

ALGEBRAIC APPROACHES FOR CENTRALIZED AND DISTRIBUTED FAULT IDENTIFICATION IN DISCRETE EVENT SYSTEMS*

Yingquan Wu and Christoforos N. Hadjicostis[†]

Abstract

In this paper we develop algebraic approaches for fault identification in discrete event systems that are described by Petri nets. We consider faults in both Petri net transitions and places, and assume that system events are not directly observable but that the system state is periodically observable. The particular methodology we explore incorporates redundancy into a given Petri net in a way that enables fault detection and identification to be performed efficiently, in a centralized or distributed manner, using algebraic decoding techniques. The guiding principle in adding redundancy is to keep the number of additional Petri net places small while retaining enough information to be able to systematically detect and identify faults when the system state becomes available. The end result is a *redundant Petri net embedding* that uses $2k$ additional places and enables the simultaneous identification of $2k - 1$ transition faults and k place faults (that may occur at various instants during the operation of the Petri net). The proposed identification scheme has worst-case complexity of $O(k(m + n))$ (where m and n are respectively the number of transitions and places in the given Petri net) and can be extended to distributed settings in ways that require negligible additional hardware.

Keywords — Petri nets, discrete event systems, fault detection and identification, algebraic decoding, distributed fault diagnosis.

I INTRODUCTION

A commonly used approach to fault diagnosis in dynamic systems is to introduce analytical redundancy (characterized in terms of a parity space) and diagnose faults based on parity relations [1, 2]. The methodology in [3] uses a similar idea to monitor faults in DES that can be modeled by Petri nets. In its most general form, this approach encodes the state (marking) of the original Petri net by embedding it into a redundant one in a way that preserves the state, evolution and properties of the original Petri net, while enabling an external mechanism to perform fault diagnosis. More specifically, faults in the Petri net transitions and/or places are identified via linear parity checks on the overall *encoded* state of the redundant Petri net embedding. Unlike analytical redundancy schemes for dynamic systems, the process of constructing a redundant Petri net embedding corresponds to adding

*This material is based upon work supported in part by the National Science Foundation under NSF Career Award No 0092696 and NSF ITR Award No 0085917, and in part by the Air Force Office of Scientific Research under Award No AFOSR DoD 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.

[†]The authors are with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Address for correspondence: Christoforos N. Hadjicostis, 148 CSL, 1308 West Main Street, Urbana, IL 61801-2307, USA. E-mail: chadjic@uiuc.edu.

redundancy in the system — in the form of additional places (sensors) and the connections (acknowledgments) associated with them. The benefit of this added redundancy is the ability to *guarantee* the quick detection of up to a certain (predetermined) number of faults in the system.

In this paper we consider a setting similar to the one in [3] (fault identification in a DES that can be modeled by a Petri net) where activity (transition firing) is *unobservable* but the state (Petri net marking) is *periodically observable*. More specifically, at the end of a period we observe the final state (marking) of the redundant Petri net embedding and, based on this information, we need to detect and identify faults that may have occurred in this period. To achieve this, we construct redundant Petri net embeddings in which the identification of multiple and mixed (transition and/or place) faults, even when certain state information is missing, can be done systematically via algebraic coding/decoding techniques. Apart from fault detection and identification guarantees, our goal in choosing an appropriate redundant Petri net embedding is to keep the amount of redundancy (as indicated by the number of additional places/sensors) small. Since we are primarily interested in being able to handle complex, possibly distributed systems, we also develop a variation of our approach that is scalable and can be used in distributed settings.

As we show in this paper, the use of a redundant Petri net embedding with $2k$ additional places (and the connections and tokens associated with them) allows the simultaneous identification of up to $2k - 1$ transition faults and up to k place faults. The worst-case complexity of the fault identification procedure is $O(k(m + n))$ where m and n are the number of Petri net transitions and places respectively. The identification procedure is based on algebraic techniques, such as traditional decoding methods (e.g., Berlekamp-Massey decoding [4, 5]) and more recently developed methodologies for solving systems of composite power polynomial equations [6, 7]. Note that in order to achieve such efficiency in the identification process we need to add redundancy into the original Petri net.

Since the occurrence of faults in large-scale discrete event systems (DES) can degrade their overall performance in unpredictable and possibly devastating ways, fault diagnosis and management in such systems has received considerable attention over the last two decades. In the remainder of this section we review this previous work, focusing primarily on approaches that are closely related to the scheme we present in this paper. In [8, 9], the DES is represented by a finite state machine (FSM) whose behavior is modeled as a regular language and faults are diagnosed using a diagnoser, i.e., an appropriately designed FSM that is capable of performing fault detection and identification by analyzing observable activity in the given FSM. Within this automata-based modeling formulation, these ideas are further extended in [10] where the authors study an integrated approach for diagnosis and control in DES by determining the sub-language of the legal language of a given DES that is supremely controllable, observable and diagnosable. Related approaches have also appeared for cases where failure specifications are given in linear temporal logic [11] or when there exist additional optimization objectives or other restrictions/requirements [12, 13, 14].

Apart from automata-based modeling formulations, previous work has also considered different DES models. For instance, the authors of [15] use a template language framework to represent a discrete event process, leading to on-line fault monitoring schemes for confirming correct system operation. In [16], the DES is modeled as a time Petri net and an analytical procedure for performing fault detection by back-firing transitions is developed. In [17], Petri net modeling is coupled with parameter trend and fault tree analysis to perform fault diagnosis. The monitoring of flexible manufacturing systems using colored timed Petri nets is investigated by [18]. Distributed and

asynchronous control/diagnosis in DES has been addressed more recently (see, for example, [19, 20, 21, 22, 23]). Also related to the work presented in this paper are diagnosis methodologies that identify faults in distributed settings using partially stochastic networks [24, 25].

The work in this paper is closely related to previous work that deals with observability properties of Petri net systems. For example, the authors of [26] (which provides an excellent background on this topic) present an approach to estimate the partially observable initial marking of a Petri net, given that the net structure is known and given that event occurrences are observable. In our setup, the initial and final states of the Petri net are known and our aim is to detect and identify faults that may have taken place during the evolution of the net, given that the event sequence is *not* observable but that the net structure is known. As evident from our analysis, our ability to detect and identify such faults depends critically on the amount of redundancy we add into the system. Finally, unlike [27, 28, 29, 30, 26], we do not discuss the issue of controlling the Petri net in a certain desirable way although it is possible that our approach here can be used in conjunction with such control strategies.

II Preliminaries

A Petri net notation

This section describes the notation and background needed for our Petri net development; more details can be found in [31, 32]. Petri nets can be used to model a variety of information and processing systems, including concurrent, asynchronous, distributed, nondeterministic, and/or stochastic systems, [32]. The functionality of a Petri net \mathcal{S} is best described by a directed, bipartite graph with two types of nodes: *places* (denoted by $\{P_1, P_2, \dots, P_n\}$ and drawn as circles) and *transitions* (denoted by $\{T_1, T_2, \dots, T_m\}$ and drawn as rectangles). Weighted directed arcs connect transitions to places and vice versa (but there are no connections from a place to a place or from a transition to a transition). The arc weights have to be nonnegative integers (we use b_{ij}^- to denote the weight of the arc from place P_i to transition T_j and b_{ij}^+ to denote the weight of the arc from transition T_j to place P_i).

Transitions denote system activity that causes the rearrangement, generation and consumption of tokens (which can be regarded as the resources that are available in the system). Places function as storage locations for tokens so that each place has a nonnegative integer number of tokens stored in it. At any given time instant t , the *marking* (state) of the Petri net is given by the number of tokens at its places. Transition T_j is *enabled* (i.e., it is allowed to take place) only if each of its input places P_i has at least b_{ij}^- tokens. When transition T_j takes place (we say that transition T_j fires), it removes b_{ij}^- tokens from each input place P_i , and deposits b_{ij}^+ tokens to each output place P_i . If $\mathbf{q}_s[t]$ denotes the state (marking) of the Petri net at time epoch t and $\mathbf{B}^- \triangleq [b_{ij}^-]$ (respectively, $\mathbf{B}^+ \triangleq [b_{ij}^+]$) denotes the $n \times m$ matrix with b_{ij}^- (respectively, b_{ij}^+) at its i th row, j th column position, then the state evolution of Petri net \mathcal{S} is captured by

$$\mathbf{q}_s[t+1] = \mathbf{q}_s[t] + \mathbf{B}^+ \mathbf{x}[t] - \mathbf{B}^- \mathbf{x}[t] = \mathbf{q}_s[t] + \mathbf{B} \mathbf{x}[t], \quad (1)$$

where $\mathbf{B} \triangleq \mathbf{B}^+ - \mathbf{B}^-$ and the input vector $\mathbf{x}[t] \in (\mathbb{Z}^+)^m$ indicates the transitions that take place (fire) at time epoch t ($\mathbf{x}[t]$ is usually assumed to be a unit vector with a single nonzero entry at its j th position indicating that transition T_j has fired). Note that transition T_j is enabled at time epoch t if and only if $\mathbf{q}_s[t] \geq \mathbf{B}^-(\cdot, j)$ (where the inequality is taken element-wise and $\mathbf{B}^-(\cdot, j)$ denotes the j th column of \mathbf{B}^-).

B Fault model

Following the development in [3] we consider three different fault models, i.e., models that allow us to abstract away from the particulars of a system implementation and the hardware failure modes associated with it. Naturally, given a particular DES and its corresponding Petri net model, we need to ensure that our fault model effectively captures the faults that are expected in the system by mapping them into a manageable algebraic representation; we elaborate on this issue at the end of this section.

(i) A *transition fault* models a fault in the mechanism that implements a certain Petri net transition. We say that transition T_j has a *post-condition fault* if no tokens are deposited at its output places (even though the tokens from its input places are consumed). Similarly, we say that transition T_j has a *pre-condition fault* if the tokens that are supposed to be removed from the input places are not removed (even though tokens are deposited at the corresponding output places).

Let $\mathbf{e}_T^+ \in (\mathbb{Z}^+)^m$ denote an indicator vector of post-condition faults and $\mathbf{e}_T^- \in (\mathbb{Z}^+)^m$ denote an indicator vector of pre-condition faults. More specifically, the j th entry of \mathbf{e}_T^+ (\mathbf{e}_T^-) is a nonnegative integer indicating the number of post-condition (pre-condition) faults that have affected transition T_j . Then, the erroneous state $\mathbf{q}_f[t]$ at time epoch t can be expressed as

$$\mathbf{q}_f[t] = \mathbf{q}_s[t] - \mathbf{B}^+ \mathbf{e}_T^+ + \mathbf{B}^- \mathbf{e}_T^-, \quad (2)$$

where $\mathbf{q}_s[t]$ is the state that would have been reached under fault-free conditions [3].

(ii) A *place fault* models a fault that corrupts the number of tokens in a single place of the Petri net. A place fault at time epoch t results in an erroneous state $\mathbf{q}_f[t]$ that can be expressed as

$$\mathbf{q}_f[t] = \mathbf{q}_s[t] + \mathbf{e}_P, \quad (3)$$

where $\mathbf{q}_s[t]$ is the state that would have been reached under fault-free conditions and \mathbf{e}_P is an n -dimensional vector with a unique nonzero integer entry. More specifically, if $e_P^i < 0$ then the number of tokens in the i th place has decreased due to the fault, whereas if $e_P^i > 0$ then the number of tokens in the i th place has increased. Notice that when state information from a certain place is missing, one can assign zero (or an arbitrary value) as the number of tokens for this place and treat this situation as a place fault; therefore, place faults can also be used to model missing state information.

(iii) The *additive fault model* is a generalization of the above fault models and is based on explicitly modeling each fault f by its additive effect \mathbf{e}_f on the fault-free state $\mathbf{q}_s[t]$ that the Petri net would be in had the fault been absent, so that

$$\mathbf{q}_f[t] = \mathbf{q}_s[t] + \mathbf{e}_f \quad (4)$$

for some appropriate $\mathbf{e}_f \in \mathbb{Z}^n$.

Note that by combining (2) and (3), we can model the composite effect of transition and place faults on the erroneous state at time epoch t as

$$\mathbf{q}_f[t] = \mathbf{q}_s[t] - \mathbf{B}^+ \mathbf{e}_T^+ + \mathbf{B}^- \mathbf{e}_T^- + \mathbf{e}_P. \quad (5)$$

It is also worth noting that the additive fault model can capture the effects of multiple faults. For example, if faults

$\mathbf{e}_{1,T}^+, \mathbf{e}_{1,T}^-, \mathbf{e}_{1,P}$ occur at time epoch t_1 and faults $\mathbf{e}_{2,T}^+, \mathbf{e}_{2,T}^-, \mathbf{e}_{2,P}$ occur at time epoch t_2 ($t_1 < t_2$), then the resulting erroneous state at time epoch t_2 will be

$$\mathbf{q}_f[t_2] = \mathbf{q}_s[t_2] - \mathbf{B}^+(\mathbf{e}_{1,T}^+ + \mathbf{e}_{2,T}^+) + \mathbf{B}^-(\mathbf{e}_{1,T}^- + \mathbf{e}_{2,T}^-) + (\mathbf{e}_{1,P} + \mathbf{e}_{2,P}) .$$

We will show that multiple such faults can be identified by performing checks periodically/non-concurrently, as long as we add enough redundancy to guarantee that information about the occurrence of faults is not lost. Our goal will be to add a small amount of redundancy so that we can achieve our goal using efficient detection/identification algorithms.

