Decentralized Supervision of Petri Nets

Marian V. Iordache and Panos J. Antsaklis

***Abstract*—This note extends previous results on the supervision of Petri nets (PNs) to the decentralized setting. While focusing on the extension of supervision based on place invariants (SBPI), the proposed approach is more general and could be applied to other types of supervision as well. We begin by introducing d-admissibility as an extension to the decentralized setting of the centralized admissibility concept. We define also structural d-admissibility, as the counterpart of the simple sufficient conditions for centralized admissibility in the context of the SBPI. Note that (structural) d-admissibility is only sufficient for a specification to be enforcible with the same permissiveness as in the centralized setting with full controllability and observability. However, structural d-admissibility can be checked with low polynomial complexity. Based on the d-admissibility concept, we propose two suboptimal methods to design decentralized supervisors. The first method is to find a centralized solution, and then distribute the centralized supervisory policy by means of communication. The amount of communication can be minimized by means of an integer linear program (ILP). The second method is to transform the specification to a (more restrictive) d-admissible specification by means of an ILP. In the case of decentralized supervision with communication, the ILP can be used to minimize the amount of communication required by the solution.**

***Index Terms*—Decentralized control, Petri nets, supervisory control.**

I. Introduction

We consider Petri net (PN) structures of the form $\mathcal{N} = (P, T, F, W)$, where P is the set of places, T the set of transitions, F the set of transition arcs, and W the weight function. A decentralized supervisor $\mathcal{S}$ consists of a set of supervisors $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_n$, operating in parallel, such that a given specification is satisfied. We will denote the supervisor $\mathcal{S}$ by $\bigwedge_i \mathcal{S}_i$, since a transition may be fired if no supervisor $\mathcal{S}_i$ disables it. Each supervisor can control (observe) a subset $T_{c,i}(T_{o,i})$ of the transitions T. The set of transitions that are uncontrollable (unobservable) to $\mathcal{S}_i$ are denoted by $T_{uc,i} = T \setminus T_{c,i}(T_{uo,i} = T \setminus T_{o,i})$.

Let $(\mathcal{N}, T_{c,1}, \ldots T_{c,n}, T_{o,1}, \ldots T_{o,n})$ denote the system $\mathcal{N}$ with sets of controllable and observable transitions $T_{c,1}, T_{c,2}, \ldots T_{c,n}$ and $T_{o,1}, T_{o,2}, \ldots T_{o,n}$. The main problem considered in this note is *given a system* $(\mathcal{N}, T_{c,1}, \ldots T_{c,n}, T_{o,1}, \ldots T_{o,n})$ *and a* global *specification, find* $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_n$ *whose simultaneous operation guarantees that the global specification is satisfied*. Note that the problem does not give specifications for each supervisor $\mathcal{S}_i$, but rather a global specification that needs to be somehow decomposed in specifications for each supervisor $\mathcal{S}_i$.

Depending on whether communication between the supervisors $\mathcal{S}_i$ is allowed or not, we can distinguish between several cases: supervision with no communication, supervision with constraints on the communication between the supervisors, and supervision with unrestricted communication. If communication is available, two types of information can be exchanged. A supervisor $\mathcal{S}_i$ can communicate with another supervisor $\mathcal{S}_j$ to observe a transition $t \in T_{o,j} \setminus T_{o,i}$ or to control a transition $t \in T_{c,j} \setminus T_{c,i}$. Examples of factors that could restrict communication are bandwidth limitations and lack of communication links between certain supervisors. To address such constraints, communication costs are associated with each transition.

The type of specifications considered in this note are

$$L\mu \leq b \tag{1}$$

where $L \in \mathbb{Z}^{n_c \times |P|}$, $b \in \mathbb{Z}^{n_c}$, and μ is the marking of $\mathcal{N}$. An overview of the centralized enforcement of (1) as well as some other necessary definitions are included in Section II. D-admissibility is introduced in Section III. For d-admissible constraints (1), a decentralized supervisor can be easily obtained, without any need for communication. However, one way to enforce d-inadmissible constraints is to enable the supervisors $\mathcal{S}_i$ to remotely control/observe transitions that are not directly available, communication achieving a virtual d-admissibility. Then, an integer linear program (ILP) can be used to minimize the communication cost. This approach is considered in Section IV. The approach is suboptimal, as the strategy of creating virtual d-admissibility may not lead to the least communication cost. In the cases in which communication is restricted or unavailable, the design of decentralized supervisors is more difficult, but can be approached using an ILP, as shown in Section V. The approach is suboptimal, as it may not produce the least restrictive solution, when it exists. Section VI concludes our presentation with a manufacturing example adapted from [1].

The decentralized supervision of discrete event systems (DES) has been studied for automata models. For PN models, distributed supervision has been considered in [2], for specifications given from the beginning in a distributed form. We are not aware of any other related work on PN models. Compared to related work on automata models, note that d-admissibility is not equivalent to controllability and coobservability [3], as a d-inadmissible specification could still be feasible. On the other hand, structural d-admissibility can be verified with low polynomial complexity. Note also that the complexity of the methods presented here depends on the size of the PN, not on the reachability graph (which defines the equivalent automaton of a PN), which may not even be finite. As in [4], communication consists of events rather than state estimates or observation strings [5], [6]. The vast majority of the decentralized control papers consider language specifications. Here we consider specifications (1), which are a particular type of state predicates. The relation between state predicate specifications and language specifications is as follows: any language can be represented as a state predicate on a system consisting of the plant and a "memory" DES [7]. In the automata setting, the existence of a decentralized solution enforcing state predicates is studied in [8]. In the context of PN models, the enforcement of (1) has been studied by numerous authors, such as [9]–[12]. The decentralized control of DES has been proposed for various applications, including manufacturing [1], [2], failure detection [13], and communication protocols [14]. An earlier version of our work can be found in [15].

II. Preliminaries

Given a PN $\mathcal{N}$ of marking μ, a *constraint* will be denoted by $l\mu \leq c$ with $l \in \mathbb{Z}^{1 \times |P|}$ and $c \in \mathbb{Z}$, while a *set of constraints* by $L\mu \leq b$ with $L \in \mathbb{Z}^{n_c \times |P|}$, $b \in \mathbb{Z}^{n_c}$, and $n_c \geq 1$. Note that $\mathcal{N}$ represents the **plant**. The supervision based on place invariants (SBPI) provides a supervisor in the form of a PN $\mathcal{N}_s = (P_s, T, F_s, W_s)$ with

$$D_s = -LD \tag{2}$$

$$\mu_{0,s} = b - L\mu_0 \tag{3}$$

Manuscript received August 1, 2003; revised July 18, 2005. Recommended by Associate Editor R. Boel. This work was supported in part by the Lockheed Martin Corporation, in part by the National Science Foundation under Grant NSF ECS99-12458, and in part by DARPA/IXO-NEST Program AF-F30602-01-2-0526.

