Designs of Bisimilar Petri Net Controllers With Fault Tolerance Capabilities

Lingxi Li, *Student Member, IEEE*, Christoforos N. Hadjicostis, *Senior Member, IEEE*, and Ramavarapu S. Sreenivas, *Senior Member, IEEE*

***Abstract*—This paper proposes an approach for providing tolerance against faults that may compromise the functionality of a given controller modeled by a Petri net. The method is based on embedding the given Petri net controller into a larger (redundant) Petri net controller that retains the original functionality and properties, and uses additional places, connections, and tokens to impose invariant conditions that allow the systematic detection and identification of faults via *linear parity checks*. In particular, this paper considers two types of redundant Petri net controllers: 1) *nonseparate* redundant Petri net controllers have the same functionality as the given Petri net controller and allow for fault detection and identification, but do not necessarily retain the given controller intact; and 2) *separate* redundant Petri net controllers are a special case of the nonseparate redundant controllers that retain the given Petri net controller intact but enhance it with additional places to enable fault detection and identification. The work in this paper obtains complete characterizations of both types of redundant controllers along with necessary and sufficient conditions for them to be *bisimulation equivalent* to the given original Petri net controller. In addition, this paper discusses how each type of redundant controllers can be designed to have desirable fault detection and identification capabilities. When the bisimulation equivalence requirement is not directly enforced, nonseparate redundant controllers can potentially have advantages over separate ones (e.g., they can use fewer connections to detect and identify the same number of faults). An example of a Petri net controller for a production cell and its fault tolerance capabilities using separate and nonseparate embeddings is used to illustrate the approach.**

***Index Terms*—Bisimulation equivalence, fault tolerance, Petri nets, redundant Petri net controllers.**

I. Introduction

FAULTS in large-scale dynamic systems can compromise their functionality in complex ways and, depending on the underlying application, can have devastating consequences or even lead to loss of life. For this reason, fault tolerance has received considerable attention in the areas of control, communication, and industrial systems. Previous research has extensively studied fault-tolerant implementations for discrete-time dynamic systems by constructing encoded embeddings of them. For instance, the work in [1]–[3] studied linear finite-state machines, and the work in [4]–[7] generalized this approach to arbitrary finite-state machines. A similar approach that is also based on encoding the state of the system was used in [8]–[10] to achieve fault detection and identification in discrete-event systems that can be modeled by Petri nets. This approach encodes the state of a given Petri net into a larger redundant one in a way that preserves the state and evolution of the original Petri net, while enabling an external mechanism to detect, identify, and correct faults that may corrupt the state of the redundant system. The construction of the redundant Petri net incorporates additional places and connections associated with transitions in the original system; faults in the Petri net are then detected and identified via linear parity checks on the overall *encoded* state of the redundant Petri net.

This paper considers a setting similar to [10] and aims at providing fault tolerance to certain plant controllers that can be modeled by Petri nets. The representation of plant/controller systems as Petri nets can take advantage of the modeling capabilities of Petri nets, particularly in hybrid and supervisory controllers. The approach in this paper extends the techniques in [10] to settings where the redundant and original Petri net controllers need to be bisimulation equivalent; it can be applied to any controller that can be modeled as a Petri net, such as the ones developed in [11]–[17] (a good survey can be found in [18]). Note that in deriving necessary and sufficient conditions for bisimulation equivalence, the linear algebraic tools used here are reminiscent of the techniques used in the context of Petri net structural analysis (see, for instance, [19] and references therein).

This paper considers two types of *redundant Petri net controllers*, both of which allow systematic detection and identification of faults (malfunctions) that might take place during the controller operation.

1) The first type of redundant Petri net controllers studied here uses a nonseparate redundant implementation that retains the behavior of the given controller (in terms of the transitions that are enabled at any given time) while incorporating enough redundancy to enable the detection and identification of faults. The term *nonseparate* indicates that the given controller is not immediately identifiable within the redundant controller net. In other words, one cannot directly locate a subset of places and connecting arcs whose marking and weights are

Manuscript received December 22, 2005; revised April 2, 2006 and August 15, 2006. The work in this paper was supported in part by the National Science Foundation under NSF Career Award 0092696, NSF ITR Award 0085917, NSF EPNES Award 0224729, and NSF CNS Award 0437415, and by the Air Force Office of Scientific Research (AFOSR) under URI Award F49620-01-1-0365URI. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF or AFOSR. This paper was recommended by Associate Editor M. P. Fanti.

L. Li and C. N. Hadjicostis are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801-2307 USA, and also with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: lingxili@uiuc.edu; chadjic@uiuc.edu).

R. S. Sreenivas is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801-2307 USA, and also with the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

Digital Object Identifier 10.1109/TSMCA.2007.909559

identical to the marking and weights of the corresponding places and arcs in the original controller. Although the nonseparate redundant Petri net controller does not necessarily keep the given controller intact, it allows one to recover the state and functionality of the given controller and provides certain flexibility in the controller design.

2) The second type of redundant controllers is based on embedding the given Petri net controller into a separate redundant controller, i.e., a larger controller that keeps the given controller intact, but strategically incorporates additional places and connections in a way that enables the systematic detection and identification of faults in the redundant controller (the separate redundant controller keeps the original controller intact because it contains a set of places and connections with marking and arc weights that are identical to the marking and arc weights of the corresponding places and connections in the original controller). The additional (redundant) places, connections, and tokens are used to impose invariant conditions that serve as consistency checks; as a result, by performing linear parity checks on the combined marking of the given controller [including the original controller places and the additional (redundant) places], this methodology is able to systematically detect and identify faults.

This paper also discusses conditions under which both types of redundant controllers are *bisimulation equivalent* to the given controller (i.e., any transition sequence enabled in the given controller is also enabled in the redundant one, and vice versa). Under the reasonable assumption that the given Petri net controller is presumably designed to exactly meet the control specification of the plant to be controlled, bisimulation equivalence is clearly a property that needs to be imposed on the redundant controllers (so that the desired control functionality is retained by the redundant controllers). It is shown in this paper that under the conditions of bisimulation equivalence, both types of redundant controllers have the same fault detection and identification capabilities (in terms of the number of connections needed to detect and identify a given number of faults). However, if one is only using the redundant controller as a monitor for providing fault tolerance (i.e., if the bisimulation equivalence requirement is not directly enforced), nonseparate redundant controllers provide more design flexibility and can have advantages over separate ones (e.g., they can use fewer connections to detect and identify the same number of faults).

The remainder of this paper is organized as follows. Section II provides a quick review of Petri nets and discusses some related work on bisimulation equivalence in Petri nets. Section III presents the fault models in the setup and provides an illustrative example of a production cell. Section IV describes the construction of nonseparate redundant Petri net controllers, along with the necessary and sufficient conditions for them to be bisimulation equivalent to the given controller. Section V briefly describes the construction of separate redundant Petri net controllers and the conditions for them to achieve bisimulation equivalence. Examples are provided to illustrate fault detection and identification schemes using both types of redundant controllers in Section VI. Section VII demonstrates (through analysis and examples) some of the potential advantages of nonseparate redundant controllers over separate ones. Section VIII presents the conclusions and future research directions.

II. Notations and Preliminaries

This section provides some basic definitions and terminology that will be used throughout this paper. More details about Petri nets can be found in [20]–[22].

A. Petri Nets

A Petri net structure is a weighted bipartite graph $N = (P, T, A, \omega)$, where $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of n *places* (drawn as circles), $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of m *transitions* (drawn as rectangles), $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (from places to transitions and from transitions to places), and $\omega : A \to \{1, 2, 3, \ldots\}$ is the *weight function* on the arcs.

At each time epoch t, a *marking* is a vector $q[t] : P \to \{0, 1, 2, \ldots\}^n$ that assigns to each place of the Petri net a nonnegative integer number of tokens (drawn as black dots). The initial marking $q[0]$ denotes the number of tokens in each place of the Petri net at the initial time epoch $t = 0$. A Petri net with a given initial marking is denoted by $PN = (N, q[0])$.

