Chapter 3 – Redundancy, Spares and Repairs *Subtitle: Detailed synthesis and analysis of FT systems.*

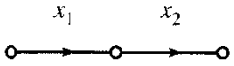
FT system designs should be reliable while using components/units/subsystems with very low failure rates, but it is hard and expensive to obtain failure data from tests since it takes a lot of test time and time is money. Also second-order factors may become more critical than anticipated (dependent failures, common mode, coverage).

Apportionment – distributing the subsystem reliabilities to achieve an overall system R within a cost constraint. Generally a number of good solutions will be produced from attempting to design a system that meets an overall system reliability R_o These family of solutions in conjunction with **parameter sensitivities** can produce a good suboptimal solution that is less sensitive to parameter changes than the true optimum which classically is very sensitive to key parameter(s).

For independent subsystem elements, an initial starting point for subsystem reliabilities in satisfying the overall system reliability R_o goal

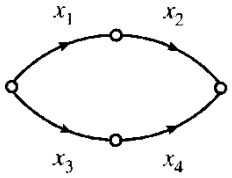
$R_o = \prod_{i=1}^{k} r_i = (r_i)^k$ thus $r_i = (R_o)^{1/k}$ thus apportion the subsystem reliabilities (r_i) to achieve an overall system goal R_o

Unit vs Component Redundancy – start with an initial system made up of two components x_1 and $x_2$which we'll label $R_a(p)$



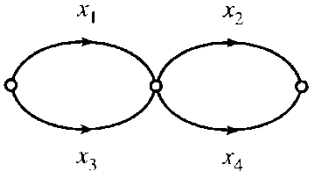
$R_a(p)$

To increase the system reliability with (parallel) redundant components x_3 and x_4 - one can add the two redundant components as a **UNIT**



$R_b(p)$

or adding the redundant components x_3 and x_4 as a **COMPONENT** (individual) redundancy



$R_c(p)$

Given: x_1 and x_2 are independent and identical so $P(x_1) = P(x_2) = p$

thus $R_a(p) = P(x_1)P(x_2) = p^2$ *Series Redundancy*

and $R_b(p) = P(x_1x_2 + x_3x_4) = 2R_a - R_a^2 = p^2 (2 - p^2)$ knowing that for two R_a's in parallel $R_b = R_a + R_a - R_a R_a$

Unit Redundancy

$R_c(p) = P(x_1 + x_3) P(x_2 + x_4) = (2p - p^2)(2p - p^2) = p^2 (2 - p)^2$ (again from IIU in parallel)

Component Redundancy

To compare component redundancy $R_c(p)$ with unit redundancy $R_b(p)$

$R_c(p) / R_b(p) = (2 - p)^2 / (2 - p^2) = 1 + [2(1 - p)^2] / [(2 - p^2)]$ Equation (3.10)

easier just to plug values of p between 0 and 1 to show that $R_c(p) / R_b(p) \geq 1$

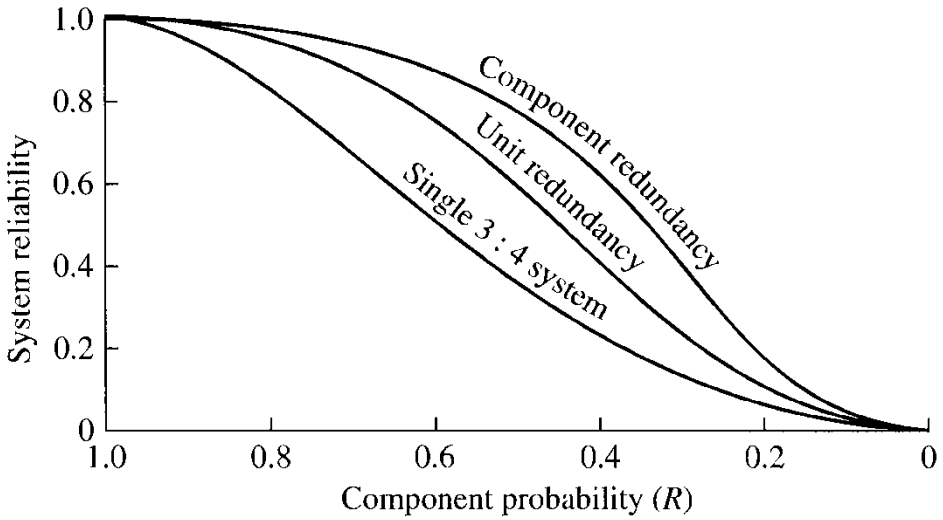
Because $0 < p < 1$ then the term $2 - p^2 > 1$ thus [ ] / [ ] > 0 which means