Before closing this session, we would like to add a remark on fault modeling. Clearly, a pre-condition (post-condition) fault on a transition that has n_T input (output) places can also be treated as a combination of n_T place faults. Similarly, a fault whose additive effect is captured by a vector \mathbf{e}_f with n_f nonzero entries can also be treated as a combination of n_f place faults. In order for the fault identification mechanism to be able to resolve such conflicts, we aim at determining the *minimum* number of transition and/or place faults that explain the behavior observed in the Petri net. The underlying assumption in this formulation is that all transition and/or place faults are equally likely and independent (therefore, the most likely explanation is the one that involves the minimum number of faults).

C Redundant Petri net embeddings

In [3] the identification of faults in a given Petri net \mathcal{S} is facilitated by the construction of a *redundant Petri net embedding* \mathcal{H} . More specifically, d places are added to the original Petri net \mathcal{S} to form a composite Petri net \mathcal{H} whose state (marking) $\mathbf{q}_h[t]$ is η -dimensional ($\eta = n + d, d > 0$) and under fault-free conditions satisfies

$$\mathbf{q}_h[t] = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{C} \end{bmatrix} \mathbf{q}_s[t] \quad (6)$$

for all time epochs t . Here, $\mathbf{q}_s[t]$ is the state of the original Petri net \mathcal{S} , \mathbf{I}_n denotes the $n \times n$ identity matrix and \mathbf{C} is a $d \times n$ integer matrix to be designed. In order to guarantee that (6) remains valid for all t , the state evolution of \mathcal{H} is chosen to be of the form

$$\begin{aligned} \mathbf{q}_h[t+1] &= \mathbf{q}_h[t] + \underbrace{\begin{bmatrix} \mathbf{B}^+ \\ \mathbf{CB}^+ - \mathbf{D} \end{bmatrix}}_{\mathcal{B}^+} \mathbf{x}[t] - \underbrace{\begin{bmatrix} \mathbf{B}^- \\ \mathbf{CB}^- - \mathbf{D} \end{bmatrix}}_{\mathcal{B}^-} \mathbf{x}[t] \\ &= \mathbf{q}_h[t] + (\mathcal{B}^+ - \mathcal{B}^-) \mathbf{x}[t], \end{aligned} \quad (7)$$

where \mathbf{D} is a $d \times m$ integer matrix, also to be designed. The d additional places together with the n initial places comprise the places of the *redundant Petri net embedding* \mathcal{H} . In [3] it is shown that if matrices \mathbf{C} and \mathbf{D} have integer nonnegative entries and satisfy $\mathbf{CB}^+ - \mathbf{D} \geq \mathbf{0}$ and $\mathbf{CB}^- - \mathbf{D} \geq \mathbf{0}$ (element-wise), then a properly initialized redundant Petri net embedding \mathcal{H} (i.e., one that satisfies Eq. (6) at $t = 0$) admits any firing sequence that is admissible in the original Petri net \mathcal{S} . In other words, these choices for \mathbf{C} and \mathbf{D} ensure that the additional places do not inhibit any of the functionality in the original Petri net. Note that if the additional places do *not*

function as controllers but simply as sensors, then these constraints are not present (because the redundant Petri net embedding is then guaranteed to admit any firing sequence that is admissible in the original Petri net).

By using a redundant Petri net embedding, we essentially introduce additional sensors in the system in a way that encodes the original state $\mathbf{q}_s[t]$ into a *codeword* $\mathbf{q}_h[t]$ that consists of the original state and the state of the added places. Although codeword-like constraints may already be present in the state of a given Petri net (in the form of place invariants for instance), in general we need to introduce additional places to enforce such constraints. The validity of the codeword can be checked by using the parity check matrix $\mathbf{P} \triangleq [-\mathbf{C} \ \mathbf{I}_d]$ to verify that the *syndrome*

$$\begin{aligned} \mathbf{s}[t] &\triangleq \mathbf{P}\mathbf{q}_f[t] \\ &= [-\mathbf{C} \ \mathbf{I}_d]\mathbf{q}_f[t] \end{aligned} \tag{8}$$

(where $\mathbf{q}_f[t]$ is the possibly faulty state at time epoch t) is zero. A nonzero syndrome at time epoch t indicates the presence of one or more faults. Faults can be identified based on the syndrome $\mathbf{s}[t]$ if matrices \mathbf{C} and \mathbf{D} are designed so that each combination of faults results in a unique syndrome. Notice that in order to verify (8) the fault identification mechanism needs to know the number of tokens in each place of the redundant Petri net embedding.

D Problem formulation

We assume that the firing of transitions in the redundant Petri net is not directly observable while the Petri net marking is *periodically observable*. We aim to identify faults based on the observed marking at the end of a period. We use the term “non-concurrent” to capture the fact that diagnosis is performed over a period of several time epochs, in this case once every N time epochs. We assume that each transition may not suffer both pre-condition and post-condition faults within the epoch interval $[1, N]$ (actually, if a particular transition suffers both a pre-condition and a post-condition fault within $[1, N]$, their effects will be cancelled, making their non-concurrent detection impossible).

Let $\mathbf{e}_T^+ \in (\mathbb{Z}^+)^m$ denote an indicator vector of post-condition faults and $\mathbf{e}_T^- \in (\mathbb{Z}^+)^m$ denote an indicator vector of pre-condition faults within the epoch interval $[1, N]$. Assuming no place faults, the erroneous state $\mathbf{q}_f[N]$ at time epoch N is given by

$$\mathbf{q}_f[N] = \mathbf{q}_h[N] - \mathcal{B}^+ \mathbf{e}_T^+ + \mathcal{B}^- \mathbf{e}_T^-, \tag{9}$$

where $\mathbf{q}_h[N]$ is the state that would have been reached under fault-free conditions. The fault syndrome at time epoch N is then

$$\begin{aligned} \mathbf{s}_T[N] &\triangleq \mathbf{P}\mathbf{q}_f[N] \\ &= [-\mathbf{C} \ \mathbf{I}_d](\mathbf{q}_h[N] - \mathcal{B}^+ \mathbf{e}_T^+ + \mathcal{B}^- \mathbf{e}_T^-) \\ &= [-\mathbf{C} \ \mathbf{I}_d] \left(\mathbf{q}_h[N] - \begin{bmatrix} \mathbf{B}^+ \\ \mathbf{C}\mathbf{B}^+ - \mathbf{D} \end{bmatrix} \mathbf{e}_T^+ + \begin{bmatrix} \mathbf{B}^- \\ \mathbf{C}\mathbf{B}^- - \mathbf{D} \end{bmatrix} \mathbf{e}_T^- \right) \end{aligned}$$

and is easily calculated to be

$$\mathbf{s}_T[N] = \mathbf{D}\mathbf{e}_T, \tag{10}$$

where $\mathbf{e}_T \triangleq \mathbf{e}_T^+ - \mathbf{e}_T^-$. Clearly, the identification of transition faults based on the syndrome $\mathbf{s}_T[N]$ is completely determined by matrix \mathbf{D} . For example, if we choose all columns of \mathbf{D} to be distinct (and not the negatives of each other), we can identify any single transition fault (as well as its type); as discussed later, more sophisticated designs of \mathbf{D} allow us to efficiently identify multiple transition faults using algebraic techniques.

Assuming no transition faults, place faults within the time epoch interval $[1, N]$ result in a corrupted state

$$\mathbf{q}_f[N] = \mathbf{q}_h[N] + \mathbf{e}_P, \quad (11)$$

where $\mathbf{q}_h[N]$ is the state that would have been reached under fault-free transitions and $\mathbf{e}_P \in \mathbb{Z}^n$ denotes the (accumulated) place fault vector. The fault syndrome in this case is given by

$$\mathbf{s}_P[N] \triangleq \mathbf{P}\mathbf{q}_f[N] = \mathbf{P}\mathbf{e}_P, \quad (12)$$

i.e., the identification of place faults is exclusively determined by \mathbf{P} . For example, if we choose \mathbf{C} such that any two columns of $\mathbf{P} = [-\mathbf{C} \mathbf{I}_d]$ are linearly independent, then any single place fault is identifiable. Again, as discussed later, more sophisticated choices of \mathbf{C} allow us to identify multiple place faults. Moreover, careful choices of both \mathbf{D} and \mathbf{C} enable the efficient identification of multiple and mixed (transition and/or place) faults.

III MATHEMATICAL BACKGROUND

In this paper we frequently make references to operations in $\text{GF}(p)$, the Galois field of order p , where p is a prime number. In these fields, addition and multiplication can essentially be treated as addition and multiplication modulo p .

For notational simplicity, we define

$$\Lambda_\tau(x_1, x_2, \dots, x_r) \triangleq (-1)^\tau \sum_{1 \leq i_1 < i_2 < \dots < i_\tau \leq r} x_{i_1} x_{i_2} \dots x_{i_\tau}, \quad \tau \leq r, \quad (13)$$

and, for purposes of consistency, we set $\Lambda_0(x_1, x_2, \dots, x_r) = 1$ and $\Lambda_\tau(x_1, x_2, \dots, x_r) = 0$ for any $\tau > r$. We also define

$$\mathcal{S}_\tau(x_1, x_2, \dots, x_r) \triangleq \sum_{i=1}^r x_i^\tau, \quad \tau \geq 0, \quad (14)$$

and, when there is no ambiguity in the context, we use Λ_τ , \mathcal{S}_τ to represent $\Lambda_\tau(x_1, x_2, \dots, x_r)$, $\mathcal{S}_\tau(x_1, x_2, \dots, x_r)$ respectively. Clearly, we have the general equality

$$x^r + \Lambda_1(x_1, x_2, \dots, x_r)x^{r-1} + \Lambda_2(x_1, x_2, \dots, x_r)x^{r-2} + \dots + \Lambda_r(x_1, x_2, \dots, x_r) = \prod_{i=1}^r (x - x_i). \quad (15)$$

Proposition 1 *Let x_1, x_2, \dots, x_r be r variables and $p > r$. Then,*

(i) Λ_i and \mathcal{S}_i , $i = 1, 2, \dots, r$, satisfy the following relations:

$$\begin{cases} \mathcal{S}_1 + \Lambda_1 & = 0, \\ \mathcal{S}_2 + \Lambda_1 \mathcal{S}_1 + 2\Lambda_2 & = 0, \\ & \vdots \\ \mathcal{S}_r + \Lambda_1 \mathcal{S}_{r-1} + \dots + \Lambda_{r-1} \mathcal{S}_1 + r\Lambda_r & = 0. \end{cases} \quad (16)$$

(ii) If $\mathcal{S}_1 = s_1, \mathcal{S}_2 = s_2, \dots, \mathcal{S}_r = s_r$, then there is at most one solution (x_1, x_2, \dots, x_r) (up to reordering of the elements). Specifically, x_1, x_2, \dots, x_r are the r roots of the equation

$$x^r + \Lambda_1 x^{r-1} + \Lambda_2 x^{r-2} + \dots + \Lambda_{r-1} x + \Lambda_r = 0, \quad (17)$$

where the Λ_i , $i = 1, 2, \dots, r$, can be obtained uniquely from (16). \square

The equations in (16) are known as Newton's identities; refer to [33, 4, 5] for details. We highlight that the above equality is consistent to setting $\Lambda_\tau(x_1, x_2, \dots, x_r)$ to zero for any $\tau > r$ in the previous paragraph. Clearly, for $i = 1, 2, \dots, r$, we have

$$\begin{aligned} \Lambda_i &\stackrel{\Delta}{=} \mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_i) \\ &= \frac{(-1)^i}{i!} \begin{vmatrix} \mathcal{S}_1 & 1 & 0 & \dots & 0 & 0 \\ \mathcal{S}_2 & \mathcal{S}_1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{S}_{i-1} & \mathcal{S}_{i-2} & \mathcal{S}_{i-3} & \dots & \mathcal{S}_1 & i-1 \\ \mathcal{S}_i & \mathcal{S}_{i-1} & \mathcal{S}_{i-2} & \dots & \mathcal{S}_2 & \mathcal{S}_1 \end{vmatrix} \end{aligned} \quad (18)$$

and the Λ_i are independent of the number of original variables x_1, x_2, \dots (note that since Λ_i has a factor of $\frac{1}{i!}$, the \mathcal{F}_i for $i = 1, 2, \dots, r$ are well defined only if $p > r$). Now, we deduce this observation to a particular condition, as identified in the following proposition.

