Spectral Techniques for Off-Line Testing and Diagnosis of Computer Systems

Mark G. Karpovsky, Fellow IEEE

Research Laboratory on Design and Testing of Computer Hardware
Department of Electrical, Computer and Systems Engineering
Boston University
Boston, MA 02215

Abstract - In this paper we present some of our recent results on applications of spectral techniques over finite fields to the problems of testing and diagnosis of computer systems.

I. Linear Transforms for Testing of Computer Systems.

Consider a computer system of (not necessarily identical) processing elements (PEs) (this system may be a computer board, multichip module, array processor or computer network). If the system has \( n \) output PEs and every PE has at most \( m \) output pins, then the response of the system at any given time can be represented as \( y = (y_0, \ldots, y_{n-1}) \), where \( y_i \in Z_q \), \( q = 2^m \), \( Z_q \) is the field with \( q \) elements and \( E \) \( y \in Z_q \).

Let \( E \subseteq Z_q^{n} \) be the set of all possible errors in the system such that as a result of the error \( e \in E \) \( y \) is replaced by \( y + e \).

Problems of compression of test responses for error detection can be formulated in the following way:

P1. Given \( E \subseteq Z_q^{n} \), construct a transform \( y \mapsto H(y) \).

\[ H(y) = \{ H_j(y) | j = 0, 1, \ldots, r-1 \}, \quad H_j(y) \in Z_q \]

with minimal \( r \) such that \( H(y + e) \neq H(y) \) for all \( y \in Z_q^n \) and \( e \in E \).

P2. Given a distribution \( P(E) = \{ P_0, P_1, \ldots, P_{M-1} \} \), where \( M = q^n \) and \( P_i = \text{Prob} \{ e = i \} \), construct a transform \( y \mapsto H(y) \).

\[ H(y) = \{ H_j(y) | j = 0, 1, \ldots, r-1 \}, \quad H_j(y) \in Z_q \]

minimizing \( \text{Prob} \{ H(y + e) = H(y) \} \) (we assume that all \( y \in Z_q^n \) are uniformly distributed).

We note that \( Z = H(y) \in Z_q^{r} \)

\( (Z = (Z_0, \ldots, Z_{r-1}), Z_j = H_j(y), Z_j \in Z_q) \), and if \( r < n \) by monitoring \( Z \in Z_q^{r} \) instead of \( y \in Z_q^n \) one can achieve a considerable reduction in overheads required for testing.

Let us consider the case \( r = 1 \). Select

\[ Z = Z_1 = H_1(y_0, y_1, \ldots, y_{n-1}) = \sum_{i=0}^{q-1} y_i \alpha^i, \quad \text{where } \alpha \text{ is a primitive element in } Z_q \text{, and all the operations in (1) are in } Z_q \text{ (} n \leq q - 1 \text{).} \]

This approach is known as the signature analysis and it is widely used for off-line testing and for built-in self-testing (BIST) [1, 2].

Computation of the signature, \( Z \), can be implemented in \( n \) steps by the following recursive procedure:

\[ Z^{(i-1)} = \alpha Z^{(i)} + y_{n-i} \quad (i = n, n-1, n-2, \ldots, 1). \]

\[ Z^{(n)} = 0, Z^{(i)} \in Z_q \text{ and } Z^{(0)} = Z. \]

This procedure can be implemented in hardware by a \( m \)-bit linear feedback shift register (LSFR) with parallel input and with the characteristic equation defined by (2) \( (y_{n-i} \text{ is the new input, } Z^{(i)} \text{ is the previous state and } Z^{(i-1)} \text{ is the new state).} \]

The hardware implementation of an \( m \)-bit LSFR requires only \( m \) flip-flops and several XOR gates in feedback loops [1].

We note that \( Z_{i} \) defined by (1) is the first coefficient of the Fourier expansion of \( y = (y_0, y_1, \ldots, y_{n-1}) \) over \( Z_q \) [3, 4].

The network (LSFR) computing \( Z_{i} \) can be considered as a

*This work was supported by the National Science Foundation (USA) under Grant MIP 9208487, the NATO under Grant 910411 and Volkswagen Foundation (Germany).
network computing syndromes for the $q$-ary $(n, n-1)$ Reed-Solomon code of length $n$ with distance $Z$ [5]. In a view of this one can see that (1) provides for the solution of $P_{1}$ for single errors, i.e., when $E = \{ e | \| e \| = 1 \}$, where $\| e \|$ is the number of nonzero components in $e = (e_0, e_1, ..., e_{n-1})$. Let us consider the efficiency of the signature analysis defined by (1) for solution of $P_{2}$. In this case we assume that errors in different components of $y$ are independent (components of $e$ are independent), and all components of $e$ have the same distribution

$$P(e) = \left( p_0, p_1, ..., p_{q-1} \right)$$

$$\left(q = 2^m, \quad R_i = \text{Prob} \{ e_j = i \text{ for all } j = 0, 1, ..., n-1 \} \right).$$

We will present below our results on probabilities

$$P_{AL} = \text{Prob} \{ H_i(y + e) = H_i(y) \} \quad \text{(which are also known as masking or aliasing probabilities)}$$

for three important classes of error distributions which correspond to so-called "symmetrical errors", "independent errors", and "errors of a given multiplicity".

