Testability & Reliability

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6161

Modern Manufacturing System Engineering

Testability and Reliability

- What is Testability?
  - Importance of Testing
  - Onboard and off-board diagnosis
  - Multiple Fault Diagnosis Methods
  - Sequential Fault Diagnosis
- What is Reliability?
  - Importance of Reliability
  - Reliability Definitions
  - Device Reliability
  - System Reliability Modeling

What is Testability?

- Management Definition
  - Testability is the ability to generate, evaluate, and apply tests to improve quality, reduce life-cycle costs, and minimize time-to-profit
- Engineering Definition
  - Testability is the extent to which a design (or fielded system) can be tested for the detection and isolation of (manufacturing) defects or (field) failures

- A Testable System Implies
  - better fault coverage and fault isolation
  - shorter testing times
  - higher quality product

shorter time-to-market

⬇

lower life-cycle costs

Failures and Defects

- Failure
  - renders a system unable to perform its normal function according to specification
  - caused by external environment or by some internal defect (e.g., design, manufacturing)

- Defect or Fault
  - an imperfection in either the design or the structure of a product
  - a defect may or may not lead to failure (but is a non-conformance to specifications)

Examples of Failures

- Incorrect and Marginal Designs
  - identify design problems early to improve and verify designs
  - typical problems: incorrect schematics, timing issues, changing specs,...
  - one solution: specification-based testing
- Production Defects
  - flaws during manufacturing and assembly processes
  - both permanent and transient defects
- Operational and Maintenance failures
  - packaging and transportation (shock and vibration resulting from dropped boxes)
  - product abuse (dropping a product, operate in overheated conditions, improper storage, environment (temperature, radiation,…)
  - wear and aging
  - static electricity
  - power surges

Importance of Testing

- Quality Assurance
- Evaluating a Manufacturing Process
- Identifying Faulty Components for Repair
- Cost of fixing problems in the field increases exponentially!

LEVEL OF ASSEMBLY	COST PER FAILURE ($)
COMPONENT LEVEL	1
CIRCUIT BOARD LEVEL	10
BOX LEVEL	100
SYSTEM LEVEL	1000
FIELD OPERATION LEVEL	2000-20,000

Latest Example: Boeing 787 grounded for Li-ion Battery Problems

Classification of Tests

- Based on Purpose
  - detection tests
  - diagnostic tests
- When Performed
  - design verification - simulation
  - manufacturing tests - behavioral, parametric
  - field tests - maintenance, diagnosis (on-board, off-board (remote/automatic/manual)
- Level
  - system
  - subsystem
  - chip
  - circuit
- Test Application
  - external testing
  - self-test

Fault Detection, Diagnosis and Prognosis

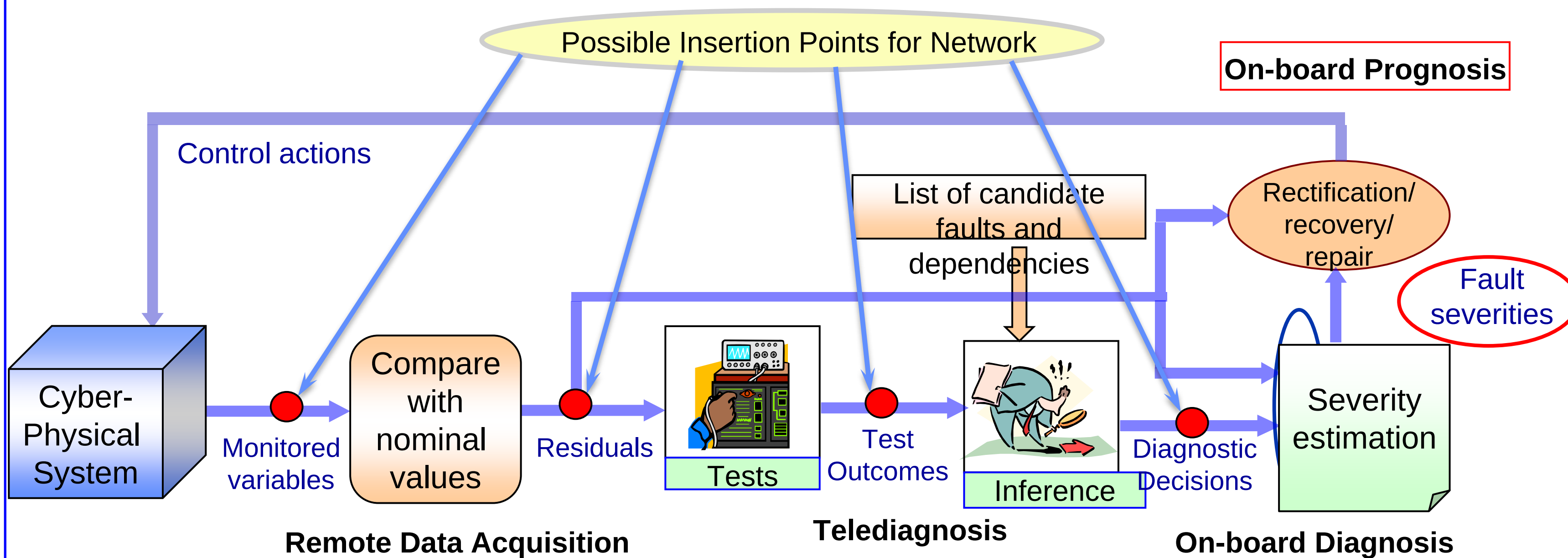
- Fault Detection
  - The process of recognizing deviations of a system from its "normal" behavior using available measured data
- Fault Isolation

Diagnosis = Isolation + Identification

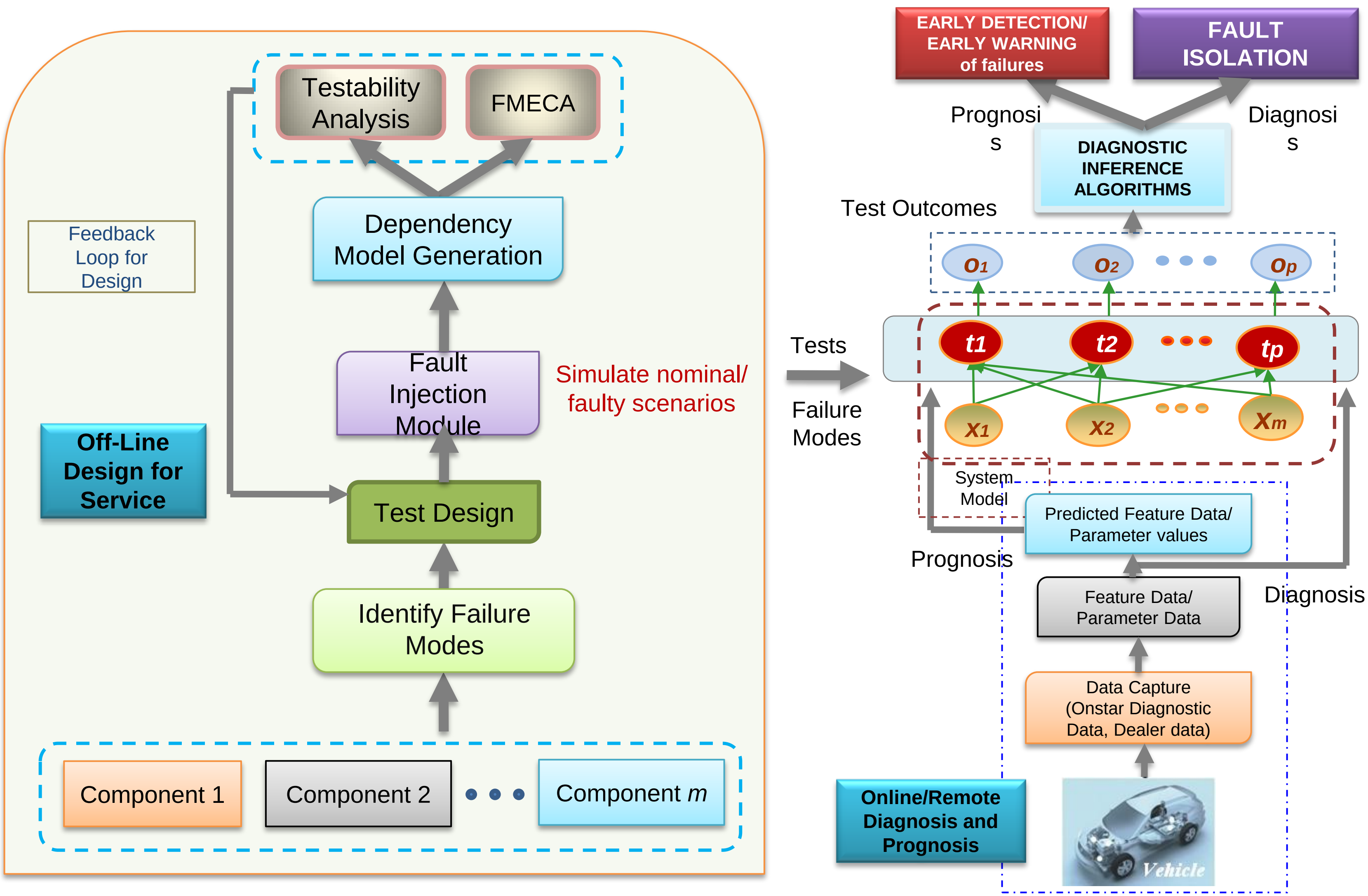
  - The process of localizing faults to physical regions (components) of the system
- Fault Identification
  - Involves the severity of fault estimation, or the identification of *fault models*
- Fault Prognosis

Prognosis = Early Diagnosis + RUL Estimation

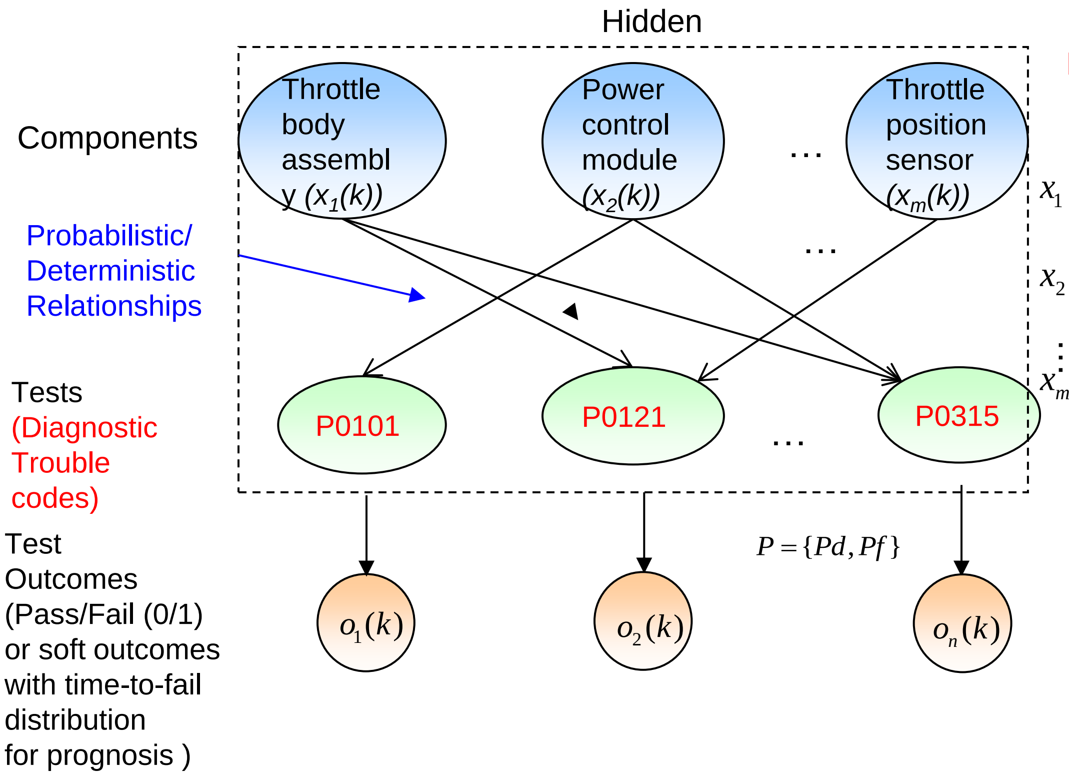
  - The process of estimating the fault evolution over time
  - Involves estimation of residual useful life (RUL) of components and subsystems



