Hierarchical Multitasking Control of Discrete Event Systems: Computation of Projections and Maximal Permissiveness

Klaus Schmidt * **Max H. de Queiroz** ** **José E.R. Cury** **

* *Department of Electronic and Communication Engineering, Cankaya University, Ankara, Turkey (e-mail: schmidt@cankaya.edu.tr)*
** *Department of Automation and Systems, Federal University of Santa Catarina, Brazil (e-mail: {max,cury}@das.ufsc.br)*

Abstract: This paper extends previous results on the hierarchical and decentralized control of *multitasking* discrete event systems (MTDES). *Colored observers*, a generalization of the observer property, together with *local control consistency*, allow to derive sufficient conditions for synthesizing modular and hierarchical control that are both *strongly nonblocking* (SNB) and *maximally permissive*. A polynomial procedure to verify if a projection fulfills the above properties is proposed and in the case they fail for a given projection an algorithm is proposed to find an extension of the set of events to be projected, in order to fulfill the sufficient conditions for SNB and maximally permissive hierarchical control.

Keywords: Discrete event systems, hierarchical control, multitasking, maximal permissiveness.

1. INTRODUCTION

Several approaches that combine modular and hierarchical control have been recently exploited in the literature of Discrete Event Systems (DES) (Leduc et al., 2005; Hill and Tilbury, 2008; Feng and Wonham, 2008; Schmidt et al., 2008; Su et al., 2010). Decomposition and abstraction of the system models, considered by all those approaches, allow for the use of efficient computational DES control synthesis methods which in general provide solutions that can be implemented by a set of modular supervisors.

In (de Queiroz et al., 2005), a multitasking DES (MTDES) approach is introduced to deal with the case where multiple classes of tasks and multiple liveness specifications are considered in the system. In (Schmidt et al., 2007), multitasking control is extended with hierarchical and decentralized ideas. To this end, a colored version of both natural projections and observers (Wong and Wonham, 2004) is employed such that the resulting hierarchical control architecture is (strongly) nonblocking (SNB), namely, nonblocking w.r.t. all classes of system tasks. As a first contribution of this paper, the sufficient conditions for SNB control in (Schmidt et al., 2007) are extended by *local control consistency* (LCC, (Schmidt and Breindl, 2010)) and *mutual controllability* (Lee and Wong, 2002) in order to achieve SNB, maximally permissive hierarchical and decentralized multitasking control.

In practice, efficient algorithms are needed to verify if the above conditions hold for given colored projections or to compute appropriate colored projections. In this sense, this paper brings two further contributions. First, a condition to verify the joint colored observer and LCC property is proposed. This condition is based on a generalization of a result by Wong and Wonham (2004) to MTDES, and it can be verified in polynomial time. We then consider the case where the verification of the colored observer condition and/or LCC fails for a given projection and propose an algorithm to find an extension of the set of events to be projected, in order to fulfill the sufficient conditions for SNB and maximally permissive hierarchical control. To this end, we employ a generalization of the *event set extension* algorithm in Feng and Wonham (2009) that is first proposed in (Schmidt and Breindl, 2010).

The outline of the paper is as follows. Basic definitions and previous work are summarized in Section 2. Section 3 states the conditions for SNB and maximally permissive control. Algorithms for the verification and computation of appropriate projections are provided in Section 4. The results are illustrated by a detailed example in Section 5.

2. PRELIMINARIES

2.1 Basic Notation

For a finite alphabet Σ, the set of all finite *strings* over Σ is denoted as Σ^* and the *empty string* is ϵ such that $\epsilon s = s\epsilon = s$ for all $s \in \Sigma^*$. A *language* over Σ is a subset $L \subseteq \Sigma^*$, and $\overline{L}$ describes the *prefix-closure* of L. The *natural projection* $p : \Sigma^* \rightarrow \hat{\Sigma}^*$, $\hat{\Sigma} \subseteq \Sigma$ is defined iteratively: (1) let $p(\epsilon) := \epsilon$; (2) for $s \in \Sigma^*$, $\sigma \in \Sigma$, let $p(s\sigma) := p(s)\sigma$ if $\sigma \in \hat{\Sigma}$, or $p(s\sigma) := p(s)$ otherwise. The inverse of p is $p^{-1} : \hat{\Sigma}^* \rightarrow 2^{\Sigma^*}$, $p^{-1}(t) := \{s \in \Sigma^* | \, p(s) = t\}$. A condition for p that is relevant to this paper is the *observer condition* (Wong and Wonham, 2004).

Definition 1. Let $L \subseteq \Sigma^*$ be a language and $\hat{\Sigma} \subseteq \Sigma$. $p : \Sigma^* \rightarrow \hat{\Sigma}^*$ is an L-observer if for all $s \in \overline{L}$, $t \in \hat{\Sigma}^*$,

$$p(s)t \in p(L) \Rightarrow \exists u \in \Sigma^* \; s.t. \; su \in L \wedge p(su) = p(s)t. \quad (1)$$

As proposed by de Queiroz et al. (2005), we consider multitasking DES (MTDES), where each system *task* is associated to a color in a *color set* C. Such MTDES is characterized by its *colored behavior* $\Lambda_C \in Pwr(Pwr(\Sigma^*)\times C)$ that consists of a set of pairs $(L_c(\Lambda_C), c)$, whereby $L_c(\Lambda_C)$ represents the strings in the MTDES that complete a task of color $c \in C$. Similarly, the language marked by $B \subseteq C$ is $L_B(\Lambda_C) := \bigcup_{b\in B} L_b(\Lambda_C)$, and we say that Λ_C is *strongly nonblocking* (SNB) w.r.t. $B \subseteq C$ if $\forall b \in B$, $\overline{L_b(\Lambda_C)} = \overline{L_C(\Lambda_C)}$. The subset relation $\Lambda_B \subseteq \Lambda_C$ for two colored behavior Λ_B, Λ_C holds if $B \subseteq C$ and for all $b \in B$, $L_b(\Lambda_B) \subseteq L_b(\Lambda_C)$. We define the *colored projection* $m : Pwr(Pwr(\Sigma^*) \times C) \to Pwr(Pwr(\hat{\Sigma}^*) \times C)$ for $\hat{\Sigma} \subseteq \Sigma$ such that, for $\Lambda_C \in Pwr(Pwr(\Sigma^*) \times C)$ and for all $c \in C$: $L_c(m(\Lambda_C)) = p(L_c(\Lambda_C))$. Now, the observer condition is generalized for colored projections.

