Petri Net Control Toolbox

Marco Ambu

May 26, 2001

1 Introduction

The techniques of supervisory control can be used to control Place/Transition Petri nets. Generalized Mutual Exclusion Constraints (GMECs) are a subset of static specifications for Petri Nets . GMECs are useful because each constraint can be set adding a monitor place to the net.

2 Definitions

Given a Place/Transition Petri Net $N = (P, T, Pre, Post)$, where

- $P = \{p_1, ..., p_m\}$ is a set of m places
- $T = \{t_1, ..., t_n\}$ is a set of t transitions
- $Pre : P \times T \rightarrow N$; $Pre(p, t)$is the number of arcs from p to t
- $Post : P \times T \rightarrow N$; $Post(p, t)$ is the number of arcs from t to p

and a marking M, where

- $M : P \rightarrow N$; $M(p)$ is the number of tokens in the place p

a marked Place/Transition Net is a couple $\langle N, M_0 \rangle$:

- N is a Place/Transition net
- M_0 is the initial marking of the net

The reachability set of the net $\langle N, M_0 \rangle$ is $R(N, M_0)$.

A GMEC is a couple $(\overrightarrow{w}, k)$:

- $\overrightarrow{w} \in \mathbf{Z}^m$
- $k \in \mathbf{N}$

that defines a set of legal states $\mathcal{L}(\overrightarrow{w}, k) = \{M \in N^m \mid \overrightarrow{w}^T \cdot M \leq k\}$ and a set of forbidden states $\mathcal{F}(\overrightarrow{w}, k) = \mathbf{N}^m \backslash \mathcal{L}(\overrightarrow{w}, k)$.

Given a net $\langle N, M_0 \rangle$ and a GMEC $(\overrightarrow{w}, k)$, the reachability set of the controlled net is $\mathcal{M}(N, M_0, \overrightarrow{w}, k) = R(N, M_0) \cap \mathcal{L}(\overrightarrow{w}, k)$.

The monitor place s that imposes a GMEC $(\overrightarrow{w}, k)$ for a net $\langle N, M_0 \rangle$ is calculated as

- $C_s = -\overrightarrow{w}^T \cdot C$ is the row of the incidence matrix of s
- $M_0(s) = k - \overrightarrow{w}^T \cdot M_0$ is the initial marking of s

A transition is controllable if it can be disabled for firing, else it is uncontrollable.

The set of transitions T can be subdivided in two disjoint sets

- T_u: set of uncontrollable transitions
- Tc: set of controllable transitions

3 Control by monitor place in presence of uncontrollable transitions

The monitor place that sets a GMEC for a net with only controllable transitions can be calculated if the initial marking of the net belongs to the legal states of the constraint (if $M_0 \in \mathcal{L}(\overrightarrow{w}, k)$).

If the net has one ore more uncontrollable transitions, a GMEC is defined controllable if the monitor place has arcs only to controllable transitions; a GMEC is defined uncontrollable it the monitor place has one or more arcs to an uncontrollable transition.

If a GMEC $(\overrightarrow{w}, k)$ is uncontrollable it is possible to find a set of controllable GMECs $(\overrightarrow{w}_i, k_i)$ that satisfy the given uncontrollable constraint $(\overrightarrow{w}, k)$. The GMECs $(\overrightarrow{w}_i, k_i)$ are more restrictive than $(\overrightarrow{w}, k)$, in fact they reduce the reachability set of the controlled net, $R(N, M_0, \overrightarrow{w}_i, k_i) \subset R(N, M_0, \overrightarrow{w}, k)$.

4 Functionality of the toolbox

The Petri Net Control Toolbox is a set of Matlab functions for dealing with the control of Place/Transition Petri Nets using GMECs and monitor places.

You can verify if a GMEC is controllable for a given net with uncontrollable transitions. If a set of GMEC is given, you can verify which GMEC of the set is not controllable for a given net with uncontrollable transitions.

In presence of uncontrollable GMECs it is possible to find the set of minimally restrictive controllable GMECs that satisfy the uncontrollable constraint.

Given a controllable GMEC or a set of GMECs it is possible to find the monitor places for a given net.

5 Functions

Parameters of the toolbox functions:

- C (m x n): incidence matrix of $\langle N, M_0 \rangle$
- M0 (m x 1): column vector of initial marking of $\langle N, M_0 \rangle$
- u (1 x uc): row vector containing the numbers of the uncontrollable transitions
- w (m x 1): column vector $\vec{w}$ of the GMEC $(\vec{w}, k)$
- k (1 x 1): scalar k of the GMEC $(\vec{w}, k)$
- W (m x h): matrix with each $\vec{w}_i$ of the set of h GMECs (W, K)
- K (h x 1): column vector with each k_i of the set of h GMECs (W, K)

5.1 Checkgmec

Sintax

```
checkgmec(C,M0,w,k,u)
```

Description

Checkgmec function tests if the given GMEC is controllable for the marked Petri Net and if the initial marking belongs to the set of reachability of the net with the constraint.

