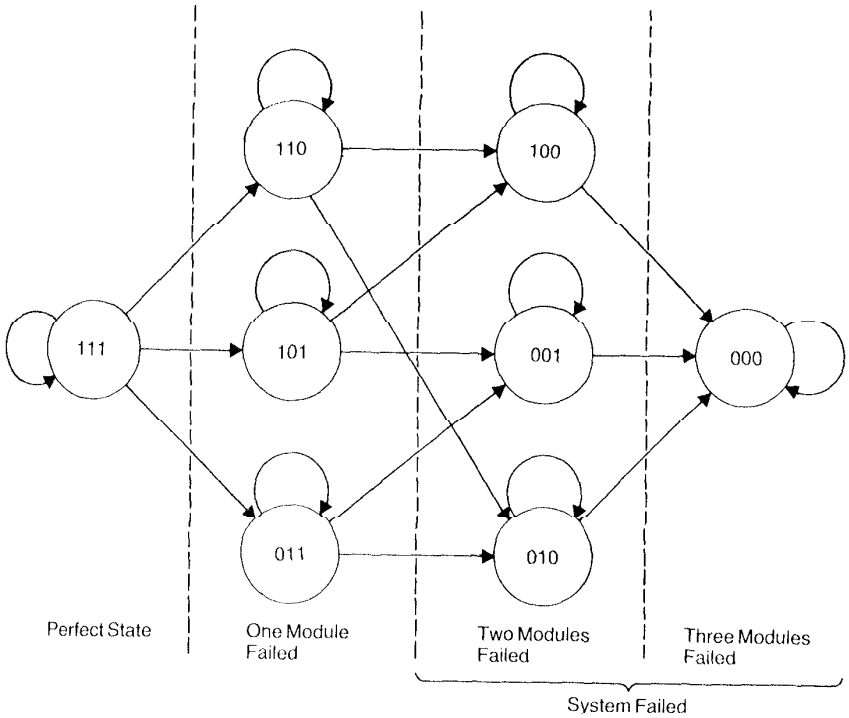

Fig. 4.16 State diagram showing possible states and state transitions for a TMR system.

The states in the diagram shown in Fig. 4.16 can be partitioned into three major categories: the *perfect* state (111) in which all modules function correctly; the *one-failed* states (110), (101), and (011) in which a single module has failed, and the *system-failed* states (100), (001), (010), and (000), in which enough modules have failed to cause the system to fail. State partitioning will be useful later when we attempt to reduce the Markov model.

As stated earlier, each state transition has associated with it a transition probability that describes the probability of that state transition occurring within a specified period of time. In the case of the TMR example that we have been considering, each transition represented in Fig. 4.16 is the result of a single module failure. If we assume that each module in the TMR system obeys the exponential failure law and has a constant failure rate of λ, the probability of a module being failed at some time $t + \Delta t$, given that the module was operational at time t, is given by

$$1 - e^{-\lambda \Delta t}$$

The exponential can be written in a series expansion as

$$e^{-\lambda \Delta t} = 1 + (-\lambda \Delta t) + \frac{(-\lambda \Delta t)^2}{2!} + \cdots$$

such that we have

$$1 - e^{-\lambda \Delta t} = 1 - \left[1 + (-\lambda \Delta t) + \frac{(\lambda \Delta t)^2}{2!} + \cdots\right] = (\lambda \Delta t) - \frac{(-\lambda \Delta t)^2}{2!} - \cdots$$

For small values of Δt, the expression reduces to simply

$$1 - e^{-\lambda \Delta t} \approx \lambda \Delta t$$

In other words, the probability that a module will fail within the time period Δt is approximately $\lambda \Delta t$.

Referring to our example on the TMR system, the transition probabilities can now be specified for each possible state transition. Figure 4.17