Definition 2. Let $\Lambda_C \in Pwr(Pwr(\Sigma^*) \times C)$ be a coloring behavior and $\hat{\Sigma} \subseteq \Sigma$, p, m as above. m is a (colored) Λ_C-observer iff p is an $L_c(\Lambda_C)$-observer for each color $c \in C$.

Finally, the *synchronous composition* of $M_{C_1} \in Pwr(Pwr(\Sigma_1^*) \times C_1)$ and $N_{C_2} \in Pwr(Pwr(\Sigma_2^*) \times C_2)$ is

$$\begin{aligned} M_{C_1}||N_{C_2} :=& \{(L_c(M_{C_1})||L_c(N_{C_2}), c), \forall c \in C_1 \cap C_2\} \\ &\cup \{(L_c(M_{C_1})||L_{C_2}(N_{C_2}), c), \forall c \in C_1 - C_2\} \\ &\cup \{(L_{C_1}(M_{C_1})||L_c(N_{C_2}), c), \forall c \in C_2 - C_1\}. \end{aligned}$$

We model an MTDES as a *colored marking generator* (CMG) $H = (Q, \Sigma, C, \delta, \chi, q_0)$ with the set of *states* Q, the *alphabet* Σ, the *color set* C, the *transition function* $\delta : Q \times \Sigma \to Q$, the *marking function* $\chi : Q \to Pwr(C)$ and the *initial state* q_0. We extend δ to strings in the usual way, and define the *eligible event function* $\Gamma : Q \to Pwr(\Sigma)$ such that $\Gamma(q) = \{\sigma \in \Sigma | \delta(q, \sigma) \text{ exists}\}$ for $q \in Q$. The *generated language* of H is $L(H) := \{s \in \Sigma^* | \delta(q_0, s) \text{ exists}\}$, the language *marked* by $c \in C$, is given by $L_c(H) := \{s \in L(H) | c \in \chi(\delta(q_0, s))\}$, and the colored behavior of a CMG H is given by $\Lambda_C(H) := \{(L_c(G), c) | c \in C\}$. A formal definition of the *synchronous composition* $H_1||H_2$ of two CMGs H_1 and H_2 is given by de Queiroz et al. (2005). A CMG H is SNB w.r.t. $B \subseteq C$ if $\Lambda_C(H)$ is SNB w.r.t. B and $\overline{L_B(\Lambda_B(H))} = L(H)$.

2.2 Multitasking Supervisory Control

Let an MTDES be modeled by a CMG $H = (Q, \Sigma, C, \delta, \chi, q_0)$ whose alphabet is partitioned into controllable events Σ_c and uncontrollable events Σ_{uc}. We assume w.l.o.g. that a *colored specification* $A_D \subseteq Pwr(\Sigma^*) \times D$ is constructed from a safety specification $K = \overline{K} \subseteq L(H)$ and liveness conditions defined by the color set C and a set of new colors E s.t. $E \cap C = \emptyset$ and $D = C \dot{\cup} E$ as follows.

$$A_D = \{(L_c, c) |\ c \in D \text{ s.t. } L_c = K \cap L_c(H) \text{ for } c \in C \text{ and } L_c \subseteq K \text{ for } c \in E\}. \quad (2)$$

A *coloring supervisor* $S : L(H) \to Pwr(\Sigma) \times Pwr(E)$ associates to each string of the plant a set of enabled events and a set of colors (of E) marking the string as a completed task of these colors. For $S(s) = (\gamma, \mu)$, let $\mathcal{R}(S(s)) = \gamma$ and $\mathcal{I}(S(s)) = \mu$. The events that can occur in S/H after the occurrence of a string $s \in L(H)$ are $\mathcal{R}(S(s)) \cap \Gamma(\delta(q_0, s))$. A string $s \in L(S/H)$ is marked by a color $c \in C$ if $s \in L_c(H)$ or by a color $e \in E$ if $e \in \mathcal{I}(S(s))$. A coloring supervisor S is *admissible* if $\forall s \in L(H)$, $\Sigma_{uc} \cap \Gamma(\delta(q_0, s)) \subseteq \mathcal{R}(S(s))$, and SNB if $\forall d \in D$, $\overline{L_d(S/H)} = L(S/H)$.

A colored specification behavior A_D as defined in (2) is controllable w.r.t. $L(H)$ and $\Sigma_{uc} \subseteq \Sigma$ if $\overline{L_D(A_D)}\Sigma_{uc} \cap L(H) \subseteq \overline{L_D(A_D)}$. The set of all controllable subbehaviors of A_D w.r.t. $L(H)$ and Σ_{uc} is denoted as $\mathcal{C}(L(H)) = \{A'_D \subseteq A_D | \overline{L_D(A'_D)}\Sigma_{uc} \cap L(H) \subseteq \overline{L_D(A'_D)}\}$. Since $\mathcal{C}(L(H))$ is closed under arbitrary union (de Queiroz et al., 2005), there uniquely exists a *supremal controllable subbehavior* of A_D w.r.t. $L(H)$ and Σ_{uc}. It is formally defined as $SupCSNB(A_D, H, D)$ and can be computed in $\mathcal{O}(|A_D|^2|H|^2|D|)$. A coloring supervisor S such that $\Lambda_D(S/H) = SupCSNB(A_D, H, D)$ exists if $L_d(SupCSNB(A_D, H, D)) \neq \emptyset$ for all $d \in D$ and is SNB and *maximally permissive*.