M. V. Iordache is with the School of Engineering and Engineering Technology, LeTourneau University, Longview, TX 75607-7001 USA (e-mail: MarianIordache@letu.edu).

P. J. Antsaklis is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: antsaklis.1@nd.edu).

Digital Object Identifier 10.1109/TAC.2005.863894

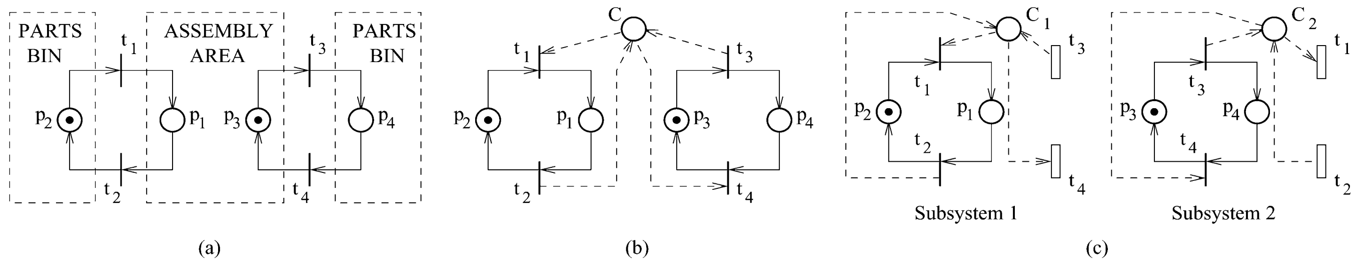

Fig. 1. (a) PN model. (b) Centralized supervision. (c) Decentralized supervision.

where D is the incidence matrix of the plant $\mathcal{N}, \mathcal{D}_f$ is the incidence matrix of the supervisor, $\mu_{0,s}$ the initial marking of the supervisor, and μ_0 is the initial marking of $\mathcal{N}$. The places of the supervisor are called **control places**. The supervised system, that is the **closed-loop** system, is a PN of incidence matrix $D_c = [D^T(-LD)^T]^T$. As an example, the control places C_1 and C_2 in Fig. 3 enforce $\mu_1 + \mu_2 + \mu_3 \leq 1$ and $\mu_4 + \mu_5 + \mu_6 \leq 1$, respectively.

Let μ_c be the marking of the closed-loop, and let $\mu_c|_{\mathcal{N}}$ denote μ_c restricted to the plant $\mathcal{N}$. $t \in T$ is **closed-loop enabled** if μ_c enables t; t is **plant-enabled**, if $\mu_c|_{\mathcal{N}}$ enables t in $\mathcal{N}$. The supervisor **detects** t if t is closed-loop enabled at some reachable marking μ_c and firing t changes the marking of some control place. The supervisor **controls** t if there is a reachable marking μ_c such that t is plant-enabled but not closed-loop enabled. Given μ_c, the supervisor **disables** t if there is a control place C such that $(C, t) \in F_s$ and $\mu_c(C) < W_s(C, t)$. A supervisor is admissible, if it only controls controllable transitions and it only detects observable transitions. The constraints $L\mu \leq b$ are **admissible** if the supervisor defined by (2)–(3) is admissible. When inadmissible, the constraints $L\mu \leq b$ are transformed (if possible) to an admissible form $L_a\mu \leq b_a$ such that $L_a\mu \leq b_a \Rightarrow L\mu \leq b$ [11]. Then, the supervisor enforcing $L_a\mu \leq b_a$ is admissible, and enforces $L\mu \leq b$ as well. A plant $\mathcal{N}$ with sets of controllable and observable transitions T_c and T_o will be denoted by $(\mathcal{N}, T_c, T_o)$.

To illustrate the decentralized setting, consider a manufacturing system in which two robots transport parts to a common assembly area. The system is modeled by the PN of Fig. 1(a), where $\mu_2 = 1 (\mu_4 = 1)$ when robot 1 (robot 2) is in the parts bin, and $\mu_1 = 1 (\mu_3 = 1)$ when robot 1 (robot 2) robot is in the assembly area. Thus, the system consists of two subsystems, corresponding to the two robots. The sets of controllable transitions of the two subsystems are $T_{c,1} = \{t_1, t_2\}$ and $T_{c,2} = \{t_3, t_4\}$. The sets $T_{o,1}$ and $T_{o,2}$ can also be defined, describing the transitions that can be observed in each subsystem.

III. Decentralized Admissibility

To distinguish between centralized and decentralized admissibility, we call the former *c-admissibility* and the latter *d-admissibility*. Let μ_0 denote the initial marking of the plant.

Definition 1: Given $(\mathcal{N}, \mu_0, T_{c,1} \ldots T_{c,n}, T_{o,1} \ldots T_{o,n})$, a constraint is **d-admissible** if there is a set $\mathcal{C} \subseteq \{1, 2, \ldots n\}, \mathcal{C} \neq \emptyset$, such that the constraint is c-admissible with respect to $(\mathcal{N}, \mu_0, T_c, T_o)$, where $T_c = \bigcup_{i \in \mathcal{C}} T_{c,i}$ and $T_o = \bigcap_{i \in \mathcal{C}} T_{o,i}$. A set of constraints is **d-admissible** if each of its constraints is d-admissible.

It is important to notice that the definition of d-admissibility does not require the sets $T_{o,i}$ to share common transitions, as the sets $\mathcal{C}$ may be a singletons. However, the definition can take advantage of situations in which there are sets $T_{o,i}$ that are not disjoint. As an example, consider the PN structure of Fig. 1(a) for $T_{uc,1} = T_{uo,1} = \{t_3, t_4\}, T_{uc,2} = T_{uo,2} = \{t_1, t_2\}$, and initial marking $\mu_0 = [1,\, 1,\, 1,\, 1]^T$. Thus, $T_{o,1}$ and $T_{o,2}$ are disjoint. Notice that the set of constraints $L\mu \leq b$ describing $\mu_1 \leq 1$ and $\mu_4 \leq 1$ is d-admissible: The constraint $\mu_1 \leq 1$ satisfies Definition 1 for $\mathcal{C} = \{1\}$ and $\mu_4 \leq 1$ for $\mathcal{C} = \{2\}$. An example of a constraint that is not d-admissible is $\mu_1 + \mu_3 \leq 1$. Now, assuming $T_{uo,1} = T_{uo,2} = \emptyset, \mu_1 + \mu_3 \leq 1$ is d-admissible for $\mathcal{C} = \{1, 2\}$, while not c-admissible with respect to any of $(\mathcal{N}, T_{c,1}, T_{o,1})$ or $(\mathcal{N}, T_{c,2}, T_{o,2})$.

