$$P_f = \int_0^t pdf(x)dx$$

Current Time:

10:08 AM

When Markov Analysis ?

Vito Faraci Jr.

Personal Info

$$P_f = \int_0^t pdf(x)dx$$

- Vito Faraci Jr.
- Mathematician by education, engineer by trade.
- Worked for small consulting company on L.I. for past 16 years presenting seminars to FAA on Probability, Reliability, FTA, FMEA, & Markov Analysis (MA).
- Have given MA lectures to Lockheed Martin System Safety Groups, Sandia National Labs, and to reliability engineers at Bombardiere Aircraft Co.
- Have several papers published by RIAC and System Safety Society on Reliability and MA.

Agenda

$$P_f = \int_0^t pdf(x)dx$$

Subject	Sheet #	Time (approx)
▪ Introduction	5	2 min
▪ Failure Rate vs. Failure Logic	8	2 min
▪ Combinatorial vs. Non-combinatorial Logic	10	3 min
▪ FTA vs. MA Advantages / Disadvantages	14	5 min
▪ Answer to "Why Markov?"	19	1 min
▪ Answer to "When Markov?"	20	2 min
▪ Time Failure Dependency	21	8 min
▪ More about System Reliability	32	2 min
▪ Markov Analysis basics	34	5 min
▪ State Logic Compared with And/Or Logic	40	5 min
▪ Standby	47	5 min
▪ Sequence Failure	52	2 min
▪ State-Dependent Behavior	55	2 min
▪ Reconfiguration	61	3 min
▪ Pf Approximations	65	1 min
▪ Summary	69	2 min
▪ Questions	72	10 min

Acronyms

$$P_f = \int_0^t pdf(x)dx$$

- ARP – Aerospace Recommended Practice
- Combo – Combinatorial Logic
- DE – Differential Equation
- FAA – Federal Aviation Administration
- FR – Failure Rate
- FTA – Fault Tree Analysis
- MA – Markov Analysis
- Non-combo – Non-combinatorial Logic
- P_f – Probability of Failure
- RBD – Reliability Block Diagram
- ROF – Required Order Factor
- SDE – Simultaneous Differential Equations
- SSD – State Sequence Diagram

Introduction (Background Info)

$$P_f = \int_0^t pdf(x)dx$$

Fault Tree Analysis (FTA) was introduced in 1962 at Bell Labs, and is the most commonly used tool for calculating System Reliability and qualitative and quantitative risk analyses.

For a period of time its limitations were unquestioned but were known only to a few. Starting in the early 80s, a group of NASA mathematicians performed some impressive studies that clearly exposed a very subtle limitation. In an effort to overcome this limitation, NASA developed several algorithms, and described in detail an approach using Markov Analysis (MA), designed not to replace, but to support FTAs.

With respect to System Reliability , the integration of MA with FTA has been a giant step forward. Engineers can now solve more accurately a much larger set of “Risk” problems than they could before.

Introduction cont.

$$P_f = \int_0^t pdf(x)dx$$

MA was introduced in 1907 by a Russian mathematician by the name of A. A. Markov. It is interesting to note that although this knowledge has been around for some time, the engineering community had waited until the 1980s to taken advantage of this science. For example,

a) NASA has been employing Markov methods for Probabilistic Risk Assessments (PRA) for the Shuttle systems, and
b) FTA and Reliability Software manufacturers have integrated Markov techniques into their Risk Assessment SW Programs.

It is this author's opinion, that due to a lack of documentation written in a clear common language, knowledge of Markov as applied to Reliability still remains a little "sketchy" within the engineering community.

Introduction cont.

$$P_f = \int_0^t pdf(x)dx$$

Objective:

- Explain ***when*** and ***why*** Markov methodology should be used
- Emphasis on
  - ***qualitative*** aspect (the logic) rather than
  - quantitative (mathematical definitions, axioms, theorems, and equations) which is a subject on its own.
- To fully understand the ***when*** and ***why*** of Markov, one must first have a good understanding of:

a) the difference between *failure rate* and *failure logic*, and

b) the difference between *combinatorial* and *non-combinatorial* logic which crops up so very often in the world of Probability

Failure Rate vs. Failure Logic

The study of Markov with respect to Reliability, requires that the difference between failure rate and failure logic be absolutely clear. The next slide will define and help clarify the distinction.

Failure Rate vs. Failure Logic cont.

$$P_f = \int_0^t pdf(x)dx$$

Failure Rate

A failure characteristic of an individual component or system. It is a measure of the number of failures that occur within a unit interval of time. A failure rate can be constant (does not vary with time), or non-constant (varies with time).

Failure Logic

Deals with the relationship of failures of 2 or more devices that occur in a system. Failure logic can be combinatorial or non-combinatorial.

Examples of Failure Logic

What is the probability that components A and B both fail?
What is the probability that A fails and B does not?
What is the probability that A and B both fail, and A fails before B?

Combinatorial vs. Non-combinatorial Logic

The study of Markov with respect to Reliability, also requires an understanding of combinatorial and non-combinatorial logic which will be explained on the next few slides.

Combinatorial vs. Non-combinatorial Logic cont.

$$P_f = \int_0^t pdf(x)dx$$

Combinatorial Logic

Two or more input states define one or more output states. Output states are related by defined rules that are independent of previous states.

- Logic depends solely on combinations of inputs
- Time is neither modeled or recognized
- Outputs change when inputs change irrespective of time
- Output is a function of, and only of, the present input

Simply stated: (Can be taken as a definition)

Combinatorial logic is any logic that can be expressed using And Gates & Or Gates (Boolean Algebra)

Combinatorial vs. Non-combinatorial Logic cont. $P_f = \int_0^t pdf(x)dx$

Non-combinatorial Logic

Logic of output(s) depends on combinations of present input states, and combinations of previous input states.

Excerpt from Wikipedia

In digital circuit theory, sequential logic is a type of logic circuit whose output depends not only on the present input but also on the history of the input. This is in contrast to combinational logic, whose output is a function of, and only of, the present input. In other words, sequential logic has state (memory) while combinational logic does not.

Simply stated: (Can be taken as a definition)

Non-combinatorial logic is any logic that cannot be expressed using And Gates & Or Gates (Boolean Algebra)

Quick Answer to "Why Markov?"

In Reliability the role of Markov is to obtain *more accurate System Reliability calculations* in places where FTA has difficulty.

However, there are some Markov *disadvantages* that were cited by NASA and other sources that will be presented in the next few slides.

FTA vs. MA Advantages / Disadvantages

FTA vs. MA **Advantages / Disadvantages** cont.

$$P_f = \int_0^t pdf(x)dx$$

Fault Tree Advantages:

- Acts as a visual tool which can be used to pinpoint system weaknesses.
- Exhibits clear representation of logical processes that lead to a system or sub-system failure (clear qualitative representation of failure propagation).
- Reveals relatively simple equations for P_f calculations yielding quantitative analyses that do not require high powered math.
- Proves to be a very effective tool for the fault isolation process.

FTA vs. MA Advantages / Disadvantages cont.

$$P_f = \int_0^t pdf(x)dx$$

Fault Tree Limitations: (From ARP 4761 Issue 1996-12)

- Difficult to allow for transient & intermittent faults or standby systems with spares.
- If a system has many failure conditions, separate fault trees may need to be constructed for each one.
- Difficult to represent systems where failure rates or repair rates are state dependent (change between states).

FTA vs. MA Advantages / Disadvantages cont.

$$P_f = \int_0^t pdf(x)dx$$

Excerpt from NASA Ref. Publication 1348:

Traditionally, the reliability analysis of a complex system has been accomplished with combinatorial mathematics. The standard fault-tree method of reliability analysis is based on such mathematics. Unfortunately, the fault-tree approach is somewhat limited and incapable of analyzing systems in which reconfiguration is possible. Basically, a fault tree can be used to model a system with:

Only permanent faults (no transient or intermittent faults)

- No time failure dependencies
- No sequence failure dependencies
- No state-dependent behavior
- No reconfiguration

FTA vs. MA Advantages / Disadvantages cont.

$$P_f = \int_0^t pdf(x)dx$$

From ARP4761 Issue 1996-12

- MA does not have these limitations.
- Sequence dependent events are included and handled naturally.
- Covers a much wider range of system behaviors.

FTA vs. MA Summary
Answer to “Why Markov?”

$$P_f = \int_0^t pdf(x)dx$$

This following summary answers the “Why Markov?” question with a little more detail

- Fault Tree Analysis (FTA) –Handles combinatorial type problems both qualitatively and quantitatively extremely well. <u>However</u> FTA has difficulty with non-combinatorial problems in both areas.

- Markov Analysis (MA) – Handles non-combinatorial as well as combinatorial problems. <u>However</u>, not quite as intuitive as FTA, and requires higher power mathematics for the quantitative analyses.

Answer to "When Markov ?"

$$P_f = \int_0^t pdf(x)dx$$

Close examination of the above NASA 1348 and FAA excerpts reveals that "Markov" techniques should be used when non-combinatorial logic is encountered. Most commonly for the following 4 situations which will be further discussed in this presentation.

- Standby systems with spares
- Sequence failure dependencies
- State-dependent behavior
- Reconfiguration

Note:
NASA 1348 also stated that FTA has difficulty with *time failure dependencies*, but it doesn't appear on the above list.

Time Failure Dependency

Explanation of why it doesn't appear on the Markov list.

Time Failure Dependency
NASA 1348 Clarification

$$P_f = \int_0^t pdf(x)dx$$

NASA 1348 also stated that FTA has difficulties handling time failure dependencies. However, this problem has nothing to do with failure logic. This problem is simply due to the fact that most FTA algorithms are not written to handle the more complicated math required for devices that have time failure dependencies i.e. non-constant failure rate devices.

Primary component failure rates, whether constant or non-constant, have no effect on logic of failure or Markov requirements.

For purposes of clarification, *time failure dependency* will be discussed in more detail in the next few slides.

Time Failure Dependency cont.

$$P_f = \int_0^t pdf(x)dx$$

Failure characteristics of any primary component will fall into one of two categories, either <u>constant</u> failure rate or <u>non-constant</u> failure rate. Components whose failure rates vary with time are said to have a "Time Failure Dependency" .

Time Failure Dependency cont.