$R_c(p) / R_b(p) \geq 1$ thus component redundancy $R_c(p)$ is superior to unit (system) redundancy $R_b(p)$

The textbook goes on to show this is the case in all situations (see Figure 3.4 on page 89)
……well almost all the time except for the situations described at the end of the Section 3.3

Comparing these component and unit configurations for an **r-out-of-n** system,
if r = n the structure is a series system and the previous analysis applies.
If r = 1, the r-out-of-n system structure reduces to n parallel elements where component and unit redundancy are identical. The comparison is meaningful for the cases $2 \leq r < n$

As an example, the following Figure 3.5 shows the comparison of component and unit redundancy used in a 3-out-of-4 system (8 modules used in component & unit redundancy)

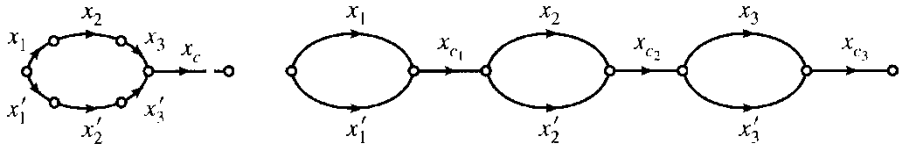


Again component redundancy is superior which can also be proven by tie-set analysis.

If we were to take into account the circuitry required to implement the redundancy (the switches, couplers and voters used to reconfigure the redundant system after a detected failure) then the analysis becomes more complicated and in some cases this extra circuitry can negate the improvement in the redundant configuration.

For the unit versus component redundancy examples → factor in the reliability of the switches necessary to reconfigure the various parallel elements after a detected failure

For system configurations made up of three components x_1, x_2, x_3 in unit or component redundancy, configurations with the addition of the switch reliabilities necessary to reconfigure after a detected failure in one of the elements results in the following:



Unit redundancy R_a with one switch x_c

Component Redundancy R_b with three x_c switches

The textbook solves the reliability expressions for R_a and R_b as before but with switches in series with each of the parallel branches. In addition the textbook uses a constant K for the reliability of the assumed identical switches such that $P(x_c) = Kp$

Then in the comparison of R_a and R_b one solves for the value of K that would make the two configurations equal (or show the effect of R_b's three switches in reducing the component reliability gain to be the same as unit reliability with its one switch).

$$K^2 = \frac{(2p^3 - p^6)}{(2p - p^2)^3 p^2} \qquad (3.16b)$$

Substituting reliability values for p, results in a value of K that makes the two configurations equal. With the switch reliability $P(x_c) = Kp$, then this value of K will determine the switch reliability that makes unit and component reliability equal.

Example: If p = 0.9 then Eq 3.16b yields K = 1.08577850 and Kp = 0.97720 The two configurations are equal (unit = component) if the coupler **failure** probability is 0.0228 If the coupler failure probability is less than 22.8% of the component failure probability, then component redundancy is better. (Component P_f) (X_c) = Coupler P_f then $X_c = 0.228$ Or conversely, if $P_f(x_c) > 22.8\%$ then unit redundancy is better. The effect of three 'lousy' switches on the 3 elements of component redundancy makes the unit redundancy with its one switch better. Lesson – sometimes redundant complexity can do more harm than good. Switch (coupler) reliability can have significant impact in certain situations.