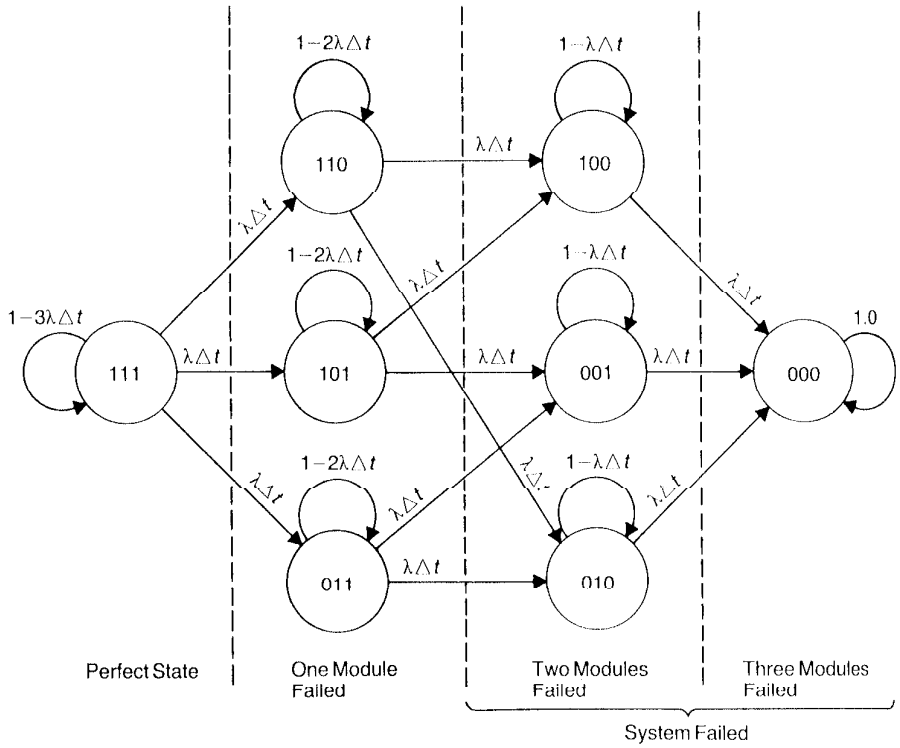

Fig. 4.17 Markov model of the TMR system showing possible states, state transition, and state transition probabilities.

shows the resulting state diagram for the Markov model of the TMR system, including the specification of each state transition probability. It is possible, however, to reduce the Markov model of Fig. 4.17. As mentioned earlier, the states of the TMR model can be partitioned into three major classes: the perfect state, the one-failed state, and the system-failed state. If we appropriately define the state transition probabilities, the several states within the TMR model can be combined.

Suppose that we let state 3 correspond to the state in which all three modules in the TMR system are functioning correctly; state 2 is the state in which two modules are working correctly; state F is the failed state in which two or more modules have failed. The resulting Markov model can be illustrated as shown in Fig. 4.18. The state transition probabilities shown in Fig. 4.18 have been derived to account for one of several failures occurring. For example, the probability of transitioning from state 3 to state 2 depends on the probability of any one of three modules failing. Consequently, the transition probability assigned to the transition from state 3 to state 2 is $3\lambda(\Delta t)$. Likewise the transition probability assigned to the state transition from state 2 to state F is $2\lambda(\Delta t)$.

The equations of the Markov model of the TMR system can be written easily from the state diagram shown in Fig. 4.18. The probability of the system being in any given state S at some time $t + \Delta t$ depends on the probability that the system was in a state from which it could transition to state S *and* the probability of that transition occurring. For example, the probability that the TMR system will be in state 3 at time $t + \Delta t$ depends on the probability that the system was in state 3 at time t (since the system can only transition to state 3 from state 3) and the probability of the system transitioning from state 3 back into state 3.

In mathematical terms, we have

$$p_3(t + \Delta t) = (1 - 3\lambda\,\Delta t)p_3(t)$$

where $p_3(t)$ is the probability of being in state 3 at time t and $p_3(t + \Delta t)$ is the probability of being in state 3 at time $t + \Delta t$. In a similar fashion, the equations for the remaining two states can be written as

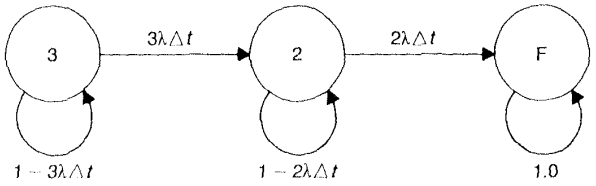

Fig. 4.18 Reduced Markov model of the TMR system with a minimal number of states.

$$p_2(t + \Delta t) = (3\lambda \Delta t)p_3(t) + (1 - 2\lambda \Delta t)p_2(t)$$
$$p_F(t + \Delta t) = (2\lambda \Delta t)p_2(t) + p_F(t)$$

assuming that the system will remain in the failed state if it ever enters the failed state. $p_2(t + \Delta t)$ is the probability of being in state 2 at time $t + \Delta t$; $p_2(t)$ is the probability of being in state 2 at time t; $p_F(t + \Delta t)$ is the probability of being in state F at time $t + \Delta t$; and $p_F(t)$ is the probability of being in state F at time t.

The equations of the Markov model of the TMR system can be written in matrix form as

$$\begin{bmatrix} p_3(t + \Delta t) \\ p_2(t + \Delta t) \\ p_F(t + \Delta t) \end{bmatrix} = \begin{bmatrix} (1 - 3\lambda \Delta t) & 0 & 0 \\ 3\lambda \Delta t & (1 - 2\lambda \Delta t) & 0 \\ 0 & 2\lambda \Delta t & 1 \end{bmatrix} \begin{bmatrix} p_3(t) \\ p_2(t) \\ p_F(t) \end{bmatrix}$$

The resulting matrix equation can be written in a condensed form as

$$\mathbf{P}(t + \Delta t) = \mathbf{A}\mathbf{P}(t)$$

where

$$\mathbf{P}(t + \Delta t) = \begin{bmatrix} p_3(t + \Delta t) \\ p_2(t + \Delta t) \\ p_F(t + \Delta t) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} (1 - 3\lambda\Delta t) & 0 & 0 \\ 3\lambda\Delta t & (1 - 2\lambda\Delta t) & 0 \\ 0 & 2\lambda\Delta t & 1 \end{bmatrix}$$

$$\mathbf{P}(t) = \begin{bmatrix} p_3(t) \\ p_2(t) \\ p_F(t) \end{bmatrix}$$

$\mathbf{P}(t)$ is the probability state vector at time t, $\mathbf{P}(t + \Delta t)$ is the probability state vector at time $t + \Delta t$, and $\mathbf{A}$ is the transition matrix.