Proposition 2 Let x_1, x_2, \dots, x_r be r variables and s_1, s_2, \dots, s_τ be τ known parameters in $GF(p)$ with $p > \tau$. Then, $\mathcal{F}_\tau(s_1 + \mathcal{S}_1, s_2 + \mathcal{S}_2, \dots, s_\tau + \mathcal{S}_\tau)$ is a linear combination of Λ_l , $l = 1, 2, \dots, r$. Moreover,

$$\mathcal{F}_\tau(s_1 + \mathcal{S}_1, s_2 + \mathcal{S}_2, \dots, s_\tau + \mathcal{S}_\tau) = \sum_{i=0}^{\min\{r, \tau\}} \mathcal{F}_{\tau-i}(s_1, s_2, \dots, s_{\tau-i}) \cdot \Lambda_i. \quad (19)$$

\square

The proof can be found in [6, 7].

IV CENTRALIZED FAULT IDENTIFICATION

In this section we discuss centralized fault detection and identification, first for transition faults, then for place faults and eventually for combinations of transition and place faults. In each case, the diagnoser aims at identifying the combination of transition and/or place faults that has minimum cardinality and explains the observed behavior.

A Identification of transition faults

In this section we discuss centralized identification of *up to k transition faults* within the epoch interval $[1, N]$. The expression in Eq. (10) indicates that k or less transition faults are identifiable via a parity check if and only if the syndrome for transition faults $\mathbf{s}_T[N] = \mathbf{D}\mathbf{e}_T$ is unique for any \mathbf{e}_T such that $|\mathbf{e}_T| \triangleq \sum_{i=1}^m |e_T^i| \leq k$ (by assumption no cancellations take place in \mathbf{e}_T , i.e., no transition suffers both pre-condition and post-condition faults within the epoch interval $[1, N]$, which implies that $|\mathbf{e}_T| = |\mathbf{e}_T^+| + |\mathbf{e}_T^-|$). In the following analysis, we aim at designing matrix \mathbf{D} to achieve this objective.

Consider the following choice for matrix \mathbf{D} :

$$\mathbf{D}_{k+1} \triangleq \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & m \\ 1 & 2^2 \bmod p & 3^2 \bmod p & \dots & m^2 \bmod p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^k \bmod p & 3^k \bmod p & \dots & m^k \bmod p \end{pmatrix}, \quad (20)$$

where p is a prime number larger than m (the subscript $_{k+1}$ is used to indicate the row dimension of \mathbf{D} , which is essentially the number of additional places). Note that matrix \mathbf{C} does not directly¹ enter the development here and we consider it later when we discuss identification of place faults.

We now demonstrate that the above choice for matrix \mathbf{D} allows the identification of up to k transition faults by establishing that the fault syndrome $\mathbf{s}_T[N] \triangleq [s_0 \ s_1 \ s_2 \ \dots \ s_k]^T = \mathbf{D}\mathbf{e}_T$ is nonzero and unique for any \mathbf{e}_T that satisfies $|\mathbf{e}_T| \triangleq \sum_{i=1}^m |e_T^i| \leq k$. Note that $s_0 = |\mathbf{e}_T^+| - |\mathbf{e}_T^-|$ (for notational simplicity we let $\tau \triangleq s_0$). Without loss of generality, we assume that $\tau \geq 0$ and show that each combination of up to k faults causes a unique (nonzero) syndrome $\mathbf{s}_T[N]$, implying that all combinations of up to k transition faults are identifiable. Suppose that a combination of $\tau + i$ post-condition faults and i pre-condition faults at transitions $[x_1, x_2, \dots, x_{\tau+i}, x_{\tau+i+1}, \dots, x_{\tau+2i}]$ ($\tau + 2i \leq k$) results in the same syndrome as a combination of $\tau + j$ post-condition faults and j pre-condition faults at transitions $[y_1, y_2, \dots, y_{\tau+j}, y_{\tau+j+1}, \dots, y_{\tau+2j}]$ ($\tau + 2j \leq k$). Then, the following equation array holds:

$$\left\{ \begin{array}{l} \sum_{l=1}^{\tau+i} (x_l \bmod p) - \sum_{l=\tau+i+1}^{\tau+2i} (x_l \bmod p) = s_1 = \sum_{l=1}^{\tau+j} (y_l \bmod p) - \sum_{l=\tau+j+1}^{\tau+2j} (y_l \bmod p), \\ \sum_{l=1}^{\tau+i} (x_l^2 \bmod p) - \sum_{l=\tau+i+1}^{\tau+2i} (x_l^2 \bmod p) = s_2 = \sum_{l=1}^{\tau+j} (y_l^2 \bmod p) - \sum_{l=\tau+j+1}^{\tau+2j} (y_l^2 \bmod p), \\ \vdots \\ \sum_{l=1}^{\tau+i} (x_l^k \bmod p) - \sum_{l=\tau+i+1}^{\tau+2i} (x_l^k \bmod p) = s_k = \sum_{l=1}^{\tau+j} (y_l^k \bmod p) - \sum_{l=\tau+j+1}^{\tau+2j} (y_l^k \bmod p), \end{array} \right. \quad (21)$$

where $\tau + 2i \leq k$, $\tau + 2j \leq k$. We can weaken and rewrite (21) in the following form:

$$\left\{ \begin{array}{l} \sum_{l=1}^{\tau+i} x_l + \sum_{l=\tau+j+1}^{\tau+2j} y_l \equiv \sum_{l=1}^{\tau+j} y_l + \sum_{l=\tau+i+1}^{\tau+2i} x_l, \\ \sum_{l=1}^{\tau+i} x_l^2 + \sum_{l=\tau+j+1}^{\tau+2j} y_l^2 \equiv \sum_{l=1}^{\tau+j} y_l^2 + \sum_{l=\tau+i+1}^{\tau+2i} x_l^2, \\ \vdots \\ \sum_{l=1}^{\tau+i} x_l^k + \sum_{l=\tau+j+1}^{\tau+2j} y_l^k \equiv \sum_{l=1}^{\tau+j} y_l^k + \sum_{l=\tau+i+1}^{\tau+2i} x_l^k, \end{array} \right. \quad (22)$$

¹Recall that if the additional places function as controllers, then one has to ensure that the redundant Petri net embedding \mathcal{H} admits all firing sequences that are admissible in the original Petri net \mathcal{S} by choosing matrix \mathbf{C} to have nonnegative integer entries and satisfy $\mathbf{C}\mathbf{B}^+ - \mathbf{D} \geq \mathbf{0}$ and $\mathbf{C}\mathbf{B}^- - \mathbf{D} \geq \mathbf{0}$ (element-wise).

where the equality is now taken modulo p . For notational simplicity, in the sequel we use the notation “ \equiv ” to stand for equality modulo p . Since $\tau + i + j \leq k$ and since no cancellations are allowed, we can use Proposition 1 to argue that $i = j$ and that, ignoring order, $\{x_1, x_2, \dots, x_{\tau+2i}\} = \{y_1, y_2, \dots, y_{\tau+2j}\}$. This establishes the uniqueness of syndrome $s_T[N]$ under this combination of transition faults. Therefore, the matrix \mathbf{D} defined in (20) allows the identification of k or less transition faults.

Next, we establish an efficient identification algorithm. Without loss of generality, we assume that $\tau + r$ post-condition faults and r pre-condition faults have taken place ($\tau + 2r \leq k$) and consider the following equation array:

$$\begin{cases} \sum_{l=1}^{\tau+r} (x_l \bmod p) - \sum_{l=\tau+r+1}^{\tau+2r} (x_l \bmod p) & = s_1, \\ \sum_{l=1}^{\tau+r} (x_l^2 \bmod p) - \sum_{l=\tau+r+1}^{\tau+2r} (x_l^2 \bmod p) & = s_2, \\ & \vdots \\ \sum_{l=1}^{\tau+r} (x_l^{\tau+2r} \bmod p) - \sum_{l=\tau+r+1}^{\tau+2r} (x_l^{\tau+2r} \bmod p) & = s_{\tau+2r}. \end{cases} \quad (23)$$

We can weaken and rewrite (23) in the following form:

$$\begin{cases} \sum_{l=1}^{\tau+r} x_l & \equiv s_1 + \sum_{l=\tau+r+1}^{\tau+2r} x_l, \\ \sum_{l=1}^{\tau+r} x_l^2 & \equiv s_2 + \sum_{l=\tau+r+1}^{\tau+2r} x_l^2, \\ & \vdots \\ \sum_{l=1}^{\tau+r} x_l^{\tau+2r} & \equiv s_{\tau+2r} + \sum_{l=\tau+r+1}^{\tau+2r} x_l^{\tau+2r}. \end{cases} \quad (24)$$

We recall that

$$\Lambda_l(x_1, x_2, \dots, x_{\tau+r}) = 0$$

for $l > \tau + r$. Using this observation in conjunction with Proposition 2, we conclude that for $l > \tau + r$

$$\begin{aligned} 0 &= \Lambda_l(x_1, \dots, x_{\tau+r}) \equiv \mathcal{F}_l(s_1 + \sum_{j=\tau+r+1}^{\tau+2r} x_j, \dots, s_l + \sum_{j=\tau+r+1}^{\tau+2r} x_j^l) \\ &= \sum_{j=0}^r \mathcal{F}_{l-j}(s_1, \dots, s_{l-j}) \cdot \Lambda_j(x_{\tau+r+1}, \dots, x_{\tau+2r}). \end{aligned}$$

By taking $l = \tau + r + 1, \tau + r + 2, \dots, \tau + 2r$, we obtain the following *linear* equation array:

$$\begin{cases} \sum_{j=1}^r \mathcal{F}_{\tau+r+1-j}(s_1, \dots, s_{\tau+r+1-j}) \cdot \Lambda_j(x_{\tau+r+1}, \dots, x_{\tau+2r}) \equiv -\mathcal{F}_{\tau+r+1}(s_1, \dots, s_{\tau+r+1}), \\ \sum_{j=1}^r \mathcal{F}_{\tau+r+2-j}(s_1, \dots, s_{\tau+r+2-j}) \cdot \Lambda_j(x_{\tau+r+1}, \dots, x_{\tau+2r}) \equiv -\mathcal{F}_{\tau+r+2}(s_1, \dots, s_{\tau+r+2}), \\ & \vdots \\ \sum_{j=1}^r \mathcal{F}_{\tau+2r-j}(s_1, \dots, s_{\tau+2r-j}) \cdot \Lambda_j(x_{\tau+r+1}, \dots, x_{\tau+2r}) \equiv -\mathcal{F}_{\tau+2r}(s_1, \dots, s_{\tau+2r}). \end{cases} \quad (25)$$

As is readily seen, system (25) can be efficiently solved by the Berlekamp-Massey algorithm (cf. [4]) at the computational complexity of $O(k^2)$. We are now ready to discuss the specifics of the identification procedure:

Transition Fault Identification Procedure

1. The first stage utilizes (16) (or (18)) to compute sequentially $\mathcal{F}_1(s_1), \mathcal{F}_2(s_1, s_2), \dots, \mathcal{F}_k(s_1, s_2, \dots, s_k)$; this stage requires $O(k^2)$ operations.

2. For $r = \lfloor (k - |\tau|) / 2 \rfloor$, apply the Berlekamp-Massey algorithm to solve for $\Lambda_l(x_{\tau+r+1}, \dots, x_{\tau+2r})$, $l = 1, 2, \dots, r$; this can be accomplished with $O(r^2)$ operations.
3. Substitute the obtained $\Lambda_l(x_{\tau+r+1}, \dots, x_{\tau+2r})$, $l = 1, 2, \dots, r$, into Eq. (17) and, using Proposition 1, obtain the solution of $x_{\tau+r+1}, \dots, x_{\tau+2r}$ by testing $1, 2, \dots, m$, one by one; this is successful only if the solution exists and takes $O(mr)$ steps.
4. With known $x_{\tau+r+1}, \dots, x_{\tau+2r}$, we obtain $\mathcal{S}_i(x_1, x_2, \dots, x_{\tau+r}) = s_i - \sum_{l=\tau+r+1}^{\tau+2r} x_l^i$, $i = 1, 2, \dots, \tau + r$, with $O(r(\tau + r))$ steps; we can then follow (16) to get $\Lambda_1(x_1, \dots, x_{\tau+r}), \dots, \Lambda_{\tau+r}(x_1, \dots, x_{\tau+r})$ in $O((\tau + r)^2)$ steps.
5. Substitute the values obtained in (iii) into (17) and follow Proposition 1 to get the solution of $x_1, \dots, x_{\tau+r}$, again by testing $1, 2, \dots, m$ one by one (this is successful only if the solution exists).