Approximate Reliability Expressions (or what you can do without a computer and not much analysis time)

Using truncated series expansions to approximate the terms e^{-z} that occur many times in reliability analysis; specifically, using the Maclaurin series expansion

$$e^{-Z} = 1 - Z + \frac{Z^2}{2!} - \frac{Z^3}{3!} + \cdots + \frac{(-Z)^n}{n!} + \cdots \qquad (3.17)$$

Which can also be written as a series with n terms and a remainder which accounts for all the missing terms after $(-Z^n/n!)$

$$e^{-Z} = 1 - Z + \frac{Z^2}{2!} - \frac{Z^3}{3!} + \cdots + \frac{(-Z)^n}{n!} + R_n(Z) \qquad (3.18)$$

where

$$R_n(Z) = (-1)^{n+1} \int_0^Z \frac{(Z-\xi)^n}{n!} e^{-\xi}\, d\xi \qquad (3.19)$$

Textbook looks at the truncated series expansions for the hazard function z(t) = f(t) / R(t) and the MTTF using a series expansion for the exponential within the integral definition.

3.5 Parallel Redundancy

We've already looked at parallel redundancy especially for components with the same failure rate but it is worthwhile looking at a graphical comparison of three reliabilities (perfect switches/couplers)

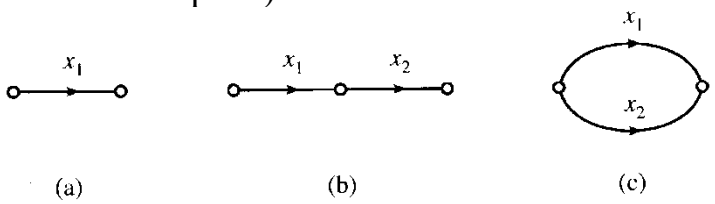
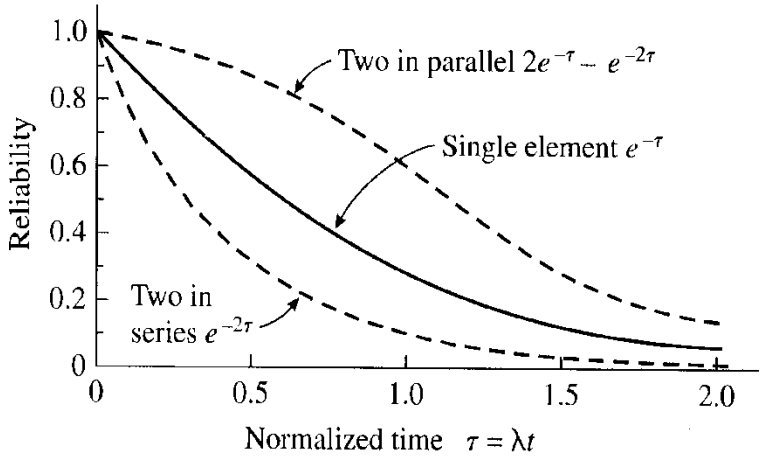


Figure 3.9 Three reliability structures: (a) single element, (b) two series elements, and (c) two parallel elements.

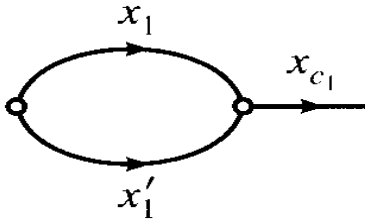


Comparison for constant-hazard components

Series → Worst Case
Parallel → Best Case

3.5.2 Dependent and Common Mode Effects

Common Mode failures are failures that affect all of the elements in a redundant system. An example would be a power supply (the common element) that feeds two elements in a cold or hot standby system such that if the power supply failed, then all modules along with switch circuitry would cease to operate. The switch in the standby system would be considered a common mode element – its failure would impact both the prime and backup modules. Common mode elements sound easy but can be difficult to determine especially in a complex system. Markov, FEMA and associated reliability modeling techniques will sometimes surprisingly reveal common mode failures.