One Possible Realization of the IDPP



Diagnostic Inference: A Tri-partite Graph Model



Fault model: Diagnostic matrix (D-matrix)

	t_1	t_2	t_3	t_4	…	t_n
x_1	1	0	0	0	…	0
x_2	1	0	1	1	…	0
⋮	…	…	…	…		0
x_m	0	0	0	0	…	1

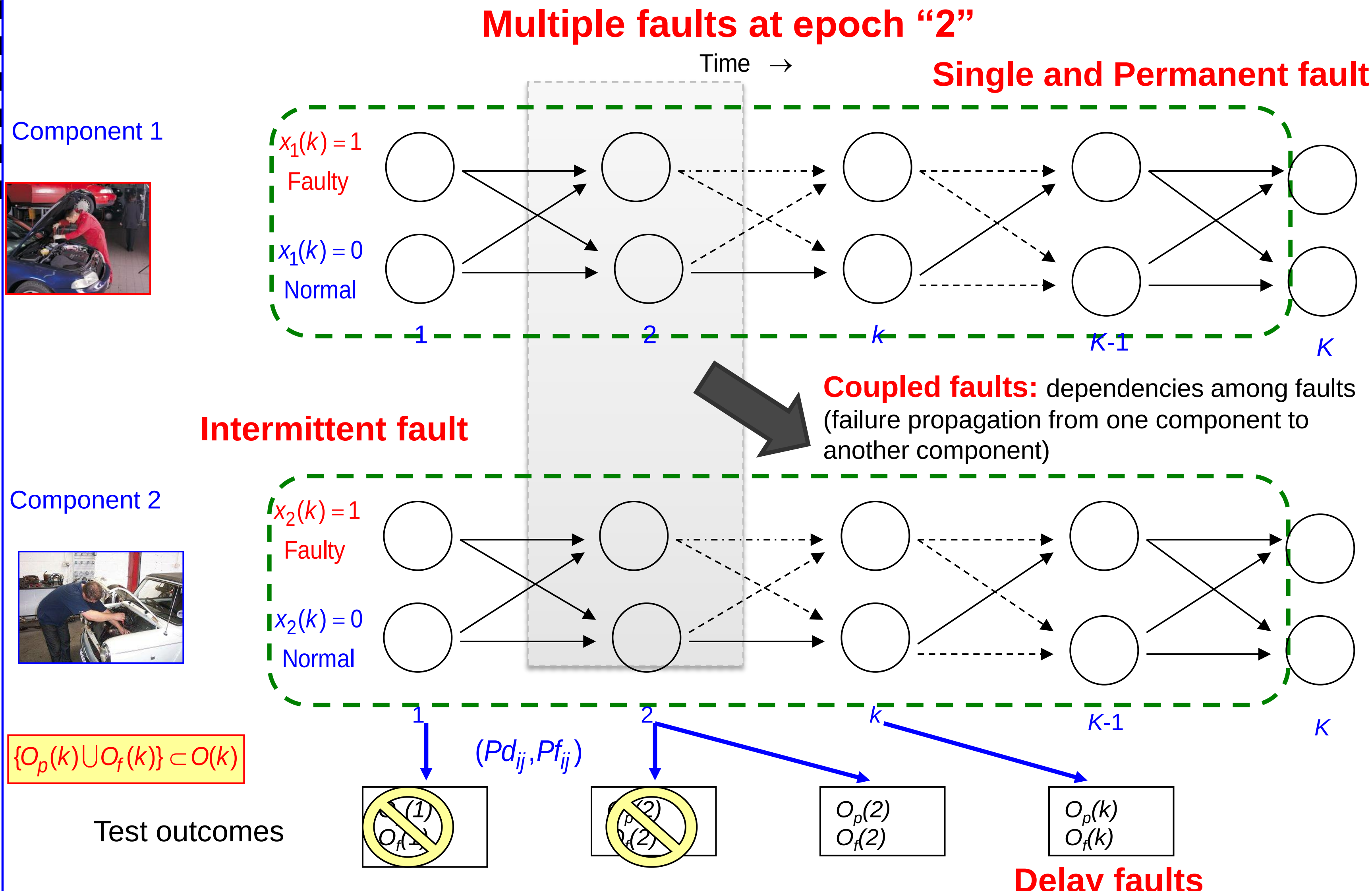
- Obtained via simulation
- Dependency graphs
- Error Correcting Codes
  Learned from Data

- Component states, tests and test outcomes represent the nodes of digraph
  - True states of the component states and tests are hidden
  - $P = \{Pd, Pf\}$ represents the detection and false alarm probability pair
- NP-hard combinatorial optimization problem (even in the static case!)

Fault Diagnosis Problems and Terms

- Fault Assumptions
  - Single fault: If only a single component is faulty
  - Multiple faults: If more than one component is faulty
    - If multiple faults result in similar test signature (same rows in D-matrix) ⇒ Ambiguous faults
    - If union of multiple fault signatures is similar to one or more fault signatures ⇒ Hidden or masking faults (caused by insufficient observability due to inadequate sensors/test design)
    - If multiple faults are dependent on each other ⇒ ***Coupled faults***
  - Fault Dynamics:
    - Component once failed, remains in that state ⇒ Permanent faults
    - Malfunction of the component occurs only at intervals with/without specific patterns ⇒ Intermittent faults
    - If faults take time to propagate or tests are observed with delays ⇒ ***Delay faults***
- Test reliabilities
  - Reliable/perfect tests: *No Missed detections or False alarms*
  - Unreliable/imperfect tests (more practical): *Missed detections and/or False alarms*

Illustration of Different Fault Types



Dynamic Multiple Fault Diagnosis

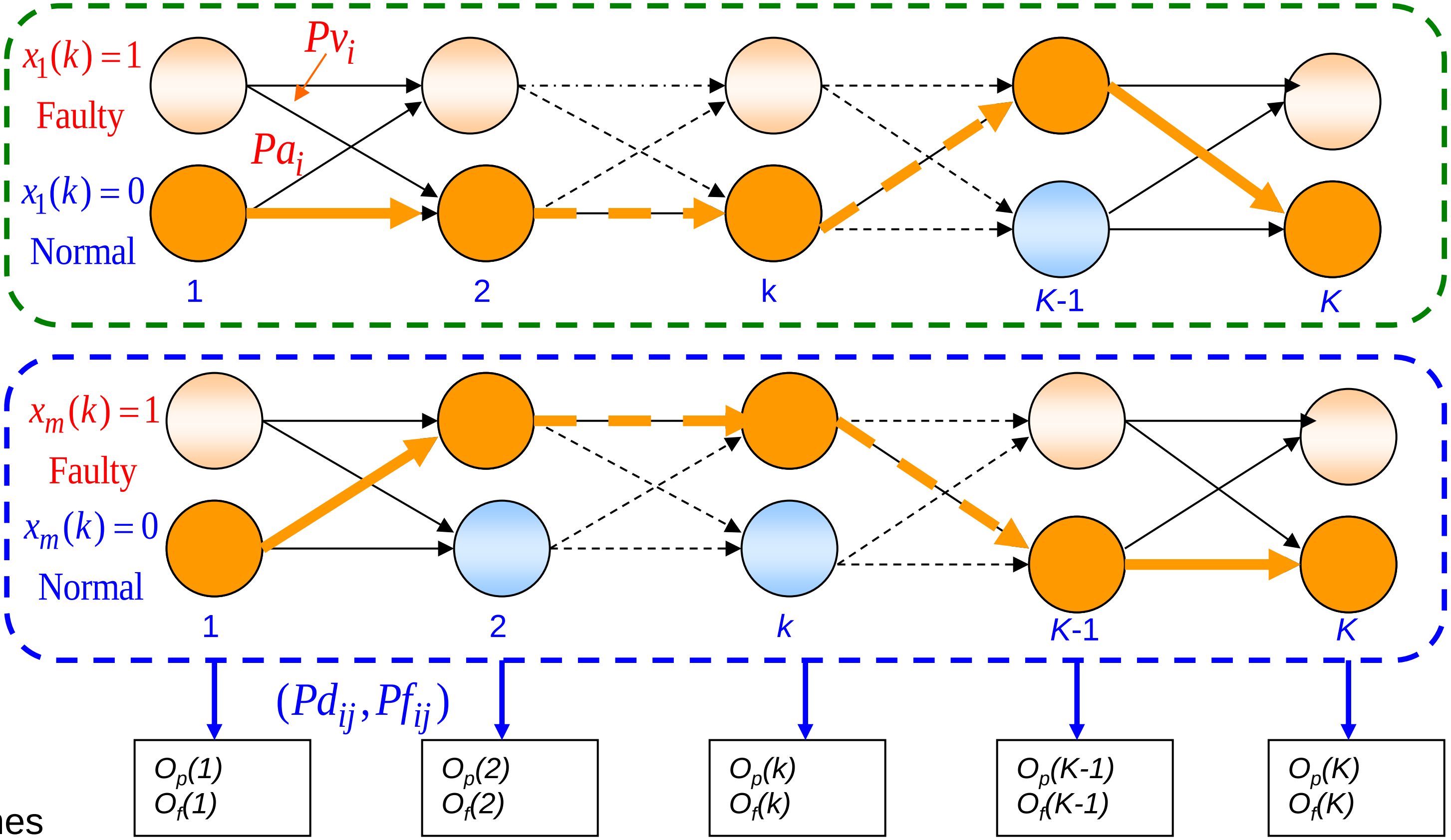
- Determine the most likely evolution of fault states, one that best explains the observed test outcomes over time

Singh *et. al.*, 2009
IEEE T-SMC: Part A

Time →

HMM 1
(Component 1)

HMM *m*
(Component *m*)