Constant Failure Rate Characteristic

$$P_f = \int_0^t pdf(x)dx$$

Failure characteristic of a <u>constant</u> failure rate device

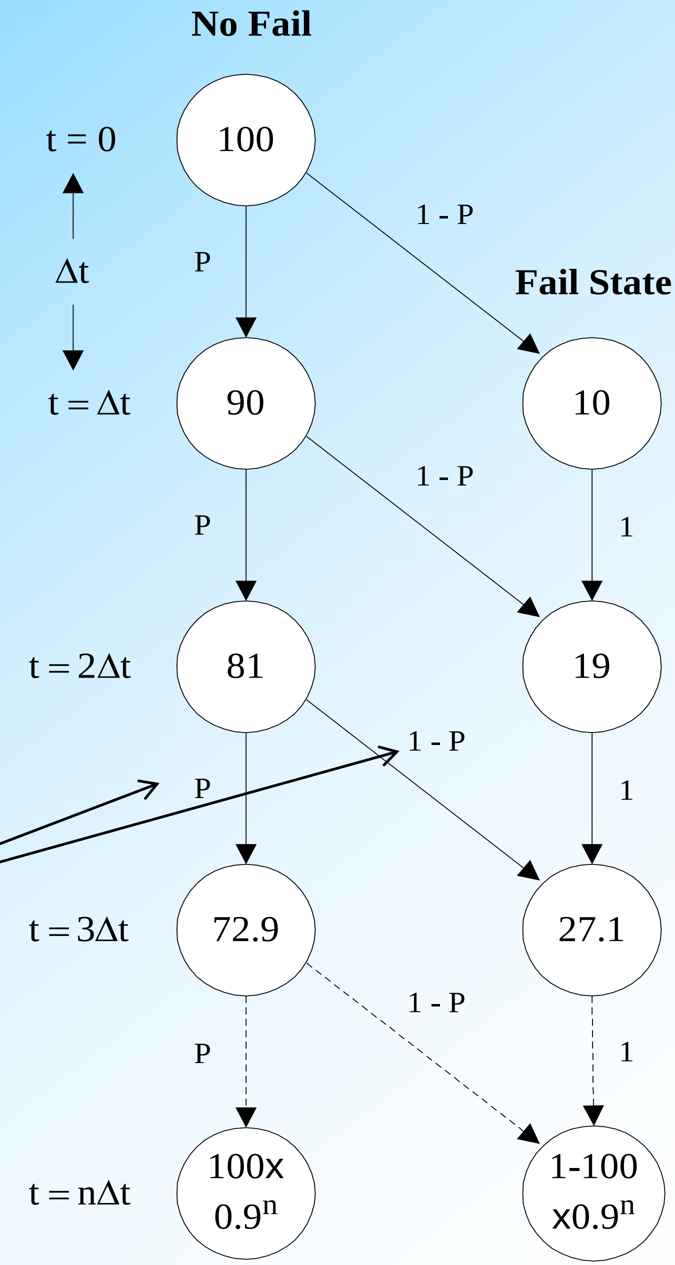
probability of success = $P = e^{-\lambda t} = 0.9$

probability of fail = $F = 1 - P$

where e = 2.71828,

λ = constant failure rate,

t = time.



Transition Probabilities

Note:
Percentage of failures are always the same (constant) for each time interval Δt.

Time Failure Dependency cont.

Non-Constant Failure Rate

$$P_f = \int_0^t pdf(x)\,dx$$

The following is an example of a probability of failure (P_f) equation of a non-constant failure rate device. In this particular case, the device exhibits a "normal" distribution of failure:

$$P_f = \frac{1}{s\sqrt{2\pi}} \int_0^t e^{-\left(\frac{(x-u+hl)^2}{2s^2}\right)} dx$$

where

u = mean time to failure,

s = standard deviation,

hl = hours previously logged,

t = time.

x = dummy variable

Note:

The above equation is "non-integrable" which presents an additional computational challenge when calculating P_f.

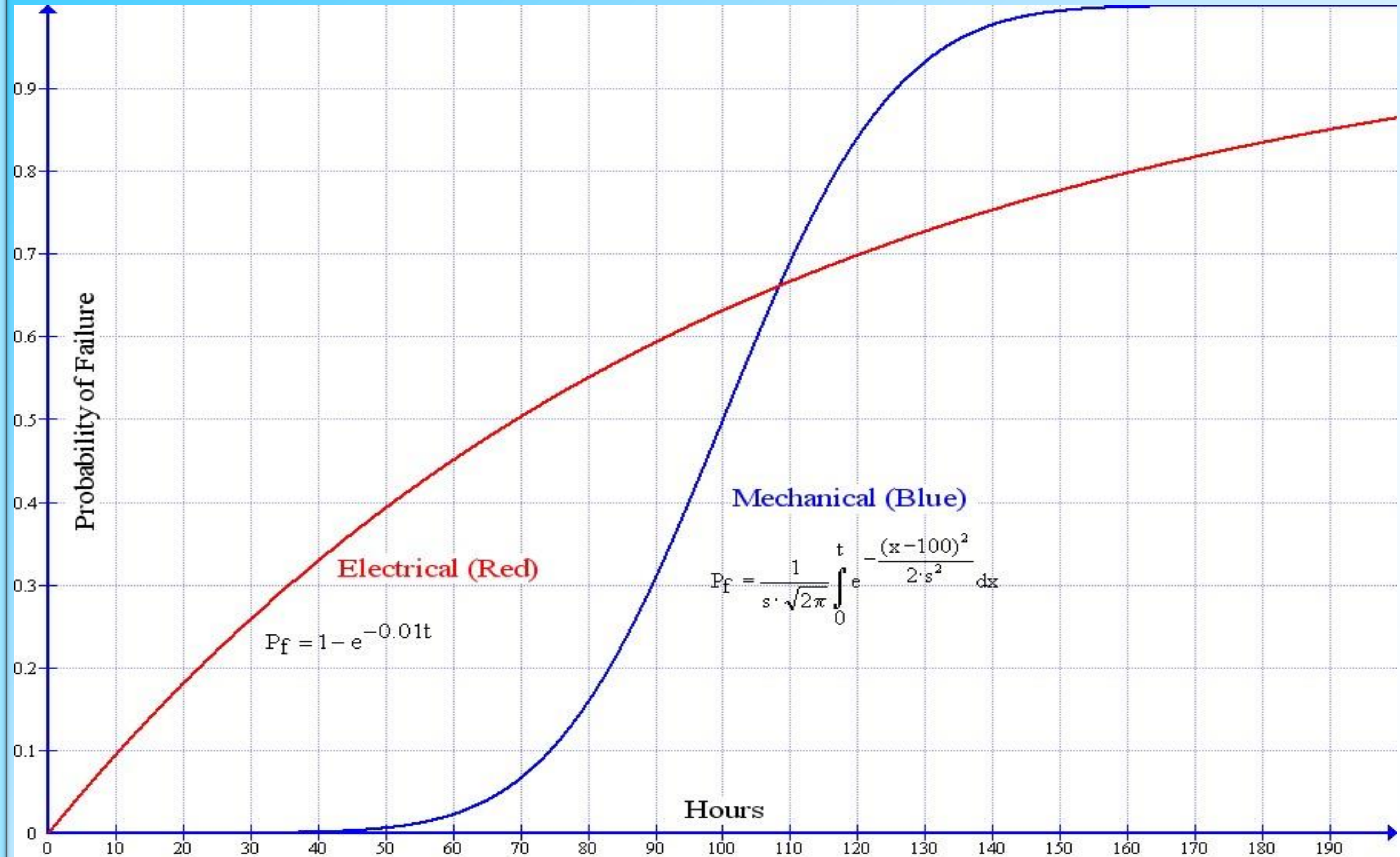
Time Failure Dependency cont.

Constant FR vs. Non-Constant FR

$$P_f = \int_0^t pdf(x)\,dx$$

Comparison of a Constant and a Non-constant Failure Rate Device with same MTBF

Probability of failure vs. Time graph



Time Failure Dependency

Example of Mixed Failure Rate System

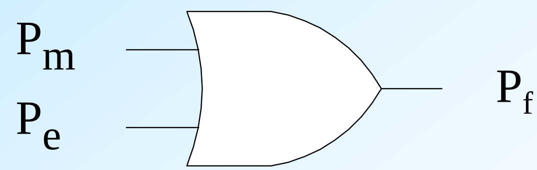
$$P_f = \int_0^t pdf(x)\,dx$$

System Reliability is defined to be the "*probability that a system will perform its specified function successfully within a specified period of time*."

Consider the following system comprised of a mechanical component in series with an electrical component.

RBD

Logic Diagram



P_m (Mechanical)

$$P_m(t) = \frac{1}{s\cdot\sqrt{2\pi}}\int_0^t e^{-\frac{(x-100)^2}{2\cdot s^2}}\,dx$$

P_e (Electrical)

$$P_e(t) = 1 - e^{-0.01t}$$

$$P_f(\text{System}) = P_m(t) + P_e(t) - P_m(t)\cdot P_e(t)$$

Time Failure Dependency

Example of Mixed Failure Rate System cont.

$$P_f = \int_0^t pdf(x)dx$$

Major point of previous slide:

$$P_f(\text{System}) = P_m(t) + P_e(t) - P_m(t) \cdot P_e(t) \quad \text{(Boolean Expression)}$$

When integrating devices with mixed failure rates in a combinatorial situation, all that is required is basic Boolean Algebra or what is called "and/or" logic. There is no need for Markov or any other special techniques.

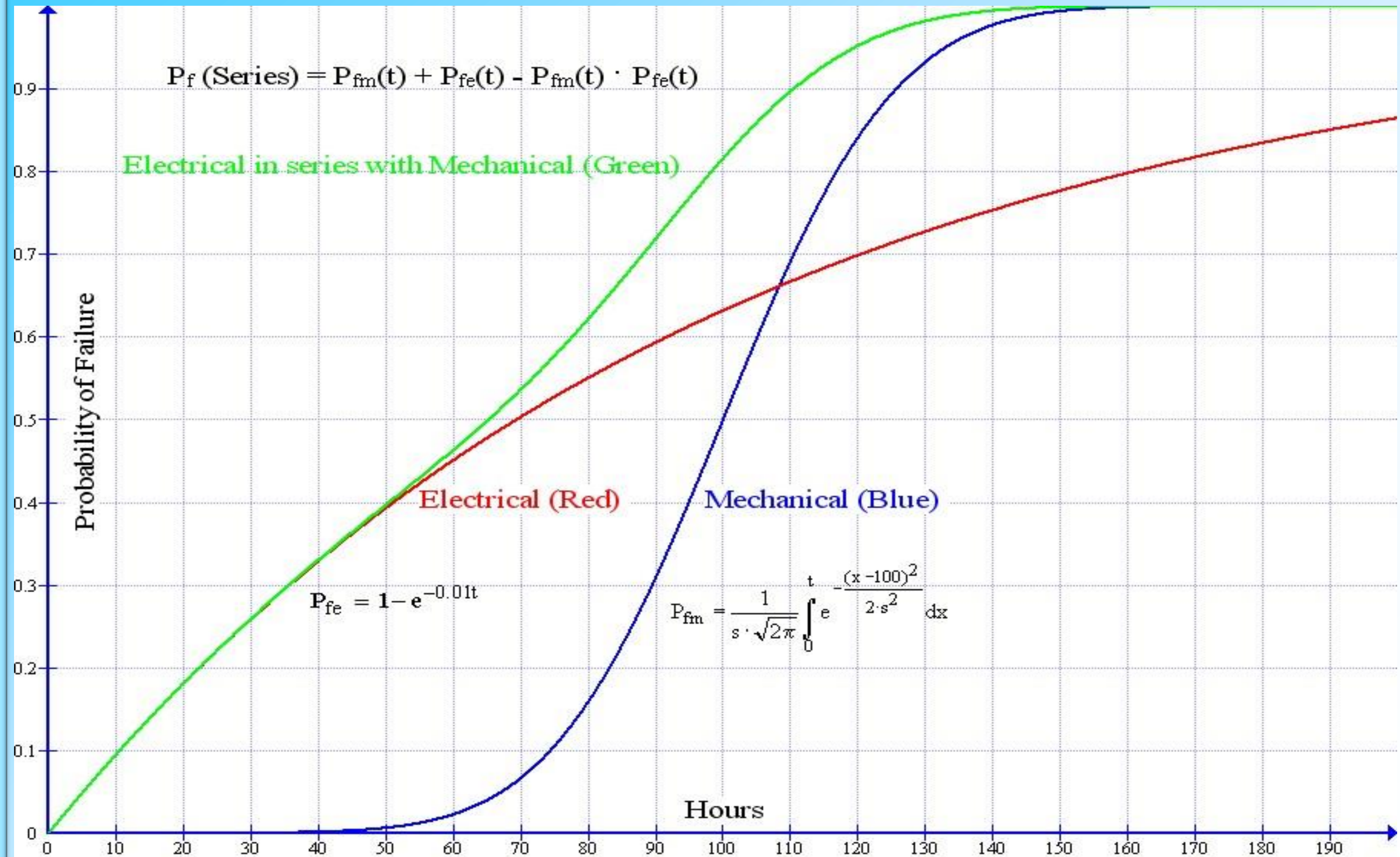
Time Failure Dependency cont.