2.3 Set Theory

We present basic results from set theory as employed by Feng and Wonham (2009); Wong and Wonham (2004). We denote $\mathcal{E}(Q)$ the set of all *equivalence relations* on the set Q. For $\mu \in \mathcal{E}(Q)$, $[q]_\mu$ is the *equivalence class* containing $q \in Q$. The set of equivalence classes of μ is written as $Q/\mu := \{[q]_\mu | q \in Q\}$ and the *canonical projection* $\text{cp}_\mu : Q \to Q/\mu$ maps an element $q \in Q$ to its equivalence class $[q]_\mu$. Let $f : Q \to R$ be a function. The equivalence relation $\ker f$ is the *kernel* of f and is defined as follows: for $q, q' \in Q$,

$$q \equiv q' \mod \ker f \quad \Leftrightarrow \quad f(q) = f(q').$$

Given two equivalence relations η and μ on q, $\mu \leq \eta$, i.e. μ refines η, if $q \equiv q' \mod \mu \Rightarrow q \equiv q' \mod \eta$ for all $q, q' \in Q$. In addition, we define the *meet* operation $\wedge$ for $\mathcal{E}(Q)$ as follows. For any two elements $\mu, \eta \in \mathcal{E}(Q)$, it holds for all $q, q' \in Q$ that

$$q \equiv q' \mod (\mu \wedge \eta) \Leftrightarrow q \equiv q' \mod \mu \text{ and } q \equiv q' \mod \eta.$$

Let Q and R be sets and $f : Q \to 2^R$ be a function. It is also assumed that $\varphi \in \mathcal{E}(R)$, and the canonical projection cp_φ is naturally extended to sets. The equivalence relation $\varphi \circ f$ on Q is defined for $q, q' \in Q$ by

$$q \equiv q' \mod \varphi \circ f \Leftrightarrow \text{cp}_\varphi(f(q)) = \text{cp}_\varphi(f(q')).$$

Now let $f_i : Q \to 2^Q$ be functions, where i ranges over an index set $\mathcal{I}$. Then $S := (Q, \{f_i | i \in \mathcal{I}\})$ is called a *dynamic system* (Wong and Wonham, 2004). The equivalence relation $\varphi \in \mathcal{E}(Q)$ is called a *quasi-congruence* (QC) for S if

$$\varphi \leq \bigwedge_{i\in\mathcal{I}} (\varphi \circ f_i).$$

The *quotient CMG* (QCMG) $H_{\mu,\hat{\Sigma}} = (Y, \hat{\Sigma} \cup \{\sigma_0\}, \nu, C, \kappa, y_0)$ of a CMG $H = (Q, \Sigma, \delta, C, \chi, q_0)$ for an equivalence relation $\mu \in \mathcal{E}(Q)$ and an alphabet $\hat{\Sigma} \subseteq \Sigma$ is introduced analogous to (Wong and Wonham, 2004). It holds that $Y := Q/\mu$ is the *quotient set* with the associated *canonical projection* $\text{cp}_\mu : Q \to Y$. The initial state is $y_0 = \text{cp}_\mu(q_0)$. Also $\sigma_0 \notin \hat{\Sigma}$ is an additional label and the coloring function $\kappa : Y \to Pwr(C)$ is defined such that $\kappa(y) = \bigcup_{q \in cp_\mu^{-1}(y)} (\chi(q))$. The nondeterministic *induced transition function* $\nu : Y \times (\hat{\Sigma} \cup \{\sigma_0\}) \to 2^Y$ of $G_{\mu,\hat{\Sigma}}$ is defined as

$$\nu(y, \sigma) := \begin{cases} \{\text{cp}_\mu(\delta(q, \sigma)) | q \in \text{cp}_\mu^{-1}(y)\} & \text{if } \sigma \in \hat{\Sigma} \\ \{\text{cp}_\mu(\delta(q, \gamma)) | \gamma \in (\Sigma - \hat{\Sigma}), \\ q \in \text{cp}_\mu^{-1}(y)\} - \{y\} & \text{if } \sigma = \sigma_0. \end{cases}$$

3. MAXIMALLY PERMISSIVE HIERARCHICAL MULTITASKING CONTROL

In this section, we extend the SNB hierarchical control as in (Schmidt et al., 2007) to maximal permissiveness.

3.1 Local Supervisory Control

We consider a system that is modeled by multiple plant CMGs $H_i = (Q_i, \Sigma_i, C_i, \delta_i, \chi_i, q_{0,i})$ with the uncontrollable alphabets $\Sigma_{i,\mathrm{uc}}$, $i = 1, \ldots, n$ such that for all $i, j = 1, \ldots, n$ with $i \neq j$, $\Sigma_{i,\mathrm{uc}} \cap (\Sigma_j - \Sigma_{j,\mathrm{uc}}) = \emptyset$. The overall plant is $H := ||_{i=1}^n H_i$ and the uncontrollable alphabet is $\Sigma_{\mathrm{uc}} := \bigcup_{i=1}^n \Sigma_{i,\mathrm{uc}}$. The desired system behavior is given by n local specifications $A_{i,D_i} \subseteq Pwr(Pwr(\Sigma_i^*) \times C_i)$, $i = 1, \ldots, n$ for the local models and a global specification $\hat{A}_D \subseteq Pwr(Pwr(\hat{\Sigma}) \times D)$ that is formulated over the alphabet $\hat{\Sigma} \subseteq \Sigma$ and the color set $D \supseteq \bigcup_{i=1}^n D_i =: D'$.

We propose to perform the supervisor synthesis in two steps. First, supervisors $S_i : L(H_i) \rightarrow Pwr(\Sigma_i) \times Pwr(D_i - C_i)$ are designed for the local plant components such that $\Lambda_{D_i}(S_i/H_i) = SupCSNB(A_{i,D_i}, H_i, D_i)$ for $i = 1, \ldots, n$. Then, the joint behavior of the locally controlled plants is modeled by $G := ||_{i=1}^n S_i/H_i$. It has to be noted that G is usually not SNB w.r.t. D', which will be addressed in the following synthesis step. In addition, it holds that maximal permissiveness of the control can be lost by the local synthesis as is stated for the case of a single marking in (Schmidt and Breindl, 2010).

For illustration, we investigate H_1 and H_2 in Fig. 1 with the color sets $C_1 = \{c_1\}$, $C_2 = \{c_2\}$ and the uncontrollable events $\Sigma_{1,\mathrm{uc}} = \Sigma_{2,\mathrm{uc}} = \{\alpha\}$ (controllable events are marked by a tick). The overall plant $H := H_1||H_2$ has the color set $C := C_1 \cup C_2$. Then, the local liveness specification $\Lambda_{C_1}(D_1||H_1)$ is not controllable w.r.t. $L(H_1)$, whereas $\Lambda_C(D_1||H)$ is controllable w.r.t. $L(H)$. Here, maximal permissiveness is violated since the local supervisor synthesis lacks the information that α cannot occur after $\mathtt{a} \in L(H_1)$ because of the synchronization with H_2.

