Fault-Hiding Control Reconfiguration for a Class of Discrete Event Systems

Th. Wittmann * **J. H. Richter** ** **T. Moor** *

* *Friedrich-Alexander Universität Erlangen-Nürnberg*
** *Siemens AG, I IA ATS 43, Nürnberg*

Abstract: Fault-hiding control reconfiguration aims at hiding a fault from the nominal controller while the reconfigured closed-loop system possesses admissible behaviour. The necessary degrees of freedom are created by placing a reconfiguration block between nominal controller and faulty plant. We aim at a guaranteed non-conflicting, complete and controllable behaviour of the self-reconfiguring closed-loop system, in particular for an *arbitrary* solution to the nominal control problem. Thereby, the nominal controller design and the design of the reconfiguration block are completely decoupled. This is desirable from a practical perspective, since in this way additional fault-tolerant control capabilities can be retrofit to an existing control system. In this paper, we propose a self-reconfiguring control architecture, state our reconfiguration problem in terms of finite languages and address the synthesis of discrete event dynamic reconfiguration blocks. To illustrate our results, we provide a running example.

Keywords: Discrete Event Systems, Fault-Tolerant Control, Control Reconfiguration, Fault-Hiding

1. INTRODUCTION

Industrial PLCs are programmed by experts in practice, and this is not expected to change in the near future. In this paper we thus introduce a concept for fault-tolerant control of discrete event processes that can coexist with manually programmed PLCs and that can be retrofit to existing controlled plants.

Inspired by the fault-hiding principle, we place a reconfiguration block between controller and faulty plant. Thereby, we influence the signals in the nominal closed-loop system in order to hide a fault from the controller. The reconfiguration block shall be inactive during nominal operation and take over control only after fault occurrences. Motivated from industrial applications, where the actual controller implementation is unknown, we demand the reconfiguration block to be functional for an *arbitrary* but specification-conforming controller.

Much research in recent years has focused on control reconfiguration, see [Blanke et al., 2006] and subsequent literature. In [Schmidt, 2012] control reconfiguration is considered in the context of reconfigurable machine tools. Fault-tolerant control based on control reconfiguration is reported in [Paoli et al., 2008, 2011, Nke and Lunze, 2013]. The fault-hiding principle was originally developed for linear continuous-time system [Lunze and Steffen, 2006] and later extended to selected classes of non-linear continuous-time systems [Richter et al., 2011]. However, to the authors best knowledge there is no approach to fault-hiding control reconfiguration in discrete event systems, which is our main contribution.

Our approach relies on the supervisory control theory (SCT), as proposed in [Ramadge and Wonham, 1987, 1989]. In this paper the nominal controller is a solution to a control problem under partial observation [Lin and Wonham, 1988], which we extend by an operator interface. Thereby, integrating a reconfiguration block to hierarchically structured control systems is simplified. Furthermore, we use fault-accommodating control [Wittmann et al., 2012] to address sensor, actuator and plant faults, thus we do not rely on a separate diagnosis or controller switching mechanism. Our design goals w.r.t. the self-reconfiguring closed-loop system are non-conflictingness, completeness, controllability and specification conformance. We set up a suitable candidate for the closed-loop system and thereof we extract an admissible reconfiguration block by projection.

Our main results consist in sufficient conditions for the synthesis of a reconfiguration block that ensures non-conflictingness, completeness, controllability and the conformance with the design specifications for an arbitrary nominal controller.

The remainder of this paper is structured as follows. A concise revision of formal languages and SCT is provided in Section 2. In Section 3 we state the nominal control problem and recall fault-accommodating control. In Section 4 we address fault-hiding control reconfiguration in discrete event systems, and illustrate our results using an example.

2. PRELIMINARIES

Let Σ be a *finite alphabet*, i.e., a finite set of symbols (or *events*) $\sigma \in \Sigma$. A finite sequence of symbols $\sigma_i \in \Sigma$, $i \leq n$ is called a *string* $s = \sigma_1\sigma_2 \ldots \sigma_n$, where $n \in \mathbb{N}$. The length of a string $s \in \Sigma^*$ is denoted $|s| \in \mathbb{N}_0$. The *empty string* ϵ is characterised by $|\epsilon| = 0$. The *Kleene-closure* Σ^* is the

set of all finite strings over the alphabet Σ including the empty string.

If, for two strings $s, r \in \Sigma^*$, there exists $t \in \Sigma^*$ such that $s = rt$, we say r is a *prefix* of s, and write $r \leq s$. A *formal language* (or short a *language*) $L \subseteq \Sigma^*$ over Σ, is a subset of the Kleene-closure Σ^*. The *prefix* of a language L is defined by $\operatorname{pre} L := \{r \in \Sigma^* \mid \exists s \in L : r \leq s\}$. A language L is *prefix-closed* if $L = \operatorname{pre} L$.

The *natural projection* $\mathrm{p}_\circ \colon \Sigma^* \to \Sigma_\circ^*$ with $\Sigma_\circ \subseteq \Sigma$, is defined iteratively: (1) let $\mathrm{p}_\circ \epsilon = \epsilon$; (2) for an arbitrary $s \in \Sigma^*$ and $\sigma \in \Sigma$, let $\mathrm{p}_\circ(s\sigma) = \mathrm{p}_\circ(s)\sigma$ if $\sigma \in \Sigma_\circ$, or, if $\sigma \notin \Sigma_\circ$, then $\mathrm{p}_\circ(s\sigma) = \mathrm{p}_\circ(s)$. The set-valued inverse $\mathrm{p}_\circ^{-1}$ of $\mathrm{p}_\circ$ is defined by $\mathrm{p}_\circ^{-1}(r) = \{s \in \Sigma^* \mid \mathrm{p}_\circ(s) = r\}$ for $r \in \Sigma_\circ^*$. We use the convention that projections and inverse projections are denoted p_- and p_-^{-1}, respectively, with a subscript to indicate the respective range and domain. When extended to languages, the projection distributes over unions, and the inverse projection distributes over unions and intersections.

The *synchronous composition* of two languages $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$ is defined by $L_1 \parallel L_2 := (\mathrm{p}_1^{-1} L_1) \cap (\mathrm{p}_2^{-1} L_2)$, where p_1 and p_2 denote the natural projections from $\Sigma = (\Sigma_1 \cup \Sigma_2)^*$ to Σ_1^* and Σ_2^*, respectively.

