# Chapter 4 *N*- Modular Redundancy

Using of multiple levels of redundancy for fault tolerance was established:

(1.) Using automata theory (logic gates, state machines, combinatorial/sequential logic) to model digital circuits and computational operations.

(2.) As a means of making reliable computers from less reliable components.

The classic *n*-modular redundant example is triple modular redundant: TMR R(3,0)

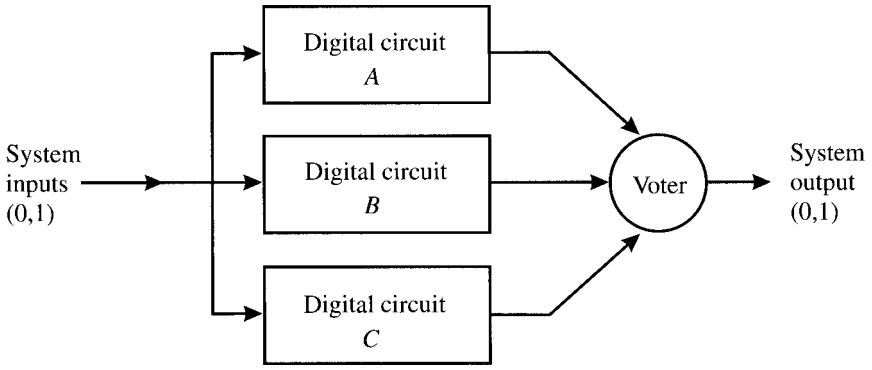



**Figure 4.1** Triple modular redundancy.

Identical modules synchronized to the level of being able to compare and vote on the outputs (usually means lock-step operations).

Requires identical system INPUTS (usually voted before agreeing on the **one** input set that will be 'submitted' to the 3 modules system inputs).

Requires a very high precision and fault tolerant clock.

Assuming an ideal voter, $R_v(t) = 1.0$ then

$\mathbf{R_{TMR}} = P(A \cdot B + A \cdot C + B \cdot C)$

Assuming independent and identical modules, then using the binomial theorem

$$B(r; n, p) = \binom{n}{r} p^r (1-p)^{n-r}$$

$$\mathbf{R_{TMR}} = B(3:3) + B(2:3) = \binom{3}{3} p^3(1-p)^0 + \binom{3}{2} p^2(1-p)^1$$

all three systems operating (one combination) + 2-out-of-3 systems operating (three possible combinations)

$$\mathbf{R_{TMR} = 3\,p^2 - 2\,p^3 = 3\,R_m^2 - 2\,R_m^3} = 3\,e^{-2\lambda t} - 2e^{-3\lambda t}$$ (for constant failure rate)

If a more realistic simplex voter is incorporated into the reliability formulation then

$R_{TMR} = R_v(3\,p_c^2 - 2\,p_c^3)$ where $R_v$ is the reliability of the voter. (Eq 4.2)

## Section 4.4.3 System Error Rate

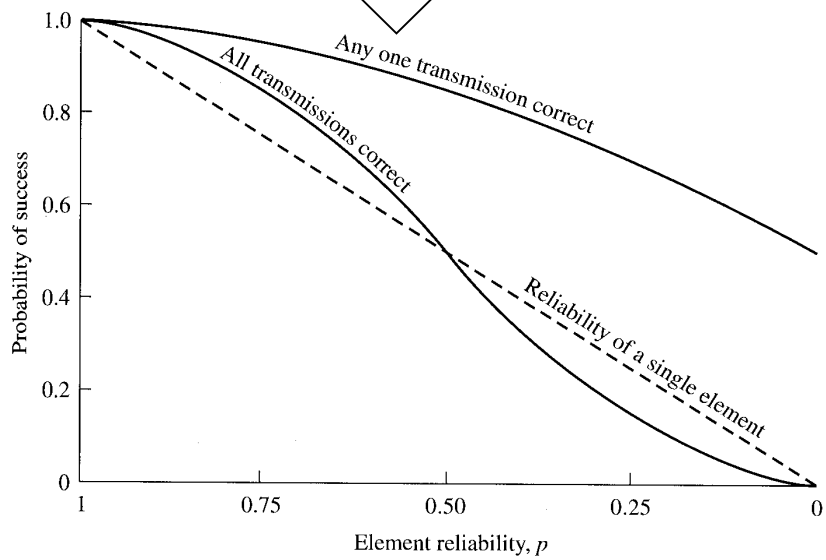
Formulate the probability of a correct output from the TMR system taking into consideration that a lot of failures are transient and $R_{TMR}$ is a worst-case analysis result.

1. If the correct output is 1 then success ($P_1$) is not sending a 0 and conversely if the correct output is 0, then success ($P_0$) is not sending a 1
2. Assume all states and all modules fail independently
3. Assume that sending a 1 or a 0 is equally likely so $P_{success} = (P_0 + P_1) / 2$

where $P_0 = 1 - P(A_1'B_1'+A_1'C_1'+B_1'C_1')$ and $P_1 = 1 - P(A_0'B_0'+A_0'C_0'+B_0'C_0')$

$P_0$ is the probability of a 0 output which is unity minus the probability of two or more '1 failures'

Then $P_{success} = ½ + ¾\,p - ¼\,p^3$ which is Eq 4.8 – any one xmission correct
p = element (module) reliability
full development of equation on page 150




Note that if the level of element (module) reliability is low, then $R_{SIMPLEX} > R_{TMR}$

Another oddity can be shown using just single-value metrics like $MTTF_{TMR}$

$R(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$ then $MTTF_{TMR} = \int R(t) = \frac{3}{2\lambda} - \frac{2}{3\lambda} = \frac{5}{6\lambda}$ which is less than $\frac{1}{\lambda}$

Thus $MTTF_{TMR} < MTTF_{SIMPLEX}$ which is a pitfall of using single valued metrics.
$5/6\,\lambda < 1/\lambda$

With just one failure, TMR continues to operate while continuously voting out the one failed module. We'll annotate this as a TMR $_{3-2}$ which reaches a trapped state after two failures. When the second failure occurs, the voter can't work since there is no longer a majority of working modules and it cannot determine which of the modules has failed.

