Supervisory Controller Design to Enforce Reversibility and Liveness in Colored Petri Nets

Aydın Aybar and Mustafa Çimen

Abstract: Colored Petri net model which is a model of discrete event systems is considered in this work. A supervisory controller which enforces reversibility and liveness simultaneously is presented. Furthermore, the algorithms, written by pseudo-code, are presented for the supervisory controller design. A program is developed to implement these algorithms.

Keywords: Colored Petri nets, discrete event systems, liveness, reversibility, supervisory controller.

1. INTRODUCTION

Petri net model which was introduced by C. A. Petri [1], has been a popular model for discrete event systems [2]. Many types of this modelling method (such as fuzzy, timed, colored, etc.) have been developed and used for various aims (for example, [3-8]).

The colored Petri net model, which is a type of Petri nets, is considered in this work. In this model, any place can have one or more different colored tokens and any transition can have different colors. While a place with a group of tokens represents the information about a process, the same place with another group of tokens represents the information about another process. When a transition with a color represents either start or completion of an event, the same transition with another color represents either start or completion of another event. Thus, for a discrete event system, the number of places and transitions in the colored Petri net model is less than the number of places and transitions in the generalized Petri net model.

Manufacturing systems can easily be modelled to analyse some properties by using the colored Petri net model. Two of the most important properties are the cyclic behavior and the occurence of all events. These properties, which are explained such that a system can reach to the initial status and make all desired process, are respectively corresponding to reversibility and liveness in Petri net model. Therefore, reversibility and liveness of colored Petri nets are considered in this work.

Many works have been considered to analyse and develop the supervisory controller for various properties of Petri nets (e.g., [3,4,7,9-12]). In [12], the major properties such as boundedness, reversibility and liveness were considered at the same time and a supervisory controller design approach was introduced to guarantee all of these properties simultaneously. This approach was extended to decentralized controller design for generalized Petri nets in [7]. In addition, the structural controller design approaches for some properties of Petri nets were presented by various researches (for example, [11,13]).

In this work, a supervisory controller design approach which guarantees reversibility and liveness in colored Petri nets is introduced. This controller approach is based on [12]. After the definitions of these properties are given for colored Petri nets, the algorithms, which construct the reversible set, check the consistency, and design the controller, are developed. Furthermore, a software which is developed to implement for these algorithms is presented in this work.

2. COLORED PETRI NETS

Colored Petri net is denoted by a tuple $G(P,T,N,O,C,\Omega,m_0)$. Here, P is the set of places, T is the set of transitions, Ω is the set of color, $C: T\cup P \to \Omega$ is the color function, $C(P)$ is the set of color for all places, $C(T)$ is the set of color for all transitions, $N(p,t): C(T) \to \mathbb{Z}^{1\times|C(P)|}$, is the matrix that specifies the weights of arcs directed from place p to transition t, where $\mathbb{Z}$ is the set of non-negative integer numbers, $O(t,p): C(T) \to \mathbb{Z}^{1\times|C(P)|}$, is the matrix that specifies the weights of arcs directed

Manuscript received July 25, 2005; revised April 11, 2007; accepted May 28, 2007. Recommended by Editorial Board member Won-jong Kim under the direction of Editor Tae-Woong Yoon.

Aydın Aybar and Mustafa Çimen are with the Department of Electrical and Electronics Engineering, Anadolu University, 26555, Eskişehir, Turkey (e-mails: {aaybar, mustafac}@anadolu.edu.tr).

from transition t to place p, and m_0 is the initial marking.

$M(p,\rho)$ indicates the number of $\rho \in C(P)$ color assigned by marking M to place p. Note that, if the place p^+ has not ρ^*, then $M(p^+,\rho^*)=0$. $M(p)$ is the vector which denotes the number of the place color and $M: P \rightarrow \mathbb{Z}^{1\times|C(P)|}$, is the marking matrix,

$$M=[M(p_1)^T\ M(p_2)^T\ \ldots M(p_n)^T]^T.$$

Here, $P=\{p_1,p_2,\ldots,p_n\}$, $[.]^T$ denotes the transpose of $[.]$.