Example of Mixed Failure Rates & Combinatorial

$$P_f = \int_0^t pdf(x)dx$$

Constant & Non-constant FR Devices in Series

Probability of failure vs. Time graph



Time Failure Dependency cont.

State Sequence Diagram (SSD)

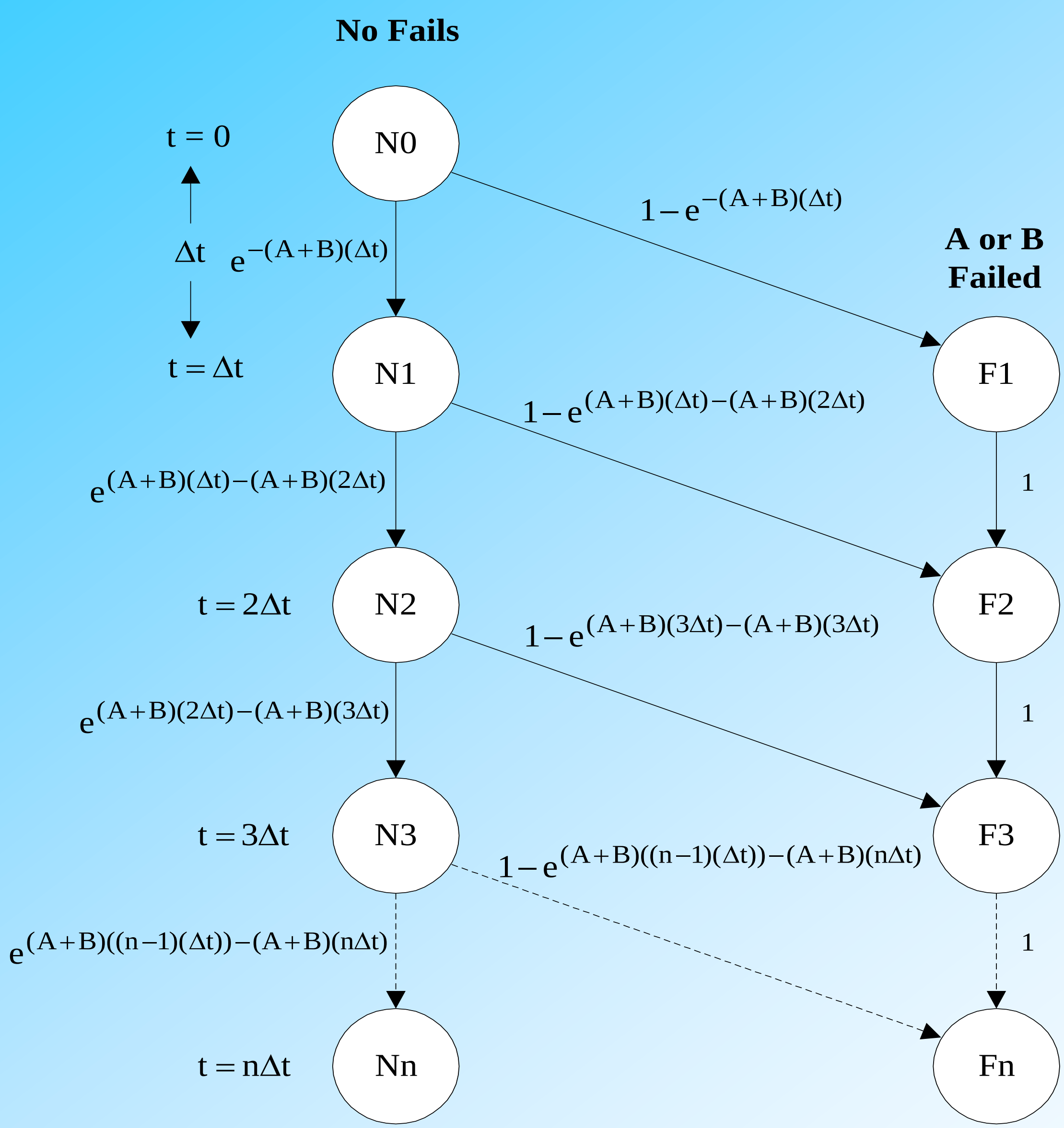
$$P_f = \int_0^t pdf(x)dx$$

A deeper understanding of the qualitative aspects (the logic) for solving System Reliability problems can be obtained with a useful tool known as a State Sequence Diagram (SSD) as illustrated next.

State Sequence Logic for 2 Devices in Series

State Sequence Diagram (SSD)

$$P_f = \int_0^t pdf(x)dx$$



N_i = probability of success at $t = i\Delta t$

F_i = probability of failure at $t = i\Delta t$

A(t) = Failure Rate of Component A

B(t) = Failure Rate of Component B

Expressions = transition probabilities

More about System Reliability

$$P_f = \int_0^t pdf(x)\,dx$$

System Reliability concentrates on math modeling the failure characteristics of a system. This is accomplished by deriving failure characteristics (math models) of each individual component, and then ***integrating these individual component failure characteristics*** based on component relationships operating within the system. This science relies very heavily on applied mathematics, and recall that it has its qualitative and quantitative aspects.

The rest of this presentation is devoted to the study of this subject.

More about System Reliability cont.

$$P_f = \int_0^t pdf(x)dx$$

Failure Logic as well as Failure Rates must be considered:

When calculating System Reliability, not only must failure rates of primary components be considered, but also the system's failure logic.

Example: (Consider a simple system comprised of 2 components A and B.)

What is the probability of both A and B failing <u>and</u> A failing before B?

Important note:

The failure rate of primary a component can be constant or non-constant. This variable by itself has no effect on a system's failure logic, and does not determine requirements for Markov.

Markov Analysis Basics

A quick overview of some Markov Basics should help with further understanding of the logic involved.

Markov Analysis Basics

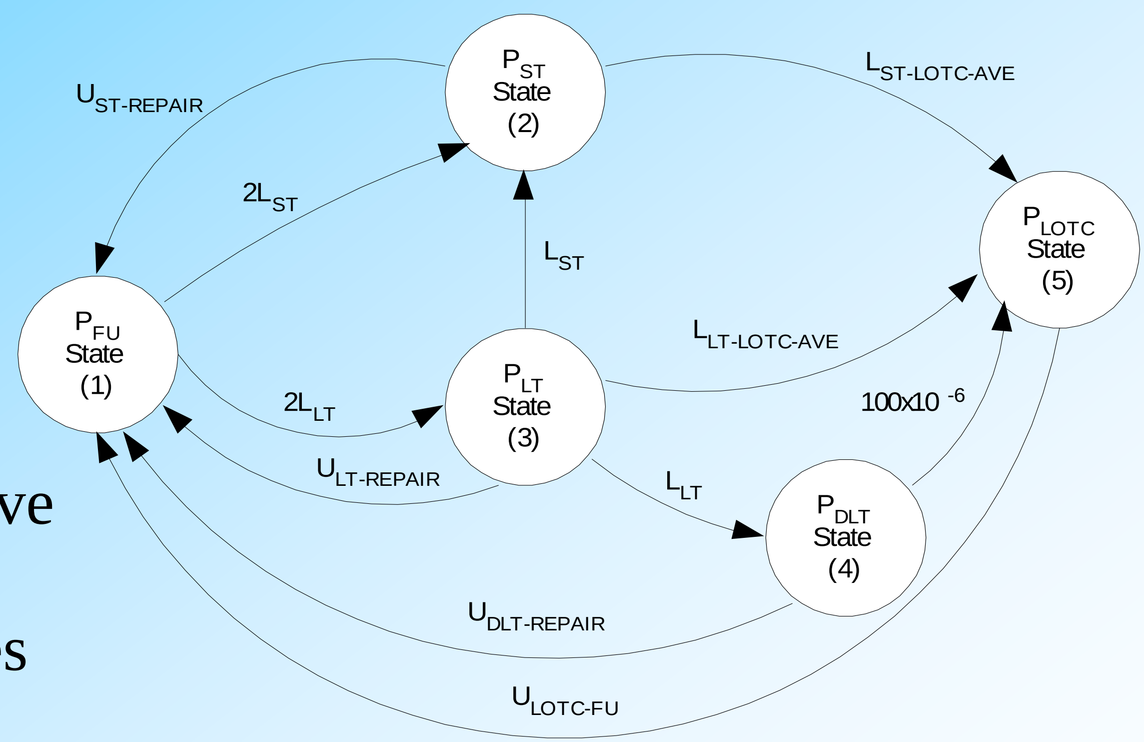
$$P_f = \int_0^t pdf(x)dx$$

If a system or component can be in one of two states (i.e. failed, non-failed), and if we can define the probabilities associated with these states on a discrete or continuous basis, the probability of being in one or other at a future time can be evaluated using a state-time analysis. In reliability and availability analysis, failure probability and the probability of being returned to an available state are the variables of interest. The best known state-space technique is Markov Analysis.