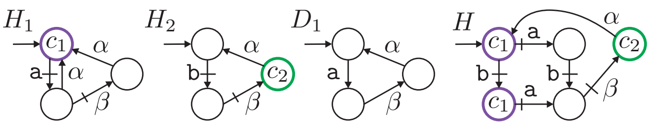

Fig. 1. Illustration of mutual controllability.

This situation can be avoided if the plant components are *mutually controllable* (MC) (Lee and Wong, 2002). [1]

Definition 3. The CMGs H_i and H_j are MC if

$$L(G_i)(\Sigma_{j,\mathrm{u}} \cap \Sigma_i) \cap p_{\Sigma_i \cup \Sigma_j \rightarrow \Sigma_i}(p^{-1}_{\Sigma_i \cup \Sigma_j \rightarrow \Sigma_j}(L(G_j)) \subseteq L(G_i)$$

$$L(G_j)(\Sigma_{i,\mathrm{u}} \cap \Sigma_j) \cap p_{\Sigma_j \cup \Sigma_i \rightarrow \Sigma_j}(p^{-1}_{\Sigma_j \cup \Sigma_i \rightarrow \Sigma_i}(L(G_i)) \subseteq L(G_j)$$

Lemma 1. Let H_i and H_j be MC for $i, j = 1, \ldots, n$ and write $A'_{D'} := ||_{i=1}^n A_{i,D_i}$. Then,

$$SupCSNB(A'_{D'}, H, D') \subseteq ||_{i=1}^n SupCSNB(A_{i,D_i}, H_i, D_i).$$

That is, the local control is not conservative if MC is fulfilled. Moreover, if MC does not hold for two components H_i and H_j, their composition $H_i||H_j$ can be used in order to avoid the MC violation.

[1] The proofs for all statements are provided in (Schmidt et al., 2010).

3.2 Hierarchical Abstraction

In the second synthesis step, we proceed as in (Schmidt et al., 2007) to ensure SNB. We compute a *high-level plant* $\hat{H}$ over an alphabet $\hat{\Sigma} \subseteq \Sigma$: each locally controlled plant component S_i/H_i is abstracted to the alphabet $\hat{\Sigma}_i := \hat{\Sigma} \cap \Sigma_i$, $i = 1, \ldots n$. With the set $\Sigma_{i,\cap} := \bigcup_{k=1,k\neq i}^n (\Sigma_i \cap \Sigma_k)$ of *shared events* of component H_i, it is required that $\Sigma_{i,\cap} \subseteq \hat{\Sigma}_i$. Using the natural projections $\hat{p}_i : \Sigma_i^* \rightarrow \hat{\Sigma}_i^*$, $p : \Sigma^* \rightarrow \hat{\Sigma}^*$ and the colored projections $\hat{m}_i : Pwr(Pwr(\Sigma_i^*) \times D_i) \rightarrow Pwr(Pwr(\hat{\Sigma}_i^*) \times D_i)$, $m : Pwr(Pwr(\Sigma^*) \times D') \rightarrow Pwr(Pwr(\hat{\Sigma}^*) \times D')$, $\hat{H}$ is defined such that

$$L(\hat{H}) = ||_{i=1}^n \hat{p}_i(L(S_i/H_i)) = p(L(G)),$$
$$\Lambda_{D'}(\hat{H}) = ||_{i=1}^n \hat{m}_i(\Lambda_{D_i}(S_i/H_i)) = m(\Lambda_{D'}(G)).$$

The uncontrollable events are chosen as $\hat{\Sigma}_{\mathrm{uc}} := \Sigma_{\mathrm{uc}} \cap \hat{\Sigma}$.

The abstraction process is illustrated on the right-hand side of Fig. 2. Next, $\hat{A}_D$ is used as a high-level specification, and an SNB coloring *high-level supervisor* $\hat{S} : L(\hat{H}) \rightarrow Pwr(\hat{\Sigma}) \times Pwr(E)$ for $E := D - D'$ is computed such that $\Lambda_D(\hat{S}/\hat{H}) = SupCSNB(\hat{A}_D, \hat{H}, D)$. The control action of the corresponding *low-level supervisor* $S : L(G) \rightarrow Pwr(\Sigma) \times Pwr(E)$ is then defined for each $s \in L(G)$ as

$$S(s) := \big(\hat{S}(p(s)) \cup (\Sigma - \hat{\Sigma}), \mathcal{I}(\hat{S}(p(s)))\big).$$

Thus, the overall closed loop is characterized by $L(S/G) = L(\hat{S}/\hat{H})||L(G)$ and $\Lambda_D(S/G) = \Lambda_D(\hat{S}/\hat{H})||\Lambda_{D'}(G)$.

The control action of S is shown in Fig. 2.

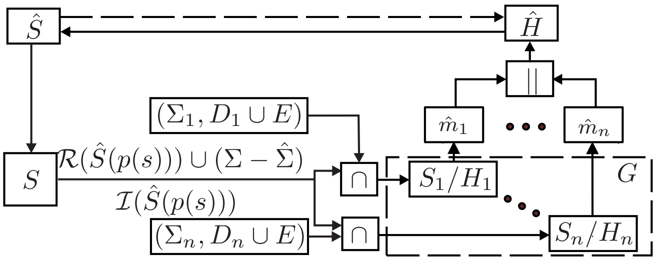

Fig. 2. Hierarchical and decentralized control architecture.

It is shown by (Schmidt et al., 2007) that the overall closed loop is SNB if each $\hat{m}_i$, $i = 1, \ldots, n$ is a colored observer. However, that condition is not sufficient for maximal permissiveness as explained in Fig. 3 with the CMG G and its abstraction $\hat{H}$ over $\hat{\Sigma} = \{\alpha, \beta, \gamma\}$. Here, the occurrence of $\gamma \in \Sigma_{\mathrm{uc}} = \{\beta, \gamma, \mathtt{a}, \mathtt{b}, \mathtt{d}, \mathtt{f}\}$ cannot be disabled in $\hat{H}$, although γ could be prevented in G by disabling the controllable local events $\mathtt{c}$ and $\mathtt{g}$.