Let b_{ij}^- denote the integer weight of the arc from place p_i to transition t_j, and b_{ij}^+ denote the integer weight of the arc from transition t_j to place p_i $(1 \le i \le n, 1 \le j \le m)$. Note that $b_{ij}^-(b_{ij}^+)$ is taken to be zero if there is no arc from place p_i to transition t_j (or vice versa). Let the *input incident matrix* $B^- = [b_{ij}^-]$ (*output incident matrix* $B^+ = [b_{ij}^+]$) be the $n \times m$ matrix with b_{ij}^- (b_{ij}^+) at its ith row, jth column position. The state (or marking) evolution of Petri net N is given by

$$q[t+1] = q[t] + (B^+ - B^-)x[t] \equiv q[t] + Bx[t] \qquad (1)$$

where the *incident matrix* is defined to be $B \equiv B^+ - B^-$, $q[t]$ is the marking of the Petri net at time epoch t, and the firing vector $x[t]$ is restricted to have exactly one nonzero entry with value "1" (when the jth entry is "1," transition t_j fires at time epoch t). Note that this setting implies that the Petri nets considered in this paper are asynchronous (i.e., at each time epoch, only one transition may fire).

Note that transition t_j is *enabled* at time epoch t if and only if $q[t] \ge B^-(:, j)$, where the inequality is taken elementwise, and $B^-(:, j)$ denotes the jth column of B^- (this ensures that the marking $q[t]$ remains nonnegative at all time epochs). If transition t_j is enabled at marking $q[t]$, it may *fire* and yield the marking $q[t+1] = q[t] + Bx_j[t]$, where $x_j[t]$ indicates that the jth entry of $x[t]$ is nonzero with value "1." The firing of t_j is denoted by $q[t] \xrightarrow{t_j} q[t+1]$, and the marking $q[t+1]$ is said to be *reachable* from the marking $q[t]$.

A *labeling function* $\sigma : T \to \Sigma$ assigns to each transition in the net a label from a given alphabet Σ.

Definition 1: A *labeled* Petri net is a tuple $LN = (P, T, A, \omega, \Sigma, \sigma, q[0], Q_m)$, where (P, T, A, ω) is a Petri net structure, Σ is an alphabet of symbols, $\sigma : T \to \Sigma$ is the transition labeling function, $q[0]$ is the initial marking, and Q_m is the set of final markings [21].

Remark 1: Unlike [21], this paper does not specify the set of final markings of the net, i.e., it considers *P-type* Petri net languages whose final marking set includes all reachable markings from the initial marking $q[0]$ [22].

Definition 2: A *free-labeled* Petri net is a labeled Petri net where all transitions have a distinct label, i.e., if $\sigma(t_i) = \sigma(t_j)$, then $t_i = t_j$ [22].

B. Bisimulation Equivalence

Many notions of bisimulation equivalence for Petri nets have been proposed; most of these notions are based on the markings of two nets (e.g., [23]). Olderog [24] proposed another notion of bisimulation equivalence called *place bisimulation*, which is based on the places of two nets rather than their markings. In this paper, the former concept of bisimulation equivalence is considered.

The general notion of bisimulation equivalence (also referred to as *bisimilarity*) and its decidability has been extensively studied in [25]–[30]. Jancar [29] showed that bisimilarity checking for labeled Petri nets is undecidable. However, for the free-labeled Petri nets studied in this paper, bisimilarity coincides with language equivalence, which is known to be reducible to the reachability problem and decidable [31]. The definition of bisimulation equivalence between free-labeled Petri nets is discussed below (in what follows, $\mathcal{N} = \{0, 1, 2, \ldots\}$ denotes the set of nonnegative integer numbers).

Let $PN_1 = (N_1, q_1[0]) = (P_1, T_1, A_1, \omega_1, q_1[0])$ and $PN_2 = (N_2, q_2[0]) = (P_2, T_2, A_2, \omega_2, q_2[0])$ be two free-labeled Petri nets, and let $\mathcal{R} \subseteq \mathcal{N}^{P_1} \times \mathcal{N}^{P_2}$ be a relation between their markings $q_1[t]$ and $q_2[t']$.

Definition 3: $\mathcal{R}$ is a *bisimulation*[1] if and only if for all $q_1[t]$, $q_2[t']$ such that $q_1[t]\mathcal{R}q_2[t']$.

1) For each enabled transition $t_i \in T_1$ such that $q_1[t] \xrightarrow{t_i} q_1[t+1]$, there exists an enabled transition $t_j \in T_2$ such that $q_2[t'] \xrightarrow{t_j} q_2[t'+1]$ with $q_1[t+1]\mathcal{R}q_2[t'+1]$.
2) For each enabled transition $t_j \in T_2$ such that $q_2[t'] \xrightarrow{t_j} q_2[t'+1]$, there exists an enabled transition $t_i \in T_1$ such that $q_1[t] \xrightarrow{t_i} q_1[t+1]$ with $q_1[t+1]\mathcal{R}q_2[t'+1]$.

When $q_1[t]\mathcal{R}q_2[t']$ for a bisimulation $\mathcal{R}$, Petri nets $(N_1, q_1[t])$ and $(N_2, q_2[t'])$ are called *bisimilar* and are denoted by $(N_1, q_1[t]) \sim (N_2, q_2[t'])$.

Example 1: Assume that two free-labeled nets $PN = (P, T, A, \omega, q_1)$ and $PN' = (P', T', A', \omega', q_1')$ have the reachability graphs shown in Fig. 1, where q_1 denotes the initial marking of net PN, and q_1' denotes the initial marking of net PN'. Consider a relation $\mathcal{R}$ between the markings of two nets as $\mathcal{R} \subseteq \mathcal{N}^P \times \mathcal{N}^{P'} = \{(q_1, q_1'), (q_2, q_2'), (q_2, q_2''), (q_3, q_3'), (q_3, q_4'), (q_4, q_3'), (q_4, q_4')\}$. It is not hard to show that $\mathcal{R}$ satisfies the two conditions identified in Definition 3 and is, therefore, a bisimulation. Thus, the two Petri nets PN and PN' are bisimilar.[2]

[1]The notion of bisimulation equivalence used in this paper is also called strong bisimulation equivalence in some other references [30].

[2]For more details about bisimulation equivalence between two Petri nets and examples or applications, the interested reader is referred to [23], [26], and [29].

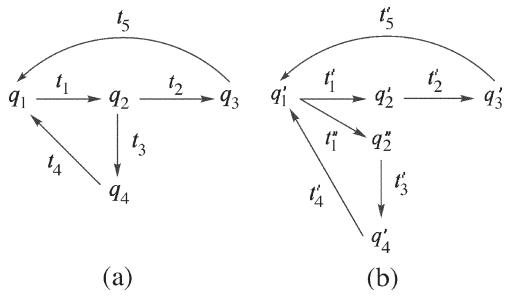

Fig. 1. Reachability graphs of two bisimilar free-labeled Petri nets.

Remark 2: From Example 1, it is clear that Definition 3 requires that a bisimulation exists between the markings of two free-labeled Petri nets, but does not impose any direct conditions on transitions (i.e., possibly different transition sequences are allowed as long as their firings result in the bisimulation between the markings of the two nets). However, the definition of bisimulation in this paper requires that the transitions in the two Petri nets are in one-to-one correspondence, i.e., at any time epoch t, a transition that is enabled in one net must be also enabled in the other net. In addition, the relations $\mathcal{R}$ require that the markings of the two nets are related by a linear transformation (i.e., there exist matrices L and G of appropriate dimensions such that $q_1[t] = Lq_2[t]$ and $q_2[t] = Gq_1[t]$). This becomes clearer in the definition that follows.

Let $\mathcal{C} = (P_c, T, A_c, \omega_c, q_c[0])$ denote the net of the given controller and $\mathcal{H} = (P_h, T, A_h, \omega_h, q_h[0])$ be the net of a corresponding redundant controller.

Definition 4: $(\mathcal{C}, q_c[t]) \sim (\mathcal{H}, q_h[t])$ if and only if for all $q_c[t] \in \mathcal{C}$ and all $q_h[t] \in \mathcal{H}$ such that $q_c[t] = Lq_h[t]$ and $q_h[t] = Gq_c[t]$, the following are true.