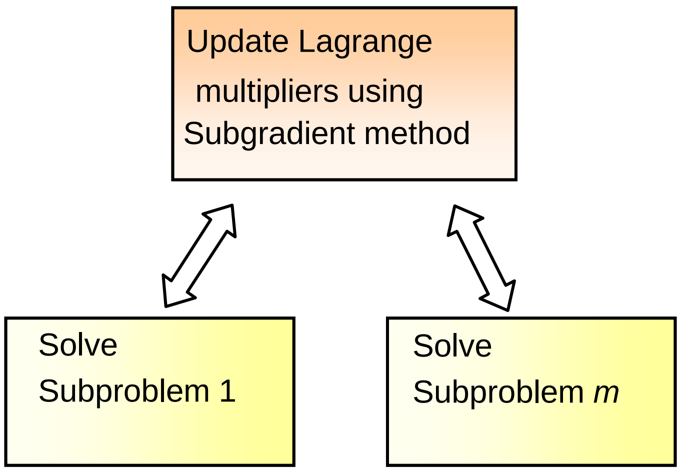
Test outcomes

Problem: Find maximum *a posteriori* (MAP) solution:

$$\hat{X}^K = \arg \max_{X^K=\{\underline{x}(1),\underline{x}(2),..,\underline{x}(K)\}} \Pr(X^K \mid O^K)$$

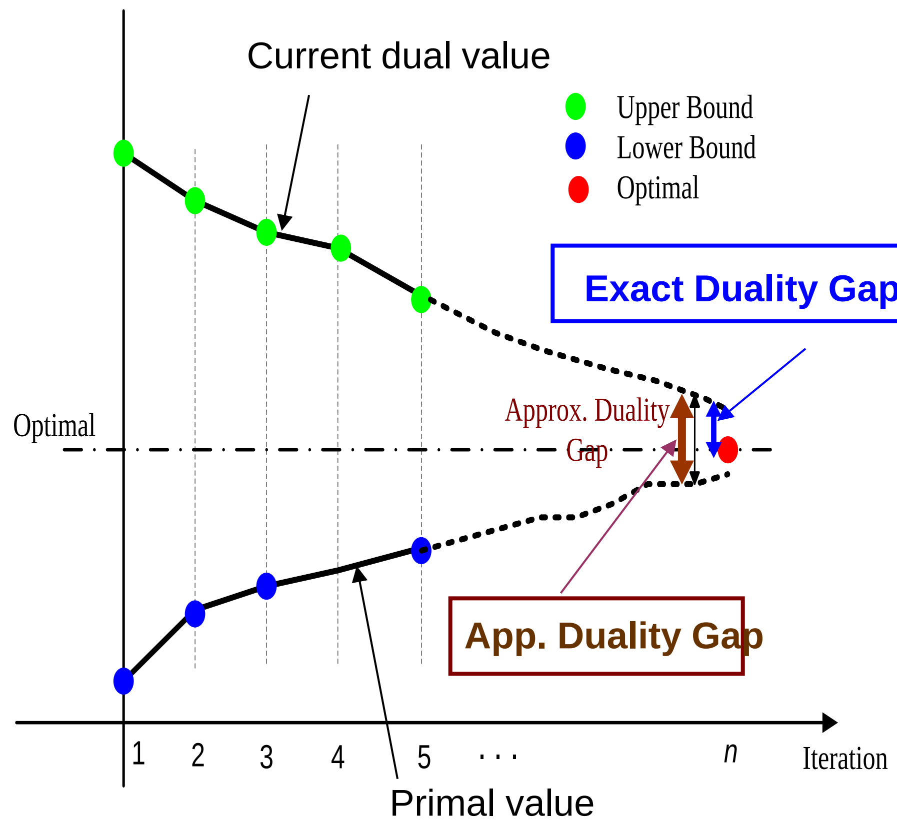
Optimization Framework and Novel Feature

- Two-level coordinated solution framework



- Primal-dual decomposition
- Separable problems at lower level
  - Coordination via multipliers
  - Distributed implementation
- L-ranked solutions via Murty's decomposition

- Measurable performance



- High Diagnostic Accuracy

MFD Problems and Algorithms

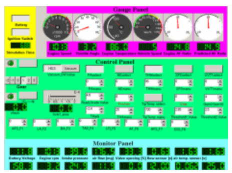
- Single Frame (Static) MFD Algorithms
  - Perfect Test Case: Set covering Algorithms (2003)
  - Imperfect Test Case:
    - Lagrangian Relaxation Algorithm (LRA, 1998; IEEE T-SMCC)
    - Approximate Belief Revision Algorithm (ABR, 2008; IEEE T-SMCA)
    - Deterministic Simulated Annealing (DSA, 2009; IEEE T-SMCA)
    - L-ranked Solutions via Murty's Decomposition (1998; IEEE T-SMCC)
- Multi-frame (Dynamic) MFD Algorithms – *Infer multiple, coupled and intermittent faults with fault propagation and observation delays*
  - Perfect Test Case: Dynamic set covering and Delay Dynamic Set Covering (Kodali, 2013; IEEE T-SMCA)
  - Imperfect Test Case
    - Deterministic Simulated Annealing + Markov-chain based smoothing (2009; SMC-A)
    - LRA + (Soft Decision, Hard Decision) Viterbi Algorithms (2009; IEEE T-SMCA)
    - Gauss-Seidel or Jacobi–based Coordinate Ascent Algorithm (Kodali, 2013; SMC-A)
    - Block Coordinate Ascent and Viterbi (BCV) or Annealed MAP (Zhang, 2013; SMC-A)

DMFD: Real World Applications

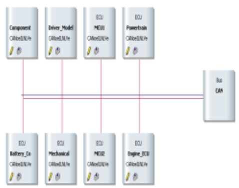
Automotive

- Anti-lock/Regenerative Braking
- CRAMAS® Engine Data
- Li-ion Batteries
- Fuel pumps, ETCS, EPGS

CRAMAS® Platform



Regenerative Braking



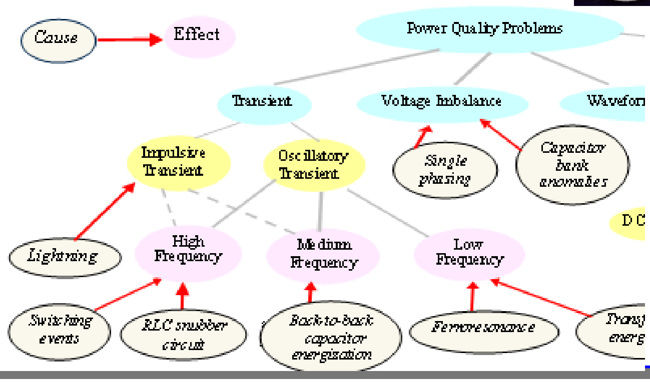
Aerospace

- PW2500
- Black Hawk and Sea Hawk T-700 Engines
- Non-toxic Orbital Maneuvering System and Reaction Control System (NT-OMS/RCS)
- International Space Station
- Ares-1x Rocket

Jet Engine

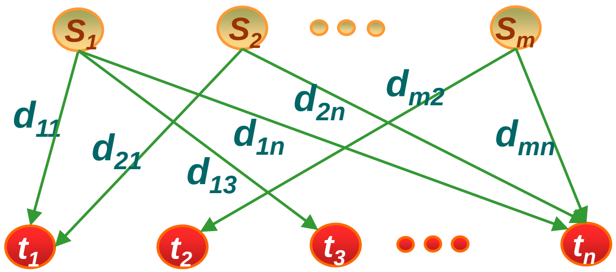
Power/Buildings

- Power Quality Monitoring
- HVAC Chillers



Guided Troubleshooting

- Military Vehicles, Fork lift trucks
- Optical Scanning Machines, Semiconductor Fabrication Facilities
- Medical Equipment



Static Test Sequencing Problem and Algorithms

My first foray into diagnostics)

- **Motivation**
  - Design of optimal test sequencing procedures
  - Application for off-equipment (off-board) diagnosis
- **Simplest Test Sequencing Problem**
  - A set of m failure sources with prior probabilities, $\underline{p} = \{p(s_1), p(s_2), \ldots, p(s_n)\}$
  - A corresponding set of n test costs, $C = \{c_1, c_2, \ldots, c_n\}$
  - An optimal test sequence which attains the minimum expected cost

$$\min_{\{P_i\}^m_{i=0}} J = \sum_{i=0}^{m} \left\{ \sum_{j=1}^{|P_i|} C_{p_i[j]} \right\} p(s_i)$$

  - P_i denotes the sequence of tests applied to isolate the system state s_i
  - Optimization is done over all admissible test sequences
- **Optimal algorithms**
  - Dynamic Programming: High Storage and computational requirement $O(3^n) \Rightarrow n < 13$
  - AND/OR Graph Search and information theory $\Rightarrow$ 50-100 components
- **Suboptimal algorithms**
  - Information heuristic algorithm (can be arbitrarily off from optimal)
  - Rollout strategies with information gain heuristics (near-optimal and practical)
- **Extensions to realistic systems:** setup operations, precedence constraints, multi-outcome tests, unreliable tests, multi-mode test sequencing, blocks of tests, modular diagnosis, rectification,…

Dynamic Test Sequencing: Active probing during DMFD is an open research problem in the context of diagnosis. Done in dynamic sensor management.