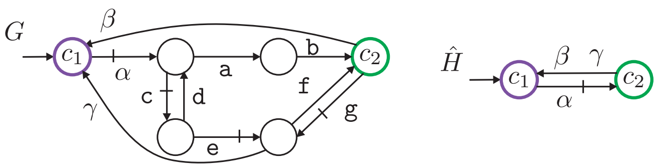

Fig. 3. Violation of maximally permissive control.

To resolve this issue, we show that maximally permissive control holds if p fulfills the additional requirement of *local control consistency* (LCC) (Schmidt and Breindl, 2010).

Definition 4. Let $L = \overline{L} \subseteq \Sigma^*$ be a prefix-closed language, Σ_{uc} be a set of uncontrollable events and $\hat{\Sigma} \subseteq \Sigma$. The projection $p : \Sigma^* \rightarrow \hat{\Sigma}^*$ is locally control consistent (lcc) for L and Σ_{uc} if for all $s \in L$ and for all $\sigma_{\mathrm{uc}} \in \Sigma_{\mathrm{uc}} \cap \hat{\Sigma}$ with $p(s)\sigma_{\mathrm{uc}} \in p(L)$, it holds that either $\nexists u \in (\Sigma - \hat{\Sigma})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L$ or $\exists u \in (\Sigma_{\mathrm{uc}} - \hat{\Sigma})^*$ s.t. $su\sigma_{\mathrm{uc}} \in L$.

That is, if there is an uncontrollable extension of a string $p(s)$ in $p(L)$, not all corresponding extensions of s in the original language L should contain controllable events such that it is not possible to locally disable such strings. We now state the main result of this section.

Theorem 2. If all H_i, H_j, $i \neq j$ are MC, $\hat{m}_i$ is an $\Lambda_{D_i}(S_i/H_i)$-observer and p_i is lcc for $L(S_i/H_i)$ and $\Sigma_{i,\mathrm{uc}}$, $i = 1, \ldots, n$, then the overall closed loop $\hat{S}/\hat{H}||(||_{i=1}^{n} S_i/H_i)$ is SNB and maximally permissive:

$$\forall d \in D : \overline{L_d\big(\Lambda_D(\hat{S}/\hat{H})||\Lambda_{D'}(G)\big)} = L(\hat{S}/\hat{H})||L(G),$$

$$\Lambda_{\max} := SupCSNB(\hat{A}_D||A'_{D'}, H, D) = \Lambda_D(\hat{S}/\hat{H})||\Lambda_{D'}(G).$$

Hence, the described hierarchical architecture is suitable for SNB and maximally permissive control. Moreover, is has to be noted that our approach directly extends to a multi-level hierarchy by considering high-level closed loops as low-level plants for the next hierarchical level.

4. COMPUTATIONS FOR PROJECTIONS

In practice, efficient algorithms are needed to either verify if the conditions stated in the previous section hold for given colored projections or to compute appropriate colored projections. This section addresses both issues for a CMG $H = (Q, \Sigma, \delta, C, \chi, q_0)$, the abstraction alphabet $\hat{\Sigma} \subseteq \Sigma$, the uncontrollable alphabets $\Sigma_{\mathrm{uc}} \subseteq \Sigma$ and $\hat{\Sigma}_{\mathrm{uc}} := \Sigma_{\mathrm{uc}} \cap \hat{\Sigma}$, the natural projection $p : \Sigma^* \rightarrow \hat{\Sigma}^*$ and the colored projection $m : Pwr(Pwr(\Sigma^*) \times C) \rightarrow Pwr(Pwr(\hat{\Sigma}^*) \times C)$.

4.1 Verification

We generalize a result by (Wong and Wonham, 2004) that is stated for a single marking to the case of multiple markings considered in this paper.

Theorem 3. Let H be a CMG and define the dynamic system $\tilde{H}_{\mathrm{MT}} = (Q, \{\Delta_\sigma | \sigma \in \hat{\Sigma}\} \cup \{\Delta_c | c \in C\})$ with

$\Delta_\sigma : Q \rightarrow 2^Q$:$q \mapsto \{\delta(q, u\sigma u') | uu' \in (\Sigma - \hat{\Sigma})^*\}$,

$\Delta_c : Q \rightarrow 2^Q$:$q \mapsto \{\delta(q, u) | u \in (\Sigma - \hat{\Sigma})^* \wedge c \in \chi(\delta(q, u))\}$.

Denote the coarsest quasi-congruence (QC) for $\tilde{H}_{\mathrm{MT}}$ as

$$\mu^*_{\mathrm{MT}} := \sup\{\mu \in \mathcal{E}(Q) | \mu \leq \bigwedge_{\sigma \in \hat{\Sigma}} (\mu \circ \Delta_\sigma) \wedge \bigwedge_{c \in C} (\mu \circ \Delta_c)\}.$$

m is a $\Lambda_C(H)$-observer iff the quotient CMG (QCMG) $H_{\mu^*_{\mathrm{MT}}, \hat{\Sigma}}$ is deterministic and without any σ_0-transitions.

The quasi-congruence μ^*_{MT} groups states with the same observed future event extensions (Δ_σ) and reachable colors (Δ_c) in the same equivalence class.

The colored observer verification is illustrated by the CMG H with the color set $C = \{c_1, c_2\}$ in Fig. 4. We first consider $\hat{\Sigma}_1 = \{\alpha, \beta\}$ and determine the dynamic system $\tilde{H}_{1,\mathrm{MT}} = (Q, \{\Delta_\alpha, \Delta_\beta, \Delta_{c_1}, \Delta_{c_2}\})$ with the coarsest QC $\mu^*_{1,\mathrm{MT}}$ as indicated by the shaded areas in Fig. 4. Here, the transitions with events in $\hat{\Sigma}_1$ and colors in C correspond to the maps Δ_σ and Δ_c in Theorem 3, respectively. It can be seen that the corresponding QCMG $H_{\mu_{1,\mathrm{MT}}, \hat{\Sigma}_1}$ contains σ_0 transitions such that $m_1 : Pwr(Pwr(\Sigma^*) \times C) \rightarrow Pwr(Pwr(\hat{\Sigma}_1) \times C)$ is not a $\Lambda_C(H)$-observer. In contrast, $m_2 : Pwr(Pwr(\Sigma^*) \times C) \rightarrow Pwr(Pwr(\hat{\Sigma}_2) \times C)$ with $\hat{\Sigma}_2 = \{\alpha, \beta, \mathtt{a}, \mathtt{b}\}$ is a $\Lambda_C(H)$-observer.