C1) If $q_c[t] \xrightarrow{t_j} q_c[t+1]$, then $q_h[t] \xrightarrow{t_j} q_h[t+1]$ such that $q_c[t+1] = Lq_h[t+1]$ and $q_h[t+1] = Gq_c[t+1]$.
C2) If $q_h[t] \xrightarrow{t_j} q_h[t+1]$, then $q_c[t] \xrightarrow{t_j} q_c[t+1]$ such that $q_c[t+1] = Lq_h[t+1]$ and $q_h[t+1] = Gq_c[t+1]$.

Remark 3: Note that a redundant controller net $\mathcal{H}$ has the same transitions as the given controller net $\mathcal{C}$, i.e., it does not include any extra transitions. Thus, the definition of bisimulation in this paper requires that, at any time epoch t, any transition enabled in the given controller can be also enabled in the redundant one (condition C1) and vice versa (condition C2).

C. Matrix and Vector Inequalities

The following notation is used throughout this paper.

Let $\mathcal{Q}$ be the set of rational numbers. Given matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ in $\mathcal{Q}^{n \times m}$, A (B) is said to be *nonnegative* if $A \geq 0$ ($B \geq 0$), i.e., if $a_{ij} \geq 0$ ($b_{ij} \geq 0$) for every $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$. Define $A \geq B$ if $a_{ij} \geq b_{ij}$ for every $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$; similarly, let $A \ngeq B$ if at least one entry of A is smaller than the corresponding entry of B (i.e., $a_{ij} < b_{ij}$ for some $i \in \{1, 2, \ldots, n\}$ and some $j \in \{1, 2, \ldots, m\}$). $A \leq B$ and $A \nleq B$ are defined in a similar way. Also, a column vector $x = [x_i]$ in $\mathcal{Q}^n$ is said to be *nonnegative* (denoted by $x \geq 0$) if $x_i \geq 0$ for every $1 \leq i \leq n$.

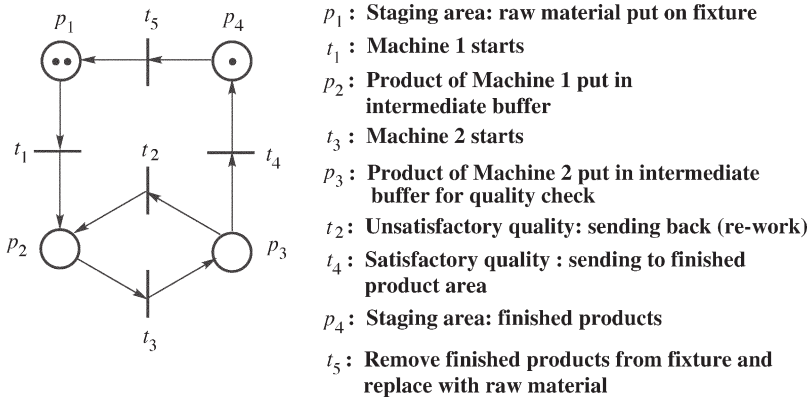

Fig. 2. Petri net model of a production cell.

III. Fault Models

Clearly, the effectiveness of a fault model for a given system significantly depends on the particular application. This paper considers a rather general *place fault* model, which captures faults that cause the corruption of the number of tokens in a particular place of the Petri net controller (by increasing or decreasing the number of tokens in that place by an integer amount). These faults can be caused by sensor failures or internal hardware faults in the controller. For instance, a sensor may indicate that the number of products on a conveyor belt of a manufacturing system is zero, whereas, in reality, there are products on the belt (or vice versa). In this example, this type of sensor faults could arise for a brief duration due to transient occlusions/reflections in the path of optical sensors in nonsterile manufacturing environments and could result in an erroneous interpretation of the system state in the controller. Apart from sensor faults, the fault model of this paper is also motivated by internal hardware faults that occur due to various types of interference in the digital implementation of the Petri net controllers. These types of faults result in bit corruption ("0" becomes "1" or vice versa) in the controller implementation and have been the topic of extensive studies in fault-tolerant computing systems [32]–[34]. Note that sensor faults that result in noisy measurements (as considered, for instance, in [35]) are not a good match for the type of faults considered here, but could be handled by embedding that (continuous-like) fault model within the (discrete-like) fault model of this paper.

A place fault at time epoch t results in an erroneous state (marking) $q_f[t]$ that can be expressed as

$$q_f[t] = q[t] + e_{p_i} \tag{2}$$

where $q[t]$ is the state that would have been reached under fault-free conditions, and e_{p_i} is the place error vector with a single nonzero element at its ith entry. If the ith entry of e_{p_i} is negative, then the number of tokens in the ith place has decreased due to the fault; if it is positive, then the number of tokens in the ith place has increased.

Note that other types of faults (e.g., the transition faults in [8]–[10]) can be seen as a combination of multiple place faults. This paper focuses on the detection and identification of place faults; however, interested readers could refer to [9] and [10] for more details about the detection and identification of transition faults.

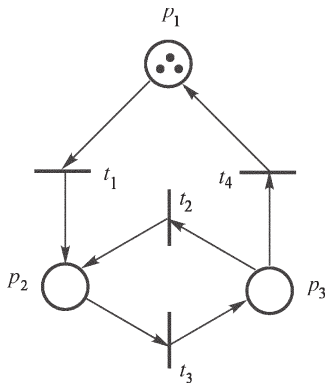

Fig. 3. Simplified Petri net model.

Example 2: Consider a production cell in a manufacturing system, which processes raw material via two machines (machines 1 and 2) as captured by the net on the left of Fig. 2 (a detailed description of each place/transition activity in the Petri net is provided to the right of Fig. 2).

For simplicity, the running Petri net example in this paper starts with a plant that uses only one place p_1 to represent the staging area; this results in the simplified Petri net plant of Fig. 3.

Suppose that the storage limit on the intermediate buffers (i.e., the number of tokens in places p_2 and p_3) is 2 and 1, respectively. One possible Petri net controller (which can be designed using the techniques in [15]) to enforce these constraints is given in Fig. 4: it has two places, i.e., p_{c1} and p_{c2}, and its arcs are drawn with dashed lines.

Note that the Petri net controller in Fig. 4 will be implemented using digital hardware (e.g., a microcontroller or a programmable logic controller). In such cases, a sensor failure or a bit flip in the digital hardware could trigger the place faults considered in this paper. For instance, the corruption of the token number in controller place p_{c1} captures scenarios where the controller is under the impression that more/less products are present in a fixture or a buffer (different control actions to force the storage constraints can lead to different token numbers in controller places) or scenarios where the bit corruption occurs in the digital hardware that implements the controller (which directly causes the corruption of the number of tokens in the controller places). Note that place faults could be also modeled by adding two unobservable fault transitions to

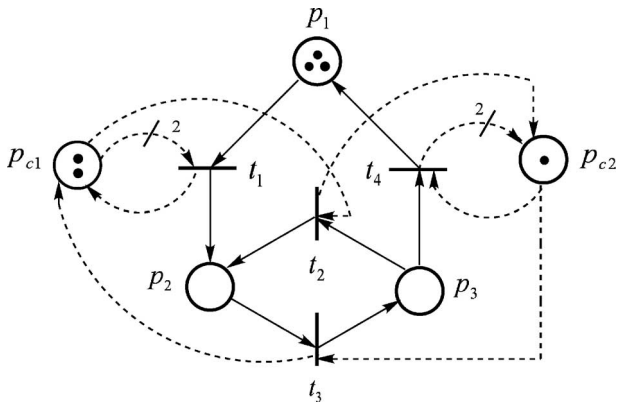

Fig. 4. Petri net controller enforcing storage constraints on the plant of Fig. 3.

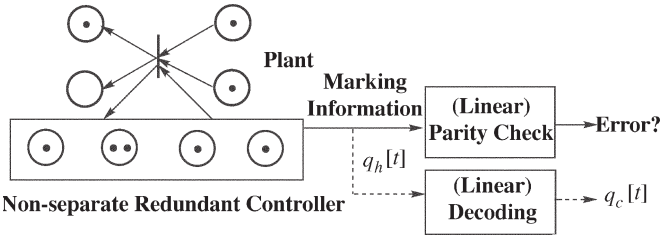

Fig. 5. Nonseparate redundant Petri net controller.

each controller place (p_{c1} and p_{c2}): one transition would have no input places, and the controller place would serve as its sole output place (to model the addition of tokens in that place), whereas the other transition would have no output places and would have the controller place as its sole input place (to model the subtraction of tokens from that place). To avoid cluttering the diagram, these four unobservable fault transitions are not drawn in Fig. 4.

