Module 2: Combinatorial Modeling Methods
Introduction to Combinatorial Methods

• Combinatorial validation methods are the simplest kind of analytical/numerical techniques and can be used for reliability and availability modeling under certain assumptions.

• Assumptions are that component failures are independent, and for availability, repairs are independent.

• When these assumptions hold, simple formulas for reliability and availability exist.
Lecture Outline

• Review definition of reliability
• Failure rate
• System reliability
  – Maximum
  – Minimum
  – $k$ of $N$
• Reliability formalisms
  – Reliability block diagrams
  – Fault trees
  – Reliability graphs
• Reliability modeling process
Reliability

• One key to building highly available systems is the use of reliable components and systems.

• Reliability: The reliability of a system at time \( t \) \( (R(t)) \) is the probability that the system operation is proper throughout the interval \([0, t]\).

• Probability theory and combinatorics can be directly applied to reliability models.

• Let \( X \) be a random variable representing the time to failure of a component. The reliability of the component at time \( t \) is given by
  \[
  R_X(t) = P[X > t] = 1 - P[X \leq t] = 1 - F_X(t).
  \]

• Similarly, we can define unreliability at time \( t \) by
  \[
  U_X(t) = P[X \leq t] = F_X(t).
  \]
Failure Rate

What is the rate that a component fails at time $t$? This is the probability that a component that has not yet failed fails in the interval $(t, t + \Delta t)$, as $\Delta t \to 0$.

Note that we are not looking at $P[X \in (t, t + \Delta t)]$. Rather, we are seeking $P[X \in (t, t + \Delta t) \mid X > t]$.

$$P[X \in (t, t + \Delta t) \mid X > t] = \frac{P[X \in (t, t + \Delta t), X > t]}{P[X > t]}$$

$$= \frac{P[X \in (t, t + \Delta t)]}{1 - F_X(t)}$$

$$= \frac{f_X(t)}{1 - F_X(t)} = r_X(t)$$

$r_X(t)$ is called the failure rate or hazard rate.

* this is a heurist
Typical Failure Rate

Break in  Normal operation  Wear out

\( r_X(t) \)

time
System Reliability

While $F_X$ can give the reliability of a component, how do you compute the reliability of a system?

System failure can occur when one, all, or some of the components fail. If one makes the *independent failure assumption*, system failure can be computed quite simply. The independent failure assumption states that all component failures of a system are independent, i.e., the failure of one component does not cause another component to be more or less likely to fail.

Given this assumption, one can determine:

1) Minimum failure time of a set of components
2) Maximum failure time of a set of components
3) Probability that $k$ of $N$ components have failed at a particular time $t$. 
Maximum of \( n \) Independent Failure Times

Let \( X_1, \ldots, X_n \) be independent component failure times. Suppose the system fails at time \( S \) if all the components fail.

Thus, \( S = \max \{X_1, \ldots, X_n\} \)

What is \( F_s(t) \)?

\[
F_s(t) = P[S \leq t] \\
= P[X_1 \leq t \ \text{AND} \ X_2 \leq t \ \text{AND} \ \ldots \ \text{AND} \ X_n \leq t] \\
= P[X_1 \leq t] \ P[X_2 \leq t] \ \ldots \ P[X_n \leq t] \quad \text{By independence} \\
= F_{X_1}(t)F_{X_2}(t)\ldots F_{X_n}(t) \quad \text{By definition} \\
= \prod_{i=1}^{n} F_{X_i}(t)
\]
Minimum of $n$ Independent Component Failure Times

Let $X_1, \ldots, X_n$ be independent component failure times. A system fails at time $S$ if any of the components fail. Thus, $S = \min\{X_1, \ldots, X_n\}$.