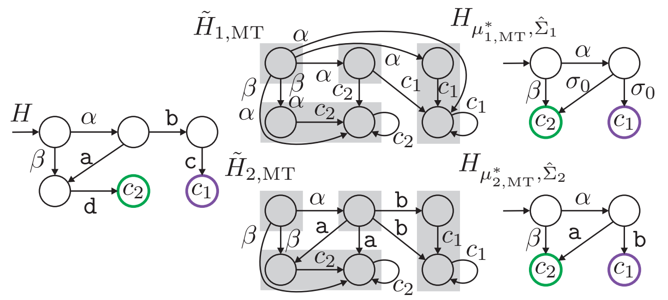

Fig. 4. Colored observer verification.

Furthermore, it is shown in (Schmidt and Breindl, 2010) that LCC can be described by a dynamic system $\tilde{H}_{\mathrm{LCC}}$.

Lemma 4. Define the dynamic system $\tilde{H}_{\mathrm{LCC}} = (Q, \{\Delta_\sigma | \sigma \in \hat{\Sigma}\} \cup \{\Delta_{\sigma_{\mathrm{uc}},\mathrm{LCC}} | \sigma_{\mathrm{uc}} \in \hat{\Sigma}_{\mathrm{uc}}\})$, with $\Delta_{\sigma_{\mathrm{uc}},\mathrm{LCC}} : Q \rightarrow 2^Q$ s.t.

$$\Delta_{\sigma_{\mathrm{uc}},\mathrm{LCC}}(q) := \begin{cases} \bigcup_{\sigma \in \hat{\Sigma}} \Delta_\sigma(q) & \text{if } \exists u \in (\Sigma_{\mathrm{uc}} - \hat{\Sigma}_{\mathrm{uc}})^* \text{ s.t. } \\ & \delta(q, u\sigma_{\mathrm{uc}}) \text{ exists or } \nexists u \in (\Sigma \\ & -\hat{\Sigma})^* \text{ s.t. } \delta(q, u\sigma_{\mathrm{uc}}) \text{ exists} \\ \emptyset & \text{otherwise.} \end{cases}$$

p is an $L(H)$-observer and lcc for $L(H)$ and Σ_{uc} iff $H_{\mu^*_{\mathrm{LCC}}, \Sigma_0}$ is deterministic and contains no σ_0-transitions, whereby μ^*_{LCC} is the coarsest QC for $\tilde{H}_{\mathrm{LCC}}$.

Theorem 3 and Lemma 4 allow the joint verification of the colored observer condition and LCC.

Corollary 5. Let H be a CMG, Σ_{uc} be the uncontrollable alphabet and define the dynamic system $\tilde{H}_{\mathrm{MT,LCC}} = (Q, \{\Delta_\sigma | \sigma \in \hat{\Sigma}\} \cup \{\Delta_c | c \in C\} \cup \{\Delta_{\sigma_{\mathrm{uc}},LCC} | \sigma_{\mathrm{uc}} \in \hat{\Sigma}_{\mathrm{uc}}\})$. Let $\mu^*_{\mathrm{MT,LCC}}$ be the coarsest QC for $\tilde{H}_{\mathrm{MT,LCC}}$. Then, m is a $\Lambda_C(H)$-observer and p is lcc for $L(H)$ and Σ_{uc} iff $H_{\mu^*_{\mathrm{MT,LCC}}, \hat{\Sigma}}$ is deterministic and without σ_0-transitions.

Hence, it is only required to determine the coarsest QC for $\tilde{H}_{\mathrm{MT,LCC}}$ in order to verify the colored observer condition and LCC which can be done with a complexity of $\mathcal{O}(|Q|^3 \cdot |\delta|)$ (Wong and Wonham, 2004), where $|Q|$ and $|\delta|$ denote the number of states and transitions of H, respectively.

4.2 Computation of Projections

We finally consider the case, where the verification of the colored observer condition and/or LCC according to Corollary 5 fails for a given CMG H over the alphabet Σ, an abstraction alphabet $\hat{\Sigma} \subseteq \Sigma$ and a set of uncontrollable events $\Sigma_{\mathrm{uc}} \subseteq \Sigma$. Then, it is desired to find an appropriate extension of $\hat{\Sigma}$ in order to fulfill the sufficient conditions for SNB and maximally permissive hierarchical control. To this end, we employ the following generalization of the *event set extension* algorithm in Feng and Wonham (2009) that is first proposed in (Schmidt and Breindl, 2010).

Algorithm 1. **Input:** H, $\hat{\Sigma}$

1. Compute the QC $\mu^*_{\mathrm{MT,LCC}}$ and the QCMG $H_{\mu^*_{\mathrm{MT,LCC}},\hat{\Sigma}}$.
2. **if** $H_{\mu^*_{\mathrm{MT,LCC}},\hat{\Sigma}}$ is deterministic without σ_0-transitions
   return $\hat{\Sigma}$
   else
   event set extension of $\hat{\Sigma}$ based on $H_{\mu^*_{\mathrm{MT,LCC}},\hat{\Sigma}}$ as in (Feng and Wonham, 2009) and **go to** 1.

The algorithm is based on the computation of the QCMG $H_{\mu^*_{\mathrm{MT,LCC}},\hat{\Sigma}}$ that is performed in step 1. Then, either the verification of the colored observer condition and LCC according to Corollary 5 is successful and the current alphabet $\hat{\Sigma}$ is returned, or an extension of $\hat{\Sigma}$ is required. In the latter case, the suboptimal event set extension algorithm in Feng and Wonham (2009) is applied. It adds events to $\hat{\Sigma}$ in order to remove σ_0-transitions and resolve the possible nondeterminism in $H_{\mu^*_{\mathrm{MT,LCC}},\hat{\Sigma}}$. The observer algorithm iterates until an appropriate alphabet extension is found. Its complexity is determined by the complexity $\mathcal{O}(|Q|^4 \cdot |\delta|^3)$ of the event set extension.

