Finite-State Machine Embeddings for Nonconcurrent Error Detection and Identification

Christoforos N. Hadjicostis, Member, IEEE,

Abstract—In digital sequential systems that operate over several time steps, a state-transition fault at any time step during the operation of the system can corrupt its state and render its future functionality useless. Such state-transition faults are usually handled by embedding the given sequential system into a larger one, in a way that preserves the state evolution and properties of the original system while enabling an external mechanism to perform checks to detect, identify and correct errors in the encoded state of this redundant system. Checking is typically performed concurrently (i.e., at the end of each time step) and can potentially cause high power consumption or an overall slowdown in the system; more importantly, concurrent checking imposes significant reliability requirements on the error-detection/identification mechanism. In this paper, we develop a methodology for systematically constructing embeddings of finite-state machines so that the external mechanism can capture transient state-transition faults via checks that are performed in a nonconcurrent manner (e.g., periodically instead of every time step). More specifically, by characterizing nonconcurrent error-detecting/identifying capabilities in terms of state encoding constraints and redundant dynamics, the proposed approach can be used to construct a redundant version of the given FSM that allows the external mechanism to detect and identify errors due to past state-transition faults based on an analysis of the current, possibly corrupted FSM state. As a result, the checker in such designs can operate at a slower speed than the rest of the system which relaxes the stringent requirements on its reliability.

Index Terms—finite-state machines, fault tolerance, transient faults, nonconcurrent error detection and correction, error recovery

I. INTRODUCTION

In an effort to utilize redundancy in efficient ways, researchers have extensively used coding and embedding techniques to achieve fault tolerance. Apart from the numerous examples in communication and data storage applications [1], [2], these approaches have also been successful in protecting computational systems against transient or permanent hardware faults. Examples of such approaches include arithmetic codes [3], [4], algorithm-based fault tolerance (ABFT) schemes [5], [6], [7], [8], [9] and algebraic embeddings [10], [11], [12]. In this spirit, a number of researchers focused on protecting discrete-time (DT) dynamic systems, including linear time-invariant dynamic systems [13], [14], linear finite-state machines (LFSMs) [15], [16], [17], [18], and arbitrary finite-state machines (FSMs) [19], [20], [21], [22], [23], [24], [25], [12]. To achieve fault tolerance, these approaches essentially map the state of a given DT dynamic system into the higher dimensional state space of a larger, redundant DT dynamic system, in a way that preserves the evolution and state of the original one while enabling an external mechanism to detect and correct faults that corrupt the state of the redundant system. Reference [26] provides an overview of these techniques for both combinational and dynamic systems.

Most of the techniques mentioned above are based on concurrent checks, i.e., checks that are performed at the end of each time step, and detect and identify errors in the system by analyzing violations on enforced state encodings. The goal is to immediately account for errors, i.e., to ensure that an error that occurs at time step $t$ will be detected, identified and corrected by the concurrent checking and correction that is performed at the end of time step $t$. In this paper, we protect arbitrary FSMs by developing coding and embedding techniques that allow nonconcurrent error detection and identification. In these nonconcurrent checking schemes, the error-detection/identification mechanism only checks for faults at nonconsecutive time steps (e.g., periodically) instead of every step, each time detecting and identifying errors that may have occurred since the last check. Note that traditional fault-tolerant constructions that are designed for concurrent checking will typically fail to catch faults if checks are performed periodically.

The motivating application behind the proposed schemes is our desire to protect modern digital sequential systems, including embedded controllers and micro-controllers, against transient state-transition faults, i.e., faults that do not appear on a consistent basis but only occasionally manifest themselves, causing the corruption of the state at a particular time step (transient faults are usually caused by glitches due to noise, electromagnetic interference, or environmental factors [27], [28], [29]). Current trends towards higher clock speeds, lower power consumption and smaller transistor sizes lead to a significant increase in the frequency of transient (or so-called “soft”) errors. Since the occurrence of a transient state-transition fault during a particular time step causes the corruption of the system state and the likely propagation of errors at future time steps, the main contribution of this paper lies in the development of methodologies that allow for efficient, nonconcurrent error detection and identification of such faults. The constructions in this paper should be viewed as redundant versions of a given FSM, designed and implemented so as...
to allow the detection and identification of up to a certain number of transient state-transition faults. Since the error detector/corrector will typically be implemented along with the redundant implementation of the sequential circuit, we assume that the detector/corrector has access to the augmented state of this redundant version of the original FSM (i.e., the state of the redundant circuit). One particularly attractive feature of the proposed schemes is that the error-detection/identification mechanism checks for faults periodically instead of every time step, which relaxes the stringent reliability requirements that are normally imposed on the error-detection/identification mechanism.

The development in this paper is a nonlinear extension of the nonconcurrent error detection and identification schemes that were presented in [30], [31] for DT linear time-invariant dynamic systems and in [32], [33] for LFSMs. There are several key issues that arise in making the transition to arbitrary FSMs and we discuss some of them in the theorems and examples that are presented in this paper. Although some of the techniques in [33] can also be used in this more general case, the proofs and, more importantly, the interpretation of the results is conceptually harder in the case of FSMs; what allows us to utilize the techniques in [33] is our ability to track additive errors via a suitable choice for a redundant FSM.

This paper is organized as follows. In Section II we provide an introduction to relevant previous work on fault diagnosis and monitoring as well as a quick overview of nonconcurrent error detection, identification and correction schemes for fault-tolerant LFSMs. In Section III we construct FSM embeddings that allow for nonconcurrent detection and identification of errors due to transient state-transition faults that may have occurred during the operation of the FSM since the last check. In Section IV we discuss some further issues, including comments on how rollback-based mechanisms can be used in conjunction with the proposed schemes to obtain reliable FSMs out of unreliable components.

II. BACKGROUND

A. Related Previous Work on Fault Diagnosis and Monitoring

Before we describe our proposed approach for achieving nonconcurrent error detection and identification in FSMs, we point out that there is a large volume of related previous research. For example, within the systems and control community, there has been a long-standing interest in fault diagnosis in large-scale DT dynamic systems. This work includes diagnosis in discrete event systems (DES), such as finite automata [34], [35], timed systems [36], [37], [38], communication networks [39], [40], [41], and Petri net models [42], [43], [44], [45]. The goal in most of these settings is to obtain an external mechanism that operates alongside with the given system and is able to detect and identify faults from a given, predetermined set. The approach is inherently nonconcurrent and is based on locating a set of invariant properties of the system.

1In concurrent schemes, the error-detection/identification mechanism is commonly assumed fault-free; this assumption is reasonable only if the complexity of error detection and identification is considerably less than the complexity of the system under consideration.

Related approaches have also appeared within the computer engineering community under the name fault monitoring. For FSMs, this idea essentially translates to assigning a (possibly non-unique) signature to each state of the given FSM, so that a smaller FSM (called a monitor) that receives the same input as the original system can keep track of the signatures of the sequence of states that are visited. Errors are detected at various checkpoints by finding inconsistencies between the signature of the actual FSM state and the signature reached in the monitor. Faults that lead to erroneous state sequences with correct signatures are not detected (see, for example, [21], [52], [22], [23], [24], [12]). These monitoring techniques are inherently nonconcurrent and aim at exploiting the dynamic structure (state-transition functionality) of the given FSM in order to obtain redundancy-efficient schemes; they usually assume that the monitor is fault-free.

The nonconcurrent error detection, identification and correction approach that we develop in this paper for arbitrary FSMs has a similar overall objective in the sense that we need to nonconcurrently detect and identify errors. There are, nevertheless, important differences that one needs to keep in mind when studying our development. For instance, the FSM to be diagnosed or monitored in our schemes is not fixed as is the case in fault diagnosis and monitoring schemes; since we are designing the digital sequential system, we have the flexibility of choosing the redundancy in our FSM design to allow for more efficient/effective error detection or identification. On the other hand, our nonconcurrent schemes need to be able to partially overcome errors in the monitor itself, which is in sharp contrast with fault diagnosis and fault monitoring schemes (where the monitor/checker are commonly assumed fault-free). Finally, nonconcurrent error detection and identification schemes for arbitrary FSMs need to be designed so that all errors out of a given list can be identified. This goal is imperative in microprocessor or control units, and needs to be achieved efficiently by taking advantage of the different time scales at which the sequential system and the error-detection/identification mechanism are operating.