For symmetrical errors we assume [6, 7, 8]:

$$P_i = \frac{1-p_i}{p(2^m-1)} = 1 - p_i \quad \text{for some } 0 < p < 1.$$

In this case using results from [6, 10] it is possible to show that

$$P_{AL} = \text{Prob} \{ H(y + e) = H(y) \} =$$

$$= 2^{-m} \left[ 1 - 2^{-m}(1-p)^n + 2^{-m} - 2^{-m} p(2^m-1) \right].$$

(4)

For independent errors [6, 7, 9] we consider every $e_j \in Z_q$ ($q = 2^m$) as an $m$-bit binary vector, and we assume that distortions in the binary components of the vectors are independent. If $\| i \|$ is a number of ones in the binary representation of $i$ ($i = 0, 1, ..., 2^m-1$), then we have for independent errors

$$P_i = \left( \frac{m}{i} \right) p_i^i (1-p_i)^{m-i}.$$

(5)

where $p_i$ is a probability of a distortion of one bit, and for $n = 2^m - 1$ we have [6, 7]

$$P_{AL} = 2^{-m} \left[ 1 - 2^{-m}(1-2p) + 2^{-m} \right] - (1-p)^{2^m-1}.$$

(6)

For errors of a given magnitude [6]

$$P_i = \begin{cases} 1-p, & i = 0; \\ p, & i = a; \\ 0, & \text{otherwise}; \end{cases} \quad \text{for some } a \in Z_q - 0$$

(7)

and for $n = 2^m - 1$

$$P_{AL} = 2^{-m} \left[ 1 - (2^m - 1)(1-2p) + 2^{-m} - (1-p)^{2^m-1} \right].$$

(8)

For large $n$ (say, $n > 50$) for all three models $P_{AL} \rightarrow 2^{-m}$. Analysis of aliasing probabilities for benchmark circuits [6] shows that for small $n$ the independent error model predicts minimum aliasing, and the symmetrical model predicts aliasing more accurately than the other models.

Let us consider now the case $r > 1$. In this case we select the following system of $r$ $q$-ary functions

$$Z_j = H_j(y_0, y_1, ..., y_{m-1}) = \sum_{i=0}^{m-1} y_i \alpha^j_i \quad \text{for } j = 0, 1, ..., r-1$$

(9)

(\( \alpha \) is a primitive in $Z_q$, $q = 2^m$).

This approach is known as multisignature analysis, and it was also used for offline testing and diagnosis [11, 12, 13]. The hardware implementation of this approach requires $r$ $m$-bit LFSRs with characteristic equations $Z^{(m)} = \alpha^{r} Z^{0} + y_{r-1}, Z^{0} = 0, Z^{j} = Z_{j}; j = 0, 1, ..., r-1 (10)$

In this case $Z_0, Z_1, ..., Z_{r-1}$ defined by (9) are the first $r$ coefficients of the Fourier expansion of $y = (y_0, y_1, ..., y_{m-1})$ over $Z_q$. The network computing $Z_0, Z_1, ..., Z_{r-1}$ (r m-bit LFSRs) computes syndromes for the $q$-ary $(n, n-r)$ Reed-Solomon codes of length $n$ with distance $r+1$. Thus, (9) provides for the minimal solution of $P_{1}$ for the case when at most $r$ components of $e = (e_0, e_1, ..., e_{n-1})$ are not equal to 0, i.e., at most $r$ components of $y = (y_0, y_1, ..., y_{m-1})$ are distorted $\| e \| > r\). For a large $n$ we have for the above error models $P_{AL} \rightarrow 2^{-m} = \gamma^{-r}$ for all $e$ such that $\| e \| > r$. (For $P_{AL} = 0$ for $0 < \| e \| \leq r$).

Let us consider a more general case when the set, $E$, of errors is defined by the topology of the system.

Let $X$ be the set $\{ X_0, X_1, ..., X_{n-1} \}$ of $N$ processing elements. Consider a digraph $G$ having $X$ as a set of vertices and a set $U = \{ U_0, U_1, ..., U_{M-1} \}$ of directed edges (m-
bit communication links) between vertices of $G$. We shall also assume that the graph has no cycles and all $n$ output vertices are reachable from at least one input vertex. Let $y(r) = (y_0, y_1, ..., y_{n-1}) \in Z_q^n$ be an output vector for the system represented by the graph $G$, where $y_i \in Z_q$ is an output of the corresponding output vertex $i = 0, 1, ..., n-1$.

The problem to be considered is the problem of error detection under the assumption of single vertex failures in the graph $G$. A failure in the graph (system of processing elements) refers to a physical malfunction that causes an undesired event. We consider a fault in the graph which alters its output value to $\hat{y} = (\hat{y}_0, \hat{y}_1, ..., \hat{y}_{n-1})$, where $\hat{y}_i \in Z_q$.

The error in the graph's output $y$ can be characterized by the error vector $e = (e_0, e_1, ..., e_{n-1})$ where $e_i = \hat{y}_i - y_i$ for $i = 0, 1, ..., n-1$.

Let us first define an error set $E(G)$ characterized by the underlying graph $G$. In our definition of an error set we assume that at most one vertex or any number of incoming edges to this vertex may fail and a fault in the graph manifest itself by distorting all successor vertices outputs, i.e. error propagates along a directed path.