5. MANUFACTURING SYSTEM EXAMPLE

5.1 Description

We consider a slight modification of the flexible manufacturing system (FSM) in (Schmidt et al., 2007). Its aim is to produce blocks with conical and cylindrical pins from raw blocks and raw pegs. Fig. 5 gives an overview of the system and Fig. 6 shows the CMG models of the system components. Here, C1 and C2 represent conveyor belts that allow the input of blocks and pegs to the system, M is a mill that produces a hole in a block, L is a lathe that can produce a conical or cylindrical pin from a peg and R is a robot that governs the exchange of parts among the system components. R is also connected to the conveyor belt C3 and the assembly machine AM. C3 transports cylindrical pins to the painting device PD, while AM assembles pins and blocks in order to obtain the ready products that are indicated by the colors co and cy in H_{AM}.[2]

The system restrictions are represented by buffers (B) between the system components that can hold one part and should neither overflow nor underflow. In our framework, they are expressed by colored specification behaviors that are modeled by the CMGs $M_{\mathrm{B}i}$, $i = 1, \ldots, 7$, in Fig. 7. In addition, M_{B3}, M_{B4}, M_{B6} and M_{B7} introduce restrictions on the sequential system behavior. The specified colors e_i, $i = 1, \ldots, 7$ require that all buffers can become empty independently, while the color o in M_{B3} and M_{B4} ensures that parts can be processed simultaneously in M and L.

The synchronous composition of the eight plant CMGs leads to a CMG H with 3 456 states and color set $C = \{\mathrm{co}, \mathrm{cy}\}$, and the overall specification behavior is represented by a CMG $M := H||(||_{i=1}^{7} M_{\mathrm{B}i})$ with 2 560 states and the color set $D = \{\mathrm{e}_i | 1 \leq i \leq 7\} \cup \{\mathrm{o}, \mathrm{co}, \mathrm{cy}\}$. The resulting maximally permissive colored behavior $SupCSNB(\Lambda_D(M), G, D)$ leads to a monolithic closed loop with 36 360 states.

[2] For a more detailed description, please consult (de Queiroz et al., 2005; Schmidt et al., 2007).

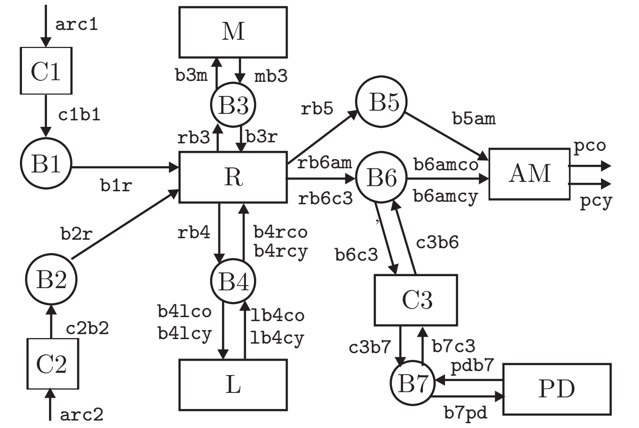

Fig. 5. Flexible manufacturing system (FMS) overview.

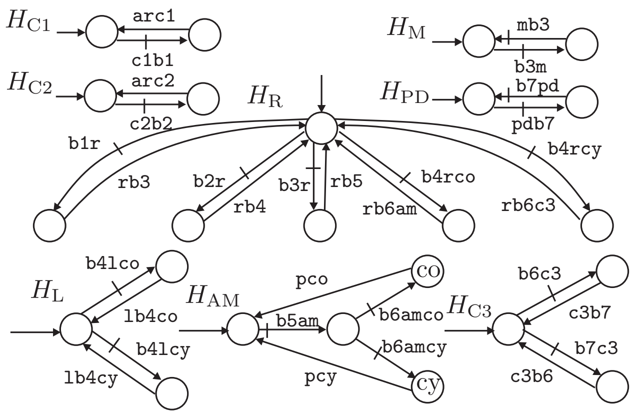

Fig. 6. Flexible manufacturing system models.

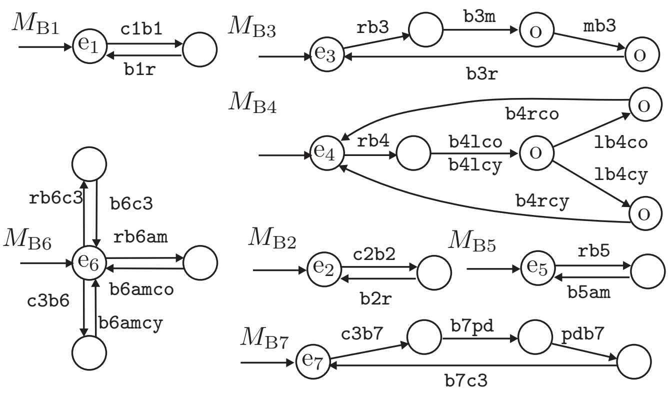

Fig. 7. Buffer specifications for the FMS

5.2 Hierarchical Synthesis

We now apply the hierarchical synthesis framework described in Section 3 in order to reduce the computational effort. All computations are carried out by the "multi-tasking" plug-in of the software library `libFAUDES` (lib, 2009) for DES. For comparison, we both employ colored observers and colored observers that additionally fulfill LCC for the hierarchical abstraction. The hierarchical architecture used in this example is shown in Fig. 8, and the plant components are listed in Table 1. It can be verified that all components are mutually controllable according to Definition 3. Regarding the hierarchical synthesis, $\hat{H}_i$ represents the abstraction of S_i/H_i, $i = 1, \ldots, 7$ that is computed using the algorithm in Section 4.2 for the case of colored observers and additional LCC. In all cases, the initial abstraction alphabet for each component is chosen as the set of shared events with the other components/specifications as required in Section 3.2.

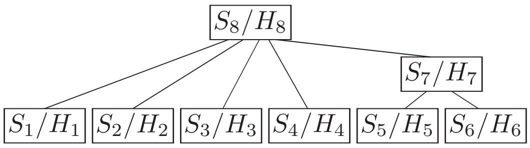

Fig. 8. Hierarchy of closed-loop systems.