B. Nonconcurrent Error Detection, Identification and Correction in LFSMs

The work presented in this paper is an extension of the nonconcurrent error detection and identification schemes that were developed for linear finite-state machines (LFSMs) in [32], [33]. An LFSM $S$ has state evolution of the form

$$q[t + 1] = Aq[t] \oplus Bx[t],$$

(1)
where $t$ is the discrete-time index, $\mathbf{q}[t]$ is the state vector and $\mathbf{x}[t]$ is the input vector. We assume that $\mathbf{q}[t]$ is $d$-dimensional, $\mathbf{x}[t]$ is $u$-dimensional, and $\mathbf{A}$ and $\mathbf{B}$ are constant matrices of appropriate dimensions. All vectors and matrices have entries from $GF(q)$, the Galois field of order $q$ (i.e., the LFSM $\mathcal{S}$ has $q^d$ states). In Eq. (1) and for the rest of this paper, matrix-vector multiplication (denoted by juxtaposition) and vector-vector addition (denoted by $\oplus$) are performed as usual except that individual element multiplications and additions are performed in $GF(q)$.

The LFSM $\mathcal{S}$ in Eq. (1) is protected in [18], [32], [33] by constructing a larger LFSM $\mathcal{H}$ with state evolution

$$
\xi[t+1] = \mathbf{A}\xi[t] \oplus \mathbf{B}\mathbf{x}[t]
$$

(2)

where $\xi[t]$ is of dimension $\eta = d+m$ for some positive integer $m$. The initial state $\xi[0]$ and matrices $\mathbf{A}$, $\mathbf{B}$ chosen so that, under fault-free conditions, the state $\xi[t]$ for $t \geq 0$ provides complete information about $\mathbf{q}[t]$, the state of the original LFSM $\mathcal{S}$, and vice-versa. More specifically, in [18], [32], [33] these decoding and encoding constraints are restricted to be linear, i.e., it is assumed that there exist matrices $\mathbf{L}$ and $\mathbf{G}$ (with entries in $GF(q)$) such that, when $\xi[0] = \mathbf{G}\mathbf{q}[0]$ and under fault-free conditions,

$$
\mathbf{q}[t] = \mathbf{L}\xi[t],
$$

(3)

$$
\xi[t] = \mathbf{G}\mathbf{q}[t]
$$

(4)

for all $t \geq 0$. Under the above conditions, the development in [18] provides a characterization of the class of appropriate redundant LFSMs in terms of the redundant dynamics of the modes that are added to the LFSM and the coupling from the redundant (added) to the non-redundant (original) modes.

Nonconcurrent error detection and identification for LFSMs is analyzed in [32], [33] where it is assumed (without loss of generality) that the LFSM begins operation at time step 0 and that the first nonconcurrent check is performed at the end of time step $L - 1$ (beginning of time step $L$). If a total of $D$ transient faults occur during time steps $L - k_1 - 1$, $L - k_2 - 1$, ..., $L - k_D - 1$, originally corrupting state variables $i_1$, $i_2$, ..., $i_D$ by initial additive errors $v_1$, $v_2$, ..., $v_D$, respectively, the error syndrome at the end of time step $L - 1$ (which can be calculated based on the erroneous state of the redundant LFSM) needs to provide enough information for the detection and identification of all errors [i.e., the identification of the originally affected variables $(i_1, i_2, ..., i_D)$, the values by which they were initially corrupted $(v_1, v_2, ..., v_D)$ and the time steps during which the errors took place $(L - k_1 - 1, L - k_2 - 1, ..., L - k_D - 1)$]. The work in [32], [33] presented explicit constructions of redundant LFSMs that can be used to provide fault tolerance to the LFSM $\mathcal{S}$ in Eq. (1). The corresponding redundant LFSM uses $2D$ additional state variables and enables the identification (respectively, detection) of up to $D$ (respectively, $2D$) errors due to transient state-transition faults in the interval $[0, L - 1]$. The approach makes use of Bose-Chaudhuri-Hocquenghem (BCH) coding techniques and is optimal in the sense that it uses the minimal possible number of additional state variables.

### III. Nonconcurrent Error Detection and Identification in FSMs

#### A. Notation and Preliminaries

Consider the FSM $\mathcal{S}$ with state set $Q = \{q_1, q_2, ..., q_N\}$, input set $X = \{x_1, x_2, ..., x_U\}$ and output set $Y = \{y_1, y_2, ..., y_J\}$. Its state $q[t]$ and input $x[t]$ at time step $t$ specify its next state $q[t+1]$ via the next-state function

$$
q[t+1] = \delta(q[t], x[t])
$$

(5)

and its output via the output function

$$
y[t] = \lambda(q[t], x[t])
$$

(6)

We assume that functions $\delta$ and $\lambda$ are defined for all pairs in $Q \times X$. (One can always redefine them so that this is the case without loss of generality, e.g., by adding a “dummy” state with an associated “dummy” output, and by forcing any undefined transitions and outputs to result in this “dummy” state/output.)

In this paper we focus on detecting, identifying, and correcting state-transition faults, i.e., faults that result in the corruption of the state of FSM $\mathcal{S}$ at a particular time step. We focus on transient faults, i.e., faults that do not appear on a consistent basis but only manifest themselves with a certain probability (transient faults are usually caused by glitches due to noise, electromagnetic interference, or environmental factors [27], [28], [29]). Due to the memory of the system, a transient fault that corrupts the state of the FSM at a particular time step can influence its future behavior, potentially rendering its functionality useless. Since we are designing the redundant version of the original FSM, we allow the error correcting mechanism to have full access to the augmented state (for example, in the case of a sequential system, the augmented state as well as the error correcting mechanism will presumably be implemented on the same circuit). The mapping $\lambda$ will not be relevant for our discussions here since we focus on ensuring that the sequence of states is correct (as long as the hardware that implements the mapping $\lambda$ is designed in a fault-tolerant manner, a correct state sequence will produce a correct output sequence). Note that transient state-transition faults are actually harder to deal with than transient output faults because a transient fault that corrupts the output of the FSM at a particular time step does not have any adverse effects on future system states or outputs.

In general, the implementation of the FSM $\mathcal{S}$ in Eq. (5) as a sequential system will rely on encoding each of the $N$ states into a binary vector with $b$ bits, where $b$ is an integer satisfying $2^b \geq N$ (so that there are enough binary strings to represent all $N$ states). The states of the FSM $\mathcal{S}$ can be captured by a set of $N$ $b$-dimensional binary vectors

$$
Q_s = \{q_s^{(1)}, q_s^{(2)}, ..., q_s^{(N)}\}
$$
that correspond to states \( \{q_1, q_2, ..., q_N\} \) respectively. These vectors can certainly be viewed as \( q \)-dimensional vectors with elements in \( GF(2) \) (i.e., with elements “0” or “1”) or as \( q \)-dimensional vectors with elements in some finite field \( GF(q) \) with \( q > 2 \) [where, of course, the elements “0” or “1” are mapped to two of the elements in \( GF(q) \)]. One can also view the vectors in \( \mathcal{Q}_s \) as lower dimensional vectors with elements in a higher dimensional field \( GF(q) \). For example, the 8-dimensional binary vector \( \mathbf{q}_s^{(j)} = [1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1]^T \), can be treated as the 4-dimensional vector \( \mathbf{q}_s^{(j)} = [10 \ 11 \ 00 \ 01]^T \in GF(4) \) [where elements “10,” “11,” “00” and “01” represent elements in \( GF(4) \)], or as the 2-dimensional vector \( \mathbf{q}_s^{(j)} = [1011 \ 0001]^T \in GF(2^4) \) [where elements “1011” and “0001” represent elements in \( GF(2^4) \)]. For the purposes of this paper, we will be content to think of the vectors in the set \( \mathcal{Q}_s \) as \( q \)-dimensional vectors with entries in \( GF(q) \) where \( q^d \geq N \). The breakdown of the vector dimensions and the size of the finite field is not particularly important for the analysis in this section but plays a role in the complexity of the redundant implementation and the error-detection/identification mechanism.

