On the enforcement of a class of nonlinear constraints on Petri nets

YuFeng Chen a,c, Zhi Wu Li b,a,d, Kamel Barkaoui c, Alessandro Giua e,f,a

a School of Electro-Mechanical Engineering, Xidian University, No. 2 South Taibai Road, Xi’an 710071, China
b Institute of Systems Engineering, Macau University of Science and Technology, Macau
c Cedric Lab and Computer Science Department, Conservatoire National des Arts et Métiers, Paris 75141, France
d Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
e Aix Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, 13397, Marseille, France
f DIEE, University of Cagliari, 09123 Cagliari, Italy

ARTICLE INFO

Article history:
Received 29 July 2013
Received in revised form 17 October 2014
Accepted 29 January 2015
Available online 23 March 2015

Keywords:
Discrete event system
Petri net
Supervisory control
Supervisor
Nonlinear constraint

ABSTRACT

This paper deals with the enforcement of nonlinear constraints on Petri nets. A supervisory structure is proposed for a class of nonlinear constraints. In order to enforce a nonlinear constraint on a Petri net, we propose a transition transformation technique to replace a transition in an original net by a set of transitions. Then, a control place is designed to control the firing of these transitions, aiming to enforce the nonlinear constraint. The proposed supervisor is maximally permissive in the sense that it can make all markings in the admissible-zone reachable and all markings in the forbidden-zone unreachable. The proposed method is applicable to bounded Petri nets. Finally, a number of examples are provided to demonstrate the proposed approach.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Petri nets (Murata, 1989) are a powerful tool to model and analyze discrete event systems (DESs). They have been widely used for deadlock control, scheduling and planning, and performance evaluation in a variety of resource allocation systems (Barkaoui & Abdallah, 1995; Li, Liu, Hanisch, & Zhou, 2012; Li, Wu, & Zhou, 2012; Zhang et al., 2015). Supervisory control is a suitable mechanism to enforce external constraints on a system to be controlled. In the framework of Petri nets, a supervisor that enforces supervisory control specifications is often represented by a set of control places.

Constraints associated with reachable states in a DES are a typical and important control specification in supervisory control theory of DESs. Many specifications can be converted into linear constraints. For example, deadlock problems in Petri nets are usually dealt with by finding a set of constraints, with respect to the markings, that can prevent the system from reaching deadlock states (Barkaoui, Couvreur, & Klai, 2005; Chen & Li, 2013; Li & Zhao, 2008; Li & Zhou, 2009). Most control requirements in system control design can be directly represented by a set of constraints.

Generally, there are two classes of constraints in Petri nets: linear and nonlinear. Linear constraints, also called generalized mutual exclusion constraints (GMECs) (Giua, DiCesare, & Silva, 1992; Ma, Li, & Giua, in press), play an important role in the development of supervisors for a system modeled by Petri nets. Many efforts have been done to enforce a GMEC by constructing a place invariant (PI) (Banaszak & Krogh, 1990; Chen, Li, & Zhou, 2012; Yamalidou, Moody, Lemmon, & Antsaklis, 1996). The PI-based approach is well-established and widely used by researchers and engineers. Yamalidou et al. (1996) study a variety of GMECs and design control places to enforce them by constructing PIs. Iordache and Antsaklis (2005, 2007) present an approach to the implementation of disjunctive GMECs. The work in Iordache and Antsaklis (2006) provides a good survey on the design of control places by PI based methods. Up to now, a lot of work has been done to deal with deadlocks by Petri nets (Chen, Li, & Barkaoui, 2014; Ghaffari, Rezg, & Xie, 2003; Huang, Jeng, Xie, & Chung, 2006; Li & Zhou, 2004, 2006, 2008; Liu, Li, & Zhou, 2010; Wu, Zhou, & Li, 2008). In fact, almost all of them compute control places by PIs (Chen, Li, & Zhou,
A transition \( t \in T \) is enabled at marking \( M \) if \( \forall p \in s^*t \), \( M(p) \geq W(p, t) \). This fact is denoted as \( M(t) \). Once an enabled transition \( t \) fires, it yields a new marking \( M' \), denoted as \( M(t) \), \( M'(p) = M(p) - W(p, t) + W(t, p) \). The set of reachable markings of net \( N \) with initial marking \( M_0 \) is denoted by \( R(N, M_0) \). It can be graphically expressed by a reachability graph, denoted as \( G(N, M_0) \). It is a directed graph whose nodes are markings in \( R(N, M_0) \) and arcs are labeled by the fired transitions.

Let \( (N, M_0) \) be a system network with \( N = (P, T, F, W) \). A transition \( t \in T \) is live if \( \forall M \in R(N, M_0), \exists M'(t) \in R(N, M) \). \( (N, M_0) \) is live if \( \forall t \in T, t \) is live. It is dead if \( \exists t \in T, M_0(t) \).

2.2. Generalized mutual exclusion constraint

A GMEC (Giua et al., 1992) is a control requirement that limits a weighted sum of tokens contained in a subset of places. Let \( |N| \) be the incidence matrix of a plant with \( n \) places and \( m \) transitions. A GMEC can be expressed as:

\[
\sum_{i=1}^{n} w_i \cdot \mu_i \leq k
\]

where \( \mu_i \) denotes the number of tokens in place \( p_i \) at any reachable marking, and \( w_i \) and \( k \) are non-negative integers. Eq. (1) can be represented as a vector form, i.e.,

\[
\bar{w}^T \cdot \bar{\mu} \leq k
\]

where \( \bar{w} \) is a weight vector of nonnegative integers with \( \bar{w}(i) = w_i \), \( \bar{\mu} \) is a vector of nonnegative integers with \( \bar{\mu}(i) = \mu_i \) and \( k \) is a positive integer. A GMEC is usually denoted as \( (\bar{w}, k) \).