In a special case you can use error detection to lock-out the failed modules (deselect); however, the voter will no longer have a majority voting capability. After the second failure, one can make a manual decision to deselect the module that has behaved erratically and downmode to a simplex system assuming the other module is operating properly. One can use such things as error logs of repairs, # of operational hours, # of detected transient errors, etc., to select the working module. For this type of TMR which ends up as a simplex system (1 module) after the second failed module is removed/deselected, add a 1-out-of-3 binomial probability term to the $R_{TMR\ 3-2}$ equation. This downmoded system will be annotated as:

$$R_{TMR\ 3\text{-}2\text{-}1} = R_{TMR\ 3\text{-}2} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} p^1 (1-p)^2 = \underbrace{3p^2 - 2p^3}_{R_{TMR\ 3\text{-}2}} + \underbrace{3p(1-p)^2}_{\text{probablily of } \mathbf{1:3}} = \mathbf{p^3 - 3p^2 + 3p}$$

and for constant failure rates $R(t)_{3\text{-}2\text{-}1} = e^{-3\lambda t} - 3e^{-2\lambda t} + 3e^{-\lambda t}$

$$\text{MTTF} = \int R(t)\,dt \text{ thus } MTTF_{3\text{-}2\text{-}1} = \frac{11}{6\lambda}$$ (an improvement over the classic TMR whose $MTTF_{3\text{-}2} = 5/6\ \lambda$)

Using truncated series expansion of the reliability equations (page 152):

Simplex $R(t) = e^{-\lambda t} \cong 1 - \lambda t$ $\quad R_{TMR\ 3\text{-}2} \cong 1 - 3(\lambda t)^2$ $\quad R_{TMR\ 3\text{-}2\text{-}1} \cong 1 - \lambda^3 t^3$

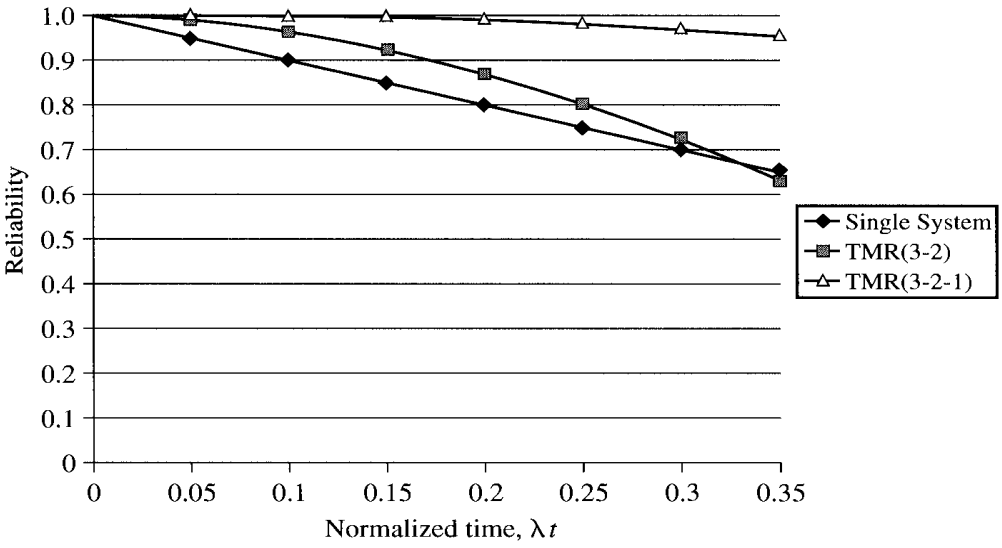
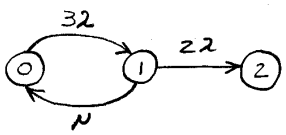



**Figure 4.3** Comparison of the reliability functions of a single system, a TMR 3–2 system, and a TMR 3–2–1 system in the high-reliability region.

Normally in high reliability systems, you don't want to ever give up any working resources so a TMR (3-2-1) is a viable design decision even if it requires a manual decision (most likely an automatic decision with a manual over-ride option).

Another means of keeping resources in a $R_{TMR\ 3\text{-}2}$ system → Repair. (Simplified Model)

EXAMPLE MARKOV MODELING FOR A TMR SYSTEM WITH REPAIR




$p_0$ = FULLY OPERATIONAL (ALL 3)

$p_1$ = 1 FAILURE

$p_2$ = 2 FAILURES, SYSTEM FAILS

$$\vec{T} = \begin{bmatrix} -3\lambda & 3\lambda & 0 \\ \mu & -\mu-2\lambda & 2\lambda \\ 0 & 0 & 0 \end{bmatrix}$$

$P_{00}$ $P_{01}$ $P_{02}$
$P_{10}$ $P_{11}$ $P_{12}$
$P_{20}$ $P_{21}$ $P_{22}$

$$MTTF = \frac{5}{6\lambda} + \frac{\mu}{6\lambda^2}$$

TMR WITHOUT REPAIR

$MTTF = \frac{5}{6\lambda}$

THUS REPAIR ADDS TERM $\mu/6\lambda^2$

WHICH IS A CONSIDERABLE IMPROVEMENT

EXAMPLE ON PG 170 SHOWS AN INCREASE BY A FACTOR OF 3 !

THUS ONE SHOULD STRIVE TO DESIGN A SYSTEM WITH ON-LINE REPAIR WHICH INFERS BRINGING A "MODULE" BACK INTO THE SYSTEM

Repair for a TMR is very difficult to execute especially for real-time systems. However, most 'failures' are intermittent/transient and you really don't have to repair - just validate that the voted-out system is still good. Keep it available and bring it back into the TMR system using the history of the voting on this 'failed' module. One should keep it locked out from sending commands when voted out but note that if later it is successfully passing on the majority votes with the other two modules, then bring it back into the TMR system.