What is $F_S(t)$?

$$F_S(t) = P[S \leq t] = P[X_1 \leq t \text{ OR } X_2 \leq t \text{ OR } \ldots \text{ OR } X_n \leq t]$$

Trick: If $A_i$ is an event, and $\overline{A}_i$ is the set complement such that $A_i \cup \overline{A}_i = \Omega$ and $A_i \cap \overline{A}_i = \emptyset$, then

$$P[A_1 \text{ OR } A_2 \text{ OR } \ldots \text{ OR } A_n] = 1 - P[\overline{A}_1 \text{ AND } \overline{A}_2 \text{ AND } \ldots \text{ AND } \overline{A}_n]$$

This is an application of the law of total probability (LOTP).
Minimum cont.

\[ F_s(t) = P[X_1 \leq t \text{ OR } X_2 \leq t \text{ OR } \ldots \text{ OR } X_n \leq t] \]
\[ = 1 - P[X_1 > t \text{ AND } X_2 > t \text{ AND } \ldots \text{ AND } X_n > t] \quad \text{By trick} \]
\[ = 1 - P[X_1 > t] \cdot P[X_2 > t] \ldots P[X_n > t] \quad \text{By independence} \]
\[ = 1 - (1 - P[X_1 \leq t])(1 - P[X_2 \leq t]) \ldots (1 - P[X_n \leq t]) \quad \text{By LOTP} \]
\[ = 1 - \prod_{i=1}^{n}(1 - F_{X_i}(t)) \]
Let $X_1, \ldots, X_n$ be component failure times that have identical distributions (i.e., $F_{X_1}(t) = F_{X_2}(t) = \ldots$). The system has failed by time $S$ if $k$ or more of the $N$ components have failed by $S$

$$F_S(t) = P[\text{at least } k \text{ components failed by time } t]$$

$$= P[ \text{exactly } k \text{ failed OR exactly } k + 1 \text{ failed OR } \ldots \text{ OR exactly } N \text{ failed}]$$

$$= P[\text{exactly } k \text{ failed}] + P[\text{exactly } k + 1 \text{ failed}] + \ldots + P[\text{exactly } N \text{ failed}]$$

What is $P[\text{exactly } k \text{ failed}]$?

$$= P[k \text{ failed and } (N - k) \text{ have not}]$$

$$= \binom{N}{k} F_X(t)^k (1 - F_X(t))^{N-k}$$

where $F_X(t)$ is the failure distribution of each component.

Thus, $F_S(t) = \sum_{i=k}^{N} \binom{N}{i} F_X(t)^i (1 - F_X(t))^{N-i}$

- by independence and axiom of probability.
**k of N in General**

For non-identical failure distributions, we must sum over all combinations of at least \( k \) failures.

Let \( G_k \) be the set of all subsets of \( \{X_1, \ldots, X_N\} \) such that each element in \( G_k \) is a set of size at least \( k \), i.e.,

\[
G_k = \{g_i \subseteq \{X_1, \ldots, X_N\} : |g_i| \geq k\}.
\]

The set \( G_k \) represents all the possible failure scenarios.

Now \( F_S \) is given by

\[
F_S(t) = \sum_{g \in G_k} \left( \prod_{X \in g} F_X(t) \right) \left( \prod_{X \notin g} \left(1 - F_X(t)\right) \right)
\]
Component Building Blocks

Complex systems can be analyzed hierarchically.

Example: A computer fails if both power supplies fail or both memories fail or the CPU fails.

System problem is one of a minimum: the system fails when the first of three subsystems fails... proper formulation is

- Power supply subsystem is a maximum: both must fail
- Memory subsystem is a maximum: both must fail

\[ F_S(t) = 1 - (1 - F_{P1}(t)F_{P2}(t))(1 - F_{M1}(t)F_{M2}(t))(1 - F_C(t)) \]

Probability at least 1 power source is up at \( t \)

Probability all 3 subsystems are up at \( t \)
**Summary**

A system comprises $N$ components, where the component failure times are given by the random variables $X_1, \ldots, X_N$. The system fails at time $S$ with distribution $F_S$ if:

<table>
<thead>
<tr>
<th>Condition:</th>
<th>Distribution:</th>
</tr>
</thead>
<tbody>
<tr>
<td>all components fail</td>
<td>$F_S(t) = \prod_{i=1}^{N} F_{X_i}(t)$</td>
</tr>
<tr>
<td>one component fails</td>
<td>$F_S(t) = 1 - \prod_{i=1}^{N} \left(1 - F_{X_i}(t)\right)$</td>
</tr>
<tr>
<td>$k$ components fail, identical distributions</td>
<td>$F_S(t) = \sum_{i=k}^{N} \binom{N}{i} F_{X_i}(t)^i \left(1 - F_{X_i}(t)\right)^{N-i}$</td>
</tr>
<tr>
<td>$k$ components fail, general case</td>
<td>$F_S(t) = \sum_{g \subseteq G_k} \left( \prod_{X \in g} F_{X_i}(t) \right) \left( \prod_{X \notin g} \left(1 - F_{X_i}(t)\right) \right)$</td>
</tr>
</tbody>
</table>
Reliability Formalisms

There are several popular graphical formalisms to express system reliability. The core of the solvers is the methods we have just examined. In particular, we will examine

- Reliability Block Diagrams
- Fault Trees
- Reliability Graphs

There is nothing particularly special about these formalisms except their popularity. It is easy to implement these formalisms, or design your own, in a spreadsheet, for example.
Reliability Block Diagrams

- Blocks represent components.
- A system failure occurs if there is no path from source to sink.

Series:
- System fails if any component fails.

Parallel:
- System fails if all components fail.

$k$ of $N$:
- System fails if at least $k$ of $N$ components fail.
Example

A NASA satellite architecture under study is designed for high reliability. The major computer system components include the CPU system, the high-speed network for data collection and transmission, and the low-speed network for engineering and control. The satellite fails if any of the major systems fail.

There are 3 computers, and the computer system fails if 2 or more of the computers fail. Failure distribution of a computer is given by $F_C$.

There is a redundant (2) high-speed network, and the high-speed network system fails if both networks fail. The distribution of a high-speed network failure is given by $F_H$.

The low-speed network is arranged similarly, with a failure distribution of $F_L$. 
RBD Example

\[ F_S(t) = 1 - \left( 1 - \left( \sum_{i=2}^{3} \binom{3}{i} F_C^i(t)(1 - F_C(t))^{3-i} \right) \right) \left( 1 - (F_H(t))^2 \right) \left( 1 - (F_L(t))^2 \right) \]
RBD Example

\[ F_S(t) = 1 - \left(1 - \sum_{i=2}^{3} F_C^i(t)(1 - F_C(t))^{3-i}\right) \left(1 - (F_H(t))^2 \right) \left(1 - (F_L(t))^2 \right) \]

Probability all three systems survive to t
RBD Example

\[ F_S(t) = 1 - \left( 1 - \left( \sum_{i=2}^{3} \binom{3}{i} F_C^i(t)(1 - F_C(t))^{3-i} \right) \right) \left( 1 - (F_H(t))^2 \right) \left( 1 - (F_L(t))^2 \right) \]

Probability low speed network survives to t
RBD Example

\[ F_S(t) = 1 - \left( 1 - \left( \sum_{i=2}^{3} \binom{3}{i} F_C^i(t)(1 - F_C(t))^{3-i} \right) \right) \left( 1 - (F_H(t))^2 \right) \left( 1 - (F_L(t))^2 \right) \]

Probability both components of low speed network fail by t
Background: Series-Parallel Graphs

- Analysis of RBD is based on the notion of *series-parallel graphs*
- Defined recursively as follows:
- A series-parallel graph is comprised of nodes, edges, and subgraph connectors (SGC)
  - An edge connects a node and an SGC, or two SGCs
  - Every series-parallel graph has an SCG as source, and an SCG as sink
- All series-parallel graphs have one of the following forms:
  - Single node
  - Series of two SPGs
    (understanding that the sink of one SPG is identically the source of the other)
  - Parallel of two SPGs
    (understanding that the source and sink endpoints connect (respectively) to the source and sink endpoints of the SPGs)

Every node has a failure distribution
Failure time of series subgraph is minimum failure time of SPGs in series
Failure time of parallel subgraph is maximum failure time of SPGs in parallel

Given a graph, identify the series-parallel decomposition
Series-Parallel Decomposition of NASA example

Use formula for series, applied to failure distribution of component subgraphs, which are ??