Clearly, the overall decoding/identification complexity of the procedure described above is $O(k^2m)$. Note that, for $k = 1$, the first row of matrix \mathbf{D} in (20) is redundant; thus, we only need

$$\mathbf{D}_1 \triangleq (1, 2, \dots, m), \quad k = 1. \quad (26)$$

For $k = 2$, the first row is again redundant. This is justified by the fact that the equation array

$$\begin{cases} x_1 \equiv x_2 + x_3 \\ x_1^2 \equiv x_2^2 + x_3^2 \end{cases}$$

does not have a nonzero (non-trivial) solution² in $\text{GF}(p)$; thus, there is no confusion on the number of faults. Therefore, for $k = 2$, we can use the following matrix \mathbf{D} :

$$\mathbf{D}_2 \triangleq \begin{pmatrix} 1 & 2 & 3 & \dots & m \\ 1 & 2^2 \bmod p & 3^2 \bmod p & \dots & m^2 \bmod p \end{pmatrix}, \quad k = 2. \quad (27)$$

Note that there is a variety of ways in which one can introduce additional places in order to capture pre-condition or post-condition transition faults. For example, a simple (but expensive) approach would be to do the following: (i) add one additional place for each transition; (ii) connect each additional place with its corresponding transition with one input and one output arc, both with unit weight; (iii) initialize each additional place with k tokens. Clearly, if the number of tokens in an additional place is $k' \neq k$, then the corresponding transition has suffered a fault; in fact, if $k' > k$ then the transition has suffered $k' - k$ pre-condition faults, whereas if $k' < k$ then the transition has suffered $k - k'$ post-condition faults. Clearly, the task of the fault identification mechanism under this scenario is very simple, however, the amount of redundancy added in the system is prohibitively expensive, particularly for systems with a large number of transitions. Another significant disadvantage of this approach is that a single *place* fault in one of the additional places will lead to an erroneous diagnosis. The next section discusses how one can construct redundant embeddings that allow the systematic identification of place faults (assuming no transition faults); we then discuss more elaborate approaches which can be used to handle combinations of transition and place faults.

²We observe that $x_2^2 + x_3^2 \equiv (x_2 + x_3)^2$ yields $2x_2x_3 \equiv 0$ which implies that either $x_2 \equiv 0$ or $x_3 \equiv 0$.

B Identification of place faults

We now focus on designing matrix $\mathbf{P} \triangleq [-\mathbf{C} \ \mathbf{I}_d]$ so as to identify up to k place faults with some d additional places (assuming no transition faults). This is equivalent to choosing the integer entries of the $d \times n$ matrix \mathbf{C} so that the syndrome at time epoch N , which was shown in Section II.D to be equal to

$$\mathbf{s}[N] = \mathbf{P}\mathbf{e}_P,$$

has a unique solution $\mathbf{e}_P \in \mathbb{Z}^\eta$ with at most k nonzero entries. This is reminiscent of a decoding problem in which a message of length n is extended to a codeword of length $n + d = \eta$ in a way that allows the correction of a maximum of k errors in the codeword. We will pursue this connection more explicitly for a special class of linear block codes, namely Reed-Solomon codes in $\text{GF}(p)$ [4, 5]. For simplicity, for the remainder of this section all operations are defined in $\text{GF}(p)$, unless stated otherwise. Let α be a primitive element in $\text{GF}(p)$, i.e., an element such that $\{1, \alpha^1, \alpha^2, \dots, \alpha^{p-2}\} = \{1, 2, 3, \dots, p-1\}$. The parity check matrix of a Reed-Solomon code is defined in $\text{GF}(p)$ as

$$\mathbf{H}_t = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 & \dots & \alpha^{p-2} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \dots & \alpha^{2(p-2)} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \dots & \alpha^{3(p-2)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^t & \alpha^{2t} & \alpha^{3t} & \dots & \alpha^{t(p-2)} \end{pmatrix}.$$

The set of codewords in the above code is formed by all $(p-1)$ -dimensional vectors $\mathbf{c} = [c_1 \ c_2 \ \dots \ c_{p-1}]^T$ that satisfy $\mathbf{H}_t \mathbf{c} = \mathbf{0}$. This particular Reed-Solomon code has minimum Hamming distance³ $t+1$, i.e., the Hamming distance between any two codewords \mathbf{c} and $\tilde{\mathbf{c}}$ satisfies $d_H(\mathbf{c}, \tilde{\mathbf{c}}) \geq t+1$, which implies that the corruption of up to $\lfloor \frac{t}{2} \rfloor$ entries in a codeword is correctable. Moreover, the solution of

$$\mathbf{s} = \mathbf{H}_t \mathbf{e}$$

for a vector \mathbf{e} with at most $\lfloor \frac{t}{2} \rfloor$ nonzero entries in $\text{GF}(p)$ can be achieved efficiently with computational complexity $O(tp)$ using the Berlekamp-Massey algorithm.

To transform our problem to the decoding of a Reed-Solomon code, we make (for now) the reasonable assumption that the number of erroneous tokens in a place is bounded. More specifically, we assume that e_{P_i} , the erroneous number of tokens in place P_i , is within the interval $[-\frac{p-1}{2}, \frac{p-1}{2}]$, where p is a large enough prime number. Under this assumption, e_{P_i} can be interpreted to fall within $[0, p-1]$ in $\text{GF}(p)$ (by naturally mapping e_{P_i} to $e_{P_i} \pmod{p}$). This artificially imposed requirement enables us to transform our identification problem to the decoding of Reed-Solomon codes. More specifically, what we need to do is to transform the problem in (12) (in which \mathbf{P} is a systematic parity check matrix with integer entries) to the form $\mathbf{s} = \mathbf{H}_t \mathbf{e}$ discussed above. Note that one can always write

$$\mathbf{H}_t = \Phi[-\tilde{\mathbf{C}} \ \mathbf{I}_t],$$

³The Hamming distance $d_H(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x} = (x_1, x_2, \dots, x_\eta)$ and $\mathbf{y} = (y_1, y_2, \dots, y_\eta)$ with elements in $\text{GF}(p)$ is the number of entries at which \mathbf{x} and \mathbf{y} differ [4, 5].

where matrix Φ denotes the last t columns of \mathbf{H}_t (because Φ forms a nonsingular Vandermonde matrix [4, 5]). Clearly, if we set $\mathbf{P} = [-\tilde{\mathbf{C}} \ \mathbf{I}_t] = \Phi^{-1}\mathbf{H}_t$ (by interpreting the entries as integer entries in $[0, p-1]$, i.e., by inverting the natural mapping mentioned above), then we can easily reduce (12) to a decoding problem. By taking both sides of (12) modulo p we obtain

$$\mathbf{s}_P[N] = \mathbf{P}\mathbf{q}_f[N] = [-\tilde{\mathbf{C}} \ \mathbf{I}_t]\mathbf{e}_P \Rightarrow \mathbf{s}_P[N] \equiv \Phi^{-1}\mathbf{H}_t\mathbf{e}_P \Rightarrow \Phi\mathbf{s}_P[N] \equiv \mathbf{H}_t\mathbf{e}_P,$$

where the symbol “ \equiv ” denotes equality modulo p . Clearly, with this choice of matrix \mathbf{P} , we can identify k place faults using $d = 2k$ additional places. The construction can be split into the following two cases:

(i) When $\eta = p - 1$ (that is, when $\eta + 1$ is prime), we can set

$$\mathbf{P}_{2k} = \Phi^{-1}\mathbf{H}_{2k}, \tag{28}$$

where Φ denotes the matrix composed of the last $2k$ columns of \mathbf{H}_{2k} . To efficiently identify place faults, we pre-process the syndrome $\mathbf{s}_P[N] = \mathbf{P}_{2k}\mathbf{e}_P$ by (left) multiplying by matrix Φ , i.e., we obtain the modified syndrome

$$\mathbf{s}'_P[N] \triangleq \Phi\mathbf{s}_P[N],$$

which satisfies

$$\mathbf{s}'_P[N] \equiv \mathbf{H}_{2k}\mathbf{e}_P$$

and can be solved efficiently for \mathbf{e}_P using the Berlekamp-Massey algorithm.

(ii) When $\eta < p - 1$, we can extend \mathbf{e}_P to a $(p - 1)$ -dimensional vector by appending 0's (i.e., $\tilde{\mathbf{e}}_P = [\mathbf{e}_P^T \ 0 \ 0 \ \dots \ 0]^T$) and then follow the same steps as in the first case. An alternative understanding is to define the *punctured* parity check matrix $\tilde{\mathbf{H}}_{2k}$ to be the first η columns of \mathbf{H}_{2k} , such that

$$\tilde{\mathbf{H}}_{2k} = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 & \dots & \alpha^{\eta-1} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \dots & \alpha^{2(\eta-1)} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \dots & \alpha^{3(\eta-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{2k} & \alpha^{4k} & \alpha^{6k} & \dots & \alpha^{2k(\eta-1)} \end{pmatrix}. \tag{29}$$

The parity check matrix \mathbf{P}_{2k} can be defined as a systematic version of $\tilde{\mathbf{H}}_{2k}$ in the form

$$\mathbf{P}_{2k} = \Phi^{-1}\tilde{\mathbf{H}}_{2k}, \tag{30}$$

where Φ denotes the matrix composed of the last $2k$ columns of $\tilde{\mathbf{H}}_{2k}$. As in case (i), we pre-process the syndrome and apply the Berlekamp-Massey algorithm on the modified syndrome to identify up to k faults. In this case, however, the Berlekamp-Massey algorithm can be simplified: the algorithm reduces the system of equations to a polynomial equation and tests for all possible solutions; in our case, since we only need to test for $1, \alpha, \alpha^2, \dots, \alpha^{\eta-1}$, the operational complexity is $O(k\eta)$ as opposed to $O(kp)$.

In summary, by employing Reed-Solomon codes over $\text{GF}(p)$, we can use $2k$ additional places to identify k place

faults (where in each affected place the number of tokens that are added or subtracted does not exceed $(p-1)/2$). The complexity of the identification procedure is $O(k\eta)$ operations. Note that one can easily relax the assumption that the number of erroneous tokens in each place is bounded. For example, the number of erroneous tokens in each place P_i can always be decomposed in the form

$$e_{P_i} = k_i^{(0)} + pk_i^{(1)} + p^2k_i^{(2)} + \dots$$

(where $k_i^{(\ell)}$ for $\ell = 0, 1, 2, \dots$ satisfy $0 \leq k_i^{(\ell)} \leq p-1$), so that we can express the vector \mathbf{e}_P as

$$\mathbf{e}_P = \mathbf{e}_P^{(0)} + p\mathbf{e}_P^{(1)} + p^2\mathbf{e}_P^{(2)} + \dots.$$

Clearly, $\mathbf{e}_P^{(0)}$ can be calculated as before by applying the Berlekamp-Massey algorithm on

$$\mathbf{s}_P^{(0)}[N] \equiv \Phi \mathbf{s}_P[N] \equiv \mathbf{H}_{2k} \mathbf{e}_P^{(0)} \pmod{p}.$$

Once $\mathbf{e}_P^{(0)} \pmod{p}$ is obtained, we can calculate

$$\mathbf{s}_P^{(1)}[N] \equiv \Phi \left[\frac{1}{p} \left(\mathbf{s}_P[N] - \mathbf{P}_{2k} \mathbf{e}_P^{(0)} \right) \right] \equiv \mathbf{H}_{2k} \mathbf{e}_P^{(1)} \pmod{p}$$

and use it in the same way as we used $\mathbf{s}_P^{(0)}[N]$ to calculate $\mathbf{e}_P^{(1)}$. This process can be repeated to calculate $\mathbf{e}_P^{(2)}, \mathbf{e}_P^{(3)}, \dots$

C Simultaneous Identification of Transition and Place Faults

In this section we discuss the synthesis of an identification scheme for mixed transition and place faults. By combining Eqs. (10) and (12), and the analysis in Section II.D, we have the following fault syndrome at time epoch N :

$$\begin{aligned} \mathbf{s}[N] &\triangleq \mathbf{P} \mathbf{q}_f[N] = \mathbf{P}(\mathbf{q}_h[N] - \mathcal{B}^+ \mathbf{e}_T^+ + \mathcal{B}^- \mathbf{e}_T^- + \mathbf{e}_P) \\ &= \mathbf{D} \mathbf{e}_T + \mathbf{P} \mathbf{e}_P. \end{aligned} \tag{31}$$

So far, we have argued that by using a redundant Petri net embedding with $2k$ additional places we are able to identify *either* $2k-1$ transition faults *or* k place faults. In this section we show that with $2k$ additional places it is possible to *simultaneously* identify $2k-1$ transition faults *and* k place faults. As in the previous sections, we assume that no transition suffers simultaneous pre-condition and post-condition faults during the epoch interval $[1, N]$ and that the number of erroneous tokens added to (or subtracted from) each place does not exceed $+\frac{p-1}{2}$ (or $-\frac{p-1}{2}$).