By introducing a non-negative slack variable \( \mu_s \), Eq. (2) becomes

\[
\bar{w}^T \cdot \bar{\mu} + \mu_s = k
\]

where \( \mu_s \) represents the marking of control place \( p_s \), generally called a monitor. The firing of a transition \( t \) modifies the tokens in \( p_s \) by a constant:

\[
\Delta(t) = -\bar{w}^T \cdot [N]\(\bullet, t\).
\]

In fact, \( VM_1, M_2 \in R(N, M_0) \) with \( M_1 = M_2 + [N]\(\bullet, t\) \), we have \( \Delta(t) = M_1(p_s) - M_2(p_s) \). Thus, the incidence vector \( [N]_i \) of \( p_i \) can be computed by:

\[
[N]_i = -\bar{w}^T \cdot [N].
\]

The initial marking \( M_0(p_s) \) of \( p_s \) can be calculated as follows:

\[
M_0(p_s) = k - \bar{w}^T \cdot M_0.
\]

3. Generalizations of arbitrary marking constraints

In this section, we present basic concepts of nonlinear constraints in Petri nets in the sense of reachability graph analysis. A constraint for a Petri net is in general a predicate with respect to the states (markings) of the Petri net. Let \( c \) be a constraint that restricts the tokens contained in a subset of places of a Petri net model \( (N, M_0) \). In this work, the constraints are only associated with markings while no firing vectors of transitions are considered.

**Definition 1.** Let \( c \) be a constraint and \( M \in R(N, M_0) \) a marking of a net \( (N, M_0) \). The function \( F(c, M) \) is defined as \( F(c, M) = 1 \) if \( M \) satisfies \( c \) and \( F(c, M) = 0 \) otherwise.

Given a constraint \( c \), the reachable markings of a net are classified into two groups: admissible ones that satisfy \( c \) and inadmissible ones that do not satisfy \( c \), as defined below:

**Definition 2.** Let \( c \) be a constraint of a Petri net model \( (N, M_0) \). A marking \( M \in R(N, M_0) \) is said to be admissible with respect to \( c \) if \( F(c, M) = 1 \). The set of admissible markings of \( c \) is denoted by \( M_c \). A reachable marking \( M \in (N, M_0) \) is said to be inadmissible with respect to \( c \) if \( F(c, M) = 0 \). The set of inadmissible markings of \( c \) is denoted by \( M_{\neg c} \).
A Reachability Graph

\[ \text{Fig. 1. The AZ and the FZ in a reachability graph.} \]

Given a constraint for a Petri net model \((N, M_0)\), we assume that its initial marking always satisfies the constraint. Then, the reachability graph of \((N, M_0)\) can be classified into two parts: an admissible-zone (AZ) and a forbidden-zone (FZ). There may exist some admissible markings that cannot be reached from the initial marking through admissible markings only. In this case, these admissible markings cannot be reached if all inadmissible markings are forbidden, i.e., they should be included in the FZ though they are admissible. Hence, the AZ includes the maximal set of admissible markings of \(c\), which are reachable from the initial marking without leaving \(M_c\), whose set is denoted as \(M_c^*\), and the FZ contains all the other reachable markings, i.e., all the inadmissible markings of \(c\) and the admissible markings that cannot be reached without leaving the AZ, whose set is denoted as \(M_c^*\). It is obvious that \(M_c^* \subseteq M_c\) and \(M_c^* \subseteq M_c^*\). The partition of a reachability graph is demonstrated in Fig. 1.

A supervisor is \textit{maximally permissive}, or said to be \textit{optimal}, if it can always disable any transition whose firing leads to a marking in the FZ and does not disable any transition whose firing leads to a marking in the AZ. In this sense, a maximally permissive supervisor for a constraint \(c\) should keep all the admissible markings in the AZ of \(c\) and exclude the reachability of any marking in the FZ of \(c\).

A border forbidden marking (BFM) of \(c\) is a marking in the FZ that is a direct successor of some marking in the AZ, as shown in Fig. 1. Mathematically, the set of BFMs, denoted by \(M_B\), is defined as follows:

\textbf{Definition 3.} Let \(c\) be a constraint on \((N, M_0)\). The set of BFMs is defined as \(M_B = \{M | M \in M_c^*, \exists \hat{M} \in M_c^* \setminus M, \exists t \in T_s. M’(t) = M\}\).

If all BFMs cannot be reached, their successors cannot be reached. Thus, there is no need to compute the whole reachability graph of a net system.

\textbf{4. Design of supervisory structures for nonlinear constraints}

In this section, we define a new class of constraints that are inspired by GMECs (Giua et al., 1992) but not linear. We propose a supervisory structure to implement a nonlinear constraint, which can optimally enforce it, i.e., all admissible markings in the AZ of the constraint are reachable.

\textbf{4.1. Synthesis of an optimal supervisory structure}

In this section, we develop an approach to design a supervisor for a class of nonlinear constraints, namely an additive separable function, as defined below.

\textbf{Definition 4.} An additive separable constraint \(c\) involves the sum of a number of functions \(f_i(\mu_i)\) \((i \in \{1, 2, \ldots, n\})\) and a constant \(\beta\), formally,

\[ f_1(\mu_1) + f_2(\mu_2) + f_3(\mu_3) + \cdots + f_n(\mu_n) \leq \beta \]  \hspace{1cm} (7)

where \(f_i(\mu_i)\) is a nonlinear function of \(\mu_i\) and \(\mu_i\) denotes the marking of \(p_i\), \(i \in \{1, 2, \ldots, n\}\).