Given two languages L, $K \subseteq \Sigma^*$, and a set of uncontrollable events $\Sigma_{\mathrm{uc}} \subseteq \Sigma$, we say K is *controllable* w.r.t. $(L, \Sigma_{\mathrm{uc}})$, if $\operatorname{pre} K \Sigma_{\mathrm{uc}} \cap (\operatorname{pre} L) \subseteq \operatorname{pre} K$ and *relatively closed* w.r.t. L if $K = \operatorname{pre} K \cap L$ [Ramadge and Wonham, 1987]. With $\Sigma_\circ \subseteq \Sigma$ the set of observable events, we say K is *prefix-normal* w.r.t. $(L, \Sigma_\circ)$, if $\operatorname{pre} K = \operatorname{pre} L \cap (\mathrm{p}_\circ^{-1} \mathrm{p}_\circ \operatorname{pre} K)$. A language $K \subseteq \Sigma^*$ is *complete*, if for all $s \in \operatorname{pre} K$ there exists $\sigma \in \Sigma$ such that $s\sigma \in \operatorname{pre} K$ [Kumar et al., 1992] and $\Sigma - \Sigma_{\mathrm{S}}$ *complete*, with $\Sigma_{\mathrm{S}} \subseteq \Sigma$, if for all $s \in \operatorname{pre} K$ exist $t \in \Sigma_{\mathrm{S}}^*$, $\sigma \in \Sigma - \Sigma_{\mathrm{S}}$, such that $st\sigma \in \operatorname{pre} K$ [Schmidt et al., 2008]. Note that controllability, prefix-normality and completeness are retained under arbitrary union.

3. NOMINAL AND FAULT-ACCOMMODATING CONTROL

3.1 Nominal Control

Our nominal control problem is a variation [1] of modular supervisory control [Lin and Wonham, 1988], see Fig. 1 for the nominal closed-loop system.

The *nominal* overall alphabet Σ_{N} is partitioned according to

$$\Sigma_{\mathrm{N}} := \Sigma_{\mathrm{CON}} \,\dot\cup\, \Sigma_{\mathrm{UCON}} \,\dot\cup\, \Sigma_{\mathrm{HI}} \,\dot\cup\, \Sigma_{\mathrm{LO}},$$

with Σ_{CON} the *controllable events* and Σ_{UCON} the *uncontrollable events*. *Low-level events* Σ_{LO} facilitate modular control and *high-level events* Σ_{HI} provide an interface to a superordinated operator.

Both the *nominal plant* $L_{\mathrm{N}} \subseteq \Sigma_{\mathrm{P}}^*$ and the *nominal controller* $H_{\mathrm{N}} \subseteq \Sigma_{\mathrm{C}}^*$ are dynamic systems, where

$$\Sigma_{\mathrm{C}} := \Sigma_{\mathrm{UCON}} \,\dot\cup\, \Sigma_{\mathrm{CON}} \,\dot\cup\, \Sigma_{\mathrm{HI}}$$
$$\Sigma_{\mathrm{P}} := \Sigma_{\mathrm{UCON}} \,\dot\cup\, \Sigma_{\mathrm{CON}} \,\dot\cup\, \Sigma_{\mathrm{LO}}$$

[1] Formal differences between our framework and [Lin and Wonham, 1988] are purely cosmetic but convenient for the purpose of fault-hiding control reconfiguration.

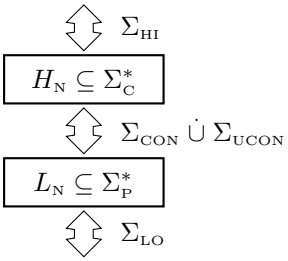

Fig. 1. Nominal closed-loop system

holds. Furthermore, we model the *nominal closed-loop system* by $L_{\mathrm{N}} \parallel H_{\mathrm{N}}$.

Definition 1. A *nominal control problem* is a pair $(L_{\mathrm{N}}, E_{\mathrm{N}})$, where $L_{\mathrm{N}} \subseteq \Sigma_{\mathrm{P}}^*$ is a nominal plant model and $E_{\mathrm{N}} \subseteq \Sigma_{\mathrm{N}}^*$ is a nominal specification. A *solution* to a nominal control problem is a controller $H_{\mathrm{N}} \subseteq \Sigma_{\mathrm{C}}^*$ satisfying

(C0) Closedness, i.e. $H_{\mathrm{N}} = \operatorname{pre} H_{\mathrm{N}}$,

such that the nominal closed-loop system $L_{\mathrm{N}} \parallel H_{\mathrm{N}}$ statisfies

(NC1) Non-conflictingness, i.e.
$(\operatorname{pre} L_{\mathrm{N}}) \parallel H_{\mathrm{N}} = \operatorname{pre}(L_{\mathrm{N}} \parallel H_{\mathrm{N}})$
(NC2) Completeness, i.e.
$(\forall s \in (\operatorname{pre} L_{\mathrm{N}}) \parallel H_{\mathrm{N}} \exists \sigma \in \Sigma_{\mathrm{N}})[s\sigma \in (\operatorname{pre} L_{\mathrm{N}}) \parallel H_{\mathrm{N}}]$
(NC3) Controllability w.r.t. $(L_{\mathrm{N}}, \Sigma_{\mathrm{uc}})$ with
$\Sigma_{\mathrm{uc}} := \Sigma_{\mathrm{UCON}} \,\dot\cup\, \Sigma_{\mathrm{LO}}$, i.e.
$(\operatorname{pre} L_{\mathrm{N}}) \parallel H_{\mathrm{N}} \Sigma_{\mathrm{uc}} \cap (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L_{\mathrm{N}}) \subseteq (\operatorname{pre} L_{\mathrm{N}}) \parallel H_{\mathrm{N}}$
(NC4) Specification Conformance, i.e. $L_{\mathrm{N}} \parallel H_{\mathrm{N}} \subseteq E_{\mathrm{N}}$. □

The following fact characterises a solution to a nominal control problem by a closed-loop candidate $K_{\mathrm{N}} \subseteq \Sigma_{\mathrm{N}}^*$. It's proof is omitted for brevity but available at request.