For notational convenience, we define the following state transition mappings: for each input \( x_u \in X_U \), let \( \delta_{x_u} : \mathcal{Q}_s \rightarrow \mathcal{Q}_s \) be the mapping that defines the state transition functionality under input \( x_u \), i.e.,

\[
\delta_{x_u}(\mathbf{q}_s^{(j)}) = \mathbf{q}_s^{(l)}
\]

eff \( q_l = \delta(q_j, x_u) \). With this notation

\[
\mathbf{q}[t+1] = \delta_{x}[t](\mathbf{q}[t]) \ , \quad (7)
\]

where \( x[t] \in X_U \) is the input at time step \( t \), and \( \mathbf{q}[t], \mathbf{q}[t+1] \in \mathcal{Q}_s \) are the machine states at time steps \( t \) and \( t+1 \). Mappings \( \delta_{x_u}(\cdot), x_u \in X_u \), can be viewed as nonlinear mappings in \( \mathcal{Q}_s \) [i.e., nonlinear mappings in the \( d \)-dimensional vector space with entries in \( GF(q) \)]. Given a sequence of inputs \( x[\tau_1], x[\tau_1+1], x[\tau_2+2], ..., x[\tau_2] \), we also define

\[
\delta^{x[\tau_1]}(\mathbf{q}_s^{(j)}) = \delta_{x[\tau_1]}(\delta_{x[\tau_1+1]}(\delta_{x[\tau_2]}(\delta_{x[\tau_2+1]}(...(\delta_{x[\tau_1]}(\mathbf{q}_s^{(j)}))...)))
\]

(8)

(if \( \tau_2 < \tau_1 \), then \( \delta^{x[\tau_1]}(\cdot) \) is defined to be the identity mapping in \( \mathcal{Q}_s \)).

### B. Embeddings of Arbitrarily Encoded FSMs

To protect against transient state-transition faults, we will construct a larger FSM \( \mathcal{H} \) with an \( \eta \)-dimensional state vector \( \xi[t] \) with entries in \( GF(q) \), where \( \eta = d+m \) for some \( m > 0 \). Using the notation introduced in the previous section, the state evolution of \( \mathcal{H} \) can be described by

\[
\xi[t+1] = \Delta_{x}[t](\xi[t]) \ , \quad (9)
\]

where \( x[t] \in X_U \) is the input at time step \( t \), and \( \xi[t], \xi[t+1] \in \mathcal{Q}_h \) are the machine states at time steps \( t \) and \( t+1 \). The \( \eta \)-dimensional vector space with entries in \( GF(q) \). The initial state \( \xi[0] \) and all transition mappings \( \Delta_{x_u}(\cdot), x_u \in X \), will be chosen so that under fault-free conditions the state \( \xi[t] \) of FSM \( \mathcal{H} \) for \( t \geq 0 \) provides complete information about \( \mathbf{q}[t] \), the state of the original FSM \( \mathcal{S} \), and vice-versa. More specifically, we will restrict ourselves to decoding and encoding mappings that are linear in \( GF(q) \), i.e., we will require that there exist a \( d \times \eta \) matrix \( L \) and an \( \eta \times d \) matrix \( G \) with entries in \( GF(q) \) such that, when \( \xi[0] = G\mathbf{q}[0] \) and under fault-free conditions

\[
\mathbf{q}[t] = L\xi[t] \ , \quad (10)
\]

\[
\xi[t] = G\mathbf{q}[t] \quad (11)
\]

for all \( t \geq 0 \). Note that \( LG \) does not necessarily have to equal \( I_q \) because the valid states \( \mathbf{q}[t] \) do not necessarily span the \( d \)-dimensional vector space in \( GF(q) \).

As in the case of LFSMs in [32, 33], the \( d \)-dimensional vector \( \mathbf{q}[t] \) is uniquely represented by the \( \eta \)-dimensional vector (codeword) \( \xi[t] = G\mathbf{q}[t] \), where \( G \) has full column-rank and is essentially the \( \eta \times d \) generator matrix of an \( (\eta, d) \) linear code in \( GF(q) \) [1], [2]. Concurrent error detection is straightforward: under fault-free conditions, the redundant state vector must be in the column space of \( G \); therefore, all we need to check is that, at each time step \( t \), the redundant state \( \xi[t] \) lies in the column space of \( G \) (in coding theory terminology, we need to check that \( \xi[t] \) is a codeword of the linear code generated by \( G \) [1], [2]). Equivalently, we can check that \( \xi[t] \) is in the null space of an appropriate \textit{parity check matrix} \( H \), so that \( H\xi[t] = 0 \). Here, the parity check matrix \( H \) is an \( m \times \eta \) matrix that has (full) row-rank and satisfies \( HG = 0 \). Evaluating the parity check \( s[t] = H\xi[t] \) at every time step (to check whether is zero or not) amounts to concurrent checking. Concurrent error correction associates with each valid state in \( \mathcal{H} \) (of the form \( G\mathbf{q}[s] \)), a unique subset of invalid states that get corrected to that particular valid state. Notice that concurrent error detection and identification can be performed using any of the methods used in the communications setting [1], [2].

The FSM \( \mathcal{H} \) defined in Eq. (9) that satisfies the constraints in Eqs. (10) and (11) if properly initialized will be called a \textit{redundant implementation} for FSM \( \mathcal{S} \) in Eq. (7). The nonlinearities associated with the next-state transition mappings of the original and redundant FSMs [see Eqs. (7) and (9)] make it difficult to obtain a direct characterization of the class of redundant FSMs as was done in [32, 33] for LFSMs. This class of redundant FSMs, however, can be indirectly described in terms of the following corollary.

**Corollary 3.1:** If FSM \( \mathcal{H} \) is a redundant implementation for FSM \( \mathcal{S} \), then

\[
L\Delta_{x_u}(G_{s}^{(j)})) = \delta_{x_u}(G_{s}^{(j)}), \quad x_u \in X, \quad s^{(j)} \in \mathcal{Q}_s . \quad (12)
\]

Versions of the above theorem that do not impose linear decoding and encoding constraints can be found in [26]. The restriction to linear constraints is important because it allows us to develop a particular class of redundant FSMs that satisfies the condition in Corollary 3.1 and is able to track the propagation of errors due to transient state transition faults.

\[^3\text{This subset usually contains \( \eta \)-dimensional vectors with small Hamming distance from the associated valid codeword. The Hamming distance between two vectors } x = (x_1, x_2, ..., x_n) \text{ and } y = (y_1, y_2, ..., y_n) \text{ (with elements from the specified finite field } GF(q) \text{) is the number of entries at which } x \text{ and } y \text{ differ} [1], [2]. The minimum Hamming distance } d_{\text{min}} \text{ of a code (collection of vectors of length } \eta \text{) determines its error-detecting/correcting capabilities: a code can detect errors at } d_{\text{min}} - 1 \text{ entries; it can correct errors at } \left\lfloor \frac{d_{\text{min}} - 1}{2} \right\rfloor \text{ entries.} \]
The class of redundant FSMs that will be used throughout the remainder of this paper to construct our nonconcurrent error detection and identification schemes is the one described in the theorem below.