We can express this one, because node distributions are known
Fault Trees

- Components are leaves in the tree
- A component fails = logical value of \textit{true}, otherwise \textit{false}.
- The nodes in the tree are boolean AND, OR, and \(k\) of \(N\) gates.
- The system fails if the root is \textit{true}.

**AND gates**

\textit{true} if all the components are \textit{true} (fail).

**OR gates**

\textit{true} if any of the components are \textit{true} (fail).

**\(k\) of \(N\) gates**

\textit{true} if at least \(k\) of the components are \textit{true} (fail).
Fault Tree Example

2 of 3

OR

AND

AND

C1 C2 C3

H1 H2

L1 L2
Reliability Graphs

- The *arcs* represent components and have failure distributions.
- A failure occurs if there is no path from source to sink.

Can implement series:

Can implement parallel:
Reliability Graph Example

Reliability graphs can implement more complex interactions.
- Graphs from RBD and FT are essentially “series-parallel”, a special class of graph
- A reliability graph is more general

For example, a telephone network “fails” if there is no path from source to sink.

How do we solve this?
Solving by Conditioning

Recall that \( P[E \mid F] = \frac{P[E \cap F]}{P[F]} \)

If \( F \) and \( \bar{F} \) are complementary events, i.e.,
\[
F \cup \bar{F} = \Omega \quad \text{and} \quad F \cap \bar{F} = \emptyset
\]

then there is a trick:
\[
P[E] = P[E \cap F] + P[E \cap \bar{F}]
\]
\[
P[E] = P[E \mid F]P[F] + P[E \mid \bar{F}]P[\bar{F}]
\]

If you can solve \( P[E \mid F], P[F], P[E \mid \bar{F}], \) and \( P[\bar{F}] \), then you can solve \( P[E] \).
First, condition the system on link C being failed. Then the system becomes the series AD in parallel with the series BE.

\[ S \] is time of failure

\[
Pr\{A \text{ and } D \text{ alive at } t\}
\]

\[
F_{S|C \text{ Fail}}(t) = P[S \leq t | C \leq t] = (1 - (1 - F_A(t))(1 - F_D(t))(1 - (1 - F_B(t))(1 - F_E(t))))
\]

and \( P[C \leq t] = F_C(t) \)
Second, *condition* the system on link C being up.

- System fails if either both A&B fail, or both D and E fail --- a series

\[
F_{S|C\text{up}}(t) = P[S \leq t \mid C > t] = 1 - (1 - F_A(t)F_B(t))(1 - F_D(t)F_E(t)),
\]

and \( P[C > t] = 1 - P[C \leq t] = 1 - F_C(t) \)

Thus, \( F_S(t) = F_{S|C\text{Fail}}(t)F_C(t) + F_{S|C\text{up}}(t)(1 - F_C(t)) \).
**Conditioning Fault Trees**

It is also possible to use conditioning to solve more complex fault trees. If the same component appears more than once in a fault tree, it violates the independent failure assumption. However, a conditioned fault tree can be solved.

Example: A component \( C \) appears multiple times in the fault tree conditioned on \( C \) being up (i.e. have \( C=0 \) as input in the fault tree):

- AND gates with \( C \) as input become 0 (impossible for all components to be failed)
- OR gates with \( C \) as input remain OR gates without \( C \) as input
- \( k \) of \( N \) gates become \( k \) of \( (N-1) \) gates

Conditioned on \( C \) being down (i.e. have \( C=1 \) as input in the fault tree):

- AND gates with \( C \) as input remain AND gates without \( C \) as input
- OR gates with \( C \) as input become 1 (any failure causes the OR to fail)
- \( k \) of \( N \) gates become \( (k-1) \) of \( N-1 \) gates

\[
F_S(t) = F_{S|C\text{ Fail}}(t)F_C(t) + F_{S|C\text{ Up}}(t)(1 - F_C(t))
\]

Where \( S|C \text{ Fail} \) is the system given that \( C \) has failed and \( S|C \text{ Up} \) is the system given that \( C \) has not failed.
Reliability/Availability Point Estimates

• Frequently, the desired measure of a reliability model is the reliability at some time $t$. Thus, the distribution of the system reliability is superfluous; $R(t)$ is the only thing of interest.