Fact 2. Given a nominal control problem $(L_{\mathrm{N}}, E_{\mathrm{N}})$ and a candidate language $K_{\mathrm{N}} \subseteq \Sigma_{\mathrm{N}}^*$ satisfying

(K1) Controllability w.r.t. $(L_{\mathrm{N}}, \Sigma_{\mathrm{uc}})$, with
$\Sigma_{\mathrm{uc}} := \Sigma_{\mathrm{UCON}} \,\dot\cup\, \Sigma_{\mathrm{LO}}$, i.e.
$\operatorname{pre} K_{\mathrm{N}} \Sigma_{\mathrm{uc}} \cap (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L_{\mathrm{N}}) \subseteq \operatorname{pre} K_{\mathrm{N}}$
(K2) Prefix-Normality w.r.t. $(L_{\mathrm{N}}, \Sigma_{\mathrm{C}})$, i.e.
$\operatorname{pre} K_{\mathrm{N}} = (\mathrm{p}_{\mathrm{P}}^{-1} \operatorname{pre} L_{\mathrm{N}}) \cap (\mathrm{p}_{\mathrm{C}}^{-1} \mathrm{p}_{\mathrm{C}} \operatorname{pre} K_{\mathrm{N}})$
(K3) Relative Closedness w.r.t. L_{N}, i.e.
$K_{\mathrm{N}} = (\operatorname{pre} K_{\mathrm{N}}) \cap \mathrm{p}_{\mathrm{P}}^{-1} L_{\mathrm{N}}$
(K4) Completeness, i.e.
$(\forall s \in \operatorname{pre} K_{\mathrm{N}} \exists \sigma \in \Sigma_{\mathrm{N}})[s\sigma \in \operatorname{pre} K_{\mathrm{N}}]$
(K5) Specification Conformance, i.e. $K_{\mathrm{N}} \subseteq E_{\mathrm{N}}$,

then $H_{\mathrm{N}} := \mathrm{p}_{\mathrm{C}} \operatorname{pre} K_{\mathrm{N}}$ is a solution to the given nominal control problem. Conversely, if $H_{\mathrm{N}} \subseteq \Sigma_{\mathrm{C}}^*$ is a solution to the given nominal control problem, then the nominal closed-loop system $K_{\mathrm{N}} := L_{\mathrm{N}} \parallel H_{\mathrm{N}}$ possesses the properties (K1)-(K5). Furthermore, given $H_{\mathrm{N}}^{\uparrow} := \mathrm{p}_{\mathrm{C}} \operatorname{pre} K_{\mathrm{N}}^{\uparrow}$ where $K_{\mathrm{N}}^{\uparrow}$ denotes the supremal sublanguage (of Σ^*) w.r.t. (K1)-(K5) and the supremal solution $H_{\mathrm{N}}^{\Uparrow}$ to the given nominal control problem, then the closed-loop behaviours achieved by $H_{\mathrm{N}}^{\uparrow}$ and $H_{\mathrm{N}}^{\Uparrow}$ are identical, i.e. $L_{\mathrm{N}} \parallel H_{\mathrm{N}}^{\uparrow} = L_{\mathrm{N}} \parallel H_{\mathrm{N}}^{\Uparrow}$ holds. □

An algorithm for computing the supremal sublanguage w.r.t. (K1)-(K5), can be derived according to [Moor et al., 2012]. A software implementation is given in [libFAUDES, 2006-2013]. In the following we will not distinguish between $H_{\mathrm{N}}^{\Uparrow}$ and $H_{\mathrm{N}}^{\uparrow}$.

Example 1. Consider a machine that processes a single workpiece using one of two different modes. Process start and process completion are to be reported to the high-level operator. Table 1 summarizes the relevant events.

Table 1. Simple machine alphabet

event	semantics	alphabet
a1/a2	start process 1/2	Σ_{CON}
A1/A2	completion process 1/2	Σ_{UCON}
ah/Ah	report process start/completion	Σ_{HI}

An automaton representation of the nominal plant L_{N} is drawn in Fig. 2. The nominal specification E_{N} and the

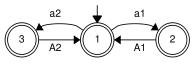

Fig. 2. Nominal plant model L_{N}

minimally restrictive nominal controller $H_{\text{N}}^{\uparrow}$ are pictured in Fig. 3. Note that the semantics of the high-level events ah and Ah are defined by the specification E_{N} and implemented solely by the controller $H_{\text{N}}^{\uparrow}$.

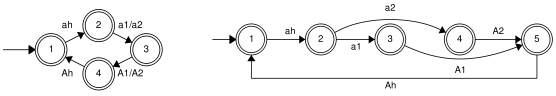

Fig. 3. Nominal specification E_{N} and nom. controller $H_{\text{N}}^{\uparrow}$

3.2 Fault-Accommodating Modelling and Control

Fault-accommodating models [Wittmann et al., 2012] are a general modelling framework for systems subject to spontaneously occurring sensor, actuator and plant faults. We model a fault by a distinct low-level event F. Given a nominal plant model $L_{\text{N}} \subseteq \Sigma_{\text{P}}^*$ and a model of the fault's history together with its impact on the plant $L_{\text{D}} \subseteq \Sigma_{\text{FP}}^*$ with

$$\Sigma_{\text{FP}} := \Sigma_{\text{P}} \dot\cup \{F\} = \Sigma_{\text{CON}} \dot\cup \Sigma_{\text{UCON}} \dot\cup \Sigma_{\text{LO}} \dot\cup \{F\},$$

then we call a pair $(L_{\text{N}}, L_{\text{D}})$ a *fault-accommodating model*. With a given fault-accommodating model $(L_{\text{N}}, L_{\text{D}})$ we associate the *fault-accommodating behaviour* [2] $L_{\text{F}} = L_{\text{N}} \cup L_{\text{D}}$. We could show that

$$L_{\text{F}} = L_{\text{N}} F \Sigma_{\text{N}}^* \cap L_{\text{D}}$$

holds, given that $\operatorname{pre} L_{\text{D}} \cap \Sigma_{\text{N}}^* \subseteq \operatorname{pre} L_{\text{N}}$ and $L_{\text{D}} \cap \Sigma_{\text{N}}^* \subseteq L_{\text{N}}$.

Example 2. (Ex. 1 cont'd). *Both modes can be subject to a sensor fault, i.e. starting in mode 1 is confirmed with ending in mode 2. The fault-accommodating model L_{F} is pictured in Fig. 4. Observe that the states 1,2 and 3*

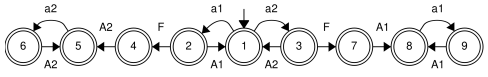

Fig. 4. Fault-accommodating plant L_{F}

correspond to the nominal plant model but the fault may only occur in the states 2 and 3. The remaining states describe the fault's impact on the plant.