The construction of a decentralized supervisor enforcing d-admissible constraints is illustrated on the PN of Fig. 1(a) with $T_{c,1} = \{t_1, t_2\}, T_{c,2} = \{t_3, t_4\}, T_{uo,1} = T_{uo,2} = \emptyset, \mu_0 = [0,\, 1,\, 1,\, 0]^T$, and specification $\mu_1 + \mu_3 \leq 1$. The centralized solution is shown in Fig. 1(b). The constraint is d-admissible for $\mathcal{C} = \{1, 2\}$. Given two variables $x_1, x_2 \in \mathbb{N}$, a decentralized supervisor $\mathcal{S}_1 \wedge \mathcal{S}_2$ enforcing $\mu_1 + \mu_3 \leq 1$ can be defined by the following rules.

Operation of $\mathcal{S}_1$

- Initialize $x_1 = 0$.
- Disable t_1 if $x_1 = 0$.
- If t_2 or t_3 fires, $x_1 = x_1 + 1$.
- If t_1 or t_4 fires, $x_1 = x_1 - 1$.

Operation of $\mathcal{S}_2$

- Initialize $x_2 = 0$.
- Disable t_4 if $x_2 = 0$.
- If t_2 or t_3 fires, $x_1 = x_1 + 1$.
- If t_1 or t_4 fires, $x_2 = x_2 - 1$.

Note that $\mathcal{S}_1$ and $\mathcal{S}_2$ differ only in the second rule: one disables t_1, while the other t_4. A graphical representation of $\mathcal{S}_1$ and $\mathcal{S}_2$ is possible, once we reexamine the meaning of arcs going from control places to transitions. In Fig. 1, $\mathcal{S}_1$ is represented by C_1 and $\mathcal{S}_2$ by C_2; x_1 is the marking of C_1 and x_2 the marking of C_2. Graphically, C_1 and C_2 are copies of the control place C of the centralized supervisor. However, (C_1, t_4) and (C_2, t_1) model observation, not control. This is due to the fact that $\mathcal{S}_1$ never disables t_4 and $\mathcal{S}_2$ never disables t_1. As C_1 and C_2 have the same initial marking as C, their markings stay equal at all times. So, whenever t_1 should be disabled, the disablement action is implemented by C_1, and whenever t_4 is to be disabled, the disablement action is implemented by C_2.

In the general case, the construction of a supervisor enforcing a d-admissible constraint $l\mu \leq c$ ($l \in \mathbb{N}^{1 \times |P|}$ and $c \in \mathbb{N}$) is as follows.

Algorithm 1: Supervisor Design for a D-Admissible Constraint

1) Let μ_0 the initial marking of $\mathcal{N}, \mathcal{C}$ the control place of the centralized SBPI supervisor $\mathcal{N}_s = (P_s, T, F_s, W_s)$ enforcing $l\mu \leq c$, and $\mathcal{C}$ the set of Definition 1.
2) For all $i \in \mathcal{C}$, let $x_i \in \mathbb{N}$ be a state variable of $\mathcal{S}_i$.
3) Define $\mathcal{S}_i$, for $i \in \mathcal{C}$, by the following rules:

- Initialize $x_i = c - l\mu_0$.
- If $t \in T_{c,i}, t \in C\bullet$ and $x_i < W_s(C, t)$, then $\mathcal{S}_i$ disables t.
- If t fires, $t \in T_{o,i}$ and $t \in \bullet C$, then $x_i = x_i + W_s(t, C)$.
- If t fires, $t \in T_{o,i}$ and $t \in C\bullet$, then $x_i = x_i - W_s(C, t)$.

To enforce a d-admissible set of constraints $L\mu \leq b$, the construction above is repeated for each constraint $l\mu \leq c$. Note that in our graphical representation, the supervisors $\mathcal{S}_i$ correspond to $|\mathcal{C}|$ copies of the control place C of the centralized supervisor, where each copy has the same initial marking as C.

As stated in the next result, Algorithm 1 provides supervisors that are feasible and maximally permissive. A decentralized supervisor $\mathcal{S}_d = \bigwedge_{i \in \mathcal{C}} \mathcal{S}_i$ is **feasible** if all $\mathcal{S}_i$ are feasible. $\mathcal{S}_i$ is **feasible** if $\mathcal{S}_i$ only observes transitions $t \in T_{o,i}$ and disables transitions t only if not plant-enabled or if $t \in T_{c,i}$. Further, let's notice that a decentralized supervisor attains maximal permissiveness when it is as permissive as the centralized supervisor of the construction (2)–(3). Let $\mathcal{S}$ be the supervisor of (2)–(3) (for simplicity, the same notation $\mathcal{S}$ is used both for the enforcement of sets of constraints $L\mu \le b$ and the enforcement of single constraints $l\mu \le c$.)

Theorem 1: The decentralized supervisor $\mathcal{S}_d$ constructed in Algorithm 1 is feasible, enforces the desired constraint, and is as permissive as the centralized supervisor $\mathcal{S}$.

Proof: Feasibility is an immediate consequence of Algorithm 1. The remaining part of the theorem can be proved by showing that a firing sequence σ is enabled by the centralized supervisor at the initial marking iff enabled by the decentralized supervisor at the initial marking. This follows easily from the observations that (a) at all markings μ and for all $i \in \mathcal{C} : x_i = l\mu - c$ and (b) the centralized supervisor is c-admissible with respect to $(\mathcal{N}, T_c, T_o)$, where T_c and T_o are from Definition 1. A complete proof can be found in [15]. ■

Let T_o^M be the set of transitions detected by $\mathcal{S}$ and T_c^M the set of transitions controlled by $\mathcal{S}$. For instance, in Fig. 1(b) $T_c^M = \{t_1, t_4\}$ and $T_o^M = \{t_1, t_2, t_3, t_4\}$. The d-admissibility of a constraint can be tested as follows.

Algorithm 2: Checking the d-admissibility of a constraint

1) Find T_o^M and T_c^M.
2) Let $\mathcal{C}$ be the set of indices i satisfying $T_{o,i} \supseteq T_o^M$.
3) If $\mathcal{C} = \emptyset$, declare the constraint not d-admissible and exit.
4) Define $T_c = \bigcup_{i \in \mathcal{C}} T_{c,i}$.
5) Does T_c satisfy $T_c \supseteq T_c^M$? If yes, declare the constraint d-admissible. Otherwise, declare the constraint not d-admissible.