A transition $t\in T$ for color $\tau \in C(t)$ is enabled if and only if $M(p,\rho)\geq N(p,t)(\tau,\rho)$, $\forall p\in P$, $\forall \rho \in C(p)$, $t\in T$ and $\tau\in C(t)$ when $N(p,t)(\tau,\rho)(p,\rho)>0$. Here, $N(p,t)(\tau)$ is the vector which indicates the number of all elements of $C(P)$ in place p for $\tau\in C(t)$, and $N(p,t)(\tau,\rho)$ denotes the number of color ρ at $N(p,t)(\tau)$.

An enabled transition t for color τ can fire at M, yielding the new marking:

$$M'(p,\rho)=M(p,\rho)-N(p,t)(\tau,\rho)+O(t,p)(\tau,\rho),$$

$\forall p\in P, \forall \rho \in C(p)$. Note that, since t for τ can fire only when $M(p,\rho)-N(p,t)(\tau,\rho)>0$, above equation yields $M'(p,\rho)\in Z$, $\rho\in C(p)$. A firing sequence g consists of pairs of the enabled transitions and their colors $(t_i,\tau_j)(t_n,\tau_h)$, where $t_i,t_n\in T$ and $\tau_j\in C(t_i)$, $\tau_n \in C(t_n)$. A marking M' is said to be reachable from M if there exist a sequence starting from M and yielding M'. The set denoted by $R(G,m_0)$ is the set of all markings reachable from the initial marking. We let $E(G,M)$ to denote the set of pairs of transition and its color which are enabled at M in Petri net G, and also $\gamma(M,g)$ to denote the transition function, which gives the yielded marking when the sequence g fires starting from M.

The considered properties of colored Petri nets are given as follow:

Definition 1: A colored Petri net is said to be reversible if $m_0\in R(G,M)$, $\forall M\in R(G,m_0)$.

Definition 2: A colored Petri net is said to be live if, for every $M\in R(G,m_0)$ and for every (t,τ), $\forall \tau\in C(t)$, there exists a sequence g such that t for τ can fire at $\gamma(M,g)$.

The example colored Petri nets is considered to explain the notation in this work. This example net is shown in Fig. 1.

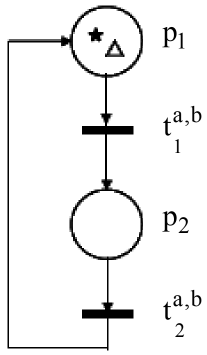

Fig. 1. Example colored Petri net.

This net is described as $P=\{p_1,p_2\}$, $T=\{t_1,t_2\}$, $\Omega=\{\star,\vartriangle,a,b\}$, $C(P)=\{\star,\vartriangle\}$, $C(T)=\{a,b\}$, and $m_0=\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}$. In addition, the input and output matrices are constructed as follow:

$$\begin{aligned}
&N(p_1,t_1)(a)=[1\ 0], && O(t_1,p_1)(a)=[0\ 0],\\
&N(p_1,t_1)(b)=[0\ 1], && O(t_1,p_1)(b)=[0\ 0],\\
&N(p_1,t_2)(a)=[0\ 0], && O(t_2,p_1)(a)=[1\ 0],\\
&N(p_1,t_2)(b)=[0\ 0], && O(t_2,p_1)(b)=[0\ 1],\\
&N(p_2,t_1)(a)=[0\ 0], && O(t_1,p_2)(a)=[1\ 0],\\
&N(p_2,t_1)(b)=[0\ 0], && O(t_1,p_2)(b)=[0\ 1],\\
&N(p_2,t_2)(a)=[1\ 0], && O(t_2,p_2)(a)=[0\ 0],\\
&N(p_2,t_2)(b)=[0\ 1], && O(t_2,p_2)(b)=[0\ 0].
\end{aligned}$$

The new marking, M, which is obtained by firing transition t_1 for color a at the initial marking m_0, is determined as

$$\begin{aligned}
M(p_1,\star)&=m_0(p_1,\star)-N(p_1,t_1)(a,\star)+O(t_1,p_1)(a,\star)\\
&=1-1+0=0,\\
M(p_1,\vartriangle)&=m_0(p_1,\vartriangle)-N(p_1,t_1)(a,\vartriangle)+O(t_1,p_1)(a,\vartriangle)\\
&=1-0+0=1,\\
M(p_2,\star)&=m_0(p_2,\star)-N(p_2,t_1)(a,\star)+O(t_1,p_2)(a,\star)\\
&=0-0+1=1,\\
M(p_2,\vartriangle)&=m_0(p_2,\vartriangle)-N(p_2,t_1)(a,\vartriangle)+O(t_1,p_2)(a,\vartriangle)\\
&=0-0+0=0,\\
M&=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}.
\end{aligned}$$

3. REVERSIBILITY AND LIVENESS

The algorithms for reversibility and liveness are given in this section.

