

Coding Approaches to Fault-Tolerant Dynamic Systems

Christoforos Hadjicostis

Decision and Control Laboratory
Coordinated Science Laboratory and Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

MAIN RESEARCH THRUSTS

- **Monitoring and control of complex networked systems** (NSF ITR, NSF ECS)
 - Infrastructure constraints, fundamental limitations, layered architectures
 - Hardware, communication and/or algorithmic overhead
- **Fault-tolerant dynamic systems** (NSF Career, AFOSR URI)
 - System dynamics and structure, coding for protection
 - Special-purpose architectures (e.g., communication, signal processing)
- **Error control and noise-tolerance in digital sequential circuits** (NSF ITR)
 - Novel digital designs
 - Implications to speed, cost, power consumption
- **Fault-tolerant operation of energy processing systems** (NSF EPNES)
- **Soft-decision decoding, path diversity in networked systems**

DEFINITIONS AND MOTIVATION

Fault tolerance describes ability to

- Withstand internal faults
- Produce desirable overall “behavior” (e.g., correct or acceptable output)

Necessary or desirable in

- Life-threatening circumstances (military, transportation, medical)
- Systems in inaccessible environments (space missions)
- Reliable systems from unreliable components
(faster, less expensive, less power)

Fault monitoring implies ability to detect and identify faults (\Rightarrow **fault diagnosis**)

Exploits: (i) **Redundancy**

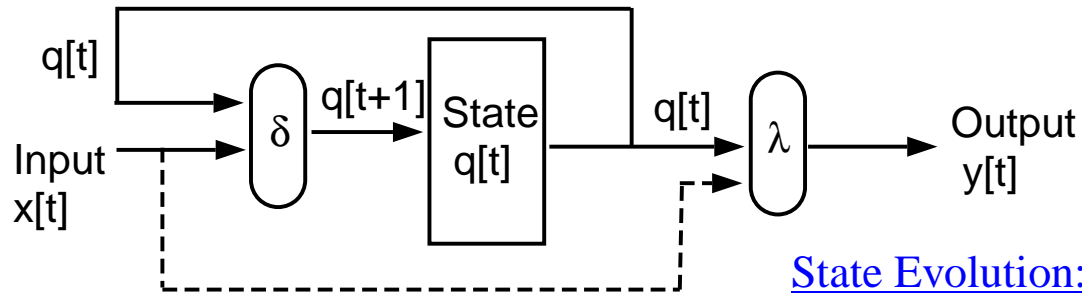
(ii) **Role of system dynamics and structure**

(iii) **Tradeoffs (detection delay, algorithmic complexity, redundancy)**

RELATED WORK ON FAULT TOLERANCE AND DIAGNOSIS

- **Communication systems (channel noise, error-correcting codes)**
- **Computational circuits (hardware faults, modular redundancy)**
 - Each component fails with *constant* probability
 - Earlier work by von Neumann, Shannon, Winograd, Elias and others; later work by Pippenger, Gács, Hajek, Feder, Reischuk
- **Special-purpose systems (Algorithm-Based Fault Tolerance)**
 - Protect against *fixed* number of faults, fault-free corrector
 - Recent work by Abraham et al., Redinbo, Musicus, Beckmann
- **Fault diagnosis in discrete event systems**
 - Observability limitations, distributivity constraints, complexity of diagnoser
 - Work by Teneketzis, Lafortune, Benveniste, Wonham, Holloway, Giua
- **Fault tolerance and monitoring in finite-state machines (concurrent check)**
 - Work by Reed, Redinbo, Kinney, Shen, Leveugle

DYNAMIC SYSTEMS

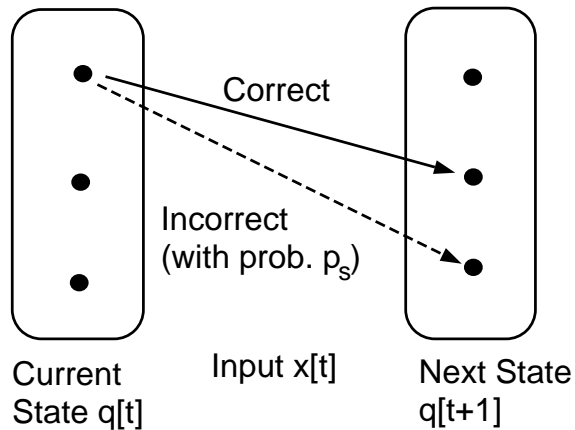


State Evolution: $q[t + 1] = \delta(q[t], x[t])$

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

Examples: Digital filters, encoders/decoders, FSMs, algorithmic computations

Faults in state transitions: Errors propagate in future time steps

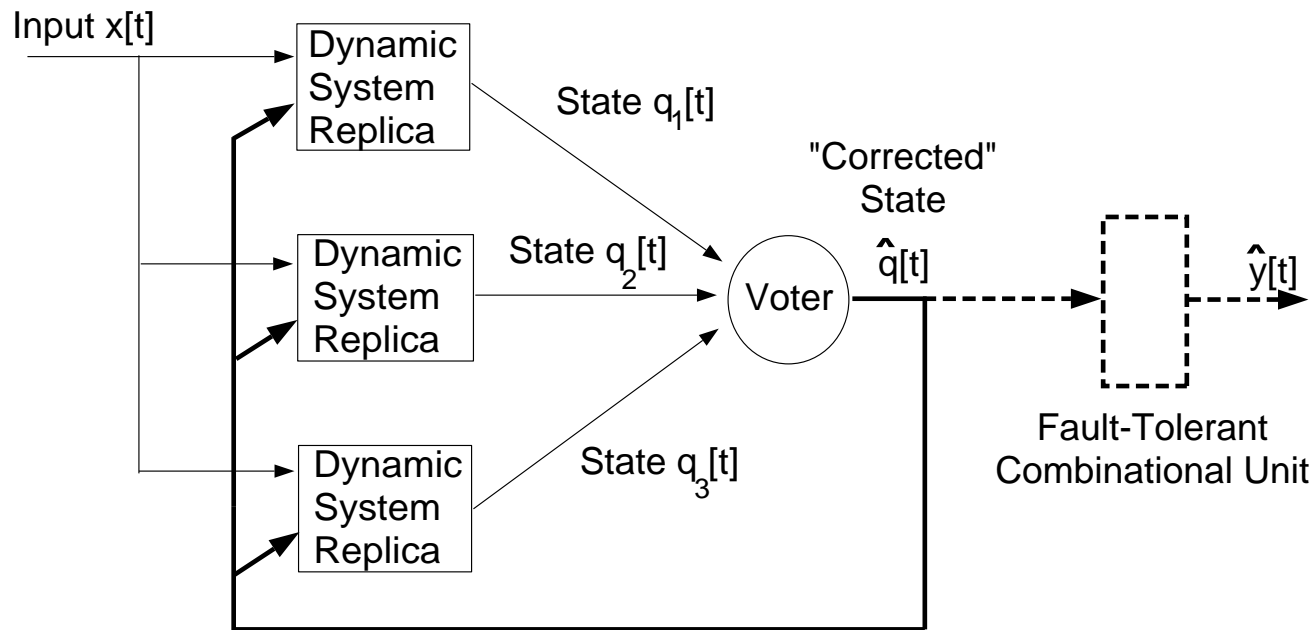


After N time steps:

$$\Pr[\text{correct state trajectory}] = (1 - p_s)^N$$

Error propagation is costly!