X_c the coupler/switch is common to parallel elements X_1 and X_1'

Most analysis so far has assumed independent failure mechanisms. For example with two parallel elements, both units must fail in order for the system to fail (P_f) thus
$R = P_s = P(x_1 + x_2)$ which results in conditional probabilities when the intersection terms are expanded

$P_s = 1 - P_f = 1 - P(\bar{x}_1\bar{x}_2) = 1 - P(\bar{x}_1)P(\bar{x}_1|\ \bar{x}_2)$ if the failures of x_1 and x_2 are **dependent**

versus $P_s = 1 - P(\bar{x}_1)P(\bar{x}_2)$ if the failures of x_1 and x_2 are **independent**.

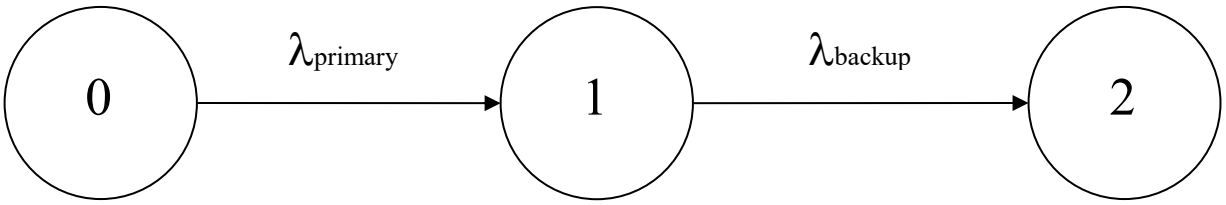
The example given in the textbook is for two parallel space communication channels that have a failure dependency that affects both channels. If $P_f = 0.01$ for each channel then for independent failures the parallel system $P_s = 0.9999 = 99.99\%$. [$P_s = R_1 + R_2 - R_1R_2$]
But if atmospheric interference results in a dependent failure mechanism for both channels where 25% of the failures are due to the atmospheric interference
$P(x_1'|\ x_2') = 0.25$
then for dependent failures $R = P_s = 1 - P(x_1')P(x_1'|\ x_2') = 1 - (0.1)(0.25) = 0.9975$
This reduces the parallel system reliability improvement over a single-string communications system from 100 times better to only an improvement of 4 times better given a dependent channel failure of 25% - probability that channel 1 has failed given that channel 2 has failed → $P(x_1'|\ x_2')$

Dependent mode failures are normally 'overlooked' because the analysis methods rely on independent failures to keep the mathematical (probability) complexity to a minimum.

Standby Systems

We've discussed a simple 'cold' standby system including the Markov Model solution at the end of Appendix B. Let's take another look at (cold) standby systems in comparison to hot standby systems which are also parallel systems (both modules turned on at t = 0) taking into account the necessary switch needed to reconfigure the system when the primary/on-line element fails and the backup/spare element must be engaged.

Figure 3.11 shows a probabilistic (Markov) model for a cold standby system. For the simplified model:



S_0 - primary module engaged and working = 0, backup module good but turned off

S_1 - primary module failed (λ_1), switch over backup module

S_2 - trapping state, backup module failed (λ_2) thus at t both modules have failed system ceases operation

$$\bar{\mathbf{T}} = \begin{matrix} -\lambda_1 & \lambda_1 & 0 \\ 0 & -\lambda_2 & \lambda_2 \\ 0 & 0 & 0 \end{matrix}$$ state transition rate matrix

$$\bar{\mathbf{P}}'(t) = \bar{\mathbf{P}}(t) \cdot \bar{\mathbf{T}}$$

from inspection of $\bar{\mathbf{T}}$

$d P_{s0}(t)/dt = - \lambda_1 P_{s0}(t)$ no failures

$d P_{s1}(t)/dt = \lambda_1 P_{s0}(t) - \lambda_2 P_{s1}(t)$ one failure, switch over