Problems of bringing a module back into the system?

- Most operating systems are not deterministic.
- Realigning large amounts of system memory takes time.
- Knowing a minimal state for re-initialization of the system is a good approach.
- Computers are so fast with multiple co-processors and *n* number of cores that it makes bringing a repaired system back on-line very difficult and time consuming.

## Section 4.4 *N*-Modular Redundancy ($N > 3$)

Since voting is normally used in *n*-modular redundancy schemes, *n* should be odd.
(An even number of modules can result in a split vote, e.g., 2 on 2)

Assuming a single perfect voter with odd number of modules (2n + 1 votes):

$$R = \sum_{i=n+1}^{2n+1} B(i:2n+1) = \sum_{i=n+1}^{2n+1} \binom{2n+1}{i} p^i (1-p)^{2n+1-i} \qquad (4.17)$$

For modules with constant failure rates, $R(t) = e^{-\lambda t}$, then with *n* modules as a function of mission time (normalized to $\lambda t$ where $\lambda = 1$) produces:

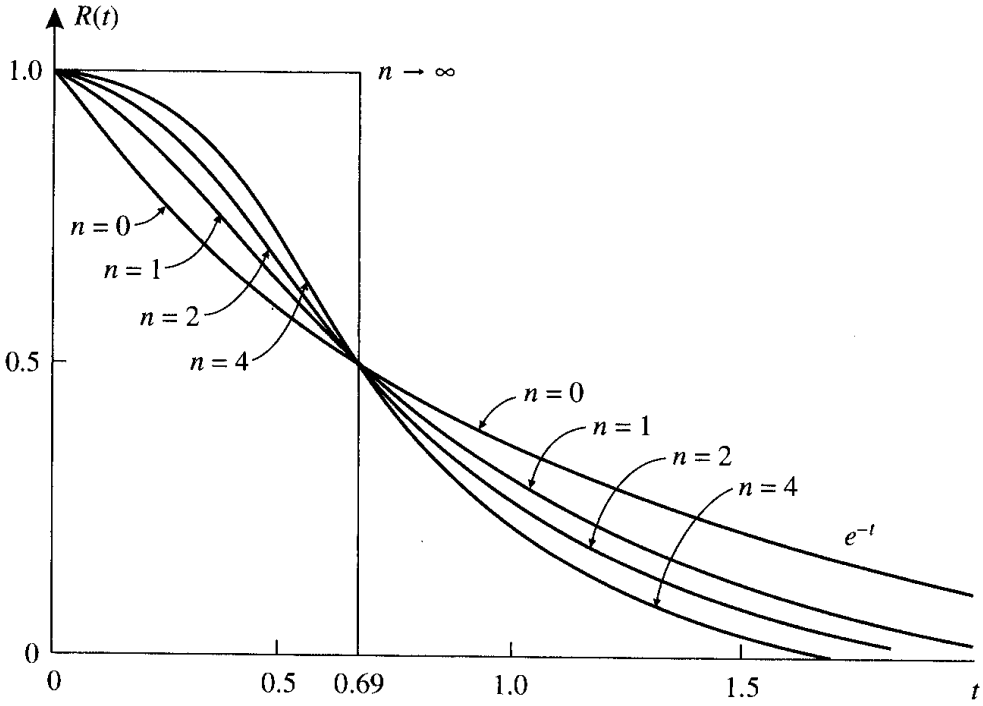


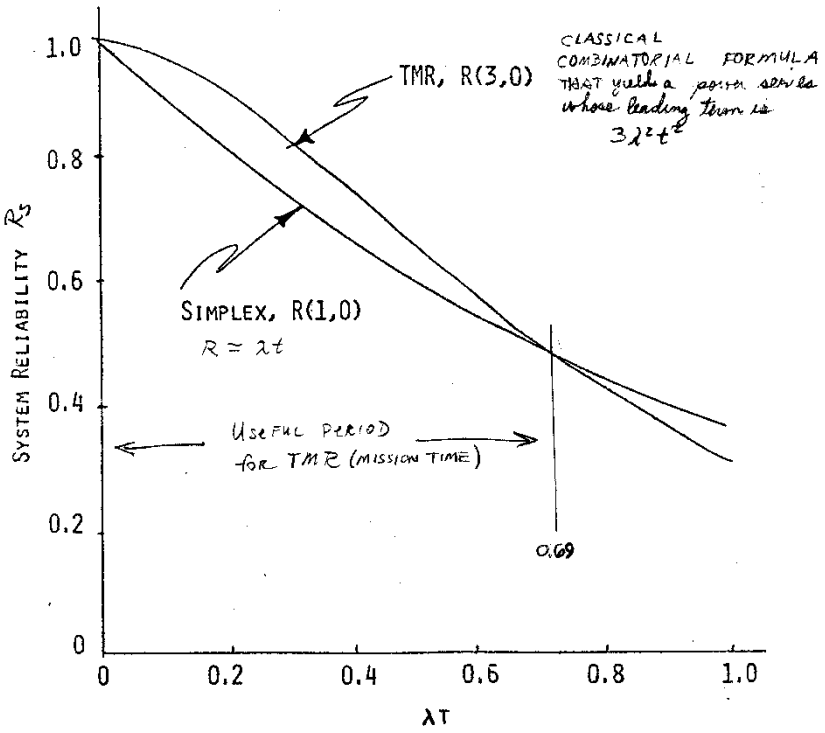

**Figure 4.4** Reliability of a majority voter containing $2n + 1$ circuits. (Adapted from Knox-Seith [1963, p. 12].) (Note that graph is normalized for $\lambda = 1$.)

Note that as $n \rightarrow \infty$, $MTTF_{n \rightarrow \infty} = 0.69 / \lambda$ The normalized y-axis ($\lambda$ t) shows where simplex reliability becomes a better solution than n-module reliability → a cross-over.