5.2 Checkgmec2

Sintax:

```
[Wc, Kc, Wu, Ku] = checkgmec2(C,M0,W,K,u)
```

Description

Checkgmec2 function tests if each GMEC of the set W is controllable for the marked Petri net and if the initial marking belongs to the set of reachability of the net with each of the constrains. The function returns two sets: (W_c, K_c) with the controllable GMECs and (W_u, K_u) with the uncontrollable GMECs.

5.3 Controllablegmec

Sintax

```
[W,K] = controllablegmec(C,M0,w,k,u)
```

Description

Controllablegmec finds all the minimally restrictive controllable constraints that satisfy the given uncontrollable constraint. If the given GMEC is controllable the function returns it. If the given GMEC is uncontrollable the function returns a set (W, K) of controllable GMECs that can be used to set the uncontrollable constraint.

5.4 Gmecsbs

Sintax

```
[W, K] = gmecsbs(C,M0,w,k,u)
```

Description

Gmecsbs is an explanation of the algorithm used to find the controllable GMECs that satisfy an uncontrollable GMEC used in Controllablegmec function. It gives the same results of Controllablegmec.

5.5 Monitorplace

Sintax

```
[Cs, Ms0] = monitorplace(C,M0,w,k,u)
```

Description

Monitorplace function finds the row of the incidence matrix and initial marking of the monitor place that satisfies the controllable GMEC.

5.6 Monitorplace2

Sintax

```
[CS, MS0] = monitorplace2(C,M0,W,K,u)
```

Description

Monitorplace2 function finds the rows of the incidence matrix and initial marking of each monitor place used to satisfy the set of controllable GMECs.

Appendix A

Source code of the algorithm used to find the controllable GMECs

```
function [W, K] = controllablegmec(C, M0, w, k, u)
% Help [...]
ni = nargin;
no = nargout;
error(nargchk(5,5,ni));
% check of input size
[Cm,Cn] = size(C);
[M0m,M0n] = size(M0);
[wm,wn] = size(w);
[km,kn] = size(k);
[um,un] = size(u);
if (Cm~=M0m | Cm~=wm | M0n~=1 | wn~=1 | km~=1 | kn~=1 | um~=1 | un>Cn)
   fprintf('   Error: check the size of inputs\n');
elseif (max(u)>Cn)
```

```
   fprintf('   Error: check values of u vector\n');
% verify that M0 belongs to legal markings given by (w,k)
elseif (k-w'*M0 < 0)
   fprintf('   Initial marking not belonging to  R(N,M0,w,k)\n');
   fprintf('   This method can not be used\n');
else
   if (un == Cn)
      fprintf('\n   All the transitions are uncontrollable!\n');
end
   Cu = C(:,u);
   if (isempty(find(w'*Cu > 0)))
      %the GMEC is controllable
      fprintf('\n   (w,k) is controllable\n');
      if (no ~= 0)
         W = w;
         K = k;
      end
   else
      %the GMEC is not controllable
      fprintf('\n   (w,k) is not controllable\n\n');
      GMEC = [];
      trchecked = zeros(1,un);
      % ALGORITHM START
      nw = 1;
      wCu = [w'*Cu zeros(nw,Cm) ones(nw,1)];
      % inizial matrix
      A = [Cu eye(Cm) zeros(Cm,1); wCu];
      nw = size(A,1) - Cm;
      while (nw~=0)
         j = find(A(Cm+1,1:un)>0);
         % covering control
         if (~(~isempty(j) & trchecked(j(1))))
            r = find(Cu(:,j(1))<0); % vettore colonna
            rowr = size(r,1);
            for h=1:rowr
               wCu = A(r(h),:).*A(Cm+1,j(1))-A(Cm+1,:).*A(r(h),j(1));
               if (isempty(find(wCu(1:un)>0)))
                  % w1 =  r1 + r2 * w
                  w1 = wCu(un+1:un+Cm)+wCu(un+Cm+1)*w';
                  % k1 = r2(k+1)-1
                  k1 = wCu(un+Cm+1)*(k+1)-1;
                  GMEC = [GMEC; w1 k1];
               else
                  A = [A; wCu];
               end
            end
         end
```

```
            trchecked(j(1)) = 1;
            rowA = size(A,1);
            % delete analyzed row
            A = [A(1:Cm,:); A(Cm+2:rowA,:)];
            nw = size(A,1) - Cm;
         end
         [nGMEC,h] = size(GMEC);
         fprintf('    %d controllable GMEC founded\n',nGMEC);
         if (no == 0)
            for i=1:nGMEC
               fprintf('\n w%d = ',i);
                  disp(GMEC(i,1:h-1));
                  fprintf(' k%d = ',i);
               disp(GMEC(i,h));
            end
         else
            W = GMEC(:,1:h-1);
            K = GMEC(:,h);
         end
      end
   end
```

References

[1] Francesco Basile, Pasquale Chiacchio, Alessandro Giua, "Suboptimal Supervisory Control of Petri Nets in presence of Uncontrollable Transitions via Monitor Places"