**Theorem 3.1:** Let the decoding, encoding and parity check matrices of a redundant implementation \( H \) for FSM \( S \) be given by

\[
\begin{align*}
L & = \begin{bmatrix} I_d & 0 \end{bmatrix}, \\
G & = \begin{bmatrix} I_d & C \end{bmatrix}, \\
H & = \begin{bmatrix} -C & I_m \end{bmatrix}.
\end{align*}
\]

FSM \( H \) is a redundant implementation for FSM \( S \) [i.e., the state of FSM \( H \)] satisfies the decoding and encoding constraints in Eqs. (10) and (11) if it is initialized in state \( \xi[0] = Gq[0] \) and its next-state transition mapping under each input \( x_u \in X_U \) satisfies (for any \( d \times m \) matrix \( A_{12x_u} \) and any \( m \times m \) matrix \( A_{22x_u} \) with entries in \( GF(q) \))

\[
\Delta_{x_u}(\xi[t]) = \begin{bmatrix}
\delta_{x_u}(q[t]) \oplus A_{12x_u} C q[t] \oplus A_{12x_u} C r[t] \\
C \delta_{x_u}(q[t]) \oplus (C A_{12x_u} C \oplus A_{22x_u} C) q[t] \oplus (C A_{12x_u} \oplus A_{22x_u}) C r[t]
\end{bmatrix}
\]

where \( \xi[t] \equiv \begin{bmatrix} q[t] \\
\xi_r[t] \end{bmatrix} \) is a fault-free condition [here \( q[t] \) is a \( d \)-dimensional vector (in \( Q_x \)) and \( \xi_r[t] \) is an \( m \)-dimensional vector, both with entries in \( GF(q) \)].

Note that the coupling matrices \( A_{12x_u}, x_u \in X_U \), and the redundant dynamics matrices \( A_{22x_u}, x_u \in X_U \), are completely free for us to choose and they essentially parameterize the redundant FSM implementations of Theorem 3.1 (particular choices for \( A_{12x_u} \) and \( A_{22x_u} \) that aid in nonconcurrent error detection and identification will be discussed later on). Also note that, with a little bit of abuse of notation, we can write the next-state transition mapping in Eq. (13) as

\[
\Delta_{x_u}(\xi[t]) = \begin{bmatrix}
\delta_{x_u}(q[t]) \oplus A_{12x_u} C r[t] \\
C \delta_{x_u}(q[t]) \oplus (C A_{12x_u} C \oplus A_{22x_u} C) q[t] \oplus (C A_{12x_u} \oplus A_{22x_u}) C r[t]
\end{bmatrix}
\]

In terms of this view, it is clear that our motivation for studying this class of redundant FSM implementations stems from the redundant LFSM constructions in [32], [33]. Note that the influence of the coupling and the redundant dynamics on the overall FSM state is restricted to be linear.

**Proof:** Clearly, \( L, G \) and \( H \) is a valid triple for decoding, encoding and parity check matrices (i.e., \( L(Gq[t]) = q[t] \) and \( HG = 0 \) with \( H \) having full row-rank \( m \)). The remainder of the proof is by induction. Clearly, \( \xi[0] \) satisfies \( \xi[0] = Gq[0] \). Let us denote the state \( \xi[t] \) at time step \( t \) as

\[
\xi[t] = \begin{bmatrix} q[t] \\
\xi_r[t] \end{bmatrix}
\]

where, under fault-free conditions at least, \( \xi_r[t] \) satisfies \( \xi_r[t] = Cq[t] \) (so that \( \xi[t] = Gq[t] \)). The next state can be calculated as shown in Figure 1 and, since \( \xi[t + 1] \) satisfies the decoding, encoding and parity check restrictions imposed by matrices \( L, G \) and \( H \), we can conclude the proof of the theorem by induction. \( \square \)

Note that the above theorem can easily be generalized to a case where the next-state transition mappings in Eq. (13) are replaced by

\[
\Delta_{x_u}(\xi[t]) = \begin{bmatrix}
\delta_{x_u}(q[t]) \oplus \delta_{12x_u}(Cq[t]) \oplus \delta_{12x_u}(\xi_r[t]) \\
C \delta_{x_u}(q[t]) \oplus (C A_{12x_u} C \oplus A_{22x_u} C) q[t] \oplus (C A_{12x_u} \oplus A_{22x_u}) C r[t]
\end{bmatrix}
\]

for all \( x_u \in X \). In this generalization, the linear mappings \( A_{12x_u} \) and \( A_{22x_u} \) are replaced by the (nonlinear) mappings \( \delta_{12x_u}(\cdot) \) and \( \delta_{22x_u}(\cdot) \).

**C. Transient State-Transition Faults and Error Propagation**

We are interested in designing the redundant FSM \( H \) so that we can protect against transient state-transition faults. We assume that a transient fault during the calculation of the next state at time step \( t \) causes an erroneous value at exactly one of the state variables in the next state vector \( \xi[t + 1] \) (this will be the case if FSM \( H \) is implemented so that the next value of each state variable is calculated by separate hardware	extsuperscript{4} [26], [14]). It will become clear in our analysis later on that this assumption can be relaxed in a straightforward manner; we adopt it at this point to make the discussion more transparent.

To analyze nonconcurrent error detection and identification, we assume (without loss of generality) that the FSM begins

\textsuperscript{4}This would be true, for example, if the value of each bit in the next state vector is calculated by separate hardware.
operation with $\xi[0] = Gq[0]$, and that the first nonconcurrent parity check is performed at the end of time step $L-1$ (i.e., at the beginning of time step $L$). Assume for now that a single transient state-transition fault takes place in the interval $[0, L-1]$. More specifically, the fault takes place during the execution of time step $t = L - k - 1$, $0 \leq k \leq L-1$, originally affecting the $i$th state variable by an initial (additive) error value $v$ so that the erroneous state vector at time step $t = L - k$ is given by

$$
\xi_f[L - k] = \xi[L - k] \oplus ve_i ,
$$

where $\xi[L - k]$ is the state that FSM $\mathcal{H}$ would be in under fault-free conditions and $e_i$ is an $n$-dimensional column vector with a unique nonzero entry with value “1” at its $i$th position. Note that this additive error model is quite general: any fault that corrupts a single state variable can be modeled in the above fashion for appropriate $v$ and $e_i$.

A (concurrent) parity check at the end of time step $L - k - 1$ would result in the syndrome

$$
\mathbf{s}[L - k] \equiv H\xi_f[L - k] = vH(:, i) ,
$$

where $H(:, i)$ denotes the $i$th column of matrix $H$ (i.e., concurrent error-detecting/identifying capabilities depend solely on matrix $H$). Assuming no other fault occurs in the interval $[L - k, L - 1]$, we can use Eq. (8) to find the erroneous state of FSM $\mathcal{H}$ at the end of time step $L - 1$ (beginning of time step $L$):

$$
\xi_f[L] = \Delta_x[L - 1] \xi_f[L - k] = \Delta_x[L - 1] (\xi[L - k] \oplus ve_i) ,
$$

(15)

where $\xi[L - k]$ is the error-free state at time step $L - k$. For an arbitrary redundant implementation, the above expression cannot be further simplified, making it hard to evaluate the nonconcurrent syndrome $\mathbf{s}[L] \equiv H\xi_f[L]$ at the end of time step $L - 1$. For the class of FSMs that are given in Theorem 3.1, however, things can be simplified in the way indicated in the following theorem.

**Theorem 3.2:** Let the redundant implementation $\mathcal{H}$ for FSM $\mathcal{S}$ be constructed as in Theorem 3.1. A single error due to a transient state-transition fault that occurred during the execution of time step $L - k - 1$ and corrupted the $i$th state variable by an additive value $v$ results in the nonconcurrent syndrome

$$
\mathbf{s}[L] \equiv H\xi_f[L] = v \left( \prod_{\tau=L-k}^{L-1} A_{22x[i]} \right) He_i .
$$

More generally, an error that occurred during the execution of time step $L - k - 1$ and corrupted the state of the system by an additive error vector $e$ (so that $\xi_f[L - k] = \xi[L - k] \oplus e$) results in the nonconcurrent syndrome

$$
\mathbf{s}[\ell] \equiv H\xi_{f}[\ell] = \left( \prod_{\tau=L-k}^{\ell-1} A_{22x[i]} \right) He_i ,
$$

for $\ell \geq L - k$.

