Control and Cybernetics

vol. **36** (2007) No. 1

Decentralized structural control approach for Petri nets

by

Aydın Aybar

Department of Electrical and Electronics Engineering
Anadolu University
26470 Eskişehir, Turkey
e-mail: aaybar@anadolu.edu.tr

Abstract: The structural controller, described by adding the control places to the Petri nets, is introduced in this work to lead the Petri net to the desired marking vectors. An algorithm (*Algorithm I*) is developed to determine the control places for the given Petri net. The connections and the initial marking of each control place are determined in this algorithm. Moreover, a decentralized structural control approach, based on overlapping decompositions, is introduced in this work. In this decentralized approach, all disjoint Petri subnets, which are obtained by using overlapping decompositions, are determined. The control places for each Petri subnet are determined by using the given algorithm. Then, the control places for the orginal Petri net are obtained by the control places of each PSN by another algorithm (*Algorithm II*) and these places are added to the orginal Petri net. Therefore, a decentralized structural controller which guarantees to lead the Petri net to the desired marking vectors is obtained.

Keywords: discrete-event systems, Petri nets, decentralized control, structural controller.

1. Introduction

Many supervisory controller design approaches using Petri nets have been presented in the literature (e.g., Holloway and Krogh, 1990; Sreenivas and Krogh, 1992; Giua and DiCasare, 1994; Ezpeleta et al., 1995; Barkaoui and Abdallah, 1995; Seenivas, 1997; Haoxun, 1998; Aybar and İftar, 2001, 2003b; Tordache et al., 2001; Iordache and Antsaklis, 2001; Aybar et al., 2005). Some approaches for the forbidden states were presented (for example, Sreenivas and Krogh, 1992; Sreenivas, 1997; Aybar and İftar, 2003b; Aybar et al., 2005; Uzam and Wonham, 2006). In these cited works, deadlock-freeness, liveness, reversibility, and boundedness, which are basic properties of Petri nets (Zhou and DiCesare, 1993) were considered to design the supervisory controllers.

The structural approaches were given to implement the controllers in Holloway and Krogh (1990), Haoxun (1998), Barkaoui and Abdallah (1995), Iordache et al. (2001), Iordache and Antsaklis (2001). In these related works, the control places were considered. The feedback logic control for a class of Petri nets, which contain the control places and the arcs only directed from the control places to the transitions (the output connection), was designed by these control places in Holloway and Krogh (1990). In another work (Haoxun, 1998), the control methodology based on S-Decreases was introduced for same class of Petri nets. The structural analysises (*trap, siphon, P-invariant*) were used for the control approaches (Barkaoui and Abdallah, 1995; Iordache et al., 2001; Iordache and Antsaklis, 2001). By using the control places which have the input and output connections, deadlock-freeness and liveness were guaranteed in these related works.

In this work, the structural control approach, which guarantees to lead the Petri net to the desired marking vectors is considered. It is assumed that any desired marking vector is reachable from the initial marking. Since the set of the desired marking vectors for the overall Petri net is used, this approach is called the centralized structural control approach in this work. This controller approach is explained so that the control places are added to the Petri nets. An algorithm, written by using pseudo-codes, is presented to determine the control places (the input and output connections and the initial marking of each place are determined). In the presented algorithm, *Algorithm I*, the important difference is that the set of desired marking vectors, instead of structural analysis (Barkaoui and Abdallah, 1995; Iordache et al., 2001; Iordache and Antsaklis, 2001) is used. Moreover, a program in the Microsoft Visual Basic programming language is developed to implement this algorithm. The definition of the Petri net is supplied using an ascii input file for this program. The results of the program are displayed on the screen.

After the centralized approach is introduced, the decentralized structural control approach is also presented to guarantee the occurrence of the desired marking vectors in this work. The decentralized control design has also been investigated for the discrete event systems (DESs) in many works (e.g., Ramadge and Wonham, 1989; Lin and Wonham, 1990; Rudie and Wonham, 1992; Cho and Lim, 1999; Aybar and İftar, 2003b; Aybar et al., 2005). This control design approach is chosen to reduce the computational complexity (for detailes about the computational complexity in DES, see Esparza and Nielsen, 1994; Aybar and İftar, 2003b; Aybar et al., 2005). The overlapping decompositions approach, which is used in the decentralized controller design was first introduced by Ikeda and Siljak (1980) for the continuous-state systems. This approach was then used for DESs by İftar and Özgüner (1998) for a state vector modeling, and Aybar and İftar (2002) for Petri net modeling.

In the decentralized approach, after all disjoint Petri subnets are obtained, the control places are determined for these subnets. Since each subnet is smaller than the overall Petri net, it is, in general, much easier to determine the control places for Petri subnets than to determine the control places for the overall original Petri net. Here, *Algorithm I* is used for the determination of the control places for each PSN. Finally, the control places for the given (orginal) Petri net are obtained by using the control places of each Petri subnets. This procedure is done by another algorithm, *Algorithm II*. We prove that this decentralized controller leads the original Petri net to the desired marking vectors.

2. Preliminaries

2.1. Petri net model

A Petri net is denoted as a five-tuple $G(P, T, N, O, m_0)$, where P is the set of *places*, T is the set of *transitions*, ($P \cap T = \emptyset$ and $P \cup T \neq \emptyset$), $N : P \times T \rightarrow \{0, 1\}$ is the *input matrix* that specifies the arcs directed from transitions to places, $O : P \times T \rightarrow \{0, 1\}$ is the *output matrix* that specifies the arcs directed from transitions to places and m_0 is the *initial marking*. It is assumed that all transitions are controllable in this work.

$M : P \rightarrow \{0, 1\}$ is a *marking vector*, $M(p)$ indicates the number of *tokens*, assigned by marking M to place p. A transition $t \in T$ is *enabled* if and only if $M(p) \geq N(p, t)$ for all $p \in P$. An enabled transition $t \in T$ can *fire* at M, yielding the new marking vector:

$$M'(p) = M(p) + O(p, t) - N(p, t), \quad \forall p \in P \ . \tag{1}$$

A *firing sequence* g is a sequence of enabled transitions $t_1 t_2 \ldots t_k$, where $t_1, t_2, \ldots, t_k \in T$. A marking M' is said to be *reachable* from M if there exist a firing sequence starting from M (i.e., the first transition of the sequence fires at M) and yielding M' (i.e., the final transition of the sequence yields M'). The set of all marking vectors, of G, reachable from M is denoted by $R(G, M)$. The *transition function*, denoted by $\rho(G, M, g)$, defines the yielded marking when the sequence g fires starting from M (ρ is in fact a partial function, since it is not defined, if g contains transitions which are not enabled). Note that we let $E(G, M)$ to denote the set of transitions which are enabled at $M \in R(G, m_0)$.