IV. Nonseparate Redundant Petri Net Controllers

A. Designs of Nonseparate Redundant Petri Net Controllers

This section proposes systematic constructions for the class of nonseparate redundant controllers illustrated in Fig. 5. The state $q_h[t]$ of the nonseparate redundant Petri net controller implementation is encoded in a form that allows one to recover the state $q_c[t]$ of the given Petri net controller, if desired, and also perform fault detection and identification. However, the given (original) Petri net controller is not necessarily kept intact.

Let $\mathcal{C}$ be a Petri net controller with n_c places, m transitions, and state evolution

$$q_c[t+1] = q_c[t] + \left(B_c^+ - B_c^-\right) x[t] \equiv q_c[t] + B_c x[t] \quad (3)$$

where $x[t]$ is the firing vector (which indicates the transition that fires in the Petri net controller at time epoch t), B_c^+ is the output incident matrix, B_c^- is the input incident matrix, and $B_c \equiv B_c^+ - B_c^-$ is the incident matrix of the given Petri net controller. Let $q_c[0] \geq 0$ be *any* initial marking and $\mathcal{X} = \{x[0], x[1], \ldots\}$ be *any* admissible (legal) firing sequence under this initial marking.

Consider a Petri net $\mathcal{H}$ with $\eta = n_c + d$ $(d > 0)$ places, m transitions, and state evolution given by

$$q_h[t+1] = q_h[t] + \left(\mathcal{B}_c^+ - \mathcal{B}_c^-\right) x[t] \equiv q_h[t] + \mathcal{B}_c x[t] \quad (4)$$

where $\mathcal{B}_c^+$, $\mathcal{B}_c^-$, and $\mathcal{B}_c \equiv \mathcal{B}_c^+ - \mathcal{B}_c^-$ are the incident matrices of $\mathcal{H}$.

Definition 5: $\mathcal{H}$ is a *nonseparate redundant controller* of controller $\mathcal{C}$ if $(\mathcal{C}, q_c[0]) \sim (\mathcal{H}, Gq_c[0])$ for all $q_c[0]$.

Lemma 1: $\mathcal{H}$ is a nonseparate redundant controller of controller $\mathcal{C}$ if there exist: 1) an $n_c \times \eta$ decoding matrix L; 2) an $\eta \times n_c$ full-column rank encoding matrix G with nonnegative integer entries; and 3) an $\eta \times m$ matrix $\mathcal{D}$ with nonnegative integer entries such that $\mathcal{D} \leq \min(GB_c^+, GB_c^-)$, so that

1) $q_c[t] = Lq_h[t]$ and $q_h[t] = Gq_c[t]$ for all time epochs $t \geq 0$;
2) matrices $\mathcal{B}_c^+$ and $\mathcal{B}_c^-$ can be written as

$$\mathcal{B}_c^+ = GB_c^+ - \mathcal{D}$$
$$\mathcal{B}_c^- = GB_c^- - \mathcal{D}.$$

Definition 4 requires linear encoding and decoding; Lemma 1 further argues that the encoding matrix G has to have nonnegative integer entries and allow a parametrization of $\mathcal{B}_c^+$ and $\mathcal{B}_c^-$ in terms of matrices B_c^+, B_c^-, and $\mathcal{D}$. The proof of Lemma 1 can be found in [34], which shows that, with the above choices, one is guaranteed that under any initial condition of the controller net $\mathcal{C}$, a firing sequence $\mathcal{X}$ that is admissible in $\mathcal{C}$ is also admissible in $\mathcal{H}$.

Thus, a nonseparate redundant Petri net controller $\mathcal{H}$ that satisfies Definition 5 will admit any transition that is allowed to fire in $\mathcal{C}$ (i.e., condition C1 in Definition 4 always holds). To achieve bisimulation equivalence in the nonseparate case, however, it is necessary to ensure that any transition enabled by $\mathcal{H}$ is also enabled by $\mathcal{C}$ (i.e., condition C2 in Definition 4 holds). If transition t_j is enabled in $\mathcal{H}$ at time epoch t, it follows that

$$q_h[t] \geq \left(GB_c^- - \mathcal{D}\right) x_j[t].$$

Clearly, this implies that

$$Gq_c[t] \geq \left(GB_c^- - \mathcal{D}\right) x_j[t]$$
$$G\left(q_c[t] - B_c^- x_j[t]\right) \geq -\mathcal{D}x_j[t].$$

Given the inequality above, the problem reduces to finding (additional) conditions on matrices G and $\mathcal{D}$ such that for each j, $1 \leq j \leq m$, the column vector $q_c[t] - B_c^- x_j[t]$ has nonnegative integer entries (which implies that transition t_j can fire in the given Petri net controller). To find these conditions on G and $\mathcal{D}$, some notation is introduced first.

Given a column vector $z \in \mathcal{Q}^{n_c}$, let the symbol z_i $(i \in \{1, 2, \ldots, n_c\})$ denote the ith component of z.

Definition 6: The *support* of z is the set of nonzero indexes of z and is denoted by $\|z\|$ $(\subseteq \{1, 2, \ldots, n_c\})$, i.e.,

$$\|z\| = \{i | z_i \neq 0\}.$$

Suppose that $G = [g_1\, g_2\, \ldots\, g_{n_c}]$ is an $\eta \times n_c$ matrix with nonnegative entries, where each column $g_i \in \mathcal{Q}^\eta$, $i \in \{1, 2, \ldots, n_c\}$.

Definition 7: G has *independent column support* if[3]

$$\forall i \in \{1,2,\ldots,n_c\}, \left(\|g_i\| - \bigcup_{j \in \{\{1,2,\ldots,n_c\}-i\}} \|g_j\| \right) \neq \emptyset$$

where $\emptyset$ denotes the empty set.

Definition 8: Let g_{ki} denote the element at the kth row, ith column position of matrix G; column g_i is said to have *unique support* at its kth row if

$$\left(\|g_i\| - \bigcup_{j \in \{\{1,2,\ldots,n_c\}-i\}} \|g_j\| \right) = \{k\}.$$

For example, I_n, the $n \times n$ identity matrix, has independent column support, and each column I_i, $i \in \{1,2,\ldots,n\}$, has unique support at its ith row. Likewise, the matrix $\begin{bmatrix} I_n \\ C \end{bmatrix}$, where C is an arbitrary matrix of n columns, has independent column support.

Remark 4: Independent column support of a matrix is not an invariant property. A set of vectors that form a matrix with independent column support in one reference frame will not define a matrix with independent column support when viewed in a different reference frame.

Lemma 2: If matrix $G = [g_1\, g_2\, \ldots\, g_{n_c}] \in \mathcal{Q}^{\eta \times n_c}$ has independent column support, matrix G has full-column rank.

Proof: If matrix G has independent column support, from Definition 7, it follows that

$$\forall i \in \{1,2,\ldots,n_c\}, \left(\|g_i\| - \bigcup_{j \in \{\{1,2,\ldots,n_c\}-i\}} \|g_j\| \right) \neq \emptyset.$$

Equivalently, each column g_i has at least one nonzero entry, for example, at its k_ith row for some $k_i \in \{1,2,\ldots,\eta\}$, where other columns have entries of zero. For $a_1, a_2, \ldots, a_{n_c} \in \mathcal{Q}$

$$\sum_{i=1}^{n_c} a_i g_i = 0$$

if and only if $a_1 = a_2 = \cdots = a_{n_c} = 0$ (because if $a_i \neq 0$, the k_ith entry of $\sum_{i=1}^{n_c} a_i g_i$ is nonzero). Therefore, the columns of matrix G are linearly independent, and the result follows. ■

Theorem 1: Let $G = [g_1\, g_2\, \ldots\, g_{n_c}] \in \mathcal{Q}^{\eta \times n_c}$ be a matrix with nonnegative entries. Then, $(\forall z \in \mathcal{Q}^{n_c}, (Gz \geq 0 \Rightarrow z \geq 0)) \Leftrightarrow G$ has independent column support.