D-matrix based Measures

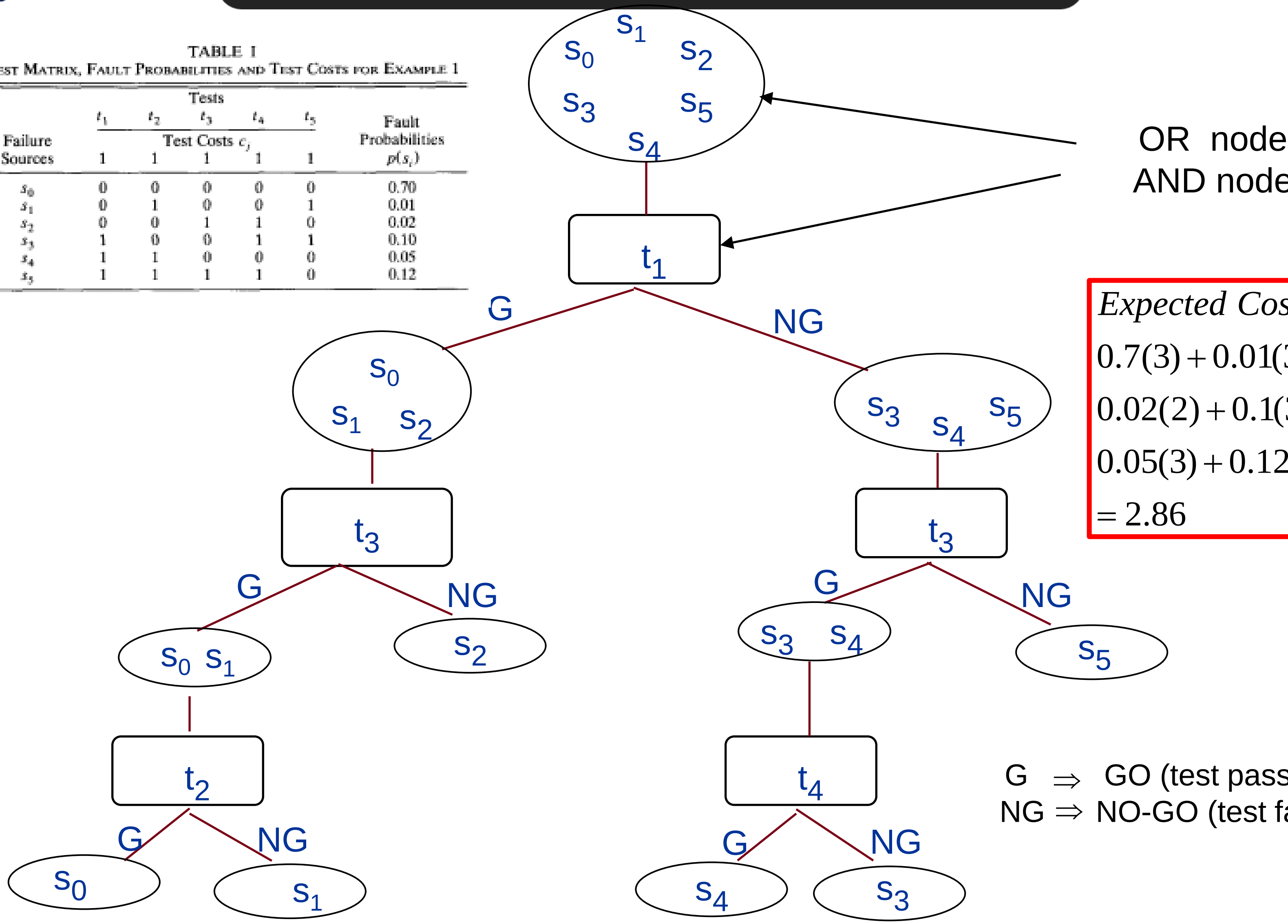
- **D-matrix (Diagnostic Dictionary, Fault Dictionary)**
  - Assume perfect test case for simplicity
  - d_{ij} = 1 if failure source s_i is detectable by test t_j
- **Undetectable faults**
  - set of faults in the system that cannot be detected using the available tests
  - correspond to **null rows** in the D-matrix
- **Redundant Tests**
  - set of tests that have the same detection signature, i.e., detect the same set of faults, is termed redundant
  - correspond to **identical columns** in D-matrix
- **Ambiguity Groups**
  - set of faults that have the same observability signature, i.e., detected by the same set of tests, is termed "ambiguity set"
  - correspond to **identical rows** in D-matrix
- **Hidden Failures**
  - set of failures that are detected only by a subset of tests that detect a given fault
  - correspond to the **set of rows which are subsets of a given row** of D-matrix
- **Masking False Failures**
  - an irreducible set of faults, which when occur simultaneously produce the same symptoms as some other fault, is termed a "masking set"
  - corresponds to **an irreducible set of rows of D-matrix which when logically added (OR-ed) would produce some other row** of D-matrix

Sequential Test Algorithm

TABLE I

Test Matrix, Fault Probabilities and Test Costs for Example 1

Failure Sources	t_1	t_2	t_3	t_4	t_5	Fault Probabilities $p(s_i)$
Test Costs c_j	1	1	1	1	1	
s_0	0	0	0	0	0	0.70
s_1	0	1	0	0	1	0.01
s_2	0	0	1	1	0	0.02
s_3	1	0	0	1	1	0.10
s_4	1	1	0	0	0	0.05
s_5	1	1	1	1	0	0.12



Expected Cost :

$$0.7(3)+0.01(3)+0.02(2)+0.1(3)+0.05(3)+0.12(2)=2.86$$

G ⇒ GO (test passes)

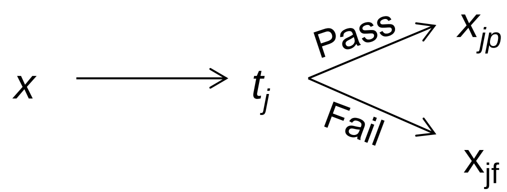
NG ⇒ NO-GO (test fails)

Structure of the Test Algorithm

- Solution exists if and only if **no** two rows of the D-matrix are identical
  - need number of tests, $n \geq \log_2 (m+1)$; m = number of failure modes
  - since the problem is finite, existence of solution implies the existence of an optimal solution
- Solution is a deterministic and sequential algorithm
  - decides which test to perform next depending upon the outcomes of previously applied tests $\Rightarrow$ state-dependent/closed-loop/adaptive
- Algorithm has AND/OR decision tree structure
  - OR nodes labeled by ambiguity status (~ states)
  - AND nodes denote tests at OR nodes (~ decisions)
  - initial OR node = state of complete ambiguity
  - terminal nodes (goal nodes, leaves) = s_i (or) residual ambiguity
  - each test is performed at most once on a path (for perfect tests)
  - weighted length of the tree = expected test cost
  - identifies redundant tests (i.e., identical columns of D-matrix and tests not used in the test algorithm)

Dynamic Programming Approach

- Application of a test t_j at an OR node x partitions x into two disjoint subsets, x_{jp} and x_{jf}



- Optimal cost-to-go at OR node x

$$h^*(x) = \min_j \{c_j + p(x_{jp}) h^*(x_{jp}) + p(x_{jf}) h^*(x_{jf})\}$$

$$p(x_{jp}) = \frac{\sum_{s_i \in x} (1 - d_{ij}) p(s_i)}{p(x)}; \; p(x_{jf}) = 1 - p(x_{jp}); \; p(x) = \sum_{s_i \in x} p(s_i)$$

Bottom up Algorithm

- Alternate version of DP (unconditional version)

$$Let\ v^*(x) = p(x) h^*(x)$$

$$Then,\ v^*(x) = \min_j \{p(x) c_j + v^*(x_{jp}) + v^*(x_{jf})\}$$

- Computational complexity grows exponentially with $n \Rightarrow O(3^n)$

Analogy between Testing and Coding

- Sequence of test results generates a binary prefix-free coding of the failure sources $\{s_0, s_1, \ldots, s_m\}$
  - pass outcome (G) = 0 and fail outcome (NG) = 1
- Noiseless coding problem
  - (m+1) binary messages $S = \{s_0, s_1, \ldots, s_m\}$ with pmf $\{p(s_i): i=0,1,2,..,m\}$ must be sent over a noiseless communication channel
  - Develop an efficient coding scheme to minimize the expected word length

$$w(S) = \sum_{i=0}^{m} w(s_i)\, p(s_i); \; w(s_i) = \text{length of code word for } s_i$$

Solution:
Huffman code

- Analogy

failure sources	$\leftrightarrow$	messages
test results	$\leftrightarrow$	codeword
test algorithm	$\leftrightarrow$	coding scheme
constrained by available tests	----	unconstrained

When test costs are equal

$w(S) \le h^*(S)$

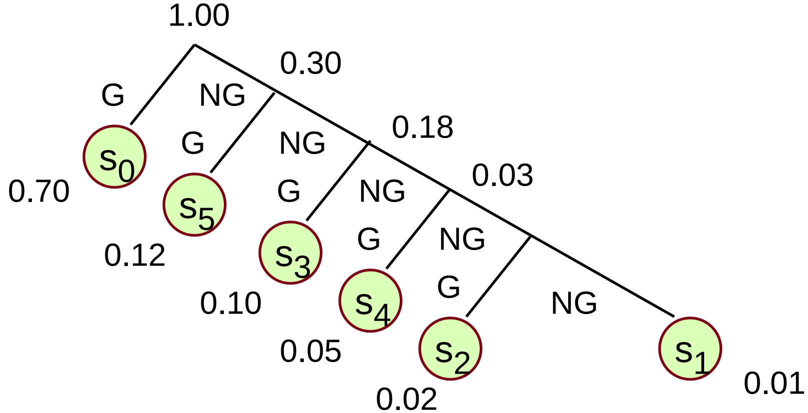
$w(x) \le h^*(x)\ \forall\, x$

What about when test costs are unequal?

<

Lower Bound on Cost-to-go Function

- Illustration of Huffman Code



- Useful properties of Huffman code
  - conditional Huffman code length from any node x, $w(x) \leq$ conditional average no. of tests, $l(x)$ for any test algorithm $\Rightarrow$ test point efficiency = $w(S)/l(S)$
  - lower bound on the optimal cost-to-go

$$HEF = h(x) = \frac{1}{p(x)} \sum_{s_i \in x} p(s_i) \sum_{j=1}^{w(s_i)} c_{[j]} \leq h^*(x); \; c_{[1]} \leq c_{[2]} \leq \ldots \leq c_{[n]}$$

- Simplified lower bound

$$HEF_1 = h(x) = \sum_{j=1}^{\lfloor w(S) \rfloor} c_{[j]} + \left(w(S) - \lfloor w(S) \rfloor\right) c_{\lfloor w(S) \rfloor + 1} \leq h^*(x)$$

Test Sequencing as AND/OR Graph Search

- AND/OR graph expresses the structure of test sequencing problem in the form of partial ordering among sub-problems (a la DP)
  - initial node of complete ambiguity, S = test sequencing problem to be solved
  - intermediate nodes = test sequencing sub-problems (OR, AND nodes)
  - goal (terminal) nodes = nodes of zero ambiguity, s_i (i.e., primitive sub-problems with known solution)
  - if an OR node x is in the solution tree, only one successor AND node (x, t_j) is in the solution tree. test t_j is the optimal test at OR node, x
  - if an AND node (x, t_j) is in the solution tree, then the immediate successor OR nodes x_{jp} and x_{jf}, are also in the solution tree (problem decomposition)
- Equivalent to splitting DP recursion into two parts
  - OR node : $h^*(x) = \min_j \{ c_j + h^*(x, t_j) \}$
  - AND node : $h^*(x, t_j) = p(x_{jp}).h^*(x_{jp}) + p(x_{jf}\text{f}).h^*(x_{jf})$
- **Key Idea**: Replace $h^*(x)$ by HEF $h(x)$, an easily computable estimate of optimal cost-to-go $\Rightarrow$ Top-down algorithm