For instance, \(f_1(\mu_1) = \mu_1 \cdot \mu_2\) is a nonlinear function of \(\mu_1\). The support \(\|c\|\) of an additive separable constraint \(c\) is defined as the set of places \(p_i\) such that \(f_i(\mu_i)\) is not the zero function, i.e., \(\|c\| = \{p_i | f_i(\mu_i) \neq 0\}\). An additive separable constraint can be transformed into an equality by introducing a non-negative slack variable \(\mu_s\) (the marking of control place \(p_s\)), as presented below:

\[ f_1(\mu_1) + f_2(\mu_2) + f_3(\mu_3) + \cdots + f_n(\mu_n) + \mu_s = \beta \] \hspace{1cm} (8)

In such a case the firing of a transition \(t\) at a marking \(M\) modifies the slack variable \(\mu_s\) of a quantity that depends on the marking \(M\):

\[ \Delta(t, M) = f(M) - f(M + [N](\mu, t)) \] \hspace{1cm} (9)

where \(f(\mu) = f_1(\mu_1) + f_2(\mu_2) + f_3(\mu_3) + \cdots + f_n(\mu_n)\). Eq. (9) can be rewritten as

\[ \Delta(t, M) = \sum_{i=1}^{n} [f_i(M(p_i)) - f_i(M(p_i) + [N](p_i, t))] \] \hspace{1cm} (10)

\textbf{Property 1.} Let \( t \) be a transition and \( N_{\Delta t} = \{ \Delta(t, M) | M \in R(N, M_0) \} \). Then, \( |N_{\Delta t}| \) (the cardinality of \( N_{\Delta t} \)) is finite.

\textbf{Proof.} It can be easily obtained by the fact that \( R(N, M_0) \) (the cardinality of \( R(N, M_0) \)) is finite.

In fact, \( \Delta(t, M) \) has no relation with the marking of a place \( p_i \) if \( [N](p_i, t) = 0 \) or \( p_i \notin \|c\| \). Hence, Eq. (10) can be simplified as

\[ \Delta(t, M) = \sum_{p_i \in \{^*t \cup ^*t'\} \cap \|c\|} [f_i(M(p_i)) - f_i(M(p_i) + [N](p_i, t))] \] \hspace{1cm} (11)

Eq. (11) motivates us to transform a transition \( t \) into a set of transitions to represent the different modified quantities of control place \( p_s \). By Eq. (11), \( \Delta(t, M) \) is a sum of the token modifications \( [f_i(M(p_i)) - f_i(M(p_i) + [N](p_i, t))] \), where \( p_i \in \{^*t \cup ^*t'\} \cap \|c\| \). Hence, we can design a supervisory structure for each nonlinear function \( f_i(\mu_i) \) respectively and then combine them together to enforce the nonlinear constraint. Without loss of generality, let us design a supervisory structure for \( f_1(\mu_1) \). For the sake of brevity, let \( g(\mu_2, \mu_3, \ldots, \mu_n) = f_2(\mu_2) + f_3(\mu_3) + \cdots + f_n(\mu_n) \). Then, Eq. (8) can be written as

\[ f_1(\mu_1) + g(\mu_2, \mu_3, \ldots, \mu_n) + \mu_s = \beta \] \hspace{1cm} (12)

In order to enforce the nonlinear constraint, each of the input and output transitions of the places involved in the nonlinear constraints is replaced by a set of transitions. Next, we show details of the design of the supervisory structure. We consider the supervisory structure for \( f_1(\mu_1) \) in Eq. (7). An algorithm is presented as follows.

\textbf{Algorithm 1.} Design of a supervisory structure for a nonlinear function

\textbf{Input:} A bounded Petri net model \((N, M_0)\) and a nonlinear function \(f_1(\mu_1)\)

\textbf{Output:} A supervisory structure for \( f_1(\mu_1) \)

\begin{enumerate}
\item Let \( K_{p_1} \) be the upper bound of \( p_1 \) and \( \mathbb{P}_{\mathcal{R}_{p_1}} = \{ (x, y) | y \leq K_{p_1}, y = x \} \). \forall t_j \in \mathcal{T}_1 \cup \mathcal{P}_1^*, \text{ a set of transitions } \mathcal{T}_2 \text{ is designed to replace } t_j, \text{ where } \mathcal{T}_2 = \{ t_j + \mathcal{R}_{p_1} | (x, y) \in \mathbb{P}_{\mathcal{R}_{p_1}}, z = [N](p_1, t_j) \}. \end{enumerate}

1 Since the complexity of the supervisory structure increases with the upper bound of \( p_1 \), we make \( K_{p_1} \) as small as possible. For example, if the capacity of \( p_1 \) decided by the original net structure is 3 but the tokens in \( p_1 \) is limited to be no more than 2 by a given constraint, then \( K_{p_1} \) should be the smaller one, i.e., 2.
and make all admissible markings reachable.

By introducing a non-negative slack variable µ, (13) is used to illustrate the proposed approach.

\[ \mu_1 \cdot \mu_1 + \mu_2 \leq 4. \tag{13} \]

By introducing a non-negative slack variable µ, the inequality constraint can be transformed into an equality as follows:

\[ \mu_1 \cdot \mu_1 + \mu_2 + \mu_3 = 4 \tag{14} \]

where µ3 represents the marking of control place p3. Suppose that the net to be controlled is shown in Fig. 2.²

It can be seen that place p1 is unbounded in the original net but we can obtain its upper bound by Eq. (13), i.e., Kp1 = 2. Then, we can design the supervisor as shown in Fig. 3. There are four added transitions: t1, t2, t3, t4, and t5, where t1 and t2 in the original net model are replaced by \( t_{11} \) and \( t_{21} \), respectively. It can be seen that a transition \( t_{ij} \) is enabled at a marking M if \( M(p_i) = x \) and \( M(p_j) \geq W(p_j, t_{ij}) \). Once \( t_{ij} \) fires at M, the marking of p1 changes to \( M(p_1) = f_i(x) - f_j(x) \). The reachability graph of the controlled net in Fig. 3 is shown in Fig. 4. We can verify that the controlled net is live with 10 reachable markings as shown in Table 2. That is to say, the proposed supervisor can implement the nonlinear constraint and make all admissible markings reachable. 