$d P_{s2}(t)/dt = \lambda_2 P_{s1}(t)$ trapped state (failed state)

we found that $P_{s0}(t) = e^{-\lambda_1 t}$ (Eq 3.50) $P_{s1}(t) = \lambda_1 / (\lambda_1 - \lambda_2) [e^{-\lambda_1 t} - e^{-\lambda_2 t}]$ (Eq 3.56)

if we are in states P_{S0} or P_{S1} then the system is operating and $R(t) = P_{s0}(t) + P_{s1}(t)$

However if the two failure rates are the same (which was the case for the Markov Model in the Appendix B lecture) then $\lambda_1 = \lambda_2$ and the expression for $P_{s1}(t)$ becomes 0/0 … but using l'Hospital's rule and taking the derivative of the numerator and the denominator separately with respect to λ_2 then taking the limit as $\lambda_1 \rightarrow \lambda_2$ results in

$R(t) = e^{-\lambda t} + \lambda t e^{-\lambda t}$ or by knowing that $R(t) = 1 - P_{S2}(t)$ since state 2 is the failed state

[This is the same as the solution for the Markov Model in the Appendix B lecture.]
R(t) is the probability of zero failures in t hours (both systems are good) **PLUS** the probability of one failure in t hours (the on-line system has failed but the standby system is good). This assumes that switching from the primary to backup was without errors including detection of the failure in the primary module.

To consider a more realistic model, one that takes into account the necessary switching circuitry to switch from the on-line/primary module to the standby/backup module after the failure of the on-line module has been detected (which we will call **coverage** in Section 3.8.4). Assume that the switching mechanism is a simplex device (only one switch in series with the parallel configuration of the standby system). An 'imperfect' switch would just multiply the reliability of the standby system R(t) since the switch is in series and thus would degrade the overall system reliability:

$$R(t) = R_{switch}\,(R_{standy\ system}) = \epsilon^{-\lambda_s t}(\epsilon^{-\lambda t} + \lambda t\,\epsilon^{-\lambda t})$$

The comparison of this scenario is shown in Figure 3.13

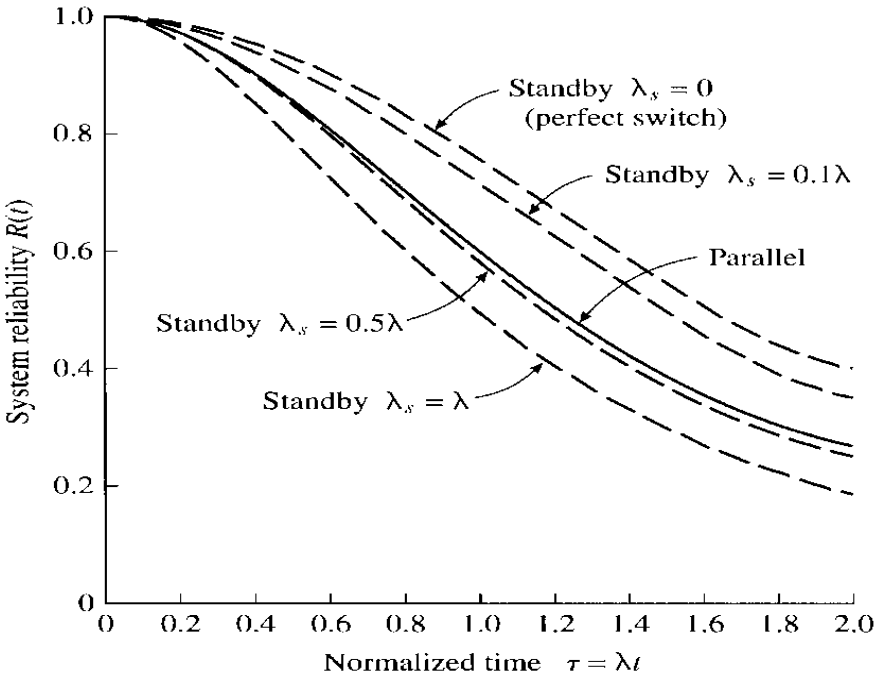


Figure 3.13 A comparison of a two-element ordinary parallel system with a element standby system with imperfect switch reliability.