**Proof:** Due to the fault during time step $L-k-1$, the erroneous state at time step $L - k$ is given by

$$
\xi_f[L - k] = \xi[L - k] \oplus ve_i ,
$$

where $\xi[L - k] = \left[ \begin{array}{c} q[L - k] \\ Cq[L - k] \end{array} \right]$ is the error-free state at time step $L - k$. We will use induction to show that for $\ell \geq L - k$ the syndrome at time step $\ell$, defined as $\mathbf{s}[\ell] = H\xi_f[\ell]$, satisfies

$$
\mathbf{s}[\ell] = v \left( \prod_{\tau=L-k}^{\ell-1} A_{22x[i]} \right) He_i .
$$

Since $\xi_f[L - k] = \xi[L - k] \oplus ve_i$, the basis of induction is satisfied for $\ell = L - k$. Assuming that $\mathbf{s}[\ell]$ satisfies the induction hypothesis, i.e., that $\mathbf{s}[\ell] = v \left( \prod_{\tau=L-k}^{\ell-1} A_{22x[i]} \right) He_i$, we now argue that

$$
\mathbf{s}[\ell + 1] = v \left( \prod_{\tau=L-k}^{\ell} A_{22x[i]} \right) He_i .
$$

For ease of notation we will also write $\xi_f[\ell]$ as

$$
\xi_f[\ell] = \left[ \begin{array}{c} \xi_{fs}[\ell] \\ \xi_{fr}[\ell] \end{array} \right] ,
$$

where $\xi_{fs}[\ell]$ represents the top $d$ elements in the erroneous state vector and $\xi_{fr}[\ell]$ denotes the bottom $m$ elements in the state vector. At time step $\ell + 1$, the error propagates according to the next-state transition mapping in Eq. (13) as shown in Figure 2.

When we evaluate the parity check at the end of time step...
\[ \ell \text{ we get} \]
\[
\begin{align*}
    s[\ell + 1] & \equiv H \xi_f[\ell + 1] \\
    & = \begin{bmatrix} -C & I_m \end{bmatrix} \xi_f[\ell + 1] \\
    & = -A_{22,2} \xi_f[\ell] + A_{22,1} \xi_f[\ell] \\
    & = A_{22,\ell} [ -C & I_m ] \xi_f[\ell] \\
    & = A_{22,\ell} H \xi_f[\ell] \\
    & = A_{22,\ell} v \left( \prod_{\tau=L-k}^{\ell-1} A_{22,\tau} \right) H e_i \\
    & = v \left( \prod_{\tau=L-k}^{\ell} A_{22,\tau} \right) H e_i ,
\end{align*}
\]

where in the previous to last step we used the induction hypothesis.

Clearly, if the erroneous state due to a fault during time step \( L - k - 1 \) was given by
\[ \xi_f[L - k] = \xi[L - k] \oplus e , \]

where \( \xi[L - k] \) is the error-free state at time step \( L - k \), the parity check at the end of time step \( \ell, \ell \geq L - k \), would be given by
\[ s[\ell + 1] = \left( \prod_{\tau=L-k}^{\ell} A_{22,\tau} \right) H e . \]

At this point the proof of the theorem is complete. \( \square \)

**Corollary 3.2:** Let the redundant implementation \( \mathcal{H} \) for FSM \( S \) be constructed as in Theorem 3.1 with \( A_{22,xu} = A_{22} \) for all \( x_u \in X_U \). A single error due to a transient state-transition fault that occurred during the execution of time step \( L - k - 1 \) and corrupted the ith state variable by an additive value \( v \) results in the nonconcurrent syndrome
\[ s[L] \equiv H \xi_f[L] = v A_{22}^k H e_i . \]

We now generalize the above results to the case of multiple faults within the interval \([0, L - 1]\). We assume that a total of \( D \) transient state-transition faults occur during the executions of time steps \( L - k_1 - 1, L - k_2 - 1, \ldots, L - k_D - 1 \), originally corrupting state variables \( v_1, v_2, \ldots, v_D \) by initial additive errors \( v_1, v_2, \ldots, v_D \), respectively. Without loss of generality, we assume that \( 0 \leq k_D \leq k_{D-1} \leq \ldots \leq k_2 \leq k_1 \leq L - 1 \). The following theorem and corollary are generalizations of Theorem 3.2 and Corollary 3.2.

**Theorem 3.3:** Let the redundant implementation \( \mathcal{H} \) for FSM \( S \) be constructed as in Theorem 3.1. Assume that a total of \( D \) transient faults take place in the interval \([0, L - 1]\). More specifically, transient faults occur during the executions of time steps \( 0 \leq L - k_1 - 1 \leq L - k_2 - 1 \leq \ldots \leq L - k_D - 1 \leq L - 1 \), originally corrupting state variables \( v_1, v_2, \ldots, v_D \) by initial additive errors \( v_1, v_2, \ldots, v_D \). Then, the nonconcurrent syndrome \( s[L] \) is given by
\[ s[L] \equiv H \xi_f[L] = \sum_{j=1}^{D} \left\{ v_j \left( \prod_{\tau=L-k_j}^{L-1} A_{22,\tau} \right) H e_i \right\} . \]

**Proof:** Let \( k_{D+2} = k_{D+3} = 0 \) and \( v_{D+1} = 0 \); we will use induction to show that for \( n = 1, 2, \ldots, D + 1 \), the syndrome at time step \( \ell \) with \( L - k_n \leq \ell < L - k_{n+1} \) satisfies
\[ s[\ell] \equiv H \xi_f[\ell] = \sum_{j=1}^{n} \left\{ v_j \left( \prod_{\tau=L-k_j}^{\ell-1} A_{22,\tau} \right) H e_i \right\} . \]

Clearly, from Theorem 3.2 we know that the induction hypothesis is satisfied for \( n = 1 \) and \( \ell \) such that \( L - k_1 \leq \ell < L - k_2 \). Assume that \( s[\ell] \) satisfies the induction hypothesis for some integer \( n \in \{1, 2, \ldots, D\} \), i.e., assume that for \( \ell \) such that \( L - k_n \leq \ell < L - k_{n+1} \), the syndrome \( s[\ell] \) satisfies
\[ s[\ell] = \sum_{j=1}^{n} \left\{ v_j \left( \prod_{\tau=L-k_j}^{\ell-1} A_{22,\tau} \right) H e_i \right\} . \]

We now argue that the same will be true for \( n + 1 \). The state at time step \( L - k_{n+1} \) will be
\[ \xi_f[L - k_{n+1}] = \xi[L - k_{n+1}] \oplus e \oplus v_{n+1} e_i , \]

where \( e \) is chosen so that \( \xi[L - k_{n+1}] \oplus e \) captures the erroneous state in which the redundant implementation \( \mathcal{H} \) would be in at time step \( L - k_{n+1} \) due to the presence of state-transition faults during the execution of previous time steps (but in the absence of the fault during the execution of time step \( L - k_{n+1} \)). Clearly, the parity check at the end of time step \( L - k_{n+1} - 1 \) satisfies
\[ H \xi_f[L - k_{n+1}] = H \xi[L - k_{n+1}] \oplus e \oplus v_{n+1} H e_i , \]

where we used the fact that \( H e_i \) should be equal to the syndrome we get at time step \( L - k_{n+1} \) if no error takes place during time step \( L - k_{n+1} - 1 \) (see Theorem 3.2).