---

### Table 1

<table>
<thead>
<tr>
<th>p</th>
<th>W(p, t_{i,j}⁻)</th>
<th>W(t_{i,j}⁻, p)</th>
<th>M0(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1 _ _ p1</td>
<td>W(p1, t1)</td>
<td>W(t1, p1)</td>
<td>M0(p1)</td>
</tr>
<tr>
<td>p2</td>
<td>W(p2, t2)</td>
<td>W(t2, p2)</td>
<td>M0(p2)</td>
</tr>
<tr>
<td>p3</td>
<td>W(p3, t3)</td>
<td>W(t3, p3)</td>
<td>M0(p3)</td>
</tr>
<tr>
<td>p4</td>
<td>W(p4, t4)</td>
<td>W(t4, p4)</td>
<td>M0(p4)</td>
</tr>
</tbody>
</table>

---

### Example 1

A simple constraint, Eq. (13), is used to illustrate the proposed approach.

\[ \mu_1 \cdot \mu_1 + \mu_2 \leq 4. \tag{13} \]

By introducing a non-negative slack variable µ, the inequality constraint can be transformed into an equality as follows:

\[ \mu_1 \cdot \mu_1 + \mu_2 + \mu_3 = 4 \tag{14} \]

where µ3 represents the marking of control place p3. Suppose that the net to be controlled is shown in Fig. 2.²

It can be seen that place p1 is unbounded in the original net but we can obtain its upper bound by Eq. (13), i.e., Kp1 = 2. Then, we can design the supervisor as shown in Fig. 3. There are four added transitions: t1, t2, t3, t4, and t5, where t1 and t2 in the original net model are replaced by \( t_{11} \) and \( t_{21} \), respectively. It can be seen that a transition \( t_{ij} \) is enabled at a marking M if \( M(p_i) = x \) and \( M(p_j) \geq W(p_j, t_{ij}) \). Once \( t_{ij} \) fires at M, the marking of p1 changes to \( M(p_1) = f_i(x) - f_j(x) \). The reachability graph of the controlled net in Fig. 3 is shown in Fig. 4. We can verify that the controlled net is live with 10 reachable markings as shown in Table 2. That is to say, the proposed supervisor can implement the nonlinear constraint and make all admissible markings reachable. 

---

### Table 2

<table>
<thead>
<tr>
<th>i</th>
<th>M1(p1)</th>
<th>M2(p1)</th>
<th>M3(p1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Theorem 1.** Let M and M’ be two markings in a plant net model \((N, M_0)\) with \( M’(p_1) \) = \( M(p_1) \) and \( M’(p_2) \) = \( M(p_2) \), i.e., \( i \in \{2, 3, \ldots, n\} \), where \([N]\) is the incidence matrix. Suppose that M is a reachable marking that satisfies Eq. (12), then the proposed supervisory structure due to Algorithm 1 can ensure that

1. M’ is reachable from M and satisfies the PI equality Eq. (12) if M’ satisfies Eq. (7); and
2. M’ is unreachable if M’ does not satisfy Eq. (7).

² Note that we only show the subnet generated by \( [p_1, p_2] \cup [p_1 \cup p_1 \cup p_2 \cup p_2] \). Since Eq. (13) is concerned with the tokens in p1 and p2 only.
Proof. First, we consider Case (1). Since $M$ satisfies Eq. (12), we have $f_1(M(p_1)) + g(M) + M(p_1) = \beta$, where $g(M) = f_2(M(p_2)) + f_3(M(p_3)) + \cdots + f_N(M(p_N))$. Since $M'$ satisfies Eq. (7), $f_1(M'(p_1)) + g(M') \leq \beta$ is true, where $g(M') = f_2(M'(p_2)) + f_3(M'(p_3)) + \cdots + f_N(M'(p_N))$. In this case, we have $M(p_1) = \beta - f_1(M'(p_1)) - g(M) \geq f_1(M'(p_1)) + g(M') - f_1(M'(p_1)) - g(M) = f_1(M'(p_1)) - f_1(M'(p_1))$. Next, we consider two subcases (1.a) $f_1(x) - f_1(y) < 0$ and (1.b) $f_1(x) - f_1(y) > 0$.

For Case (1.a), consider $p_1$ and the added place $\bar{p}_1$. The added transition $t_{x,y}^\alpha$ with $x = M(p_1)$ and $y = M(p_1)$ can fire at $M$ owing to $M(p_1) \geq f_1(M'(p_1)) - f_1(M'(p_1))$ and $W(p_1, t_{x,y}^\alpha) = f_1(y) - f_1(x)$. If $t_{x,y}^\alpha$ fires, we obtain a new marking $M'$ with $M(p_1) = M(p_1) - (f_1(y) - f_1(x))$. By $M(p_1) = M(p_1)$, $i \in \{2, 3, \ldots, n\}$, $g(M') = g(M)$ holds. Therefore, we have $f_1(M'(p_1)) + g(M') + M(p_1) = \beta$, i.e., $M'$ is reachable from $M$ and satisfies the PL inequality Eq. (12).

Similarly, for Case (1.b), since $t_j$ is enabled at $M$, $t_{x,y}^\alpha$ is enabled at $M$ with $x = M(p_1)$ and $y = M'(p_1)$. Thanks to $W(t_{x,y}^\alpha) = f_1(x) - f_1(y)$, once $t_{x,y}^\alpha$ fires, it yields a marking $M'$, i.e., $M'$ is reachable with $M'(p_1) = M(p_1) + f_1(y) - f_1(x)$. By $M'(p_1) = M(p_1)$, $i \in \{2, 3, \ldots, n\}$, $g(M') = g(M)$ is true. Hence, we have $f_1(M'(p_1)) + g(M) + M(p_1) = \beta$, i.e., $M'$ is reachable and satisfies the PL inequality Eq. (12).

Next, we consider Case (2). By Eq. (12), we have $M(p_1) = \beta - f_1(M'(p_1)) - g(M)$. Since $M'$ does not satisfy Eq. (7), $\beta < f_1(M'(p_1)) + g(M)$ is true. Thus, we have $M(p_1) < f_1(M'(p_1)) + g(M) = f_1(M'(p_1)) - f_1(M'(p_1))$, leading to the fact that the added transition $t_{x,y}^\alpha$ with $x = M(p_1)$ and $y = M'(p_1)$ is disabled by $p_1$. As a result, $M'$ is unreachable. 