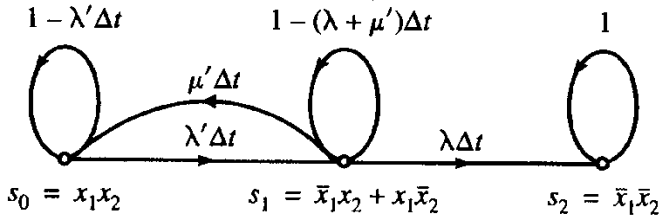
A Parallel configuration could be considered a hot standby with a perfect switch as compared to Fig 3.13 (cold standby system)

We can refine our model even more by considering: *(1)* that the switch only fails when switching from S_0 to S_1 (it shouldn't switch when the on-line module is good) thus the $\epsilon^{-\lambda_s t}$ only multiplies the second term in R(t) not both, *(2)* switching failures even if the on-line module is good (a detection failure), *(3)* switch jitter, *(4)* switch delays (by the time the spare module is engaged the entire system has failed), *(5)* non-similar failure rates ($\lambda_1 \neq \lambda_2$), *(6)* non-identical modules, *(7)* non-independent failures (failure of the on-line module impacts the spare module), *(8)* common-mode effects (high operating temperatures fail both modules), *(9)* the backup module fails even when it is turned off (quiescent/latent failures), *(10)* the on-line module has failed at t = 0, *(11)* factoring Kranz's Law into the model ("failure is not an option"), *(12)* Murphy's Law is factored into P_f (if can fail at the worst possible time – it will), etc., etc., …….

3.8 Repairable Systems

We discussed repairable systems in Appendix B and the implications of availability **A(t)** along with a Markov model of a single component with repair μ *Why not consider R(t)?*

The textbook now considers a three-state Markov reliability rate model (dual processor or a hot/cold standby system)



where $\lambda' = 2\lambda$ for an ordinary system
$\lambda' = \lambda$ for a standby system
$\mu' = \mu$ for one repairman
$\mu' = k\mu$ for more than one repairman ($k > 1$)

Figure 3.14 A Markov reliability model for two identical parallel elements and k repairmen.

Does this make sense for k > 2 ?? Substitutions into this generalized solution can be dangerous.

I prefer the simplified model without the Δ t's & self-loops (the probability of no state change or 1 – probability of leaving) and deriving the differential equations from the state transition rate matrix, i.e., a matrix based schemes. Section 3.8.2 goes through the same exercise with a non-simplified Markov Model for writing the associated differential equations and then using Laplace transforms to solve these differential equations.

$$\mathbf{P'}(t) = \mathbf{P}(t) \cdot \mathbf{T} \qquad \mathbf{T} = \begin{matrix} -\lambda' & \lambda' & 0 \\ \mu' & -(\lambda + \mu') & \lambda \\ 0 & 0 & 0 \end{matrix}$$

state transition rate matrix
simplified model

from inspection of $\bar{\mathbf{T}}$

$$\frac{dP_{s0}(t)}{dt} = -\lambda' P_{s0}(t) + \mu' P_{s1}(t) \qquad \textit{from 1st column of } \mathbf{T}$$

$$\frac{dP_{s1}(t)}{dt} = \lambda' P_{s0}(t) - (\lambda + \mu') P_{s1}(t) \qquad \textit{from 2nd column of } \mathbf{T}$$

$$\frac{dP_{s2}(t)}{dt} = \lambda P_{s1}(t) \qquad \textit{from 3rd column of } \mathbf{T}$$

where $R(t) = P_{s0}(t) + P_{s1}(t) = 1 - P_{s2}(t)$

The author references the Laplace transform solution for the above differential equations. The Siewiorek textbook reference details the involved solution in its Chapter 5 – *Evaluation Criteria* for a very similar Markov model so to give you some respect for the gruesome differential equation solution for the reliability R(t)