3.1. Reversibility in Color Petri Nets

The construction of the reversible set is presented in sub-section. This set is defined as

$$R_r := \{M \in R(G, m_0) \mid \exists\, g \text{ such that } \gamma(M, g) = m_0\}.$$

The algorithm, *Reversible-Set*, which was first introduced for generalized Petri nets by [12], is extented to the colored Petri nets in this work. This algorithm, shown in Appendix A, is developed to construct the set R_r. *Reversible-Set* requests the description of the given colored Petri net, by G (the set of places and transitions, the input and output matrices, the color function, the set of color, and initial marking) and the reachability set. The reachability set is constructed by the algorithm *R-Set*, given Appendix B (see for detail [14]). The algorithm *R-Set* constructs the all markings which are obtained from the initial marking and returns the reachability set, denoted by R. Moreover, $E(G,M)$, which obtains the set of the pairs of enabled transition and its color at $M \in R(G, m_0)$, is used by above algorithms (see Appendix C).

Reversible-Set finds the markings which are reached to the initial marking by firing any transition with color. These markings are added to the set R_r. If m_0 is only one element of R_r, then it is not possible to find the controller which enforce reversibility in the given colored Petri net. It is known that if any marking $M \in R_r$, then there always exists a sequence g such that $\gamma(M, g) = m_0$.

3.2. Liveness in Color Petri Nets

Liveness is analysed in the reversible colored Petri nets such that the consistency is checked in the desired nets. Consistency, which is defined for generalized Petri nets [2], is presented to describe liveness for colored Petri nets.

Definition 3: A colored Petri net is said to be consistent if there exists a sequence, g_0, starting from the initial marking such that g_0 contains each transition $t \in T$ and all elements of $C(t)$ at least once and, when fires, leads back to m_0.

Lemma 1: If and only if the given colored Petri net is consistent, then a controller, which enforces both reversibility and liveness, exists for a given colored Petri net.

Proof: This lemma was developed and proved for generalized Petri net in [12]. The proof in [12] is extended to the colored Petri nets. If the given colored Petri net is not consistent, then either some $t \in T$ for any element of $C(t)$ can not fire or the initial state can not be reached once some $t \in T$ for any element of $C(t)$ fires. Therefore, this net is neither reversible nor live. Furthermore, a controller can not force the colored Petri net since there does not exists a sequence which contains each transition $t \in T$ and each color of $C(t)$ at least once. □

The consistency algorithm, Appendix D, which was presented for the generalized Petri nets by [12], is extented to the colored Petri nets in this work. This algorithm determines whether or not the colored Petri net is consistent. If a sequence, which contains each transition $t \in T$ for each element of $C(t)$, is searched in the reversibile set in this algorithm, then it is possible to enforce reversibility and liveness. Otherwise, it is not possible (see Lemma 1).

4. CONTROLLER DESIGN

A supervisory controller, which is based on the forbidden markings, is introduced to enforce reversibility and liveness in the colored Petri net. This controller is defined as the control function such that $K : R(G, m_0) \times T \times C(T) \to \{0,1\}$. Here, $K(M, t, \tau) = 0$ denotes that the transition t for the color τ is disabled at the marking M, and $K(M, t, \tau) = 1$ denotes that the transition t for the color τ is enabled at the marking M (note that, the pair (t, τ) is denoted by Θ in this work).

The controller is described as

$$K(M, \Theta) := \begin{cases} 1, & \text{if } \gamma(M, \Theta) \in R_D \\ 0, & \text{otherwise,} \end{cases}$$

$\forall M \in R(G, m_0)$ and $\forall\, \Theta \in E(G, M)$. Here, $R_D \subset R$ (G, m_0) denotes the set of the desired markings. In this work, the set R_r is taken as R_D. If the given colored Petri net is consistent, then the reversibility and liveness is enforced for the given colored Petri net. Note that, if there exists the reversible set for the given colored Petri net which is not consistent, then a controller function, which only enforces reversibility, can be found.