Let $E_j = \{(e_0^{(j)}, e_1^{(j)}, ..., e_{n-1}^{(j)})\} (j = 0, ..., N-1)$ be a set of error patterns corresponding to a fault in vertex $X_j$, where $e_i^{(j)} \in Z_q - 0$ if there exists a directed path from $X_j$ to an output and $e_i^{(j)} = 0$ otherwise. The set $E(G) = \bigcup_{j=0}^{N-1} E_j$ of all possible error patterns corresponding to all single vertex failures in $G$ called the error set for $G$.

To illustrate the above definition let us now consider the $p$-ary star network topology (see Fig. 1). For the $p$-ary star, the single central processing element (root) is connected to all others, $N = p+1$ and $n = p$. Due to single vertex or processing element failures we have the following nonzero errors in the $p$-ary star:

$$E(G) = \begin{bmatrix} e_0, & e_1, & ..., & e_{p-1} \\ e_0, & 0, & ..., & 0 \\ 0, & e_1, & ..., & 0 \\ 0, & 0, & ..., & e_{p-1} \end{bmatrix},$$

(11)

where $e_i \in Z_q - 0$. Thus, we have $|E(G)| = (q-1)^p + p(q-1)$ nonzero error vectors for $p$-ary star over $Z_q$.

Let $Z = Z_q^n$, $H$ is a $(r \times n)$ $q$-ary transform matrix over $Z_q \{y \in Z_q^n, \ Z \in Z_q^r\}$. Consider a graph, $G(E)$, having the error set $E(G) = \{0 \in E(G)\}$ as a set of vertices and $U = \{(E_i, E_j) | E_i, E_j \in E(G), \ E_i + E_j \in E(G)\}$ as a set of edges. Denote by $\chi(E)$ the chromatic number for $G(E) (\chi(E))$ is a minimal number of colors required to color vertices of $G(E)$ in such a way that no two neighboring vertices have the same color; techniques for graph coloring with lower and upper bounds for $\chi(E)$ can be found e.g. in [14]).

Then using results from [14, 15] we have for a minimal number $r_{\min}$ of transform coefficients required for detection of $E(G)$:

$$\text{min}\left[n \left[-\log_q(q^n - |E(G)|)\right], \log_q(\chi(E) + 1)\right] \leq r_{\min} \leq \left[\log_q(\left|E(G)\right|(q-1)+1)\right]$$

(12)

We will present now solutions for the data compression problem P1 for the two important topologies: trees and Fast Fourier Transform (FFT) networks.

Let $T_p$ be a $p$-ary full tree of height $h$ ($p \geq 2, h \geq 2$). The height of the tree is the length of a longest path from the root to any leaf. (Here we assume that input vertex is the root and output vertices are $n = p^h - 1$ leaves of the tree). Then

$$|E(G)| = \sum_{i=0}^{p^h-1} p^i(q-1).$$

(13)

For example, 4-ary tree $T_4$ is represented in Fig. 2.
For this tree the error set \( E(G) \) is:

\[
E(G) = \{
(e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7),
(e_0, e_1, e_2, e_3, 0, 0, 0, 0),
(0, 0, 0, 0, e_4, e_5, e_6, e_7),
(0, 0, 0, 0, 0, 0, 0, 0),
(e_0, e_1, 0, 0, 0, 0, 0, 0),
(0, 0, 0, 0, e_4, e_5, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0),
\}.
\]

where \( e_i \in \mathbb{Z}_q - 0 \).

It is possible to show [15, 16], that for a \( p \)-ary full tree with
\[
h \leq \log_p \left( 1 + \left( \frac{q-1}{q} \right)^{-1} \right)
\]
for any \( p, r_{\min} = h \) and
\[
H = H^{(h)} = \begin{pmatrix}
H^{(h-1)} & H^{(h-1)} & \cdots & H^{(h-1)}
\end{pmatrix}
\]
where \( W \) is a row of one 1 followed by \( n-1 = p^h - 1 \) zeros, and
\[
H^{(2)} = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & 0 & 0 & \cdots & 0
\end{bmatrix}
\]

For example, for \( T_3 \) represented by Fig. 2 (\( p=4, \ h=3 \)) we have

\[
H = H^{(3)} = \begin{bmatrix}
1111 & 1111 & 1111 & 1111 \\
1000 & 1000 & 1000 & 1000 \\
1000 & 0000 & 0000 & 0000
\end{bmatrix}
\]

For the \( n \)-point FFT, there are \( N = n \log_2 n \) vertices interconnected with \( \log_2 n \) levels of butterfly structures, e.g., the graph for the 8-point FFT with decimation-in-frequency (DIF) is shown in Fig. 3.

If we consider input fanout branches as possible source of errors, then there are \( n(\log_2 n + 1) \) single faults in the \( n \)-point FFT graph. Due to these single faults we have the following nonzero errors for the 8-point FFT graph of Fig. 3:

\[
E(G) = \begin{bmatrix}
(e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7),
(e_0, e_1, e_2, e_3, 0, 0, 0, 0),
(0, 0, 0, 0, e_4, e_5, e_6, e_7),
(0, 0, 0, 0, 0, 0, 0, 0),
(e_0, e_1, 0, 0, 0, 0, 0, 0),
(0, 0, 0, 0, e_4, e_5, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0),
\end{bmatrix}
\]

where \( e_i \in \mathbb{Z}_q - 0 \) and
\[ |E(G)| = \sum_{r=0}^{n} 2^{r} (q-1)^{2^{r}}, n = 2^{r}. \]

The recursive construction for matrices \( H \) for DIF FFT topologies is given by
\[
H = H^{(n)} = \begin{bmatrix} H^{(n/2)} & H^{(n/2)} \\ W & \end{bmatrix},
\]
where \( W = 100\ldots0 \) and \( H^{(2)} = \begin{bmatrix} 11 \\ 10 \end{bmatrix} \).