A d-admissible constraint can be implemented using a minimal set $\mathcal{C}_{min} \subseteq \mathcal{C}$ of indices i such that $T_c^M \subseteq \bigcup_{i \in \mathcal{C}_{min}} T_{c,i}$. Further, checking whether a set of constraints is d-admissible involves checking each constraint individually with Algorithm 2.

Proposition 1: Algorithm 2 is correct.

Proof: A constraint is declared d-admissible if $\mathcal{C} \neq \emptyset$ and $T_c \supseteq T_c^M$. The definition of T_o^M and T_c^M implies that the constraint is c-admissible with respect to $(\mathcal{N}, T_c, T_o)$ (where $T_o = \bigcap_{i \in \mathcal{C}} T_{o,i}$). Then, in view of Definition 1, the algorithm is right to declare the constraint d-admissible. Next, assume a d-admissible constraint. Then, there is a set $\mathcal{C}' \neq \emptyset$ such that the constraint is c-admissible with respect to $(\mathcal{N}, T_c', T_o')$ (where $T_c' = \bigcup_{i \in \mathcal{C}'} T_{c,i}$ and $T_o' = \bigcap_{i \in \mathcal{C}'} T_{o,i}$). Then $T_o' \supseteq T_o^M$ and $T_c' \supseteq T_c^M$; $T_o' \supseteq T_o^M \Rightarrow \mathcal{C}' \subseteq \mathcal{C} \Rightarrow T_c \supseteq T_c' \Rightarrow T_c \supseteq T_c^M$. Consequently, the algorithm declares the constraint to be d-admissible. ■

In general, it may not be possible to compute the sets T_c^M and T_o^M without some reachability analysis. Alternatively, estimates $T_c^e \supseteq T_c^M$ and $T_o^e \supseteq T_o^M$ can be used instead of T_c^M and T_o^M. However, in this case the algorithm only checks a sufficient condition for d-admissibility, and so it can no longer detect constraints that are not d-admissible. In the case of the SBPI, a constraint $l\mu \le c$ is implemented by a control place C, as described by (2)–(3). Thus, some estimates T_c^e and T_o^e are $T_c^e = C\bullet$ and $T_o^e = \bullet C \cup C\bullet$. Here, T_c^e differs from T_c^M if there is $t \in C\bullet$ that is never both plant-enabled and closed-loop disabled in $(\mathcal{N}, \mu_0, \mathcal{S})$. Also, T_o^e differs from T_o^M if there is some $t \in \bullet C \cup C\bullet$ that is dead in $(\mathcal{N}, \mu_0, \mathcal{S})$. A constraint $l\mu \le c$ is **structurally d-admissible** if it satisfies the test of Algorithm 2 when $T_c^e = C\bullet$ and $T_o^e = \bullet C \cup C\bullet$ are used instead of T_c^M and T_o^M. Structural d-admissibility is the decentralized equivalent of the following admissibility conditions of [11]:

$$lD_{uc} \le 0 \text{ and } lD_{uo} = 0 \tag{4}$$

where D_{uc} and D_{uo} are the restrictions of the incidence matrix of the plant to the sets of uncontrollable and unobservable transitions, respectively. Structural d-admissibility coincides with d-admissibility in all examples considered in this section.

IV. Distributing Centralized Supervisory Policies

The previous section has shown that the design of supervisors enforcing d-admissible constraints can be done easily, as in Algorithm 1. It remains to consider the enforcement of constraints that are not d-admissible. Two main approaches are possible here. One is to solve the problem first in a centralized setting, by assuming all locally observable and controllable transitions as observable and controllable to a central supervisor. Then, a communication policy could be used for a decentralized implementation of the centralized solution. Alternatively, another approach is to solve the problem directly in the decentralized setting. The first approach is considered in this section, while the other will be treated in Section V.

In our decentralized setting, communication can be used to increase the sets $T_{c,i}$ and $T_{o,i}$ of transitions that a supervisor $\mathcal{S}_i$ can control and observe. This is achieved by transmitting control decisions/transition firings to/from a supervisor $\mathcal{S}_j$ that can control/observe the transition of interest. Any transition added by communication to $T_{c,i}(T_{o,i})$ and used by $\mathcal{S}_i$ for control (observation) is said to be remotely controlled (observed). Note that given a set $\mathcal{C}$, communication cannot increase $T_o = \bigcap_{i \in \mathcal{C}} T_{o,i}$ above the attainable upper bound $\overline{T_o} \supseteq T_o$, where $\overline{T_o} = \bigcup_{i=1}^{n} T_{o,i}$. In the same way, $T_c = \bigcup_{i \in \mathcal{C}} T_{c,i}$ cannot be increased above $\overline{T_c} = \bigcup_{i=1}^{n} T_{c,i}$. Indeed, $T \setminus \overline{T_c}(T \setminus \overline{T_o})$ is the set of transitions uncontrollable (unobservable) to all supervisors $\mathcal{S}_j$.

We begin with an algorithm that uses only the communication of transition firings, without resorting to the transmission of control decisions. We assume that we start with a specification that is c-admissible with respect to $(\mathcal{N}, \overline{T_c}, \overline{T_o})$. If that is not the case, the specification could be transformed to a (more restrictive) c-admissible form using one of the literature approaches, such as [11].

Algorithm 3: Decentralized supervisor design with local control

1) Let $\mathcal{S}$ be the centralized SBPI supervisor enforcing the specification. Let T_{cs} be the set of transitions controlled by $\mathcal{S}$ and T_{os} the set of transitions detected by $\mathcal{S}$.
2) Find a set $\mathcal{C}$ such that $T_c = \bigcup_{i \in \mathcal{C}} T_{c,i} \supseteq T_{cs}$.[1]
3) In view of the d-admissibility requirement that $\bigcap_{i \in \mathcal{C}} T_{o,i} \supseteq T_{os}$, the communication is designed as follows: For all $t \in T_{os} \cap (\bigcup_{i \in \mathcal{C}} T_{uo,i})$, a supervisor $\mathcal{S}_j$ such that $t \in T_{o,j}$ transmits the firings of t to all supervisors $\mathcal{S}_k$ with $t \in T_{uo,k}$ and $k \in \mathcal{C}$.
4) Design the decentralized supervisor by applying Algorithm 1 to $\mathcal{N}, \mathcal{C}$ and $T_{o,i} = T_{o,i} \cup T_{os} \forall i \in \mathcal{C}$.