The colored Petri net under the controller K, which is called as the controlled net, is denoted by the eight tuple $G_c(P, T, N, O, C, \Omega, m_0, K)$. In this net, the controller K enables or disables the considered transition. The reachability set of the controlled net is determined as the set R_D. Thus, this controlled net is reversible and live.

An algorithm for the controller function is given in Appendix E. This algorithm requires the current marking, the transition and its color, and the set of the desired markings.

The structural controller, described by adding the control places to the Petri nets, was introduced to guarantee the occurence of the desired marking vectors [15] can be considered for colored Petri nets. The algorithms in [15] can be easily extented to colored Petri nets. It is possible that the desired markings could occur in the colored Petri net with the control places.

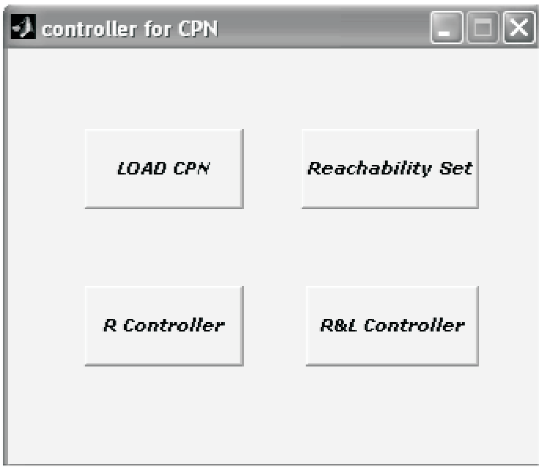

Fig. 2. The interface for the developed program.

5. PROGRAM

In this section, the program is introduced for the presented algorithms. These algorithms are implemented by using Matlab such that each algorithm is developed as a matlab function.

The interface, shown in Fig. 2, contains four buttons which are denoted by the matlab functions of the presented algorithms. The "Load CPN" button must be initially used. After this button is clicked, the name of the input file which is constructed by the user must be entered. This file contains the definition of the considered colored Petri net (for example, the file format is given Appendix H).

By clicking on the "Reachability Set" button, the matlab function which implements the algorithm shown in Appendix B is run to determine the reachability set $R(G,m_0)$. Each element of the set $R(G,m_0)$ also appears on the matlab display. The "R Controller" button can be used to guarantee only reversibility. By clicking on this button, the algoritms, shown in Appendix G, is executed. This button constructs the reversible set and the controller for the reversibility enforcement. All of them appears on the screen. If two basic properties, reversibility and liveness, are enforced simultaneously, then the "R & L Controller" button can be used to enforce reversibility and liveness. When this button is clicked, the consistency is firstly checked by using the algorithm in Appendix D, and then the controller which enforces reversibility and liveness simultaneously is obtained. If the considered colored Petri net is consistent, then the results of the controller function appear on the screen such that only disabled the transition and its color are displayed.