[2] We will drop the distinction between model and behaviour in the following.

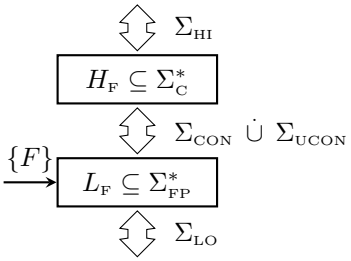

Fig. 5. Fault-accommodating closed-loop system

Using fault-accommodating models, the problem of fault-tolerant control can be reduced to a standard control problem, similar to the nominal control problem declared over a different overall alphabet

$$\Sigma_{\text{F}} := \Sigma_{\text{N}} \dot\cup \{F\} = \Sigma_{\text{CON}} \dot\cup \Sigma_{\text{UCON}} \dot\cup \Sigma_{\text{HI}} \dot\cup \Sigma_{\text{LO}} \dot\cup \{F\}.$$

Formally, a *fault-accommodating control problem* is a pair $(L_{\text{F}}, E_{\text{F}})$, where $L_{\text{F}} \subseteq \Sigma_{\text{FP}}^*$ denotes a *fault-accommodating plant model* and $E_{\text{F}} \subseteq \Sigma_{\text{F}}^*$ denotes a *fault-accommodating specification*, and a *solution* to the fault-accommodating control problem is a *fault-accommodating controller* $H_{\text{F}} \subseteq \Sigma_{\text{C}}^*$, $H_{\text{F}} = \operatorname{pre} H_{\text{F}}$ such that the closed-loop system $L_{\text{F}} \parallel H_{\text{F}}$ is non-conflicting, complete, controllable w.r.t. $(L_{\text{F}}, \Sigma_{\text{uc}})$, $\Sigma_{\text{uc}} = \Sigma_{\text{UCON}} \dot\cup \Sigma_{\text{LO}} \dot\cup \{F\}$ and conformal with E_{F}. Note that we use E_{F} to define the semantics of the operator interface in the case of a fault, see Ex. 3. The resulting closed-loop system is pictured in Fig. 5.

Example 3. (Ex. 2 cont'd). *After a fault, the process at hand is no longer available and the controller may switch to the alternative mode without reporting back to the operator. The fault-accommodating specification E_{F} and the respective minimally restrictive fault-accommodating controller H_{F} are pictured in Fig. 6. Note that H_{F} is*

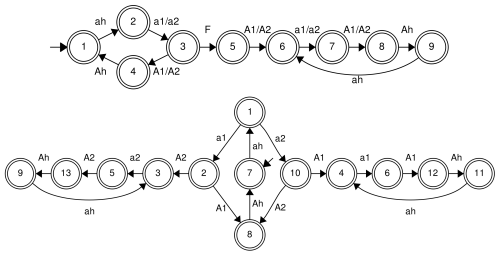

Fig. 6. Fault-accommodating spec. E_{F} and controller H_{F}

admissible for the nominal as well as the faulty plant. In this example the specification is weak enough to retain the semantics of the high-level events.

In industrial practice an expert solution is replaced by a fault-accommodating controller, which is undesirable. However, following the concept of fault-hiding control reconfiguration introduced in the next section, we can benefit from expert experience up to the fault occurence.

4. FAULT-HIDING CONTROL RECONFIGURATION

This section develops a formal framework for the design of a reconfiguration block R that operates the fault-accommodating plant L_{F} in a suitable way, while it mimics nominal plant behaviour w.r.t. the *virtualised* (decoupled) nominal controller H_{V}. The resulting *self-reconfiguring closed-loop structure* is pictured in Fig. 7.

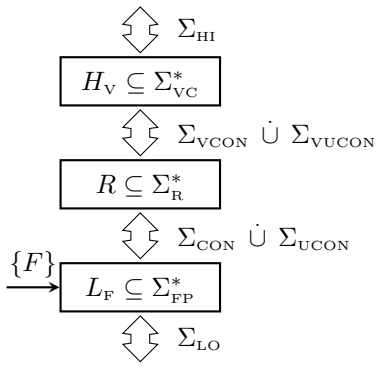

Fig. 7. Self-reconfiguring closed-loop system

To create the necessary degrees of freedom, we formally decouple the nominal controller from the fault-accommodating plant by introducing *virtual events*. Each virtual event has a unique physical counterpart with corresponding semantics. A *virtual controller* is defined over virtual events and the high-level events, but follows the nominal controller's dynamical laws.

4.1 Self-reconfiguring Closed-Loop Structure

Formally, we introduce the set of *virtual controllable events* Σ_{VCON} and the set of *virtual uncontrollable events* Σ_{VUCON} disjoint with Σ_{F} and in a bijective relation ρ with Σ_{CON} and Σ_{UCON}:

$$\rho(\,\Sigma_{\mathrm{CON}}\,) = \Sigma_{\mathrm{VCON}}\,, \quad \rho(\,\Sigma_{\mathrm{UCON}}\,) = \Sigma_{\mathrm{VUCON}}\,.$$

We extend this relationship to strings by the function $h : \Sigma^* \to \Sigma^*_{\mathrm{V}}$, defined by $h(\epsilon) := \epsilon$ and

$$h(s\sigma) := \begin{cases} h(s)\rho(\sigma) & \text{if } \sigma \in \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}}, \\ h(s)\sigma & \text{otherwise.} \end{cases}$$

The extension of the function $h(s)$ to languages is straight forward.

For a concise notation, we introduce the *virtual overall alphabet* Σ_{V}, the *virtual plant alphabet* Σ_{VP}, and the *virtual controller alphabet* Σ_{VC} according to:

$$\begin{aligned} \Sigma_{\mathrm{V}} &:= \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}} \dot\cup \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO}}\,, \\ \Sigma_{\mathrm{VP}} &:= \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}} \dot\cup \Sigma_{\mathrm{LO}}\,, \\ \Sigma_{\mathrm{VC}} &:= \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}} \dot\cup \Sigma_{\mathrm{HI}}\,. \end{aligned}$$

Given a nominal controller H_{N}, a nominal plant model L_{N} and a nominal specification E_{N}, we obtain the corresponding *virtualised controller*, the *virtualised plant* and the *virtualised specification* as follows:

$$\begin{aligned} L_{\mathrm{V}} &:= h(L_{\mathrm{N}}) \subseteq \Sigma^*_{\mathrm{VP}} \\ E_{\mathrm{V}} &:= h(E_{\mathrm{N}}) \subseteq \Sigma^*_{\mathrm{V}} \\ H_{\mathrm{V}} &:= h(H_{\mathrm{N}}) \subseteq \Sigma^*_{\mathrm{VC}}\,. \end{aligned}$$

The bijective relation between nominal events and their virtual counterparts, implies that H_{V} solves the control problem $(\,L_{\mathrm{V}},\, E_{\mathrm{V}}\,)$ if and only if H_{N} solves $(\,L_{\mathrm{N}}, E_{\mathrm{N}}\,)$.