Markov Analysis Basics cont.

$$P_f = \int_0^t pdf(x)dx$$

- State Diagram represents various system states
- Transition rate is function of failure or repair rate
- States must be finite in number
- States are mutually exclusive
- The sum of the probabilities must equal 1



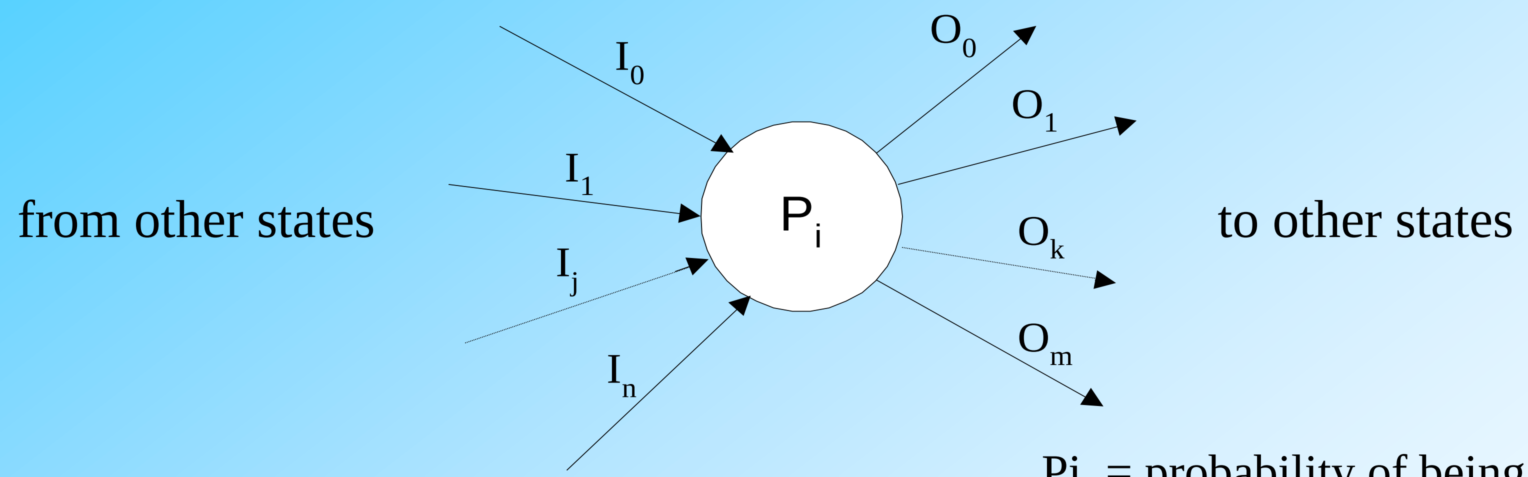
Mutually Exclusive – System can never be in any more than one state at any given time

Markov Analysis Basics

Few words on the Quantitative Aspect

$$P_f = \int_0^t pdf(x)dx$$

The following is a typical state taken from a State Diagram with n input transitions with failure rates I_j, and m output transitions constant failure rates O_k.



Pi = probability of being in state I

Unfortunately Pi cannot be calculated immediately. Precise calculations of Pi can at times be very difficult and may require advanced topics from Calculus and even Advanced Calculus. Topics such as solutions of sets of Simultaneous Differential Equations (SDE), Matrix Algebra, and/or Convolution Integrals which are all subjects by themselves.

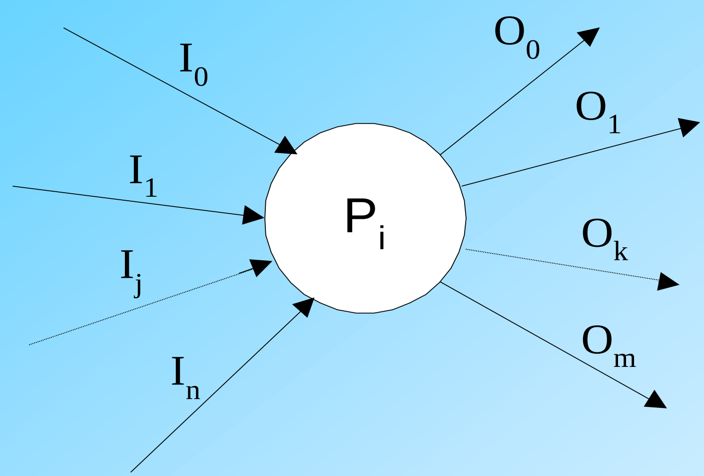
However, the following is a quick and simple illustration of one such quantitative approach using SDEs.

Markov Analysis Basics

Few words on the Quantitative Aspect cont.

$$P_f = \int_0^t pdf(x)dx$$

from other states



to other states

Each state has a corresponding DE. The DE corresponding to a typical state is:

$$\frac{dP(i)}{dt} = \sum_{j=1}^{n} I_j P(j) - \left(\sum_{k=1}^{m} O_k \right) P(i)$$

Note: Transitions <u>into</u> a state result in <u>positive</u> terms in the DE, while transitions <u>leaving</u> a state yield <u>negative</u> terms.

For the sake of simplifying notation let $P_i = P(i)$.

Markov Analysis Basics cont.

Few words on the Quantitative Aspect cont.

$$P_f = \int_0^t pdf(x)dx$$

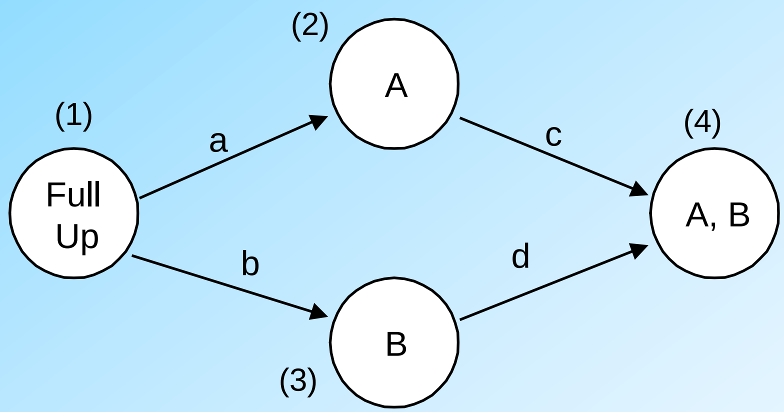
Example : Two devices in Parallel (constant failure rates assumed)

Transitions **into** a state result in **positive** terms in the DE, while transitions **leaving** a state yield **negative** terms.

$$\frac{dP_2}{dt} = aP_1 - cP_2$$

P_i = Probability of being in State i

$$\frac{dP_1}{dt} = -(a+b)P_1$$



$$\frac{dP_4}{dt} = cP_2 + dP_3$$

$$\frac{dP_3}{dt} = bP_1 - dP_3$$

Caution: This method is not reliable for non-constant failure devices.

State Logic Compared with And/Or Logic

Recall non-combinatorial logic cannot be expressed using Logic Gates and therefore State Diagrams are used instead to capture the logic.

State Logic Compared with And/Or Logic

Combinatorial Type Problems

$$P_f = \int_0^t pdf(x)dx$$

As mentioned before Markov handles combinatorial as well as non-combinatorial problems.

Although there is no need of Markov for solving combinatorial type problems, (FTA handles them well enough) the next few slides will demonstrate several examples for the sake of illustration and comparison.

Note:

The following comparison examples are limited to "constant failure rate" type problems. Quantitative solutions to "non-constant failure rate" type problems may require other math methods.

State Logic Compared with And/Or Logic

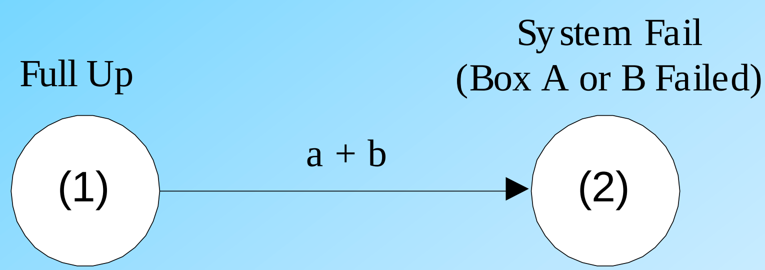
2 Components in Series (Combo Type)

$$P_f = \int_0^t pdf(x)\,dx$$

Two black boxes start operation at the same time. Box 1 has failure rate a and Box 2 has failure rate b. Successful system operation requires that both boxes be working.

Find P_f = Probability of System Failure.

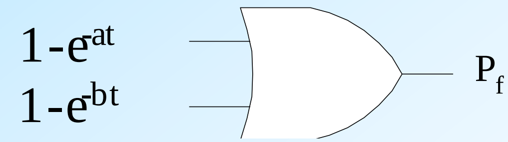
State Diagram



$$dP(2)/dt = (a+b)P(1) \Rightarrow$$

$$P_f = P(2) = 1 - e^{-(a+b)t}$$

And/Or Logic



$$P_f = P_a + P_b - P_a \cdot P_b \Rightarrow$$

$$P_f = 1 - e^{-(a+b)t}$$

Note: P(n) = Probability of State (n)

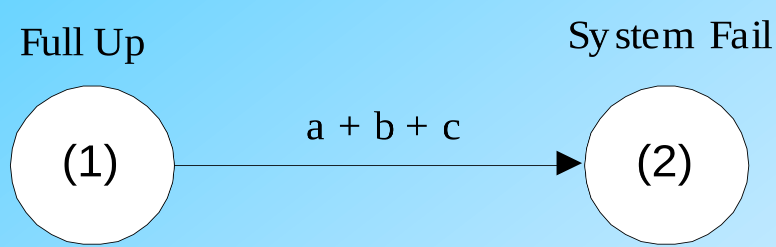
State Logic Compared with And/Or Logic

3 Components in Series (Combo Type)

$$P_f = \int_0^t pdf(x)dx$$

Three black boxes start operation at the same time. Boxes A, B, and C have failure rates a, b, and c respectively. Successful system operation requires that all three boxes be working. Find P_f the Probability of System Failure.

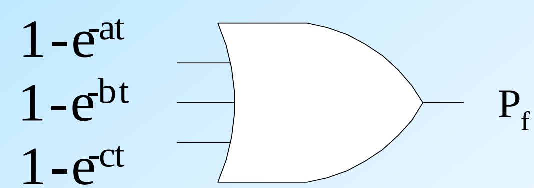
State Diagram



$$\frac{dP(2)}{dt} = (a+b+c)\cdot P(1)$$

$$\Rightarrow P_f = P(2) = 1 - e^{-(a+b+c)t}$$

And/Or Logic



$$P_f = P_a + P_b + P_c - P_a \cdot P_b - P_a \cdot P_c - P_b \cdot P_c + P_a \cdot P_b \cdot P_c$$

$$\Rightarrow P_f = 1 - e^{-(a+b+c)t}$$

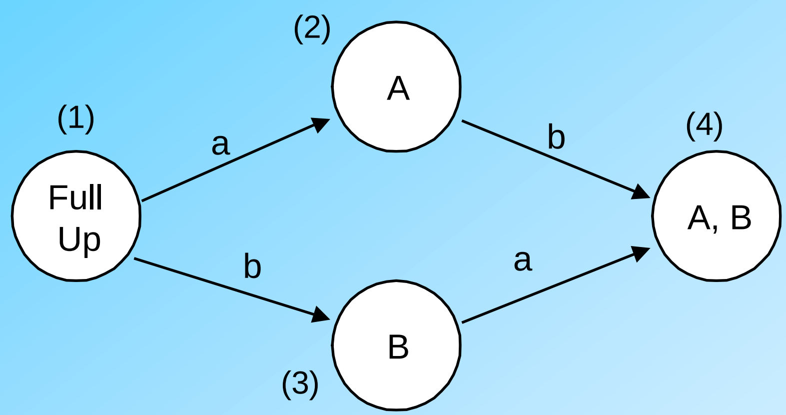
State Logic Compared with And/Or Logic

2 Components Active Redundant (Parallel) (Combo Type)

$$P_f = \int_0^t pdf(x)dx$$

Two black boxes start operation at the same time. Box A has failure rate a and Box B has failure rate b. Successful system operation requires that Box A or Box B or both be working. Find P_f the Probability of System Failure.