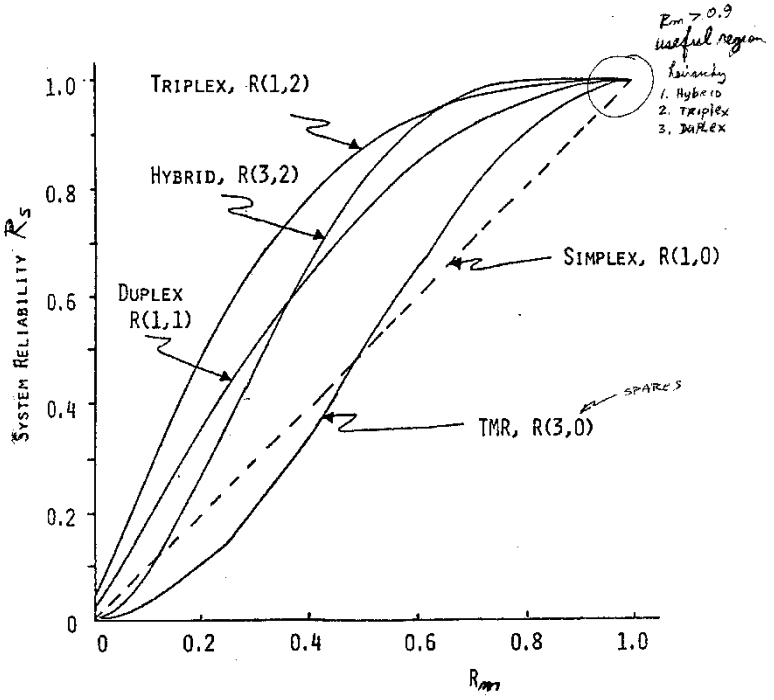
n-module redundancy is only better than a single system for $\lambda t < 0.69$, which is the high reliability region of the above graph.

Thus *n*-modular redundancy is only useful for specific mission times and only in the high-reliability regions for various values of *n* and $\lambda$.

**TMR Reliability versus λt** (essentially the same graph as Fig 4.4 but simpler)

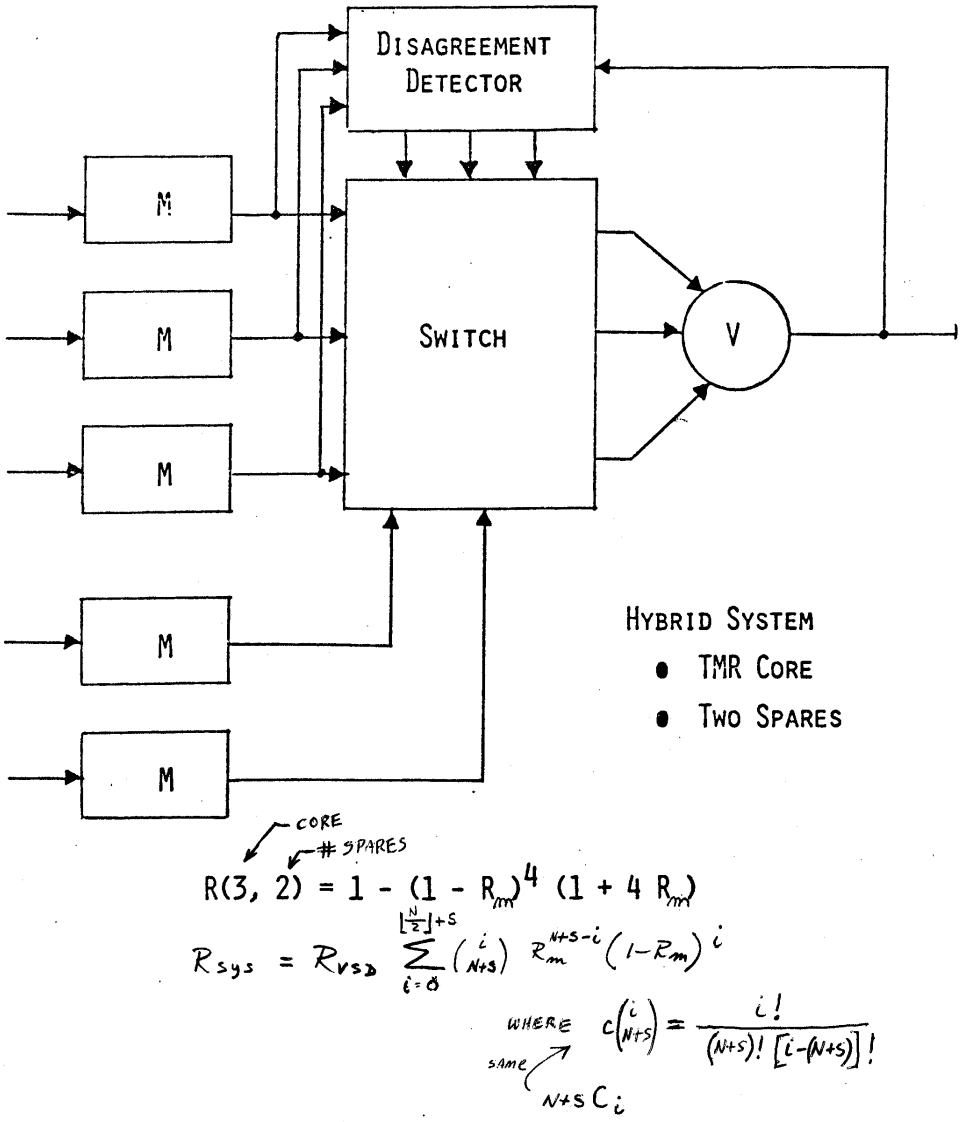



Comparing reliabilities for various configurations




Nomenclature: R(# of redundant modules, # of spares) thus R(3,2) is TMR with two spares, a hybrid configuration.