2.2. Overlapping decompositions and expansions

Overlapping decompositions and expansions of Petri nets were first introduced in Aybar and İftar (2002). To obtain an overlapping decomposition of a Petri net, overlapping subnets of a Petri net are first identified by examining the topological structure of the Petri net. These Petri subnets (PSNs) are identified such that the only interconnection between the subnets are through the overlapping part, i.e., no arc should be directed from one transition / place in one subnet to a place / transition in another subnet unless at least one of these transitions / places is in the overlapping part of the two subnets. Moreover, for any transition, t, in the overlapping part of two PSNs, any element of

$\mathcal{U}_T(t) := \{p \in P \mid O(p,t) = 1\}$ and of $\mathcal{V}_T(t) := \{p \in P \mid N(p,t) = 1\}$ must also be in the overlapping part of the same two PSNs. Moreover, for any place, p, in any overlapping part we must have $\mathcal{U}_P(p) := \{t \in T \mid N(p,t) = 1\} \subset T$.

The overlappingly decomposed Petri net is expanded as follows (Aybar and İftar, 2002):

i) A place or a transition in the overlapping part of n subnets is repeated n times and each repeated place / transition is assigned to a different subnet. Arcs between such places / transitions are also repeated and each repeated arc is assigned the same weight as the original arc.

ii) Two transitions with proper arcs are added between any two repeated places, such that each transition, when fires, transfers one token from one repeated place to the other.

iii) All the tokens which are initially assigned to a place in an overlapping part of the original Petri net are assigned to one of the repeated places corresponding to that place. Numbers of tokens in any place, which is not in any overlapping part, remain unchanged.

As a result of this procedure, an expanded Petri net (EPN), $\tilde{G}(\tilde{P}, \tilde{T}, \tilde{N}, \tilde{O}, \tilde{m}_0)$, which consists of S disjoint PSNs, is obtained from an original Petri net (OPN), $G(P, T, N, O, m_0)$, which was decomposed into S overlapping PSNs. Each place / transition / arc of a PSN corresponds to a place / transition / arc of the OPN. The PSNs are interconnected through the transitions introduced in step (ii) of the above procedure. Step (iii) of the above procedure determines the initial marking, $\tilde{m}_0$, of the EPN. The set of places of the EPN is given by $\tilde{P} := \bigcup_{i=1}^{S} P_i$, where P_i is the set of places of the i^{th} PSN. The set of transitions of the EPN is given by $\tilde{T} := \left(\bigcup_{i=1}^{S} T_i\right) \bigcup \mathcal{T}$, where T_i are, respectively, the sets of transitions of the i^{th} PSN, and $\mathcal{T}$ is the set of transitions between the PSNs (as introduced in step (ii) of the above procedure).

3. The control places

This section explains how the control places, which are the basis of the centralized approach, are determined. An algorithm (*Algorithm I*) is developed to determine the control places such that the connections of the control places to the transitions of the given nets are obtained and the number of token of each control places is determined.

Algorithm I requires the Petri net definition G and the set $R_d(G, m_0)$. Here, $R_d(G, m_0)$ denotes the set of the desired marking vectors. It is organized in three parts corresponding to the determination of the input matrix and the output matrix, and the initial marking. $N_c : P_c \times T \to \{0, 1\}$ denotes the input matrix for the control places, specifying the arcs directed from transitions to places, $O_c : P_c \times T \to \{0, 1\}$ denotes the output matrix for the control places, specifying the arcs directed from places to transitions and $M_{c_0} : P_c \to \{0, 1\}$ denotes the initial marking of the control places. Here, P_c denotes the set of the control places. In the first part of *Algorithm I*, the set $T_f = \{t \in T \mid \rho(G, M, t) \notin R_d(G, m_0),\ M \in R_d(G, m_0)\}$, which denotes the set of forbidden transitions, is constructed and then one control place is used for each element of T_f. It is known that, since any element of T_f is obtained as depending on any desired marking vector, this transition may be fired at any other desired vector. The physical connections of these places to the transitions are defined by the input matrix and the output matrix, described in the second part of *Algorithm I.* In the last part of *Algorithm I*, the initial marking of the new added control places, M_{c_0}, is determined.

Algorithm I: Control Place Algorithm

```
Pc := ∅, Tf := ∅, k = 1
/ Part I: Construct the input matrix for the control places /
For i = 1 to |Rd|
 Tx = E(G, [Rd]i)
 For j = 1 to |Tx|
  If ρ(G, [Rd]i, [Tx]j) ∉ Rd Then
   Tf ← Tf ∪ {[Tx]j}
   Pc ← PcÛ{pc^k}
   Nc(pc^k, [Tx]j) = 1
   k ← k + 1
  End
 End
End
For i = 1 to |Pc|
 For j = 1 to |T|
  If Nc(pc^j, [T]i) ≠ 1 Then
   Nc(pc^j, [T]i) = 0
  End
 End
End
/ Part II: Construct the output matrix for the control places /
For i = 1 to |Pc|
 For j = 1 to |Tf|
  For n = 1 to |Rd|
   Ty = E(G, [Rd]n)
   For r = 1 to |Ty|
    If Nc(pc^i, [Tf]j) = 1 And ρ(G, ρ(G, [Rd]n, [Ty]r), [Tf]j) ∈ Rd Then
     Oc(pc^i, [Ty]r) = 1
    End
   End
  End
 End
End
```

For $i = 1$ to $|P_c|$
 For $j = 1$ to $|T|$
  If $O_c(p_c^j, [T]_i) \neq 1$ Then
   $O_c(p_c^j, [T]_i) = 0$
  End
 End
End
/ Part III: Construct the initial marking for the control places /
For $j = 1$ to $|P_c|$
 $M_{c_0}(p_c^j) = 0$
End
For $i = 1$ to $|T_f|$
 For $j = 1$ to $|P_c|$
  If $N_c(p_c^j, [T_f]_i) = 1$ And $\rho(G, m_0, [T_f]_i) \in R_d$ Then
   $M_{c_0}(p_c^j) = 1$
  End
 End
End