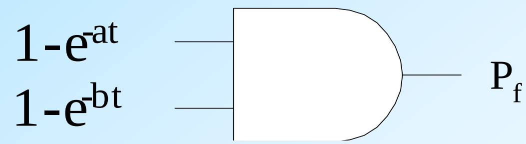
State Diagram



$$dP(4)/dt = bP(2) + aP(3)$$

$$\Rightarrow P_f = P(4) = (1 - e^{-at})(1 - e^{-bt})$$

And/Or Logic



$$P_f = (1 - e^{-at})(1 - e^{-bt})$$

Note:
Failure rates are not effected by state changes.

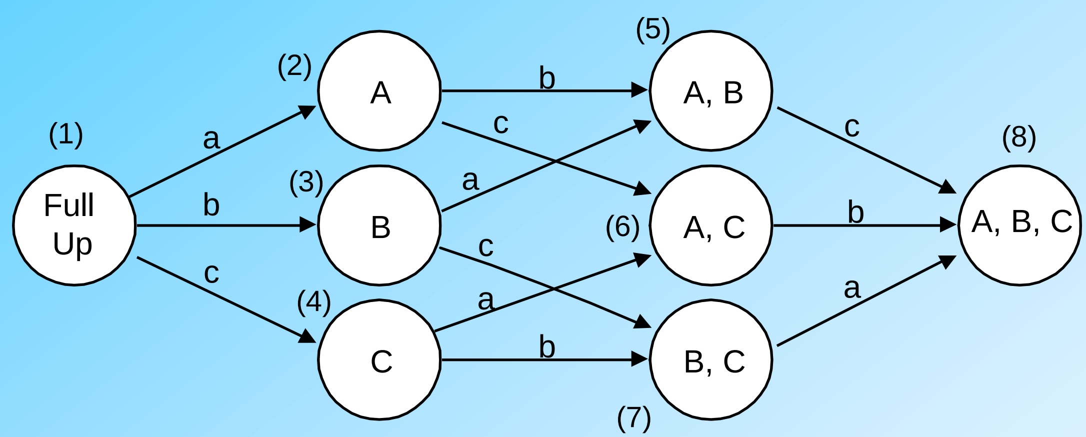
State Logic Compared with And/Or Logic

3 Components Active Redundant (Parallel) (Combo Type)

$$P_f = \int_0^t pdf(x)dx$$

Three black boxes start operation at the same time. Box A, B, and C have failure rate a, b, and c respectively. Successful system operation requires that Box A, B, or C be working. Find P_f the Probability of System Failure.

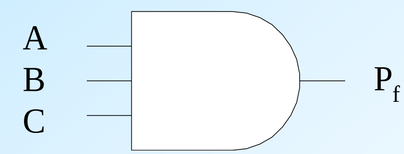
State Diagram



$$dP(8)/dt = cP(5) + bP(6) + aP(7)$$

$$\Rightarrow P_f = P(8) = (1-e^{-at})(1-e^{-bt})\,(1-e^{-ct})$$

And/Or Logic



$$A = (1-e^{-at})$$

$$B = (1-e^{-bt})$$

$$C = (1-e^{-ct})$$

$$P_f = (1-e^{-at})(1-e^{-bt})\,(1-e^{-ct})$$

When Markov?

$$P_f = \int_0^t pdf(x)dx$$

When Markov? For the following situations:

- **Standby systems with spares**
- **Sequence failure dependencies**
- **State-dependent behavior**
- **Reconfiguration**

Standby

$$P_f = \int_0^t pdf(x)dx$$

When Markov? For the following situations:

- **Standby systems with spares**
- Sequence failure dependencies
- State-dependent behavior
- Reconfiguration

Standby cont.

$$P_f = \int_0^t pdf(x)dx$$

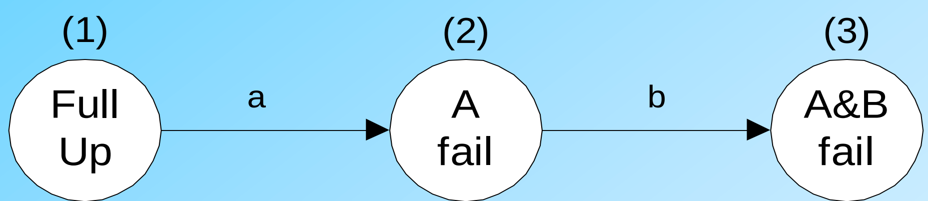
Solutions of non-combinatorial problems require different techniques other than traditional combinatorial logic such as that found in FTAs. In particular one of the simplest non-combinatorial type problem that has intrigued mathematicians is the classic “Standby Problem”.

Standby cont.

$$P_f = \int_0^t pdf(x)dx$$

Box A has failure rate a and Box B has failure rate b. Box A is turned on while Box B remains powered off in standby mode. Immediately upon detection of Box A failure, Box B is turned on. Calculate the probability that both boxes are failed. (Assume a perfect switch.)

State Diagram



And/Or Gates

?

$$\frac{dP_1}{dt} = -aP_1 \quad \Rightarrow \quad P_1 = e^{-at}$$

$$\frac{dP_2}{dt} = aP_1 - bP_2 \Rightarrow \quad P_2 = \frac{a}{a-b}\left(e^{-bt} - e^{-at}\right)$$

$$\frac{dP_3}{dt} = bP_3 \quad \Rightarrow \quad P_3 = 1 + \frac{b}{a-b}e^{-at} - \frac{a}{a-b}e^{-bt}$$

P_i = probability of State i

Standby cont.

$$P_f = \int_0^t pdf(x)dx$$

Deriving solutions (quantitative aspect) for **non-combinatorial** type problems, whether failure rates are constant or not, is somewhat non-trivial and is a subject for another paper. However again for the sake of a deeper understanding of the qualitative aspect, a State Sequence Diagram (SSD) may be helpful.

Standby cont.

State Sequence Diagram for "Standby"

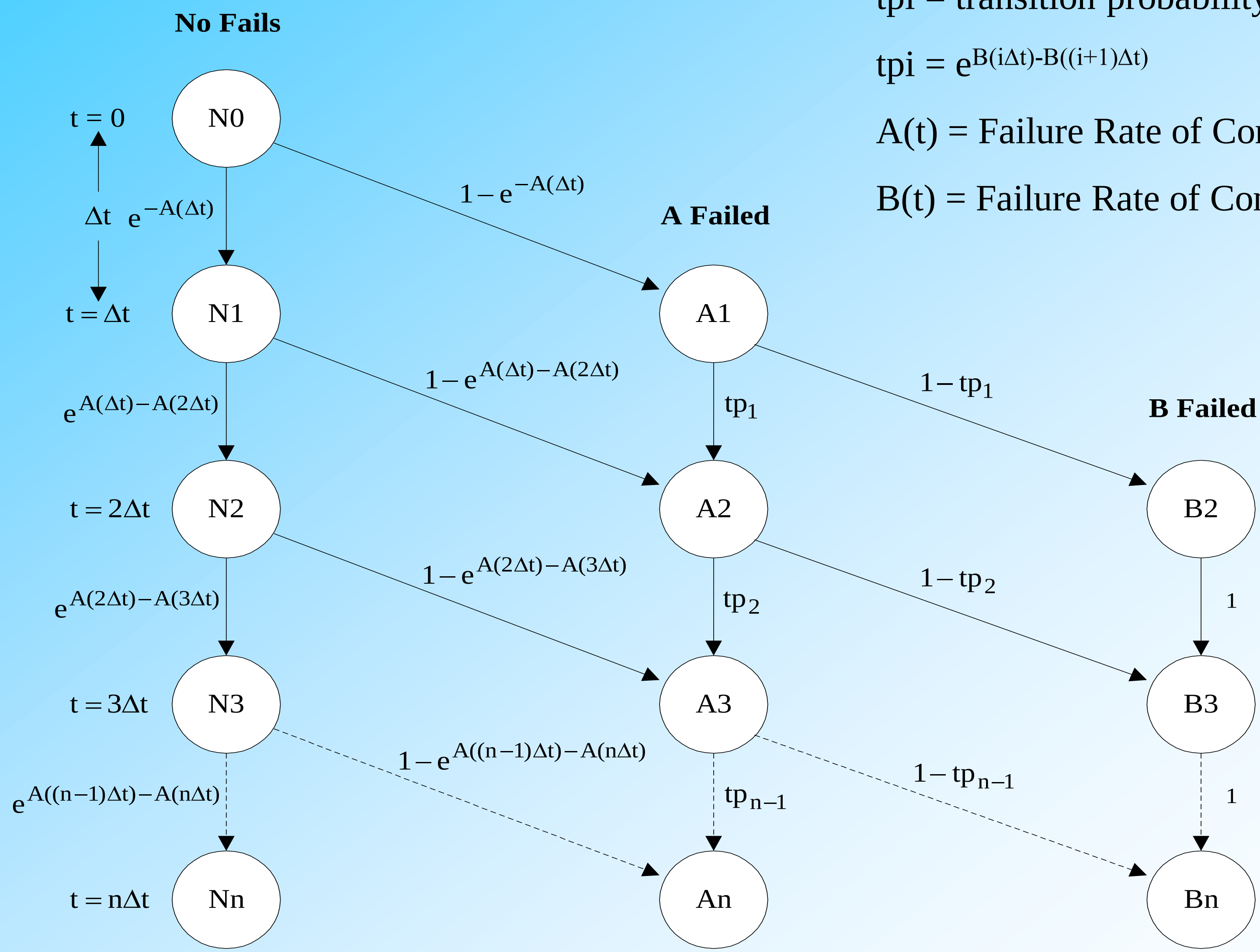
$$P_f = \int_0^t pdf(x)dx$$

tpi = transition probability

tpi = $e^{B(i\Delta t)-B((i+1)\Delta t)}$

A(t) = Failure Rate of Component A

B(t) = Failure Rate of Component B



Sequence Failure

$$P_f = \int_0^t pdf(x)dx$$

When Markov? For the following situations:

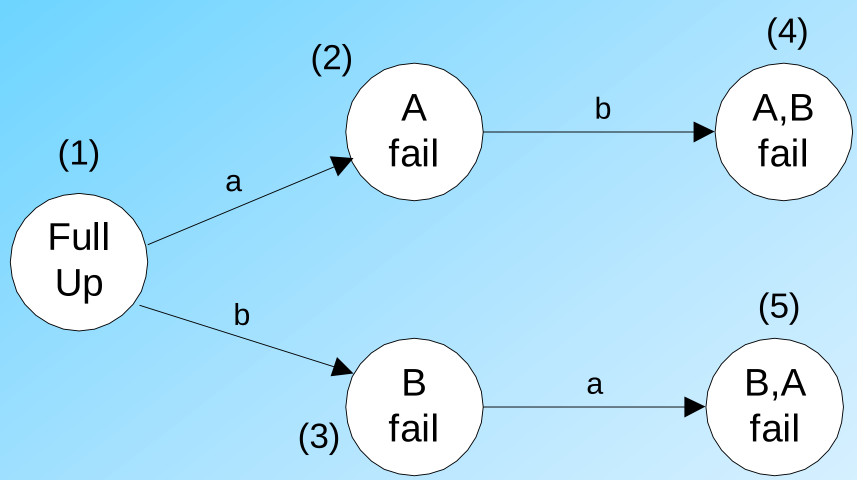
- Standby systems with spares
- **Sequence failure dependencies**
- State-dependent behavior
- Reconfiguration

Sequence Failure cont.

$$P_f = \int_0^t pdf(x)dx$$

Two components are in operation. Find the probability that both Boxes A and B fail <u>and</u> that Box A fails before Box B. Also find the probability that both Boxes fail <u>and</u> that Box B fails before Box A.