Theorem 1 indicates that the proposed supervisory structure due to Algorithm 1 can implement the nonlinear function $f_1(\mu_1)$. In this case, we can design such a supervisory structure for each nonlinear function $f_\mu(i)$ ($i \in \{1, 2, \ldots, n\}$) in Eq. (7) and merge it into a supervisory by the shared places $p_1$ with $M_0(p_1) = \beta$. Then, the obtained supervisory can implement the nonlinear constraint Eq. (7). In the following, an algorithm is presented to merge the supervisory structure for each nonlinear function in Eq. (7).

Algorithm 2. Design of a supervisory structure for an additive separable constraint

Input: A bounded Petri net $(N, M_0)$ and an additive separable constraint $c$

Output: A supervisory structure for $c$

1. Add a control place $p_1$ with $M_0(p_1) = \beta - f(M_0)$.
2. foreach $[p_i \in [c]]$ do { Add a complementary place $\bar{p}_i$ of $p_i$ with $M_0(\bar{p}_i) = K_{p_i} - M_0(p_i)$, where $K_{p_i}$ is the upper bound of $p_i$.
3. foreach $[t_j \in [p_1]]$ do { A set of transitions $t_j$ is designed to replace $t_j$, where $t_j^\alpha = \{t_{x,y}^\alpha \mid (x, y) = [p_i, \bar{p}_i], z = [N](p_i, t_j)\}$. For all $t_j^\alpha$, add arcs $W(\bar{p}_i, t_j^\alpha) = K_{p_i} - p_1$ and $W(p_1, t_j^\alpha) = x$. For all $p_k \in P\setminus[p_i]$, $W(t_j^\alpha, p_k) = W(p_k, t_j^\alpha)$.
4. Remove the transition $t_j$.
5. Output the obtained supervisory structure.
6. End.

Proposition 1. Let $(N^a, M_0^a)$ be a supervisor obtained by Algorithm 2. Then, the obtained supervisor can implement the nonlinear constraint Eq. (7).

Proof. Let $M$ and $M'$ be two markings in a plant net model $(N, M_0)$ with $M' = M + [N](e, t)$. Suppose that $M$ is a reachable marking that satisfies Eq. (12). Then, we can prove that the proposed supervisory structure by Algorithm 2 can ensure that

1. $M'$ is reachable from $M$ and satisfies the PL equality Eq. (8) if $M'$ satisfies Eq. (7); and
2. $M'$ is unreachable if $M'$ does not satisfy Eq. (7).

For Case (1), we first consider that the tokens in two places $p_1$ and $p_2$ are changed by firing $t_j$, i.e., $[N](p_1, t_j) \neq 0$ and $[N](p_2, t_j) \neq 0$. Let $x_1 = M(p_1)$, $x_2 = M(p_2)$, $y_1 = M'(p_1)$, and $y_2 = M'(p_2)$. If $M'$ satisfies Eq. (7), then there exists a transition $t_{x,y}^\alpha$ in the set of added transitions representing $t_j$. Similar to the proof of Theorem 1, $t_{x,y}^\alpha$ is enabled at $M$ only if $x_1 = M(p_1)$ and $x_2 = M(p_2)$. In this case, $t_{x,y}^\alpha$ can be fired and once it fires, it yields the new marking $M'$ with $M'(p_1) = y_1$ and $M'(p_2) = y_2$. It can also be verified that $[N](t_{x,y}^\alpha) = f_1(x_1) - f_1(y_1) + f_2(x_2) - f_2(y_2)$, where $[N]$ is the incidence vector of the control place $p_1$. Then, we have $M'(p_1) = M(p_1) + f_1(x_1) - f_1(y_1) + f_2(x_2) - f_2(y_2)$. Since $M'$ satisfies Eq. (8), i.e., $f_1(x_1) + f_2(x_2) + \cdots + f_N(M(p_1)) = M(p_1) = \beta$, we have $f_1(y_1) = f_1(y_2) + f_2(y_2) + f_2(y_2) + \cdots + f_N(M(p_1)) + M(p_1) = f_1(y_1) + f_1(y_2) + f_2(y_2) + \cdots + f_N(M(p_1)) + M(p_1) + f_1(x_1) - f_1(y_1) + f_2(x_2) - f_2(y_2) = f_1(x_1) + f_2(x_2) + \cdots + f_N(M(p_1)) + M(p_1) = \beta$. Hence, $M'$ satisfies the PL equality Eq. (8). Now, we can similarly prove that Case (1) holds if the tokens in more than two places are changed by firing $t_j$.

Next, we consider Case (2). Similarly, we first consider that the tokens in two places $p_1$ and $p_2$ are changed by firing $t_j$. By Eq. (12), we have $M(p_1) = \beta - f_1(M(p_1)) - f_1(M'(p_1)) - \cdots - f_1(M(p_N))$. Since $M'$ does not satisfy Eq. (7), $\beta < f_1(M'(p_1)) + f_2(M'(p_2)) + \cdots + f_N(M'(p_N))$ is true. Thus, we have $M(p_1) < f_1(M'(p_1)) + f_2(M'(p_2)) + \cdots + f_N(M'(p_N)) - f_1(M'(p_1)) - f_1(M'(p_2)) - \cdots - f_1(M(p_N)) - f_1(M'(p_1)) - f_1(M'(p_2)) - \cdots - f_1(M(p_N)).$ leading to the fact that the added transition $t_{x,y}^\alpha$ with $x = M(p_1)$ and $y = M'(p_1)$ is disabled by $p_1$. As a result, $M'$ is unreachable. Finally, we conclude that the supervisory obtained by Algorithm 2 can implement the nonlinear constraint Eq. (7).
4.2. Structural complexity

In this section, we discuss the structural complexity of the proposed supervisor. The number of added places is \( n + 1 \) since there are \( n \) complementary places \( \overline{0} \) (\( i = 1, 2, \ldots, n \)) and a control place \( p_r \). Next, we discuss the number of the added transitions as follows.