Top-down Search Algorithm

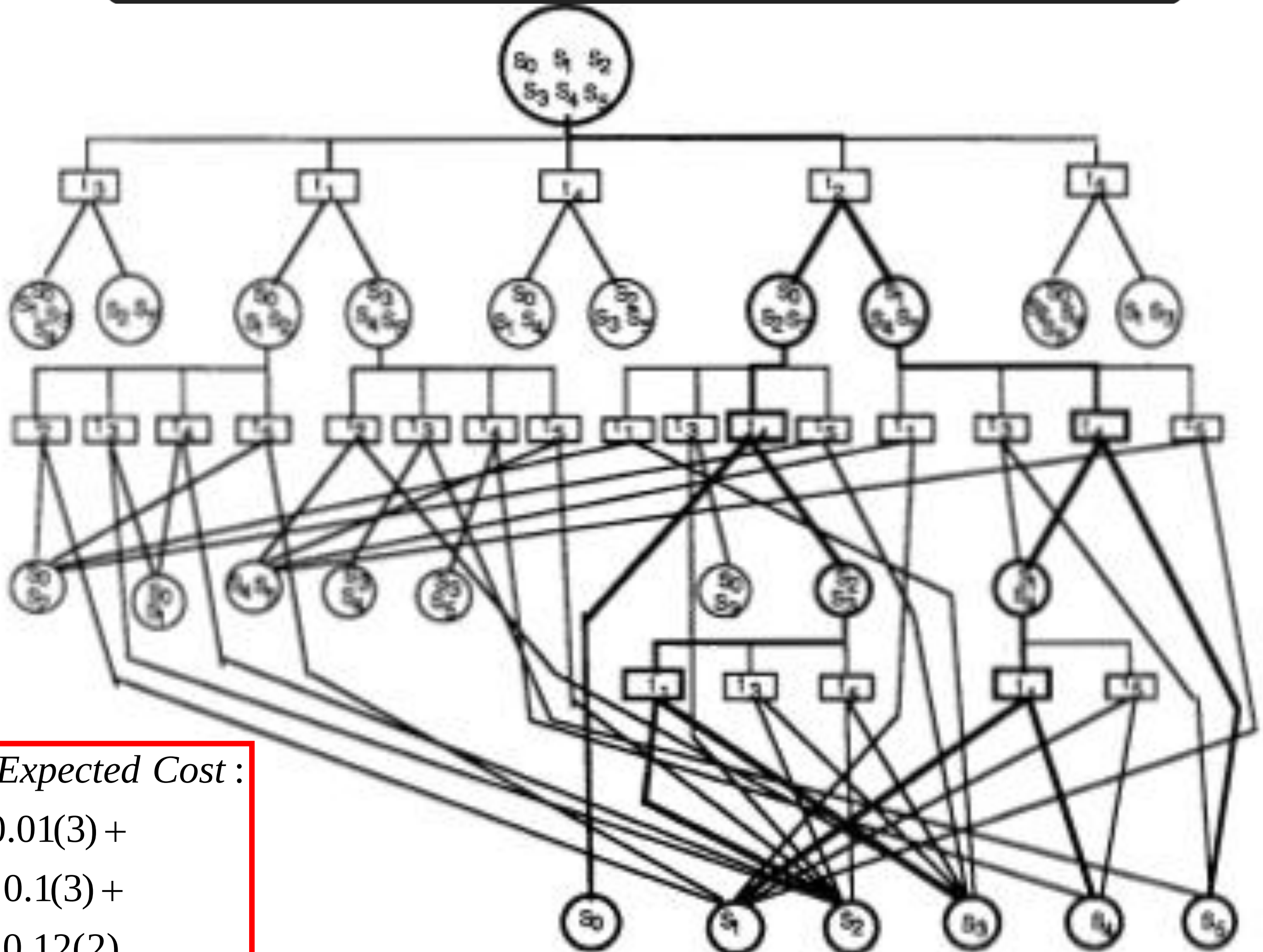
- Ordered best-first search algorithm AO*
  - expand only that node with most promise
  - node selection based on HEF
- Three basic operations performed repeatedly
  - top-down graph traversing
    - follow the best current (marked) partial solution graph
    - accumulate unexpanded terminal nodes
  - node selection and expansion
    - select unexpanded node with highest HEF, $h(x)$
    - expand x with each feasible test tj to get xjp, xj
    - if any successors = s_i, label them solved
    - add successors to graph (if not already present)
  - bottom-up cost revision (minor variation of DP)
    - update cost-to-go of expanded node
    - propagate change backward to the initial node

$$e = \min_j \{c_j + p(x_{jp}) f(x_{jp}) + p(x_{jf}) f(x_{jf})\}$$

$$\text{with intial } f(x_{jp}) = HEF_1(x_{jp}) = h(x_{jp});\ f(x_{jf}) = HEF_1(x_{jf}) = h(x_{jf})$$

- Leads to optimal solution because $f(S) \leq h^*(S) \leq f(S) \Rightarrow f(S) = h^*(S)$

Illustration of AO* Algorithm



$$Optimal\ Expected\ Cost:$$
$$0.7(2)+0.01(3)+$$
$$0.02(3)+0.1(3)+$$
$$0.05(3)+0.12(2)$$
$$=2.18$$

Applications of Static Test Sequencing

- Practical applications:
  - Helicopters, Military Trucks
  - Jet Engines
  - Medical Equipment
  - Fork lifts and military trucks
  - Optical Scanning Machines, Semiconductor Fabs……
- Controlled, blin… g machines
  - Used for qu… g

Hi-Tech Case Study

Technician Equipment	Elapsed Time to Troubleshoot
Expert Without TEAMS	7 Hours and 27 Minutes
Novice #1 With TEAMS	24 Minutes and 36 Seconds
Novice #2 Without TEAMS	8 Hours
Novice #3 With TEAMS	24 Minutes
Novice #4 With TEAMS	15 Minutes

Fault #	Expert Time to Diagnose the F…	Non-Expert Time to Diagnose the Fault	Non-Expert Time to Diagnose the Fault using TEAMS-RDS
1	15 min		15 min.
2	15 min	2 Hrs - gave up after wrong solution	15 min.
3	15 min	30 min. - got close but failed	15 min.
4	5 min.	1 Hr	5 min.
5	10 min	Gave up	5 min.

- Non-expert technicians can achieve the diagnostic capability equivalent to or better than that of diagnostic experts

Optimal Test (Sensor) Selection

- Problem: Optimal test selection while minimizing the total costs of tests subject to lower bound constraints on fault detection and fault isolation
  - Imperfect multi-outcome tests, and delays due to fault propagation, reporting and transmission
- Model
  - A set of failure sources, $S = \{s_1, s_2, \ldots, s_m, s_{m+1}\}$ and s_{m+1} is fault-free state
  - Probability of failure states, $P = \{p_1, p_2, \ldots, p_m, p_{m+1}\}$
  - A set of tests, $T = \{t_1, t_2, \ldots, t_n\}$ and test costs $C = \{c_1, c_2, \ldots, c_n\}$
  - A diagnostic dictionary matrix $D = [d_{ij}]_{(m+1)\times n}$ $d_{ij} = \text{Prob}\{\text{test } t_j \text{ fails} \mid \text{failure } s_i \text{ has occurred}\}$
- Problem Formulation

$$\min \sum_{j=1}^{n} c_j x_j$$
$$s.t.\ P_D(X) \geq \underline{P_D},$$
$$P_I(X) \geq \underline{P_I},$$
$$x_j \in \{0,1\},\ j = 1,2,\ldots,n$$

$$P_D(X) = \frac{1}{1-p_{m+1}} \sum_{i=1}^{m} p_i \left[1 - \prod_{j=1}^{n} (1-d_{ij})^{x_j}\right]$$

Detection probability of fault s_i

$$P_I(X) = \frac{1}{1-p_{m+1}} \sum_{i=1}^{m} p_i \{\prod_{\substack{k=1\\k\neq i}}^{m+1} [1 - \prod_{j=1}^{n} (1-d_{ij}-d_{kj}+2d_{ij}d_{kj})^{x_j}]\}$$

- Approach: Genetic algorithm (GA) and Lagrangian relaxation algorithm (LRA)
  - Genetic algorithm (GA) – for imperfect test selection with delayed and multiple test outcomes
  - Lagrangian relaxation algorithm (LRA) – for perfect test selection with multiple test outcomes
    - *Key advantage:* Provides an approximate duality gap (an upper bound on sub-optimality)

Optimal Test Selection: Simulation Results

- GA and LRA for perfect test selection problem

LC: Total cost of tests selected by LRA
LD: Deviation of LRA result from the best known result
LT: Average computation time of LRA
DG: Approximate duality gap of LRA
GC: Total cost of tests selected by GA
GD: Deviation of GA solution from the best known result
GT: Average computation time of GA
ENUM: Optimal cost of test set obtained by exhaustive search
ET: Average computation time for exhaustive search

Perfect *multi-outcome* Test Selection Problem

	m=10, n=10	m=10, n=15	m=15, n=15	m=30, n=40	m=50, n=60
LC	1.74	2.01	2.32	2.50	2.36
LD (%)	0	0	0.21	4.15	5.07
LT (s)	49.06	43.11	61.42	96.86	277.45
DG (%)	5.51	9.13	14.38	33.27	42.29
GC	1.74	2.01	2.32	2.40	2.25
GD (%)	0	0	0	0	0
GT (s)	0.96	1.42	1.50	4.44	12.68
ENUM	1.74	2.01	2.32	-	-
ET(s)	0.57	22.66	46.43	-	-

Performance of GA for the imperfect *multi-outcome* test selection problem

	m=10, n=10	m=10, n=15	m=15, n=15	m=30, n=40	m=50, n=60
GC	2.83	2.63	3.516	1.88	1.91
GD(%)	0	0	0	-	-
GT(s)	0.84	0.99	1.014	3.76	9.18
ENUM	2.83	2.63	3.5165	-	-
ET(s)	1.42	49.69	101.55	-	-

- Both GA and LRA generate very good test sets for the multi-outcome test selection problem
- Performance of GA is generally better than LRA
- Also applied to analog circuit test selection problems with excellent results

Origins & Importance of Reliability

- Formalized design techniques in early 19th century
  - standardizing commonly used parts (e.g., fasteners, bearings)
  - units of a given type tend to break or wear out in the same way
  - correlation between application loading and useful operating life (e.g., operating life of a bearing inversely proportional to rotational speed of inner ring and cube of radial load)
  - "reliability of a product is no better than the reliability of its least reliable component"

- Reliability becomes an engineering science
  - probability of successfully completing a prescribed mission
  - multiple engines versus single engine air planes (between WW I and WW II)
  - quantitative analysis techniques due to Robert Lusser and Erich Pieruschka (German VI missile during WW II) …. "a reliability chain is weaker than its weakest link"
  - requirements for reliability became part of military procurements during late 1950's

- Historically important in critical applications
  - military, aerospace, industrial, communications, patient monitors, power systems,..

- Recent trends
  - harsher environments, novice users, increasing repair costs, larger systems,...