Since we can always express the faulty state at time step \( L - k_{n+1} \) in terms of an additive error \( e' \) as
\[ \xi_f[L - k_{n+1}] = \xi[L - k_{n+1}] \oplus e' \]

with \( e' = e \oplus v_{n+1} e_i \) satisfying
\[ H e' = \sum_{j=1}^{n+1} \left\{ v_j \left( \prod_{\tau=L-k_j}^{L-k_{n+1}-1} A_{22,\tau} \right) H e_i \right\} , \]

we can apply Theorem 3.2 again to argue that for \( \ell \) such that \( L - k_{n+1} \leq \ell < L - k_{n+2} \) we will have
\[ s[\ell] = \left( \prod_{\tau=L-k_{n+1}}^{\ell} A_{22,\tau} \right) H e' \]
\[ = \sum_{j=1}^{n+1} \left\{ v_j \left( \prod_{\tau=L-k_j}^{\ell-1} A_{22,\tau} \right) H e_i \right\} . \]

At this point, the proof of the induction hypothesis is complete. \( \square \)

\footnote{If \( k_2 = k_1 \), then the induction hypothesis is trivially satisfied and we can start at \( n = 2 \).}
Corollary 3.3: Let the redundant implementation \( \mathcal{H} \) for FSM \( S \) be constructed as in Theorem 3.1 where \( A_{22} \) for all \( x_u \in X_U \). Assume that a total of \( D \) transient faults occur in the interval \([0, L - 1]\). More specifically, transient faults occur during the execution of time steps \( 0 \leq L - k_1 - 1 \leq L - k_2 - 1 \leq \ldots \leq L - k_D - 1 \leq L - 1 \), originally corrupting state variables \( i_1, i_2, \ldots, i_D \) by initial additive errors \( v_1, v_2, \ldots, v_D \). Then, the nonconcatent syndrome \( s[L] \) is given by

\[
s[L] = H \xi_f[L] = \sum_{j=1}^{D} v_j A_{22}^j H e_{ij} .
\]

D. Multiple Error Detection and Identification

In this section, we focus on redundant FSMs that are constructed as in Corollary 3.3 (so that each FSM input is associated with the same redundant dynamics as given by matrix \( A_{22} \)). Clearly, the corollary implies that if \( D \) transient state-transition faults occur during the execution of time steps \( 0 \leq L - k_1 - 1 \leq L - k_2 - 1 \leq \ldots \leq L - k_D - 1 \leq L - 1 \), originally corrupting state variables \( i_1, i_2, \ldots, i_D \) by initial additive errors \( v_1, v_2, \ldots, v_D \), then the syndrome \( s[L] = H \xi_f[L] \) at the end of time step \( L - 1 \) (beginning of time step \( L \)) will be a linear combination of \( D \) columns of the 
\( m \times L \eta \) syndrome matrix

\[
S = \begin{bmatrix} H & A_{22} & H^2 & \ldots & A_{22}^{L-1} & H \end{bmatrix} .
\]

In order to be able to detect \( D \) or less errors in the interval \([0, L - 1]\) based on the syndrome \( s[L] \), we need all linear combinations of any subset of \( D \) columns of \( S \) to be non-zero; to be able to uniquely identify the originally affected variables \( i_1, i_2, \ldots, i_D \), the values by which they were initially corrupted \( (v_1, v_2, \ldots, v_D) \) and the time steps during which the errors took place \( (L - k_1 - 1, L - k_2 - 1, \ldots, L - k_D - 1) \) based on the syndrome \( s[L] \), we need all linear combinations of any subset of \( D \) columns of \( S \) to be different from a linear combination of any other subset of \( D \) columns of \( S \).

Assuming that the redundant implementations of the given FSM are constructed so that detection and identification of \( D \) (or less) errors is possible, the following procedure summarizes how the corresponding errors can be detected and identified.

1. At the end of time step \( L - 1 \) (beginning of time step \( L \)), calculate \( s[L] = H \xi_f[L] \); if \( s[L] = 0 \), no errors have taken place.

2. Find the unique linear combination of \( D' \) \( (D' \leq D) \) columns of the syndrome matrix \( S \) that results in \( s[L] = \sum_{j=1}^{D'} \alpha_j S(:, \ell_j) \). Identify the unique combination of errors of the form \( v_j, e_{ij} \) and \( k_j, j \in \{1, 2, \ldots, D'\} \) (where \( j \) th column is taken during the execution of time step \( L - k_j - 1 \) and originally affects the \( (i_j) \) th state variable by an initial value \( v_j \)), such that \( s[L] = \sum_{j=1}^{D'} v_j H A^j e_{ij} \) as follows:

\[
v_j = \alpha_j , \quad i_j = 1 + (\ell_j - 1 \mod \eta) , \quad k_j = \frac{\ell_j - i_j}{\eta} .
\]

(To see this, recall that the columns of matrix \( S \) in Eq. (17) have been arranged so that the first \( \eta \) columns correspond to errors during time step \( L - 1 \), the next \( \eta \) columns correspond to errors during time step \( L - 2 \), and so forth. Clearly, index \( k_j \) defined above indicates the \( \eta \)-member set of columns within which column \( S(:, \ell_j) \) falls; similarly, index \( i_j \) indicates the position of \( S(:, \ell_j) \) within that set.)

Note that if \( A_{22} \) is set to zero, the syndrome matrix \( S \) in Eq. (17) will not have any nonconcurrent error detection or identification capabilities. Also notice that for the class of redundant implementations that satisfy the restrictions in Corollary 3.3, the \( m \times L \eta \) syndrome matrix \( S \) in Eq. (17) is the same as the one for LFSMs in [32, 33]. Therefore, we can obtain redundant implementations with the desired properties by appropriately adopting the designs there. More specifically, if matrices \( A_{22} \) and \( H \) are chosen carefully, then the \( m \times L \eta \) syndrome matrix \( S \) in Eq. (17) will have the rank properties required for error detection and identification (namely, any subset of \( 2D \) of its columns will be linearly independent). Following the construction in [32, 33], we first recall the definition and properties of a Vandermonde matrix in \( GF(q) \).

Definition 3.1: Let \( V(x_1, x_2, \ldots, x_r) \) denote the \( 2D \times r \) matrix

\[
V(x_1, x_2, \ldots, x_r) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_r \\
x_1^2 & x_2^2 & \cdots & x_r^2 \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{2D-1} & x_2^{2D-1} & \cdots & x_r^{2D-1} \end{bmatrix}
\]

with elements in \( GF(q) \).

Recall that Vandermonde matrices of the form \( V(x_1, x_2, \ldots, x_{2D}) \) are invertible if and only if \( x_i \neq x_j \) for \( 1 \leq i < j \leq 2D \) (see, for example, [2]); therefore, if the parameters \( \{x_1, x_2, \ldots, x_r\} \) are distinct, then any \( 2D \) columns of \( V \) will be linearly independent. We are now in position to prove the following theorem. The proof is omitted because it is similar to the one in [32, 33].

Theorem 3.4: Let the decoding, encoding and parity check matrices of a redundant implementation \( \mathcal{H} \) for FSM \( S \) be given by

\[
\begin{align*}
L &= \begin{bmatrix} I_d & 0 \end{bmatrix}, \\
G &= \begin{bmatrix} I_d & \mathbf{C} \end{bmatrix}, \\
H &= \begin{bmatrix} -\mathbf{C} & \mathbf{I}_m \end{bmatrix}.
\end{align*}
\]

Assume that the next-state transition mapping under each input \( x_u \in X_U \) is chosen to satisfy

\[
\Delta_{x_u}(\xi[t]) = \begin{bmatrix} \delta_{x_u}(q[t]) \oplus A_{12x_u} C q[t] \oplus A_{12x_u} \xi_r[t] \\
\mathbf{C} \delta_{x_u}(q[t]) \oplus (\mathbf{C} A_{12x_u} \mathbf{C} + A_{22} \mathbf{C}) q[t] \oplus (\mathbf{C} A_{12x_u} + A_{22} \mathbf{C}) \xi_r[t] \end{bmatrix}
\]

for all \( \xi[t] = \begin{bmatrix} q[t] \\ \xi_r[t] \end{bmatrix} \) (here \( q[t] \) is a \( d \)-dimensional vector in \( Q_d \) and \( \xi_r[t] \) is an \( m \)-dimensional vector, both with entries in \( GF(q) \)). Any \( D \) or less errors due to transient state-transition faults in the interval \([0, L - 1]\) can be detected and identified by a parity check at the end of time step \( L - 1 \) (beginning of time step \( L \)) if the following conditions are satisfied:

\[
\Delta_{x_u}(\xi[t]) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]
The constants \( x, x_1, x_2, \ldots, x_\eta \) in the specified finite field \( GF(q) \) are chosen so that

1) \( x_i \neq x_j \) for \( 1 \leq i < j \leq \eta \),
2) \( x^k x_i \neq x^{k'} x_j \) for integers \( k, k' \) such that \( 0 \leq k < k' \leq L - 1 \).