The virtual controller H_{V} and the fault-accommodating plant L_{F} are linked via a *reconfiguration block* $R \subseteq \Sigma^*_{\mathrm{R}}$, where

$$\Sigma_{\mathrm{R}} = \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}} \dot\cup \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}}\,.$$

The behaviour of the self-reconfiguring closed-loop behaviour is given by $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$.

4.2 Formal Reconfiguration Problem

We consider a reconfiguration block to be admissible if the resulting self-reconfiguring closed-loop system is non-conflicting, complete, controllable w.r.t. $L_{\mathrm{F}} \parallel H_{\mathrm{V}}$ and if it satisfies a given specification language $E \subseteq \Sigma^*$, where

$$\Sigma := \Sigma_{\mathrm{CON}} \dot\cup \Sigma_{\mathrm{UCON}} \dot\cup \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO}} \dot\cup \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}} \dot\cup \{F\} \quad (1)$$

We subdivide the specification E according to $E = E_{\mathrm{F}} \parallel E_{\mathrm{R}}$, with a fault-accommodating specification E_{F} and a *reconfiguration specification* E_{R}. In this paper we use E_{R} to enforce *inactivity conditions* as in [Richter, 2011], i.e. one-by-one event dispatching before the occurrence of a fault; formally,

$$\begin{aligned} E_{\mathrm{R}} = (&(\{\rho(\sigma)\sigma \mid \sigma \in \Sigma_{\mathrm{CON}}\} \cup \\ &\{\sigma\rho(\sigma) \mid \sigma \in \Sigma_{\mathrm{UCON}}\} \cup \Sigma_{\mathrm{HI}} \cup \Sigma_{\mathrm{LO}})^* F \Sigma^*). \end{aligned} \quad (2)$$

Obviously, given a virtualised controller H_{V} and a fault-accommodating plant L_{F} the corresponding reconfiguration block is the solution to the control problem $(L_{\mathrm{F}} \parallel H_{\mathrm{V}}, E)$. However, in industrial practice the nominal control strategy is designed and implemented by human experts. Thus, we may not assume that a sufficiently precise controller model is available and we are faced with the design of a reconfiguration block that is functional for an *arbitrary* nominal controller.

Definition 3. A *reconfiguration problem* is a tuple $(\,L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}}\,)$, where $L_{\mathrm{N}} \subseteq \Sigma^*_{\mathrm{P}}$ is a nominal plant model, $E_{\mathrm{N}} \subseteq \Sigma^*_{\mathrm{N}}$ is a nominal specification, $L_{\mathrm{F}} \subseteq \Sigma^*_{\mathrm{FP}}$ is a fault-accommodating plant model. Furthermore, $E_{\mathrm{F}} \subseteq \Sigma^*_{\mathrm{F}}$ is a fault-accommodating specification and $E_{\mathrm{R}} \subseteq \Sigma^*$ is a reconfiguration specification, with $E = E_{\mathrm{F}} \parallel E_{\mathrm{R}}$ relatively closed w.r.t. the formal plant $L_{\mathrm{F}} \parallel H^{\uparrow}_{\mathrm{V}}$, i.e. $E = (\mathrm{pre}\, E) \cap L_{\mathrm{F}} \parallel H^{\uparrow}_{\mathrm{V}}$. A *solution* to a reconfiguration problem is a reconfiguration block $R \subseteq \Sigma^*_{\mathrm{R}}$ with,

(R0) Closedness, i.e. $R = \mathrm{pre}\, R$,

such that for an arbitrary $H_{\mathrm{V}} = \mathrm{h}(\,H_{\mathrm{N}}\,)$ where $H_{\mathrm{N}} \subseteq \Sigma^*_{\mathrm{C}}$ solves $(\,L_{\mathrm{N}}, E_{\mathrm{N}}\,)$, the self-reconfiguring control loop $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ satisfies

(RC1) Nonconflictingness, i.e.
$(\,\mathrm{pre}\, L_{\mathrm{F}}\,) \parallel R \parallel H_{\mathrm{V}} = \mathrm{pre}\,(\,L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}\,)$
(RC2) Completeness, i.e.
$(\forall s \in (\mathrm{pre}\, L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}} \exists \sigma \in \Sigma)[s\sigma \in (\mathrm{pre}\, L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}]$
(RC3) Controllability w.r.t. $(L_{\mathrm{F}} \parallel H_{\mathrm{V}}, \Sigma_{\mathrm{uc}})$, where
$\Sigma_{\mathrm{uc}} = \Sigma_{\mathrm{VUCON}} \dot\cup \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO}} \dot\cup \{F\}$, i.e.
$(\mathrm{pre}\, L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}} \Sigma_{\mathrm{uc}} \cap (\mathrm{pre}\, L_{\mathrm{F}}) \parallel H_{\mathrm{V}} \subseteq (\mathrm{pre}\, L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$
(RC4) Specification Conformance, i.e. $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}} \subseteq E$. □

4.3 Reconfiguration Block Synthesis

Note that $H^{\uparrow}_{\mathrm{V}}$ is an upper bound to all virtualised solutions H_{V} to the nominal control problem $(L_{\mathrm{N}}, E_{\mathrm{N}})$. Since H_{V} is unknown, we use $H^{\uparrow}_{\mathrm{V}}$ instead. We interpret $L_{\mathrm{F}} \parallel H^{\uparrow}_{\mathrm{V}}$ as a formal plant model controlled by the reconfiguration block R. However, a solution to the control problem $(L_{\mathrm{F}} \parallel H^{\uparrow}_{\mathrm{V}}, E)$ is not necessarily a solution to the reconfiguration problem $(\,L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}}\,)$, since the liveness properties non-conflictingness and completeness are not guaranteed for an arbitrary nominal controller.