The notation is used in this algorithm as follows: for a set X, $|X|$ denotes the number of the elements of X, and $[X]_i$ denotes the i^{th} element of X ($i = 1, 2, \ldots, |X|$). All the sets are assumed to be ordered sets. When a new element is added to a set of size n, the new element is taken as the $(n+1)^{\text{st}}$ element. Both $\cup$ and $\hat{\cup}$ are used to denote the *set union*. $A \mathbin{\hat{\cup}} B$ is used, rather than $A \cup B$, whenever it is known apriori that $A \cap B = \emptyset$. To evaluate $Z = A \cup B$, the set Z is first initialized as A; for each element (from first to last), b, of B, it is then checked whether $b \in A$. If $b \notin A$, then b is added to set Z. To evaluate $Z = A \mathbin{\hat{\cup}} B$, on the other hand, elements of A and of B are simply appended to form Z. For a vector M, $M(i)$ denotes the i^{th} element of M. For an n-dimensional vector A and a scalar a, $[A\ a]$ denotes the $(n+1)$-dimensional vector obtained by appending a to A.

This algorithm is implemented by a program in the Microsoft Visual Basic language. This program displays the input and output connections and the initial markings of all control places.

The Petri net G with the control places is called the *controlled Petri net* (CPN), denoted by the tuple $\bar{G}(\bar{P}, T, \bar{N}, \bar{O}, \bar{m}_0)$. The set of places is $\bar{P} := P \cup P_c$, the initial marking is $\bar{m}_0 = [m_0^T\ M_{c_0}^T]^T$ (here, $[.]^T$ denotes the transpose of $[.]$), and $\bar{N}$ / $\bar{O} := \bar{P} \times T \to \{0, 1\}$, denotes the input / output matrix such as

$$\bar{N}(p,t) := \begin{cases} N(p,t), & \text{if } p \in P \\ N_c(p,t), & \text{if } p \in P_c \end{cases}, \bar{O}(p,t) := \begin{cases} O(p,t), & \text{if } p \in P \\ O_c(p,t), & \text{if } p \in P_c \end{cases} \quad \text{for } t \in T.$$

We now prove that only desired states occur in CPN.

THEOREM 1 *If CPN is obtained by using the above method, then* $R^{-cp}(\bar{G}) = R_d(G, m_0)$. *Here,* $R^{-cp}(\bar{G}) := \{M^* : P \to \mathcal{N} \mid M^*(p) = \bar{M}(p),\ \forall p \in P,\ \forall \bar{M} \in R(\bar{G}, \bar{m}_0)\}$ *contains the marking vectors without the markings of the control places.*

Proof. Since the initial marking for CPN, $\bar{m}_0$, contains the initial marking of OPN, $m_0 \in R^{-cp}(\bar{G})$ and $m_0 \in R_d(G, m_0)$. It is known that any $M \in R_d(G, m_0)$ is reachable from m_0 and $R_d(G, m_0)$ is constructed starting from the initial marking, m_0.

1) For $m_0 \in R_d(G, m_0)$ and $\bar{m}_0 \in R(\bar{G}, \bar{m}_0)$.

a) If $M^o = \rho(G, m_0, t_i) \in R_d(G, m_0)$, $t_i \in E(G, m_0)$ (i.e., $m_0(p) = 1$, $\forall p \in \bullet t_i(G)$), then there is no control place or one control place, p_c^x with $\bar{m}_0(p_c^x) = 1$ obtained by using *Algorithm I* for the transition t_i. Here, $\bullet t_i(G)$ denotes the set of input places into the transition t in G.

1a-1) If there is no control place, then t_i is enabled at $\bar{m}_0$, because of $\bullet t_i(G) = \bullet t_i(\bar{G})$. $t_i \in E(\bar{G}, \bar{m}_0)$ and $\bar{M}^o = \rho(\bar{G}, \bar{m}_0, t_i) \in R(\bar{G}, \bar{m}_0)$ are obtained. In this case, there exists the marking vector $M^* \in R^{-cp}(\bar{G})$ for $\bar{M}^o$.

1a-2) If there exists a control place, p_c^x, with $\bar{m}_0(p_c^x) = 1$, then t_i is enabled at $\bar{m}_0$, since $\bar{m}_0(p) = 1$, $\forall p \in \bullet t_i(\bar{G}) = \bullet t_i(G) \cup \{p_c^x\}$. In this case, there exists the marking vector $M^* \in R^{-cp}(\bar{G})$ for $\bar{M}^o$.

We obtain $M^* = M^o$, $M^* \in R^{-cp}(\bar{G})$ and $M^o \in R_d(G, m_0)$.

b) If $\rho(G, m_0, t_i) \notin R_d(G, m_0)$, $t_i \in E(G, m_0)$, then the control place, p_c^b, ($\bar{m}_0(p_c^b) = 0$), which is connected to this transition t_i as the input is obtained by using *Algorithm I.* In this case, t_i is not enabled at $\bar{m}_0$, since $\bar{m}_0(p_c^b) = 0$ for $p_c^b \in \bullet t_i(\bar{G})$.

c) If $t_i \notin E(G, m_0)$, then $t_i \notin E(\bar{G}, \bar{m}_0)$, because of the structure of the net.

2) For $M^o \in R_d(G, m_0)$ and $\bar{M}^o \in R(\bar{G}, \bar{m}_0)$.

a) If $M' = \rho(G, M^o, t_j) \in R_d(G, m_0)$, $t_j \in E(G, M^o)$ (i.e., $M^o(p) = 1$, $\forall p \in \bullet t_j(G)$ for G), then there is no control place or one control place, p_c^y with $\bar{M}^o(p_c^y) = 1$, obtained by using *Algorithm I* for the transition t_j.

2a-1) If there is no control place, then t_j is enabled at $\bar{M}^o$, because of $\bullet t_j(G) = \bullet t_j(\bar{G})$. $t_j \in E(\bar{G}, \bar{M}^o)$ and $\bar{M}' = \rho(\bar{G}, \bar{M}^o, t_j) \in R(\bar{G}, \bar{m}_0)$ are obtained. In this case, there exists the marking vector $M^\ddagger \in R^{-cp}(\bar{G})$ for $\bar{M}'$.