Algorithm 3 identifies the class of solutions that can be obtained when only observations of transition firings are communicated. There are other types of solutions, as illustrated next. Assume that in Fig. 1(a) $T_{c,1} = T_{o,1} = \{t_1, t_2\}, T_{c,2} = T_{o,2} = \{t_3, t_4\}$, and the specification is $\mu_1 + \mu_3 \le 1$. Algorithm 3 produces $\mathcal{C} = \{1, 2\}$ and requires t_1 and t_2 to be communicated by $\mathcal{S}_1$ to $\mathcal{S}_2$, and t_3 and t_4 by $\mathcal{S}_2$ to $\mathcal{S}_1$. $\mathcal{S}_1$ and $\mathcal{S}_2$ can be illustrated as in Fig. 1(c). However, this solution is not unique. For instance, another possibility would be to have all control decisions made by $\mathcal{S}_1$. Communication would ensure that $\mathcal{S}_1$ remotely controls t_4 and remotely observes t_3 and t_4. $\mathcal{S}_2$ would simply communicate t_3 and t_4 to $\mathcal{S}_1$ and execute the decisions of $\mathcal{S}_1$ concerning t_4. This solution is illustrated in Fig. 2. Either of these two solutions could

[1] At least one solution exists: $\mathcal{C} = \{1 \dots n\}$. Indeed, $\mathcal{S}$ admissible w.r.t. $(\mathcal{N}, \overline{T_c}, \overline{T_o})$ implies $T_{cs} \subseteq \overline{T_c}$, where $\overline{T_c} = \bigcup_{i=1}^{n} T_{c,i}$.

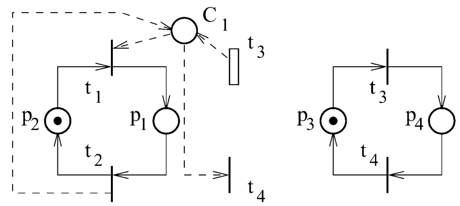

Fig. 2. Example of control with communication.

be optimal, depending on the relative cost of communicating observations versus control decisions. The following algorithm can be used to find the best solution, based on a cost function and an ILP

To characterize communication, let α_{ij} and ε_{ij} be binary variables defined as follows:

- $\alpha_{ij} = 1$ iff the transition t_j is communicated to $\mathcal{S}_i$;
- $\varepsilon_{ij} = 1$ iff the transition t_j is remotely controlled by $\mathcal{S}_i$.

Note that in the broadcast case $\alpha_{ij} = \alpha_j$ for all i, and $\varepsilon_{ij} = \varepsilon_j$ for all i, where the latter means that either all or none of the supervisors $\mathcal{S}_i$ are allowed to remotely control t_j. In practice, remote control could be implemented by allowing the supervisors to announce when their control decision (t_j enabled or t_j disabled) changes.

Assume that the specification has n_c constraints. Let $T_c^k(T_o^k)$ be the set of transitions controlled (detected) by the centralized SBPI supervisor that enforces the k-th constraint. Note that $T_c^k \subseteq T_{cs}$ and $T_o^k \subseteq T_{os}$. Let δ_{ik} denote binary variables indicating whether $\mathcal{S}_i$ participates in the control decision making for the constraint k. For instance, in Fig. 1(c) and Fig. 2 there is only one constraint (namely $\mu_1 + \mu_3 \leq 1$), so $k = 1$. In Fig. 1(c) $\delta_{11} = \delta_{21} = 1$, while in Fig. 2, $\delta_{11} = 1$ and $\delta_{21} = 0$, because $\mathcal{S}_2$ makes no control decisions.

If $\mathcal{S}_i$ participates in the control decision making for the constraint k (i.e., if $\delta_{ik} = 1$), then d-admissibility requires it to observe all transitions in T_o^k. This is written as

$$\alpha_{ij} \geq \delta_{ik} \qquad \forall j \in \{f : t_f \in T_o^k \setminus T_{o,i}\} \qquad \forall i = 1 \ldots n, \forall k = 1 \ldots n_c \tag{5}$$

Further, every transition $t \in T_c^k$ needs to be controlled by some $\mathcal{S}_i$. If $\mathcal{S}_i$ controls $t_j \in T_c^k$ and $t_j \notin T_{c,i}$, then we need $\varepsilon_{ij} = 1$. Formally, $\forall t_j \in T_c^k \exists i = 1 \ldots n : (\delta_{ik} = 1) \wedge [t_j \in T_{c,i} \vee (t_j \notin T_{c,i} \wedge \varepsilon_{ij} = 1)]$. This can be expressed by (6)–(7)

$$\forall k = 1 \ldots n_c : \quad \sum_{i=1}^{n} \delta_{ik} \geq 1 \tag{6}$$

$$\forall k = 1 \ldots n_c, \forall x = 1 \ldots n$$
$$\forall j \in \{y : t_y \in T_c^k\} : \quad \delta_{xk} \leq \varepsilon_{xj} + \sum_{i \in I_j} \delta_{ik} \tag{7}$$

where $I_j = \{i : t_j \in T_{c,i}\}$. Without loss of generality, we have assumed $T_c^k \neq \emptyset$. We can use an ILP to minimize a cost of the form

$$\min \sum_{i,j} \alpha_{ij} c_{ij} + \sum_{i,j} \varepsilon_{ij} f_{ij} + \sum_{i,k} \delta_{ik} h_{ik} \tag{8}$$

which penalizes communication and the number of supervisors implementing control. The minimization is subject to (5)–(7). As discussed later in Section V, additional inequalities for communication constraints could be incorporated in the ILP.

Algorithm 4: Design minimizing communication

1) Solve (8) subject to (5)–(7).
2) For each $k = 1 \ldots n_c$, apply Algorithm 1 on $\mathcal{N}$ with $\mathcal{C} = \{i : \delta_{ik} = 1\}, T_{c,i} = T_{c,i} \cup \{t_j : \varepsilon_{ij} = 1\}$, and $T_{o,i} = T_{o,i} \cup \{t_j : \alpha_{ij} = 1\}$.

The decentralization approach of this section could be used for more general specifications, such as modular language specifications, as long as the computation of the sets T_c^k and T_o^k is convenient. As mentioned in the previous section, structural analysis can easily provide estimates $T_c^{k,e} \supseteq T_c^k$ and $T_o^{k,e} \supseteq T_o^k$ for the SBPI.