UNIVERSAL APPROACH: MODULAR REDUNDANCY (VON NEUMANN)

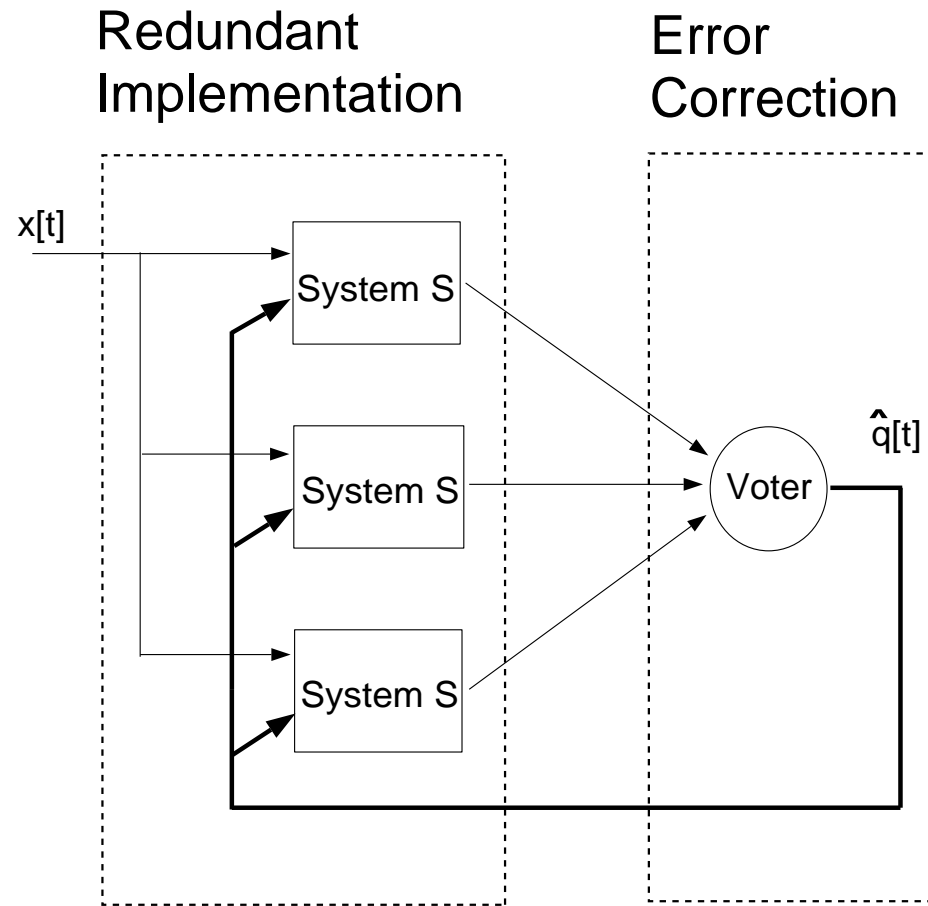


Problems with modular redundancy

- Replication
- Checking overhead/slowdown
- Reliability of checking mechanism

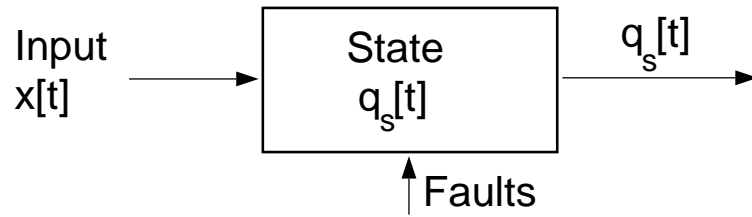
} Will be addressing these issues!

AVOIDING REPLICATION

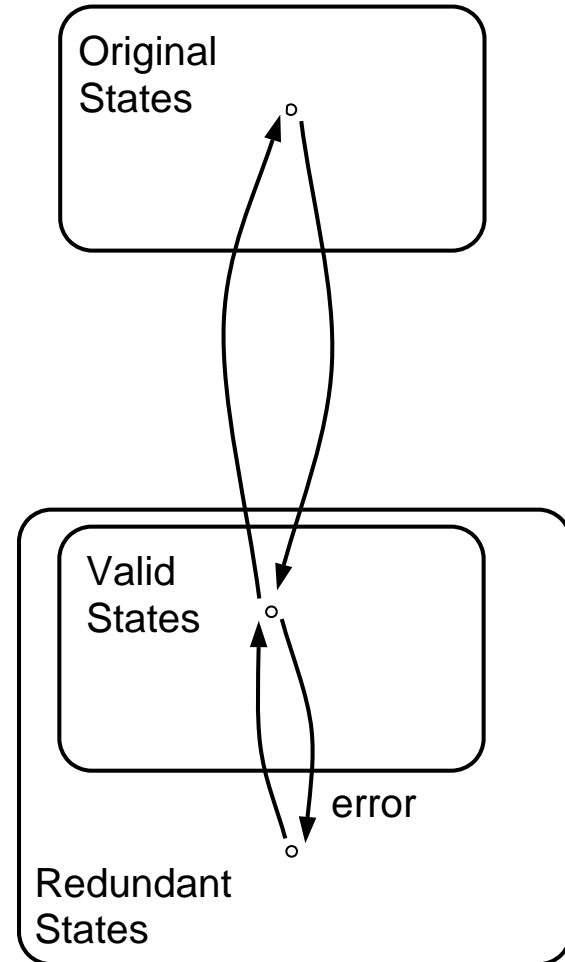
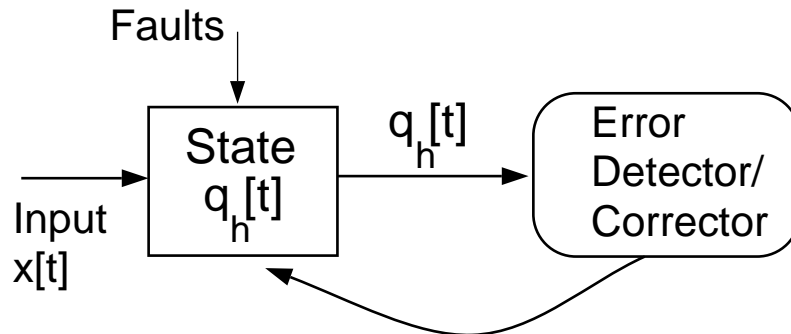


What are good alternatives to replication?

REDUNDANT IMPLEMENTATIONS

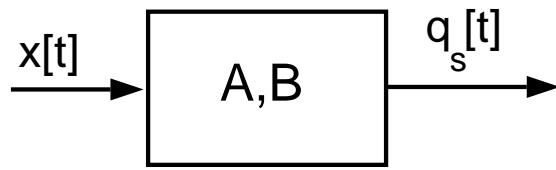


Replace with larger dynamic system:



REDUNDANT IMPLEMENTATIONS OF LTI DYNAMIC SYSTEMS

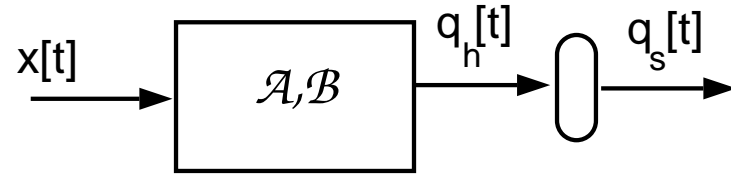
Original System



$$\mathbf{q}_s[t + 1] = \mathbf{A}\mathbf{q}_s[t] + \mathbf{B}\mathbf{x}[t]$$

\mathbf{q}_s is d -dimensional

Redundant Implementation



$$\mathbf{q}_h[t + 1] = \mathcal{A}\mathbf{q}_h[t] + \mathcal{B}\mathbf{x}[t]$$

\mathbf{q}_h is η -dimensional ($\eta = d + s$)

- **Concurrent simulation:** $\mathbf{q}_s[t] = \mathbf{L}\mathbf{q}_h[t]$
 - **Encoding constraints:** $\mathbf{q}_h[t] = \mathbf{G}\mathbf{q}_s[t]$
- } **Linear (not necessary)**
- **Fault detection:** If $\mathbf{q}_h[t]$ is *not* in the column space of \mathbf{G} , or

$$\mathbf{P}\mathbf{q}_h[t] \neq \mathbf{0},$$

$$\mathbf{P}\mathbf{G} = \mathbf{0},$$

\mathbf{P} has full-row rank s

CHARACTERIZATION OF REDUNDANT IMPLEMENTATIONS

<p><u>Original System</u></p> $\mathbf{q}_s[t + 1] = \mathbf{A}\mathbf{q}_s[t] + \mathbf{B}\mathbf{x}[t]$ <p>\mathbf{q}_s is d-dimensional</p>	$\left. \begin{array}{l} \xrightarrow{\mathbf{q}_h[t] = \mathbf{G}\mathbf{q}_s[t]} \\ \xleftarrow{\mathbf{q}_s[t] = \mathbf{L}\mathbf{q}_h[t]} \end{array} \right\}$	<p><u>Redundant Implementation</u></p> $\mathbf{q}_h[t + 1] = \mathcal{A}\mathbf{q}_h[t] + \mathcal{B}\mathbf{x}[t]$ <p>\mathbf{q}_h is η-dimensional ($\eta = d + s$)</p>
--	--	--