For example, for the 8-point DIF FFT graph represented by Fig. 3:
\[
H = H^{(8)} = \begin{bmatrix} 11 & 11 & 11 & 11 \\ 10 & 10 & 10 & 10 \\ 10 & 00 & 10 & 00 \\ 10 & 00 & 00 & 00 \end{bmatrix}
\]
(A construction similar to (16) can be used for FFTs with decimation-in-time [16].)

Thus, for an \( n \)-point FFT we have
\[
\tau_n = \log_2 n + 1.
\]

The flow graph of the Walsh-Hadamard Transform (WHT) [3, 4] differs from the flow graphs of the FFT only in the values of twiddle factors \( \pm 1 \) for WHT. Thus, solution (16) is valid also for networks computing WHT. Solutions for multidimensional FFTs and for Fast Chrestenson Transforms can be found in [16].

To conclude this section we note that elements of matrices \( H \) for trees and FFTs are zeros and ones only.

II. Robust Quadratic Transforms for Testing of Computer Systems.

In the previous section we considered linear data compressors \( Z = Hy, (Z \in Z_q^n, y \in Z_q^n) \). For any linear transform we have for the probability, \( P_{\text{Al}}(e) \), of masking (aliasing probability) for a given error pattern \( e \)
\[
P_{\text{Al}}(e) = \text{Prob} \{ H(y + e) = H(y) \}
\]
\[
= \text{Prob} \{ He = 0 \} = \begin{cases} 1, & e \in \text{Kern } H; \\ 0, & e \notin \text{Kern } H. \end{cases}
\]

Thus, performances of linear compressors are very sensitive to error distributions
\[
P = \{ P_0, P_1, \ldots, P_{M-1} \} \{ M = q^n, P_i = \text{Prob} \{ e = i \} \}.
\]

We will describe in this section a class of nonlinear (quadratic) \textit{robust} transforms, such that their performances do not depend on error distributions. We will show that for optimal robust transforms described by \( r \) quadratic \( q \)-ary functions, we have
\[
P_{\text{Al}}(e) = q^{-r} \text{ for any } e \neq 0.
\]

Suppose \( n = 2rs \). Consider the following nonlinear mapping \( Z_q^n \rightarrow Z_q^r \)
\[
Z = (Z_0, \ldots, Z_{r-1}) = H(y_0, \ldots, y_{r-1}) = Y_0Y_1 + Y_2Y_3 + \ldots + Y_{2^{r-2}}Y_{2^{r-1}},
\]
where \( Y_i = (y_{ir}, \ldots, y_{i(r-1)s}) \) \( (i = 0, \ldots, 2^r-1) \), and all operations in the quadratic form (19) are in \( Z_q^r \).

For \( q = 2 \) and \( r = 1 \) these quadratic forms are known as bent functions [3, 5, 18, 19, 20, 21]. In this case, \( n = 2s, Y_i = y_i \), \( Z = y_0Y_1 + y_2Y_3 + \ldots + y_{n-2}Y_{n-1} \), and we have for the autocorrelation for bent functions
\[
B(e) = \sum_{y \in Z_q^n} H(y)H(y + e) = \begin{cases} 2^{n-1} - 2^{n/2-1}, & e = 0; \\ 2^{n-1} - 2^{n/2-1}, & e \neq 0. \end{cases}
\]

Thus, \( B(e) = \text{Const.} \) (We note that \( B(e) \) can be computed by applying twice the Walsh-Hadamard Transform using the Wiener-Khintchin theorem [3]).

For any linear function \( f(y_0, \ldots, y_{r-1}) \) \( (y_i \in Z_q) \) such that \( f(y_0, \ldots, y_{r-1}) = 1 \), when \( (y_0, \ldots, y_{r-1}) \in V \) and \( f(y_0, \ldots, y_{r-1}) = 0 \) otherwise, where \( V \) is a subspace of \( Z_q^n \), we have for its autocorrelation
\[
B(e) = \sum_{y \in Z_q^n} f(y)f(y + e) = \begin{cases} 0, & e \in V; \\ 1, & e \notin V. \end{cases}
\]

Since for bent functions defined by (20), we have \( B(e) = \text{Const.} \), one can say that bent functions are at the maximal distance from any linear function [5].