Proof: ($\Leftarrow$)*:* If G has independent column support, then there can be no $z \in \mathcal{Q}^{n_c}$, where some entry of z is negative (for example, the ith component $z_i < 0$), yet $Gz \geq 0$. The proof is by contradiction: Suppose z has a negative entry at its ith position ($z_i < 0$). Then, $g_i z_i < 0$ since G has nonnegative entries, which, in turn, would mean that

$$\forall i \in \left\{ \|g_i\| - \bigcup_{j \in \{\{1,2,\ldots,n_c\}-i\}} \|g_j\| \right\},\ (Gz)_i < 0.$$

This contradicts $Gz \geq 0$, and the result follows.

($\Rightarrow$)The proof is by showing the contrapositive, i.e., if G does not have independent column support, then there exists a $z \in \mathcal{Q}^{n_c}$ with negative entries and $Gz \geq 0$. If G does not have independent column support, then $\exists i \in \{1,2,\ldots,n_c\}$ such that

$$\|g_i\| - \bigcup_{j \in \{\{1,2,\ldots,n_c\}-i\}} \|g_j\| = \emptyset.$$

Let the ith element of $z \in \mathcal{Q}^{n_c}$ be negative, i.e., choose $z_i = -\epsilon$ (where ϵ is a small positive rational number) and $z_j \geq 0$ (where $j \in \{1,2,\ldots,n_c\}$ and $j \neq i$). Let g_{ki} denote the element at the kth row, ith column position in matrix G. To let $Gz \geq 0$, it requires that

$$g_{ki} \times (-\epsilon) + \left(\sum_{j \in \{\{1,2,\ldots,n_c\}-i\}} g_{kj} \times z_j \right) \geq 0$$

where $i \in \{1,2,\ldots,n_c\}$ and $k \in \{1,2,\ldots,\eta\}$.

For entries where $g_{ki} = 0$, the inequality above holds since g_{kj} and z_j are nonnegative. For entries where $g_{ki} \neq 0$, if ϵ is chosen so that

$$0 < \epsilon \leq \frac{\sum_{j \in \{\{1,2,\ldots,n_c\}-i\}} g_{kj} \times z_j}{g_{ki}}$$

then the inequality will be satisfied. Therefore, to guarantee that the inequality above holds for all $k \in \{1,2,\ldots,\eta\}$, ϵ must be chosen to satisfy

$$0 < \epsilon \leq \min_k \left\{ \frac{\sum_{j \in \{\{1,2,\ldots,n_c\}-i\}} g_{kj} \times z_j}{g_{ki}} \right\}.$$

Clearly, if G does not have independent column support, one can always choose an ϵ such that there exists a $z \in \mathcal{Q}^{n_c}$ with negative entries such that $Gz \geq 0$, and the result follows. ■

Corollary 1: Let $G = [g_1\, g_2\, \ldots\, g_{n_c}] \in \mathcal{N}^{\eta \times n_c}$ be a matrix with nonnegative integer entries. Then, $(\forall z \in \mathcal{N}^{n_c}, (Gz \geq 0 \Rightarrow z \geq 0)) \Leftrightarrow G$ has independent column support.

Proof: The proof is similar as Theorem 1; however, the entries of z_j, $j \neq i$, in $z \in \mathcal{Q}^{n_c}$ need to be nonnegative integer entries. We can simply scale the entries of z such that within the range of

$$0 < \epsilon \leq \min_k \left\{ \frac{\sum_{j \in \{\{1,2,\ldots,n_c\}-i\}} g_{kj} \times z_j}{g_{ki}} \right\}$$

where $k \in \{1,2,\ldots,\eta\}$, one can pick up an ϵ with a positive integer entry. ■

The following theorem states that by appropriately constructing matrices G and $\mathcal{D}$, the nonseparate redundant controller $\mathcal{H}$ is guaranteed to be bisimulation equivalent to the given controller $\mathcal{C}$.

Theorem 2: Let $\mathcal{H}$ be a nonseparate redundant Petri net controller implementation of $\mathcal{C}$, as in Definition 5. Then, $(\mathcal{H}, q_h[t]) \sim (\mathcal{C}, q_c[t])$ if and only if $G \in \mathcal{N}^{\eta \times n_c}$ is a matrix with nonnegative integer entries and has independent column support, and $\mathcal{D} \in \mathcal{N}^{\eta \times m}$ is a matrix with nonnegative integer entries that satisfies $0 \leq \mathcal{D} \leq (GB_c^+, GB_c^-)$ and has zero entries in the rows corresponding to any unique support in matrix G.

[3]In set theory, $A - B = A \cap \overline{B}$, where $\overline{B}$ is the complementary set of B.

Proof: Condition C1 in Definition 4 holds by Lemma 1. For condition C2, we need to prove two directions.

($\Rightarrow$)Recall that if transition t_j is enabled in the redundant controller $\mathcal{H}$ at time epoch t, it follows that $G(q_c[t] - B_c^- x_j[t]) \geq -\mathcal{D}x_j[t]$; therefore, one needs to show that $q_c[t] - B_c^- x_j[t] \geq 0$.

Let $z \equiv q_c[t] - B_c^- x_j[t]$ and $\mathcal{D}(:, j) \equiv \mathcal{D}x_j[t]$, where $z \in \mathcal{N}^{n_c}$ and $\mathcal{D}(:, j) \in \mathcal{N}^{\eta}$. From Lemma 1, it follows that $\mathcal{D} \geq 0$. If $\mathcal{D} = 0$, Corollary 1 states that matrix G has independent column support. The additional conditions on matrix $\mathcal{D}$ are proved by contradiction: Suppose $\mathcal{D}(:, j)$ has a nonzero entry at its kth row $[\mathcal{D}(k, j) \neq 0]$, and that column g_i in matrix G has unique support at its kth row. Let z_i denote the ith entry of z. The kth row of Gz is given by

$$g_{ki} \times z_i + \sum_{j \in \{\{1,2,\ldots,n_c\} - i\}} g_{kj} \times z_j = g_{ki} \times z_i$$

(because g_i has unique support at its kth row, $g_{kj} = 0$ for $j \in \{\{1, 2, \ldots, n_c\} - i\}$). Clearly, $g_{ki} \times z_i \geq -\mathcal{D}(k, j)$ as long as z_i is chosen such that $(-\mathcal{D}(k, j)/g_{ki}) \leq z_i < 0$. In other words, one can choose z such that $Gz \geq 0$ but $z \not\geq 0$, which means that the transition can be enabled in the redundant controller net but cannot be enabled in the given controller. This is a contradiction, and the result follows.

($\Leftarrow$)If G and $\mathcal{D}$ satisfy the conditions in the theorem, by rearranging the rows and columns of G and $\mathcal{D}$ (i.e., rearranging the order of places and transitions), one can always have $G = [\frac{G_{n_c}}{C}]$, where $G_{n_c} \in \mathcal{N}^{n_c \times n_c}$ is a diagonal matrix, and $\mathcal{D} = [\frac{0_{n_c}}{D}]$, where $0_{n_c} \in \mathcal{N}^{n_c \times m}$ is a matrix with zero entries. $z \geq 0$ follows from the top n_c inequalities of $Gz \geq -\mathcal{D}(:, j)$. It is easy to check that the remaining d inequalities also hold, and the result follows. ■

This section has given a complete characterization of nonseparate redundant Petri net controllers and has obtained necessary and sufficient conditions for them to be bisimulation equivalent to the given controller. Section IV-B describes how fault detection and identification can be achieved in nonseparate redundant Petri net controllers. An example is presented in Section VI.