Standard redundant implementations (Hadjicostis & Verghese 1999):

$(\mathcal{A}, \mathcal{B})$ is a redundant implementation for (\mathbf{A}, \mathbf{B}) iff $(\mathcal{A}, \mathcal{B})$ is similar to

$$\mathbf{q}_\sigma[t + 1] = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix}}_{\mathcal{A}_\sigma} \mathbf{q}_\sigma[t] + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\mathcal{B}_\sigma} \mathbf{x}[t]$$

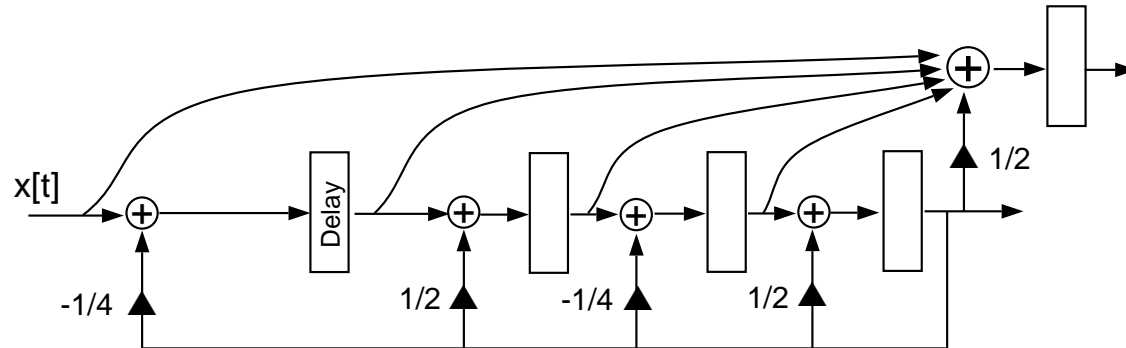
for some matrices $\mathbf{A}_{12}, \mathbf{A}_{22}$

Specifically: Invertible \mathcal{T} such that $\mathcal{A}_\sigma = \mathcal{T}^{-1}\mathcal{A}\mathcal{T}$, $\mathcal{B}_\sigma = \mathcal{T}^{-1}\mathcal{B}$

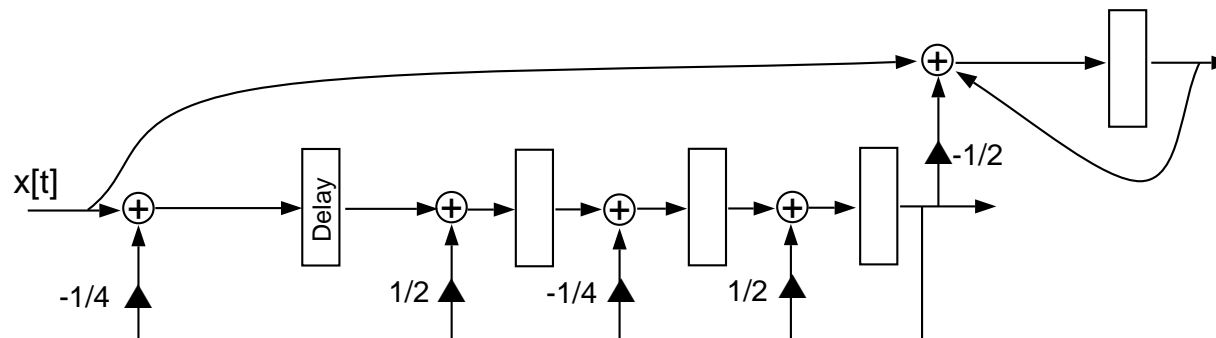
Related work: Šiljak's *inclusion principle*

DIFFERENT REDUNDANT IMPLEMENTATIONS FOR CHECKSUM SCHEME

Traditionally (Chatterjee and d'Abreu, Abraham et al., Reed):



Using previous theorem:



CONCURRENT CHECKING (FAULT DETECTION AND IDENTIFICATION)

Fault model: Single fault corrupts i th state variable

$$\mathbf{q}_f[t] = \underbrace{\mathbf{q}_h[t]}_{\text{fault-free}} + v \mathbf{e}_i$$

Justification: Constrained interconnections of adders, multipliers and delays
 \Rightarrow A single fault corrupts a single state variable

(Class of signal flow graphs, Hadjicostis & Verghese 1999)

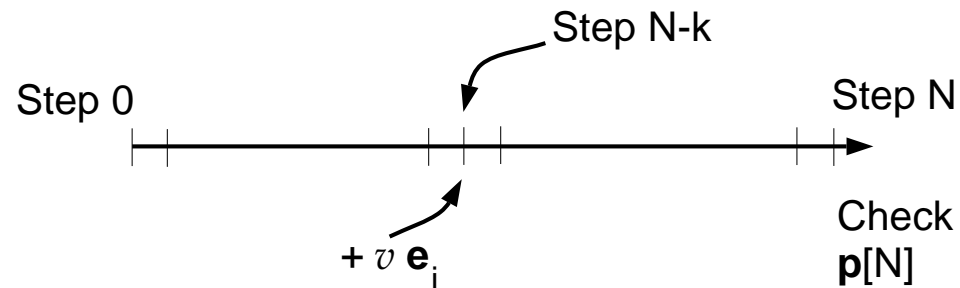
Concurrent error detection (Abraham, Chatterjee, Hadjicostis, ...):

At end of *each* time step, perform the *parity check*

$$\mathbf{p}[t] \equiv \mathbf{P} \mathbf{q}_f[t] = \mathbf{P} v \mathbf{e}_i = v \mathbf{P}(:, i) \stackrel{?}{=} \mathbf{0}$$

Error detection/correction capabilities: Constraints on matrix \mathbf{P} (coding theory)

NON-CONCURRENT CHECKING (1)



Goal: Design redundant implementation so that knowledge of $p[N]$ allows detection and identification of error(s) in interval $[0, N]$

Motivation: Relax reliability requirements on checker (e.g., periodic checking)

Need: For each fault (j), identify

- Value (v_j)
 - State variable (e_{i_j})
 - Step ($N - k_j$)
- } Error correction involves resetting past states/outputs

NON-CONCURRENT CHECKING (2)

Error model: At step $N - k_j$, fault j causes

$$\mathbf{q}_f[N - k_j] = \mathbf{q}_h[N - k_j] + v_j \mathbf{e}_{i_j}$$

Error propagation: At step N ,

$$\mathbf{q}_f[N] = \mathbf{q}_h[N] + \mathcal{A}^{k_j} v_j \mathbf{e}_{i_j}$$

Parity check: At step N ,

$$\mathbf{p}[N] \equiv \mathbf{P} \mathbf{q}_f[N] = v_j \mathbf{P} \mathcal{A}^{k_j} \mathbf{e}_{i_j}$$

Multiple (D) errors result in:

$$\mathbf{p}[N] = \sum_{j=1}^D v_j \mathbf{P} \mathcal{A}^{k_j} \mathbf{e}_{i_j}$$

Task: Construct redundant implementation for detection/identification of D errors

SYNDROME GENERATION

Observation: Syndrome $\mathbf{p}[N]$ is a linear combination of columns of

$$\mathbf{S} = \left[\mathbf{P} \quad \mathbf{PA} \quad \mathbf{PA}^2 \quad \dots \quad \mathbf{PA}^{N-1} \right]$$