2a-2) If there exist the control place, p_c^y, and $\bar{M}^o(p_c^y) = 1$, then t_j is enabled at $\bar{M}^o$. Since $t_j \in T_f$ and $M' = \rho(G, M^o, t_j) \in$

$R_d(G, m_0)$, $O_c(p_c^y, t_i) = 1$ are determined. Thus, t_j is enabled at $\bar{M}^o$. In this case, there exists the marking vector $M^\ddagger \in R^{-cp}(\bar{G})$ for $\bar{M}'$.

We obtain $M^\ddagger = M'$, $M^\ddagger \in R^{-cp}(\bar{G})$ and $M' \in R_d(G, m_0)$.

b) If $\rho(G, M^o, t_j) \notin R_d(G, m_0)$, $t_j \in E(G, m_0)$, then the control place, p_c^q ($\bar{M}^o(p_c^q) = 0$), which is connected to this transition t_j as the input is obtained by using *Algorithm I*. In this case, t_j is not enabled at $\bar{M}^o$, since $\bar{M}^o(p_c^q) = 0$ for $p_c^y \in \bullet t_j(\bar{G})$.

c) If $t_j \notin E(G, M^o)$, then $t_j \notin E(\bar{G}, \bar{M}^o)$, because of the structure of net.

There exists the marking vector, $\bar{M}^+ \in R(\bar{G}, \bar{m}_0)$ for $M^+ \in R_d(G, m_0)$ by using induction method from (1)-(2) such that $\bar{M}^+(p) = M^+(p)$, $\forall p \in P$. Therefore, the set $R^{-cp}(\bar{G})$ is equal to the set $R_d(G, m_0)$. ■

We consider the example Petri net, shown in Fig. 1. In this net, the set of places is $P = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, the set of transitions is $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, and the initial marking is $m_0 = [1\ 0\ 0\ 0\ 1\ 0]^T$ (it is known that deadlock occurs when t_1, t_4, t_2 fires at the initial marking in this Petri net).

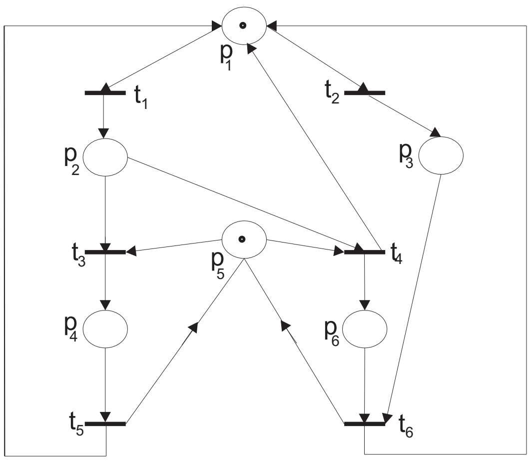

Figure 1. The example Petri net

The set of the desired markig vectors for this net is given as $R_d(G, m_0) = \{[1\,0\,0\,0\,1\,0]^T, [0\,1\,0\,0\,1\,0]^T, [0\,0\,0\,1\,0\,0]^T, [1\,0\,0\,0\,0\,1]^T, [0\,0\,1\,0\,0\,1]^T, [0\,0\,0\,0\,1\,0]^T\}$. To obtain the element of this set, *Algorithm I* can be used for this aim. When the algorithm is used, $P_c = \{p_c^1, p_c^2\}$, $T_f = \{t_1, t_2\}$, $N_c(p_c^1, t_1) = 1$, $N_c(p_c^2, t_2) = 1$, $N_c(p, t) = 0$, for $p \in P_c$, and $\forall t \in T \setminus \{t_1, t_2\}$, $O_c(p_c^1, t_5) = 1$, $O_c(p_c^1, t_6) = 1$, $O_c(p_c^2, t_4) = 1$, $O_c(p, t) = 0$, for $p \in P_c$, and $\forall t \in T \setminus \{t_5, t_6\}$, and $M_{c_0} = [1\ 0]^T$

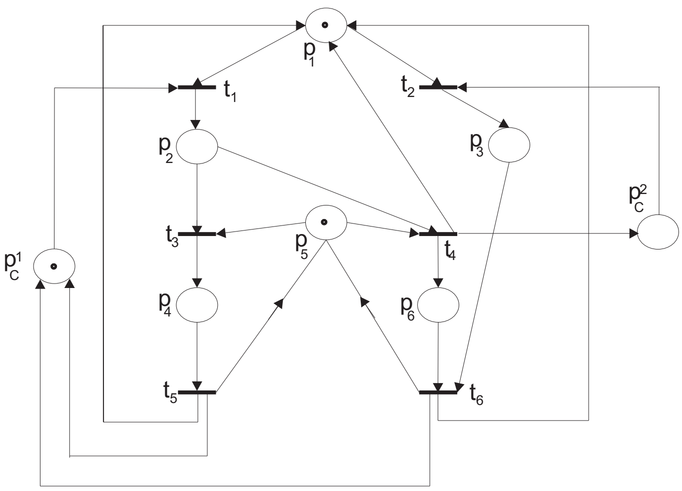

Figure 2. The controlled Petri net

are obtained. After the control places are determined for this Petri net, CPN is obtained (see Fig. 2). The control places effect the firing transitions and satisfy $R^{-cp}(\bar{G}) = R_d(G, m_0)$. Moreover, deadlock does not occur in CPN.

4. Decentralized structural control approach

Decentralized approach, which is based on overlapping decompositions of Petri nets to guarantee the occurence of the desired states is introduced in this section. It is assumed that an overlapping decomposition and a corresponding expansion, $\tilde{G}(\tilde{P}, \tilde{T}, \tilde{N}, \tilde{O}, \tilde{m}_0)$, of the given OPN, $G(P, T, N, O, m_0)$, is obtained as explained in Section 2-2.