$$R(t) = \frac{4\lambda^2 \exp\left(-(1/2)(3\lambda + \mu - \sqrt{\lambda^2 + 6\lambda\mu + \mu^2})t\right)}{(3\lambda + \mu)\sqrt{\lambda^2 + 6\lambda\mu + \mu^2} - \lambda^2 - 6\lambda\mu - \mu^2} - \frac{4\lambda^2 \exp\left(-(1/2)(3\lambda + \mu + \sqrt{\lambda^2 + 6\lambda\mu + \mu^2})t\right)}{(3\lambda + \mu)\sqrt{\lambda^2 + 6\lambda\mu + \mu^2} + \lambda^2 + 6\lambda\mu + \mu^2}$$

Shooman proposes looking at a single-value metric, the MTTF which derives from the integration of R(t) which is the sum of the first two-state probabilities $P_{S0}(t) + P_{S1}(t)$. Thus

$$MTTF = (\lambda + \mu' + \lambda') / (\lambda\lambda')$$

Substituting various failure rate values of λ' into the MTTF expression and assuming a single repairman (μ' = μ), the following table is obtained:

TABLE 3.4 Comparison of MTTF for Several Systems

Element	Formula	For $\lambda = 1$, $\mu = 10$
Single element	$1/\lambda$	1.0
Two parallel elements—no repair	$1.5/\lambda$	1.5
Two standby elements—no repair	$2/\lambda$	2.0
Two parallel elements—with repair	$(3\lambda + \mu)/2\lambda^2$	6.5
Two standby elements—with repair	$(2\lambda + \mu)/\lambda^2$	12.0

Hot Standby
Cold Standby
Hot Standby w/repair
Cold Standby w/repair

The primary observation is that a **repair strategy greatly increases the MTTF** especially for complex systems (the last two systems shown above). The strategy is also very cost effective when the ratio of μ / λ is large, which is normally the case since the magnitude of the repair rate is repairs per hour (2 units/hour) and the failure rate is very small (10^{-8} failures per hour).

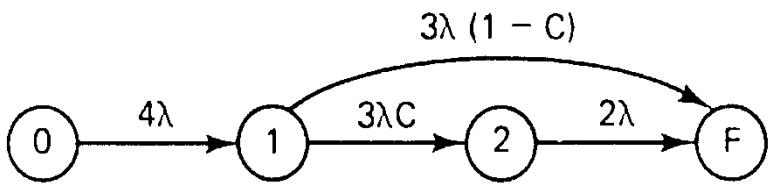
Coverage - the probability that a system can recover given that a fault has occurred. This infers that there is a probabilistic failure in the ability to detect, reconfigure and recover within a FT system.

Coverage factors in the unreliability (imperfection) within the decision process of the failure detection mechanism; in other words, a decision unit cannot detect 100% of all failures. It can only *cover* (detect) a fraction c ($0 < c < 1$) of all the possible failures. According to Sieworek, a typical diagnostic program usually detects only 80 – 90% of all possible faults (c = 0.8 to 0.9). It is probably worse in complex systems (Shuttle - use of artificial intelligence in CAU). Coverage failures are the dominant source of system failures in highly reliable systems.

Permanent faults are easier to detect than transient and intermittent faults which are much harder to detect. Detection and reconfiguration time is critical since a second fault can occur during the interval when the system is trying to reconfigure from the first failure.

Coverage is normally utilized in Markov models (Figure 3.15 shows a three-state parallel model taking into account coverage effects at states 0 and 1)

Markov model of a two-out-of-four structure with imperfect coverage for just the 2nd failure (you would have a similar situation coming out of State 0 but not necessarily State 2. Why?)



Fault Definitions

Permanent – a failure or fault that is continuous and stable, a hard fault. In hardware, a permanent fault is an irreversible physical change until repaired.

Intermittent – a fault that is only occasionally present due to unstable hardware or varying hardware or software states, e.g., as a function of load or activity.