Let us introduce a system of characteristic functions
\[
h_i(y) = 1 \text{ iff } H(y) = i.
\]

The autocorrelations functions \( B_i(e) \) for \( h_i(y) \) can be defined as
\[
B_i(e) = \sum_{y \in Z_q^n} h_i(y)h_i(y + e) = \begin{cases} 0, & H(y) = H(y + e) = i; \\ 1, & \text{otherwise.} \end{cases}
\]

(Autocorrelation functions \( B_i(e) \) can be computed by the Wiener-Khintchin theorem using the Chrestenson transform over \( Z_q^n \) [3, 4].)
It is possible to show [22, 23] that for \( i \neq 0 \)
\[
B_i(e) = \begin{cases} 
q^{2i-r} - q^{-r}, & e = 0; \\
q^{2i-r} \pm q^{-r}, & e \neq 0; 
\end{cases}
\]
and we have for the total autocorrelation
\[
B(e) = \sum_i B_i(e) = \left( \sum_i y_i H(y) = H(y+e) \right) =
\begin{cases} 
q^{2r}, & e = 0; \\
q^{2n-2r}, & e \neq 0; 
\end{cases}
\]
where \( H(y) \) defined by (19).

Thus, quadratic forms defined by (19) have the flat total autocorrelation and compressors implementing these forms are robust with a probability of masking any error
\[
P_{M}(e) = q^{-r} = 2^{-2r}.
\]

Let \( C \) be a quadratic q-ary error-detecting code defined as \( y \in C \) if
\[
H(y) = y_0 + y_2 + \ldots + y_{2n-2} + 1,
\]
where \( Y_j = (y_{j,1}, y_{j,2}, \ldots, y_{j,2n-2}) \) and all operations in (29) are in \( Z_q \). Then \( C \) is a code with the length \( n = 2rs \) and the number of codewords \( |C| = q^{22r-2} - q^{-2r} \). These codes provide for an equal protection against all possible errors for large \( q \) or large \( n \) and are optimal for the minimax criterion on error detection [22].

For a given block size, \( n \), and the number of codewords, \( |C| \), these codes minimize max \( Q(e) \), where \( Q(e) \) is the conditional error masking probability given the error pattern \( e \). For these codes for any \( e \neq 0 \)
\[
(2) \quad (q^{2n-2r} - q^{-r}) \left( q^{2r-2} - q^{-r} \right)^{-1} \leq Q(e) \leq (q^{2n-2r} + q^{-r}) \left( q^{2n-2r} + q^{-r} \right)^{-1}
\]
and
\[
Q(e) \sim q^{-r},
\]
for large \( n \) or large \( q \).

Simple encoding and decoding procedures for these codes are presented in [22].


Let \( y = (y_0, y_1, \ldots, y_{2n-1}) \), \( y_i \in Z_q, q = 2^m \) be a test response for a system-under-test. As a result of faults in the system \( y \) may be distorted into \( y + e \), where \( e = (e_0, \ldots, e_{n-1}), e_i \in Z_q, e \in E(G) \), and the set of errors, \( E(G) \), is defined by the topology of the system. Denote
\[
\text{supp}(e) = \{ \text{supp}(e_0), \ldots, \text{supp}(e_{n-1}) \},
\]
where
\[
\text{supp}(e_i) = \begin{cases} 
1, & e_i \neq 0; \\
0, & e_i = 0.
\end{cases}
\]

The soft decision diagnosis problem can be formulated in the following way. For a given \( E(G) \subseteq Z_n \) construct a transform \( Z = H(y) \) over \( Z_q \)
\[
Z_j = H(y_0, y_1, \ldots, y_{2n-1}, \ldots, y_{j-1}, y_{j+1}, \ldots, y_{2n-1}) \quad (j = 0, 1, \ldots, r-1)
\]
with minimal \( f \) such that
\[
H(y + e^{(1)}) \neq H(y + e^{(2)})
\]
for any \( e^{(1)} , e^{(2)} \in E(G) \) and \( \text{supp}(e^{(1)}) \neq \text{supp}(e^{(2)}) \).

We will consider the case when \( H \) is linear. Then (32) can be written as:
\[
H(e^{(1)}) \neq H(e^{(2)})
\]
for any \( e^{(1)}, e^{(2)} \in E(G), \text{supp}(e^{(1)}) \neq \text{supp}(e^{(2)}) \) and all operations are in \( Z_q \). We will call \( H(y) \) a fault-free signature, \( H(y + e) \) a faulty signature and \( S = He \) a syndrome of error \( e \).

If (33) is satisfied, then the location, \( \text{supp}(e) \), of an error \( e \) can be computed by analysis of the distortion \( H(y) + H(y + e) \) in the fault-free signature. The bounds on a minimum number of spectral coefficients in transforms satisfying (33) can be obtained from bound (12) for error detection by replacing the error set \( E(G) \) with
\[
E(G) \cup \left\{ e^{(1)} + e^{(2)} \right\} \overset{\text{supp}(e^{(1)}) \neq \text{supp}(e^{(2)})}{\neq \text{supp}(e^{(2)}) \cup \text{supp}(e^{(2)}) \in E(G) \}
\]

We note that any matrix \( H \) satisfying (33) is a parity check matrix of a linear \( q \)-ary \((n, n-r) \) code locating the error set \( E(G) \). We note also that a linear \((n, n-r) \) code with a parity check matrix \( H \) locating an error set \( E(G) \), \( 0 \in E(G) \) corrects \( E(G) \) if and only if for every distinct \( e^{(1)} , e^{(2)} \in E(G) \) with \( \text{supp}(e^{(1)}) \neq \text{supp}(e^{(2)}) \), there exists a pair \( e^{(3)}, e^{(4)} \in E(G) \cup \{ 0 \} \) with \( \text{supp}(e^{(3)}) \neq \text{supp}(e^{(4)}) \) such that
\[
e^{(1)} + e^{(2)} = e^{(3)} + e^{(4)}.
\]