The matrix equations for the Markov model can be viewed as a difference equation for the purpose of obtaining a solution. By assuming some initial value of the probability state vector, $\mathbf{P}(0)$, the value of $\mathbf{P}(\Delta t)$ can be obtained as $\mathbf{P}(\Delta t) = \mathbf{A}\mathbf{P}(0)$. Similarly, the value of the probability state vector at time $2\Delta t$ can be written as $\mathbf{P}(2\Delta t) = \mathbf{A}\mathbf{P}(\Delta t) = \mathbf{A}^2\mathbf{P}(0)$. In general, the solution is given as

$$\mathbf{P}(n\Delta t) = \mathbf{A}^n\mathbf{P}(0)$$

The probability of the system failing is given by the probability of the system being in the failed state. For example, in the TMR illustration, the probability of the system failing is the element of $\mathbf{P}(t)$ given by $p_F(t)$. The reliability of the TMR system can be written as

$$R_{\text{TMR}}(t) = 1 - p_F(t) = p_3(t) + p_2(t)$$

The Markov models considered thus far have been *discrete-time* models in which state transitions occur at fixed intervals Δt. It is possible to model systems using *continuous-time* Markov models in which state transitions can occur at any point in time [Nelson 1986]. The continuous-time equations can be derived from the discrete-time equations by allowing the time interval Δt to approach zero. For example, the equations of the discrete-time Markov model for the TMR system can be written as

$$\frac{p_3(t + \Delta t) - p_3(t)}{\Delta t} = -3\lambda p_3(t)$$

$$\frac{p_2(t + \Delta t) - p_2(t)}{\Delta t} = 3\lambda p_3(t) - 2\lambda p_2(t)$$

$$\frac{p_F(t + \Delta t) - p_F(t)}{\Delta t} = 2\lambda p_2(t)$$

through simple algebraic manipulations. Taking the limit as Δt approaches zero results in a set of differential equations given by

$$\frac{dp_3(t)}{dt} = -3\lambda p_3(t)$$

$$\frac{dp_2(t)}{dt} = 3\lambda p_3(t) - 2\lambda p_2(t)$$

$$\frac{dp_F(t)}{dt} = 2\lambda p_2(t)$$

The simultaneous differential equations can be solved using a number of techniques. For example, if Laplace transforms are used, we have

$$sP_3(s) - p_3(0) = -3\lambda P_3(s)$$

$$sP_2(s) - p_2(0) = 3\lambda P_3(s) - 2\lambda P_2(s)$$

$$sP_F(s) - p_F(0) = 2\lambda P_2(s)$$

where $P_3(s)$ is the Laplace transform of $p_3(t)$, $P_2(s)$ is the Laplace transform of $p_2(t)$, $P_F(s)$ is the Laplace transform of $p_F(t)$, $p_3(0)$ is the initial value of $p_3(t)$ at time $t = 0$, $p_2(0)$ is the initial value of $p_2(t)$ at time $t = 0$, and $p_F(0)$ is the initial value of $p_F(t)$ at time $t = 0$. We assume in the analysis, however, that the system starts in the perfect state at time $t = 0$, so $p_3(0) = 1$, $p_2(0) = 0$, and $p_F(0) = 0$. Consequently, the Laplace transform equations can be written as

$$P_3(s) = \frac{1}{s + 3\lambda}$$

$$P_2(s) = \frac{3\lambda}{(s + 2\lambda)(s + 3\lambda)}$$

$$P_F(s) = \frac{6\lambda^2}{s(s + 2\lambda)(s + 3\lambda)}$$

which can be rewritten as

$$P_3(s) = \frac{1}{s + 3\lambda}$$

$$P_2(s) = \frac{3}{(s + 2\lambda)} + \frac{-3}{(s + 3\lambda)}$$

$$P_F(s) = \frac{1}{s} + \frac{-3}{(s + 2\lambda)} + \frac{2}{(s + 3\lambda)}$$

Taking the inverse Laplace transform results in the solution given by

$$p_3(t) = e^{-3\lambda t}$$

$$p_2(t) = 3e^{-2\lambda t} - 3e^{-3\lambda t}$$

$$p_F(t) = 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t}$$

Recall that the reliability of the TMR system is the probability of being in either state 3 or state 2, so

$$R_{\text{TMR}}(t) = p_3(t) + p_2(t) = e^{-3\lambda t} + 3e^{-2\lambda t} - 3e^{-3\lambda t} = 3e^{-2\lambda t} - 2e^{-3\lambda t}$$

which is exactly the same result obtained using the combinatorial techniques. Also note that the sum of $p_3(t)$, $p_2(t)$, and $p_F(t)$ is 1, as expected.