Corollary 1: Detection of D errors *if and only if*
all sets of D columns of \mathbf{S} are linearly independent
 \Rightarrow Need at least D additional variables ($s \geq D$)

Corollary 2: Identification of D errors *if and only if*
all sets of $2D$ columns of \mathbf{S} are linearly independent
 \Rightarrow Need at least $2D$ additional variables ($s \geq 2D$)

Lemma (Hadjicostis 2001):

The syndrome matrix \mathbf{S} can be expressed as

$$\mathbf{S} = \left[\mathbf{P} \quad \mathbf{A}_{22}\mathbf{P} \quad \mathbf{A}_{22}^2\mathbf{P} \quad \dots \quad \mathbf{A}_{22}^{N-1}\mathbf{P} \right]$$

MAIN OBSERVATION EXPLOITED IN NON-CONCURRENT SCHEMES

Vandermonde matrix: $\mathbf{V}(x_1, x_2, \dots, x_r) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_r \\ x_1^2 & x_2^2 & \dots & x_r^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{2D-1} & x_2^{2D-1} & \dots & x_r^{2D-1} \end{bmatrix}$

Fact: Any $2D$ columns of \mathbf{V} are linearly independent if parameters $\{x_i\}$ are distinct

Diagonal matrix $\mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & x & 0 & \dots & 0 \\ 0 & 0 & x^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & x^{2D-1} \end{bmatrix}$

Then: $\mathbf{\Lambda}^k \mathbf{V}(x_1, x_2, \dots, x_r) = \mathbf{V}(x_1 x^k, x_2 x^k, \dots, x_r x^k)$

Fact: Any $2D$ columns of $[\mathbf{V} \ \mathbf{\Lambda V} \ \dots \ \mathbf{\Lambda}^k \mathbf{V}] = \mathbf{V}(x_1, \dots, x_r, \dots, x_1 x^k, \dots, x_r x^k)$ are linearly independent if all parameters involved are distinct

REDUNDANT IMPLEMENTATION FOR NON-CONCURRENT IDENTIFICATION OF D ERRORS

Redundant implementation

- Uses $s = 2D$ additional state variables
- Detects $2D$ errors; identifies D errors during time interval $[0, N]$
- \Rightarrow Uses minimal possible number of additional state variables

Procedure

1. **Find** “appropriate” parameters $x, x_1, x_2, \dots, x_\eta$ (recall $\eta = d + s$)
2. **Set** $\mathbf{C} = -\mathbf{M}^{-1}\mathbf{V}(x_1, x_2, \dots, x_d), \quad \mathbf{M} = \mathbf{V}(x_{d+1}, x_{d+2}, \dots, x_\eta)$
3. **Set** $\mathbf{A}_{22} = \mathbf{M}^{-1}\mathbf{\Lambda}\mathbf{M}, \quad \mathbf{\Lambda} = \text{diag}(1, x, x^2, x^3, \dots, x^{2D-1})$
4. **Perform similarity transformation** with $\mathcal{T} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{C} & \mathbf{I}_{2D} \end{bmatrix}$

THEOREM AND PROOF

Theorem (Hadjicostis 2001):

Resulting redundant implementation allows non-concurrent

- (i) identification of D errors, or
- (ii) detection of $2D$ errors

Why? Syndrome matrix \mathbf{S} can be written as

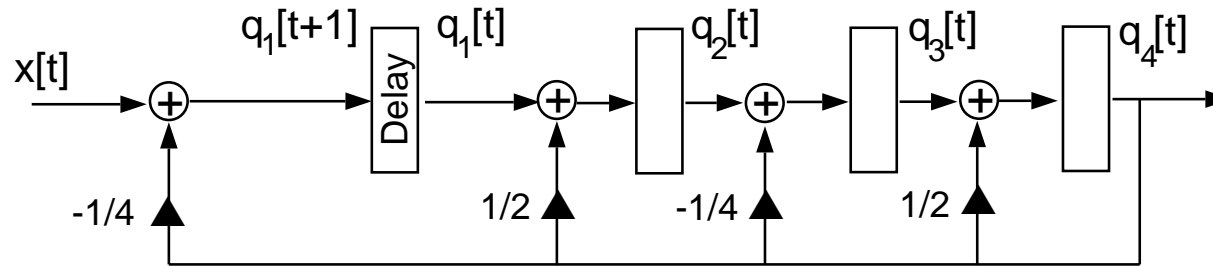
$$\mathbf{S} = \mathbf{M}^{-1} \mathbf{V}(\underbrace{x_1, \dots, x_\eta, x_1x, \dots, x_\eta x, x_1x^2, \dots, x_\eta x^2, \dots, x_1x^{N-1}, \dots, x_\eta x^{N-1}}_{\mathbf{Q}})$$

where \mathbf{Q} is a *large* ($2D \times \eta N$) Vandermonde matrix

Requirement: Choose $x, x_1, x_2, \dots, x_\eta$ so that $\{x_i x^k\}$ are *distinct*

\Rightarrow any $2D$ columns of \mathbf{Q} are linearly independent

EXAMPLE (1)



State evolution:

$$\begin{aligned}
 \mathbf{q}_s[t + 1] &= \mathbf{A}\mathbf{q}_s[t] + \mathbf{b}x[t] \\
 &= \begin{bmatrix} 0 & 0 & 0 & -1/4 \\ 1 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & -1/4 \\ 0 & 0 & 1 & 1/2 \end{bmatrix} \mathbf{q}_s[t] + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} x[t]
 \end{aligned}$$

Goal: $N = 15$, Detect and identify two errors \Rightarrow Use 4 additional variables

Step 1: Choose parameters so that $x_i x^k$ are distinct

$$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x\} = \{-4, -3, 3, 4, -2, -1, 1, 2, \frac{4}{5}\}$$

EXAMPLE (2)

Step 2: Set

$$\mathbf{M} = \mathbf{V}(-2, -1, 1, 2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 1 & 2 \\ 4 & 1 & 1 & 4 \\ -8 & -1 & 1 & 8 \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & x^3 \end{bmatrix}, \quad x = \frac{4}{5}$$

Step 3: Set

$$\mathbf{C} = -\mathbf{M}^{-1}\mathbf{V}(-4, -3, 3, 4) = \begin{bmatrix} -7.5 & -3.333 & 0.667 & 2.5 \\ 10 & 3.333 & -1.667 & -6 \\ -6 & -1.667 & 3.333 & 10 \\ 2.5 & 0.667 & -3.333 & -7.5 \end{bmatrix}$$

$$\mathbf{A}_{22} = \mathbf{M}^{-1}\mathbf{\Lambda}\mathbf{M} = \begin{bmatrix} 0.468 & -0.084 & -0.036 & 0.052 \\ 0.624 & 1.008 & 0.112 & -0.144 \\ -0.144 & 0.112 & 1.008 & 0.624 \\ 0.052 & -0.036 & -0.084 & 0.468 \end{bmatrix}$$

EXAMPLE (3)

Redundant implementation after transformation:

$$\mathbf{q}_h[t+1] = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & -1/4 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1/2 & 0 & 0 & 0 & 0 \\ \hline 0.671 & 2.412 & 2.341 & 0.368 & 0.468 & -0.084 & -0.036 & 0.052 \\ -1.035 & -2.664 & -5.589 & -1.129 & 0.624 & 1.008 & 0.112 & -0.144 \\ 0.621 & 3.744 & 9.003 & 0.465 & -0.144 & 0.112 & 1.008 & 0.624 \\ -0.257 & -3.492 & -5.755 & 0.796 & 0.052 & -0.036 & -0.084 & 0.468 \end{array} \right] \mathbf{q}_h[t] + \frac{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -7.5 \\ 10 \\ 6 \\ 2.5 \end{bmatrix}}{x[t]}$$

Syndrome matrix:

$$\mathbf{S} = \mathbf{M}^{-1} \underbrace{\left[\mathbf{S}_0 \ \mathbf{S}_1 \ \mathbf{S}_2 \ \cdots \ \mathbf{S}_{15} \right]}_{\mathbf{Q}}$$

where $\mathbf{S}_k = \mathbf{V}(-4x^k, -3x^k, 3x^k, 4x^k, -2x^k, -x^k, x^k, 2x^k)$, $x = \frac{4}{5}$