In this approach, each PSN is considered in the determination of the control places. The k^{th} PSN is denoted by $G_k(P_k, T_k, N_k, O_k, m_{k_0})$. It is assumed that an initial marking, m_{k_0}, is the part of a valid initial marking of EPN (not necessarily $\tilde{m}_0$). The k^{th} controlled PSN with the control places is denoted by $\bar{G}_k(\bar{P}_k, T_k, \bar{N}_k, \bar{O}_k, \bar{m}_{k_0})$. Here, $\bar{P}_k := P_k \cup P_c^k$, $\bar{m}_{k_0} = [m_{k_0}^T \ M_{c_{k_0}}^T]^T$, $\bar{N}_k$ and $\bar{O}_k$ are the input and output matrices, $P_c^k = \{p_c^{k1}, p_c^{k2}, ...\}$ is the set of the control places for k^{th} PSN, $M_{c_{k_0}}$ is the initial marking for the control places in k^{th} PSN.

After all PSNs are obtained, the set of the desired marking vectors for each PSN is constructed as follow:

> $R_{k_d}(G_k, m_{k_0})$ contains the marking vector M_k such that $M_k(\tilde{p}) = M(p)$, $k \in \{1, ..., S\}$, $M \in R_d(G, m_0)$, $\tilde{p} \in \Pi(p) \cap P_k$, $\forall p \in P$.

Here, $\Pi(p)$ denotes the set of places, $\tilde{p} \in \tilde{P}$, of the EPN, which correspond to the place p, and $\hat{\Pi}(\tilde{p})$ denotes the place $p \in P$ of the OPN which corresponds to $\tilde{p} \in \tilde{p}$. If $m_{k_0} \notin R_{k_d}(G_k, m_{k_0})$, it is not possible to determine the control places for k^{th} PSN. Therefore, it is assumed that $m_{k_0} \in R_{k_d}(G_k, m_{k_0})$, $\forall k \in \{1, 2, ...S\}$ in this work.

There may exist a marking vector, $M_k^* \in R_{k_d}(G, m_0)$, such that M_k^* is not reachable from m_{k_0}. Since each desired marking vector of any PSN corresponds to a part of the desired marking vector of OPN, there exists a marking vector $M_k^+ \in R_{k_d}(G, m_0)$, such that $M_k^+(\tilde{p}) = M_k^*(\tilde{p})$, $\forall \tilde{p} \in P_k \setminus P_k^r$, where P_k^r denotes the set of repeated places in the k^{th} PSN (if there is one token in the common place of OPN, then one of the repeated places of this common place has one token and the remainder of the the repeated places have no tokens in EPN). Thus, we obtain $E(G_k, M_k^*) \subset E(G_k, M_k^+)$. Therefore, M_k^* does not effect the determination of the control places for the k^{th} PSN.

After the control places of each PSN are determined by using *Algorithm I*, the control places for OPN are determined by using control places of each PSN in *Algorithm II.* This algorithm requires the control places and their connections. *Algorithm II* constructs the input and output matrices for all control places which are added to OPN. The repetition of the control places in the given algorithm (*Algorithm II*) is prevented so that if $\bullet p_x(\bar{G}) = \bullet p_y(\bar{G})$ and $p_x \bullet (\bar{G}) = p_y \bullet (\bar{G})$, then P_c is updated as $P_c \leftarrow P_c \setminus \{p_y\}$, for $p_x, p_y \in P_c$. Here, $\bullet p(\bar{G})$ denotes the set of input transitions into the place p in $\bar{G}$, and $p \bullet (\bar{G})$ denotes the set of output transitions from the place p in $\bar{G}$.

Algorithm II: Decentralized Approach Algorithm

```
For i = 1 to S
 For j = 1 to |P_c^i|
  For k = 1 to |T_i|
   If N_{c_i}(p_c^{i j}, [T_i]_k) = 1 Then
    N_c(p_c^{i j}, Φ([T_i]_k)) = 1
   End
   If O_{c_i}(p_c^{i j}, [T_i]_k) = 1 Then
    O_c(p_c^{i j}, Φ([T_i]_k)) = 1
   End
  End
 End
End
```

Here, $\Phi(\tilde{t})$ denotes the place $t \in T$ of the OPN which corresponds to $\tilde{t} \in \tilde{T}$. The initial marking for the added control places is described as $M_{c_0}(p) = M_{c_{\mu(p)_0}}(p)$, $\forall p \in P_c$. Here, $\mu : \tilde{P} \to \{1, 2, \ldots, S\}$ is defined so that $\mu(\tilde{p}) = q$, where q is such that $\tilde{p} \in P_q$. Thus, CPN is obtained by decentralized approach. Note that a program is developed to implement this algorithm. The results of this program are displayed on the screen. Now, we prove that the above approach guarantees the occurrence of the desired states for OPN.

THEOREM 2 *If CPN is obtained by using the control places of PSNs, $R^{-cp}(\bar{G}) = R_d(G, m_0)$.*

Proof. It is known that $R_k^{-cp}(\bar{G}_k) \subset R_{d_k}(G_k, m_{k_0})$ for k^{th} PSN. $R_{d_k}(G_k, m_{k_0})$ is obtained from the $R_d(G, m_0)$ under the assumption of $m_{k_0} \in R_{k_d}(G_k, m_{k_0})$, for all $k \in \{1, 2, ...S\}$. Each element of $R_k^{-cp}(\bar{G}_k)$ corresponds to the part of the desired marking vector. Therefore, the set $R^{-cp}(\bar{G})$, which is obtained by using decentralized approach, is equal to the set $R_d(G, m_0)$. ∎

While the complexity of the presented decentralized approach is obtained as the sum of the complexity of *Algorithm I* of each PSN, *Algorithm II*, and the overlapping decompositions, the complexity of the centralized approach is defined as the complexity of *Algorithm I* for the OPN. Here, the complexity of *Algorithm I* depends on the cardinality of the set of desired markings, the number of the determined control places, the number of transitions and the number of places, and the complexity of *Algorithm II* depends on the number of the determined control places and the number of transitions (Cormen et al., 1990). Since each PSN is only a part of the given OPN, besides that PSNs have lower dimensional marking vectors, and fewer number of transitions and places (note that, since the decomposition may, in most cases, be made by an eye inspection, the decomposition effort may be neglected). Therefore, the complexity of the decentralized approach is lower than the complexity of the centralized approach.

5. Example

Consider the Petri net, which was presented in Aybar and İftar (2003a), shown in Fig. 3. In this net, $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$, $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_7, t_8, t_9, t_{10}\}$, and $m_0 = [0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T$. The analysis of this net leads to the conclusion that the example net is neither live nor reversible.