Recall that the identification schemes for transition faults and place faults that we presented earlier were based on operations in $\text{GF}(p)$ (modulo p operations). To identify both types of faults simultaneously, the key idea is to incorporate regular integer operations into the design of matrices \mathbf{D} and \mathbf{C} , as well as in the identification procedure. For notational simplicity, we ignore the subscript $2k$ in the sequel. Let p be a prime number larger than both m and η , and let \mathbf{D} follow the design in (20) and \mathbf{C} be chosen such that $\mathbf{P} \triangleq [\mathbf{C} \quad \mathbf{I}] \pmod{p}$ satisfies (28)

or (30). Define \mathbf{D}^* and \mathbf{P}^* by

$$\mathbf{D}^* \triangleq -p \cdot \mathbf{D}, \quad (32)$$

$$\mathbf{C}^* \triangleq p \cdot \mathbf{1} - \mathbf{C}, \quad \mathbf{P}^* \triangleq [-\mathbf{C}^* \ \mathbf{I}] = [\mathbf{C} - p \cdot \mathbf{1} \ \mathbf{I}], \quad (33)$$

where $\mathbf{1}$ is a $2k \times n$ matrix with all entries being 1.

Note that the design in Eqs. (32) and (33) satisfies $\mathbf{C}^* > \mathbf{0}$, $\mathbf{D}^* < \mathbf{0}$ (element-wise). This guarantees that the marking of the additional places $\mathbf{C}^* \mathbf{q}_s[\cdot]$ is nonnegative and that the arc weights associated with the additional places (given by $\mathbf{C}^* \mathbf{B}^- - \mathbf{D}^*$ and $\mathbf{C}^* \mathbf{B}^+ - \mathbf{D}^*$) are nonnegative. It is possible, however, that after the occurrence of a fault some firings that are enabled in the original Petri net become disabled in the redundant Petri net embedding due to the (erroneous) marking of the additional places (this will be evident in our example in Section VI). Clearly, this is not an issue if the enabling and disabling of transitions is *not* influenced by the number of tokens in the additional places. Even when the additional places function as controllers (as in [29, 30] for instance), this problem can be avoided in straightforward ways (e.g., by adding a sufficiently large number of extra tokens to each additional place and ignoring this extra number of tokens when performing parity checks at the end of time epoch N).

We now address the identification procedure. Clearly, the syndrome $\mathbf{s}[N] \triangleq \mathbf{P}^* \mathbf{q}_f[N]$ at time epoch N satisfies

$$\mathbf{s}_P \equiv \mathbf{s}[N] \equiv [\mathbf{C} \ \mathbf{I}] \mathbf{e}_P \pmod{p}. \quad (34)$$

Left multiplying by Φ on both sides of (34), we obtain the modified syndrome

$$\mathbf{s}'_P \triangleq \Phi \mathbf{s}_P \equiv \Phi [\mathbf{C} \ \mathbf{I}] \mathbf{e}_P \equiv \mathbf{H} \mathbf{e}_P \pmod{p} \quad (35)$$

(recall that Φ is the matrix that transforms the parity check matrix of the given Reed-Solomon code to a systematic parity check matrix). When k or less place faults occur, they can be identified by the Berlekamp-Massey algorithm based on \mathbf{s}'_P . Once place faults have been successfully identified and \mathbf{e}_P has been obtained, we can compute

$$\mathbf{s}_T \triangleq (\mathbf{s}[N] - \mathbf{P}^* \mathbf{e}_P) / p = (\mathbf{D}^* / p) \mathbf{e}_T = -\mathbf{D} \mathbf{e}_T, \quad (36)$$

which immediately enables us to identify up to $2k - 1$ transition faults using the algorithm discussed in the previous section (note that the symbol “=” denotes integer equality). Overall, the identification of place faults requires $O(k\eta) = O(k^2 + kn)$ operations and the identification of transition faults requires $O(k^2m)$ operations; thus, the entire identification complexity is $O(k^2m) + O(k\eta) = O(k(m + n))$ operations.

Note that in the approach presented above the identification of transition and place faults is essentially separated. As a result, no matter how many transition faults occur, place faults are always identifiable, as long as no more than k place faults happen. Of course, there are several other ways to approach the problem. For example, if we let

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_T \\ \mathbf{e}_P \end{bmatrix},$$

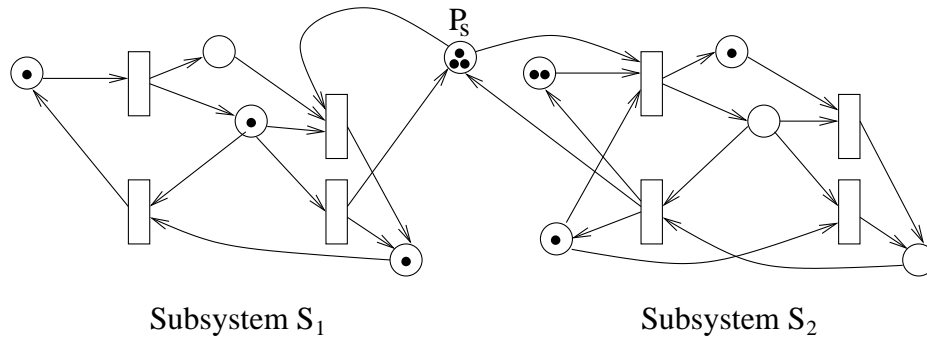


Figure 1: A system which can be conveniently decomposed into two interacting subsystems.

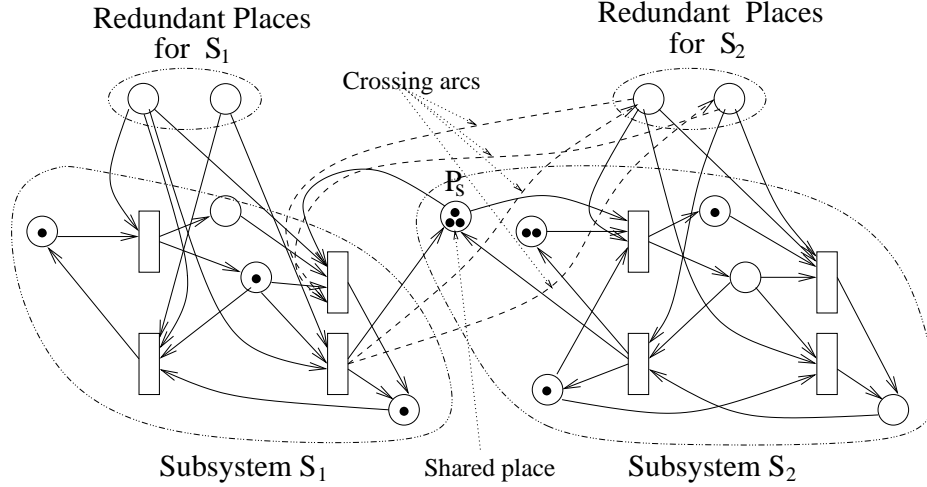


Figure 2: Distributed fault identification for the system depicted in Figure 1.

then the syndrome at time epoch N is given by

$$\begin{aligned} \mathbf{s}[N] &= \begin{bmatrix} \mathbf{D} & \mathbf{P} \end{bmatrix} \mathbf{e} \\ &= \begin{bmatrix} \mathbf{D} & -\mathbf{C} & \mathbf{I} \end{bmatrix} \mathbf{e}. \end{aligned}$$

Clearly, by defining $\mathbf{C}' = \begin{bmatrix} \mathbf{D} & -\mathbf{C} \end{bmatrix}$, we see that we can use the same approach as in Section IV.B to choose matrix \mathbf{C}' (and thus matrices \mathbf{D} and \mathbf{C}). The difference is that the prime number p has to satisfy $p > n + m$ and that this scheme uses $2k$ additional places to detect and identify a total of k faults at Petri net transitions and/or places. On the other hand, in this scenario multiple faults on the same transition are only counted as one fault.

V DISTRIBUTED FAULT IDENTIFICATION

The design of the centralized fault identification scheme that we outlined in the previous section may be difficult to implement in large systems due to communication constraints or overheads. In order to avoid these problems and develop scalable fault identification schemes, we present in this section a distributed fault identification scheme.

For simplicity, we first consider a Petri net which can be conveniently decomposed into two interacting sub-

systems,⁴ such as the one shown in Figure 1, where place P_s is shared by subsystems \mathcal{S}_1 and \mathcal{S}_2 . For the time being, we assign the shared place to subsystem \mathcal{S}_2 and design a redundant Petri net embedding for each of the two subsystems *separately*, utilizing our development in Section IV (refer to Figure 2). Clearly, when a transition associated with the shared place P_s fires in subsystem \mathcal{S}_1 , the fault identification mechanism for subsystem \mathcal{S}_2 will treat the result as a place fault in the shared place. One way to overcome this limitation is to *compensate* the fault syndrome of subsystem \mathcal{S}_2 by appropriately adjusting the number of tokens in the additional places of \mathcal{S}_2 , i.e., by adding arcs between the transitions of \mathcal{S}_1 that are associated with P_s and the additional places of the redundant embedding for subsystem \mathcal{S}_2 . More specifically, we need to consider two cases:

(i) When the firing of a transition in subsystem \mathcal{S}_1 *generates* tokens in the places of subsystem \mathcal{S}_2 , the fault syndrome seen by the fault identification mechanism of \mathcal{S}_2 is $\mathbf{s}_P^{(2)} = -\mathbf{C}^{(2)*} \mathbf{e}_P^{(2)}$ (superscript (i) is used to denote subsystem \mathcal{S}_i and superscript $*$ follows the design in Section IV.C). Here, $\mathbf{e}_P^{(2)}$ is the additive vector that describes how the number of tokens in the (original) places of subsystem \mathcal{S}_2 has been corrupted (i.e., $\mathbf{e}_P^{(2)}$ is of dimension n_2 as opposed to η_2). We can account for the erroneous syndrome $\mathbf{s}_P^{(2)}$ if we add crossing (output) arcs from the transition of \mathcal{S}_1 to the additional places of the redundant embedding for \mathcal{S}_2 and choose the arc weights to be $\mathbf{w}^{(2)+} = \mathbf{C}^{(2)*} \mathbf{e}_P^{(2)}$ (note that $\mathbf{e}_P^{(2)}$ has nonnegative entries, resulting in a nonnegative vector $\mathbf{w}^{(2)+}$). With this choice, the resulting syndrome in \mathcal{S}_2 is zero, i.e.,

$$\begin{aligned} \mathbf{P} \mathbf{q}_f^{(2)}[t] &= [-\mathbf{C}^{(2)*} \quad \mathbf{I}_d] \left(\mathbf{q}_h^{(2)}[t] + \begin{bmatrix} \mathbf{e}_P^{(2)} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{w}^{(2)+} \end{bmatrix} \right) \\ &= -\mathbf{C}^{(2)*} \mathbf{e}_P^{(2)} + \mathbf{w}^{(2)+} \\ &= \mathbf{0}. \end{aligned} \tag{37}$$

(ii) When the firing of a transition in subsystem \mathcal{S}_1 *consumes* tokens from places of subsystem \mathcal{S}_2 , causing the fault syndrome to be $\mathbf{s}_P^{(2)} = -\mathbf{C}^{(2)*} \mathbf{e}_P^{(2)}$ in \mathcal{S}_2 , then we need to add crossing (input) arcs from the additional places of the redundant embedding for \mathcal{S}_2 to this transition and choose the arc weights to be $\mathbf{w}^{(2)-} = -\mathbf{C}^{(2)*} \mathbf{e}_P^{(2)}$ (note that $\mathbf{e}_P^{(2)}$ has nonpositive entries, resulting in a nonnegative vector $\mathbf{w}^{(2)-}$). We note that this case may lead to negative values for the total number of tokens in the additional places. If this needs to be avoided, we can artificially add a sufficiently large number of tokens to each additional place and ignore this extra number of tokens when performing parity checks (see the discussion in Section IV.C).

We highlight that this compensation method ensures that transition faults in a particular subsystem do not affect fault detection and identification in other subsystems. For instance, in case (i), if a post-condition fault occurs to the transition of \mathcal{S}_1 that is connected to P_s (in the forward direction), then no tokens are deposited at the shared places or the additional places in \mathcal{S}_2 . Likewise, in case (ii), if a pre-condition fault occurs to the transition in \mathcal{S}_1 that is connected to P_s (in the backward direction), there is no effect on \mathcal{S}_2 . Finally, we note that the value of the prime number p , and accordingly the arc weights and the number of tokens in the additional places, can be significantly reduced with modular designs because they can be confined to a subsystem rather than the entire system.

More generally, when multiple subsystems exist in the modular decomposition of a large system, we can imple-

⁴This can be done in many ways; see, for example, [34, 35] and references therein.

ment the above compensation approach in a pairwise fashion. More specifically, given a decomposition of a Petri net into subsystems, we can follow the above approach to build crossing arcs for each pair of subsystems that share one or more places. This is described in the following procedure.

Construction of Distributed Redundant Embeddings:

For each subsystem \mathcal{S}_i , perform the following:

1. Establish an appropriate redundant embedding for \mathcal{S}_i .
2. For each subsystem \mathcal{S}_j , $j \neq i$, with transitions that connect to or from place(s) in \mathcal{S}_i , do:
 - (i) For each transition in subsystem \mathcal{S}_j with output arc(s) to the shared place(s), build output crossing arcs from this transition to the additional places of the redundant embedding for \mathcal{S}_i .
 - (ii) For each transition in subsystem \mathcal{S}_j with input arc(s) from the shared place(s), build input crossing arcs from the additional places of the redundant embedding for \mathcal{S}_i to this transition.