• The number of additional state variables is \( m = 2D \).
• The \( 2D \times 2D \) matrix \( A_{22} \) is of the form

\[
A_{22} = M^{-1}AM,
\]

where \( \Lambda = \text{diag}(1, x, x^2, x^3, \ldots, x^{2D-1}) \) and \( M = V(x_{d+1}, x_{d+2}, \ldots, x_\eta) \).

The \( 2D \times d \) matrix \( C \) is chosen so that

\[
C = -M^{-1}V(x_1, x_2, \ldots, x_d).
\]

As shown more explicitly in [33], the choices in Theorem 3.4 lead to

\[
A_{20}^2H = M^{-1}V(x_1 x^k, x_2 x^k, \ldots, x_d x^k, x_{d+1} x^k, \ldots, x_\eta x^k);
\]

therefore, the syndrome matrix \( S \) in Eq. (17) reduces to

\[
S = M^{-1}V(x_1 x^k, x_2 x^k, \ldots, x_\eta x^k);
\]

and, since parameters \( x, x_1, x_2, \ldots, x_\eta \) are chosen so that all \( x^k x_i \) are distinct, we are guaranteed that any \( 2D \) columns of \( V \) (and \( S \)) are linearly independent. As was the case in [32], [33], the order \( q \) of the finite field needs to be large enough so that the first requirement in Theorem 3.4 can be satisfied (clearly, the first requirement cannot be satisfied if \( q - 1 < L\eta \)). When the order of \( GF(q) \) is small, one can still construct redundant FSMs that are amenable to nonconcurrent error detection and identification by using a larger number of additional state variables. For instance, the detection and identification of up to \( D \) errors requires that any \( 2D \) of the columns of matrix \( S \) in Eq. (17) are linearly independent. In [53] it is shown that by choosing \( m > 2D \) it is possible to have such constructions over smaller fields. Also notice that if we modify slightly the choice of Vandermonde matrices in Theorem 3.4, we can obtain redundant implementations that allow the use of the Peterson-Gorenstein-Zierler (PGZ) algorithm to efficiently determine up to \( D \) errors based on the syndrome \( s[L] \) without resorting to an exhaustive search (see the discussion in [32], [33]).

### E. Example

We now illustrate the proposed techniques by developing a scheme for single error identification in the 12-bit up-counter with enable that is shown in Figure 3. If input \( x \) is set to zero, the counter remains in its current state; if \( x \) is set to one the counter increments to the next state, eventually wrapping back to state zero. In terms of our earlier notation this FSM has two inputs, \( x_1 = x \) and \( x_2 = x' \), and its state can be encoded using 12 bits, e.g., by simply considering the binary value of each state \( (q_0 \rightarrow 00 \ldots 00, q_1 \rightarrow 00 \ldots 01, q_2 \rightarrow 00 \ldots 10, \) and so forth). Note that one can obtain the next state logic that implements \( \delta \) as a logic block that receives 13 bits of input (12 bits for the current state and one bit for the input \( x \)) and produces 12 bits of output (that specify the next state).

This logic block can be produced and optimized via numerous methods, however, the actual contents of \( \delta \) are not necessary for our discussion here.

In our construction we will be treating the 12-bit state encoding as a 3-dimensional vector with elements in \( GF(16) \). To ensure that a single fault corrupts a single variable we might want to produce separate logic for each one of them (i.e., split the logic that implements \( \delta \) into three portions, each of which takes the same 13 bits of input and produces 4 bits of output\(^6\)). For the purposes of this example, we consider single error correction (double error detection) with the length of the check period being three cycles (i.e., \( D = 1 \) and \( L = 3 \)). From Theorem 3.4, we know that single error correction requires a redundant state encoding with two additional state variables \( (m = 2) \). Since we are treating the 12-bit state encoding as a 3-dimensional vector in \( GF(16) \), we have a total of five state variables and we need to choose constants \( x, x_1, x_2, x_3, x_4, x_5 \) in \( GF(16) \) such that all \( x^k x_i \) for \( 0 \leq k \leq 2 \) and \( 1 \leq i \leq 5 \) are distinct (Theorem 3.4). Clearly, we need at least \( 3 \times 5 = 15 \) distinct nonzero numbers and \( GF(16) \) is adequate for the purposes of our construction.

For completeness, the elements of \( GF(16) \) are shown in Table I, including their power, polynomial and vector representation. [Recall that multiplication in \( GF(16) \) is easier using the power notation \((\alpha^i \otimes \alpha^j)^{\text{mod}15}\) and \(0 \otimes f = f \otimes 0 = 0\) for any element \( f \) in \( GF(16) \) whereas addition is easier using the vector notation (via bit-wise XORing of the entries of each vector).] Choosing \( x_1 = \alpha, x_2 = \alpha^2, x_3 = \alpha^3, x_4 = \alpha^4, x_5 = \alpha^5 \) will guarantee that the conditions of Theorem 3.4 are met for \( L = 3 \) (because each term will be a unique power of \( \alpha \) with the last term \((x_2^2 x_5)\) being \( \alpha^{15} = \alpha^0 \)).

Based on the above choice for constants \( x, x_1, x_2, x_3, x_4, x_5 \), the matrices \( A_{22} \) and \( C \) in Theorem 3.4

\(^6\)In other words, the logic blocks that produce the three state variables should not share any hardware.
can be calculated as
\[ M = V(x_4, x_5) = \begin{bmatrix} 1 & 1 \\ \alpha^4 & \alpha^5 \end{bmatrix}, \]
\[ A_{22} = M^{-1} \text{diag}(1, \alpha^5) M = \begin{bmatrix} \alpha^{12} & \alpha^7 \\ \alpha^{11} & \alpha^7 \end{bmatrix}, \]
\[ C = -M^{-1}V(x_1, x_2, x_3) = \begin{bmatrix} \alpha^{12} & \alpha^7 \\ \alpha^{11} & \alpha^7 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 \end{bmatrix} = \begin{bmatrix} \alpha^9 & \alpha^8 & \alpha^3 \\ \alpha^7 & \alpha^2 & \alpha^14 \end{bmatrix}, \]
(recall that "-f" is the same as "+f" when performing addition in GF(2^2)). We can now easily describe the functionality of the redundant implementation: the original machine with next state transition functionality captured by
\[ q[t+1] = \delta_x[q[t]](q[t]) \]
(where q is 12 bits wide and input x[t] could be either zero or one) is replaced by the redundant machine with state evolution
\[ \xi[t+1] = \Delta_x[q[t]](\xi[t]), \]
where
\[ \xi[t] = \begin{bmatrix} q[t] (12 \text{ bits}) \\ \xi_r[t] (8 \text{ bits}) \end{bmatrix} = \begin{bmatrix} q_1[t] (4 \text{ bits}) \\ q_2[t] (4 \text{ bits}) \\ q_3[t] (4 \text{ bits}) \\ \xi_{r1}[t] (4 \text{ bits}) \\ \xi_{r2}[t] (4 \text{ bits}) \end{bmatrix} \]
and
\[ \Delta_x[q[t]] = \frac{\delta_x[q[t]](q[t])}{C \delta_x[q[t]](\xi_r[t])} \otimes A_{22} C q[t] \otimes A_{22} \xi_r[t] \]
(note that A_{12} has been set to zero for simplicity).