Let us guarantee to lead the example net to the elements of the set

$$
\begin{aligned}
R_d(G, m_0) = \{&[0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0]^T, [1\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0]^T, [0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0]^T,\\
&[0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0]^T, [0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0]^T, [0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0]^T,\\
&[0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1]^T, [0\,1\,0\,0\,0\,0\,0\,0\,1\,1\,0]^T, [0\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0]^T\}\,.
\end{aligned}
$$

We use the decentralized approach, which is given in Section 4, for this example. The overlappingly decomposed OPN is shown in Fig. 4. Its expansion, the EPN, is shown in Fig. 5. The set of places of the EPN is $\tilde{P} = P_1 \bigcup P_2$, where $P_1 = \{p_1, p_2, p_3, p_4, p_5, p_{6a}\}$ and $P_2 = \{p_{6b}, p_7, p_8, p_9, p_{10}, p_{11}\}$. The set of transitions of the EPN is $\tilde{T} = T_1 \bigcup T_2 \bigcup \mathcal{T}$, where $T_1 = \{t_1, t_2, t_3, t_4, t_5, \}$,

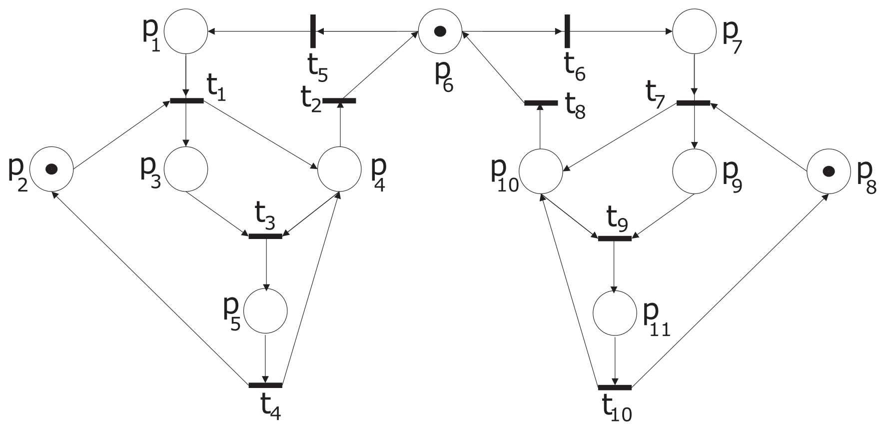

Figure 3. Orginal Petri net (Aybar and İftar, 2003a)

$T_2 = \{t_6, t_7, t_8, t_9, t_{10}\}$, and $\mathcal{T} = \{t_x, t_y\}$. The initial marking is chosen as $\tilde{m}_0 = [0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0]^T$, i.e., it is assumed that all the tokens in the overlapping places in the initial marking, are assigned to the corresponding places of the first PSN (as shown in Fig. 5). The inital markings for the first PSN and the second PSN are, respectively, $m_{1_0} = [0\ 1\ 0\ 0\ 0\ 1]^T$ and $m_{2_0} = [1\ 0\ 1\ 0\ 0\ 0]^T$ (for detailed information about the inital markings of all PSNs, see Aybar and İftar, 2002).

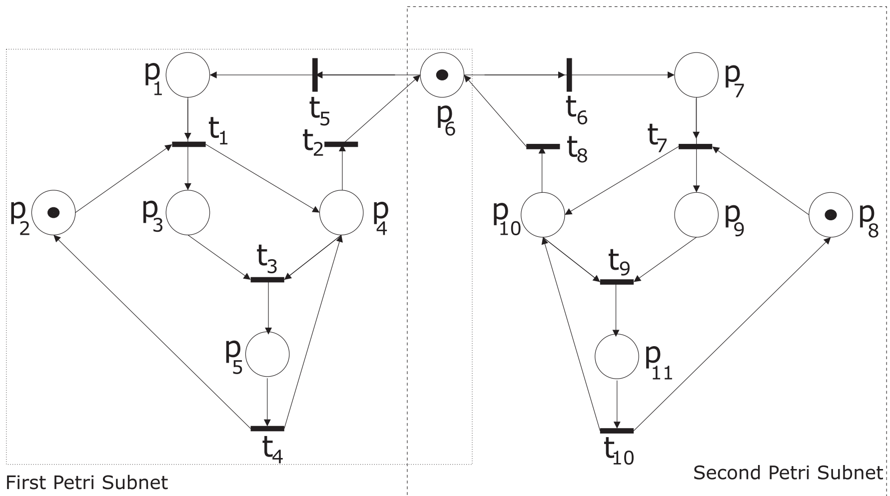

Figure 4. Overlappingly decomposed Petri net

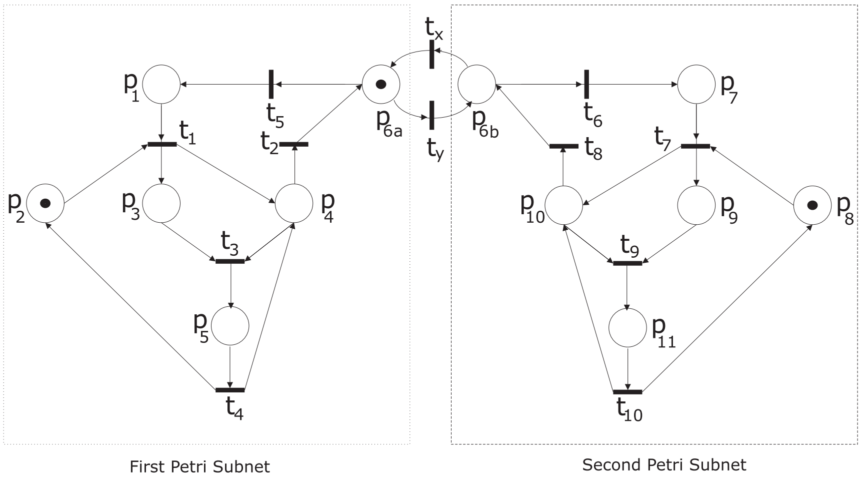

Figure 5. Expanded Petri net

After all PSNs are determined, the sets of the desired marking vectors for two PSNs are obtained as