6. EXAMPLE

We consider the manufacturing system which was

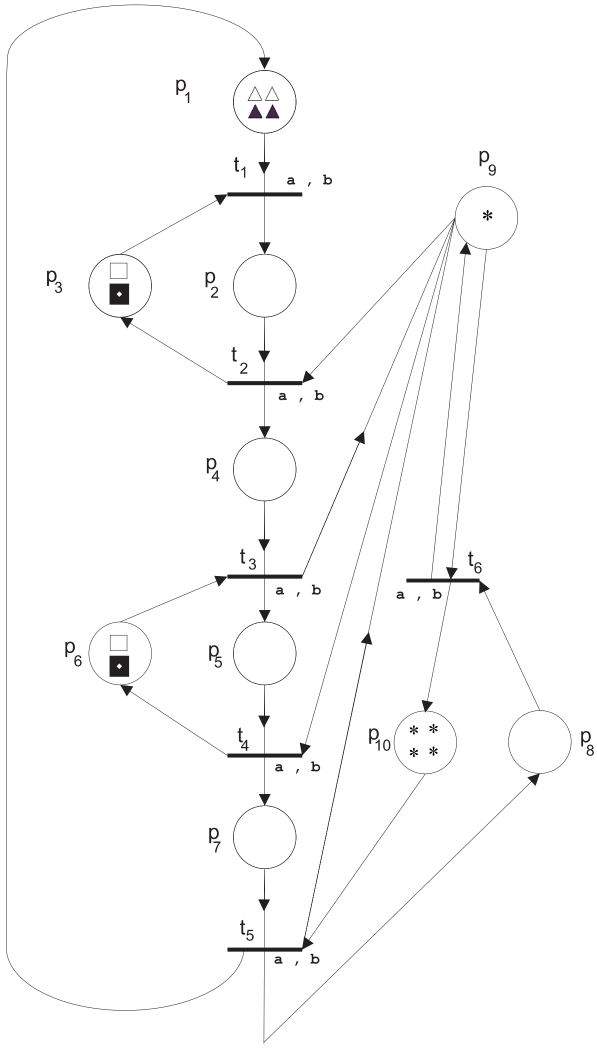

Fig. 3. Colored Petri net model of a manufacturing system.

given by [1]. This system consists of two non-identical assembly lines, each of which contains one robot, and two additional robots, both of which can serve to one of the two lines at a time. The colored Petri net model $G(P,T,N,O,C,\Omega,m_0)$ of this manufacturing system is shown in Fig. 3, where all the arcs have unity weights and the colored tokens and transitions are defined.

In this net, the set of places is $P=\{p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8,p_9,p_{10}\}$, the set of transition is $T=\{t_1,t_2,t_3,t_4,t_5,t_6\}$, the set of color is $\Omega=\{\vartriangle,\blacktriangle,\square,\blacksquare,\circ,\bullet,\star,a,b\}$, the set of colors for places is $C(P)=\{\vartriangle,\blacktriangle,\square,\blacksquare,\circ,\bullet,\star\}$, and the set of colors for transitions $C(T)=\{a,b\}$. All definition of this net is given Appendix H as the input file for matlab.

Let us analyse this colored Petri net: since deadlock occurs using the developed algorithm in this net (see [14]), the example net is neither live nor reversible. The program, which is developed by using the presented algorithms, is used to enforce reversibility and liveness.

All markings and the disabled transitions for reversibility and liveness enforcement are obtained by using the developed program. The reachability set is constructed to use for the remainder algorithms. It is found that R has 280 elements. When the "R & L Controller" button can be actived, the reversible set, $R_r \subset R$ is constructed (R_r has 224 elements), the consistency tests for this net (this net is consistent), and then the controller which enforces reversibility and liveness in colored Petri nets is obtained. Hence, in this example, the controller forbits about 8% of the original markings to achieve reversibility and liveness. Some example results of the controller functions are listed as follow:

- the fourth transition for the first color is disabled at the marking M_c^1 (since all sets have ordered format, this message shows that t_4 for color a is disabled, i.e., $K(M_c^1, t_4, a) = 0$).
- the fourth transition for the second color is disabled at the marking M_c^1.
- the second transition for the first color is disabled at the marking M_c^2.
- the second transition for the second color is disabled at the marking M_c^2.

These above messages are corresponding to the results of the controller function (for example, $K(M_c^1, t_4, a) = 0$). Consequently, the designed controller enforces reversibility and liveness in the example net (note that, $\hat{R} = \{M_c^1, M_c^2, ..., M_c^{40}\}$ is obtained by this program).

$$M_c^1 = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, M_c^2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}.$$

7. CONCLUSION

A supervisory controller design approach is considered for reversibility and liveness in colored Petri nets. The controller design, based on [12], guarantees reversibility and liveness for the given colored Petri nets. The proposed approach always finds a controller whenever it exists. Furthermore, the controller obtained is the least restrictive controller among all controllers which enforce desired properties.

The algorithms are presented to design this controller for these properties simultaneously. These algorithms also determine the reversible set and check the property consistency for liveness. A program is developed to implement these algorithms. It finds the reachability and reversible sets and designs the controllers.

The structural controller design approach which was defined by [15] can be easily extented to colored Petri nets. The new colored Petri net is constructed by adding the control places to the original colored Petri net. Thus, it is possible that the new colored Petri net is obtained as reversible and live.

Further research is underway to use the presented approach in a decentralized framework. Another direction for further research may be developing approaches based on timed and / or colored Petri nets model (for example, routing control of the computer networks and the popular problems such as chess game).

APPENDICES

Appendix A: Algorithm to construct the reversible set.