It is interesting to verify that the computer solution of the discrete-time Markov model yields the same results as the equations from the combinatorial model and the continuous-time model. For example, suppose we consider the TMR system. The primary reason for using the TMR system as an example is the relative ease with which both the Markov and the combinatorial models of the TMR system can be constructed. Recall that the combinatorial model of the TMR system that obeys the exponential failure law produces the reliability function

$$R(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$$

Table 4.3 shows the values obtained from the combinatorial model of the TMR system at various points in time compared to the values obtained from the computer solution of the discrete-time Markov model of the TMR system. The Markov model was solved by assuming an initial state vector of $\mathbf{P}(0) = (100)$ and using a time step, Δt of 0.1 seconds. The failure rate λ has been chosen as 0.1 failures per hour. The differences that exist between the

TABLE 4.3 Comparison of results from computer solution of the discrete-time Markov and the combinatorial model for the TMR system

	Reliability	
Time (t) in minutes	Combinatorial results	Markov results
1	0.99999177	0.99999171
2	0.99996674	0.99996686
3	0.99992549	0.99992561
4	0.99986792	0.99986809
5	0.99979424	0.99979442
6	0.99970472	0.99970472
7	0.99959898	0.99959916
8	0.99947786	0.99947786
9	0.99934101	0.99934095
10	0.99918842	0.99918854

Failure rate λ is 0.1 failures per hour, and time step Δt is 0.1 seconds.

two sets of numbers are within the computational accuracy used to create the numbers.

We have seen how the Markov model can be used to model systems that do not depend on fault coverage or a repair process. Now we want to examine the process of developing a Markov model that depends on the coverage factor. After examining coverage, we will investigate the Markov model of a system with repair.

The system that we wish to model is a triply redundant system that uses fault detection techniques to detect the occurrence of a fault within one of the three independent modules. The modules provide their outputs to a flux-summer such that only one of the three modules must perform correctly for the system to function correctly. Consequently, the system can tolerate as many as two module failures provided that the failures are handled appropriately. The correct way for a module failure to be handled is for the affected unit to be removed from the flux-summing operation by opening a switch. As long as the switch is closed, the associated module provides current to the flux-summer. Once the switch is opened, however, the module is completely disconnected from the flux-summer and no longer affects the operation of the system. The probability that a failure will be correctly handled is the fault coverage and is denoted as C. The basic structure of the system to be analyzed is shown in Fig. 4.19.

The Markov model of this system is similar to that of the basic TMR system with majority voting. In fact, if the coverage factor C becomes zero,

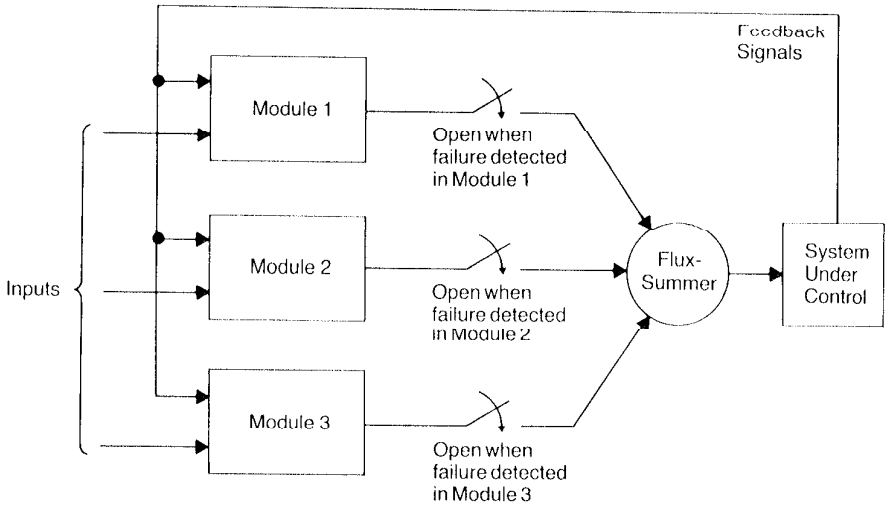

Fig. 4.19 An example hybrid redundancy technique to be used to illustrate the development of a Markov model that includes coverage.

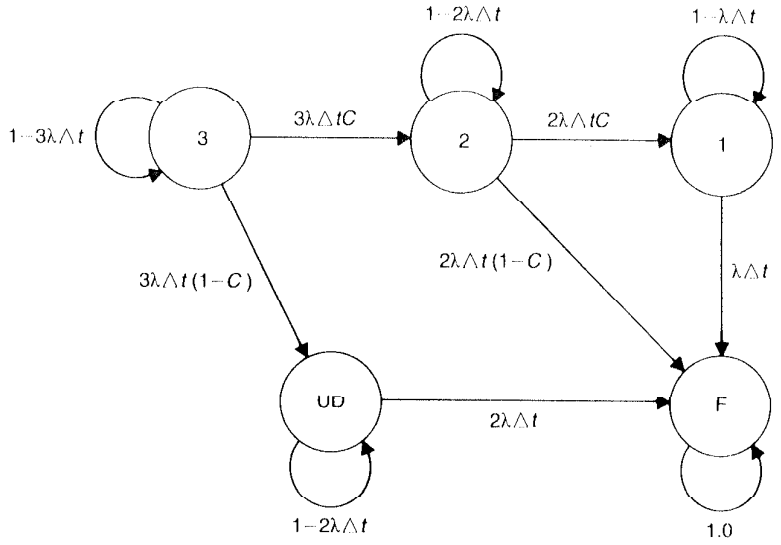

Fig. 4.20 Discrete-time Markov model of the system shown in Fig. 4.19.