**Hybrid N-Module Redundancy** (using spares to augment/repair)

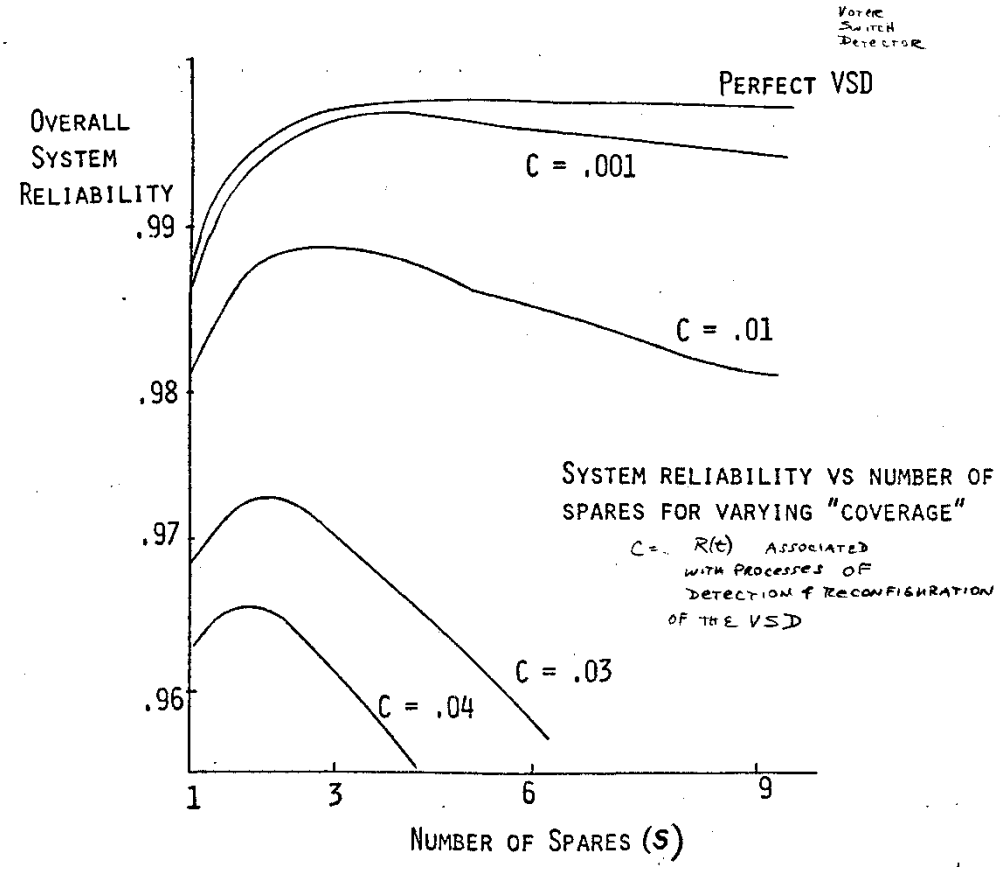



R(3,2) - TMR core with two spares and a Disagreement Detector shown above.

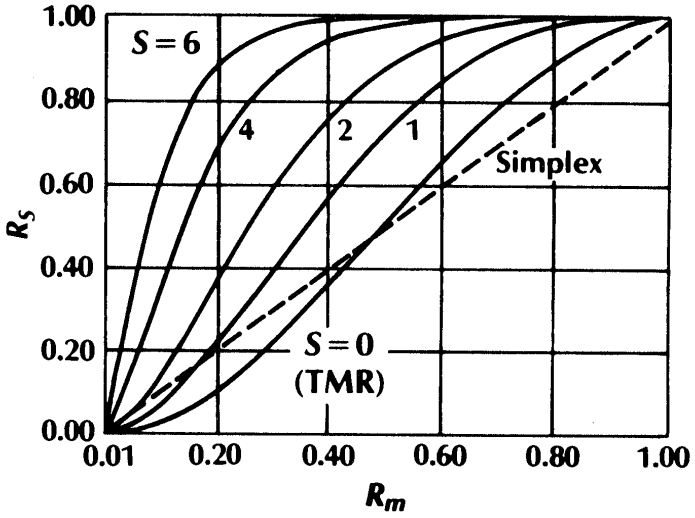
Disagreement Detector lifts up the covers on fault-tolerance showing the module that is failing the TMR votes. (NASA managers always wanted insight into what was failing).

Switch handles the 'insertion' of the spares as a means of repair (very quick if hot spares)

The number of spares can also have a non-intuitive result where analysis shows that more is not necessarily better. The reliability and more importantly the coverage of the voter-switch-detector (VSD) of the hybrid system can have a major impact with the conclusion that there is an optimum number of spares for specific configurations. One might also take into account the reliability improvements with the increasing cost as factor. (Experience shows that organizations willing to pay for 2X improvement but usually not any more.)




To show the effects of latent failures (where the standby unit has a failure when not being used) in conjunction with the specific module reliabilities $R_m$ Below is a graph showing $R_s$ = Overall System Reliability where S = number of spares with a 10% latent failure rate. (shows that one must use high reliability modules especially when S = 0, 1 or 2)




System with a Standby Failure Rate of 10% of the on-line (core) failure rate

## Section 4.5 Imperfect Voters

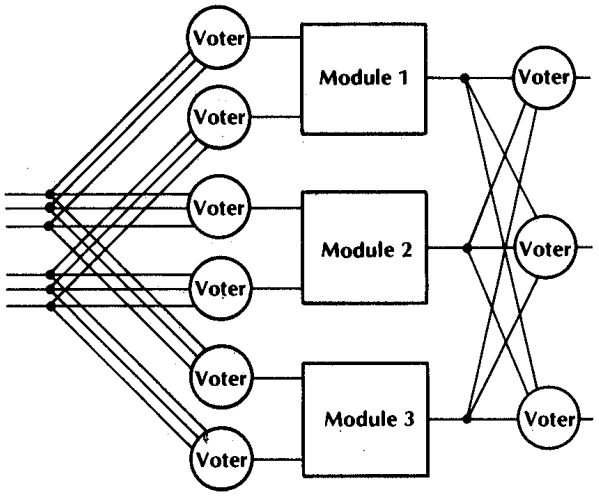
For an improvement in $\mathbf{R_{TMR}}$ where the voting scheme with the imperfect voter must be better than a single element's reliability ($R_m$), the minimum allowable reliability value of $p_v$ occurs when $R_m = 0.75$ which results in a minimum $p_v$ of 8/9 = 0.889. Thus if the voter reliability is below 0.889, a simplex circuit is more reliable for the same element/module reliability. Generally $p_v \sim 1$ since a voter is normally a simple comparator circuit and thus a voter has a negligible effect on the overall value of $R_{TMR}$