Reliability Definitions - 1

- Reliability (British Standards Institution, Quality Vocabulary, Part I, 1987)
  - ability of an item to perform a <u>required</u> function under <u>stated conditions</u> for a <u>stated period of time</u>
  - required function => specification of satisfactory and unsatisfactory operation
  - stated conditions => total physical environment (mechanical, thermal and electrical)
  - stated period of time => time during which satisfactory operation is desired ("service life")
- Quantitative Definition of Reliability, R(t)
  - conditional probability that the system has survived the interval [0,t] given that it was operational at time t =0

$$R(t) = P\{\text{ system operates during } [0,t] \mid \text{system is operational at time } t = 0\}$$

  - repair cannot take place at all or cannot take place during a mission
  - also called **non-maintained** systems
- Alternate Definition
  - maximum number of failures anywhere in the system that the system can tolerate and still function correctly

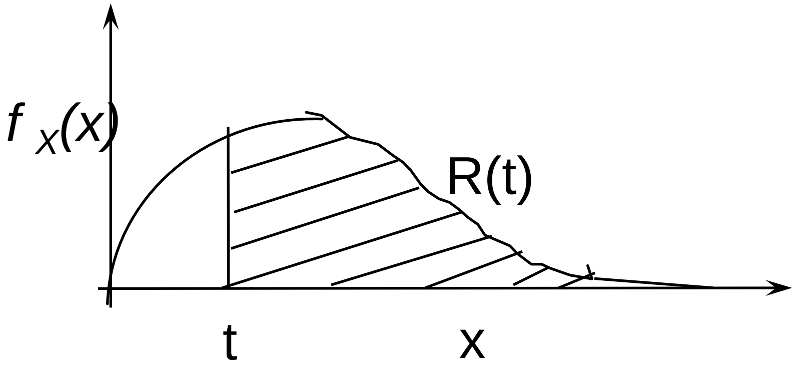
Reliability Definitions - 2

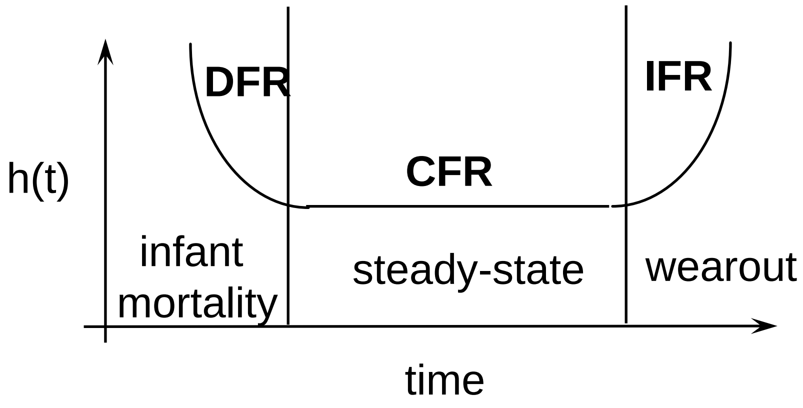
- Reliability in terms of lifetime distribution
  - X ~ lifetime or time to failure of a system and F_X (x) is the distribution function of X
  - reliability $R(t) = P\{ X > t \} = 1 - F(t)$
  - if $f_X(t)$ is the probability density function of X,

$$R(t) = \int_t^{\infty} f_X(x)dx$$

  - hazard rate (age-dependent failure rate, instantaneous failure rate), $h(t)$

$$h(t) = \frac{f_X(t)}{R(t)}$$

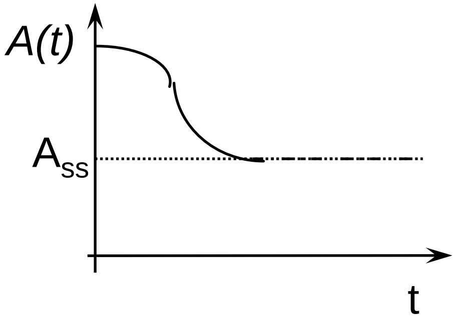




Reliability Definitions - 3

- Availability, A(t)
  - measure of the degree to which an item is in an operable state when called upon to perform
  - probability that the system is operational at time t

    A(t) = P { system is operational at time t}
  - repair is allowed $\Rightarrow$ maintained systems
  - if repair is not allowed, A(t) = R(t)
  - if $\lim_{t\to\infty}$ A(t) exists, have steady state availability, A_{ss}



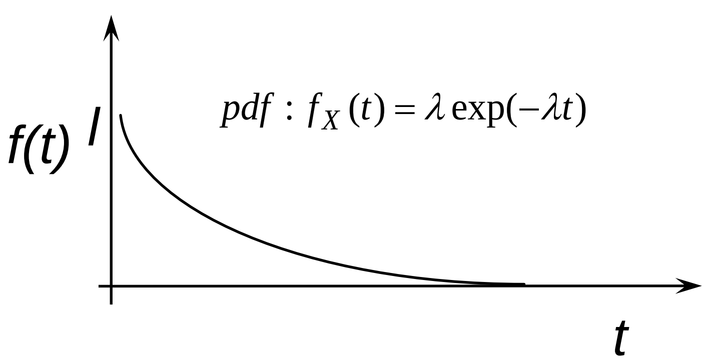
A_{ss} = expected fraction of time the system is available

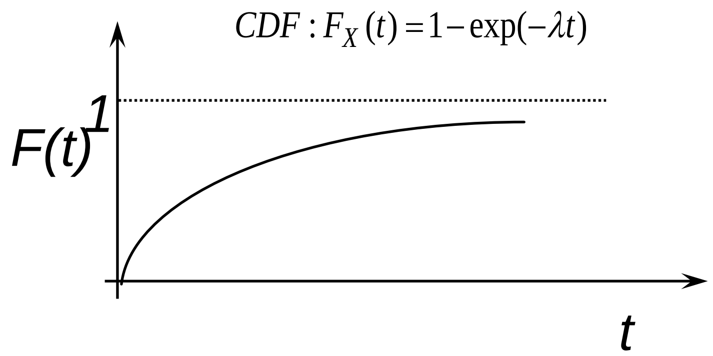
$$= \frac{UPTIME}{UPTIME + DOWNTIME}$$

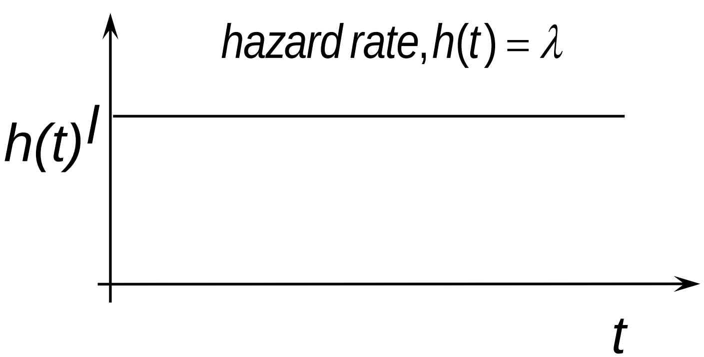
  - this equation is not valid for redundant systems with multiple UP states
- Maintainability
  - it is the degree to which an item is to be able to be restored to a specified operating condition

Failure Time Distributions - 1

- Exponential distribution
  - widely used in reliability analysis of equipment beyond the infant mortality period
  - constant failure rate (steady-state hazard rate)



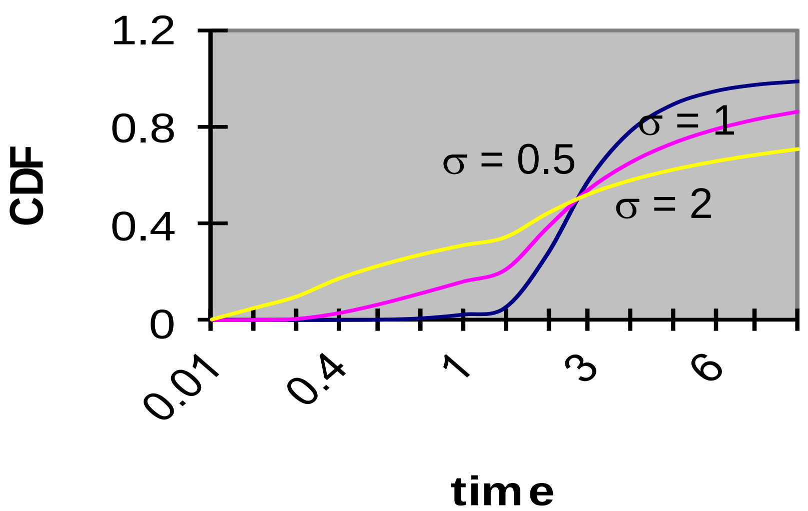


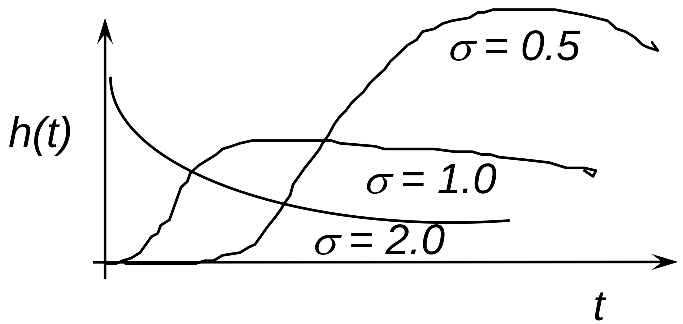


Failure Time Distributions - 2

- Lognormal distribution
  - used to describe failure time data obtained from accelerated testing of semiconductor devices
  - ln(failure time) is distributed normally

$$pdf: f(t) = \frac{1}{\sigma t \sqrt{2\pi}} \exp(-\frac{1}{2}[\frac{\ln(t)-\mu}{\sigma}]^2$$



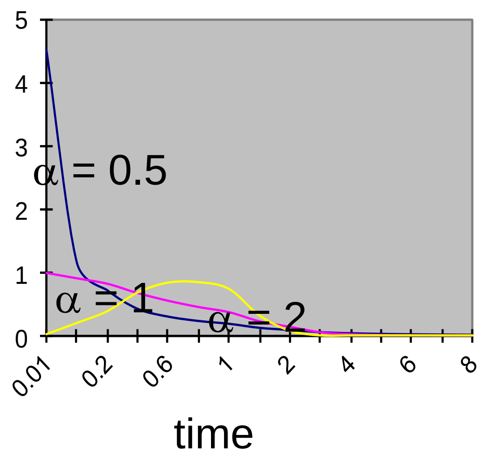


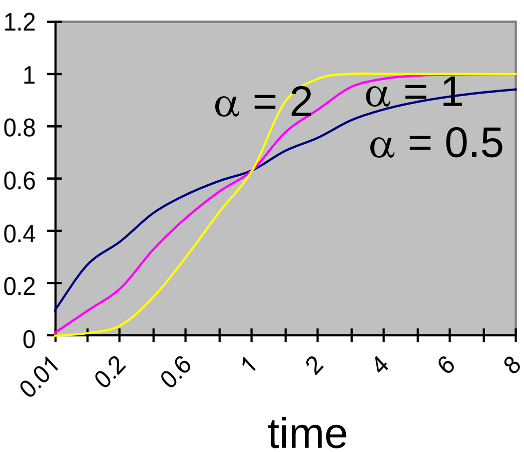
Regardless of μ and σ, the hazard rate of lognormal decreases at large times

Failure Time Distributions - 3

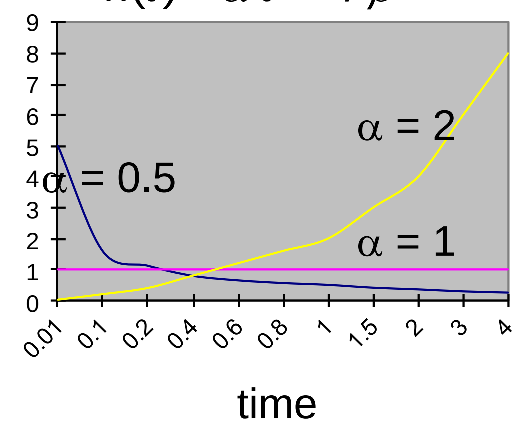
- Weibull distribution
  - the most widely used life distribution, especially in modeling infant mortality failures
  - hazard rate varies with device age

$$pdf: f(x) = \frac{\alpha}{\beta^{\alpha}} x^{\alpha-1} e^{-(\frac{x}{\beta})^{\alpha}} \qquad CDF: F(x) = 1 - e^{-(\frac{x}{\beta})^{\alpha}}$$





$$h(t) = \alpha t^{\alpha-1} / \beta^{\alpha}$$



- Weibull enables modeling of a variety of hazard (failure) rates
  - $\alpha < 1$ => decreasing failure rate with time ⇒ infant mortality period
  - $\alpha = 1$ => constant failure rate with time ⇒ exponential distribution => steady-state
  - $\alpha > 1$ => increasing failure rate with time ⇒ wear out period

Examples

- Example 1
  - the hazard rate of a piece of equipment is constant and estimated at 325,000 FITs (1 FIT= 10^{-9} failures per hour).
  - What is the probability that this device will first fail in the interval : (i) 0 to 6 months of operation? (ii) 6 to 12 months of operation? (iii) 6 to 12 months if it has survived the first 6 months?
  - if 100 of these systems are installed in the field but are not repaired when they fail, how many will still be expected to be working after 12 months?
  - what is the equipment MTTF? assuming an average repair time of 4 hours, what would the steady state availability be? how would this change if the average repair time were 50 hours?