1. First, we consider the case that a transition \( t_j \) modifies the marking of just one place \( p_i \) in the support of \( c \). Then, \( t_j \) is replaced by \( [\overline{1}]_{i_j} [\overline{1}]_{j} = [N](p_i, t_j) \) transitions.

2. Second, we consider the case that \( t_j \) modifies the marking of two places \( p_i \) and \( p_k \) in the support of \( c \). Then, at the iteration step to design the supervisory structure for \( p_i \), \( t_j \) is replaced by \( [\overline{1}]_{i_j} [\overline{1}]_{j} \) transitions, where \( z_{i_j} = [N](p_i, \{t_j\}) \). At the iteration step to design the supervisory structure for \( p_k \), for each newly added transition in \( [\overline{1}]_{i_j} [\overline{1}]_{j} \), it is replaced by \( [\overline{1}]_{i_j} [\overline{1}]_{j} \) transitions, where \( z_{i_j} = [N](p_k, \{t_j\}) \). Therefore, the total number of added transitions to replace \( t_j \) is \( [\overline{1}]_{i_j} [\overline{1}]_{j} \).

3. Third, we consider the case that \( t_j \) modifies the marking of \( r \) places \( p_{i_1}, \ldots, p_{i_r} \) in the support of \( c \). Then, the total number of added transitions to replace \( t_j \) is \( [\overline{1}]_{i_j} [\overline{1}]_{j} \cdot [\overline{1}]_{i_j} [\overline{1}]_{j} \).

4. Let \( T_r = \{ t \in \mathcal{T} \mid \langle t, t^* \rangle \cap \|c\| = r \} \) be such a set of transitions that each transition in it modifies the markings of \( r \) places in the support of \( c \), and denote the places in \( \mathcal{T}_r \) by \( \{p_{i_1}, \ldots, p_{i_r}\} \). Then, the total number of added transitions is \( \sum_{t \in \mathcal{T}_r} \prod_{k=1}^{n} [\overline{1}]_{j} \).

According to the above discussions, it can be seen that the proposed method suffers from supervisory complexity problem if there are too many transitions that modify the marking of multiple places in \( \|c\| \). The reason is that the proposed method is applicable to all additive separable constraints. In fact, the supervisory structure can be simply reduced for some special constraints. In the following, we provide two simple examples to demonstrate this point.

Example 2. We consider the following example

\[
 f_i(\mu_i) = \begin{cases} 
 0 & \text{if } \mu_i \leq a \\
 \mu_i - a & \text{if } \mu_i > a 
\end{cases} \quad (15)
\]

where \( 1 \leq a \leq K_p \) and \( K_p \) is the upper bound of \( p_i \). In Eq. (15), a transition \( t \) that modifies the marking of place \( p_i \) needs only to be split into two transitions: one that does not modify the marking of \( p_i \) and the other that modifies the marking of place \( p_i \) by one token. The corresponding supervisory structure is shown in Fig. 5, where \( t_1 \) is split into two transitions: \( t_{1-a} \) whose firing does not modify the marking of \( p_i \) at a marking \( M \) if \( M(p_i) < a \) and \( t_{1-a} \) whose firing reduces the marking of place \( p_i \) by one token at a marking \( M \) if \( M(p_i) \geq a \). Similarly, \( t_2 \) is split into two transitions: \( t_{2-a} \) whose firing does not modify the marking of \( p_i \) at a marking \( M \) if \( M(p_i) < a \) and \( t_{2-a} \) whose firing increases the marking of place \( p_i \) by one token at a marking \( M \) if \( M(p_i) > a \).

**Example 3.** A similar nonlinear function is shown in the following.

\[
 f_i(\mu_i) = \begin{cases} 
 0 & \text{if } \mu_i \leq a \\
 b & \text{if } \mu_i > a \n\end{cases} \quad (16)
\]

where \( 1 \leq a \leq K_p \) and \( K_p \) is the upper bound of \( p_i \). In Eq. (16), a transition \( t \) that modifies the marking of place \( p_i \) needs only to be split into three transitions: two that do not modify the marking of \( p_i \) and one that modifies the marking of place \( p_i \) by \( b \) tokens. The corresponding supervisory structure is shown in Fig. 6, where \( t_1 \) is split into three transitions: \( t_{1-a} \) whose firing does not modify the marking of \( p_i \) at a marking \( M \) if \( M(p_i) < a \), \( t_{1-a} \) whose firing does not modify the marking of \( p_i \) at a marking \( M \) if \( M(p_i) > a \), and \( t_{1-a} \) whose firing reduces the marking of place \( p_i \) by \( b \) tokens at a marking \( M \) if \( M(p_i) = a \). Similarly, \( t_2 \) is split into three transitions: \( t_{2-a} \) whose firing does not modify the marking of \( p_i \) at a marking \( M \) if \( M(p_i) < a + 1 \), \( t_{2-a+1} \) whose firing does not modify the marking of \( p_i \) at a marking \( M \) if \( M(p_i) > a + 1 \), and \( t_{2-a+1} \) whose firing increases the marking of place \( p_i \) by \( b \) tokens at a marking \( M \) if \( M(p_i) = a + 1 \).

Examples 2 and 3 show that the supervisory structure for a nonlinear function \( f_i(\mu_i) \) can be reduced if \( f_i(\mu_i) \) can be divided into some linear parts. Then, for each linear part, a transition \( t \) that modifies the marking of place \( p_i \) should modify the marking in the control place \( p_i \) by a constant. Hence, we need only one transition to represent the linear part. As a result, the number of the added transitions to replace \( t \) is reduced. By the two examples, we can see that the proposed approach is particularly fit for piecewise linear functions since in that case the complexity of the resulting supervisor is much better than that suggested by the worst-case analysis.