Note that if one is interested in designing distributed monitoring schemes that minimize the number of crossing arcs, then one can employ several strategies to allocate shared places to subsystems. For example, if all redundant embeddings have the same number of additional places, then, in order to minimize the number of crossing arcs between different subsystems, a shared place should be allocated to the subsystem which has the most arcs connected to that place (each such arc is connected to a transition which is in turn connected via crossing arcs with all additional places). If the embeddings of different subsystems have different number of places, then the shared place should be allocated to the subsystem that is associated with the largest product of number of additional places multiplied by the number of arcs linked to the shared place.

The proposed distributed fault identification scheme exhibits great flexibility in terms of scalability. More specifically, when a new subsystem \mathcal{S}_N is added to an existing system \mathcal{S} consisting of subsystems $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{N-1}\}$, we can follow the procedure below to upgrade the distributed redundant embeddings.

System Upgrade:

1. Establish an appropriate redundant embedding for subsystem \mathcal{S}_N .
2. For each subsystem \mathcal{S}_i , $1 \leq i \leq N - 1$, with transitions that connect to place(s) in \mathcal{S}_N , do:
 - (i) For each transition in subsystem \mathcal{S}_i with output arc(s) to place(s) in \mathcal{S}_N , build output crossing arcs from this transition to the additional places of the redundant embedding for \mathcal{S}_N .
 - (ii) For each transition in subsystem \mathcal{S}_i with input arc(s) from place(s) in \mathcal{S}_N , build input crossing arcs from the additional places of the redundant embedding for \mathcal{S}_N to this transition.
3. For each subsystem \mathcal{S}_i which contains place(s) connecting to transitions in \mathcal{S}_N , do:
 - (i) For each transition in subsystem \mathcal{S}_N with output arc(s) to the shared place(s), build output crossing arcs from this transition to the additional places of the redundant embedding for \mathcal{S}_i .
 - (ii) For each transition in subsystem \mathcal{S}_N with input arc(s) from the shared place(s), build input crossing arcs from the additional places of the redundant embedding for \mathcal{S}_N to this transition.

and the arc matrices \mathbf{B}^+ and \mathbf{B}^- in Eq. (1) are given by

$$\mathbf{B}^+ = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}^- = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Centralized Fault Identification: We first describe a centralized fault identification scheme in which we aim to simultaneously detect and identify two transition faults and one place fault. We assume that the number of erroneous tokens in any place is bounded by $[-5, 5]$. From our development in Section III, we need two additional places ($d = 2$) and, since the smallest prime number that is greater than both $m = 12$ and $\eta = 18 + 2 = 20$ is 23, we set $p = 23$. Following the construction in Section IV.C, we choose \mathbf{D}^* so that

$$\begin{aligned} \mathbf{D}^* &= -23 \times \left[\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1^2 & 2^2 & 3^2 & 4^2 & 5^2 & 6^2 & 7^2 & 8^2 & 9^2 & 10^2 & 11^2 & 12^2 \end{pmatrix} \pmod{23} \right] \\ &= \begin{pmatrix} -23 & -46 & -69 & -92 & -115 & -138 & -161 & -184 & -207 & -230 & -253 & -276 \\ -23 & -92 & -207 & -368 & -46 & -299 & -69 & -414 & -276 & -184 & -138 & -138 \end{pmatrix}. \end{aligned}$$

Since 5 is a primitive element in $\text{GF}(23)$, the punctured parity check matrix $\tilde{\mathbf{H}}$ of the Reed-Solomon code can be chosen (modulo 23) as

$$\begin{aligned} \tilde{\mathbf{H}} &\equiv \begin{pmatrix} 1 & 5 & 5^2 & 5^3 & 5^4 & 5^5 & 5^6 & 5^7 & 5^8 & 5^9 & 5^{10} & 5^{11} & 5^{12} & 5^{13} & 5^{14} & 5^{15} & 5^{16} & 5^{17} & 5^{18} & 5^{19} \\ 1 & 5^2 & 5^4 & 5^6 & 5^8 & 5^{10} & 5^{12} & 5^{14} & 5^{16} & 5^{18} & 5^{20} & 5^{22} & 5^{24} & 5^{26} & 5^{28} & 5^{30} & 5^{32} & 5^{34} & 5^{36} & 5^{38} \end{pmatrix} \\ &\quad \pmod{23} \\ &\equiv \begin{pmatrix} 1 & 5 & 2 & 10 & 4 & 20 & 8 & 17 & 16 & 11 & 9 & 22 & 18 & 21 & 13 & 19 & 3 & 15 & 6 & 7 \\ 1 & 2 & 4 & 8 & 16 & 9 & 18 & 13 & 3 & 6 & 12 & 1 & 2 & 4 & 8 & 16 & 9 & 18 & 13 & 3 \end{pmatrix} \\ &\equiv \underbrace{\begin{pmatrix} 6 & 7 \\ 13 & 3 \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} 1 & 17 & 17 & 18 & 2 & 18 & 14 & 10 & 22 & 8 & 20 & 14 & 13 & 20 & 10 & 8 & 2 & 3 & 1 & 0 \\ 19 & 19 & 12 & 9 & 12 & 17 & 22 & 7 & 13 & 21 & 17 & 1 & 21 & 22 & 13 & 9 & 2 & 16 & 0 & 1 \end{pmatrix}}_{[C \ I]}. \end{aligned}$$

According to our design rule in Eq. (33), we have

$$\begin{aligned} \mathbf{C}^* &= 23 \cdot \mathbf{1} - \begin{pmatrix} 1 & 17 & 17 & 18 & 2 & 18 & 14 & 10 & 22 & 8 & 20 & 14 & 13 & 20 & 10 & 8 & 2 & 3 \\ 19 & 19 & 12 & 9 & 12 & 17 & 22 & 7 & 13 & 21 & 17 & 1 & 21 & 22 & 13 & 9 & 2 & 16 \end{pmatrix} \\ &= \begin{pmatrix} 22 & 6 & 6 & 5 & 21 & 5 & 9 & 13 & 1 & 15 & 3 & 9 & 10 & 3 & 13 & 15 & 21 & 20 \\ 4 & 4 & 11 & 14 & 11 & 6 & 1 & 16 & 10 & 2 & 6 & 22 & 2 & 1 & 10 & 14 & 21 & 7 \end{pmatrix}. \end{aligned}$$

Thus, the arc weights to (from) the additional places from (to) the transitions in the original system (shown in Figure 4 with dotted arcs) are given by

$$\mathbf{C}^* \mathbf{B}^+ - \mathbf{D}^* = \begin{pmatrix} 29 & 52 & 123 & 114 & 120 & 147 & 219 & 205 & 222 & 233 & 306 & 277 \\ 27 & 103 & 266 & 372 & 52 & 300 & 108 & 425 & 278 & 190 & 189 & 148 \end{pmatrix}$$

and

$$\mathbf{C}^* \mathbf{B}^- - \mathbf{D}^* = \begin{pmatrix} 70 & 73 & 75 & 97 & 159 & 158 & 170 & 197 & 242 & 265 & 256 & 285 \\ 43 & 117 & 218 & 382 & 65 & 319 & 70 & 430 & 317 & 193 & 144 & 160 \end{pmatrix}.$$

Furthermore, the initial marking of the overall system is

$$\begin{aligned} \mathbf{q}_h[0] &= \begin{bmatrix} \mathbf{I}_{18} \\ \mathbf{C}^* \end{bmatrix} \mathbf{q}_s[0] \\ &= (1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 53\ 36)^T. \end{aligned}$$

(Note that the two additional places are initialized so that the encoded state $\mathbf{q}_h[0]$ satisfies the parity check $[-\mathbf{C}^* \ \mathbf{I}_2] \mathbf{q}_h[0] = \mathbf{0}$ in $\text{GF}(23)$.)

According to our choice of \mathbf{D}^* and \mathbf{C}^* , the system is expected to simultaneously detect and identify two transition faults and one place fault. We now check this capability by following a specific example. We assume that the applied firing sequence is $T_7, T_1, T_2, T_8, T_3, T_9$, and that the following faults occur: a pre-condition fault in transition T_2 (during time epoch 3), a place fault that corrupts the number of tokens in P_7 by +2 (during time epoch 5), and a post-condition fault in transition T_9 (during time epoch 6). The sequence of markings is given by

$$\begin{aligned} T_7 : \mathbf{q}_f[1] &= (1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 102\ 74)^T, & \text{Fault-free} \\ T_1 : \mathbf{q}_f[2] &= (0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 61\ 58)^T, & \text{Fault-free} \\ T_2 : \mathbf{q}_f[3] &= (0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 113\ 161)^T, & \text{Pre-condition fault in } T_2 \\ T_8 : \mathbf{q}_f[4] &= (0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 121\ 156)^T, & \text{Fault-free} \\ T_3 : \mathbf{q}_f[5] &= (0\ 1\ 0\ 1\ 1\ 1\ 0\ 2\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 2\ 1\ 169\ 204)^T, & P_7 \text{ corrupted by } +2 \\ T_9 : \mathbf{q}_f[6] &= (0\ 1\ 0\ 1\ 1\ 1\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ -73\ -113)^T, & \text{Post-condition fault in } T_9 \end{aligned}$$

where the indication on the right refers only to fault events during the corresponding time epoch. The resulting syndrome at time epoch $t = 6$ is

$$\mathbf{s}[6] = [-\mathbf{C}^* \ \mathbf{I}_2] \mathbf{q}_f[6] = \begin{pmatrix} -179 \\ -186 \end{pmatrix} \equiv \begin{pmatrix} 5 \\ 21 \end{pmatrix} \pmod{23}.$$

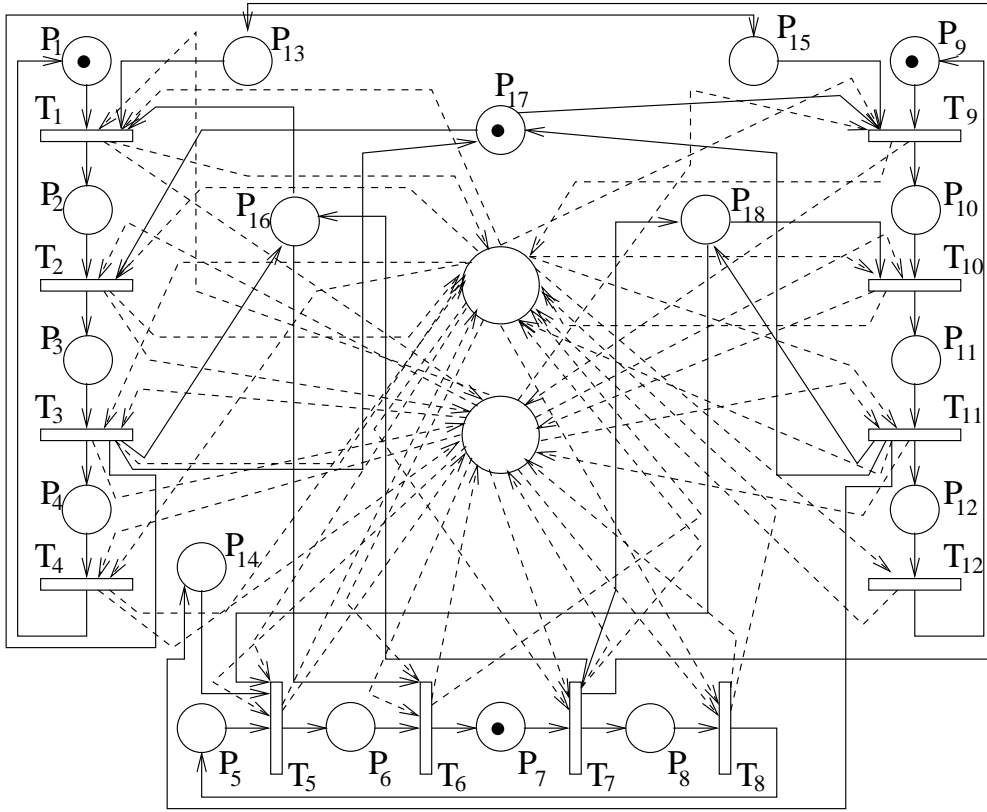


Figure 4: Centralized fault detection and identification for the manufacturing system of Figure 3 using a redundant Petri net embedding.

We proceed to identify the faults. First, by left multiplying by Φ , we obtain the modified place fault syndrome

$$\mathbf{s}'_P = \Phi \mathbf{s}[6] \pmod{23} \equiv \begin{pmatrix} 6 & 7 \\ 13 & 3 \end{pmatrix} \begin{pmatrix} 5 \\ 21 \end{pmatrix} = \begin{pmatrix} 16 \\ 13 \end{pmatrix}.$$

By inspecting the punctured parity check matrix $\tilde{\mathbf{H}}$, we easily identify place P_7 as faulty with the erroneous number of tokens being +2 (the syndrome \mathbf{s}'_P is equal to twice the seventh column of matrix $\tilde{\mathbf{H}}$). This is consistent with the faults that took place during the operation of the system.