Lemma 4. Given is a reconfiguration problem $(L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ and the virtualised minimally restrictive solution

$H_{\mathrm{V}}^{\uparrow}$ to the nominal control problem $(L_{\mathrm{N}}, E_{\mathrm{N}})$. Let $E = E_{\mathrm{F}} \parallel E_{\mathrm{R}}$ be relatively closed w.r.t. the formal plant, i.e. $E = (\operatorname{pre} E) \cap (L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow})$, and assume that the language $K \subseteq \Sigma^*$ satisfies

(M1) Controllability w.r.t. $(L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}, \Sigma_{\mathrm{uc}})$, with $\Sigma_{\mathrm{uc}} = \Sigma_{\mathrm{VUCON}} \dot\cup \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO}} \dot\cup \{F\}$ i.e. $(\operatorname{pre} K \Sigma_{\mathrm{uc}}) \cap ((\operatorname{pre} L_{\mathrm{F}}) \parallel H_{\mathrm{V}}^{\uparrow}) \subseteq \operatorname{pre} K$
(M2) Prefix-Normality w.r.t. $L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}$ and Σ_{R}, i.e. $\operatorname{pre} K = ((\operatorname{pre} L_{\mathrm{F}}) \parallel H_{\mathrm{V}}^{\uparrow}) \cap (\mathrm{p}_{\mathrm{R}}^{-1}\, \mathrm{p}_{\mathrm{R}} \operatorname{pre} K)$
(M3) Relative Closure w.r.t. $L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}$, i.e. $K = (\operatorname{pre} K) \cap (L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow})$
(M4) $\Sigma - \Sigma_{\mathrm{HI}}$-Completeness, i.e. $(\forall s \in \operatorname{pre} K\ \exists \sigma \notin \Sigma_{\mathrm{HI}},\, t \in \Sigma_{\mathrm{HI}}^*)[\, st\sigma \in \operatorname{pre} K\,]$
(M5) Weak Sensor-Consistency, i.e. $(\forall s \in \operatorname{pre} K)[(\mathrm{p}_{\mathrm{VP}}\, s)\Sigma_{\mathrm{VUCON}} \cap (\operatorname{pre} L_{\mathrm{V}}) \neq \emptyset \Rightarrow s(\Sigma - \Sigma_{\mathrm{VC}})^* \Sigma_{\mathrm{VUCON}} \cap (\operatorname{pre} K) \neq \emptyset]$
(M6) Plant Conformance, i.e. $\operatorname{pre} K \subseteq \mathrm{p}_{\mathrm{VP}}^{-1} \operatorname{pre} L_{\mathrm{V}}$
(M7) Specification Conformance, i.e. $\operatorname{pre} K \subseteq \operatorname{pre} E$.

Consider the reconfiguration block

$$R = \mathrm{p}_{\mathrm{R}} \operatorname{pre} K. \tag{3}$$

For any virtualised solution to the nominal control problem $H_{\mathrm{V}} \subseteq \Sigma_{\mathrm{VC}}^*$ the self-reconfiguring closed-loop system $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ possesses the properties (RC2)-(RC4). Furthermore R is closed (R0). □

Proof. We choose $K \subseteq \Sigma^*$ with the properties (M1)-(M7) and start with the proof of two auxiliary statements

$$\operatorname{pre} K = (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow} \tag{4}$$
$$K = L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}, \tag{5}$$

proven by the deductions

$$\begin{aligned}
\operatorname{pre} K &= (\mathrm{p}_{\mathrm{FP}}^{-1} \operatorname{pre} L_{\mathrm{F}}) \cap (\mathrm{p}_{\mathrm{VC}}^{-1} H_{\mathrm{V}}^{\uparrow}) \cap (\mathrm{p}_{\mathrm{R}}^{-1}\, \mathrm{p}_{\mathrm{R}} \operatorname{pre} K) \\
&= (\mathrm{p}_{\mathrm{FP}}^{-1} \operatorname{pre} L_{\mathrm{F}}) \cap (\mathrm{p}_{\mathrm{R}}^{-1} R) \cap (\mathrm{p}_{\mathrm{VC}}^{-1} H_{\mathrm{V}}^{\uparrow}) \\
&= (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow}, \\
K &= (\operatorname{pre} K) \cap (L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}) \\
&= ((\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow}) \cap (L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}) \\
&= L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow}.
\end{aligned}$$

Ad (R0): (R0) directly follows from the definition of R.

In order to establish (RC2)-(RC4), we choose an arbitrary virtualised solution $H_{\mathrm{V}} \subseteq \Sigma_{\mathrm{VC}}^*$ to the nominal control problem $(L_{\mathrm{N}}, E_{\mathrm{N}})$.

Ad (RC2): Picking an arbitrary $s \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$, we need to establish the existence of a $\sigma \in \Sigma$ such that $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$. In order to structure the following proof we introduce the set of events enabled by H_{V}, $\gamma_{\mathrm{H}} := \{\sigma \in \Sigma_{\mathrm{VC}} \mid (\mathrm{p}_{\mathrm{VC}}\, s)\sigma \in H_{\mathrm{V}}\}$, and the set of events simultaneously enabled by L_{F} and R, $\gamma_{\mathrm{LR}} := \{\sigma \in \Sigma_{\mathrm{R}} \cup \Sigma_{\mathrm{FP}} \mid (\mathrm{p}_{\mathrm{FP}}\, s)\sigma \in \operatorname{pre} L_{\mathrm{F}} \text{ and } (\mathrm{p}_{\mathrm{R}}\, s)\sigma \in R\}$. Since H_{V} solves a standard control problem, the closed-loop system $L_{\mathrm{V}} \parallel H_{\mathrm{V}}$ is complete. Referring to Eq. (4), completeness (M4) of K implies $\gamma_{\mathrm{LR}} \neq \emptyset$. We distinguish the following cases:

Case 1a ($\gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{HI}} \neq \emptyset$, H_{V} can evolve independently of L_{F}): Picking $\sigma \in \gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{HI}}$, we obtain $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$, as $(\Sigma_{\mathrm{R}} \cup \Sigma_{\mathrm{FP}}) \cap \Sigma_{\mathrm{HI}} = \emptyset$ holds, which implies (RC2). Thus, for the sequel we may assume $\gamma_{\mathrm{H}} \subseteq \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}}$.