State Diagram



And/Or Gates

?

$$P(1) = e^{-at} \cdot e^{-bt}$$

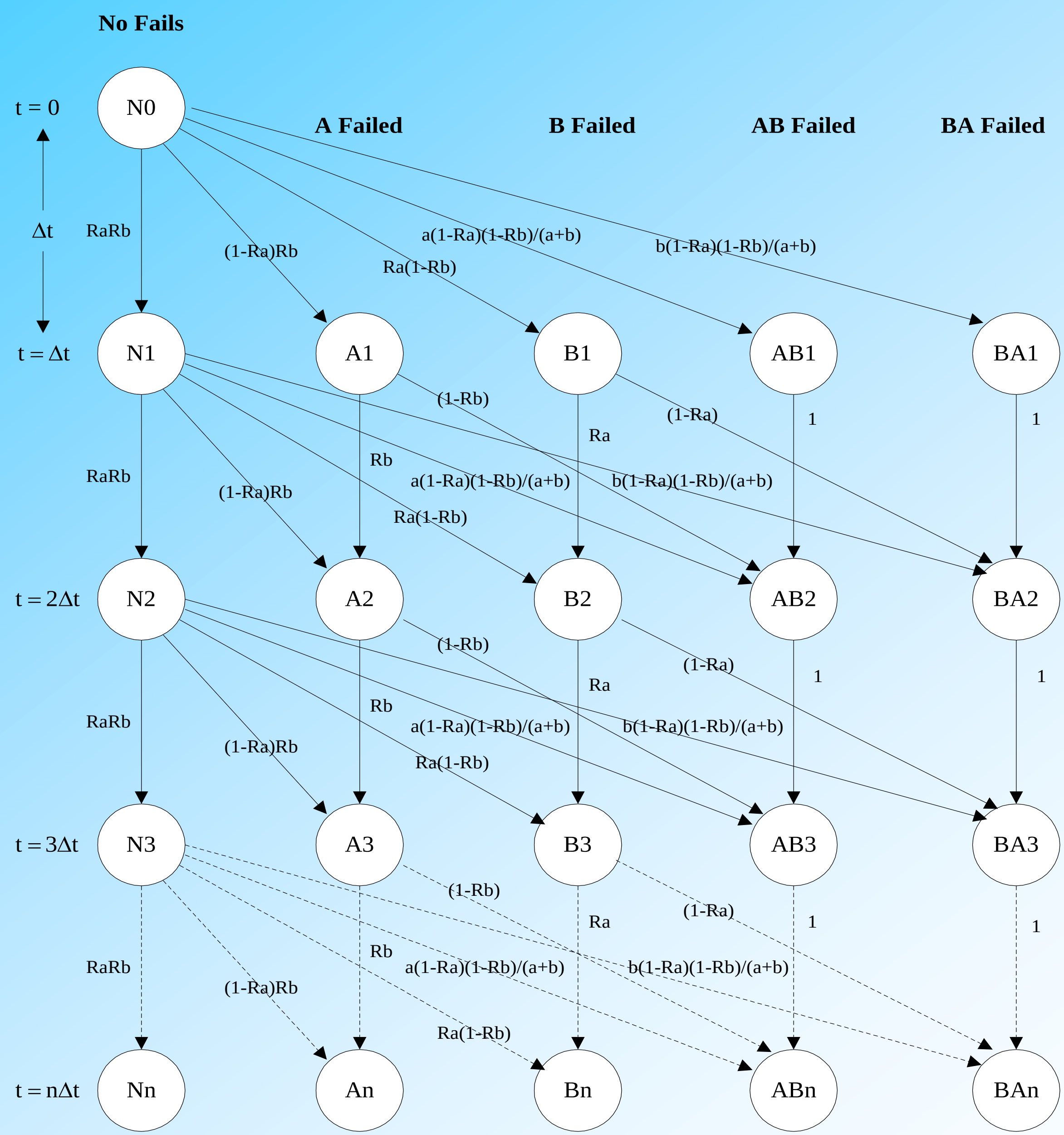
$$P(2) = (1 - e^{-at})e^{-bt} \qquad P(4) = \frac{a}{a+b} + \frac{b}{a+b} e^{-(a+b)t} - e^{-bt}$$

$$P(3) = (1 - e^{-bt})e^{-at} \qquad P(5) = \frac{b}{a+b} + \frac{a}{a+b} e^{-(a+b)t} - e^{-at}$$

Sequence Failure

State Sequence Diagram for "Sequence Failure"

$$P_f = \int_0^t pdf(x)dx$$



State-Dependent Behavior

$$P_f = \int_0^t pdf(x)dx$$

When Markov? For the following situations:

- Standby systems with spares
- Sequence failure dependencies
- **State-dependent behavior**
- Reconfiguration

State-Dependent Behavior cont.

$$P_f = \int_0^t pdf(x)dx$$

Important note:
Whenever a failure rate changes (regardless if the failure rate is constant or not) due to a change in stress, temperature, environment, etc. during a mission, it is considered a change of state, and the problem becomes non-combinatorial.

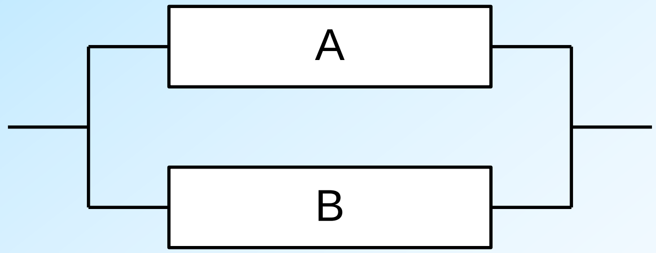
State-Dependent Behavior cont.

$$P_f = \int_0^t pdf(x)dx$$

Combo Example (Failure rates do not change from state to state)
Two black boxes start operation at the same time. Box A has failure rate a, and Box B has initial failure rate b. Successful system operation requires that Box A or Box B or both be working.
Find P_f the Probability of System Failure.

Non-Combo Example (Failure rate(s) change from state to state)
Two black boxes start operation at the same time. Box A has initial failure rate a, and then has failure rate a’ when Box B fails due to increase stress. Box B has initial failure rate b, and then has failure rate b’ when Box A fails. Successful system operation requires that Box A or Box B or both be working.
Find P_f the Probability of System Failure.

RBD



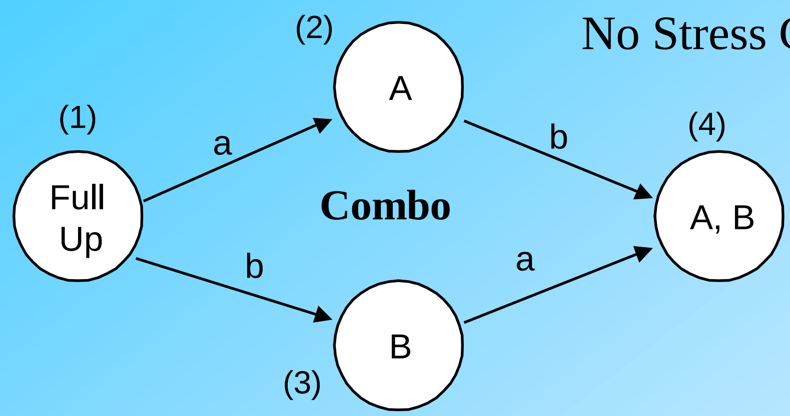
State-Dependent Behavior

Constant Failure Rate Example

$$P_f = \int_0^t pdf(x)dx$$

State Diagrams | **Equations**

No Stress Change



$$P(1) = e^{-at} \cdot e^{-bt}$$

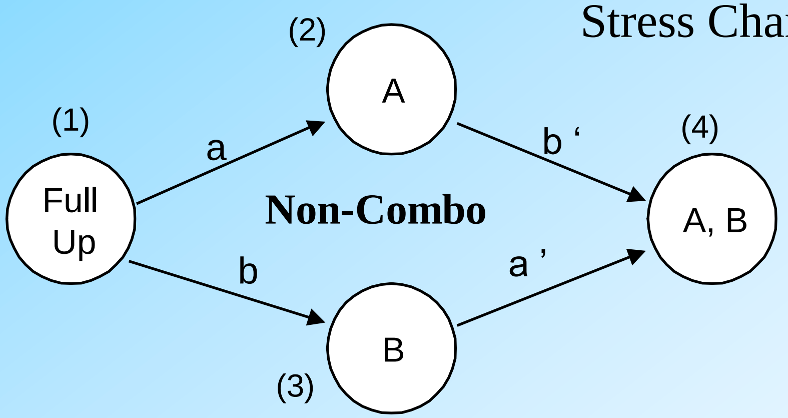
$$P(2) = e^{-bt} \cdot (1 - e^{-at})$$

$$P(3) = e^{-at} \cdot (1 - e^{-bt})$$

$$P(4) = (1 - e^{-at}) \cdot (1 - e^{-bt})$$

Note: The combinatorial nature of solution equations above.

Stress Change



$$P(1) = e^{-(a+b)t}$$

$$P(2) = \frac{a}{a+b-b'}\left(e^{-b't} - e^{-(a+b)t}\right)$$

$$P(3) = \frac{b}{a+b-a'}\left(e^{-a't} - e^{-(a+b)t}\right)$$

$$P(4) = 1 - [P(1) + P(2) + P(3)]$$

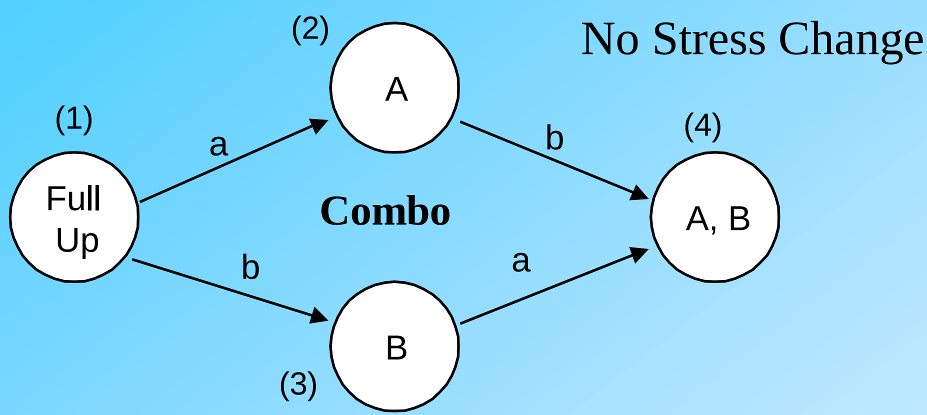
Note: Failure rates change from state to state but remain constant while in each state.