For n-module reliability

**TABLE 4.1 Minimum Voter Reliability**

| Number of redundant circuits, $2n + 1$ | 3 | 5 | 7 | 9 | 11 | ∞ |
|---|---|---|---|---|---|---|
| Minimum voter reliability, $p_v$ | 0.889 | 0.837 | 0.807 | 0.789 | 0.777 | 0.75 |

A more realistic TMR showing means of aligning the inputs (by voting here also) and minimizing the impact of the single voter's reliability limitation on the output of a $R_{TMR} = p_v(3\,R_m^2 - 2R_m^3)$ by using redundant voters that are cross strapped so they can make comparisons and vote.




The nominal way of implementing a redundant voter TMR scheme is

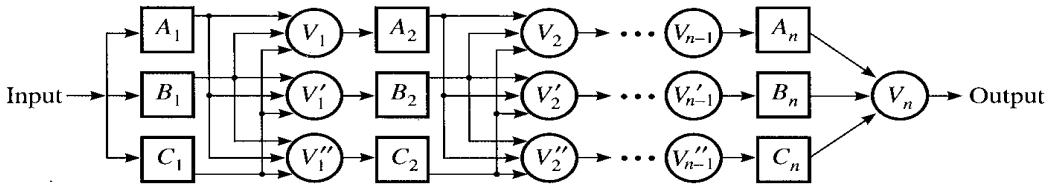



**Figure 4.8** A TMR circuit with redundant voters.

Note that in the last stage of voting, only a single voter can be employed. (Why?)

Errors do not propagate for more than one stage. If $B_1$ fails then $A_1 = C_1$ which masks the $B_1$ failure. If voter $V_1$'' fails, then $A_2$ and $B_2$ have the correct signal (but $C_2$ will be incorrect) The incorrect $C_2$ will be masked by the voting done by $V_2$ , $V_2$' and $V_2$''. Thus module failures don't propagate at all and voter failures only propagate by one stage.

The reliability of the redundant voter/TMR depends on the circuit implementation schemes (one big IC vs independent ICs and all of the various combinations for the module & voter circuits).

## Section 4.5.3 Modeling Limitations in Analytical Models

1. Transient Failures – failures can come and go at any point in time which is difficult to model.
2. Not all Failures are bad – sometimes a failed situation (stuck-at-1 or s-a-1) can be the correct output in the digital domain of 0's and 1's.
3. Bit by Bit voting – can isolate failures since voting at the word (multiple bit) level could result in non-resolvable failures.
4. All of the other items: common mode effects, optimistic/pessimistic modeling, etc.

## Section 4.6 Voter Logic

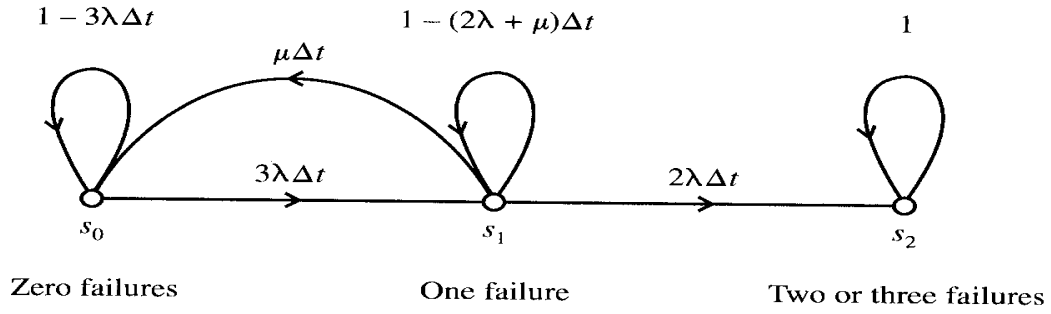
Author implements a majority voter in combinatorial logic along with an enhanced majority voter that will provide error detection (good review of Boolean algebra).

## Section 4.7 N-Modular Redundancy with Repair

Markov models that rely on Laplace transforms require the closed-form solution for the roots of an $n^{th}$ order polynomial which doesn't exist in a closed-form for $n \geq 5$. An example of this is in Appendix D5. Generally our ability to solve an $n^{th}$ order polynomial only exists for quadratic (n = 2) equations. Numerical solutions are always available for higher order equations but these provide little insight into the solution. Using simplifications and approximations discussed in Appendix B, Shooman suggests trying the following:

1. Initially represent features of a system for Markov modeling by low-order models that can be easily solved (closed-form solution).
2. Add the additional complicating effects one at a time to ascertain their effect on the model.
3. Assume that a comprehensive model will be solved numerically (using a computer and math software packages) which can then be compared to the simplifications made in the low-order model (at least to the point where one is satisfied that using the simplifications/approximations don't have a large effect/impact on the model).

**Section 4.7.3 TMR Reliability** (same model as the 'simplified' model on page 4)




State $s_2$ is a merged state for 2 or 3 failures. It is the absorbing or trapped state.