the system reduces to the basic TMR system. The Markov model of the system is shown in Fig. 4.20. The system is assumed to begin in the fault-free state, which is labeled as state 3. There are two paths by which the system can exit state 3. The first is shown as a transition to state 2 and corresponds to the failure of one of the three modules and the correct handling of that failure. The second transition that can occur from state 3 is the transition to state *UD*, which corresponds to one of the three modules failing and the failure going undetected or being handled inappropriately. Once the system enters state *UD*, it becomes equivalent to the basic TMR system with majority voting; the system cannot tolerate the second failure if the first failure is not handled correctly. The same types of transitions exist from state 2 to state 1 and the failed state. State 2 corresponds to the system having had one module to fail and having handled that failure correctly. While in state 2, the system can tolerate a failure and transition to state 1 provided that the failure is detected and handled correctly. Any undetected failures, however, take the system from state 2 to the failed state. Finally, any failures that occur while the system is in state 1 cause the system to transition immediately to the failed state.

While in any state, there is a nonzero probability that the system will remain in that particular state. For example, if the system is currently in state 3, the system remains in state 3 as long as a failure does not occur. Likewise, the system remains in states 2, 1, and *UD* if the system is presently in those states and new failures do not occur. The probability of being in the same state at the end of a Δt time period as at the beginning of that time period is calculated as $1 - p_{\text{exit}}(\Delta t)$, where $p_{\text{exit}}(\Delta t)$ is the probability of exiting a state during the Δt time period. For example, the probability of exiting state 3 during a Δt time period is the probability that any one of the three modules will fail. In other terms, the probability of exiting state 3 is $3\lambda\,\Delta t$. The probability of not exiting state 3 is, therefore, $1 - 3\lambda\,\Delta t$.

The equations of the Markov model for the system of Fig. 4.19 are developed as they were for the basic TMR system. The probability of being in state i at time $t + \Delta t$ depends on: (1) the probability of being in a state at time t from which the system can transition to state i, and (2) the associated transition probabilities. For example, the system can go to state 2 during a Δt time period if and only if it is in either state 3 or state 2. Therefore, the probability of being in state 2 at time $t + \Delta t$ is

$$p_2(t + \Delta t) = 3\lambda\,\Delta t C p_3(t) + (1 - 2\lambda\,\Delta t)p_2(t)$$

The complete set of equations for the Markov model of the system of Fig. 4.19 can be written as

TABLE 4.4 Reliability as a function of fault coverage for the system modeled using the Markov model of Fig. 4.20.

Fault coverage	Reliability (after 1 hour)
0.0	0.97460
0.1	0.97484
0.2	0.97558
0.3	0.97680
0.4	0.97852
0.5	0.98073
0.6	0.98343
0.7	0.98662
0.8	0.99030
0.9	0.99448
1.0	0.99914

Failure rate λ is 0.1 failures per hour, and time step Δt is 0.1 seconds.

$$\begin{bmatrix} p_3(t+\Delta t) \\ p_2(t+\Delta t) \\ p_1(t+\Delta t) \\ p_{UD}(t+\Delta t) \\ p_F(t+\Delta t) \end{bmatrix} = \begin{bmatrix} 1-3\lambda\Delta t & 0 & 0 & 0 & 0 \\ 3\lambda\Delta t C & 1-2\lambda\Delta t & 0 & 0 & 0 \\ 0 & 2\lambda\Delta t C & 1-\lambda\Delta t & 0 & 0 \\ 3\lambda\Delta t(1-C) & 0 & 0 & 1-2\lambda\Delta t & 0 \\ 0 & 2\lambda\Delta t(1-C) & \lambda\Delta t & 2\lambda\Delta t & 1 \end{bmatrix} \begin{bmatrix} p_3(t) \\ p_2(t) \\ p_1(t) \\ p_{UD}(t) \\ p_F(t) \end{bmatrix}$$

The reliability of the system described by the Markov model of Fig. 4.20 is the probability of being in states 3, 2, 1, or *UD*. In other words, the reliability can be written as

$$R(t) = p_3(t) + p_2(t) + p_1(t) + p_{UD}(t)$$

It is interesting to note the effect that fault coverage has on the reliability of this system. Table 4.4, for example, shows the reliability of the system after one hour as a function of the coverage factor. The time step Δt has been selected as 0.1 seconds and the failure rate λ has been chosen as 0.1 failures per hour. For perfect coverage, the system achieves a reliability of approximately 0.99914, which is the same reliability that can be achieved by the perfect parallel system with three identical modules. When the cov-

erage is zero, the system is identical to the basic TMR system with majority voting. In other words, the system will be a 2-of-3 system where two of the three modules must work for the system to work. The reliability achieved by this system when the coverage is zero is approximately 0.9746. Note in Table 4.4 that the impact of changes in the fault coverage is more significant at the higher values of fault coverage.