Once place faults have been identified, we utilize (36) to obtain

$$\mathbf{De}_T = -(\mathbf{s}[6] - \mathbf{P}^* \mathbf{e}_P) / 23 = \begin{pmatrix} 7 \\ 8 \end{pmatrix}.$$

We note that the above syndrome does not coincide with any column of $\pm \mathbf{D}$, so there must be two transition faults (if identifiable). We first consider the case of both faults undergoing pre-condition faults, which results in the following equation array:

$$\begin{cases} -x_1 - x_2 \equiv 7, \\ -x_1^2 - x_2^2 \equiv 8. \end{cases}$$

By simple calculation, we obtain $\Lambda_1(x_1, x_2) \equiv -7$ and $\Lambda_2(x_1, x_2) \equiv 11$; the corresponding polynomial is

$$x^2 - 7x + 11 = 0$$

but its roots are not proper solutions. Similarly, we eliminate the possibility of both faults being post-condition faults. We then try the case of a pre-condition fault and a post-condition fault, which translates to solving

$$\begin{cases} x_1 - x_2 \equiv 7, \\ x_1^2 - x_2^2 \equiv 8. \end{cases}$$

These equations are easily shown to have a unique solution with $x_1 = 9$ and $x_2 = 2$. Therefore, we conclude that transition T_2 suffered a pre-condition fault and transition T_9 suffered a post-condition fault, which is consistent with the faults that took place during the operation of the system.

The simplified identification procedures presented above can be done systematically (using the procedure in Section IV.A for the identification of transition faults and the Berlekamp-Massey algorithm for the identification of place faults). Note that, during the operation of the redundant Petri net system in our example, the number of tokens in the additional places becomes negative (at time epoch 6). If the additional places function as controllers (as in [29, 30]), then this implies that the firing of transition T_9 during time epoch 6 would be inhibited. If this is the case, this problem can be avoided by adding a sufficiently large number of extra tokens to each additional place (and ignoring this extra number of tokens when performing parity checks). For example, if we add 500 tokens to each additional place, the state evolution of the redundant Petri net would appear as

$$\begin{aligned} T_7 : \mathbf{q}_f[1] &= (1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 500 + 102\ 500 + 74)^T, & \text{Fault-free} \\ T_1 : \mathbf{q}_f[2] &= (0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 500 + 61\ 500 + 58)^T, & \text{Fault-free} \\ T_2 : \mathbf{q}_f[3] &= (0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 500 + 113\ 500 + 161)^T, & \text{Pre-condition fault in } T_2 \\ T_8 : \mathbf{q}_f[4] &= (0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 500 + 121\ 500 + 156)^T, & \text{Fault-free} \\ T_3 : \mathbf{q}_f[5] &= (0\ 1\ 0\ 1\ 1\ 0\ 2\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 2\ 1\ 1\ 500 + 169\ 500 + 204)^T, & P_7 \text{ corrupted by } +2 \\ T_9 : \mathbf{q}_f[6] &= (0\ 1\ 0\ 1\ 1\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 500 - 73\ 500 - 113)^T, & \text{Post-condition fault in } T_9 \end{aligned}$$

and detection and identification would progress in exactly the same way as before, given that the 500 added tokens in the additional places are ignored when performing parity check calculations.

Distributed Fault Identification: We now discuss a distributed fault identification scheme for the Petri net in Figure 3. We find it convenient to decompose the system into three subsystems: subsystem \mathcal{S}_1 ($P_1, P_2, P_3, P_4, P_{13}, P_{16}; T_1, T_2, T_3, T_4$), subsystem \mathcal{S}_2 ($P_5, P_6, P_7, P_8, P_{14}, P_{18}; T_5, T_6, T_7, T_8$) and subsystem \mathcal{S}_3 ($P_9, P_{10}, P_{11}, P_{12}, P_{15}, P_{17}, T_9, T_{10}, T_{11}, T_{12}$). Since the sole difference between the three subsystems is their initial marking, we only discuss the design of the monitor for subsystem \mathcal{S}_1 with places P_{13} and P_{16} shared between subsystems \mathcal{S}_1 and \mathcal{S}_2 . As in the centralized case, our objective is to detect and identify up to one place fault and/or up to two transition faults so that the number of additional places is given by $d_i = 2$ for each subsystem \mathcal{S}_i . As before, we allow place faults to add an erroneous number of tokens within $[-5, 5]$. Since the smallest qualified prime number is $p = 11$ (note that in this

case $p = 11$ is greater than both $m_i = 4$ and $\eta_i = n_i + d_i = 8$ for each subsystem \mathcal{S}_i), we can adopt $\mathbf{D}^{(1)*}$ such that

$$\begin{aligned}\mathbf{D}^{(1)*} &= -11 \cdot \left[\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1^2 & 2^2 & 3^2 & 4^2 \end{pmatrix} \pmod{11} \right] \\ &= \begin{pmatrix} -11 & -22 & -33 & -44 \\ -11 & -44 & -99 & -55 \end{pmatrix}.\end{aligned}$$

Since 2 is a primitive element in $\text{GF}(11)$, the punctured parity check matrix $\tilde{\mathbf{H}}$ is given by

$$\begin{aligned}\tilde{\mathbf{H}} &= \begin{pmatrix} 1 & 2 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 & 2^7 \\ 1 & 2^2 & 2^4 & 2^6 & 2^8 & 2^{10} & 2^{12} & 2^{14} \end{pmatrix} \pmod{11} \\ &= \begin{pmatrix} 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 \\ 1 & 4 & 5 & 9 & 3 & 1 & 4 & 5 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 9 & 7 \\ 4 & 5 \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} 7 & 8 & 3 & 9 & 8 & 9 & 1 & 0 \\ 10 & 1 & 3 & 10 & 3 & 4 & 0 & 1 \end{pmatrix}}_{[\mathbf{C} \ \mathbf{I}]};\end{aligned}$$

thus, we have

$$\mathbf{C}^{(1)*} = 11 \cdot \mathbf{1} - \begin{pmatrix} 7 & 8 & 3 & 9 & 8 & 9 \\ 10 & 1 & 3 & 10 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 3 & 8 & 2 & 3 & 2 \\ 1 & 10 & 8 & 1 & 8 & 7 \end{pmatrix}$$

and the two directed weight matrices in the embedded subsystem are given by

$$\left[\frac{\mathbf{B}^{(1)+}}{\mathbf{C}^{(1)*}\mathbf{B}^{(1)+} - \mathbf{D}^{(1)*}} \right] = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 14 & 30 & 37 & 48 \\ 21 & 52 & 107 & 56 \end{pmatrix}, \quad \left[\frac{\mathbf{B}^{(1)-}}{\mathbf{C}^{(1)*}\mathbf{B}^{(1)-} - \mathbf{D}^{(1)*}} \right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 20 & 25 & 41 & 46 \\ 27 & 54 & 107 & 56 \end{pmatrix}.$$

If transition T_7 (in \mathcal{S}_2) fires, it produces one token in each of P_{13} and P_{16} . From the perspective of subsystem \mathcal{S}_1 , this is regarded as a (multiple) place fault. To balance the syndrome, the crossing arc weights from T_7 to the additional places of \mathcal{S}_1 are set to be

$$\mathbf{w}_7^{(1)+} = \mathbf{C}^{(1)*} \cdot (0 \ 0 \ 0 \ 0 \ 1 \ 1)^T = \begin{pmatrix} 5 \\ 15 \end{pmatrix}.$$

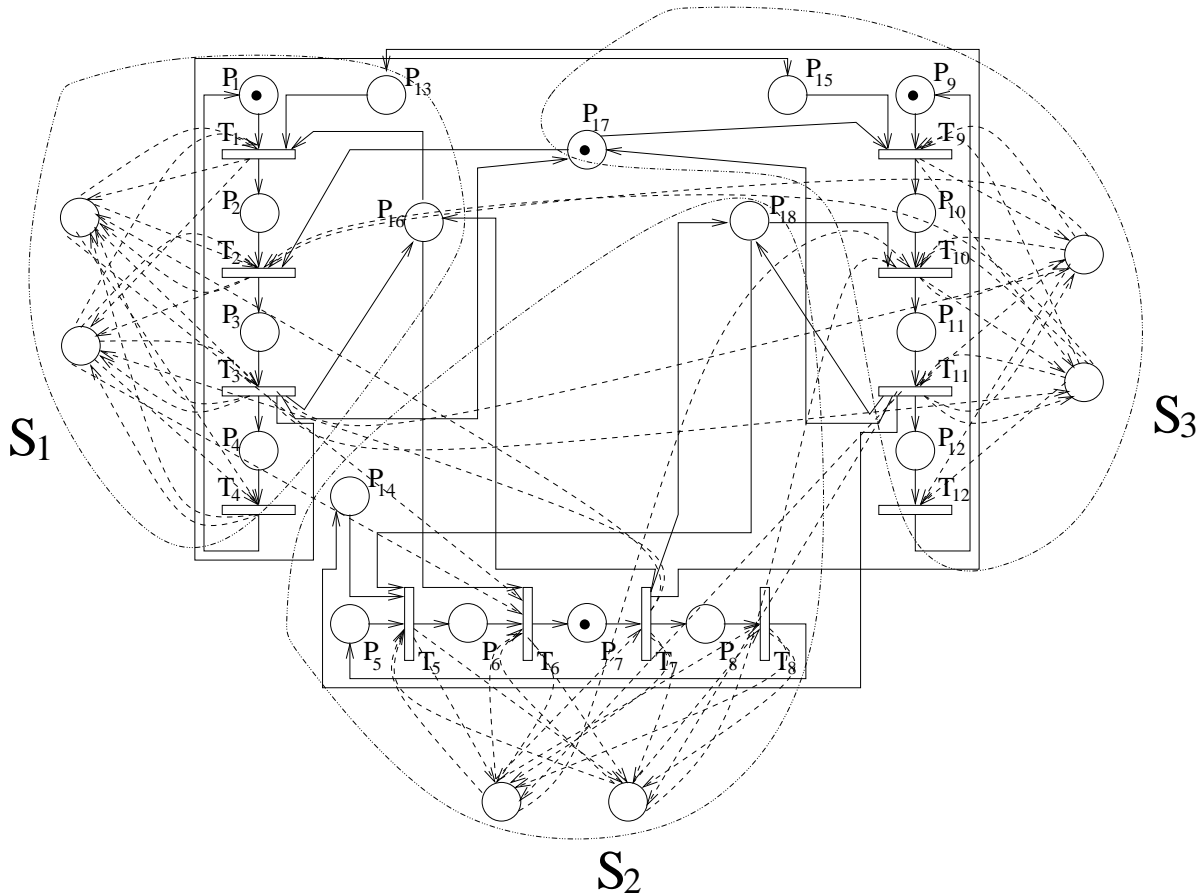


Figure 5: Distributed fault identification scheme for the system depicted in Figure 3.

If transition T_6 (in \mathcal{S}_2) fires, it consumes one token from place P_{16} . The syndrome for \mathcal{S}_1 can be balanced by setting the crossing arc weights from the additional places of \mathcal{S}_1 to transition T_6 to be

$$\mathbf{w}_6^{(1)-} = -\mathbf{C}^{(1)*} \cdot (0 \ 0 \ 0 \ 0 \ 0 \ -1)^T = \begin{pmatrix} 2 \\ 7 \end{pmatrix}.$$

The resulting distributed identification scheme (for all three subsystems and monitors) is illustrated in Figure 5, where P_{13} and P_{16} in \mathcal{S}_1 are shared with \mathcal{S}_2 , P_{14} and P_{18} in \mathcal{S}_2 are shared with \mathcal{S}_3 , and P_{15} and P_{17} in \mathcal{S}_3 are shared with \mathcal{S}_1 . As expected, the connecting structure of Figure 5 exhibits a more distributed nature than the structure of Figure 4.