The author goes thru the TMR with repair solution on pages 166-167. For simplicity if we assume $\lambda = \mu = 1$ the we obtain

$$R_{TMR} = 1.366\, e^{1.268t} - 0.3661 e^{-4.7632t}$$

For t = 1 $R_{TMR}$ w/repair = 0.3841
without repair $R_{TMR}$ = 0.3064

which shows an improvement over no repair. For larger values of μ (repair rate), the improvement is even greater. Using the Taylor series expansion approximation (3 terms)

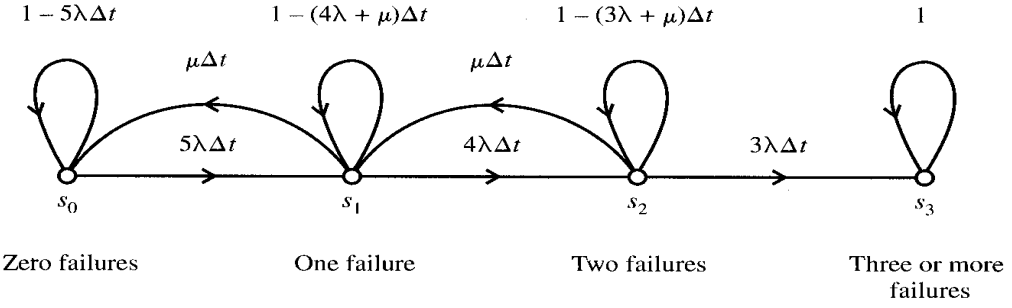
| | | |
|---|---|---|
| without repair | $R_{TMR\ 3\text{-}2} \cong 1 - 3\lambda^2t^2 + 5\lambda^3t^3$ | (4.27d) |
| with repair | $R_{TMR}(t) \cong 1 - 3\lambda^2t^2 + \lambda^2(5\lambda + \mu)t^3$ | (4.15 → 4.27e) |

thus repair adds an extra term $\lambda^2\mu t^3$ to the expansion for $R_{TMR}$ but this only affects the 3rd term (cubic). Thus for small t (initial/turn-on behavior), repair doesn't add much.
For example $\mu = 10\,\lambda$ and $t = 0.1/\lambda$ TMR w/o repair = 0.975 TMR w/repair = 0.985

**Section 4.7.4 *N*-Modular Reliability**

Systems beyond $N \geq 3$ are difficult and not used much in practice (fabrication yields, large increase in the # of test points but Shuttle N = 4). However to develop an analysis basis …

Markov Model for N = 5 (5-level majority voting) with repair (pg 170 → 173)




$$R_{5MR}(t) \cong 1 - 10\lambda^3t^3 + 2.5\lambda^3(12\lambda + 2\mu)t^4 \ldots\ldots\ldots \quad \text{Eq 4.39b}$$

for $\mu = 10\,\lambda$ and $t = 0.1/\lambda$ TMR w/o repair = 0.975 TMR w/repair = 0.985
5MR w/o repair = 0.993 5MR w/repair = 0.998
5MR reduces the ***unreliability*** by a factor of 7.5 (compare unreliability since $R_{5MR} \cong 1$)

**TABLE 4.7 Comparison of the Initial Behavior for Several Voting and Parallel Systems with Repair**

| System | Initial Reliability Equation, $\mu = 10\lambda$ | Value of $t$ at which $R = 0.999$ |
|---|---|---|
| TMR with repair | $1 - 3(\lambda t)^2 + 15(\lambda t)^3$ | $\frac{0.0192}{\lambda}$ |
| 5MR with repair | $1 - 10(\lambda t)^3 + 80(\lambda t)^4$ | $\frac{0.057}{\lambda}$ |
| Two parallel | $1 - (\lambda t)^2 + 4.33(\lambda t)^3$ | $\frac{0.034}{\lambda}$ |
| Two standby | $1 - 0.5(\lambda t)^2 + 2(\lambda t)^3$ | $\frac{0.045}{\lambda}$ |

For realistic comparisons with repair (for $R > 0.999$ with $\mu = 10\,\lambda$) then
$R_{TMR}$ ($t \leq 1{,}920$ hours) and for $R_{5MR}$ ($t \leq 5{,}700$ hours)

Parallel & Standby Systems: the high reliability regions are longer than TMR but less than 5MR

For N > 5 with so much complexity, the question of chip fabrication, additional test points, etc., raise doubt about improving reliability. In fact, a 5MR with two failed circuits is inferior to TMR.

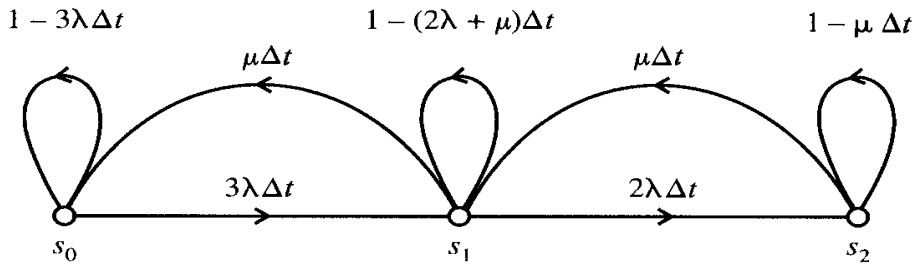
For coverage (C < 1) which impacted parallel and standby systems, it would be reasonable to believe that a voter's coverage in TMR would be better than a failure detector's coverage in a parallel/standby system. The author's analysis on pages 177 – 178, shows that a TMR voter's impact on reliability is 1/3 that of a coupler for the same decrease in reliability of the two systems (TMR versus a parallel system). Repair schemes only have an impact for large time intervals (t) and fast repair rates ($\mu$).

For a TMR system with element reliability $R_m = 0.9$ and $p_v = 0.99$
Single voter $R_{TMR} = 0.962$ Redundant voter $R_{TMR} = 0.9717$ although only a 1% increase, at these levels of reliability it is a significant increase. If the voter is even less reliable, the reliability gain with redundant voters is even greater.

The question of implementation which can be impacted by architecture modularity (fault containment regions/how to isolate failures, complexity, chip fabrication) might lead to more 'connections' which are bad news for reliability (anything mechanical is usually bad). Thus the various system architectures must be evaluated for comparisons of n-modular redundancy versus other systems.