We now consider systems that incorporate repair as a form of recovery. For example, many applications require that the repair rate's effect on a system be modeled. A system that possesses a poor repair rate can be required to have fault tolerance to the extent that the system can function while elements are being repaired. The Markov model is an extremely useful tool for analyzing the effect that repair has on a system.

Consider the Markov model of a simple system consisting of one computer and no redundancy. The single computer might be a banking system, for example, and we wish to model the failure and recovery process of this single computer. Further assume that the computer has a constant failure rate λ and a constant repair rate μ. During the time interval Δt, the computer will have a probability of failure given by $\lambda \Delta t$. Since the repair rate is analogous to the failure rate and represents the number of repairs that are expected to occur in a specific time period, the probability of a repair occurring within the time period Δt is $\mu \Delta t$. Using this information allows us to formulate the simple Markov model shown in Fig. 4.21 for the computer. State O represents the condition of the computer being completely operational, and state F represents the failed condition. If the computer system is in state O, the probability of the system transitioning to state F during the time period Δt is $\lambda \Delta t$. Likewise, if the system is in state F, the probability of transitioning to state O is $\mu \Delta t$. As we discovered during previous examples, if the system is in state O and a failure does not occur, the system remains in state O. Similarly, if the system is in state F and a repair does not occur, the system remains in state F.

The equations for the Markov model of Fig. 4.21 can be written as

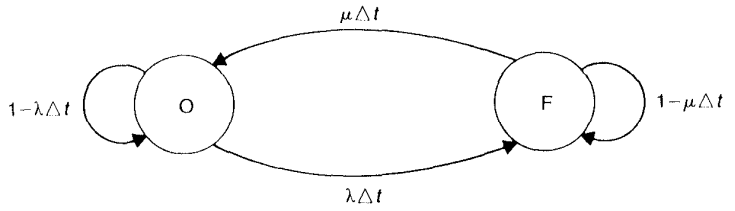

Fig. 4.21 Markov model of a simple nonredundant system with repair.

$$\begin{bmatrix} p_O(t + \Delta t) \\ p_F(t + \Delta t) \end{bmatrix} = \begin{bmatrix} 1 - \lambda\Delta t & \mu\Delta t \\ \lambda\Delta t & 1 - \mu\Delta t \end{bmatrix} \begin{bmatrix} p_O(t) \\ p_F(t) \end{bmatrix}$$

It is interesting to solve this Markov model to determine the effect that the repair rate has on the probability of the system being operational. Figure 4.22 shows the plot of the probability of the system being in state *O* versus the repair rate. The failure rate λ was selected as 0.1 failures per hour, and the time step Δt was chosen as 0.1 seconds for this example. The system was assumed to start in the operational state.

It is also instructive to determine the continuous-time solution for the model shown in Fig. 4.21. Using a procedure identical to that used to determine the continuous-time equations for the TMR system, the equations from the discrete-time Markov model are manipulated algebraically to obtain

$$\frac{p_O(t + \Delta t) - p_O(t)}{\Delta t} = -\lambda p_O(t) + \mu p_F(t)$$

$$\frac{p_F(t + \Delta t) - p_F(t)}{\Delta t} = \lambda p_O(t) - \mu p_F(t)$$

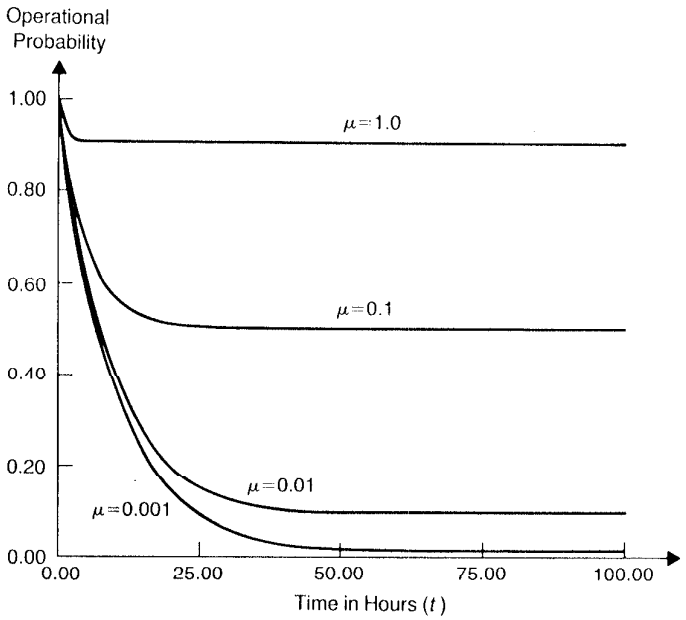

Fig. 4.22 Probability of remaining operational for the system described by the Markov model of Fig. 4.21.