Transient – a fault resulting from temporary **environmental** conditions. A soft fault.

3.9 RAID (**R**edundant **A**rray of **I**ndependent **D**isks) Systems Reliability

When solid state memory looked like it would replace hard disk drives, the disk manufactures began to make great strides in the cost, storage capacity and reliability of HDDs. RAID became a way of utilizing these opportunities in HDD technology to increase bandwidth (data transfer speed) and utilize FT techniques (redundancy & data coding techniques). HDDs are still mechanical devices and using n more of them increases the effective bandwidth (→ nBW) but also reduces the MTTF for n drives

(effective $\lambda \rightarrow n\lambda = n/MTTF$).

Textbook describes the various RAID configurations although RAID 0 (striped disks with data split between two or more disks) which is common although it doesn't increase reliability (actually decreases it – why??). RAID 0 does provide higher bandwidth (speed). NVIDIA is a leading company of motherboards and software that utilize RAID configurations. Their software allows easy re-configuration to RAID.

- RAID 1 (mirrored disks) a backup solution, using two (possibly more) disks that each store the same data so that data is not lost as long as one disk survives. Total capacity of the array is just the capacity of the single smallest disk.
- RAID 5 (striped disks with parity) combines three or more disks in a way that protects data against loss of any one disk; the storage capacity of the array is reduced by one disk.
- RAID 10 (or 1+0) uses both striping and mirroring.

Typical Commercial FT Systems – Tandem and Stratus (note M. Yin's PowerPoint slides)

Duplicated: CPUs, I/O and memory controllers, disk controllers, communication controllers and busses along with duplicated support subsystems (e.g., power, cooling).

Tandem → Compaq → HP (an FT success story although taken-over by HP)

Used predominately for transaction (real-time) processing, related to money (stocks, bond, sales).

Shooman makes the point that typical computer system metrics have the same reliability as today's automobile thus for critical computer applications you must use fault tolerance (redundancy) if you want your computer to be more reliable that your car.

Both Tandem and Stratus utilize software FT in addition to the key points of hardware FT like BIT (built-in test) which is a key hardware technique.

Tandem uses a technique of heartbeat signals and hot spares. Recovery scheme requires 'regroup' time to reconfiguration leading to re-initiation of the processing on the hot spare.

Tandem's technique gets more useful work out of its spares as compared to the Stratus technique of duplicated processing pairs where the duplicate processors perform the hardware checks and if detected, switch immediately to the spare processing spare which were performing the exact same operations. You could get into a split pair scenario (2 modules versus another 2 modules which therefore can't be voted) but highly unlikely.

All implementations today are MOTS (modified off-the-shelf).

Both systems incorporate schemes for risk management (business continuity/disaster recovery). Examples: story of the redundant systems in each of the 9/11 twin towers, story of the Stratus system used in a major railroad (catastrophic weekend A/C failure, inability to contact customer, automatic shipment of spares when failures began to be reported), story of India's stock exchange based on Stratus computer systems.

Schemes exist that use geographically dispersed systems not only for business continuity but hot spares (backups) that are utilized in all of the real-time checking mechanisms used in various FT mechanisms (using Internet high-speed backbones/excess capacity).

Overall Commercial Company FT Beliefs

1. Perceived risk to their financial health will drive commercial entities to demand improved reliability, availability and serviceability.
2. Companies will pay a premium for such improvements where it seems that the upper bound on the acceptable improvement premium is a factor of 2 → the cost to implement a hot standby.
3. For companies, especially those in the high-tech sector, the rapid change in fundamental technology results in major expenditure of resources just trying to keep up.
4. In the short term, Hardware FT is easier than Software FT but in the long term, improving software is more likely to provide more robust FT solutions.
5. Improving the "-ilities" of a system (reliability, availability, quality) requires many small (incremental) improvements.
6. Like most accident investigations, the small overlooked/incidental failures usually lead to the catastrophic failures (not paying attention to the details). The human factor is always a high probability source of failures (root cause).