SYSTEMATIC DESIGN FOR NON-CONCURRENT CHECKING

- **Jointly choose**

- (i) Encoding constraints (\mathbf{P} or \mathbf{G})

- (ii) Redundant dynamics (\mathbf{A}_{22})

- **Perform one (non-concurrent) parity check:** $\mathbf{p}[N] \equiv \mathbf{P} \mathbf{q}_f[N]$

- (i) Detect $2D$ faults

- (ii) Identify D faults

} On any variable, at any step in $[0, N]$

- **Advantages:**

- Only $2D$ additional state variables (optimal)

- Efficient identification (Peterson-Gorenstein-Ziegler decoding)

- **Did not address:** Finite-precision arithmetic effects (quantization noise)

RESEARCH DIRECTIONS RELATED TO EMBEDDINGS FOR LTI SYSTEMS

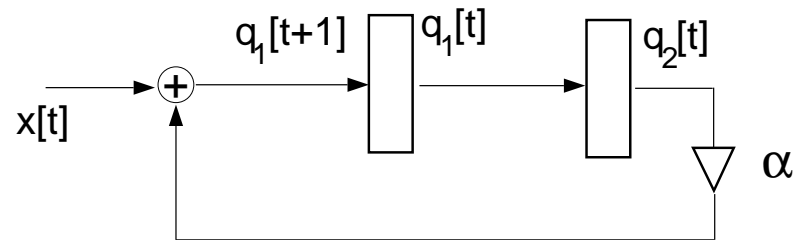
- **Encoded embeddings of LTI dynamic systems**
 - (i) Systematic, resource efficient, reflection of faults into algebraic errors
 - (ii) Generalize/combine modular redundancy and checksum schemes
 - (iii) Non-concurrent error correction and reconfiguration
 - (iv) Connections between linear coding and linear system theory

- **Related future work**
 - Applicable choices of coding constraints and redundant dynamics
 - Flexibility in choosing A_{12}
 - General hardware descriptions (e.g., factored state variables)
 - Finite-precision effects, PGZ decoding algorithm

EXTENSIONS TO LINEAR FINITE-STATE MACHINES

State evolution: $\mathbf{q}_s[t + 1] = \mathbf{A}\mathbf{q}_s[t] \oplus \mathbf{B}\mathbf{x}[t]$

where \mathbf{A} , \mathbf{B} , $\mathbf{q}_s[\cdot]$ and $\mathbf{x}[\cdot]$ have entries in $GF(q)$ ($q = p^m$ with p prime, $m \geq 1$)

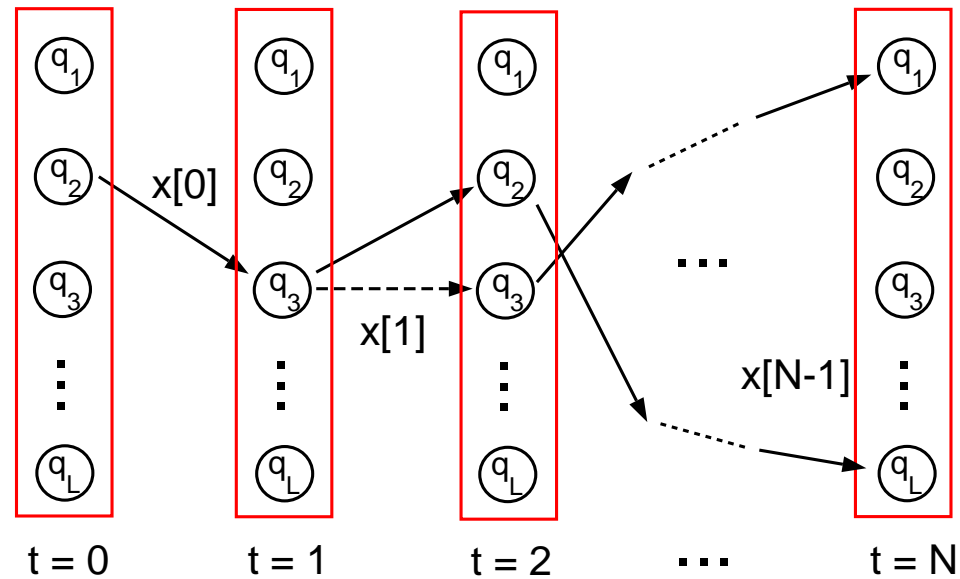


Examples: Sequence enumerators, random number generators, encoders/decoders, linear feedback shift registers, linear cellular automata

Additional bonus:

- (i) Finite-precision not a concern
- (ii) Explicit connection between linear coding and linear system theory (relationship to MDS convolutional codes of York & Rosenthal)
- (iii) Minimization of redundant hardware (Hadjicostis & Verghese 2002)

NON-CONCURRENT PROTECTION OF FSMs

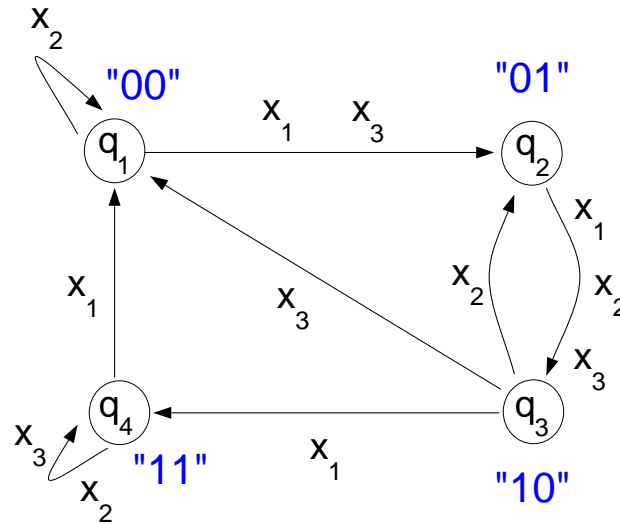


State-transition fault: Fault during $t = 1$ affects state evolution

Non-concurrent checking: Based on the observed state at time step N , how do we

- (i) Detect errors
- (ii) Systematically “diagnose” errors
(determine what went wrong, how and when)

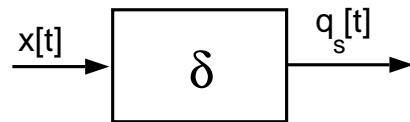
FINITE-STATE MACHINE NOTATION



- **State Set:** $Q_s = \{q_1, q_2, \dots, q_L\}$ **Input Set:** $X = \{x_1, x_2, \dots, x_K\}$
- **States viewed as vectors:** $Q_s = \{\mathbf{q}_s^{(1)}, \mathbf{q}_s^{(2)}, \dots, \mathbf{q}_s^{(L)}\}$
 - E.g., b -dimensional vectors in $GF(2)$ (where $b \geq \lceil \log_2 L \rceil$ so that $2^b \geq L$)
 - E.g., d -dimensional vectors in $GF(q)$ (where $q^d \geq L$)
- **Next-State Function:** $\delta_x(\mathbf{q}_s^{(j)}) = \delta(\mathbf{q}_s^{(j)}, x)$, defined for each $x \in X$

REDUNDANT IMPLEMENTATIONS OF FSMs

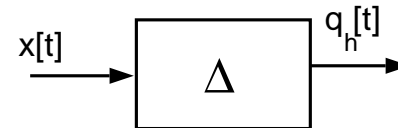
Original FSM



$$\mathbf{q}_s[t + 1] = \delta(\mathbf{q}_s[t], x[t])$$

$\mathbf{q}_s[t]$ is d -dimensional in $GF(q)$

Redundant FSM



$$\mathbf{q}_h[t + 1] = \Delta(\mathbf{q}_h[t], x[t])$$

$\mathbf{q}_h[t]$ is η -dimensional in $GF(q)$

- **Concurrent simulation:** $\mathbf{q}_s[t] = \mathbf{L}\mathbf{q}_h[t], \quad \mathbf{L} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \end{bmatrix}$
 - **Encoding constraints:** $\mathbf{q}_h[t] = \mathbf{G}\mathbf{q}_s[t], \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_d \\ \mathbf{C} \end{bmatrix}$
- } **Linear constraints**
- **Concurrent fault detection:** Verify that $\mathbf{p}[t] \equiv \mathbf{P}\mathbf{q}_h[t] = \mathbf{0}, \quad \mathbf{P} = \begin{bmatrix} -\mathbf{C} & \mathbf{I}_s \end{bmatrix}$