We note that the above condition is a necessary and sufficient condition on the error set \( E(G) \) such that if and only if this condition is satisfied any linear code locating \( E(G) \) will also correct \( E(G) \).
For example, for the p-ary star with \( p = 5 \) and \( q = 2^2 \),
\[
E(G) = \{(e_0, e_1, e_2, e_3, e_4), (e_0, 0, 0, 0, 0), (0, e_1, 0, 0, 0),
(0, 0, e_2, 0, 0), (0, 0, 0, e_3, 0), (0, 0, 0, 0, e_4)\},
\]
where \( e_i \in \{0 = 1, \alpha = 10, \alpha^2 = 11\} \). A code locating \( E(G) \) does not guarantee error correction since for two errors \( e^{(1)} = (1, 1, 1, 1, 1), e^{(2)} = (1, 1, \alpha, \alpha, \alpha) \) there is no pair \( e^{(3)}, e^{(4)} \) with different support such that \( e^{(1)} + e^{(2)} = e^{(3)} + e^{(4)} = (0, 0, 0, \alpha^2, \alpha^2, \alpha^2) \) (note that \( \alpha^2 + \alpha + 1 = 0 \)).

Using the above arguments one can see that any code over \( Z_q \) locating up to \( l \) independent errors
\[
\{E(G) = \left\{ e| 0 < \| e \| \leq 2l \right\} \}
\]
can also correct \( l \) errors. The same is also true for codes locating burst errors.

Let us consider now the case when at most \( l \) components in
\( y = (y_0, \ldots, y_{n+1}) \) may be distorted
\[
\{E = \{ e| 0 < \| e \| \leq l \} \}. 
\]
In this case \( H \) can be taken as the check matrix of the \( q \)-ary Reed-Solomon code of length \( n \) [5]
\[
H = \begin{bmatrix}
1 & 1 & 0 & \cdots & 0 \\
1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\
1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\
1 & \alpha^{r-1} & \cdots & \alpha^{(r-1)(n-1)} \\
\end{bmatrix}
\]
where \( \alpha \) is primitive in \( Z_q \), \( n \leq q - 1 \) and \( r = 2l \). For this matrix any \( r \) columns are linearly independent over \( Z_q \).

Computing \( Z = H y \), where \( H \) is defined by (34), requires \( n \) clocks and \( r \) LFSRs with parallel input. An error locating procedure for this case identifies \( \text{supp}(e) \) for a given \( H e \). This procedure based on the Euclidean algorithm [12] has a complexity \( L = O(n m) + O(\log_2 n) \). (\( L \) is a number of equivalent two-input gates, \( q = 2^m \)). For example, for \( n = 100, m = 32 \) and \( l = 5 \) we have \( L = 28,000 \) [12].

We note also that \( H \) defined by (34) with \( r = 2l \) can be used for location of more than \( l \) errors. It was shown in [13] that if \( \| e \| = t > l \), then the fraction \( w \) of localizable errors with \( \| e \| = t, l < t < 2l \), is low-bounded by
\[
w \geq (1-q^{-1})^t \frac{\binom{n}{t}}{q^t} - e^{-t} \left(\frac{q}{r}\right)^{t-1}.
\]

For example, if \( t=5, r=6, n=100, q = 2^2 \) (\( m=32 \)), then \( w = 0.979 \). Thus, by allowing a small fraction of errors not to be located, one can reduce substantially the required redundancy from \( r = 2l \) to \( r = l + 1 \).

Let us consider now the error-locating problem when the original system is the full \( p \)-ary tree of height \( h (n = p^{h-1}) \) (the \( 4 \)-ary full tree of height \( h=3 \) is presented in Fig. 2, the error set for this tree is given by (14)).

The following recursive procedure can be used for construction of \( H = H^{(h)} \) for \( p \)-ary trees [16].
\[
H^{(h)} = \begin{bmatrix}
1 & 1 & \cdots & 1 & 1 \\
1 & \alpha & \cdots & \alpha^{h-1} & \alpha^{h-1} \\
1 & \alpha^2 & \cdots & \alpha^{2h-1} & \alpha^{2h-1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
1 & \alpha^{r-1} & \cdots & \alpha^{(r-1)(h-1)} & \alpha^{(r-1)(h-1)} \\
\end{bmatrix}_{p \times p}
\]
(36)

where
\[
H^{(2)} = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \alpha & \alpha^2 & \cdots & \alpha^{p-1} \\
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
\end{bmatrix}_{p \times p}
\]
(37)

\( 4 \leq p \leq q \), and \( \alpha \) is primitive in \( Z_q \). Thus, for any \( p \)-ary, \( (4 \leq p < q) \) tree \( r = 3h - 2 \). For the case of \( p = 3 \) one can also use the above construction (36) with \( H^{(2)} = I_3 \) (3 is and identity matrix of 3 by 3). Therefore, for the \( p \)-ary tree with \( p = 3, q > p, r = 3h - 3 \).

For binary trees over \( Z_q \) the following recursive construction can be used:
\[
H^{(h)} = \begin{bmatrix}
1 & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 \\
\end{bmatrix}_{p \times p}
\]
(38)

where \( H^{(2)} = I_2 \). Thus, for the binary tree over \( Z_q \) of height \( h, r = 2h - 2 \).