Reversible - Set $[G,R]$
$R' := R \setminus \{m_0\}$
$R_r := \{m_0\}$
Do Loop Reversible
 $R^* = R'$
 $new = 0$
For $i = 1$ To $|R^*|$
 $\Theta = E(G, [R^*]_i)$
  For $j = 1$ To $|\Theta|$
   $\hat{M} = \gamma([R^*]_i, [\Theta]_j)$
    If $\hat{M} \in R_r$ Then
      $R_r \leftarrow R_r \cup [R^*]_i$
      $R' \leftarrow R' \setminus [R^*]_i$
      new =1
      Go To Jump
    End
  End
Jump : Continue
End
 If new =0 Then
  Exit Loop Reversible
 End
Loop Reversible
Return R_r

Notation used in the presentation of the algorithms: for a set X, $|X|$ denotes the number of the elements of X, and $[X]_i$ denotes the i^{th} element of X ($i = 1, 2, ..., |X|$). 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 $(n+1)^{\text{th}}$ element.

Appendix B: Algorithm to construct the reachability set.

R-Set $[G]$
$R := \{m_0\}$
$R_1 := \{m_0\}$
Do Loop Const
  $R_2 = \varnothing$
   For $i = 1$ to $|R_1|$
    $\bar{M} = [R_1]_i$
    $\Theta = E(G, [R_1]_i)$
     For $j = 1$ to $|\Theta|$
      $\hat{M} = \gamma(\bar{M}, [\Theta]_j)$
      $R_2 \leftarrow R_2 \cup \{\hat{M}\}$
     End
    End
  $R_1 = \varnothing$
  $R_1 = R_2 \setminus R$
   If $R_1 = \varnothing$ Then
     Exit Loop Const
   End
  $R \leftarrow R \cup R_1$
   Loop Const
Return R

Appendix C: Algorithm to find the set of enabled transitions.

$E(G, M)$
 $\hat{T} = \varnothing$
 For $i = 1$ to $|T|$
  For $j = 1$ to $|C(T)|$
   $flag = 0$
   For $k = 1$ to $|P|$
    If $max(N([P]_k, [T]_i)([C(T)]_j)) > 0$ Then
     For $m = 1$ to $|C(P)|$
      If $M([P]_k, [C(P)]_m) >=$
       $N([P]_k, [T]_i)([C(T)]_j, [C(P)]_m)$ Then
        $flag = 1$
      Else
         Go To Jump
      End
     End
   End
   End
   If $flag > 0$ Then
    $\hat{T} \leftarrow \hat{T} \cup \{([T]_i, [C(T)]_j)\}$
   End
   Jump: Continue
  End
End
Return $\hat{T}$

Appendix D: Algorithm to check consistency.

Consistency $[G, R_r]$
rt = "not consistent"
For $i = 1$ To $|T|$
 For $j = 1$ To $|C(T)|$
  If $[C(T)]_j \in C([T]_i)$ Then
    $\Theta \leftarrow \Theta \cup ([T]_i, [C(T)]_j)$
  End
 End
End
For $i = 1$ To $|R_r|$
 $J = E(G, [R_r]_i)$
  For $j = 1$ To $|J|$
   If $\gamma([R_r]_i, [J]_j) \in R_r$ Then
    $\Theta \leftarrow \Theta \cup [J]_j$
     If $\Theta = \varnothing$ Then
      rt = "consistent"
      Go To Jump
     End
   End
  End
End
Jump: Return rt

Appendix E: Algorithm to find the controller function

Cnt $[R_D, M, \Theta]$
If $\gamma(M, \Theta) \in R_D$ Then
  $k = 1$
Else
  $k = 0$
End
Return k

Appendix F: Algorithm to find the reversibility enforcement controller.

R Controller $[G,R]$
R_r = Reversible-Set$[G, R]$
 If $|R_r| = 1$ Then
  "not design a controller"
  Exit algorithm
 Else
 For $i = 1$ To $|R_r|$

$\Theta = E(G,[R_r]_i)$
For $j=1$ To $|\Theta|$
$K([R_r]_i,[\Theta]_j) =$ Cnt$[R_r,[R_r]_i,[\Theta]_j]$
End
End
End

Appendix G: Algorithm to find the reversibility and liveness enforcement controller.