Case 1b ($\gamma_{\mathrm{LR}} \cap \Sigma_{\mathrm{FP}} \neq \emptyset$, $L_{\mathrm{F}} \parallel R$ can evolve independently of H_{V}): Pick $\sigma \in \gamma_{\mathrm{LR}} \cap \Sigma_{\mathrm{FP}}$, we obtain $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$ as $\Sigma_{\mathrm{FP}} \cap \Sigma_{\mathrm{VC}} = \emptyset$ holds, which implies (RC2). Thus, for the sequel we may assume $\gamma_{\mathrm{LR}} \subseteq \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}}$.

In summary, we have shown that Case 1 implies completeness whenever H_{V} and $L_{\mathrm{F}} \parallel R$ can evolve independently. It remains to s tudy cases where $L_{\mathrm{F}} \parallel R$ and H_{V} must evolve together, namely $\gamma_{\mathrm{LR}} \cap \gamma_{\mathrm{H}}$ is nonempty.

Case 2a ($\gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{VCON}} \neq \emptyset$): Assume that we can choose $\sigma \in \gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{VCON}}$. Observe that $\mathrm{p}_{\mathrm{FP}}(s\sigma) = \mathrm{p}_{\mathrm{FP}}(s) \in L_{\mathrm{F}}$ and $\mathrm{p}_{\mathrm{VC}}(s\sigma) \in H_{\mathrm{V}}^{\uparrow}$ hold, the latter one from $H_{\mathrm{V}} \subseteq H_{\mathrm{V}}^{\uparrow}$. Thus, controllability (M1) implies $s\sigma \in \operatorname{pre} K$ and we conclude $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$, in particular $\sigma \in \gamma_{\mathrm{LR}}$ which implies (RC2). The remaining case is characterised by $\gamma_{\mathrm{H}} \subseteq \Sigma_{\mathrm{VUCON}}$, i.e. the controller waits for an uncontrollable event.

Case 2b ($\gamma_{\mathrm{H}} \cap \Sigma_{\mathrm{VCON}} = \emptyset$): Together with $\gamma_{\mathrm{LR}} \subseteq \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}}$ this case implies $\gamma_{\mathrm{H}} \subseteq \Sigma_{\mathrm{VUCON}}$. Referring to Eq. (4) and $H_{\mathrm{V}} \subseteq H_{\mathrm{V}}^{\uparrow}$, we have $s \in \operatorname{pre} K$. From (M6) we obtain $\operatorname{pre} K \subseteq \mathrm{p}_{\mathrm{VP}}^{-1} \operatorname{pre} L_{\mathrm{V}}$ and thus $\mathrm{p}_{\mathrm{VP}}\, s \in \operatorname{pre} L_{\mathrm{V}}$. Since H_{V} solves a standard control problem for the plant L_{V}, closed-loop completeness (NC2) implies the existence of σ such that $\mathrm{p}_{\mathrm{VP}}(s\sigma) \in \operatorname{pre} L_{\mathrm{V}}$ and $\mathrm{p}_{\mathrm{VC}}(s\sigma) \in \operatorname{pre} H_{\mathrm{V}}$. In particular, we must have $\sigma \in \Sigma_{\mathrm{VUCON}}$, i.e. the controller waits for a sensor event. Hence, weak sensor consistency (M5) implies that there exits $t \in (\Sigma - \Sigma_{\mathrm{VC}})^*$ such that $st\sigma' \in \operatorname{pre} K$ for some $\sigma' \in \Sigma_{\mathrm{VUCON}}$. Referring to Eq. (4), we have $(\mathrm{p}_{\mathrm{FP}}\, st\sigma') \in \operatorname{pre} L_{\mathrm{F}}$ and $(\mathrm{p}_{\mathrm{R}}\, st\sigma') \in R$. Then, $\gamma_{\mathrm{LR}} \subseteq \Sigma_{\mathrm{VCON}} \dot\cup \Sigma_{\mathrm{VUCON}}$ implies $t = \epsilon$ and we obtain $\sigma' \in \gamma_{\mathrm{LR}}$. Since $\mathrm{p}_{\mathrm{V}}\, s \in (\operatorname{pre} L_{\mathrm{V}}) \parallel H_{\mathrm{V}}$ and the nominal closed-loop system $L_{\mathrm{V}} \parallel H_{\mathrm{V}}$ is controllable w.r.t. $(L_{\mathrm{V}}, \Sigma_{\mathrm{VUCON}})$ (from NC3) we obtain $\mathrm{p}_{\mathrm{V}}\, s\sigma \in (\operatorname{pre} L_{\mathrm{V}}) \parallel H_{\mathrm{V}}$, in particular $s\sigma \in \mathrm{p}_{\mathrm{VC}}^{-1} H_{\mathrm{V}}$. This concludes the proof of completeness (RC2).

Ad (RC3): Pick any $s \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$ and any $\sigma \in \Sigma_{\mathrm{uc}}$, with $\Sigma_{\mathrm{uc}} := \Sigma_{\mathrm{HI}} \dot\cup \Sigma_{\mathrm{LO}} \dot\cup \Sigma_{\mathrm{UCON}} \dot\cup \Sigma_{\mathrm{VCON}}$, such that $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel H_{\mathrm{V}}$. From the supremality of $H_{\mathrm{V}}^{\uparrow}$ we have $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel H_{\mathrm{V}}^{\uparrow}$ and together with Eq. (4) we obtain $s \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}} \subseteq (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow} = \operatorname{pre} K$. From (M1) $s\sigma \in \operatorname{pre} K = (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow}$ follows. Finally, $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel H_{\mathrm{V}}$ implies $s\sigma \in (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}^{\uparrow} \cap (\operatorname{pre} L_{\mathrm{F}}) \parallel H_{\mathrm{V}} = (\operatorname{pre} L_{\mathrm{F}}) \parallel R \parallel H_{\mathrm{V}}$, which concludes the proof of controllability (RC3).

Ad (RC4): To establish (RC4), note that from the supremality of $H_{\mathrm{V}}^{\uparrow}$ and Eq. (5) we have $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}} \subseteq L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}^{\uparrow} = K$. Since K is relatively closed w.r.t. the formal plant (M3), together with (M7), we have $K \subseteq (\operatorname{pre} E) \cap (L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow})$. Since E is relatively closed w.r.t. to $L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}$, we obtain $K \subseteq E$, concluding the proof of specification complicance (RC4). ■

Each of the properties (M1),(M2),(M4) and (M5) satisfies the prerequisites of [Moor et al., 2012], thus the supremal sublanguage w.r.t. (M1)-(M7) can be computed using results from [Moor et al., 2012].