B. Fault Detection and Identification

Suppose a nonseparate redundant Petri net implementation is used to protect against place faults. If, due to a fault, the number of tokens in place p_i is corrupted by c, the faulty state is given by $q_f[t] = q_h[t] + e_{p_i}$ (where e_{p_i} is an η-dimensional array with a unique nonzero entry at its ith position, i.e., $e_{p_i} = c \times [0 \ldots 0\,1\,0 \ldots 0]^T$, and $q_h[t] = Gq_c[t]$ is the marking of redundant controller under fault-free conditions). If one chooses matrix P to be a full-row rank $d \times \eta$ matrix such that $PG = 0_{d \times n_c}$, where $0_{d \times n_c}$ is a $d \times n_c$ matrix with zero entries, the parity check at time epoch t will result in the syndrome

$$s[t] = Pq_f[t] = P\,(q_h[t] + e_{p_i}) = 0 + Pe_{p_i} = c \times P(:, i). \quad (5)$$

Clearly, a single place fault can be detected if all columns of matrix P are nonzero. If, in addition, the columns of P are not rational multiples of each other,[4] then one can detect and identify a single place fault. Similarly, by imposing additional rank conditions on matrix P, multiple place faults can be detected and identified. For more details on how to choose matrix P, the interested readers are referred to [8] and [9]. The construction of matrices G and P becomes a bit clearer in Section V when the attention is focused on the special case of separate redundant Petri net controllers. These controllers keep the given controller intact and introduce additional places and connections to enable fault detection and identification.

[4]One needs to make sure that for all pairs of columns of P, there do not exist nonzero integers a and b such that $a \times P(:, i) = b \times P(:, j), i \neq j$.

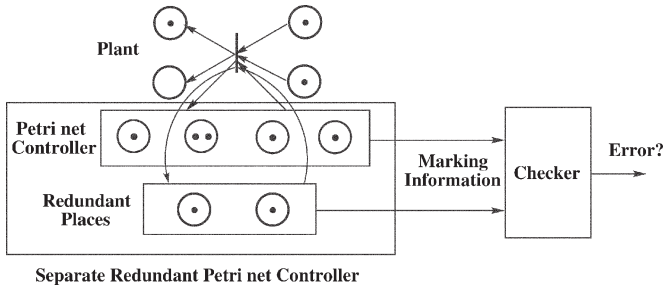

Fig. 6. Separate redundant Petri net controller.

V. Separate Redundant Petri Net Controllers

A. Designs of Separate Redundant Petri Net Controllers

A separate redundant Petri net controller retains the place/transition structure and state of the given controller, and incorporates additional places that enable the systematic detection and identification of faults. More specifically, the additional (redundant) places, connections, and tokens are used to impose invariant conditions that serve as consistency checks. As a result, by performing linear parity checks on the combined marking of the given controller places and the additional (redundant) places, one is able to detect and identify faults in the redundant Petri net controller in a systematic manner.

The overall scheme for constructing a separate redundant Petri net controller is shown in Fig. 6. Given a Petri net controller for a certain plant, the goal is to protect this controller from faults. For this reason, redundant places are added to the controller places so that the resulting separate redundant Petri net controller is bisimulation equivalent to the given controller, while, at the same time, the redundant places provide the capability of simultaneously capturing (up to a certain number of) faults. The overall redundant controller, just like the given controller, concurrently operates with the plant and takes actions based on activity and possible faults in the given controller. The checker is in charge of capturing faults by verifying that the internal state of the redundant controller is consistent.

Let $\mathcal{C}$ be a given Petri net controller with n_c places, m transitions, and state evolution as shown in (3).

Definition 9: A separate redundant Petri net controller of a given controller $\mathcal{C}$ is a nonseparate redundant Petri net $\mathcal{H}$ with $\eta \equiv n_c + d$ $(d > 0)$ places and m transitions whose state $q_h[t]$ satisfies $q_h[t] = \underbrace{\begin{bmatrix} I_{n_c} \\ C \end{bmatrix}}_{G} q_c[t]$ for all time epochs t.

Clearly, from the earlier analysis of nonseparate Petri net controllers, matrix C has to be a $d \times n_c$ matrix with

nonnegative integer entries. Moreover, if the state evolution of the separate redundant controller is written as

$$q_h[t+1] = q_h[t] + \underbrace{\begin{bmatrix} B_c^+ \\ X^+ \end{bmatrix}}_{\mathcal{B}_c^+} x[t] - \underbrace{\begin{bmatrix} B_c^- \\ X^- \end{bmatrix}}_{\mathcal{B}_c^-} x[t] \tag{6}$$

where $X^+ = CB_c^+ - D$ and $X^- = CB_c^- - D$ (for a $d \times m$ matrix D), the following lemma holds.

Lemma 3: Under fault-free conditions, a separate redundant Petri net controller $\mathcal{H}$ satisfies $(\mathcal{H}, q_h[t]) \sim (\mathcal{C}, q_c[t])$ if and only if $C \geq 0$ with integer entries and $D \geq 0$ with integer entries such that $D \leq \min(CB_c^+, CB_c^-)$.

Proof: The proof directly follows from the analysis of nonseparate Petri net controllers, where $G = [\frac{I_{n_c}}{C}]$ and $\mathcal{D} = [\frac{0_{n_c}}{D}]$. The requirement that $D \leq \min(CB_c^+, CB_c^-)$ follows from X^+ and X^- being matrices with nonnegative entries (they denote the arc weights of the additional places). ■

Remark 5: From the structural analysis of both types of redundant controllers (see the proof of Theorem 2), it is clear that any nonseparate redundant controller can be viewed as a separate redundant controller for a Petri net $\mathcal{C}'$ that can be obtained from Petri net $\mathcal{C}$ by taking each place and multiplying its initial tokens and all of its arc weights by some nonnegative integer entries.

B. Fault Detection and Identification

The separate redundant Petri net controller in Lemma 3 essentially encodes the given controller state $q_c[t]$ into a codeword $q_h[t]$ that consists of the given controller state and the state of the added places. A possibly invalid marking $q_f[t]$ can be checked by using the following parity check matrix:

$$P = [\, -C \quad I_d \,] \tag{7}$$

to verify whether the *syndrome*, defined as

$$s[t] \equiv P q_f[t] \tag{8}$$

is equal to 0.

For place faults, the syndrome at time epoch t is given by

$$s[t] \equiv P q_f[t] = P\left(q_h[t] + e_{p_i}\right) = P e_{p_i} \tag{9}$$

and detection and identification are exclusively determined by rank conditions on matrix P. For instance, to be able to detect and identify a single place fault, one needs to choose matrix C such that any two columns of matrix P are not rational multiples of each other. More sophisticated choices of matrix C (i.e., choosing the columns of matrix C to be linearly independent if possible) allow for the detection of multiple place faults [8], [9].

In the case of separate redundant Petri net controllers, fault detection and identification capability reduces to the problem of appropriately choosing matrices C and D. Since there are many choices for matrices C and D, one interesting future research direction is to develop criteria for searching among these various possibilities to find one that is optimal in a desirable way. (The objective could be to minimize the number of redundant places, the number of additional connections, and so on.) Section VI discusses examples of redundant Petri net controllers for the separate and nonseparate cases.

VI. Examples of Fault Detection and Identification Schemes

This section illustrates how to detect and identify a single place fault using redundant Petri net controllers. These concepts are illustrated using the Petri net controller in Fig. 4, which was introduced in Section III to control the production cell in Fig. 3. Note that the output and input incident matrices of the controller in Fig. 4 are given by

$$B_c^+ = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

$$B_c^- = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

The initial marking of the controller is

$$q_c[0] = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

A. Designs of Nonseparate Redundant Petri Net Controllers

The controller of Fig. 4 has two places and four transitions ($n_c = 2$, $m = 4$). To detect and identify a single place fault using a nonseparate redundant Petri net controller, two redundant places p_{c3} and p_{c4} need to be added (i.e., $d = 2$, $\eta = n_c + d = 4$), and one needs to verify that the parity check matrix P satisfies $PG = 0_{2\times 2}$ and has columns that are not rational multiples of each other (to allow the detection and identification of a single place fault).