Task: Appropriate Δ_x for $x \in X$ such that $\mathbf{G}\delta_x(\mathbf{q}_s^{(j)}) = \Delta_x(\mathbf{G}\mathbf{q}_s^{(j)})$ for all $\mathbf{q}_s^{(j)} \in \mathcal{Q}_s$

CLASS OF REDUNDANT IMPLEMENTATIONS FOR FSMs

Notation: $\mathbf{q}_h[t] = \begin{bmatrix} \mathbf{q}_{hs}[t] \\ \mathbf{q}_{hr}[t] \end{bmatrix}$ ($\mathbf{q}_{hs}[t]$ is d -dimensional, $\mathbf{q}_{hr}[t]$ is s -dimensional)

Definition of next-state transition mapping: For each input $x \in X$

$$\Delta_x(\mathbf{q}_h[t]) = \begin{bmatrix} \delta_x(\mathbf{q}_{hs}[t]) \ominus \mathbf{A}_{12_x} \mathbf{C} \mathbf{q}_{hs}[t] \oplus \mathbf{A}_{12_x} \mathbf{q}_{hr}[t] \\ \mathbf{C} \delta_x(\mathbf{q}_{hs}[t]) \ominus (\mathbf{C} \mathbf{A}_{12_x} \mathbf{C} \oplus \mathbf{A}_{22_x} \mathbf{C}) \mathbf{q}_{hs}[t] \oplus (\mathbf{C} \mathbf{A}_{12_x} \oplus \mathbf{A}_{22_x}) \mathbf{q}_{hr}[t] \end{bmatrix}$$

Verify: Under fault-free conditions (assuming appropriate initialization)

$$\mathbf{q}_h[t] = \mathbf{G} \mathbf{q}_s[t] = \begin{bmatrix} \mathbf{I}_d \\ \mathbf{C} \end{bmatrix} \mathbf{q}_s[t] \quad \Rightarrow \quad \mathbf{q}_h[t+1] = \mathbf{G} \mathbf{q}_s[t+1]$$

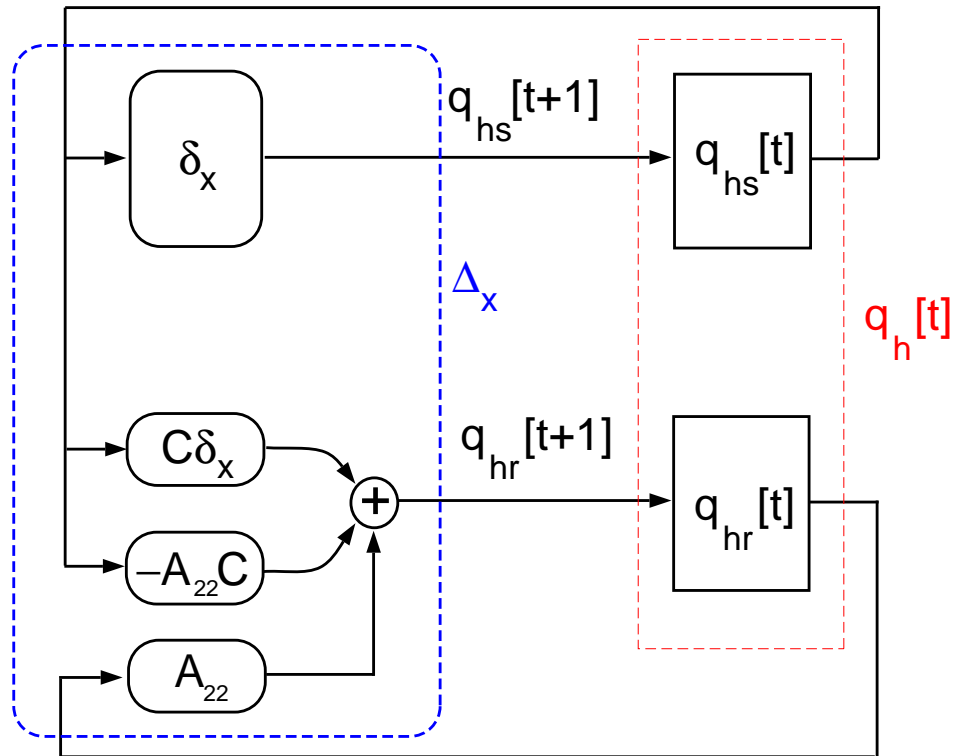
Generalization: Replace mappings \mathbf{A}_{12_x} (\mathbf{A}_{22_x}) by nonlinear mappings δ_{12_x} (δ_{22_x})

BLOCK DIAGRAM DESCRIPTION OF REDUNDANT FSM IMPLEMENTATION

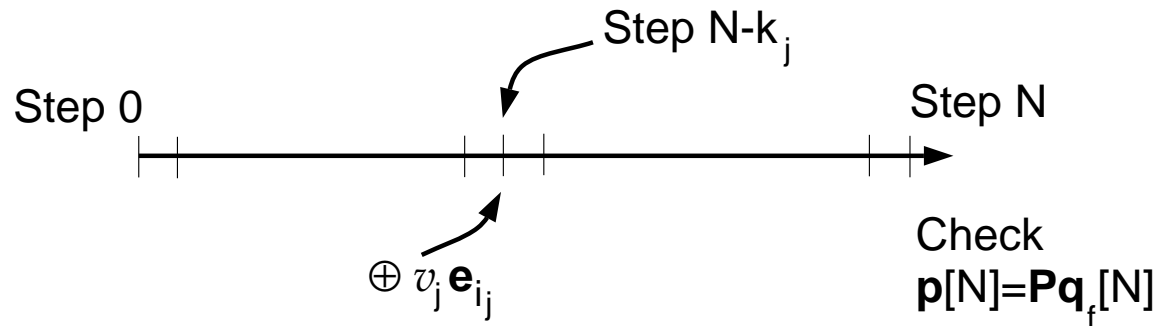
Simplifications: (i) $\mathbf{A}_{12_x} = \mathbf{0}$ for all $x \in X$

(ii) $\mathbf{A}_{22_x} = \mathbf{A}_{22}$ for all $x \in X$

Mapping Δ_x (under input $x \in X$):



NON-CONCURRENT CHECKING IN FSMs (1)



Goal: Design redundant FSM implementation so that knowledge of $p[N]$ allows us to detect and identify of error(s) in interval $[0, N]$

Need: For each fault (j), identify

- Value (v_j)
 - State variable (e_{i_j})
 - Step ($N - k_j$)
- } Error correction involves resetting past states/outputs

NON-CONCURRENT CHECKING IN FSMs (2)

Error model: Fault j corrupts the i_j th state variable by v_j during step $N - k_j$

$$\mathbf{q}_f[N - k_j] = \underbrace{\mathbf{q}_h[N - k_j]}_{\text{fault-free state}} \oplus v_j \mathbf{e}_{i_j}$$

Error propagation: At step N , $\mathbf{q}_f[N] = \mathbf{q}_h[N] \oplus \mathbf{e}$

Nonlinear machine: Error \mathbf{e} hard to characterize

Theorem: Syndrome $\mathbf{p}[N] = v_j \mathbf{A}_{22}^{k_j} \mathbf{P} \mathbf{e}_{i_j}$

Generalizes to: Syndrome $\mathbf{p}[N] = \sum_{j=1}^D v_j \mathbf{A}_{22}^{k_j} \mathbf{P} \mathbf{e}_{i_j}$