R&L Controller $[G,R]$
R_r = Reversible-Set $[G,R]$
If $|R_r|=1$ Then
"not design a controller"
Exit algorithm
Else
chk = Consistency $[G,R_r]$
If chk = "consistent" Then
For $i=1$ To $|R_r|$
$\Theta = E(G,[R_r]_i)$
For $j=1$ To $|\Theta|$
$K([R_r]_i,[\Theta]_j) =$ Cnt$[R_r,[R_r]_i,[\Theta]_j]$
End
End
Else
"not design a controller"
Exit algorithm
End
End

Appendix H: The input file (format) which is called as "definition.m" for the example colored Petri net (Fig. 3) is given as follow.

```
%i) number of place
cpnp=10;
%ii) number of  colors for places
cpnpc=7;
%iii) number of transition
cpnt=6;
%iv) number of  colors for transitions
cpntc=2;

% initialize: input, output and marking
% matrices
N=zeros(cpnt,cpnpc,cpnp,cpntc);
O=zeros(cpnp,cpnpc,cpnt,cpntc);
M(:,:,1)=zeros(cpnp,cpnpc);

% initial  marking
z=zeros(1,cpnpc);
M(:,:,1)=[2 2 0 0 0 0 0;z;
    0 0 0 0 1 1 0;z;z;
    0 0 0 0 1 1 0;z;z;
    0 0 0 0 0 0 1;
    0 0 0 0 0 0 4];

% input matrix
N(:,:,1,1)=[1 0 0 0 0 0 0;z;z;z;z;z];
N(:,:,1,2)=[0 1 0 0 0 0 0;z;z;z;z;z];
N(:,:,2,1)=[z;0 0 1 0 0 0 0;z;z;z;z];
N(:,:,2,2)=[z;0 0 0 1 0 0 0;z;z;z;z];
N(:,:,3,1)=[0 0 0 0 1 0 0;z;z;z;z;z];
N(:,:,3,2)=[0 0 0 0 0 1 0;z;z;z;z;z];
N(:,:,4,1)=[z;z;0 0 1 0 0 0 0;z;z;z];
N(:,:,4,2)=[z;z;0 0 0 1 0 0 0;z;z;z];
N(:,:,5,1)=[z;z;z;0 0 1 0 0 0 0;z;z];
N(:,:,5,2)=[z;z;z;0 0 0 1 0 0 0;z;z];
N(:,:,6,1)=[z;z;0 0 0 0 1 0 0;z;z;z];
N(:,:,6,2)=[z;z;0 0 0 0 0 1 0;z;z;z];
N(:,:,7,1)=[z;z;z;z;0 0 1 0 0 0 0;z];
N(:,:,7,2)=[z;z;z;z;0 0 0 1 0 0 0;z];
N(:,:,8,1)=[z;z;z;z;z;0 0 0 0 0 0 1];
N(:,:,8,2)=[z;z;z;z;z;z];
N(:,:,9,1)=[z;0 0 0 0 0 0 1;z;
        0 0 0 0 0 0 1;z;0 0 0 0 0 0 1];
N(:,:,9,2)=[z;0 0 0 0 0 0 1;z;
                0 0 0 0 0 0 1;z;z];
N(:,:,10,1)=[z;z;z;z;
                0 0 0 0 0 0 1;z];
N(:,:,10,2)=[z;z;z;z;
                0 0 0 0 0 0 1;z];

% output matrix
O(:,:,1,1)=[z;0 0 1 0 0 0 0;
                z;z;z;z;z;z;z;z];
O(:,:,1,2)=[z;0 0 0 1 0 0 0;
                z;z;z;z;z;z;z;z];
O(:,:,2,1)=[z;z;0 0 0 0 1 0 0;
            0 0 1 0 0 0 0;z;z;z;z;z;z];
O(:,:,2,2)=[z;z;0 0 0 0 0 1 0;
              0    0    0    1    0    0
0;z;z;z;z;z;z];
O(:,:,3,1)=[z;z;z;z;0 0 1 0 0 0 0;
                z;z;z;0 0 0 0 0 0 1;z];
O(:,:,3,2)=[z;z;z;z;0 0 0 1 0 0 0;
                z;z;z;0 0 0 0 0 0 1;z];
O(:,:,4,1)=[z;z;z;z;z;0 0 0 0 1 0 0;
                0 0 1 0 0 0 0;z;z;z];
O(:,:,4,2)=[z;z;z;z;z;0 0 0 0 0 1 0;
                0 0 0 1 0 0 0;z;z;z];
O(:,:,5,1)=[1 0 0 0 0 0 0;
                z;z;z;z;z;z;
        0 0 0 0 0 0 1;0 0 0 0 0 0 1;z];
O(:,:,5,2)=[0 1 0 0 0 0 0;
                z;z;z;z;z;z;
        0 0 0 0 0 0 1;0 0 0 0 0 0 1;z];
O(:,:,6,1)=[z;z;z;z;z;z;z;z;
            0 0 0 0 0 0 1;0 0 0 0 0 0 1];
O(:,:,6,2)=[z;z;z;z;z;z;z;z;z;z];
```