Choosing the following matrix:

$$G = \begin{bmatrix} 0 & 3 \\ 1 & 2 \\ 2 & 1 \\ 2 & 0 \end{bmatrix}$$

(which, as required, has independent column support), it follows that

$$GB_c^+ = \begin{bmatrix} 0 & 3 & 0 & 6 \\ 1 & 2 & 1 & 4 \\ 2 & 1 & 2 & 2 \\ 2 & 0 & 2 & 0 \end{bmatrix} \quad GB_c^- = \begin{bmatrix} 0 & 0 & 3 & 3 \\ 2 & 1 & 2 & 2 \\ 4 & 2 & 1 & 1 \\ 4 & 2 & 0 & 0 \end{bmatrix}.$$

Then, choosing the following matrix:

$$\mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which satisfies $0 \leq \mathcal{D} \leq \min(GB_c^+, GB_c^-)$ and has zero entries in the first and fourth rows (where matrix G has unique

support) as required, the arc weights of the resulting nonseparate redundant controller are given by

$$\mathcal{B}_c^+ = GB_c^+ - \mathcal{D} = \begin{bmatrix} 0 & 3 & 0 & 6 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 2 & 0 & 2 & 0 \end{bmatrix}$$

$$\mathcal{B}_c^+ = GB_c^- - \mathcal{D} = \begin{bmatrix} 0 & 0 & 3 & 3 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 2 & 0 & 0 \end{bmatrix}.$$

One possible parity check matrix P is given by

$$P = \begin{bmatrix} -3 & 8 & -7 & 3 \\ 2 & -6 & 6 & -3 \end{bmatrix}$$

(note that $PG = 0_{2\times 2}$). Since the columns of matrix P are not rational multiples of each other, a single place fault can be detected and identified. More specifically, if the result of $Pq_f[t]$ is a multiple of

$$\begin{bmatrix} -3 \\ 2 \end{bmatrix} \quad \left(\text{or} \begin{bmatrix} 8 \\ -6 \end{bmatrix}, \begin{bmatrix} -7 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ -3 \end{bmatrix} \right)$$

then the number of tokens in place p_{c1} (or p_{c2}, p_{c3}, p_{c4}) has been corrupted.

B. Designs of Separate Redundant Petri Net Controllers

Given the earlier choice of matrices G and $\mathcal{D}$ in the nonseparate case, one can rearrange the rows and columns of these matrices to obtain

$$G' = \left[\begin{array}{cc} 2 & 0 \\ 0 & 3 \\ \hline 1 & 2 \\ 2 & 1 \end{array}\right]$$

$$\mathcal{D}' = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 \end{array}\right].$$

Note that if one chooses

$$C = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 \end{bmatrix}$$

the output and input incident matrices of the corresponding separate redundant controller (with $G = [\frac{I_{n_c}}{C}]$) are given by

$$\mathcal{B}_c^+ = \left[\frac{B_c^+}{CB_c^+ - D}\right] = \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ \hline 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

$$\mathcal{B}_c^- = \left[\frac{B_c^-}{CB_c^- - D}\right] = \left[\begin{array}{cccc} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{array}\right].$$

This choice of matrix C ensures that the columns of the parity check matrix P are not rational multiples of each other (which implies the ability to detect and identify a single place fault). In addition to the given controller places p_{c1} and p_{c2}, the resulting separate redundant controller has places p_{c3} and p_{c4}, which enable fault detection and identification. The parity check is performed through the following matrix:

$$P = [-C \quad I_2] = \begin{bmatrix} -1 & -2 & 1 & 0 \\ -2 & -1 & 0 & 1 \end{bmatrix}.$$

If the result is a multiple of

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \left(\text{or} \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

then the number of tokens in place p_{c1} (or p_{c2}, p_{c3}, p_{c4}) has been corrupted. ■

Remark 6: In the examples above, both types of redundant controllers require two redundant places and 16 arcs (number of nonzero entries in $\mathcal{B}_c^+$ and $\mathcal{B}_c^-$) to detect and identify a single place fault.

Remark 7: For any choice of matrices G and $\mathcal{D}$ in a nonseparate redundant controller, one can always rearrange rows and columns so that $G' = [\frac{G_{n_c}}{C}]$ (with G_{n_c} a diagonal matrix with nonnegative integer entries) and $\mathcal{D}' = [\frac{0_{n_c}}{D}]$. By choosing these same matrices C and D, a separate redundant controller can be constructed, which will need the same number of places and connections to detect and identify the same number of faults (note that the arc weights are not necessarily the same because in the separate case, $G = [\frac{I_{n_c}}{C}]$).

VII. Potential Advantages of Nonseparate Petri Net Controllers

Normally, the redundant Petri net controller that is replacing the original one would have to be bisimulation equivalent to the original controller to enable and disable the same transitions as the original one. This is true if, for example, the controller needs to be maximally permissive,[5] and the original controller was designed to achieve this. Of course, under the bisimulation equivalence requirement, it has been shown that both types of redundant controllers have the same fault detection and identification capabilities. If, however, one is willing to perform some additional processing before enabling/disabling transitions in the plant, then nonseparate embeddings do not have to be bisimulation equivalent to the original controller. The reason is that a redundant embedding always allows one to recover the state of the original controller (namely, by decoding the state of the redundant embedding), which can then be used to determine whether transitions in the plant should be enabled or disabled. This situation is illustrated in Fig. 7.

Below, an example is provided to show that if the bisimulation condition is not enforced, the resulting nonseparate redundant Petri net controller may require less connections than a separate redundant controller with the same fault detection and

[5]A controller is *maximally permissive* if it does not disable any transitions unless they would be in direct violation of the control objectives [36].

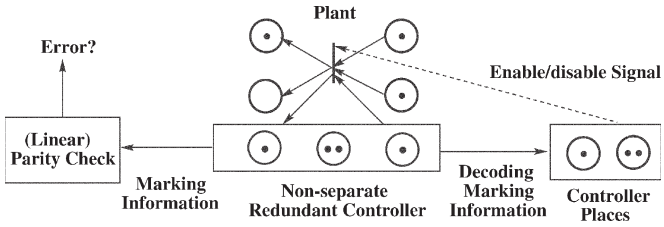

Fig. 7. Controller obtained from decoded state of nonseparate embedding.

identification capabilities. Notice that the nonseparate redundant embedding is chosen so that it always allows all transitions that would have been allowed by the original controller; this essentially means that the number of tokens in the nonseparate controller never goes negative, i.e., it is a proper Petri net.

Consider the Petri net system and its associated Petri net controller in Fig. 4. Suppose that one chooses matrix

$$G = \begin{bmatrix} 0 & 3 \\ 1 & 2 \\ 2 & 1 \\ 2 & 0 \end{bmatrix}$$

and matrix

$$\mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & 3 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

which satisfies $0 \leq \mathcal{D} \leq \min(GB_c^+, GB_c^-)$, but does not have zero entries in the first and fourth rows (i.e., bisimulation equivalence is not enforced). In this case, the arc weights in the nonseparate redundant controller are given by

$$\mathcal{B}_c^+ = GB_c^+ - \mathcal{D} = \begin{bmatrix} 0 & 3 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

$$\mathcal{B}_c^- = GB_c^- - \mathcal{D} = \begin{bmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 \end{bmatrix}.$$

The parity check matrix P is the same as in the last section and is given by

$$P = \begin{bmatrix} -3 & 8 & -7 & 3 \\ 2 & -6 & 6 & -3 \end{bmatrix}.$$

If the result of $Pq_f[t]$ is a multiple of

$$\begin{bmatrix} -3 \\ 2 \end{bmatrix} \left(\text{or} \begin{bmatrix} 8 \\ -6 \end{bmatrix}, \begin{bmatrix} -7 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ -3 \end{bmatrix} \right)$$

then the number of tokens in place p_{c1} (or p_{c2}, p_{c3}, p_{c4}) has been corrupted.

Remark 8: The total number of arcs needed in this case to detect and identify a single place fault is 14 (the previous cases needed 16 arcs). Therefore, when bisimulation equivalence is not enforced by the redundant embedding (but by the checker), nonseparate redundant Petri net controllers can have advantages over separate redundant controllers.

Remark 9: The ability of nonseparate redundant controllers to use fewer connections to detect and identify the same number of faults is a direct result of relaxing the bisimulation equivalence requirement and having additional flexibility in choosing the matrix $\mathcal{D}$. This comes at the cost of requiring a decoding of the state of the nonseparate redundant controller to determine whether a transition should be enabled or not.