4.3. An example for the proposed supervisory structure

In this section, an example is proposed to demonstrate the proposed supervisory structure.

**Example 4.** We consider the missionaries and cannibals problem (MCP) (Pressman & Singmaster, 1989). It is a well-known toy
problem in artificial intelligence, where it was used by Saul Amarel as an example of problem representation (Wikipedia, 2015). The MCP is briefly stated as follows. Three missionaries and three cannibals must cross a river by using a boat. The boat can carry at most two people. At each of the banks, if there are missionaries present in the bank, their number must be no less than that of cannibals in the same bank. Otherwise, the cannibals would eat the missionaries. The boat cannot cross the river by itself if there is no people on board.

We consider the control problem in the MCP. In each bank, the control strategy becomes:

\[
\begin{align*}
    n_m &\geq n_c & \text{if } n_c > 0 \\
    n_m &\geq 0 & \text{if } n_c = 0
\end{align*}
\]

where \(n_m \in \{0, 1, 2, 3\}\) and \(n_c \in \{0, 1, 2, 3\}\) represent the numbers of missionaries and cannibals in a bank, respectively. The control strategy can be transformed into an additive separable function as follows:

\[
f_1(n_m) + f_2(n_c) \leq 3
\]  

(17)

where \(f_1(n_m)\) is a mapping from integers to integers as shown in Table 3 and \(f_2(n_c) = n_c\).

### Table 3

<table>
<thead>
<tr>
<th>(n_m)</th>
<th>(f_1(n_m))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

We can verify that Eq. (17) can represent the control strategy in the MCP. Eq. (17) is an additive separable constraint that can be enforced by the method proposed in Section 4.1. The original Petri net modeling the MCP has six places and ten transitions, as shown in Fig. 7. Table 4 presents the meanings of transitions in the net model. It has 30 reachable markings in which 18 and 12 are admissible and inadmissible ones, respectively. Note that there are two admissible markings that can be reached by BFMs only, i.e., both are in the FZ. Thus, the two markings should not be reached once all BFMs are forbidden. By using the supervisor construction method proposed in Section 4.1, four transitions in the original net model are replaced by 16 transitions and two control places are added for the control strategies in the two banks. The incidence matrix \([N_c]\) of the controlled net model is shown in Table 5 where \([N_c](p, t) = a - b\) indicates that there is a self-loop between \(p\) and \(t\) with \(W(t, p) = a\) and \(W(p, t) = b\). In the table, \(p_{ij}\) with \(M_0(p_{ij}) = 0\) and \(p_{ij}\) with \(M_0(p_{ij}) = 3\) are used to enforce constraints \(f_1(\mu_1) + f_2(\mu_2) \leq 3\) and \(f_1(\mu_3) + f_2(\mu_4) \leq 3\) for the control purpose in the left and right banks, respectively. Note that we do not design the complementary place of \(p_1\) since \(p_1\) is just the one with \(M(p_1) = 3\) for any reachable marking \(M\). Similarly, \(p_3\) is also the complementary place of \(p_3\). The controlled net model has all the 16 admissible markings and no inadmissible marking is reachable. That is to say, the proposed method can optimally enforce the control strategies in the MCP.

### Table 4

Meanings of transitions in Fig. 7.

<table>
<thead>
<tr>
<th>(t_i)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t_1)</td>
<td>Boat carries a missionary to right bank</td>
</tr>
<tr>
<td>(t_2)</td>
<td>Boat carries a missionary to left bank</td>
</tr>
<tr>
<td>(t_3)</td>
<td>Boat carries two missionaries to right bank</td>
</tr>
<tr>
<td>(t_4)</td>
<td>Boat carries two missionaries to left bank</td>
</tr>
<tr>
<td>(t_5)</td>
<td>Boat carries a missionary and a cannibal to right bank</td>
</tr>
<tr>
<td>(t_6)</td>
<td>Boat carries a missionary and a cannibal to left bank</td>
</tr>
<tr>
<td>(t_7)</td>
<td>Boat carries two cannibals to right bank</td>
</tr>
<tr>
<td>(t_8)</td>
<td>Boat carries two cannibals to left bank</td>
</tr>
<tr>
<td>(t_9)</td>
<td>Boat carries a cannibal to right bank</td>
</tr>
<tr>
<td>(t_{10})</td>
<td>Boat carries a cannibal to left bank</td>
</tr>
</tbody>
</table>

We can verify that Eq. (17) can represent the control strategy in the MCP. Eq. (17) is an additive separable constraint that can be enforced by the method proposed in Section 4.1. The original Petri net modeling the MCP has six places and ten transitions, as shown in Fig. 7. Table 4 presents the meanings of transitions in the net model. It has 30 reachable markings in which 18 and 12 are admissible and inadmissible ones, respectively. Note that there are two admissible markings that can be reached by BFMs only, i.e., both are in the FZ. Thus, the two markings should not be reached once all BFMs are forbidden. By using the supervisor construction method proposed in Section 4.1, four transitions in the original net model are replaced by 16 transitions and two control places are added for the control strategies in the two banks. The incidence matrix \([N_c]\) of the controlled net model is shown in Table 5 where \([N_c](p, t) = a - b\) indicates that there is a self-loop between \(p\) and \(t\) with \(W(t, p) = a\) and \(W(p, t) = b\). In the table, \(p_{ij}\) with \(M_0(p_{ij}) = 0\) and \(p_{ij}\) with \(M_0(p_{ij}) = 3\) are used to enforce constraints \(f_1(\mu_1) + f_2(\mu_2) \leq 3\) and \(f_1(\mu_3) + f_2(\mu_4) \leq 3\) for the control purpose in the left and right banks, respectively. Note that we do not design the complementary place of \(p_1\) since \(p_1\) is just the one with \(M(p_1) = 3\) for any reachable marking \(M\). Similarly, \(p_3\) is also the complementary place of \(p_3\). The controlled net model has all the 16 admissible markings and no inadmissible marking is reachable. That is to say, the proposed method can optimally enforce the control strategies in the MCP.