State-Dependent Behavior

Non-constant Failure Rate Example

$$P_f = \int_0^t pdf(x)dx$$

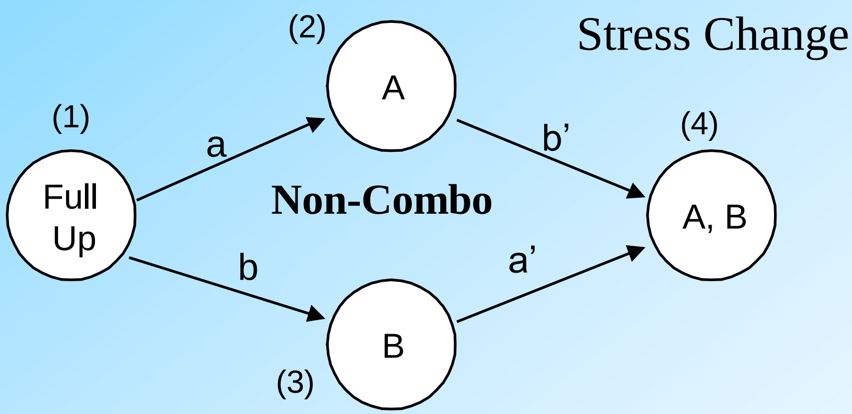
State Diagrams



Equations

$$P(1) = R_a(t) \cdot R_b(t)$$
$$P(2) = R_b(t) \cdot (1 - R_a(t))$$
$$P(3) = R_a(t) \cdot (1 - R_b(t))$$
$$P(4) = (1 - R_a(t)) \cdot (1 - R_b(t))$$

Note: In the Combo case**,** P(i) equations are simply sums and products of component Reliability equations whether failure rates are constant or not.



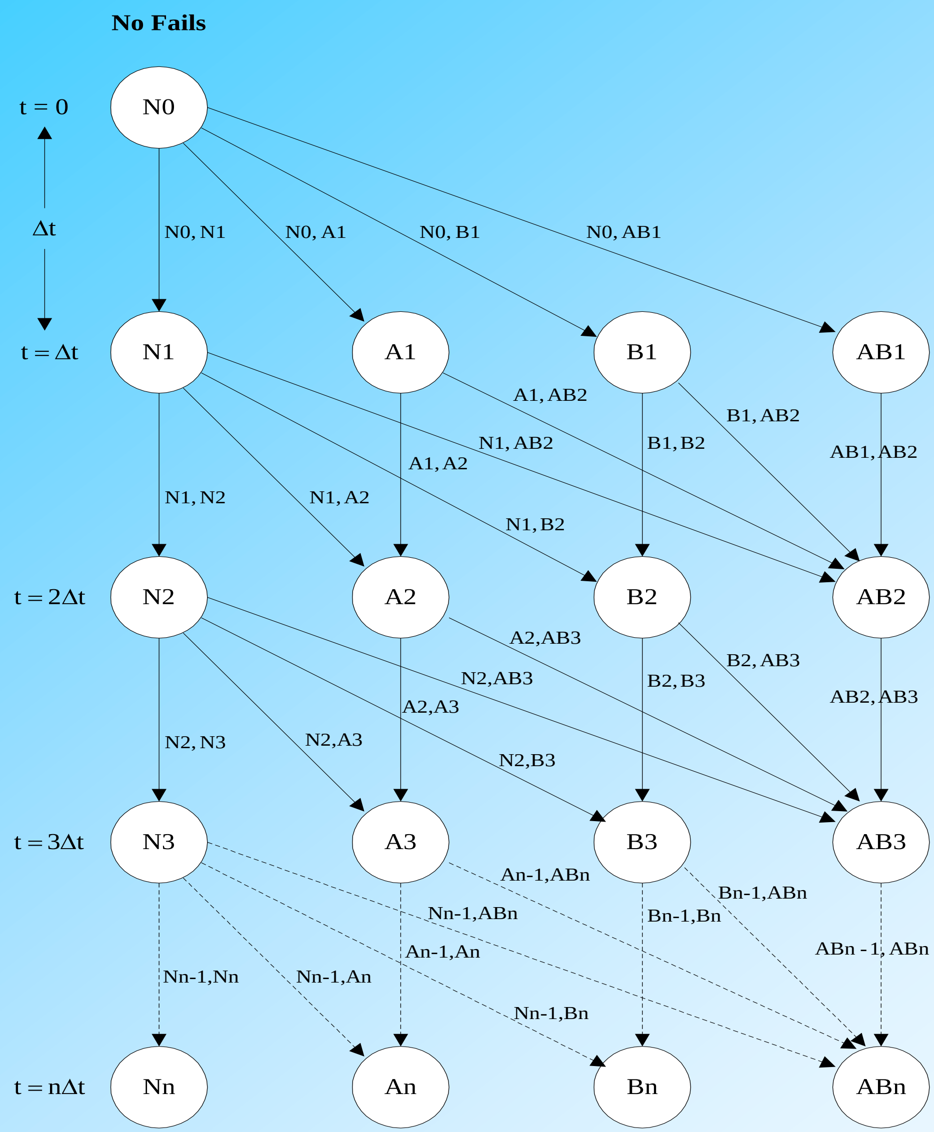
$$P(1) = R_a(t) \cdot R_b(t)$$
$$P(2) \neq R_b(t) \cdot (1 - R_a(t))$$
$$P(3) \neq R_a(t) \cdot (1 - R_b(t))$$
$$P(4) \neq (1 - R_a(t)) \cdot (1 - R_b(t))$$

Note: In the Non-combo case**,** P(i) equations are <u>not</u> simply sums and products of component Reliability equations except for P(1).

2 Devices in Parallel

State Sequence Diagram for 2 Devices in Parallel

$$P_f = \int_0^t pdf(x)dx$$



Transition Probabilities

N0, N1 $e^{-(A+B)\text{Ä}t}$
N0, A1 $e^{-B\text{Ä}t}(1-e^{-A\text{Ä}t})$
N0, B1 $e^{-A\text{Ä}t}(1-e^{-B\text{Ä}t})$
N0, AB1 $(1-e^{-A\text{Ä}t})(1-e^{-B\text{Ä}t})$
N1, N2 $e^{(A+B)\text{Ä}t\ -(A+B)2\text{Ä}t}$
N1, A2 $e^{B\text{Ä}t\ -B2\text{Ä}t}(1-e^{A\text{Ä}t\ -A2\text{Ä}t})$
N1, B2 $e^{A\text{Ä}t\ -A2\text{Ä}t}(1-e^{B\text{Ä}t\ -B2\text{Ä}t})$
N1, AB2 $(1-e^{A\text{Ä}t\ -A2\text{Ä}t})(1-e^{B\text{Ä}t\ -B2\text{Ä}t})$
A1, A2 $e^{B\text{Ä}t\ -B2\text{Ä}t}$
A1, AB2 $1-e^{B\text{Ä}t\ -B2\text{Ä}t}$
B1, B2 $e^{A\text{Ä}t\ -A2\text{Ä}t}$
B1, AB2 $1-e^{A\text{Ä}t\ -A2\text{Ä}t}$
AB1, AB2 1
N2, N3 $e^{(A+B)2\text{Ä}t\ -(A+B)3\text{Ä}t}$
N2, A3 $e^{B2\text{Ä}t\ -B3\text{Ä}t}(1-e^{A2\text{Ä}t\ -A3\text{Ä}t})$
N2, B3 $e^{A2\text{Ä}t\ -A3\text{Ä}t}(1-e^{B2\text{Ä}t\ -B3\text{Ä}t})$
N2, AB3 $(1-e^{A2\text{Ä}t\ -A3\text{Ä}t})(1-e^{B2\text{Ä}t\ -B3\text{Ä}t})$
A2, A3 $e^{B2\text{Ä}t\ -B3\text{Ä}t}$
A2, AB3 $1-e^{B2\text{Ä}t\ -B3\text{Ä}t}$
B2, B3 $e^{A2\text{Ä}t\ -A3\text{Ä}t}$
B2, AB3 $1-e^{A2\text{Ä}t\ -A3\text{Ä}t}$
AB2,AB3 1
Nn-1, Nn $e^{(A+B)(n-1)\text{Ä}t\ -(A+B)n\text{Ä}t}$
Nn-1, An $e^{B(n-1)\text{Ä}t\ -Bn\text{Ä}t}(1-e^{A(n-1)\text{Ä}t\ -An\text{Ä}t})$
Nn-1, Bn $e^{A(n-1)\text{Ä}t\ -An\text{Ä}t}(1-e^{B(n-1)\text{Ä}t\ -Bn\text{Ä}t})$
Nn-1, Abn $(1-e^{A(n-1)\text{Ä}t\ -An\text{Ä}t})(1-e^{B(n-1)\text{Ä}t\ -Bn\text{Ä}t})$
An-1, An $e^{B(n-1)\text{Ä}t\ -Bn\text{Ä}t}$
An-1, Abn $1-e^{B(n-1)\text{Ä}t\ -Bn\text{Ä}t}$
Bn-1, Bn $e^{A(n-1)\text{Ä}t\ -An\text{Ä}t}$
Bn-1, Abn $1-e^{A(n-1)\text{Ä}t\ -An\text{Ä}t}$
ABn-1, Abn 1

Reconfiguration

$$P_f = \int_0^t pdf(x)dx$$

When Markov? For the following situations:

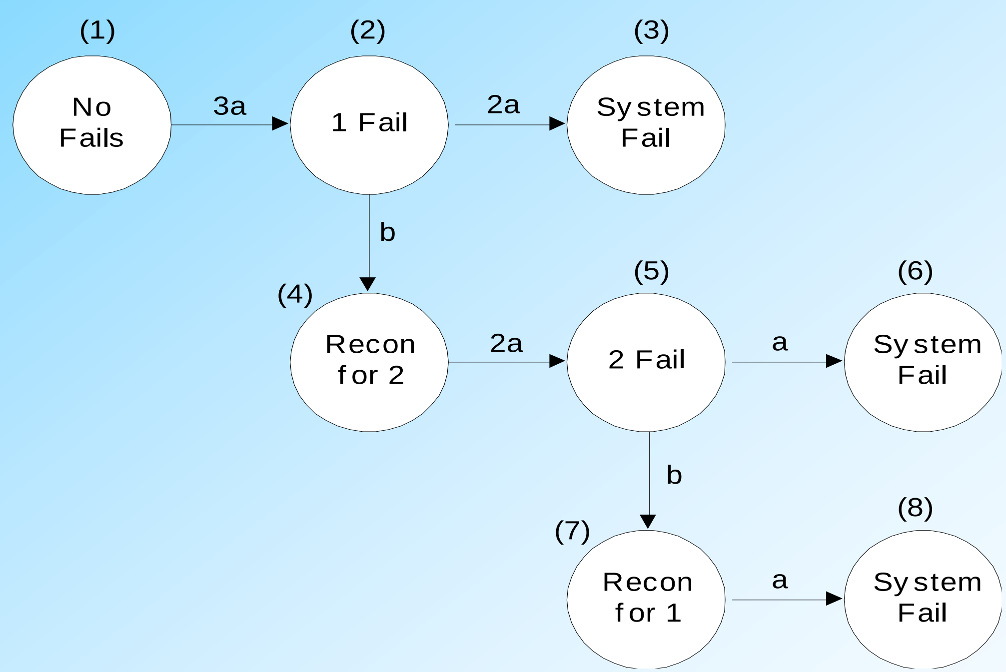
- Standby systems with spares
- Sequence failure dependencies
- State-dependent behavior
- **Reconfiguration**

Reconfiguration

Reconfiguration Example (Non-Combo Type)

$$P_f = \int_0^t pdf(x)dx$$

A system is made up of three computers with each computer having failure rate a. Upon detection of failure of any one of the three, the remaining two reconfigure themselves at rate b, and continue operating. Upon detection of a second failure, the remaining one reconfigures itself at rate b, and continues operating until it fails. Note that if a computer should fail before a reconfiguration is completed, the system fails. Find P_f .