- Example 2
  - assume the following for an integrated circuit: the steady-state hazard rate = 10 FITs, α=0.2 and the time to reach steady-state hazard rate is 10,000 hours. for a population of such devices, what percentage would be expected to fail: (i) in the first month of operation? (ii) in the first 6 months of operation? (iii) in the first 10 years of operation?

Accelerated Life (Stress) Testing

- Accelerated life (stress) testing
  - in an accelerated life test, environmental conditions, such as temperature, voltage, and humidity are altered to place a greater degree of stress on the device than there would be in actual usage. This increased level of stress is applied to *accelerate* whatever reaction is believed to lead to failure, hence the term accelerated stress testing.
- Accelerated life model
  - linear relationship between failure times at different sets of conditions

$$t_{use} = A\, t_{stress}$$

t_{use} = failure time of device at use conditions

t_{stress} = failure time of that same device under stress conditions

A = acceleration factor

- implications

$$CDF: F_u(t) = F_s(t/A);\; pdf: f_u(t) = \frac{1}{A} f_s(t/A)$$

$$reliability: R_u(t) = R_s(t/A);\; hazard\ rate: h_u(t) = \frac{1}{A} h_s(t/A)$$

For Weibull:

$$h_s(t) = A h_u(At) = A\alpha(At)^{\alpha-1}/\beta^{\alpha} = A^{\alpha} h_u(t)$$

Modeling Acceleration Constant - 1

- Types of stress: temperature, temperature cycling, operating voltage, electrical stress,….
- Acceleration constant for temperature effect, A_T:

$$A_T = e^{\frac{E_a}{k_B}[\frac{1}{T_1} - \frac{1}{T_2}]} \Rightarrow \ln A_T = \frac{E_a}{k_B}[\frac{1}{T_1} - \frac{1}{T_2}]$$

  - E_a = activation energy for temperature (0.4 *ev*); k_B = Boltzmann constant (1.38 x10^{-23} J/K = 8.6x x10^{-5}ev/K); T_1 and T_2 are temperatures (^{0} K)
  - acceleration constant for temperature cycling (general form unknown): temperature cycling of devices results in decreasing hazard rates as the number of cycles increases
- Acceleration constant for operating voltage, A_V

$$A_V = e^{\frac{C}{t_{ox}}[V_1 - V_2]} \Rightarrow \ln A_V = \frac{C}{t_{ox}}[V_1 - V_2]$$

  - C = voltage acceleration constant in angstrom/volt (300-600); t_{ox} = oxide thickness in angstroms (250); V_1 = stress voltage in volts; V_2 = operating voltage in volts

Modeling Acceleration Constant - 2

- Acceleration constant for electrical stresses (power, voltage, current) on passive components, A_E

$$A_E = e^{m[p_1 - p_0]} \Rightarrow \ln A_E = m[p_1 - p_0]$$

  - m = parameter to be determined from MIL-HDBK-217F(0.006-0.150)
  - p_1 = percent of maximum rated electrical stress
  - p_0 = reference percent of rated electrical stress (25%)

  p $\Rightarrow$ power for resistors; voltage for capacitors; current for relays and switches

- Environmental application factors, *E*

permanent structures:	1.0
ground shelters or not temperature controlled:	1.1
manholes, poles:	1.5
vehicular-mounted:	8.0

- Packaging
  - hermetic: ICs (1.0 - 3.0); Diodes & Transistors (1.0-3.0); all passive components: (1.0-3.0)
  - plastic: ICs (1.2 - 3.6); Diodes & Transistors (1.0-3.6); all passive components: (1.0-3.0)

Burn-in for Screening out Defects

- Burn-in is an effective means to screen out defective components
  - typically combines electrical stresses and temperature over a period of time in order to induce temperature- and voltage-dependent failure mechanisms in a relatively short time
  - **static burn-in**: apply dc bias at an elevated temperature to reverse bias as many junctions as possible in the device
  - **dynamic burn-in**: operate the device by simulating actual system operation (very effective)
  - for Weibull, hazard rate after burn-in:

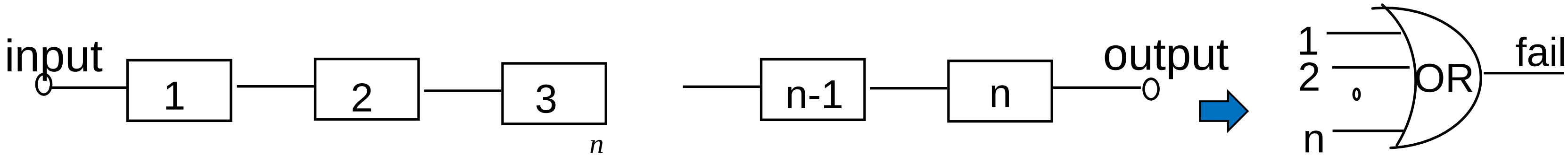
$$\text{Burn-in Time} = t_{bi};\ \text{Effective operation time due to burn-in} = t_{eff} = A_T A_V t_{bi}$$

$$h(t) = \frac{\alpha}{\beta^{\alpha}}(t_{eff} + t)^{\alpha-1}$$

- Example
  - if a device has an early-life hazard rate of 20,000 FITs, $\alpha = 0.2$ and no burn-in is performed, what % of devices will fail during the first month (730 hours) of operation? (0.04%)
  - what percentage of devices will fail during the first month if they have been burned in for 10 hours at 150^0C? (0.0015%)
  - hazard rate drops from 548 FITs with no burn-in to 21 FITs with burn-in

Reliability Block Diagrams - 1

- Series System
  - failure of any one component leads to system failure



$$\text{Reliability}: R(t) = \prod_{i=1}^{n} R_i(t)$$

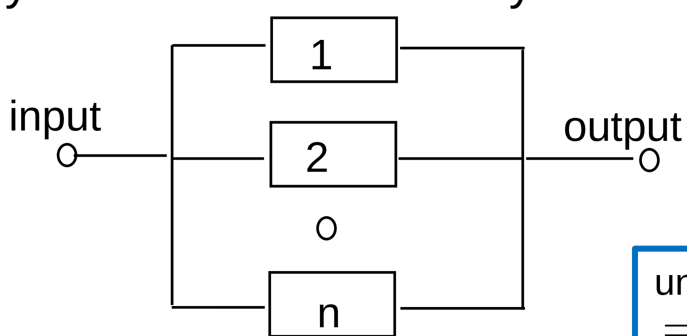
$$\textit{CDF of FailureTime}: F_X(t) = 1 - \prod_{i=1}^{n} (1 - F_{X_i}(t))$$

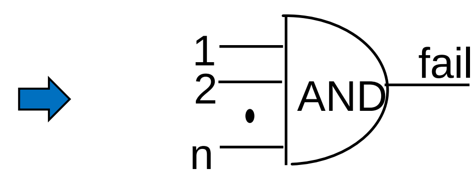
  - reliabilities multiply for a series system $\Rightarrow$ reliability is less than that of weakest element

- Parallel (redundant) system
  - a system failure occurs only if all components fail

$$\text{Reliability}: R(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t))$$

$$\textit{CDF of FailureTime}: F(t) = \prod_{i=1}^{n} F_{X_i}(t)$$





unreliabilities multiply for a parallel system $\Rightarrow$ redundancy improves system reliability

Reliability Block Diagrams - 2

- *k*-out-of-*n* system ⇒ at least *k* out of *n* components must function
  - assuming identical components

$$\text{Reliability}: R(t) = \sum_{i=k}^{n} \binom{n}{i} R(t)^{i} (1 - R(t))^{n-i}$$

$$\text{CDF of Failure Time}: F(t) = \sum_{i=n-k+1}^{n} \binom{n}{i} F(t)^{i} (1 - F(t))^{n-i}$$

  - for non-identical components

$$\text{Reliability}: R(t) = \sum_{|I| \geq k} \Big(\prod_{i \in I} R_i(t)\Big)\Big(\prod_{i \notin I} (1 - R_i(t))\Big)$$

$$\text{CDF of Failure Time}: F(t) = \sum_{|I| \geq n-k+1} \Big(\prod_{i \in I} F_i(t)\Big)\Big(\prod_{i \notin I} (1 - F_i(t))\Big)$$

I is the subset that has
at least *k* (or *n*-*k*+1) components

Reliability Block Diagrams - 3

- k-out-of-n system ⇒ at least k out of n components must function
  - CDF $F(t)$ in terms of symmetric polynomials

$$CDF\ of\ FailureTime: F(t) = \sum_{i=n-k+1}^{n} (-1)^{i+k-n-1} \binom{i-1}{n-k} S_i(\mathbf{F})$$

$$S_i(\mathbf{F}) = \sum_{|I|=i} \prod_{j \in I} F_j$$

  - $O(n^2)$ algorithm for evaluating CDF $F(t)$ for non-identical component case

$$S_i(j) = \textit{symmetric polynomial of degree i chosen out of } \mathbf{F} \text{ with j elements}$$