$R_{d_1}(G, m_0) = \{[0\ 1\ 0\ 0\ 0\ 1]^T, [1\ 1\ 0\ 0\ 0\ 0]^T, [0\ 0\ 1\ 1\ 0\ 0]^T, [0\ 0\ 0\ 0\ 1\ 0]^T,$
$[0\ 1\ 0\ 1\ 0\ 0]^T, [0\ 1\ 0\ 0\ 0\ 0]^T\}$
and
$R_{d_2}(G, m_0) = \{[1\ 0\ 1\ 0\ 0\ 0]^T, [0\ 0\ 1\ 0\ 0\ 0]^T, [0\ 0\ 1\ 0\ 1\ 0]^T, [0\ 0\ 0\ 0\ 0\ 1]^T,$
$[0\ 0\ 0\ 1\ 1\ 0]^T, [0\ 1\ 1\ 0\ 0\ 0]^T\}.$

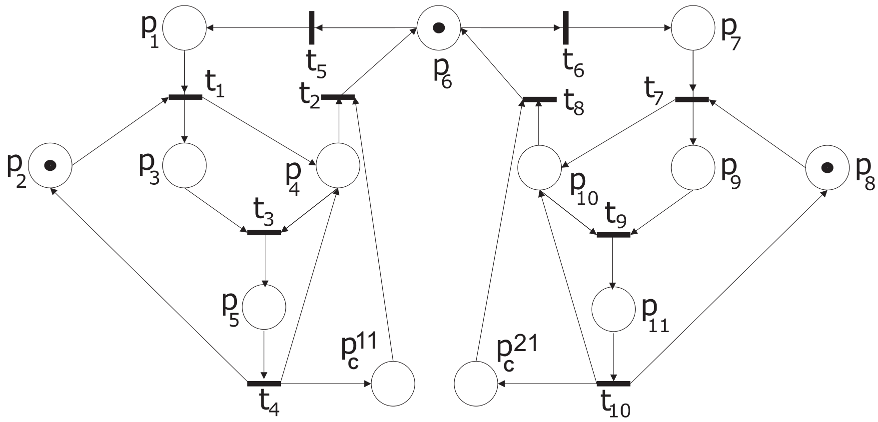

Figure 6. The controlled Petri net (Fig. 3)

Now, we determine the control places for each PSN so that $N_{c_1}(p_c^{1\,1}, t_2) = 1$, $N_{c_1}(p_c^{1\,1}, t) = 0$, $\forall t \in T_1 \setminus \{t_2\}$, $O_{c_1}(p_c^{1\,1}, t_4) = 1$, $O_{c_1}(p_c^{1\,1}, t) = 0$,, $\forall t \in T_1 \setminus \{t_4\}$, $M_{c_{10}} = [0]$ for the first PNS and $N_{c_2}(p_c^{2\,1}, t_8) = 1$, $N_{c_2}(p_c^{2\,1}, t) = 0$, $\forall t \in T_2 \setminus \{t_8\}$, $O_{c_2}(p_c^{2\,1}, t_{10}) = 1$, $O_{c_2}(p_c^{2\,1}, t) = 0$,, $\forall t \in T_2 \setminus \{t_{10}\}$, $M_{c_{20}} = [0]$. Then, by using *Algorithm II*, $N_c(p_c^{1\,1}, t_2) = 1$, $N_c(p_c^{2\,1}, t_8) = 1$, $N_c(p_c, t) = 0$, $\forall p_c \in P_c$, $\forall t \in T \setminus \{t_2, t_8\}$, $O_c(p_c^{1\,1}, t_4) = 1$, $O_c(p_c^{2\,1}, t_{10}) = 1$, $O_c(p_c, t) = 0$, $\forall p_c \in P_c$, $\forall t \in T \setminus \{t_4, t_{10}\}$, and $M_{c_0} = [0\ 0]^T$ are obtained ($P_c = \{p_c^{1\,1}, p_c^{2\,1}\}$). Thus, CPN, shown in Fig. 6, is constructed by using the approach given in Section 4. In CPN, the set of places is $\bar{P} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_c^{1\,1}, p_c^{2\,1}\}$, the set of transition is $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$, the initial marking is $\bar{m}_0 = [0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$, and the input and output matrices are given as

$$\bar{N} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \bar{O} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Using the decentralized controller approach given above, we obtain CPN, the reachability set, which contains the set of the desired marking vectors ($R^{-cp}(\bar{G}) = R_d(G, m_0)$). Note that CPN features both reversibility and liveness.

An example system which was given by Holloway and Krogh (1990) is considered to show the advantage of the decentralized approach. This system consists of five automated guided vehicles, three workstations, two part-receiving stations and one completed parts station. The respective Petri net model (Holloway and Krogh, 1990) has 64 places and 53 transitions. After overlapping decompositions and expansion are used for this net, two disjoint Petri subnets are obtained such that the first PSN has 40 places and 31 transitions, and the second PSN has 26 places and 22 transitions (since two places are in the overlapping part, these places are repeated for two subnets).

The sets of the desired markings, which have 100, 250, and 350 markings are chosen to analyse the presented approaches. The programs for *Algorithm I* and *Algorithm II* are run (on a PC with Pentium-4 microprocessor running at 2.66 GHz with 1 GB RAM) for the centralized and decentralized approaches. In the centralized approach, the program for *Algorithm I* is only used for the OPN. In the decentralized approach, the program for *Algorithm I* is used for the first and second PSNs, and another program for *Algorithm II* is run. The structural controllers, which are designed by using both the centralized approach and decentralized approach guarantee the occurrence of the desired markings of the considered Petri net. The results for the sets of the desired markings are given in Table 1. The ratio of the computational time is very small for three sets, showing that it is about 16 times faster to design the decentralized approach for the considered Petri net (Holloway and Krogh, 1990).

Table 1. Computational time

$\|R_d\|$	*Algorithm I* for the OPN	*Algorithm I* for the first PSN	*Algorithm I* for the second PSN	*Algorithm II*	Ratio
100	1649 sec.	62 sec.	47 sec.	1 sec.	$\frac{(62+47+1)}{1649} = \mathbf{0.06}$
250	19653 sec.	752 sec.	473 sec.	8 sec.	$\frac{(752+473+8)}{19653} = \mathbf{0.06}$
350	45846 sec.	2108 sec.	1082 sec.	16 sec.	$\frac{(2108+1082+16)}{45846} = \mathbf{0.06}$

6. Conclusion

The problem, which is defined as the occurrence of the desired marking vectors from the initial state in Petri nets, is considered in this work. For this problem, the structural solution, which is explained by adding of the control places to the Petri net, is presented in this work.