VIII. Conclusion and Future Work

This paper proposes methodologies for detecting and identifying faults in a plant controller that is modeled as a Petri net. Two classes of redundant Petri net implementations are defined and characterized, both of which are bisimulation equivalent to the original controller and can be used for systematically constructing detection and identification schemes for place faults. Separate redundant Petri net controllers are constructed by adding redundant places to the given Petri net controller in a way that encodes information and enables fault detection and identification to be performed using algebraic techniques, whereas nonseparate redundant Petri net controllers allow more design flexibility than the separate ones. Necessary and sufficient conditions for these redundant controllers to be bisimulation equivalent to the given Petri net controller are derived. It has been also shown that, under the bisimulation equivalence requirement, both types of redundant controllers have the same fault detection and identification capabilities; however, in situations where the redundant controllers are used as monitors to perform fault detection and identification, the nonseparate redundant controllers can have advantages over the separate ones (e.g., they can use fewer connections to detect and identify the same number of faults).

Future extensions of this work include the development of schemes for Petri nets with uncontrollable and unobservable transitions. Apart from the fact that appropriate (original) Petri net controllers might not be necessarily obtainable under these assumptions, the challenge here is that the fault detection and identification procedures may also need to be modified. Another important future direction is to characterize structures for nonseparate redundant controllers to have minimal hardware implementation cost. It would be also interesting to design distributed/hierarchical monitoring schemes for large Petri net systems by enforcing specific constraints on the Petri net embeddings.

References

[1] A. Sengupta, D. K. Chattopadhyay, A. Palit, A. K. Bandyopadhyay, and A. K. Choudhury, "Realization of fault-tolerant machines—Linear code application," *IEEE Trans. Comput.*, vol. C-30, no. 3, pp. 237–240, Mar. 1981.

[2] G. R. Redinbo, "Finite field fault-tolerant digital filtering architectures," *IEEE Trans. Comput.*, vol. C-36, no. 10, pp. 1236–1242, Oct. 1987.

[3] C. N. Hadjicostis and G. C. Verghese, "Encoded dynamics for fault tolerance in linear finite-state machines," *IEEE Trans. Autom. Control*, vol. 47, no. 1, pp. 189–192, Jan. 2002.

[4] G. X. Wang and G. R. Redinbo, "Probability of state transition errors in a finite state machine containing soft failures," *IEEE Trans. Comput.*, vol. C-33, no. 3, pp. 269–277, Mar. 1984.

[5] R. Leveugle and G. Saucier, "Optimized synthesis of concurrently checked controllers," *IEEE Trans. Comput.*, vol. 39, no. 4, pp. 419–425, Apr. 1990.
[6] C. N. Hadjicostis and G. C. Verghese, "Fault-tolerant computation in groups and semigroups," *J. Franklin Inst.*, vol. 339, no. 4/5, pp. 387–430, Jul./Aug. 2002.
[7] C. N. Hadjicostis, "Finite-state machine embeddings for nonconcurrent error detection and identification," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 142–153, Feb. 2005.
[8] C. N. Hadjicostis and G. C. Verghese, "Monitoring discrete event systems using Petri net embeddings," in *Proc. Appl. Theory Petri Nets*. London, U.K.: Springer-Verlag, 1999, vol. 1639, pp. 188–207.
[9] Y. Wu and C. N. Hadjicostis, "Algebraic approaches for fault identification in discrete-event systems," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 2048–2055, Dec. 2005.
[10] L. Li, C. N. Hadjicostis, and R. S. Sreenivas, "Fault detection and identification in Petri net controllers," in *Proc. 43rd IEEE Conf. Decision Control*, Dec. 2004, pp. 5248–5253.
[11] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 1992, vol. 2, pp. 974–979.
[12] F.-Y. Wang, "Supervisory control for concurrent discrete event dynamic systems based on Petri nets," in *Proc. 31st IEEE Conf. Decision Control*, Dec. 1992, pp. 1196–1197.
[13] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Norwell, MA: Kluwer, 1993.
[14] Y. Li and W. Wonham, "Control of vector discrete-event systems II—Controller synthesis," *IEEE Trans. Autom. Control*, vol. 39, no. 3, pp. 512–531, Mar. 1994.
[15] K. Yamalidou, J. O. Moody, M. D. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15–28, Jan. 1996.
[16] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004.
[17] M. D. Jeng and X. L. Xie, "ERCN merged nets for modeling degraded behavior and parallel processes in semiconductor manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 102–112, Jan. 2004.
[18] L. E. Holloway, B. H. Krogh, and A. Giua, "A survey of Petri net models for controlled discrete event systems," *Discret. Event Dyn. Syst.: Theory Appl.*, vol. 7, no. 2, pp. 151–190, Apr. 1997.
[19] R. Bouyekhf and A. E. Moudni, "On the analysis of some structural properties of Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 6, pp. 784–794, Nov. 2005.
[20] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
[21] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Boston, MA: Kluwer, 1999.
[22] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
[23] P. Jancar, "Decidability questions for bisimilarity of Petri nets and some related problems," in *Proc. 11th Annu. Symp. Theoretical Aspects Comput. Sci.*, 1994, pp. 581–592.
[24] E.-R. Olderog, *Strong Bisimilarity on Nets: A New Concept for Comparing Net Semantics*, vol. 354. New York: Springer-Verlag, 1988, pp. 549–573.
[25] R. Valk and G. Vidal-Naquet, "Petri nets and regular languages," *J. Comput. Syst. Sci.*, vol. 23, no. 3, pp. 299–325, Dec. 1981.
[26] R. Milner, *Communication and Concurrency*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[27] Y. Hirshfeld, "Petri nets and the equivalence problem," in *Proc. 7th Workshop Comput. Sci. Logic*, 1994, pp. 165–174.
[28] P. Jancar and F. Moller, "Checking regular properties of Petri nets," in *Proc. 6th Int. Conf. Concurrency Theory*, 1995, pp. 348–362.
[29] P. Jancar, "Undecidability of bisimilarity for Petri nets and some related problems," *Theor. Comput. Sci.*, vol. 148, no. 2, pp. 281–301, Sep. 1995.
[30] P. Jancar and J. Esparza, "Deciding finiteness of Petri nets up to bisimulation," in *Proc. 23rd Int. Colloq. Automata, Lang. Program.*, 1996, pp. 478–489.
[31] E. Mayr, "An algorithm for the general Petri net reachability problem," *SIAM J. Comput.*, vol. 13, no. 3, pp. 441–460, Aug. 1984.
[32] B. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*. Reading, MA: Addison-Wesley, 1989.
[33] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems: Design and Evaluation*, 3rd ed. Wellesley, MA: A.K. Peters, 1998.
[34] C. N. Hadjicostis, *Coding Approaches to Fault Tolerance in Combinational and Dynamic Systems*. Norwell, MA: Kluwer, 2002.
[35] D. Andreu, J.-C. Pascal, and R. Valette, "Fuzzy Petri net-based programmable logic controller," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 6, pp. 952–961, Dec. 1997.
[36] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

Lingxi Li (S'03) received the B.E. degree in automation from Tsinghua University, Beijing, China, in 2000 and the M.S. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2003. He is currently working toward the Ph.D. degree in electrical and computer engineering at the University of Illinois at Urbana-Champaign, Urbana.

His research interests include discrete-event systems and fault-tolerant dynamic systems, with applications in manufacturing systems, power systems, and networks.

Christoforos N. Hadjicostis (M'99–SM'05) received the B.S. degrees in electrical engineering in 1993, computer science and engineering in 1993, and mathematics in 1999, the M.Eng. degree in electrical engineering and computer science in 1995, and the Ph.D. degree in electrical engineering and computer science in 1999, all from the Massachusetts Institute of Technology, Cambridge.

In August 1999, he joined the faculty of the University of Illinois at Urbana-Champaign, Urbana, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering and a Research Associate Professor with the Coordinated Science Laboratory. His research interests include systems and control, fault-tolerant combinational and dynamic systems, and fault diagnosis and management in large-scale systems and networks.

Ramavarapu S. Sreenivas (S'83–M'93–SM'02) received the B.Tech degree in electrical engineering from the Indian Institute of Technology, Madras, in 1985 and the M.S. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1987 and 1990, respectively.

He was a Postdoctoral Fellow in decision and control with the Division of Applied Sciences, Harvard University, Cambridge, MA, before joining the University of Illinois at Urbana-Champaign, Urbana, in September 1992. He is currently an Associate Professor with the Department of Industrial and Enterprise Systems Engineering and a Research Associate Professor with the Coordinated Science Laboratory. His research interests include modeling, analysis, control, and performance evaluation of discrete-state/discrete-event systems.