V. Design With Constraint Transformations

A. Supervision Without Communication

In this section, we propose a method for the transformation of constraints that are not d-admissible to (more restrictive) constraints that are d-admissible. As an illustration, consider the PN of Fig. 1(a), this time with the initial marking $\mu_0 = [0, 3, 0, 3]^T, T_{c,1} = T_{o,1} = \{t_1, t_2\}, T_{c,2} = T_{o,2} = \{t_3, t_4\}$, and specification $\mu_1 + \mu_3 \leq 2$. It can be seen that there is no way to transform $\mu_1 + \mu_3 \leq 2$ to a single d-admissible constraint. However, we can transform it to the d-admissible set of constraints of $\mu_1 \leq 1$ and $\mu_3 \leq 1$, where $\mu_1 \leq 1$ is d-admissible for $\mathcal{C} = \{1\}$ and $\mu_3 \leq 1$ for $\mathcal{C} = \{2\}$.

In the general case, the problem can be stated as follows: *Given a set of constraints $L\mu \leq b$ that is not d-admissible and the sets $\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_m$, find sets of constraints $L_1\mu \leq b_1 \ldots L_m\mu \leq b_m$ d-admissible with respect to $\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_m$, respectively, such that*

$$(L_1\mu \leq b_1 \wedge L_2\mu \leq b_2 \wedge \ldots L_m\mu \leq b_m) \Rightarrow L\mu \leq b. \tag{9}$$

This framework includes the case when not all constraints $L_i\mu \leq b_i$ are necessary to implement $L\mu \leq b$, by allowing $L_i = 0$ and $b_i = 0$. Further, without loss of generality, $\mathcal{C}_1 \ldots \mathcal{C}_m$ are assumed to be given, not calculated. Indeed, we could include all possible groups $\mathcal{C}_i$, as their number is finite ($2^p - 1$ for p subsystems). This would guarantee that no possible solution of the form (9) is excluded. However, it may not be necessary to include all $\mathcal{C}_i$'s. Indeed, it is to be expected that in practice most $\mathcal{C}_i$'s have $T_o^{(i)} = \bigcap_{j \in \mathcal{C}_i} T_{o,j} = \emptyset$.

The problem is more tractable if the stronger condition

$$\left[\sum_{i=1}^{m} \lambda_i L_i \mu \leq \sum_{i=1}^{m} \lambda_i b_i\right] \Rightarrow L\mu \leq b \tag{10}$$

replaces (9), where λ_i are nonnegative scalars. Without loss of generality, (10) assumes that $L_1 \ldots L_m$ have the same number of rows. Again, without loss of generality, (10) can be replaced by

$$\left[\sum_{i=1}^{m} L_i \mu \leq \sum_{i=1}^{m} b_i\right] \Rightarrow L\mu \leq b. \tag{11}$$

We further simplify our problem to

$$L_1 + L_2 + \cdots L_m = R_1 + R_2 L \tag{12}$$

$$b_1 + b_2 + \cdots b_m = R_2(b + \mathbf{1}) - \mathbf{1} \tag{13}$$

for R_1 with nonnegative integer elements and R_2 diagonal with positive integers on the diagonal. Note that $[(R_1 + R_2 L)\mu \leq R_2(b + \mathbf{1}) - \mathbf{1}] \Rightarrow L\mu \leq b$ has been proved in [11].

Recalling (4), the admissibility requirements are written as

$$L_i D_{uc}^{(i)} \leq 0 \tag{14}$$

$$L_i D_{uo}^{(i)} = 0 \tag{15}$$

where $D_{uc}^{(i)}$ and $D_{uo}^{(i)}$ are the restrictions of D to the sets $T_{uc}^{(i)} = \bigcap_{i \in \mathcal{C}_i} T_{uc,i}$ and $T_{uo}^{(i)} = \bigcup_{i \in \mathcal{C}_i} T_{uo,i}$.

An ILP can be used to find a feasible solution to (12)–(15), where the unknowns are R_1, R_2, L_i, and b_i. In general, it is difficult to find constraints or a cost function that guarantee that the least restrictive solution is found, when a least restrictive solution exists. However, given a finite set $\mathcal{M}_I$ of markings of interest, it is possible to insure that the feasible space of the solution will include the markings of $\mathcal{M}_I$ by using the constraints

$$L_i M \leq b_i \mathbf{1}^T, \qquad i = 1 \ldots m \tag{16}$$

where $\leq$ means that each element of L_iM is less or equal to the element of the same indexes in $b_i\mathbf{1}^T$, M is a matrix whose columns are the markings of $\mathcal{M}_I$, and $\mathbf{1}^T$ is a row vector of appropriate dimension in which all elements are 1.

B. Supervision With Communication

Here, we extend the procedure of Section V-A to the case in which communication is possible. Communication is used to relax the admissibility constraints (14) and (15) by reducing the number of locally uncontrollable or unobservable transitions. However, this reduction may be limited by various communication constraints, such as bandwidth limitations. The framework of this section allows communication constraints to be incorporated in the design process, and can be used to minimize communication by defining a cost function. As in Section IV, we use the binary variables α_{ij} and ε_{ij} to describe the communication. Recall, $\alpha_{ij}=1$ iff the firings of t_j are communicated to $\mathcal{S}_i$, and $\varepsilon_{ij}=1$ iff $\mathcal{S}_i$ can remotely control the firings of t_j. Note that we have the following constraints:

$$\forall t_j \in T \setminus \overline{T_o} : \alpha_{ij} = 0 \tag{17}$$

for $\overline{T_o}=\bigcup_{i=1}^n T_{o,i}$, where $T\setminus\overline{T_o}$ is the set of transitions that cannot be observed anywhere in the system. Similarly

$$\forall t_j \in T \setminus \overline{T_c} : \varepsilon_{ij} = 0 \tag{18}$$

for $\overline{T_c}=\bigcup_{i=1}^n T_{c,i}$. For any practical purpose, we are not interested in unbounded solutions. So, we assume some upper and lower bounds are imposed on L_iD. Let B_L^i and B_U^i be the lower and upper bounds of L_iD. Note that B_L^i and B_U^i bound the weights of the arcs by which control places can be connected to the PN. In particular, if we desire the supervisor PN to be ordinary, we can set all elements of B_L^i to -1 and all elements of B_U^i to 1. In general, B_L^i and B_U^i can be set to arbitrary numbers. Given these bounds, (15) can be relaxed to

$$\begin{aligned} L_iD(\cdot,t_j) &\leq B_U^i(\cdot,t_j)\alpha_{xj} \\ L_iD(\cdot,t_j) &\geq B_L^i(\cdot,t_j)\alpha_{xj} \\ \forall t_j &\in T_{uo}^{(i)} \qquad \forall x \in X_{ij} \end{aligned} \tag{19}$$

where $X_{ij}=\{x\in\mathcal{C}_i : t_j\notin T_{o,x}\}$. This relaxes $L_iD_{uo}^{(i)}=0$ by eliminating the constraints corresponding to the transitions of $T_{uo}^{(i)}$ that have their firings communicated to the supervisors of $\mathcal{C}_i$. Similarly, (14) can also be relaxed by allowing the supervisors to remotely control transitions. Thus, if $t_j\in T_{uc}^{(i)}$, the admissibility requirement with respect to t_j can be relaxed when the remote control of t_j is allowed. Then, instead of (14) we have

$$L_iD(\cdot,t_j) \leq B_U^i(\cdot,t_j)\sum_{x\in\mathcal{C}_i}\varepsilon_{xj} \quad \text{if } t_j \notin \bigcup_{x\in\mathcal{C}_i} T_{c,x}. \tag{20}$$