To illustrate the capabilities of the resulting distributed identification scheme, we present a more complicated event sequence and show the identification procedure that occurs in subsystem \mathcal{S}_1 . The initial marking of the redundant embedding of subsystem \mathcal{S}_1 is

$$\begin{aligned} \mathbf{q}_h^{(1)}[0] &= \begin{bmatrix} \mathbf{I}_6 \\ \mathbf{C}^{(1)*} \end{bmatrix} \mathbf{q}_s^{(1)}[0] \\ &= (1 \ 0 \ 0 \ 0 \ 0 \ 4 \ 1)^T. \end{aligned}$$

Let the sequence of markings within subsystem \mathcal{S}_1 be as follows:

$$\begin{aligned}
T_7 : \mathbf{q}_f^{(1)}[1] &= (1\ 0\ 0\ 0\ 1\ 1\ 9\ 16)^T, \text{ Fault-free} \\
T_1 : \mathbf{q}_f^{(1)}[2] &= (0\ 1\ 0\ 0\ 0\ 0\ 3\ 10)^T, \text{ Fault-free} \\
T_2 : \mathbf{q}_f^{(1)}[3] &= (0\ 1\ 1\ 0\ 0\ 0\ 33\ 62)^T, \text{ Pre-condition fault in } T_2 \\
T_3 : \mathbf{q}_f^{(1)}[4] &= (0\ 1\ 0\ 3\ 0\ 1\ 29\ 62)^T, P_4 \text{ corrupted by } +2 \\
T_4 : \mathbf{q}_f^{(1)}[5] &= (0\ 1\ 0\ 2\ 0\ 1\ -17\ 6)^T, \text{ Post-condition fault in } T_4 \\
T_9 : \mathbf{q}_f^{(1)}[6] &= (0\ 1\ 0\ 2\ 0\ 1\ -17\ 6)^T, \text{ Fault-free} \\
T_{10} : \mathbf{q}_f^{(1)}[7] &= (0\ 1\ 0\ 2\ 0\ 1\ -17\ 6)^T, \text{ Pre-condition fault in } T_{10} \\
T_{11} : \mathbf{q}_f^{(1)}[8] &= (0\ 1\ 0\ 2\ 0\ 1\ -17\ 6)^T, \text{ Fault-free} \\
T_8 : \mathbf{q}_f^{(1)}[9] &= (0\ 1\ 0\ 2\ 0\ 1\ -17\ 6)^T, \text{ Fault-free} \\
T_5 : \mathbf{q}_f^{(1)}[10] &= (0\ 1\ 0\ 2\ 0\ 1\ -17\ 6)^T, \text{ Fault-free} \\
T_6 : \mathbf{q}_f^{(1)}[11] &= (0\ 1\ 0\ 2\ 0\ 0\ -19\ -1)^T, \text{ Fault-free} \\
T_7 : \mathbf{q}_f^{(1)}[12] &= (0\ 1\ 0\ 2\ 0\ 0\ -19\ -1)^T, \text{ Post-condition fault in } T_7
\end{aligned}$$

(again, if desired/necessary, we can add tokens to each additional place to ensure that the number of tokens in the additional places does not become negative and inhibit transitions that would otherwise be enabled in the original Petri net).

Let us assume that we perform a non-concurrent check at time epoch 12. We first compute the resulting syndrome

$$\mathbf{s}[12] = [-\mathbf{C}^{(1)*} \ \mathbf{I}_2] \mathbf{q}_f^{(1)}[12] = \begin{pmatrix} -26 \\ -13 \end{pmatrix} \equiv \begin{pmatrix} 7 \\ 9 \end{pmatrix}.$$

Thus, faults have been successfully detected. Left multiplying by Φ and taking the result modulo $p = 11$, we obtain the following parity check equation

$$\underbrace{\begin{pmatrix} 9 & 7 \\ 4 & 5 \end{pmatrix}}_{\Phi} \begin{pmatrix} 7 \\ 9 \end{pmatrix} \equiv \begin{pmatrix} 5 \\ 7 \end{pmatrix} \equiv \underbrace{\begin{pmatrix} 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 \\ 1 & 4 & 5 & 9 & 3 & 1 & 4 & 5 \end{pmatrix}}_{\tilde{\mathbf{H}}} \mathbf{e}_P.$$

By inspection, we obtain $\mathbf{e}_P = [0\ 0\ 0\ 2\ 0\ 0]^T$, which agrees with the faults that took place in the system.

From Eq. (36), the transition fault syndrome is given by

$$\mathbf{D}\mathbf{e}_T = -(\mathbf{s}[12] - \mathbf{P}^*\mathbf{e}_P)/11 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

and, since it does not coincide with any column of $\mathbf{D}^{(1)*}$, we conclude that there must be two transition faults (if identifiable). We first consider the case of both faults being post-condition faults, which results in the following equation array:

$$\begin{cases} x_1 + x_2 \equiv 2, \\ x_1^2 + x_2^2 \equiv 1. \end{cases}$$

By simple calculation, we obtain $\Lambda_1(x_1, x_2) \equiv 2$ and $\Lambda_2(x_1, x_2) \equiv 7$; the corresponding polynomial is

$$x^2 + 2x + 7 \equiv 0$$

and has roots 3 and 6. However, since 6 is not a valid index for a transition in this subsystem, this is an invalid solution. Similarly, we eliminate the possibility of both faults being pre-condition faults. We then try the case of a pre-condition fault and a post-condition fault, which results in the following equation array:

$$\begin{cases} x_1 - x_2 \equiv 2, \\ x_1^2 - x_2^2 \equiv 1. \end{cases}$$

This equation is easily shown to have a unique solution $x_1 = 4$ and $x_2 = 2$. Therefore, we conclude that transition T_2 suffered a pre-condition fault and transition T_4 suffered a post-condition fault, which is consistent with the faults that took place during the operation of the system.

Note that faults in other subsystems (namely the pre-condition fault in transition T_{10}) do not affect fault identification in subsystem \mathcal{S}_1 . Moreover, our design for subsystem \mathcal{S}_1 is shown to be immune to the post-condition fault in transition T_7 .

VII CONCLUSIONS AND FUTURE WORK

In this paper we have presented centralized and distributed fault identification schemes for discrete event systems that are described by Petri nets. Our setting assumes that system events (transition firings) are not directly observable but that the system state (marking) is periodically observable, and aims at capturing faults in both Petri net transitions and places. To achieve this, we introduce redundancy and construct a redundant Petri net embedding whose additional places encode information in a way that enables error detection and identification to be performed using algebraic decoding techniques. Our approach does not need to reconstruct the various possible state evolution paths associated with a given DES and has small identification overhead. More specifically, using $2k$ additional places (and the connections and tokens associated with them), the proposed scheme can simultaneously identify $2k - 1$ transition faults and k place faults that may take place in the system. The worst-case complexity of the fault identification procedure is $O(k(m + n))$ operations, where m and n are respectively the number of transitions and places in the given Petri net. The proposed fault identification scheme was also modified to accommodate distributivity in a way that requires little additional hardware.

Our current work focuses on understanding ways to introduce redundancy so that we minimize the number of additional connections (as opposed to the number of additional places). More generally, we are interested in developing low-complexity identification schemes that minimize cost functions associated with the additional places and/or connections, in both centralized and distributed settings. Another interesting future direction is to generalize the techniques introduced in this paper to settings where certain Petri net places and/or transitions are unobservable or partially observable.

References

- [1] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, 1998.
- [2] E. Y. Chow and A. S. Willsky, “Analytical redundancy and the design of robust failure detection systems,” *IEEE Trans. Automatic Control*, vol. 29, pp. 603–614, July 1984.
- [3] C. N. Hadjicostis and G. C. Verghese, “Monitoring discrete event systems using Petri net embeddings,” *Application and Theory of Petri Nets 1999 (Series Lecture Notes in Computer Science, vol. 1639)*, pp. 188–207, 1999.
- [4] R. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, Cambridge, UK, 2002.
- [5] S. B. Wicker, *Error Control Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [6] Y. Wu and C. N. Hadjicostis, “On solving composite power polynomial equations.” Submitted to *Mathematics of Computation*.
- [7] Y. Wu and C. N. Hadjicostis, “Non-concurrent fault identification in discrete event systems using encoded Petri net states,” in *Proc. of the 41st IEEE Conf. on Decision and Control*, pp. 4018–4023, 2002.
- [8] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Diagnosability of discrete-event systems,” *IEEE Trans. Automatic Control*, vol. 40, pp. 1555–1575, Sept. 1995.
- [9] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Failure diagnosis using discrete-event models,” *IEEE Trans. Control System Technology*, vol. 4, pp. 105–124, March 1996.
- [10] M. Sampath, S. Lafortune, and D. Teneketzis, “Active diagnosis of discrete-event systems,” *IEEE Trans. Automatic Control*, vol. 43, pp. 908–929, July 1998.
- [11] J. Shengbing and R. Kumar, “Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications,” in *Proc. of the 2002 American Control Conference*, pp. 128–133, 2002.
- [12] R. Debouk, S. Lafortune, and D. Teneketzis, “On an optimization problem in sensor selection for failure diagnosis,” in *Proc. of the 38th IEEE Conf. on Decision and Control*, pp. 4990–4995, 1999.
- [13] R. Debouk, S. Lafortune, and D. Teneketzis, “On the effect of communication delays in failure diagnosis of decentralized discrete event systems,” in *Proc. of the 39th IEEE Conf. on Decision and Control*, pp. 2245–2251, 2000.
- [14] J. Shengbing, R. Kumar, and H. E. Garcia, “Optimal sensor selection for discrete-event systems with partial observation,” *IEEE Trans. Automatic Control*, vol. 48, pp. 369–381, March 2003.
- [15] D. N. Pandalai and L. E. Holloway, “Template languages for fault monitoring of timed discrete event processes,” *IEEE Trans. Automatic Control*, vol. 45, pp. 868–882, May 2000.
- [16] V. S. Srinivasan and M. A. Jafari, “Fault detection/monitoring using time Petri nets,” *IEEE Trans. Syst., Man, & Cybern.*, vol. 23, pp. 1155–1162, July–Aug. 1993.
- [17] S. K. Yang and T. S. Liu, “A Petri net approach to early failure detection and isolation for preventive maintenance,” *Quality & Reliability Engineering International*, vol. 14, pp. 319–330, Sept.–Oct. 1998.
- [18] C. H. Kuo and H. P. Huang, “Failure modeling and process monitoring for flexible manufacturing systems using colored timed Petri nets,” *IEEE Trans. Robotics & Automation*, vol. 16, pp. 301–312, June 2000.
- [19] G. Barrett and S. Lafortune, “Decentralized supervisory control with communicating controllers,” *IEEE Trans. Automatic Control*, vol. 45, pp. 1620–1638, Sept. 2000.
- [20] R. K. Boel and J. H. van Schuppen, “Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers,” in *Proc. of the Sixth International Workshop on Discrete Event Systems*, pp. 175–181, 2002.
- [21] R. Sengupta, “Diagnosis and communications in distributed systems,” in *Proc. of the International Workshop on Discrete Event Systems*, pp. 144–151, 1998.

- [22] R. Debouk, S. Lafortune, and D. Teneketzis, “Coordinated decentralized protocols for failure diagnosis of discrete event systems,” *Discrete Event Dynamic Systems: Theory and Application*, vol. 10, pp. 33–86, Jan. 2000.
- [23] A. Benveniste, E. Fabre, C. Jard, and S. Haar, “Diagnosis of asynchronous discrete event systems: a net unfolding approach,” *IEEE Trans. Automatic Control*, vol. 48, pp. 714–727, May 2003.
- [24] A. Aghasaryaiu, E. Fabre, A. Benveniste, R. Boubour, and C. Jard, “Fault detection and diagnosis in distributed systems: an approach by partially stochastic Petri nets,” *Discrete Event Dynamic Systems: Theory and Applications*, vol. 8, pp. 203–231, June 1998.
- [25] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, “Diagnosis of a class of distributed discrete-event systems,” *IEEE Trans. Syst., Man, & Cybern. Part A: Systems & Humans*, vol. 30, pp. 731–752, Nov. 2000.
- [26] A. Giua and C. Seatzu, “Observability of place/transition nets,” *IEEE Trans. Automatic Control*, vol. 47, pp. 1424–1437, Sept. 2002.
- [27] Y. Li and W. M. Wonham, “Control of vector discrete-event systems — Part I: the base model,” *IEEE Trans. Automatic Control*, vol. 38, pp. 1214–1227, Aug. 1993.
- [28] Y. Li and W. M. Wonham, “Control of vector discrete-event systems — Part II: controller synthesis,” *IEEE Trans. Automatic Control*, vol. 39, pp. 512–531, March 1994.
- [29] K. Yamalidou, J. Moody, M. Lemmon and P. Antsaklis, “Feedback control of Petri nets based on place invariants,” *Automatica*, vol. 32, pp. 15–28, Jan. 1996.
- [30] J. O. Moody and P. J. Antsaklis, “Petri net supervisors for DES with uncontrollable and unobservable transitions,” *IEEE Trans. Automatic Control*, vol. 45, pp. 462–476, March 2000.
- [31] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, Boston, MA, 1999.
- [32] T. Murata, “Petri nets: properties, analysis and applications,” *Proc. of the IEEE*, vol. 77, pp. 541–580, April 1989.
- [33] J. Riordan, *An Introduction to Combinatorial Analysis*, John Wiley & Sons, New York, 1958.
- [34] A. Aybar and A. Iftar, “Overlapping decompositions and expansions of Petri nets,” *IEEE Trans. Automatic Control*, vol. 47, pp. 511–515, March 2002.
- [35] S. Christensen and L. Petrucci, “Modular analysis of Petri nets,” *The Computer Journal*, vol. 43, pp. 224–242, March 2000.
- [36] R. Y. Al-Jaar and A. A. Desrochers, “Petri nets in automation and manufacturing,” *Advances in Automation and Robotics*, G. N. Saridis, Ed., vol. 2, JAI Press, Greenwich, CT, 1990.