To illustrate error detection and identification in the redundant machine, consider a scenario where we start in state q_1 and input sequence x = 1, x = 0, x = 1 is applied (so that under fault-free operation the machine would be traversing from state q_1, to state q_2, state q_3 and state q_3). To be more precise, one should say that the redundant FSM starts in the state that corresponds to q_1 and traverses to the states that correspond to q_2, q_3; the exact fault-free state sequence is illustrated on the left of Table II. Suppose that a fault during time step 2 corrupts the third state variable and causes a transition to an erroneous state captured by the vector \( \{0, 0, \alpha^{13} \} \) (instead of the state that corresponds to q_2). Note that the corruption of the third state variable from \( \alpha^1 \) to \( \alpha^{13} \) corresponds in this particular case to a corruption of all four bits associated with that variable.

During time step 3, the redundant machine evolves according to the input it receives: the part that corresponds to the state of the original machine traverses from state q_13 to q_14, and the redundant part propagates the fault according to Eq. (18) as shown in Table II. At the end of time step 3, when a check operation is performed, we obtain
\[ s[3] \equiv H \xi_f[3] = \begin{bmatrix} \alpha^8 \\ \alpha^9 \end{bmatrix}, \]
where \( H = \begin{bmatrix} -C & I_2 \end{bmatrix} = \begin{bmatrix} \alpha^9 & \alpha^8 & \alpha^3 & 1 & 0 \\ \alpha^7 & \alpha^2 & \alpha^{14} & 0 & 1 \end{bmatrix}. \]

The syndrome matrix is constructed using Eq. (17) as
\[ S = \begin{bmatrix} H & A_{22} H & A_{22}^2 H \end{bmatrix}, \]
where
\[ H = \begin{bmatrix} \alpha^9 & \alpha^8 & \alpha^3 & 1 & 0 \\ \alpha^7 & \alpha^2 & \alpha^{14} & 0 & 1 \end{bmatrix}, \]
\[ A_{22} H = \begin{bmatrix} \alpha^1 & \alpha^5 & \alpha^{11} & \alpha^{13} & \alpha^7 \\ \alpha^4 & \alpha^{10} & \alpha^{12} & \alpha^6 & \alpha^9 \end{bmatrix}, \]
\[ A_{22}^2 H = \begin{bmatrix} \alpha^{10} & \alpha^6 & \alpha^{14} & \alpha^4 & \alpha^2 \\ \alpha^5 & \alpha^{13} & \alpha^3 & \alpha^1 & \alpha^8 \end{bmatrix}. \]

In this case, it can be seen by inspection of S that the syndrome vector is equal to \( \alpha^{12} \) times the eighth column of S. Using this information, we can now fill in the parameters of Eq. (16): since the eighth column of S is the third column of the term \( A_{22} H \), this corresponds to \( k_1 = 1 \) and \( i_1 = 3 \) with \( v_1 = \alpha^{12} \). Therefore, the nonzero \( s[3] \) can be explained by an additive
error of $\alpha^{12}$ affecting the third variable during time step 2. This means the correct value of the third state variable at time step 2 should have been $\alpha^{13} \odot \alpha^{12} = \alpha^1$, which is equal to the fault-free value shown in Table II. Note that our results earlier in this section guarantee that, if at most one fault corrupts a single state variable, the only plausible explanation for the obtained syndrome $s[3]$ is that the third state variable was corrupted by an additive error of $\alpha^{12}$ at time step 2. In general, the task of finding at most $D$ columns of the syndrome matrix $S$ such that a linear combination of them results in the syndrome $s[L]$ can be a challenging task. As discussed in [33], with the right choices for $A_{22}$ and $H$, this task can be facilitated by well-known decoding algorithms, such as the Peterson-Gorenstein Zierler (PGZ) algorithm.

IV. FURTHER DISCUSSION

The real challenge in extending nonconcurrent error detection and identification techniques to FSMs lies in our ability to add redundancy and to effectively model transient state-transition faults in a way that allows us to capture the propagation of errors in the system. The schemes proposed in this paper are successful in systematically tracking errors, including errors in the added (redundant) variables. In particular, FSM embeddings based on Bose-Chaudhuri-Hocquenghem (BCH) codes enable the use of well-known error detection and identification procedures. The proposed approach is related of the approach suggested in [23], [24] for fault monitoring using convolutional codes but has several big differences: (i) the scheme proposed here allows the identification of errors, (ii) checking in our case is performed periodically and is not a “continuous” version of signature monitoring, (iii) we allow for arbitrary types of faults in both the original system and the added hardware, (iv) we are choosing nonzero redundant dynamics (nonzero $A_{22} \odot x_u \in X$) for each input $x_u \in X$, which enables us to track errors including errors in the added (redundant) bits.

If one is interested in applying these techniques towards the construction of reliable sequential systems out of unreliable components, several questions arise. Given a certain technology with an associated probability of transient faults, how can one choose the various parameters of the nonconcurrent error detection and correction schemes proposed in this paper? Particularly relevant are the length of the window $L$, the number of additional state variables $m$, the size of the finite field $q$, and the number of errors $D$ that can be detected/identified. Note that technically the choice of $GF(q)$ is largely arbitrary; in practice, however, $q$ is extremely important in determining the complexity of the redundant implementation and the error-detection/identification mechanism (because it directly reflects on storage and computational complexity requirements).

Another issue has to do with the correction of errors which follows the identification procedure described in Section III-D. When the FSM has some particular structure, the correction procedure can potentially be simplified (see, for example, [32], [33] for the case of LFSMs). In the most general case, however, the correction procedure requires a rollback-based mechanism to perform the correction. More specifically, once errors are detected and identified, we can easily correct the state at time step $L - k_1$ (which denotes the earliest time a transient state-transition fault took place in the interval $[0, L - 1]$). More specifically, assuming that all possibly erroneous state vectors $\xi_f[1], \xi_f[2], \ldots, \xi_f[L]$ are buffered, the correct state at time step $L - k_1$ can be obtained as

$$\xi[L - k_1] = \xi_f[L - k_1] \odot v_1 e_j,$$

where $k_1$, $v_1$ and $e_j$ are obtained by the error detection/identification procedure. From then on, rollback can be used to generate the correct state trajectory for the system.

V. SUMMARY

In this paper we studied a general approach for providing fault tolerance to arbitrary FSMs via nonconcurrent error detection, identification and correction schemes. More specifically, by appropriately choosing the state encodings and the redundant dynamics of a redundant implementation of the given FSM, we systematically developed FSM embeddings that enable errors to be detected and identified based on checks that are performed periodically. Depending on the implementation of the FSM, errors could be due to hardware, software or communication faults.

There are a number of interesting open questions that are related to this development: (i) How can the flexibility in the coupling between the redundant state variables and the original state variables [given by the matrices $A_{12} \odot$ in Eq. (13) or the nonlinear mappings $\delta_{12}(\cdot)$ in Eq. (14)] be exploited to our advantage (e.g., to minimize the complexity of the overall redundant FSM implementation or to aid error correction)? (ii) How can we design systems that are optimal in terms of minimizing the redundant amount of hardware (e.g., AND and OR gates) instead of minimizing the number of additional state variables? (iii) What other codes can be adopted to this setting and what is the corresponding choice of redundant dynamics? (iv) If one is only interested in $k_1$ (the earliest time step at which a fault took place, as would be the case in rollback-based recovery), what type of codes and redundant implementations are appropriate? (v) What other connections can be made between algebraic systems theory and coding theory [54], [55], [56] in terms of designing fault-tolerant FSMs? It would also be interesting to investigate these techniques in conjunction with methodologies for mapping algebraic equations to hardware.

ACKNOWLEDGMENTS

The author would like to acknowledge the assistance of Tim Ernst in preparing the example in Section III-E. The author is also grateful to the reviewers for thorough and constructive reviews that helped improved the presentation of the results in this paper.

REFERENCES


**Christoforos N. Hadjicostis** (M’99) received S.B. degrees in Electrical Engineering, in Computer Science and Engineering, and in Mathematics, 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, MA. In August 1999 he joined the Faculty at the University of Illinois at Urbana-Champaign where he is affiliated with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory. His research interests include systems and control, fault-tolerant combinational and dynamic systems, and fault diagnosis.