Communication constraints stating that certain transitions cannot be remotely observed or controlled, can be incorporated by setting coefficients α_{ij} and ε_{ij} to zero. Constraints limiting the average network traffic can be incorporated as constraints of the form

$$\sum_{i,j}\alpha_{ij}g_{ij} + \sum_{i,j}\varepsilon_{ij}h_{ij} \leq p \tag{21}$$

where g_{ij}, h_{ij} and p are scalars. As an example, the coefficients g_{ij} could reflect average firing counts of the transitions over the operation of the system.

We may also choose to minimize the amount of communication involved in the system. Then, we can formulate our problem as

$$\min \sum_{i,j}\alpha_{ij}c_{ij} + \sum_{i,j}\varepsilon_{ij}f_{ij} \tag{22}$$

where the variables are $L_i, b_i, \alpha_{ij}, \varepsilon_{ij}, R_1$ and R_2, the coefficients c_{ij} and f_{ij} are given, and the minimization is subject to the constraints (12)–(13), (16)–(20), and $\alpha_{ij}, \varepsilon_{ij} \in \{0,1\}^{|T|}$. This problem can be solved using an ILP.

C. Liveness Constraints

A difficulty of this approach is that the permissiveness of the generated constraints can be hard to control. In the worst case, the generated constraints may cause parts of the system to unavoidably deadlock. Such a situation can be prevented by using a special kind of constraints, that we call liveness constraints.

A liveness constraint consists of a vector x such that for all $i : L_ix \leq 0$. A possible way to obtain such constraints is described next. Given a finite firing sequence σ, let x_σ be a vector such that $x_\sigma(i)$ is the number of occurrences of the transition t_i in σ. Given the PN of incidence matrix D and the constraints $L\mu \leq b$, let y be a nonnegative integer vector such that $Dy \geq 0$ and $-LDy \geq 0$. A vector y satisfying these inequalities has the following property. If σ is a firing sequence such that: i) σ can be fired without violating $L\mu \leq b$; and ii) $x_\sigma = y$, then σ can be fired infinitely often without violating $L\mu\leq b$. However, if the decentralized control algorithm generates a constraint $L_i\mu \leq b_i$ such that $L_iDy \not\leq 0$, then any firing sequence σ having $x_\sigma = y$ cannot be infinitely often fired in the closed-loop. If such a situation is undesirable, the matrices L_i can be required to satisfy $L_ix \leq 0$ for $x = Dy$. An illustration will be given in Section VI.

VI. Example

This section illustrates the approach of Section V on a manufacturing example adapted from [1]. The system is shown in Fig. 3. It consists of two machines (M_1 and M_2), four robots ($H_1 \ldots H_4$), and four buffers of finite capacity ($B_1 \ldots B_4$). The events associated with the movement of the parts within the system are marked with Greek letters. There are two types of parts. The manufacturing process of the first type of parts is represented by the following sequence of events: $\gamma_1\tau_1\pi_1\alpha_3\tau_3\pi_3\alpha_1\eta_1$. The manufacturing process of the second kind of parts is represented by $\gamma_2\tau_4\pi_4\alpha_2\tau_2\pi_2\alpha_4\eta_2$. These processes can be modeled by a PN, as shown in Fig. 3, in which the transitions are labeled by the events they represent.

The first supervisory requirements are that the buffers do not overflow. Assuming that the buffers B_1 and B_2 share common space, the requirement can be written as

$$\mu_3 + \mu_{13} \leq 2k \tag{23}$$

where $2k$ is the maximum number of parts that can be in B_1 and B_2 at the same time. Similarly, if the buffers B_3 and B_4 share a common space of the same capacity, the constraint is

$$\mu_6 + \mu_{10} \leq 2k. \tag{24}$$

Another requirement is that the number of completed parts of type 1 is about the same as the number of completed parts of type 2

$$v_8 - v_{16} \leq u \tag{25}$$

$$v_{16} - v_8 \leq u \tag{26}$$

where v_8 and v_{16} denote the number of firings of t_8 and t_{16}, respectively. Note that constraints involving the vector v can be easily represented as marking constraints in a transformed PN [16].

The constraints (23)–(24) are to be enforced assuming the following subsystems: $T_{c,1}=\{t_1\}$ and $T_{o,1}=\{t_1,t_2,t_3,t_4\}, T_{c,2}=\{t_4\}$ and $T_{o,2}=\{t_4,t_5,t_6,t_7,t_8\}, T_{c,3}=\{t_9\}$, and $T_{o,3}=\{t_9,t_{10},t_{11},t_{12}\}, T_{c,4}=\{t_{12},t_{15}\}$ and $T_{o,4}=\{t_{12},t_{13},t_{14},t_{15},t_{16}\}$. We take $\mathcal{C}_i=\{i\}$ for $i=1\ldots4$. Enforcing (23)–(24) for $k=2$ results in the control places C_1, C_2, C_3, and C_4 shown in Fig. 3. They

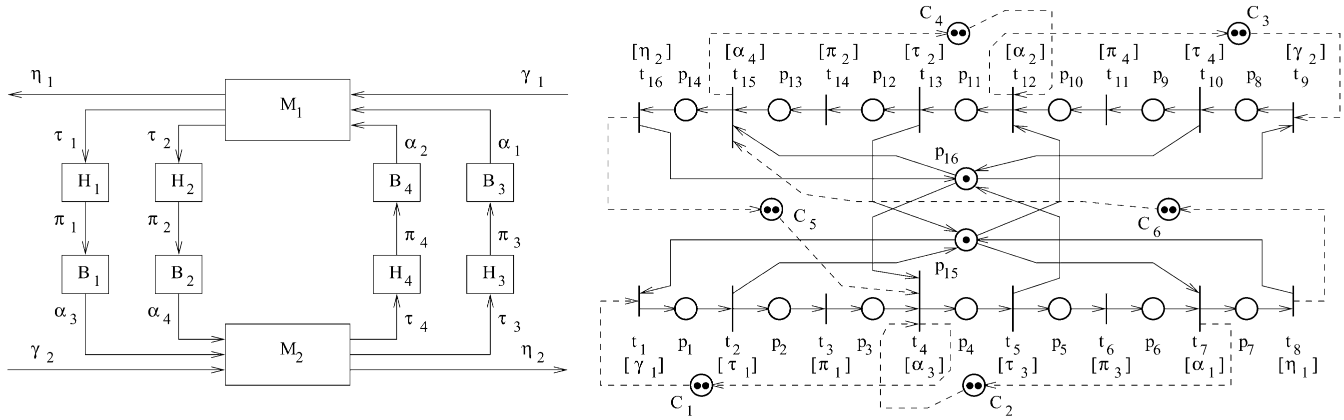

Fig. 3. Manufacturing system, PN model (solid line), and supervisor (dashed line).