REFERENCES

[1] M. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic, Norwell, MA, 1993.

[2] J. Proth and X. Xie, *Petri Nets: A Tool for Design and Management of Manufacturing*

Systems, John Wiley & Sons, West Sussex, 1996.

[3] R. S. Sreenivas and B. H. Krogh, "On Petri net models of infinite state supervisors," *IEEE Trans. on Automatic Control*, vol. 37, no. 2, pp. 274-277, 1992.

[4] R. S. Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modelled by controlled Petri nets," *IEEE Trans. on Automatic Control*, vol. 42, no. 7, pp. 928-945, 1997.

[5] A. Aybar and A. İftar, "Decentralized supervisory controller design to avoid deadlock in Petri nets," *International Journal of Control*, vol. 76, no. 13, pp. 1285-1295, 2003.

[6] M. P. Fandi, A. Guia, and C. Seatzu, "A deadlock prevention method for railway networks using monitors for colored Petri nets," *Proc. of IEEE Int. Conf. Systems, Man & Cybernetics*, Washington, D.C., USA, vol. 2, pp. 1866-1873, October 2003.

[7] A. Aybar, A. İftar, and H. Apaydin-Özkan, "Centralized and decentralized supervisory controller design to enforce boundedness, liveness, and reversibility in Petri nets," *International Journal of Control*, vol. 78, no. 8, pp. 537-553, 2005.

[8] A. Aybar and A. İftar, "Supervisory controller design for timed Petri nets," *Proc. of IEEE International Conference on System of Systems Engineering*, Los Angeles, CA, U.S.A., pp. 59-64, April 2006.

[9] A. Giua and F. DiCesare, "Blocking and controllability of Petri nets in supervisory control," *IEEE Trans. on Automatic Control*, vol. 39, no. 4, pp. 818-823, 1994.

[10] C. Haoxun, "Net structure and control logic synthesis of controlled Petri nets," *IEEE Trans. on Automatic Control*, vol. 43, no. 10, pp. 1446-1450, 1998.

[11] M. V. Iordache and P. J. Antsaklis, "Generalized conditions for liveness enforcement and deadlock prevention in Petri nets," *Proc. of the 22nd International Conference on Application and Theory of Petri Nets*, Newcastle upon Tyne, UK, pp. 184-203, June 2001.

[12] A. Aybar and A. İftar, "Controller design to enforce boundedness, liveness, and reversibility in Petri nets," *Preprints of the 7th IFAC Workshop on Intelligent Manufacturing Systems*, Budapest, Hungary, pp. 199-204, April 2003.

[13] M. V. Iordache and P. J. Antsaklis, "Design of *T*-liveness enforcing supervisors in Petri nets," *IEEE Trans. on Automatic Control*, vol. 48, no. 11, pp. 1962-1974, 2003.

[14] M. Çimen and A. Aybar, "Deadlock avoidance controller design for colored Petri nets," *Proc. of the National Automatic Control Conference* (in Turkish), Istanbul, Turkey, pp. 187-192, June 2005.

[15] A. Aybar, "Decentralized structural control approach for Petri nets," *Control and Cybernetics*, vol. 36, no. 1, pp. 1-17, 2007.

Aydın Aybar received the Ph.D. degree in Electrical and Electronics Engineering from Anadolu University, Turkey, in 2001. He is currently an Associate Professor at Anadolu University. His current research interests include discrete event dynamic systems, Petri nets, large scale systems, decentralized controller design.

Mustafa Çimen received the M.S. degree in Electrical and Electronics Engineering from Anadolu University, Turkey in 2005. His research interests include Petri nets.