$$S_1(1) = F_1$$

$$S_1(j) = S_1(j-1) + F_j \ \ for \ \ j > 1$$

$$S_j(j) = S_{j-1}(j-1) * F_j \ \ for \ \ j > 1$$

$$S_i(j) = S_i(j-1) + F_j * S_i - 1(j-1) \ for \ 1 < i < j$$

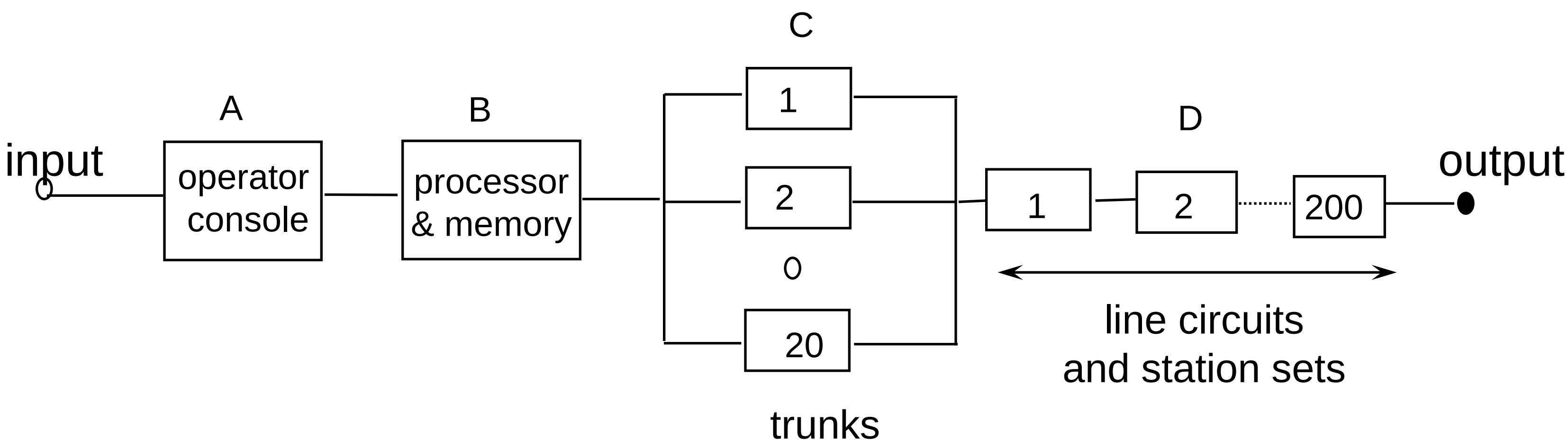
$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty (1 - F(t))dt$$

  - example: $k = 2$, $n=3$ ⇒ $F(t) = S_2(\mathbf{F}) - 2S_3(\mathbf{F})$

$$= F_1(t)\, F_2(t) + F_1(t)\, F_3(t) + F_2(t)\, F_3(t) - 2\, F_1(t)\, F_2(t)\, F_3(t)$$

Application to a PBX Example

- PBX example
  - an operator console, system processor and memory, 20 trunks and 200 lines and station sets
  - at least 18 out of 20 trunks must be working for the system to work



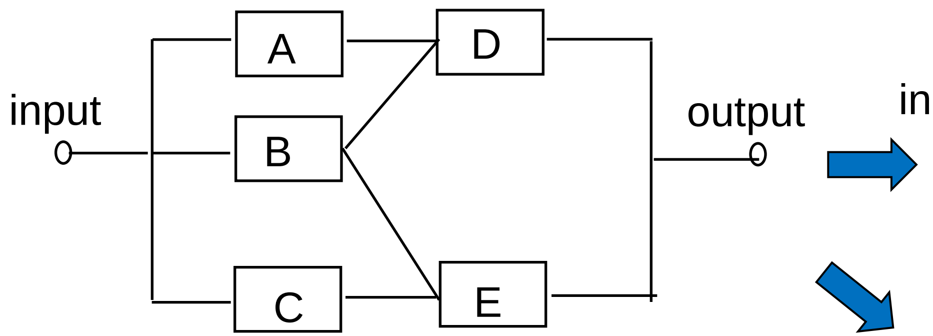
$$Reliability: R_{PBX}(t) = R_A(t)\, R_B(t)\, R_C(t)\, R_D(t)$$

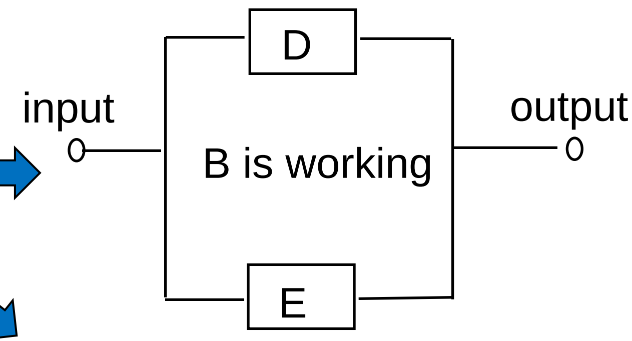
$$R_C(t) = \sum_{i=18}^{20} \binom{20}{i} (R_{trunk}(t))^i (1 - R_{trunk}(t))^{20-i}$$

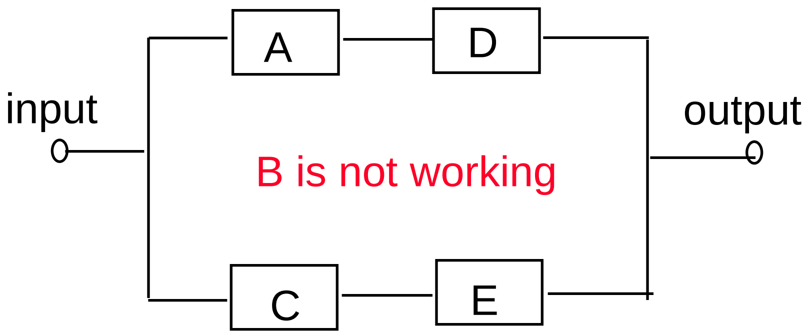
$$R_D(t) = (R_{ls}(t))^{200}\ ; R_{ls}(t) = \textit{reliability of a line circuit and its station}$$

Reliability of Complex Structures - 1

- Decomposition and Factoring method
  - what if structure can not be decomposed into series, parallel, or *k*-out-of-*n* subsystems?







$$R_{sys}(t) = R_{sys}(t|B)R_B(t) + R_{sys}(t|\bar{B})(1 - R_B(t))$$

$$R_{sys}(t|B) = 1 - (1 - R_D(t))(1 - R_E(t))$$

$$= R_D(t) + R_E(t) - R_D(t)R_E(t)$$

$$R_{sys}(t|\bar{B}) = R_A(t)R_D(t) + R_C(t)R_E(t) - R_A(t)R_D(t)R_C(t)R_E(t)$$

Minimal Path Set Method

- Minimal path sets
  - a path set is a continuous line drawn from the input to the output of the block diagram
  - a minimal path set is a **minimal set of components whose functioning ensures the functioning of the system**
  - **key**: a system will function if and only if all the components of at least one minimal path set are functioning
  - system reliability = P{ at least one minimal path is functioning}
  - example: minimal path sets are : {A,D}, {B,D},{B,E},{C,E}

$$\textit{Let } a,b,c,d,e \textit{ denote states of components } (a=1 \Rightarrow \textit{working}; a=0 \Rightarrow \textit{failed})$$

$$\begin{aligned} R_{sys} &= P\{\max(ad,bd,be,ce)=1\} \\ &= P\{1-(1-ad)(1-bd)(1-be)(1-ce)=1\} \\ &= P\{b(d+e-de)+(1-b)(ad+ce-adce)=1\} \\ &= R_B(t)(R_D(t)+R_E(t)-R_D(t)R_E(t)) \\ &\quad +(1-R_B(t))(R_A(t)R_D(t)+R_C(t)R_E(t)-R_A(t)R_D(t)R_C(t)R_E(t)) \end{aligned}$$

number of minimal paths can be exponential

  - use the fact that $a^2 = a$, etc.

Minimal Cut Set Method

- Minimal cut sets
  - a minimal cut set is a **minimal set of components whose failure ensures the failure of the system**
  - **key**: a system will fail if and only if all the components of at least one minimal cut set are not functioning
  - system reliability = P{ at least one component in each cut set is functioning}
  - example: minimal cut sets are : {A,B,C},{D,E}, {B,A,E},{B,C,D}

$$R_{sys}(t) = P\{\max(a,b,c)\max(d,e)\max(b,c,d)\max(a,b,e)=1\}$$
$$= P\{(1-(1-a)(1-b)(1-c))(1-(1-d)(1-e))$$
$$(1-(1-b)(1-c)(1-d))(1-(1-a)(1-b)(1-e))=1\}$$
$$= R_B(t)(R_D(t)+R_E(t)-R_D(t)R_E(t))$$
$$+(1-R_B(t))(R_A(t)R_D(t)+R_C(t)R_E(t)-R_A(t)R_D(t)R_C(t)R_E(t))$$

number of minimal cut sets can be exponential

Bounds on System Reliability -1

- Bounds based on minimal cut sets and minimal path sets
  - A = set of minimal paths
  - C = set of minimal cut sets
  - R_i = reliability of i^{th} component (time is implicit)

$$\prod_{X\in C}\{1-\prod_{i=1}^{n}(1-R_i)^{1-x_i}\}\le R_{sys}\le 1-\{\prod_{X\in A}(1-\prod_{i=1}^{n}R_i^{x_i})\}$$

- Example: corresponds to substituting reliability in the structure function

$$(1-(1-R_A)(1-R_B)(1-R_C))(1-(1-R_D)(1-R_E))$$
$$(1-(1-R_B)(1-R_C)(1-R_D))(1-(1-R_A)(1-R_B)(1-R_E))$$
$$\le R_{sys}\le 1-(1-R_AR_D)(1-R_BR_D)(1-R_BR_E)(1-R_CR_E)$$

Bounds on System Reliability - 2

- Key idea

$$P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{i=1}^{n} P(E_i) - \sum_{i}\sum_{j<i} P(E_i E_j) + \sum_{i}\sum_{j<i}\sum_{k<j<i} P(E_i E_j E_k)$$

$$- \ldots\ldots + (-1)^{n+1} P(E_1 E_2 \ldots\ldots E_n)$$

- Bounds based on minimal path sets
  - works good when individual reliabilities are small

$$\sum_{i\in A} P(\pi_i) - \sum_{i}\sum_{i<j} P(\pi_i \pi_j) \le R_{sys} \le \sum_{i\in A} P(\pi_i)$$

$$\pi_i = i^{th} \text{ min}\,imal\;\; path\ elements$$

$$A = \text{min}\,imal\;\; paths$$

- Bounds based on minimal cut sets
  - works good when individual reliabilities are large (close to 1)

$$\sum_{i\in C} P(F_i) - \sum_{i}\sum_{i<j} P(F_i F_j) \le 1 - R_{sys} \le \sum_{i\in C} P(F_i)$$

$$F_i = i^{th} \text{ min}\,imal\ cut\ set\ elements$$

$$C = \text{min}\,imal\ cut\ sets$$

References

- D.J. Klinger, Y. Nakada, and M.A. Menendez, AT&T Reliability Manual, Van Nostrand Reinhold, New York, 1990.
- R.A. Shaner, K.S. Trivedi and A. Pauliafito, Performance and Reliability Analysis of Computer Systems, Kulwer Academic Publishers, Boston, 1996.
- J.I. Ansell and M.J. Phillips, Practical Methods for Reliability Data Analysis, Oxford Science Publications, Clarendon Press, Oxford, 1994.
- S. Ross, Introduction to Probability Models, Academic Press, New York, 1985.

Summary

- Testability
  - Importance of Testing
  - Onboard and off-board diagnosis
  - Multiple Fault Diagnosis Methods
  - Sequential Fault Diagnosis
- Reliability
  - Importance of Reliability
  - Reliability Definitions
  - Device Reliability
  - System Reliability Modeling