Taking the limit as Δt approaches zero results in the differential equations given by

$$\frac{dp_O(t)}{dt} = -\lambda p_O(t) + \mu p_F(t)$$

$$\frac{dp_F(t)}{dt} = \lambda p_O(t) - \mu p_F(t)$$

Assuming that the initial conditions are $p_O(0) = 1$ and $p_F(0) = 0$ and using Laplace transforms results in the simultaneous equations

$$sP_O(s) = 1 - \lambda P_O(s) + \mu P_F(s)$$

$$sP_F(s) = \lambda P_O(s) - \mu P_F(s)$$

Solving the simultaneous equations for $P_O(s)$ and $P_F(s)$ yields

$$P_O(s) = \frac{1}{s + (\lambda + \mu)} + \frac{\mu}{s(s + (\lambda + \mu))}$$

$$P_F(s) = \frac{\lambda}{s(s + (\lambda + \mu))}$$

which can be rewritten as

$$P_O(s) = \frac{\frac{\mu}{\lambda + \mu}}{s} + \frac{\frac{\lambda}{\lambda + \mu}}{s + (\lambda + \mu)}$$

$$P_F(s) = \frac{\frac{\lambda}{\lambda + \mu}}{s} - \frac{\frac{\lambda}{\lambda + \mu}}{s + (\lambda + \mu)}$$

Taking the inverse Laplace transform results in the time-domain solution given by

$$p_O(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

$$p_F(t) = \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

Several interesting features are apparent in the time-domain expressions for $p_O(t)$ and $p_F(t)$. For example, as time approaches infinity, $p_O(t)$ approaches the constant value of

$$p_O(\infty) = \frac{\mu}{\lambda + \mu} = \frac{1}{\frac{\lambda}{\mu} + 1}$$

and $p_F(t)$ approaches

$$p_F(\infty) = \frac{\lambda}{\lambda + \mu} = \frac{1}{\frac{\mu}{\lambda} + 1}$$

As we will discover when discussing availability modeling, the value of $p_O(t)$ as time approaches infinity is the steady-state availability.

4.4 Safety Modeling

The safety of a system, as defined in Chapter 1, is the probability that the system will *either* perform correctly *or* will fail in a safe manner. The concepts of *safe* and *unsafe* are highly dependent upon the application. In many cases, for example, a safe course of action is to simply turn the system off after a failure occurs. In some applications, however, turning the system off can be a disastrous course of action. In any case, however, the fundamental concept of safety analysis is that a system will possess two different ways in which it can fail: one system failure is defined as *safe*, and the other is categorized as *unsafe*. The definition of safe and unsafe failures must be created uniquely for each application.

Safety can be modeled using Markov models by splitting the system failed state into two separate states. One failed state is normally labeled *FS* for *failed safe*, and the other failed state is labeled as FU for *failed unsafe*. A Markov model for a simple system containing one hardware module with a failure rate of λ and self-diagnostics with a fault detection coverage of C is shown in Fig. 4.23. Safe failures are defined in this example as those that are detected by the self-diagnostics. Consequently, unsafe failures are defined as those that are not detected by the self-diagnostics. If a failure occurs, the system transitions to either state *FS* or *FU* depending on whether or not the condition is detected.

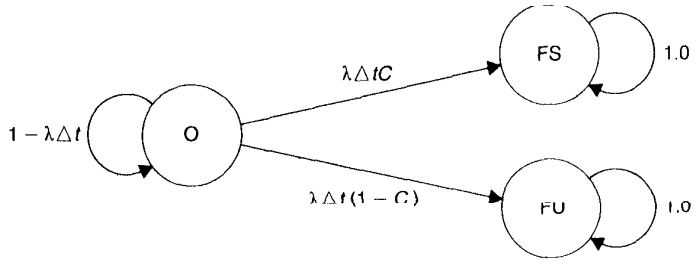

Fig. 4.23 Three-state Markov model allowing safety calculations.

The safety of the system described by the Markov model of Fig. 4.23 can be written as

$$S(t) = p_O(t) + p_{FS}(t)$$

where $S(t)$ is the safety, $p_O(t)$ is the probability of being in the operational state at time t, and $p_{FS}(t)$ is the probability of being in the failed safe state at time t. The complete equations of the discrete time Markov model can be written as

$$p_O(t + \Delta t) = (1 - \lambda \Delta t)p_O(t)$$
$$p_{FS}(t + \Delta t) = \lambda \Delta t C p_O(t) + p_{FS}(t)$$
$$p_{FU}(t + \Delta t) = \lambda \Delta t(1 - C)p_O(t) + p_{FU}(t)$$