The complexity \( L \) for the syndrome computing network implementing \( H^{(h)} y \) in terms of submers of \( Z_q \) adders and multipliers is [16]:
\[
L = (2(p^{h-1} - p) + h(p - 1))L_a + ((h-1)(p-1))L_a
\]
(39)
where $L_{\phi}$ is a complexity of a multiplier that multiplies a field element from $Z_q$ by a fixed element from the same field. This network for $h = 3$ is represented at Fig. 4.

The error locating procedure for tree errors is very simple. Let us denote

$$S = H^{(h)}y = \begin{bmatrix} S_{h-1}^\top \\ S_1^\top \\ S_2^\top \\ S_3^\top \end{bmatrix},$$

(40)

where $S_{h-1}^\top$ are syndromes due to the $[H^{(h)}; H^{(h-1)}; ...; H^{(1)}]$ part of $H^{(h)}$ (see (36)) and $S_1^\top, S_2^\top, S_3^\top$ are syndromes for the last three rows of the parity check matrix $H^{(h)}$. Let $S_1^\top$ denote the syndrome for the all 1 row.

The location algorithm to find a faulty vertex is described as follows:

1. If \( S_i = 0, i = 1, 2, ..., 3h - 2 \), no error, end.
2. Let \( j = h \).
3. If both \( S_j^\top \neq 0 \) and \( S_j^\top \neq 0 \), error location is the root of the tree of height \( j \), end.
4. For \( j > 2 \), if either \( S_j^\top = 0 \) and \( S_j^\top = 0 \), error location is in the subtree \( k, 0 \leq k \leq p - 1 \), where \( \alpha^k = S_j^{i-1} / S_j^\top \).
5. For \( j = 2 \), if either \( S_1^\top = 0 \) or \( S_2^\top = 0 \), error location is in vertex (leaf) \( k, 0 \leq k \leq p - 1 \), where \( \alpha^k = S_1^\top / S_2^\top \).
6. End.

The transform matrix $H_3$ for the 4-ary tree of height $h = 3$ is

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

(41)

Suppose that a root of subtree 1 (see Fig. 2) is faulty and $e = (0, 0, 0, 0, \alpha, 1, \alpha^2, 0, 0, 0, 0, 0, 0)$. Then

$$S = H^{(3)}e = (\alpha, \alpha^5, \alpha, \alpha^2, 0, \alpha^3).$$

(42)

This yields \( S_3^3 = 0 \), therefore the error is in the subtree 1 since \( S_1^2 / S_2^3 = \alpha^2 / \alpha = \alpha \), \( i = 1 \). Since \( S_2^3 \neq 0 \) and \( S_3^3 \neq 0 \), error is in the root of subtree 1 of height 2.

We note that the number, \( r \), of rows in $H^{(h)}$ for $p$-ary trees of height $h$ can be drastically decreased (from $r = 3h - 2$ to $r = h$) if we allow a small probability of misdiagnosis. In this case one can take [16]

$$H^{(h)} = \begin{bmatrix} H^{(h-1)} & H^{(h-1)} & ... & H^{(h-1)} \\ 11...1 & \alpha \alpha \alpha \alpha ... & \alpha p^{-1} \alpha p^{-1} \alpha p^{-1} \alpha p^{-1} \end{bmatrix}$$

(43)

where

$$H^{(2)} = \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & \alpha & \alpha^2 & ... & \alpha^{n-1} \end{bmatrix} n=p<q$$

Fig. 4 The network for computing $S = H^{(3)}y$

The number of rows in $H^{(h)}$, defined by (44) is equal to $h$. The location procedure in this case is very simple, and the probability of correct diagnosis for this procedure can be estimated as [16]

$$w = 1 - \left(1 - pq^{-1}\right)^{h-1}$$

(45)

which is small for large $q$.

**IV. Spectral Techniques for Hard Decision Diagnosis of Computer Systems.**

The hard decision diagnosis problem can be formulated in the following way. For a given $E(G) \subseteq Z_q^n$ construct a binary
matrix $H$ with a minimal number of rows such that for any $e^{(1)}, e^{(2)} \in E(G)$ with $\text{supp}(e^{(1)}) \neq \text{supp}(e^{(2)})$

$$\text{supp}(He^{(1)}) \neq \text{supp}(He^{(2)}),$$

where

$$\text{supp}(Z = Z_0, \ldots, Z_n) = (\text{supp}(Z_0), \ldots, \text{supp}(Z_n)).$$

The hard decision approach allows identification of errors $e$ by analyzing supports of their syndrome $S=He$. Magnitudes of distortions in components of a syndrome are not important for the hard decision diagnosis. The identification of locations of errors is possible if there is a one-to-one mapping

$$E(G) \leftrightarrow \{\text{supp}(He) | e \in E(G)\}.$$ 

In general, this approach requires more rows in $H$ (more observation points for the compressed response $Z=Hy$) but a decoding procedure is very simple and has a straightforward hardware implementation.

Denote $H \otimes \text{supp}(e)$ the Boolean multiplication of an $(r \times n)$ binary matrix $H$ by an $n$-bit binary vector $\text{supp}(e) = (\text{supp}(e_0), \ldots, \text{supp}(e_n))$ with addition being replaced by logical summation (OR).