**Section 4.9.2 Markov Availability Models** A(t) = probability of a system being up at any time t  A(t) > R(t)

Use Markov Models to avoid complexity of evaluating all the conditional probabilities. Note that two repairmen decouple the dependency of having to repair two failures during the same interval which is not the case for μ = 1 repairman, as follows:




*TMR Markov Availability Model with Repair* (μ = 1) *and Perfect Coverage*

Three states total; the first two states make up the Availability A(t) equation:
$S_0$ = all three modules working  $S_1$ = two modules working
For availability A(t) with one repair process → $S_2$ no longer a trapped state

The only meaningful comparison for A(t) are the steady-state equations:

| System | Eq. (4.50) | $\mu = \lambda$ | $\mu = 10\lambda$ | $\mu = 100\lambda$ |
|---|---|---|---|---|
| Two in parallel | $\frac{\mu(2\lambda + \mu)}{2\lambda^2 + 2\lambda\mu + \mu^2}$ | 0.6 | 0.984 | 0.9998 |
| Two standby | $\frac{\mu(\lambda + \mu)}{\lambda^2 + \lambda\mu + \mu^2}$ | 0.667 | 0.991 | 0.9999 |
| TMR | $\frac{\mu(3\lambda + \mu)}{6\lambda^2 + 3\lambda\mu + \mu^2}$ | 0.4 | 0.956 | 0.9994 |

Lower failure rates →

Steady-State Availability Comparisons

Repair: Two modules in (cold) Standby > two modules in Parallel > TMR

The μ = 100 λ column is probably the most realistic real-world comparison
*Note that $A(t_{ss})$ depends only on the ratio μ / λ*

Don't forget analysis assumptions:

1. No module failures while in Standby (no latent failures).
2. No consideration of the coupler/switch for the R(t) of the standby/parallel systems. The coupler/switch is more complex than the voter in a TMR system.
3. Perfect coverage, where again the detector in a standby/parallel system is more complex than a voter in a TMR system (a realistic failure detector means less than perfect coverage → lower reliability).

## Section 4.9.3 Decoupled Availability Models

For multi-module systems and repair processes, there is a dependency for the repairmen. If 1-of-n modules fail, a single repairman is dispatched to fix a failed module A but if module B fails during the repair of module A, then the repair of B is dependent on the repair of module A → P(AB) = P(A) P(B|A). Only having a single repairman, results in the second repair taking longer because the repairman is working on the first failure.

If multiple repairmen are introduced, then the repair processes are 'decoupled'. With two repairmen, the repair of module B is no longer dependent on the repair of module A. The dependent probabilities P(B|A) become independent.

Having multiple repair processes is somewhat unlikely (cost) but since $\mu >> \lambda$ the decoupled case is approached. The repairs are relatively fast (compared to $\lambda$) and there is only a small probability that module B will fail when module A is under repair

### Exact versus Approximate Equations

For constant failure and repair rates, steady-state availability APPROXIMATIONS are actually very close to exact values.

For a single element/module:

$$A_{ss} = \mu / (\lambda + \mu) = \text{uptime} / (\text{uptime} + \text{downtime})$$

For parallel system:

$$A_{ss} = \mu(2\lambda + \mu) / (\lambda + \mu)^2$$

For TMR:

$$A_{ss} = [\mu(\lambda + \mu)]^2 * [(3\lambda + \mu) / (\lambda + \mu)]$$

**TABLE 4.10 Comparison of the Exact and Approximate Steady-State Availability Equations for Various Systems**

| System | Exact, Eq. (4.50) | Approximate, Eqs. (4.54), (4.56), and (4.59) | Exact, $\mu = 100\lambda$ | Approximate, $\mu = 100\lambda$ |
|---|---|---|---|---|
| Two in parallel | $\frac{\mu(2\lambda + \mu)}{2\lambda^2 + 2\lambda\mu + \mu^2}$ | $\frac{\mu(2\lambda + \mu)}{(\lambda + \mu)^2}$ | 0.99980396 | 0.99990197 |
| Two standby | $\frac{\mu(\lambda + \mu)}{\lambda^2 + \lambda\mu + \mu^2}$ | $\left(\frac{\mu}{\lambda + \mu}\right)\left[1 - \ln\left(\frac{\mu}{\lambda + \mu}\right)\right]$ | 0.999901 | 0.999950823 |
| TMR | $\frac{\mu(3\lambda + \mu)}{6\lambda^2 + 3\lambda\mu + \mu^2}$ | $\left(\frac{\mu}{\lambda + \mu}\right)^2\left(\frac{3\lambda + \mu}{\lambda + \mu}\right)$ | 0.9994417815 | 0.999707852 |

The difference in the exact versus the approximate is very small. Gains in R(t) and A(t) do not come easy yet the small improvements can be meaningful in terms of loss-of-life, etc. Stratus A(t) = 0.9999905 Tandem A(t) = 0.9999960 (five 9's)

## Section 4.11 Advanced Voting Techniques

*N*-Modular redundancy requires voting with lockout such that a failed element/module is not used in the voting scheme once voted out n-number of times, e.g., lockout the failed module (possible repair? non-permanent failure?)

**Adjudicator Algorithms** – deals with voting situations when you may not have an absolute majority but you do have an indication on what modules are working and those that might not be working. A majority vote may fail but there may be agreement among some of the modules/elements. One mechanism used in the Boeing 747 Carousel Inertial Navigation System (n-modular with different elements to avoid common-mode faults) was to use a stored problem with a known answer as a test case for the different elements BIT).

**Adaptive Voting** – a weighted sum where each output is weighted by a coefficient, where the coefficient is the probability that the output is correct. These coefficients can be adjusted dynamically by keeping tabs of the system operation (number of agreements over time, number of transient failures, results of pre-canned test cases, etc.) This is a superior voting mechanism but it is highly dependent on the design and implementation schemes.

***Sometimes simpler is better.*** (and it costs less too!!)