Applied Reliability Symposium, North America 2011

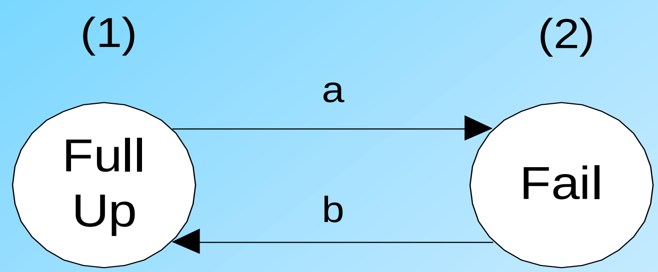
Reconfiguration

Component with Repair Example (Non-Combo Type)

$$P_f = \int_0^t pdf(x)dx$$

A Black Box has constant failure rate a and a constant repair rate b. Upon detection of a failure, the Box goes into a repair process and put back on line. Calculate the probability that the Box will be available.

State Diagram



$$P(1) = \frac{b}{a+b} + \frac{a}{a+b} e^{-(a+b)t}$$

$$P(2) = \frac{a}{a+b} - \frac{a}{a+b} e^{-(a+b)t}$$

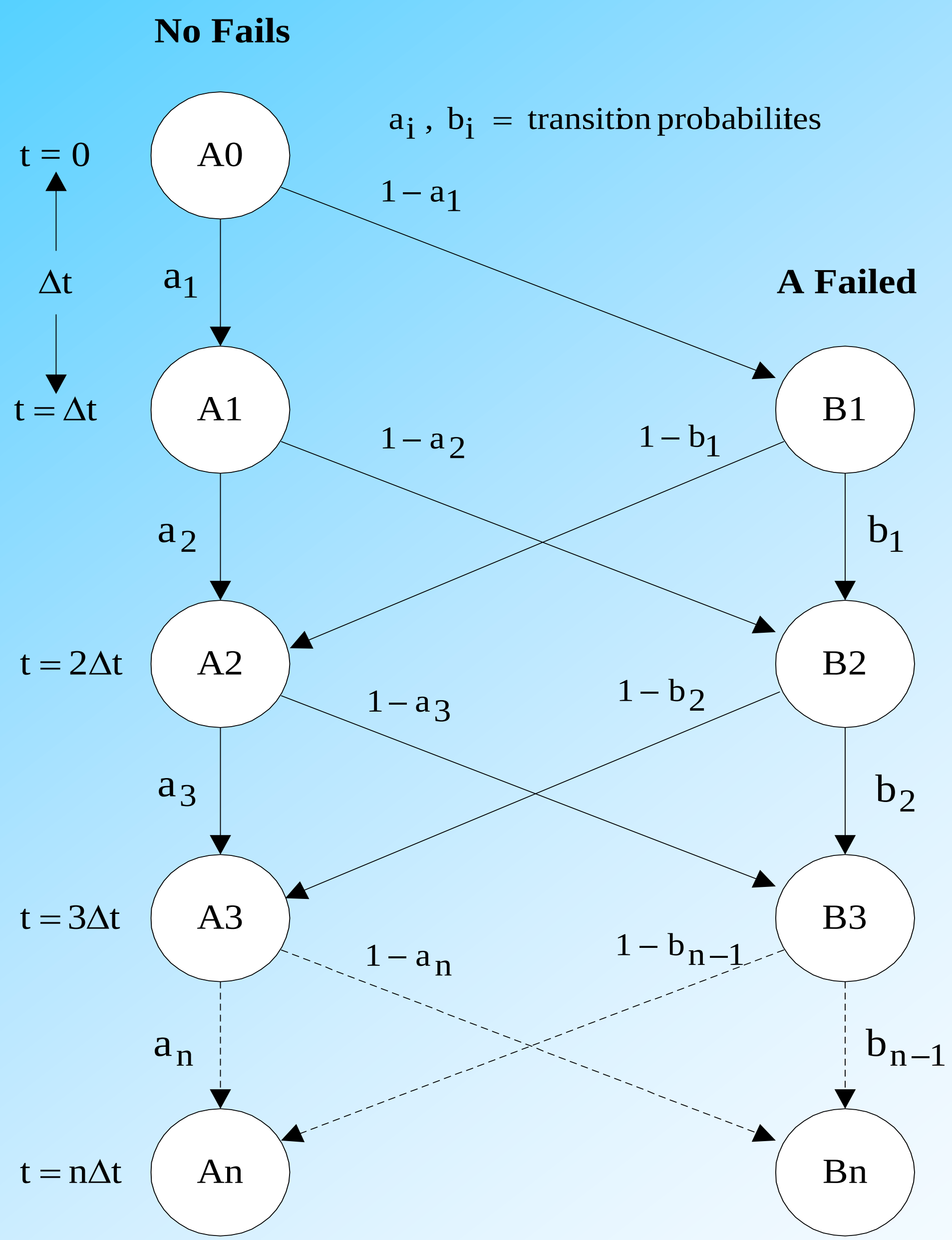
FTA

?

Reconfiguration

SSD for Component with Repair

$$P_f = \int_0^t pdf(x)dx$$



P_f Approximation Techniques

$$P_f = \int_0^t pdf(x)dx$$

A word of caution about P_f approximations:

With respect to System Reliability calculations, it must be stated that for the sake simplifying the mathematics involved, many times approximations are used. For example approximations are used to simplify:

A) failure rate equations, and

B) failure logic equations.

P_f Approximation Techniques

Failure Rate Approximation

$$P_f = \int_0^t pdf(x)dx$$

An example of a failure rate equation simplification :

P_f equations for non-constant failure rate devices like this $\rightarrow$

$$P_f = \frac{1}{s \cdot \sqrt{2\pi}} \int_0^t e^{-\frac{(x-100)^2}{2 \cdot s^2}} dx$$

are replaced with this $\rightarrow$

$$P_f = 1 - e^{-0.01t}$$

P_f Approximation Techniques cont.

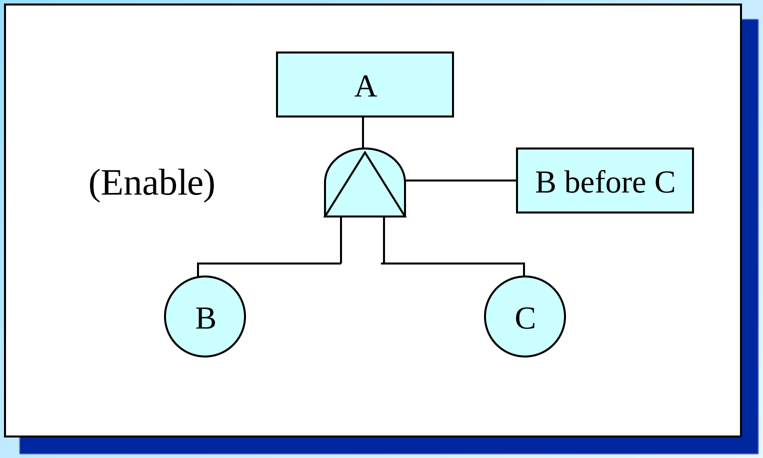
Failure Logic Approximation

$$P_f = \int_0^t pdf(x)dx$$

An example of failure logic simplification:

Cleverly adjusted Boolean expressions are used to approximate solutions (logic aspects) of non-combinatorial problems.

Example from ARP 4761 Issue 1996-12



Event A will occur only if event B occurs and subsequently event C occurs.

P_f Approximation Techniques cont.

$$P_f = \int_0^t pdf(x)dx$$

Points to Consider:

- Caution must be taken because P_f approximation techniques can lead to calculations that may be greater than or less than actual.
- When dealing with systems whose failures are not safety critical, P_f approximations may be “good enough”.
- Markov techniques very often will require extra time and work.
- However when safety critical failures are involved, P_f calculation accuracy will have a high priority, and the use of Markov may have to be considered.

Summary

Summary cont.

$$P_f = \int_0^t pdf(x)dx$$

Facts to keep in mind:

- FTA exhibits a clear representation of any combinatorial logic process, and capable of handling both constant and non-constant failure rates.
- Mixing non-constant with constant failure rate items may require higher math, but has no effect on a system's failure logic.
- MA is a supplement to, and not a replacement for FTA.
- FTA cannot express non-combinatorial logic processes, although FTAs have been used to calculate approximations in the past.
- With respect to Reliability, Markov can be thought of as a buzz word for various methodologies used for solving non-combinatorial logic problems.
- Markov techniques can also solve combinatorial problems. (Not recommended)
- State diagrams are used to represent non-combinatorial logic.
- With respect to non-combo problems, Markov adds qualitative and quantitative accuracy over FTA, but requires more work.
- Approximation techniques can lead to P_f calculations that can be more or less conservative than actual. Extra care must be taken when dealing with safety critical systems.

Summary cont.

$$P_f = \int_0^t pdf(x)dx$$

Markov Approach (for systems exhibiting non-combinatorial logic):

- Create a system RBD and partition combinatorial and non-combinatorial sections.
  (may be challenging – Differences between Combo and Non-combo problems can be very subtle)
- Use and/or logic (standard FTA methods) for all combinatorial sections.
- Create state diagrams for all non-combinatorial sections.
- Determine quantitative solution approach from state diagrams of each section.
- Calculate solutions set to determine probabilities.
  (May be challenging – Solutions methods are subjects of their own)
- Integrate solutions of all sections and determine probability of all undesirable states (events).

Markov Limitations

- Markov quantitative aspects can become difficult when a large number of states, or when non-constant failure rates are involved.

Where to Get More Information

$$P_f = \int_0^t pdf(x)dx$$

- SAE ARP 4761 Issue 1996-12
- Engineering Reliability Fundamentals & Applications – R. Ramakumar
- System Reliability Theory – A. Hoyland & M. Rausand
- Probabilistic Risk Assessment & Management for Engineers & Scientists – H. Kumamoto & E. Henley
- Modeling for Reliability Analysis – Jan Pukite & Paul Pukite
- Mil-Hdbk 338A – Electronic Reliability Design Handbook
- Mil-Std 756B – Reliability Modeling and Prediction

Contact Information

$$P_f = \int_0^t pdf(x)dx$$

vfaraci02@gmail.com

631-977-2597

Questions

$$P_f = \int_0^t pdf(x)dx$$

Thank you for your attention.

Do you have any questions?