Algorithm I is developed to determine the control places for the entire Petri nets. The centralized approach is designed by using this algorithm. Furthermore, a decentralized approach, which is based on overlapping decompositions, is introduced. After the control places of each PSN are determined by using *Algorithm I*, the control places for OPN are obtained by using *Algorithm II*. The obtained control places are easily added to the OPN and then CPN is obtained. Therefore, the elements of reachability set of CPN contain the desired markings for OPN. Moreover, the programs, which are written in Microsoft Visual Basic programming language, are organized to implement the presented algorithms. The results of each program are displayed on the screen.

Some effort is needed for the determination of overlapping decomposition of the given Petri net in the decentralized approach. This effort is neglected, though, because the decomposition may be made by an eye inspection. Moreover, the complexity of *Algorithm II* is lower than the complexity of *Algorithm I*. Therefore, the complexity of the decentralized approach mainly depends on the sum of complexities of *Algorithm I* of each PSN. Furthermore, the computation time in the program for *Algorithm I* is related to both the size of the set of desired markings and the sets of transitions and places of the given Petri net (when the definition of the OPN is used for the centralized approach, the definition of each PSN is used for the decentralized approach). Therefore, the complexity of the decentralized approach is lower than the complexity of the centralized approach. For example, the decentralized approach is faster than the centralized approach for the Petri net, which was given by Holloway and Krogh (1990).

The overlapping approach can be used in the structural theory in future studies. It is also possible to design the decentralized controllers which enforce structural deadlock - freeness and liveness - in the following studies.

References

AYBAR, A., İFTAR, A. and APAYDIN-ÖZKAN, H. (2005) Centralized and decentralized supervisory controller design to enforce boundedness, liveness, and reversibility in Petri nets. *International Journal of Control* **78**, 537–553.

AYBAR, A. and İFTAR, A. (2001) Decentralized control design for interconnected discrete–event systems. *Preprints of the 9th IFAC Symposium on Large Scale Systems*. Bucharest, Romania, 415–418.

AYBAR, A. and İFTAR, A. (2002) Overlapping decompositions and expansions of Petri nets. *IEEE Transactions on Automatic Control* **47**, 511–515.

AYBAR, A. and İFTAR, A. (2003a) Decentralized controller design to enforce boundedness, liveness and reversibility in Petri nets. *Proceeding CD-ROM of the European Control Conference*. Cambridge, UK.

AYBAR, A. and İFTAR, A. (2003b) Decentralized supervisory controller design to avoid deadlock in Petri nets. *International Journal of Control* **76**, 1285–1295.

BARKAOUI, K. and ABDALLAH, I. B. (1995) Deadlock avoidance in FMS based on structural theory of Petri nets. *Proceedings of The 1995 IEEE Symposium on The Emerging Technology and Factory Automation*. Paris, France, 499–510.

CHO, K. H. and LIM, J. T. (1999) Mixed centralized/decentralized supervisory control of discrete-event dynamic systems. *Automatica* **35**, 121–128.

CORMEN, T. H., LEISERSON, C. E. and RIVEST, R. L. (1990) *Introduction to Algorithms*. The MIT Press.

ESPARZA, J. and NIELSEN, M. (1994) Decidability issues in Petri nets. BRICS Report Series, RS-94-8, Department of Computer Science, University of Aarhus, Aarhus, Denmark.

EZPELETA, J., COLOM, J. M. and MARTINEZ, J. (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation* **11**, 173–184.

GIUA, A. and DICESARE, F. (1994) Blocking and controllability of Petri nets in supervisory control. *IEEE Transactions on Automatic Control* **39**, 818–823.

HAOXUN, C. (1998) Net structure and control logic synthesis of controlled Petri nets. *IEEE Transactions on Automatic Control* **43**, 1446–1450.

HOLLOWAY, L. E. and KROGH, B. H. (1990) Synthesis of feedback control logic for a class of controlled Petri nets. *IEEE Transactions on Automatic Control* **35**, 514–523.

İFTAR, A. and ÖZGÜNER, Ü. (1998) Overlapping decompositions, expansions, contractions, and stability of hybrid systems. *IEEE Transactions on Automatic Control* **43**, 1040–1055.

IKEDA, M. and ŠILJAK, D. D. (1980) Overlapping decompositions, expansions, and contractions of dynamic systems. *Large Scale Systems* **1**, 29–38.

IORDACHE, M. V. and ANTSAKLIS, P. J. (2001) Generalized conditions for liveness enforcement and deadlock prevention in Petri nets. *Proceedings of 22 International Conference on Application and Theory of Petri Nets (ICATPN2001).* Newcastle upon Tyne, UK., 184–203.

IORDACHE, M. V., MOODY, J. O. and ANTSAKLIS, P. J. (2001) A method for the synthesis of liveness enforcing supervisors in Petri nets. *Proceedings of the American Control Conference.* Arligton, VA., 4943–4948.

LIN, F. and WONHAM, W. M. (1990) Decentralized control and coordination of discrete-event systems with partial observation. *IEEE Transactions on Automatic Control* **35**, 1330–1337.

RAMADGE, P. J. G. and WONHAM, W. M. (1989) The control of discrete event systems. *Proceedings of the IEEE* **77**, 81–98.

RUDIE, K. and WONHAM, W. M. (1992) Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control* **37**, 1692–1708.

SREENIVAS, R. S. (1997) On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modelled by controlled Petri nets. *IEEE Transactions on Automatic Control* **42**, 928–945.

SREENIVAS, R. S. and KROGH, B. H. (1992) On Petri net models of infinite state supervisors. *IEEE Transactions on Automatic Control* **37**, 274–277.

UZAM, M. and WONHAM, W. M. (2006) A hybrid approach to supervisory control of discrete event systems coupling RW supervisors to Petri nets. *International Journal of Advanced Manufacturing Technology* **28**, 747–760.

ZHOU, M. and DICESARE, F. (1993) *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems.* Kluwer Academic, Norwell, MA.