Table 1. Supervisor computation for the FMS

closed loop	plant	specification	state count SNB/SNB & LCC
G_1	$H_1 := G_{\mathrm{C1}}\|\|G_{\mathrm{R}}$	$H_1\|\|M_{B_1}$	18/18
G_1	$H_2 := G_{\mathrm{C2}}\|\|G_{\mathrm{R}}$	$H_2\|\|M_{B_2}$	18/18
G_3	$H_3 := G_{\mathrm{M}}\|\|G_{\mathrm{R}}$	$H_3\|\|M_{B_3}$	18/18
G_4	$H_4 := G_{\mathrm{L}}\|\|G_{\mathrm{R}}$	$H_4\|\|M_{B_4}$	21/21
G_5	$H_5 := G_{\mathrm{AM}}\|\|G_{\mathrm{R}}$	$H_5\|\|M_{\mathrm{B5}}$	44/44
G_6	$H_6 := G_{\mathrm{C3}}\|\|G_{\mathrm{PD}}$	$H_6\|\|M_{\mathrm{B7}}$	6/6
G_7	$H_7 := \hat{H}_5\|\|\hat{H}_6$	$H_7\|\|M_{\mathrm{B6}}$	156/184
G_8	$H_8 := \|\|_{i=1}^4 \hat{H}_i\|\|\hat{H}_7$	H_8	1224/2904

For example, the closed loop S_6/H_6 for $H_6 = G_{\mathrm{C3}}||G_{\mathrm{PD}}$ and the specification $H_6||M_{\mathrm{B7}}$ is depicted in Fig. 9. Choosing the initial abstraction alphabet $\hat{\Sigma}_6 = \{\mathtt{b7c3}, \mathtt{c3b7}\}$ (shared events with M_{B6}), The computation of a colored observer and a colored observer with LCC yields the abstractions $\hat{H}_6$ in Fig. 9. It can be seen that adding LCC as a sufficient condition for maximally permissive control potentially leads to larger abstracted models.

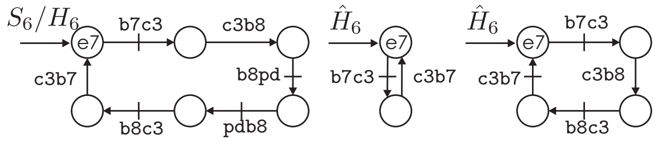

Fig. 9. G_6 and $\hat{H}_6$ without (left) and with (right) LCC.

The state counts for the remaining closed loops are listed in Table 1. In both cases, the plant H_8 is already strongly nonblocking, i.e., no supervisor has to be implemented on the highest level. Hence, 1 505/3 213 supervisor states are required to implement the control as opposed to the 36 360 states of the monolithic control. Furthermore, the trade-off between the size of the abstraction and maximal permissive control is observed in the component G_6. Here, the abstraction $\hat{H}_6$ of C3 and PD loses the information that a cylindrical pin that is located in PD can be prevented from moving to B6. Hence, different from the maximally permissive behavior, no conical pin is allowed to enter B6 whenever a cylindrical pin is present at PD.

6. CONCLUSION

This paper introduced sufficient conditions for the synthesis of maximally permissive modular supervisors that respect liveness requirements in a multitasking DES with a decentralized and hierarchical control architecture. The conditions apply to natural projections defined over the modular system models and subsets of their event sets. Algorithms were provided to check those conditions for given projections and, in case they fail, modify the projections by extending their set of projected events, in order to obtain suitable abstractions. The results have been illustrated in a multi-level control architecture for a flexible manufacturing system. Although the above results allow to synthesize optimal modular supervisors for systems with a large number of states, we believe that additional gains could be obtained by additionally eliminating redundant tasks as discussed in (Schmidt and Cury, 2009). This topic will be addressed in our future research.

REFERENCES

(2009). libFAUDES – software library for discrete event systems. URL `http://www.rt.eei.uni-erlangen.de/FGdes/faudes/index.html`.

de Queiroz, M.H., Cury, J.E.R., and Wonham, W.M. (2005). Multitasking supervisory control of discrete-event systems. *Journal on Discrete Event Dynamic Systems: Theory and Applications*, 15, 375–395.

Feng, L. and Wonham, W. (2009). On the computation of natural observers in discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*. URL `http://dx.doi.org/10.1007/s10626-008-0054-3`.

Feng, L. and Wonham, W. (2008). Supervisory control architecture for discrete-event systems. *Automatic Control, IEEE Transactions on*, 53(6), 1449–1461.

Hill, R. and Tilbury, D. (2008). Incremental hierarchical construction of modular supervisors for discrete-event systems. *International Journal of Control*, 81(9), 1364–1381.

Leduc, R., Lawford, M., and Wonham, W. (2005). Hierarchical interface-based supervisory control-part II: parallel case. *Automatic Control, IEEE Transactions on*, 50(9), 1336–1348.

Lee, S.H. and Wong, K. (2002). Structural decentralised control of concurrent DES. *European Journal of Control*, 35, 1125–1134.

Schmidt, K. and Breindl, C. (2010). Maximally permissive hierarchical control of decentralized discrete event systems. *to appear in Automatic Control, IEEE Transactions on*.

Schmidt, K. and Cury, J.E.R. (2009). Redundant tasks in multitasking control of discrete event systems. In *Workshop on Dependable Control of Discrete Systems*.

Schmidt, K., de Queiroz, M.H., and Cury, J.E.R. (2007). Hierarchical and decentralized multitasking control of discrete event systems. In *Decision and Control, IEEE Conference on*, 5936–5941.

Schmidt, K., de Queiroz, M.H., and Cury, J.E.R. (2010). Hierarchical multitasking control of discrete event systems: Computation of projections and maximal permissiveness. In *Technical Report, Cankaya University and Federal University of Santa Catarina*.

Schmidt, K., Moor, T., and Perk, S. (2008). Nonblocking hierarchical control of decentralized discrete event systems. *Automatic Control, IEEE Transactions on*, 53(10), 2252–2265.

Su, R., van Schuppen, J.H., and Rooda, J.E. (2010). Modular supervisory control of discrete-event systems with abstraction and incremental hierarchical construction. *Automatic Control, IEEE Transactions on*.

Wong, K. and Wonham, W.M. (2004). On the computation of observers in discrete-event systems. *Discrete Event Dynamic Systems*, 14(1), 55–107.