Again: Syndrome $\mathbf{p}[N]$ is a linear combination of columns of

$$\mathbf{S} = \left[\mathbf{P} \quad \mathbf{A}_{22}\mathbf{P} \quad \mathbf{A}_{22}^2\mathbf{P} \quad \cdots \quad \mathbf{A}_{22}^{N-1}\mathbf{P} \right]$$

FSM CONSTRUCTION FOR NON-CONCURRENT IDENTIFICATION

Redundant implementation

- Uses $s = 2D$ additional state variables
- Detects $2D$ errors; identifies D errors during time interval $[0, N]$

Procedure

1. **Find** “appropriate” parameters $x, x_1, x_2, \dots, x_\eta$ (recall $\eta = d + s$)
2. **Set** $\mathbf{C} = -\mathbf{M}^{-1}\mathbf{V}(x_1, x_2, \dots, x_d), \quad \mathbf{M} = \mathbf{V}(x_{d+1}, x_{d+2}, \dots, x_\eta)$
3. **Set** $\mathbf{A}_{22} = \mathbf{M}^{-1}\mathbf{\Lambda}\mathbf{M}, \quad \mathbf{\Lambda} = \text{diag}(1, x, x^2, x^3, \dots, x^{2D-1})$
4. **Set** Δ_x for each $x \in X$

FINITE FIELD CONSIDERATIONS

Design requirement: $x_i x^k$ are distinct, $1 \leq i \leq \eta$, $0 \leq k \leq N - 1$

- **Necessary condition:** Finite field needs at least ηN *nonzero* entries

$$\Rightarrow q - 1 \geq \eta N$$

- **One possibility:** Let g be a primitive element of $GF(q)$
(i.e., $\{1, g, g^2, g^3, \dots\}$ generates all nonzero elements in $GF(q)$)

$$\text{Set: } x = g^\eta, \quad x_i = g^i, \quad 1 \leq i \leq \eta$$

- Construction related to MDS convolutional codes (Rosenthal & York)

RESEARCH DIRECTIONS RELATED TO EMBEDDINGS OF FSMs

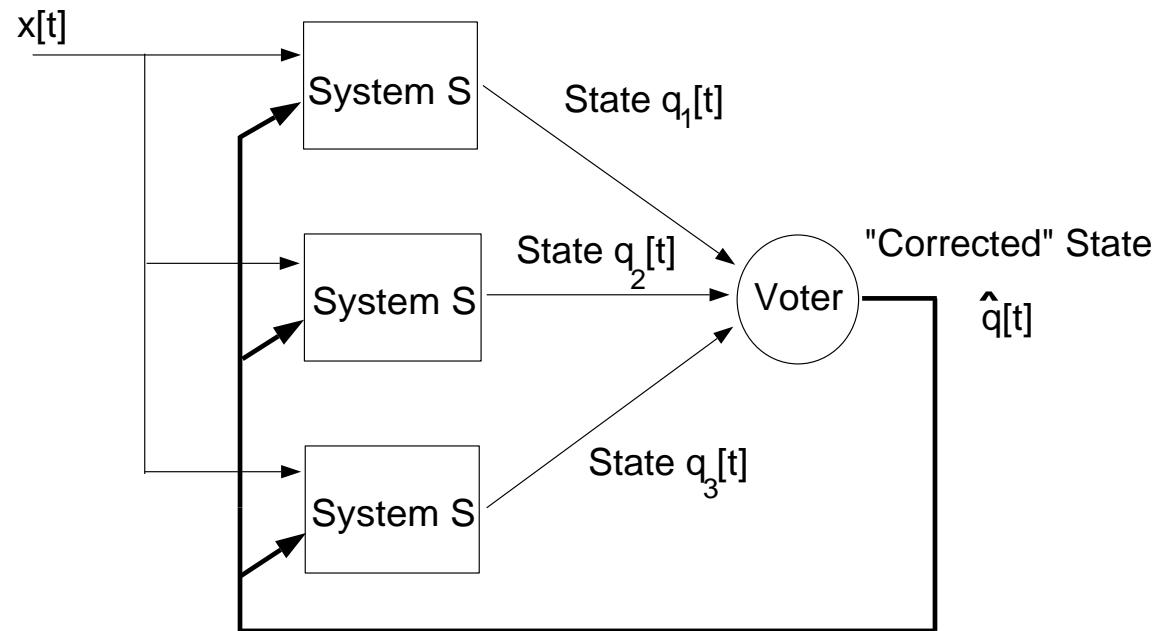
- **Encoded embeddings of FSMs**

- Systematic, resource-efficient fault tolerance for FSMs
- Reflection of hardware faults through error models
- Generalization of modular redundancy and checksum schemes
- Non-concurrent, periodic checking

- **Related future work**

- Other pairs of coding constraints and redundant dynamics
- Rollback-based correction
- Flexibility in choosing A_{12}
- Flexibility in choosing input-varying A_{12_x} and/or A_{22_x}

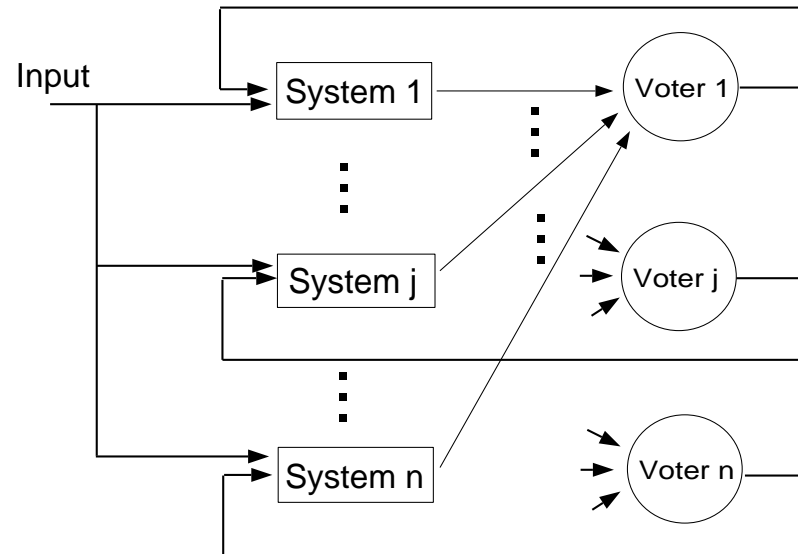
DEALING WITH VOTER FAULTS



Problem: If voter fails with prob. p_v , then after L steps

$$\Pr[\text{correct state trajectory}] \leq (1 - p_v)^L$$

DISTRIBUTED VOTING SCHEMES (HADJICOSTIS AND VERGHESE)



At each step:

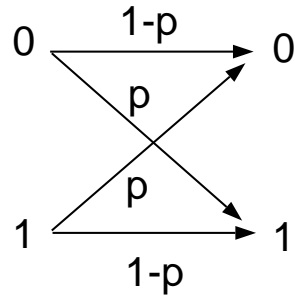
- Systems fail with prob. p_s
- Voters fail with prob. p_v

$$\Pr[\text{overall failure at or before time } L] \leq L \sum_{i=n/2}^n \binom{n}{i} p^i (1-p)^{n-i}, \quad p \equiv p_v + (1-p_v)p_s$$

Result: Probability decreases exponentially with n if $p < 1/2$

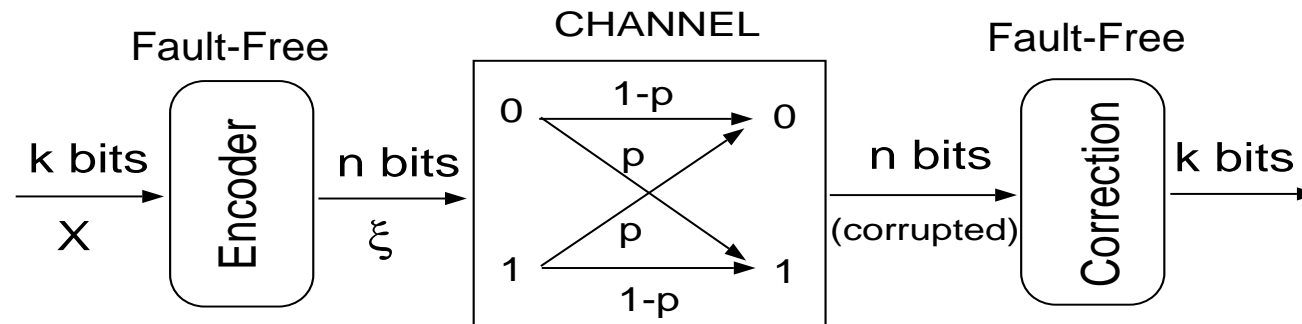
Related: Compressor graphs (Margulis, Pippenger), stable memories (Taylor, Kuznetsov)