If L_{F} is closed, the property (RC1) is trivially satisfied, which leads to our main practical result.

Corollary 5. Consider a reconfiguration problem $(L_{\mathrm{N}}, E_{\mathrm{N}}, L_{\mathrm{F}}, E_{\mathrm{F}}, E_{\mathrm{R}})$ and assume all languages are closed. If a closed reconfiguration block $R \subseteq \Sigma_{\mathrm{R}}^*$ satisfies the properties (RC2)-(RC4), then R is a solution to the reconfiguration problem. ■

4.4 Result Validation

Example 4. (Ex. 3 cont'd). *In the first step we virtualise the minimally restrictive solution to the nominal control problem to obtain $H_{\mathrm{V}}^{\uparrow}$, see Fig. 8.*

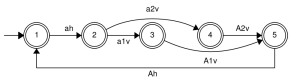

Fig. 8. Minimally restrictive nominal controller $H_{\mathrm{V}}^{\uparrow}$

As an additional design requirement we impose the inactivity condition E_{R} according to Eq. 2, see Fig. 9.

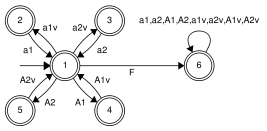

Fig. 9. Reconfiguration specification E_{R}

The supremal sublanguage K of $L_{\mathrm{F}} \parallel H_{\mathrm{V}}^{\uparrow}$ w.r.t. (M1)-(M7) is non-empty and the respective reconfiguration block $R = \mathrm{p_R}\,\mathrm{pre}\,K$ is pictured in Fig. 10.

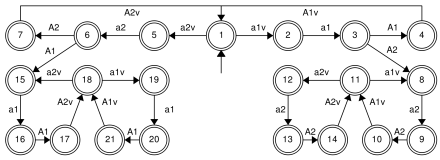

Fig. 10. Reconfiguration block R

In order to evaluate the behaviour of the self-reconfiguring closed-loop system we choose a restrictive solution H_{V} to the nominal control problem according to Fig. 11. The

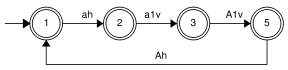

Fig. 11. Restrictive virtualised nominal controller H_{V}

behaviour of the resulting self-reconfiguring closed-loop system $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$ is shown in Fig. 12.

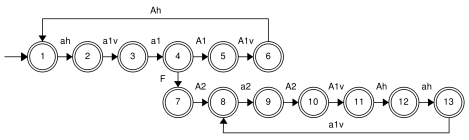

Fig. 12. Self-reconfiguring closed-loop system $L_{\mathrm{F}} \parallel R \parallel H_{\mathrm{V}}$

All languages in Example 4 are closed, thus from Corollary 5 we can guarantee admissible behaviour of the self-reconfiguring closed-loop system for an arbitrary virtual solution to the nominal control problem.

Consider the self-reconfiguring closed-loop system in Fig. 12. During nominal operation (states 1-6) the reconfiguration block is inactive and nominal control commands are dispatched one-by-one. After the fault the reconfiguration block mimics a functional nominal plant by generating the events *a1v* and *A1v* and thus hides a fault from the virtual controller.

5. CONCLUSION

From a theoretical perspective, we have established a framework for fault-hiding control reconfiguration for discrete event systems based on SCT. Our main results are sufficient conditions for the synthesis of a reconfiguration block, guaranteeing admissible behaviour of the reconfiguring closed-loop system for an arbitrary nominal controller. Due to fault-accommodating models we do not depend on external diagnosis or controller switching mechanisms but our results are effectively restricted to closed languages. Current research addresses extension of the fault-hiding control reconfiguration framework for not necessarily closed plants, and its experimental evaluation. From a practical perspective, fault-tolerant control capabilities can be retrofit to existing control systems, even if they are hierarchically structured. Since our reconfiguration block works with an arbitrary nominal controller, its design is independent of the actual nominal controller implementation: a desirable property for industrial applications.

REFERENCES

M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder. *Diagnosis and Fault-Tolerant Control*. Springer, 2006.

R. Kumar, V. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Trans. Automatic Control*, 37:1978–85, 1992.

libFAUDES. A software library for discrete event systems, 2006-2013. URL `www.rt.eei.uni-erlangen.de/FGdes/faudes`.

F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44:173–198, 1988.

J. Lunze and T. Steffen. Control reconfiguration after actuator failures using disturbance decoupling methods. *IEEE Trans. Automatic Control*, pages 1590–1601, 2006.

T. Moor, Ch. Baier, T.-S. Yoo, F. Lin, and S. Lafortune. On the computation of supremal sublanguages relevant to supervisory control. *Proc. 11th Workshop on Discrete Event Systems (WODES)*, 2012.

Y. Nke and J. Lunze. Systematic design of fault-tolerant discrete event controllers. *at Automatisierungstechnik*, pages 122–130, 2013.

A. Paoli, M. Sartini, and S. Lafortune. A fault tolerant architecture for supervisory control of discrete event systems. In *Proc. 17th IFAC World Congress*, pages 6542–47, Seoul, Korea, 2008.

A. Paoli, M. Sartini, and S. Lafortune. Active fault tolerant control of discrete event systems using online diagnostics. *Automatica*, 47 (4):639–649, 2011.

P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization*, 25:206–230, 1987.

P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proc. IEEE*, 77:81–98, 1989.

J. H. Richter. *Reconfigurable control of nonlinear dynamical systems: fault hiding approach*, volume 408 of *LNCIS*. Springer, 2011.

J. H. Richter, W.P.M.H. Heemels, N. van de Wouw, and J. Lunze. Reconfigurable control of piecewise affine systems with actuator and sensor faults: Stability and tracking. *Automatica*, 47(4):678–691, 2011. ISSN 0005-1098.

K. Schmidt. Computation of supervisors for reconfigurable machine tools. *Proc. 11th Workshop on Discrete Event Systems*, pages 227–232, 2012.

K. Schmidt, Th. Moor, and S. Perk. Nonblocking hierachical control of decentralized discrete event systems. *IEEE Trans. Automatic Control*, 53:2252–65, 2008.

Th. Wittmann, J. H. Richter, and T. Moor. Fault-tolerant control of discrete event systems based on fault-accommodating models. *Proc. 8th IFAC SAFEPROCESS*, pages 854–859, 2012.