• This condition simplifies computation because all that is necessary for solution is the reliability of the components at time $t$. Solution then becomes a straightforward computation.

• If a system is described in terms of the availability of components at time $t$, then we may compute the system availability in the same way that reliability is computed. The restriction is that all component behaviors must be independent of one another.
Reliability/Availability Tables

A system comprises $N$ components. Reliability of component $i$ at time $t$ is given by $R_{Xi}(t)$, and the availability of component $i$ at time $t$ is given by $A_{Xi}(t)$.

<table>
<thead>
<tr>
<th>Condition</th>
<th>System Reliability</th>
<th>System Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>system fails if all components fail</td>
<td>$R_S(t) = 1 - \prod_{i=1}^{n} (1 - R_{Xi}(t))$</td>
<td>$A_S(t) = 1 - \prod_{i=1}^{n} (1 - A_{Xi}(t))$</td>
</tr>
<tr>
<td>system fails if one component fails</td>
<td>$R_S(t) = \prod_{i=1}^{n} R_{Xi}(t)$</td>
<td>$A_S(t) = \prod_{i=1}^{n} A_{Xi}(t)$</td>
</tr>
<tr>
<td>system fails if at least $k$ components fail, identical distribution</td>
<td>$R_S(t) = \sum_{i=k}^{N} \binom{N}{i} (1 - R_{Xi}(t))^i R_X(t)^{N-i}$</td>
<td>$A_S(t) = \sum_{i=k}^{N} \binom{N}{i} (1 - A_{Xi}(t))^i A_X(t)^{N-i}$</td>
</tr>
<tr>
<td>system fails if at least $k$ components fail, general case</td>
<td>$R_S(t) = \sum_{g \in G_k} \left( \prod_{X \in g} (1 - R_X(t)) \right) \left( \prod_{X \in g} R_X(t) \right)$</td>
<td>$A_S(t) = \sum_{g \in G_k} \left( \prod_{X \in g} (1 - A_X(t)) \right) \left( \prod_{X \in g} A_X(t) \right)$</td>
</tr>
</tbody>
</table>
Estimating Component Reliability

• For hardware, MIL-HDBK-217-2 is widely used.
  – Not always current with modern components.
  – Lacks distributions; it only contains failure rates.
  – While not perfect, it seems to be the best source that exists. However, numbers from MIL-HDBK-217-2 should be used with caution.

• Due to the nature of software, no accepted mechanism exists to predict software reliability before the software is built.
  – Best guess is the reliability of previously built similar software.

• In all cases, numbers should be used with caution and adjusted based on observation and experience.

• No substitute for empirical observation and experience!
Modeling Process

- Reliability models are built only after proper service is specified.

- Reliability models are built to answer the question “What subsystem or components must be proper for the system to be proper?”

- Build models hierarchically out of subsystems.

- Estimation and guesses are acceptable, but state them explicitly.

- If unsure, do sensitivity analysis to see how much it matters.
Reliability Modeling Process

- Realistic systems result in large RBDs and must be managed hierarchically.

RBD Process(system)
Define the system
Define “proper service”
Create RBD out of components
for each component
  if component is simple
    obtain reliability data of component
  else
    Do RBD Process(component)
end if
Compute reliability of system
Do results meet specification?
Modify design and repeat as necessary
Review

- Reliability: review of definition
- Failure rate
- System reliability
  - Independent failure assumption
  - Minimum, maximum, $k$ of $N$
  - Reliability block diagrams, fault trees, reliability graphs
- Reliability modeling process