correspond to the subsystems 1–4, respectively, and enforce $\mu_1 + \mu_2 + \mu_3 \leq 2, \mu_4 + \mu_5 + \mu_6 \leq 2, \mu_8 + \mu_9 + \mu_{10} \leq 2$, and $\mu_{11} + \mu_{12} + \mu_{13} \leq 2$.

In order to enforce (25) and (26), we need communication of events. Indeed, without communication there is no acceptable solution. For instance, a solution is to enforce $\mu_4 + \mu_5 + \mu_6 + \mu_7 + v_8 \leq u$ in subsystem 2 and $\mu_{14} + v_{16} \leq u$ in subsystem 4. However, this implies that the manufacturing system is constrained to produce no more than $2u$ parts! To exclude such solutions of the ILP, we can introduce liveness constraints. In this example, we can add the liveness constraints $L_i x \leq 0$ for $x = Dy$ and $y = [1, 1, \ldots 1]^T$. This is to prevent the constraints generated by the algorithm from blocking the firing sequence $t_1 t_2 \ldots t_{16}$ to occur infinitely often. However, with this liveness constraint and no communication, the problem becomes infeasible. Therefore, since communication is necessary, we are interested to minimize it. Assuming broadcast ($\alpha_{ij} = \alpha_j, \varepsilon_{ij} = \varepsilon_j$, for all i) and that the cost of remote control and remote observation is nonzero and equal (i.e., $c_{ij} = f_{ij}$ in (22), the following is an optimal solution:

$$\mu_4 + \mu_5 + \mu_6 + \mu_7 + v_8 - v_{16} \leq 2 \quad (27)$$

$$\mu_{14} + v_{16} - v_8 \leq 2 \quad (28)$$

which involves communicating the occurrences of t_8 and t_{16}. The constraint (27) is implemented in the subsystem 2, and the constraint (28) in the subsystem 4. In Fig. 3, the two constraints are enforced by the control places C_5 and C_6.

Finally, note that the ILP may have several solutions with the same cost but different permissiveness. For instance, we could have $\mu_{11} + \mu_{12} + \mu_{13} + \mu_{14} + v_{16} - v_8 \leq 2$ instead of (28). Then, a second ILP could be used to select a better solution, by minimizing the sum of the positive coefficients of the constraints, while requiring the other coefficients to stay less or equal to zero (the second ILP is also subject to the constraints of the first ILP and to a constraint that fixes the communication cost to the minimal value previously computed.)

VII. CONCLUSION

The design of decentralized supervisors is computationally easy for the class of specifications identified as d-admissible. When communication between the local supervisors is allowed, specifications that are not d-admissible can be enforced by solving first a centralized design problem, and then decentralizing the solution. The decentralized solution can be obtained by solving an ILP, which reduces the communication cost while maintaining the permissiveness of the centralized solution. Alternatively, specifications that are not d-admissible can also be enforced by constraint transformations. The constraint transformation approach has the benefit that it can be used in decentralized settings with no communication or with restricted communication. It allows also the minimization of the communication cost, when communication is available. The results of this note have been presented for specifications $L\mu \leq b$ and free-labeled Petri nets. Straightforward extensions to more general specifications and plant models are possible for many of the results presented in this note.

REFERENCES

[1] F. Lin and W. Wonham, "Decentralized control and coordination of discrete-event systems with partial observation," *IEEE Trans. Autom. Control*, vol. 35, no. 12, pp. 1330–1337, Dec. 1990.

[2] H. Chen and B. Hu, "Distributed control of discrete event systems described by a class of controlled Petri nets," in *Preprints of IFAC Int. Symp. Distributed Intelligence Systems*, 1991.

[3] K. Rudie and W. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1692–1708, Nov. 1992.

[4] K. Rudie, S. Lafortune, and F. Lin, "Minimal communication in a distributed discrete-event system," in *Proc. 1999 Amer. Control Conf.*, 1999, pp. 1965–1970.

[5] G. Barrett and S. Lafortune, "Decentralized supervisory control with communicating controllers," *IEEE Trans. Autom. Control*, vol. 45, no. 9, pp. 1620–1638, Sep. 2000.

[6] J. van Schuppen, "Decentralized supervisory control with information structures," in *Proc. Int. Workshop on DES*, 1998, pp. 36–41.

[7] Y. Li and W. Wonham, "Control of vector discrete-event systems I—the base model," *IEEE Trans. Autom. Control*, vol. 38, no. 8, pp. 1214–1227, Aug. 1993.

[8] S. Takai and S. Kozama, "Decentralized state feedback control of discrete event systems," *Syst. Control Lett.*, vol. 22, no. 5, pp. 369–375, 1994.

[9] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 1992, pp. 974–979.

[10] E. Yamalidou, J. O. Moody, P. J. Antsaklis, and M. D. Lemmon, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15–28, 1996.

[11] J. O. Moody and P. J. Antsaklis, "Petri net supervisors for DES with uncontrollable and unobservable transitions," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 462–476, Mar. 2000.

[12] G. Stremersch, *Supervision of Petri Nets*. Norwell, MA: Kluwer, 2001.

[13] R. Boel and J. van Schuppen, "Decentralized failure diagnosis with costly communication between diagnosers," in *Proc. 6th Int. Workshop on Discrete Event Systems*, 2002, pp. 175–181.

[14] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varayia, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Autom. Control*, vol. 33, no. 3, pp. 249–260, Mar. 1988.

[15] M. Iordache and P. Antsaklis, "Decentralized control of Petri nets," in *Proc. Workshop on DES Control, Int. Conf. on the Application and Theory of Petri Nets*, 2003, pp. 143–158.

[16] ——, "Synthesis of supervisors enforcing general linear vector constraints in Petri nets," *IEEE Trans. Autom. Control*, vol. 48, no. 11, pp. 2036–2039, Nov. 2003.