \cdot \pi_2$ is defined by the following relation:

$$s \equiv t(\pi_1 \cdot \pi_2) \Longleftrightarrow s \equiv t(\pi_1) \wedge s \equiv t(\pi_2).$$

Blocks of $\pi_1 \cdot \pi_2$ are determined using

$$B_s^{\pi_1 \cdot \pi_2} = B_s^{\pi_2} \cap B_s^{\pi_2}.$$

Example 2. Consider the set $S = \{0, 1, 2, 3, 4, 5, 6\}$ and the partitions

$$\pi_1 = \{\{0, 1, 2\}, \{3\}, \{4, 5\}, \{6\}\},$$
$$\pi_2 = \{\{0\}, \{1, 2, 3\}, \{4\}, \{5\}, \{6\}\}.$$

The partition $(\pi_1 \cdot \pi_2)$ is given by

$$\pi_1 \cdot \pi_2 = \{\{0\}, \{1, 2\}, \{3\}, \{4\}, \{5\}, \{6\}\}.$$

Partition addition. The addition of two partitions π_1 and π_2 is defined by

$$s \equiv t(\pi_1 + \pi_2) \Leftrightarrow \exists s_0 = s, s_1, \dots, s_{n-1}, s_n = t : s_i \equiv s_{i+1}(\pi_1) \vee s_i \equiv s_{i+1}(\pi_2) \quad 0 \le i \le n-1.$$

The blocks of $\pi_1 + \pi_2$ are determined as follows: all the blocks of the two partitions containing the same element are combined, i.e.,

$$B_s^{\pi_1+\pi_2}(1) = B_s^{\pi_1} \cup B_s^{\pi_2},$$

and for all $i > 1$,

$$B_s^{\pi_1+\pi_2}(i+1) = B_s^{\pi_1+\pi_2}(i) \cup \{B | (B \in \pi_1 \vee B \in \pi_2) \wedge B \cap B_s^{\pi_1+\pi_2}(i) \neq \emptyset .\}$$

Example 3. Considering S, π_1, π_2 of Example 2,

$$\pi_1 + \pi_2 = \{\{0, 1, 2, 3\}, \{4, 5\}, \{6\}\}.$$

Partial ordering relation. Consider the relation $\le$ on the set of all possible partitions on S denoted by Π_S such that

$$\pi_1 \le \pi_2 \Leftrightarrow \begin{cases} \pi_1 \cdot \pi_2 & = \pi_1, \\ \pi_1 + \pi_2 & = \pi_2. \end{cases}$$

It can be shown (see Hartmanis and Stearns, 1966) that $\le$ is a partial ordering on the set Π_S. Also, the partitions

$$\mathbb{I} = \{\{S\}\}, \quad \mathbb{O} = \{\{s\} | \forall s \in S\}$$

are respectively the largest and the smallest partitions from Π_S, i.e.,

$$\forall \pi \in \Pi_S : \pi \le \mathbb{I} \wedge \mathbb{O} \le \pi.$$

Example 4. Consider the set S and the partitions π_1, π_2, π_3 from Example 1. The following relations are true:

$$\mathbb{I} = \{\{1, 2, 3, 4\}\}, \quad \mathbb{O} = \{\{1\}, \{2\}, \{3\}, \{4\}\},$$

$$\pi_1 \le \pi_3, \quad \pi_2 \le \pi_3, \quad \pi_1 \not\le \pi_2, \quad \pi_2 \not\le \pi_1,$$
$$\pi_1 \le \mathbb{I}, \quad \pi_2 \le \mathbb{I}, \quad \pi_3 \le \mathbb{I}, \quad \mathbb{O} \le \pi_1,$$
$$\mathbb{O} \le \pi_2, \quad \mathbb{O} \le \pi_3.$$

Substitution property and pair algebra. Let π be a partition on S. The partition π is said to have the *substitution property* with respect to the function δ if and only if

$$s \equiv t(\pi) \Rightarrow \delta(s, i) \equiv \delta(t, i)(\pi) \, \forall i \in I.$$

A partition pair of two partitions π_1 and π_2 , (π_1, π_2) is defined by

$$s \equiv t(\pi_1) \Rightarrow \delta(s, i) \equiv \delta(t, i)(\pi_2) \, \forall i \in I.$$

Example 5. Consider two sets $S = \{1, 2, 3, 4, 5, 6\}$ and $I = \{a, b\}$. Let δ be a function described by Table 3.

Table 3. Transition tables of functions δ and δ'.

δ	a	b
1	5	3
2	1	5
3	4	6
4	6	2
5	3	4
6	2	1

δ'	a	b
1'={1,3,6}	2'	1'
2'={2,4,5}	1'	2'

The partition $\pi_1 = \{\{1, 3, 6\}, \{2, 4, 5\}\}$ has the substitution property with respect to δ. The image of δ by π is the function δ_π described by Table 3. Notice that π_1 is not unique, for example, $\pi_2 = \{\{1, 4\}, \{2, 3\}, \{5, 6\}\}$ has also the substitution property with respect to δ.

Since $\pi_1$and π_2 have the substitution property, we deduce that $(\pi_1, \pi_1) \in \Delta_\delta$, $(\pi_2, \pi_2) \in \Delta_\delta$ are partition pairs. Also, the partitions $\pi_3 = \{\{1, 2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$ and $\pi_4 = \{\{1, 3, 5\}, \{2\}, \{4\}, \{6\}\}$ form a partition pair with respect to δ, i.e., $(\pi_3, \pi_4) \in \Delta_\delta$.

Example 6. Examples of computations of operators m and M are given below. The computation method can be found in the work of Hartmanis and Stearns (1966).

Consider the sets S, I and the function δ from Example 5.

$$m(\{\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}\}) = \{\{1, 3\}, \{2, 5\}, \{4\}, \{6\}\},$$

$$m(\{\{1, 4\}, \{2\}, \{3\}, \{5\}, \{6\}\}) = \{\{1\}, \{2, 3\}, \{4\}, \{5, 6\}\},$$

$$m(\{\{1, 4, 6\}, \{2\}, \{3\}, \{5\}\}) = \{\{1, 2, 3, 5, 6\}, \{4\}\},$$

$$M(\{\{1, 6\}, \{2\}, \{3\}, \{4\}, \{5\}\}) = \mathbb{O}\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\},$$

$$M(\{\{1,4\},\{2\},\{3\},\{5,6\}\})$$
$$= \{\{1,4\},\{2,3\},\{5,6\}\},$$

$$M(\{\{1,4,6\},\{2,3,5\}\})$$
$$= \{\{1\},\{2,4\},\{3\},\{5,6\}\}.$$

Received: 2 December 2009
Revised: 19 July 2010