UNRELIABLE DIGITAL COMMUNICATION LINK



Send same bit n times:

$$\Pr[\text{error}] = \sum_{i=n/2}^n \binom{n}{i} p^i (1-p)^{n-i}$$



Shannon: Can send k bits with arbitrarily small error by sending n bits as long as $\frac{k}{n} < C$ (where C is the *channel capacity* and depends only on p)

EMBEDDING k SYSTEMS INTO n REDUNDANT SYSTEMS

Consider k instantiations of an LFSM:

$$\begin{aligned}
 \mathbf{q}_1[t + 1] &= \mathbf{A}\mathbf{q}_1[t] \oplus \mathbf{b}x_1[t] \\
 \mathbf{q}_2[t + 1] &= \mathbf{A}\mathbf{q}_2[t] \oplus \mathbf{b}x_2[t] \\
 &\vdots \\
 \mathbf{q}_k[t + 1] &= \mathbf{A}\mathbf{q}_k[t] \oplus \mathbf{b}x_k[t]
 \end{aligned}$$

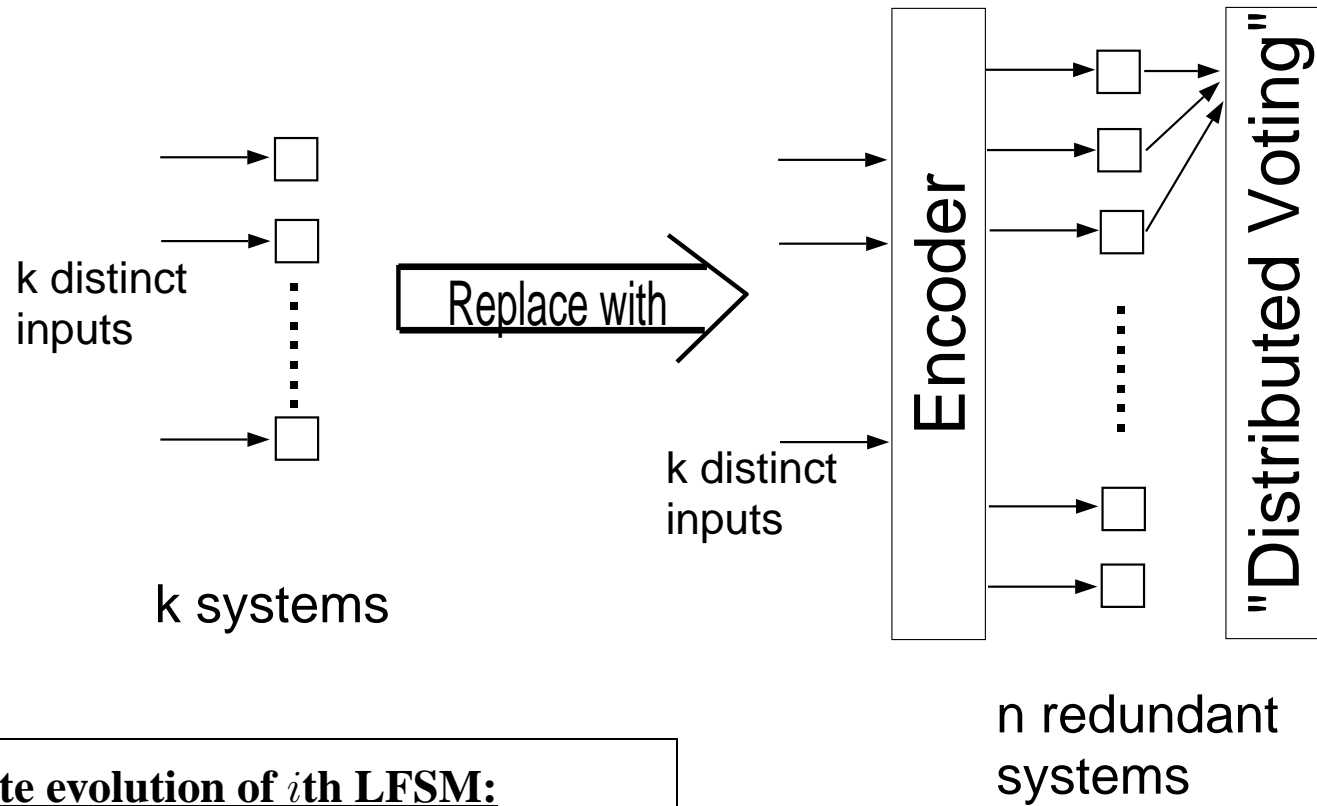
Embed k instantiations into n systems using (n, k) linear code:

$$\underbrace{\begin{bmatrix} \xi_1[\tau] & \xi_2[\tau] & \cdots & \xi_n[\tau] \end{bmatrix}}_{\mathbf{q}_h^T[\tau]} \equiv \underbrace{\begin{bmatrix} \mathbf{q}_1[\tau] & \mathbf{q}_2[\tau] & \cdots & \mathbf{q}_k[\tau] \end{bmatrix}}_{\mathbf{q}_s^T[\tau]} \mathbf{G}^T$$

if

$$\begin{bmatrix} \xi_1[t + 1] & \xi_2[t + 1] & \cdots & \xi_n[t + 1] \end{bmatrix} = \mathbf{A} \begin{bmatrix} \xi_1[t] & \xi_2[t] & \cdots & \xi_n[t] \end{bmatrix} \oplus \mathbf{b} \underbrace{\left(\begin{bmatrix} x_1[t] & x_2[t] & \cdots & x_k[t] \end{bmatrix} \mathbf{G}^T \right)}_{e \left(\begin{bmatrix} x_1[t] & x_2[t] & \cdots & x_k[t] \end{bmatrix} \right)}$$

EMBEDDING k DISTINCT LFSMs INTO n REDUNDANT LFSMs



State evolution of i th LFSM:

$$\mathbf{q}_i[t + 1] = \mathbf{A}\mathbf{q}_i[t] \oplus \mathbf{b}x_i[t]$$

RELIABLE LFSMS USING CONSTANT REDUNDANCY

Theorem (Hadjicostis and Verghese):

Using **constant redundancy** per system, we can embed k distinct LFSMs into n redundant LFSMs such that

$$\Pr[\text{overall fault at or before time } L] < LCk^{-\beta}$$

Construction:

- LFSMs use *unreliable* XOR gates (2-input)
- Encoding uses low density parity check (LDPC) codes (Gallager 1963)
- Decoding done using *unreliable* XOR gates and voters (techniques studied by Gallager (1963) and Taylor (1968))

Related work: Spielman, Gács

RESEARCH DIRECTIONS RELATED TO EMBEDDINGS OF MULTIPLE LFSMS

- Practical implementations
 - Fast/inexpensive encoders and decoders
 - Digital signal processing (linear filters)
- Generalizations (arbitrary finite-state machines)
- Permanent faults (reconfiguration)
- Theoretical work:
 - Bounds, “computational capacity”
 - Low-complexity error correction (iterative schemes, sequential decoding, “turbo” codes etc.)

OTHER RESEARCH INTERESTS

- System control, monitoring, testing, verification
 - Distributed embedded systems
 - Real-time systems
 - Software reliability
- ⇒ **Applicability of error detection/correction techniques**
- ⇒ **Implication of hardware redundancy, architectural constraints**

Other:

- Soft-decision decoding
- Probabilistic encoding, probabilistic routing
- Network reconfiguration, performance guarantees
- System reconfiguration, resource allocation