### Table 5

The incidence matrix \([N_c]\) of the controlled net model for MCP.

<table>
<thead>
<tr>
<th>(p)</th>
<th>(t_1)(^{-2})</th>
<th>(t_1)(^{-1})</th>
<th>(t_1)(^{0})</th>
<th>(t_1)(^{1})</th>
<th>(t_1)(^{2})</th>
<th>(t_1)(^{3})</th>
<th>(t_2)(^{-2})</th>
<th>(t_2)(^{-1})</th>
<th>(t_2)(^{0})</th>
<th>(t_2)(^{1})</th>
<th>(t_2)(^{2})</th>
<th>(t_2)(^{3})</th>
<th>(t_3)</th>
<th>(t_4)</th>
<th>(t_{10})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p_1)</td>
<td>2, -3</td>
<td>1, -2</td>
<td>1, -3</td>
<td>-2</td>
<td>2, -3</td>
<td>1, -2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(p_2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_3)</td>
<td>1</td>
<td>2, 1</td>
<td>3, -2</td>
<td>2</td>
<td>3, -2</td>
<td>1</td>
<td>2, -1</td>
<td>3, -2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(p_4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_7)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_9)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>(p_{10})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 7. The Petri net model of MCP.
5. Conclusions

This paper deals with the enforcement of the nonlinear constraints on bounded Petri nets. A supervisory structure is presented to implement a class of nonlinear constraints, namely additive separable functions. The proposed method can directly design a supervisor given a nonlinear constraint. A number of examples are provided to demonstrate the proposed method. A future topic is to reduce the structural complexity of the proposed supervisors. Another future work is to extend this work to design Petri net supervisors to enforce nonlinear constraints for net models with uncontrollable transitions (Luo, Shao, Nonami, & Jin, 2012; Ma, Li, & Giua, 2015; Moody & Antsaklis, 2000).

References

Yufeng Chen received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2006 and 2011, respectively. He joined Xidian University, in 2011, and is currently an Associate Professor with the School of Electro-Mechanical Engineering. He is the author or coauthor of over 20 publications. He is a coauthor with ZhiWu Li of Optimal Supervisory Control of Automated Manufacturing Systems (CRC Press, Taylor & Francis Group, 2013). His research interests include Petri net theory and applications, supervisory control of discrete event systems, and scheduling flexible manufacturing systems.
Yuewu Li received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, all from Xidian University, Xi’an, China, in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992 and now he is also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau. Over the past decade, he was a Visiting Professor at the University of Toronto, Technion-Israel Institute of Technology, Martin-Luther University of Halle-Wittenberg, Conservatoire National des Arts et Metiers (CNAM), Melissa University. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining. He is a member of Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society, and a member of IFAC Technical Committee on Discrete Event and Hybrid Systems from 2011 to 2014. He serves as a frequent reviewer for 40+ international journals including Automatica and a number of the IEEE Transactions on Control Systems Technology, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Robotics and Automation, and IEEE Transactions on Control Systems Technology, 23(1), 323–330.
Zhili Wu (M’06–SM’07) received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, all from Xidian University, Xi’an, China, in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992 and now he is also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau. Over the past decade, he was a Visiting Professor at the University of Toronto, Technion-Israel Institute of Technology, Martin-Luther University of Halle-Wittenberg, Conservatoire National des Arts et Metiers (CNAM), Melissa University. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining. He is a member of Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society, and a member of IFAC Technical Committee on Discrete Event and Hybrid Systems from 2011 to 2014. He serves as a frequent reviewer for 40+ international journals including Automatica and a member of the IEEE Transactions on Control Systems Technology, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Robotics and Automation, and IEEE Transactions on Control Systems Technology, 23(1), 323–330.
Kamel Barkaoui is a Professor of Computer Science at the Conservatoire National des Arts et Métiers (CNAM—Paris). He obtained a Ph.D. (1988) and Habilitation à Diriger des Recherches (1998) in Computer Science from Université Paris 6 (UPMC). His research interests include formal methods for verification, control and performance evaluation of concurrent and distributed systems. He has published over 90 refereed international journal articles and conference papers and received the 1995 IEEE International Conference on System Man and Cybernetics Outstanding Paper Award. He has served on PCs and as PC Chair and OC Chair for numerous international workshops and conferences. He was a guest editor of Journal of Systems and Software (JSS) and International Journal of Critical Computer-Based Systems (IJCCBS).

Alessandro Giua is a Professor of Automatic Control at the Department of Electrical and Electronic Engineering (DIEE) of the University of Cagliari, Italy and at the Information and Systems Sciences Laboratory (LSIS) of Aix-Marseille University, France. He received a Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1992. He has also held visiting positions in several institutions worldwide, including Xi’an University, Xi’an, China. His research interests include discrete event systems, hybrid systems, networked control systems, Petri nets and failure diagnosis. On these topics he has published extensively and has managed several research projects. He is the Editor in Chief of the IFAC journal Nonlinear Analysis: Hybrid Systems, and a Department Editor of Discrete Event Dynamic Systems. He has previously served in the editorial board of several journals including: IEEE Trans. on Automatic Control, IEEE Trans. on Control Systems Technology and European Journal of Control. He has been General Chair or Program Chair for international conferences such as WODES 1998, IFAC ADHS 2006 and IEEE CASE 2011 and has served in the steering committees of these series. He has also been Chair of the IFAC Technical Committee 1.3 on Discrete Event and Hybrid Systems from 2008 to 2014 and has served as a member of the program committee for several international events in this area. Currently he is serving the IEEE Control Systems Society as a member of the Board of Governors (2013–2015) and is working at the organization of the conferences IEEE CDC 2016 as General Chair and WODES 2016 as a Program Chair.