Then with probability $1 - q^{-1}$ if

$$H \otimes \text{supp}(e^{(1)}) \neq H \otimes \text{supp}(e^{(2)}),$$

then $He^{(1)} \neq He^{(2)}, e^{(1)}, e^{(2)} \in E(G) \subseteq Z_q^n$ and $\text{supp}(e^{(1)}) \neq \text{supp}(e^{(2)}).$

Let us consider now the hard decision diagnosis problem for the case of $l$ independent errors $E = \{e | 0 < \| e \|_1 \leq l\}$. We note that in this case any check matrix of a $l$-th order binary superimposed code $[24, 25, 26, 27]$ can be chosen as the transform matrix $H$.

A binary superimposed code of order $l$ consists of a set of codewords such that componentwise Boolean sum (OR) of any $l$ codewords differs from every other componentwise sum of $l$ or fewer codewords.

Thus, in a view of (47), any check matrix of a $l$-th order linear superimposed code can be chosen as a hard decision diagnostic matrix $H$.

Since there are $\binom{n}{l}$ different locations of $l$ errors, we have the following lower bound on the minimal number $r=r(n,l)$ of required spectral coefficients (rows in $H$)

$$r(n,l) \geq \left\lceil \log_2 \binom{n}{l} \right\rceil.$$ 

(48)

For the case of single errors ($l = \|e\|_1 = 1$), one can take as $H$ any binary matrix with different nonzero columns. Thus

$$r(n,1) = \left\lceil \log_2 (n+1) \right\rceil.$$ 

(49)

The case of multiple errors ($l > 1$) is not as simple and it is difficult even to estimate $r=r(n,l)$. Several good constructions of check matrices for linear superimposed codes can be found in $[24, 25, 26]$.

A hardware implementation of the hard decision diagnostic algorithms for $l = 1, n = 100, q = 2^{32}$ requires about 10,000 equivalent two-input gates $[27]$. Let us consider now a general case of hard decision diagnosis, when the error set $E(G)$ is defined by the topology $G$ of the system.

Let $N$ be the number of processing elements in the systems (nodes in $G$) and $d$ is the length (number of nodes) of the longest path in $G$.

Then we have the following attainable bounds on a minimal number of rows in $H$

$$\max \left( \left\lceil \log_2 (N + 1) \right\rceil, d \right) \leq r \leq n.$$ 

(50)

These bounds can be improved if we have additional information about the topology of the system.

Let $N(j)$ be the number of paths of length at least $j$ which do not have any endpoints in common. The following two lower bounds on $r$ have been proven in $[28, 29]$:

$$\sum_{i=1}^{R-l+1} \binom{r}{i} \geq N(j),$$

(51)

$$\sum_{i=1}^{R-d+1} \binom{r}{i} + \sum_{i=1}^{R-(j+1) \cdot d+1} \binom{r}{i} \geq M(j).$$

(52)

Lower bounds (51), (52) are valid for all values of $j = 1, 2, \ldots, d$.

For the $p$-ary full tree of height $h$ we have $d = h$, $n = p^{h-1}$,

$$N = \frac{p^h - 1}{p-1} \quad \text{and} \quad N(d) = p^{d-1}.$$ 

Thus, by (50), (51), we have the following lower bounds for the minimal number $r = r(d)$ of spectral coefficients required for hard diagnosis of $p$-ary trees of height $h = d$.

$$r(d) \geq \left\lceil \log_2 \left( \frac{p^{d-1} - 1}{p-1} + 1 \right) \right\rceil.$$ 

(53)
\[
\sum_{i=1}^{\lfloor r(d) \rfloor} \left( \begin{array}{c} r(d) \\ i \end{array} \right) \geq p^{d-1}.
\]  
(54)

Solving (54) for \( d \gg 1 \) we have asymptotically
\[
r(d) > \begin{cases} 
1.29(d-1), & p = 2; \\
1.64(d-1), & p = 3; \\
(\log_2 p)(d-1), & p \geq 4.
\end{cases}
\]  
(55)

We note, that for \( p \geq 4 \) and \( d \gg 1 \) (53) gives better lower bounds than (55).

The best known upper bounds on \( r \) for binary trees [29] are given by
\[
\begin{align*}
1.5(d-1), & \text{ for } d = 4r + 1; \\
1.5(d-1) + 1, & \text{ otherwise.}
\end{align*}
\]  
(56)

Values of \( r(d) \) for binary trees with \( N < 2^{12} \) are given in the following table:

<table>
<thead>
<tr>
<th>( d )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r(d) )</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>9-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( d )</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r(d) )</td>
<td>10-11</td>
<td>12</td>
<td>13-14</td>
<td>14-16</td>
<td>16-17</td>
</tr>
</tbody>
</table>

Table 1: Minimal Numbers, \( r(d) \), of Spectral Coefficients Required for Hard Diagnosis of Binary Trees of Height \( h = d \)

Optimal or near optimal constructions for \( [r(d) \times p^{d-1}] \) transform matrices \( H \) for \( p \)-ary trees \( h \), together with lower and upper bounds for \( r(d) \) for different \( p > 2 \) and
\[
N = \frac{p^2-1}{p-1} < 5,000
\]  
can be found in [28, 29]. The gap between bounds is small.

Constructions for optimal or near optimal transforms for hard diagnosis for two dimensional meshes and multidimensional cube topologies can be found in [29].

References