As we have done previously, the differential equations of the corresponding continuous-time Markov model can be written as

$$\frac{dp_O(t)}{dt} = -\lambda p_O(t)$$

$$\frac{dp_{FS}(t)}{dt} = \lambda C p_O(t)$$

$$\frac{dp_{FU}(t)}{dt} = \lambda(1 - C)p_O(t)$$

Taking the Laplace transform results in

$$P_O(s) = \frac{p_O(0)}{s + \lambda}$$

$$P_{FS}(s) = \frac{\lambda C p_O(0)}{s(s + \lambda)} + \frac{p_{FS}(0)}{s}$$

$$P_{FU}(s) = \frac{\lambda(1 - C)p_O(0)}{s(s + \lambda)} + \frac{p_{FU}(0)}{s}$$

where $p_O(0)$, $p_{FS}(0)$, and $p_{FU}(0)$ are the intial values of the respective state probabilities. If we assume that the system begins in state O such that $p_O(0) = 1$, $p_{FS}(0) = 0$, and $p_{FU}(0) = 0$, we obtain

$$P_O(s) = \frac{1}{s + \lambda}$$

$$P_{FS}(s) = \frac{\lambda C}{s(s + \lambda)} = \frac{C}{s} - \frac{C}{s + \lambda}$$

$$P_{FU}(s) = \frac{\lambda(1 - C)}{s(s + \lambda)} = \frac{(1 - C)}{s} - \frac{(1 - C)}{s + \lambda}$$

The time domain solutions can now be written as

$$p_O(t) = e^{-\lambda t}$$
$$p_{FS}(t) = C - Ce^{-\lambda t}$$
$$p_{FU}(t) = (1 - C) - (1 - C)e^{-\lambda t}$$

Intuitively, the equations are as expected. For example, the reliability of the system is

$$R(t) = p_O(t) = e^{-\lambda t}$$

and the probability of being in one of the two failed states is

$$p_{FS}(t) + p_{FU}(t) = 1 - e^{-\lambda t} = 1 - R(t)$$

The safety of the system is written as

$$S(t) = p_O(t) + p_{FS}(t) = C + (1 - C)e^{-\lambda t}$$

At time $t = 0$, the safety of the system is 1, as expected. As time approaches infinity, however, the safety approaches

$$S(\infty) = C$$

In other words, if the fault detection coverage is perfect ($C = 1$), the system has perfect safety. However, if the fault detection coverage is nonexistent ($C = 0$), the system will eventually fail in an unsafe manner. The safety of the system, in this example, is directly dependent upon the fault detection coverage. In subsequent sections, we investigate the safety of more complex systems.

4.5 System Comparisons

Now that we have several tools at our disposal, we can begin to examine the process of comparing two or more systems. When we make comparisons, we must be careful about the parameters that we choose to compare. For example, if we choose to compare the MTTF of two systems or the reliability of the systems, the results can be surprising. Suppose that we wish to compare a simplex system consisting of a single computer to a TMR system with three computers in a majority voting arrangement. Assume for simplicity that the majority voter is perfect. The two systems are shown in Fig. 4.24. The computers in each system are identical and are assumed to obey the exponential failure law.

Recall that the MTTF of a system is defined as

$$\text{MTTF} = \int_0^\infty R(t)\,dt$$

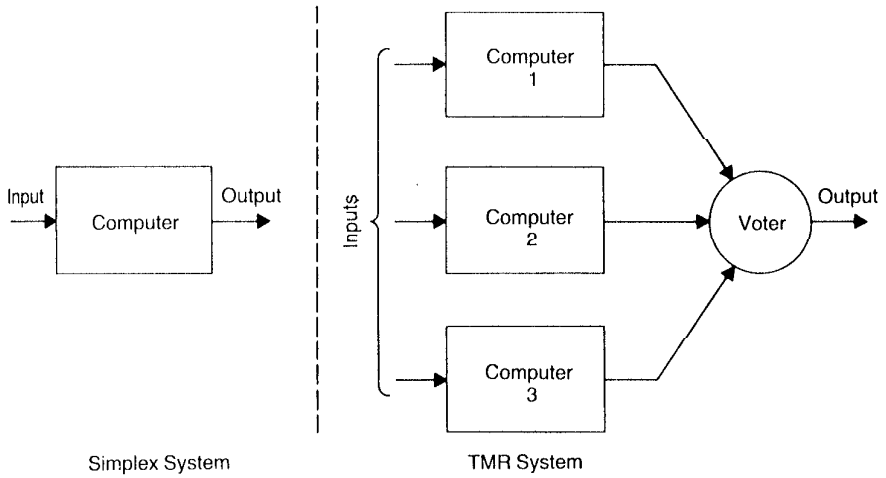

Fig. 4.24 Simplex and TMR systems to be compared to assess the benefits of redundancy.

where $R(t)$ is the reliability function of the system. The reliability function of the simplex system is

$$R_{\text{simplex}}(t) = e^{-\lambda t}$$

whereas that of the TMR system is

$$R_{\text{TMR}}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$$

If we integrate $R_{\text{simplex}}(t)$ and $R_{\text{TMR}}(t)$, we find that the MTTF of each system is given by

$$\text{MTTF}_{\text{simplex}} = \int_0^\infty e^{-\lambda t}\,dt = \frac{1}{\lambda}$$

$$\text{MTTF}_{\text{TMR}} = \int_0^\infty (3e^{-2\lambda t} - 2e^{-3\lambda t})\,dt = \frac{5}{6\lambda}$$

Thus, the MTTF of the TMR system is lower than the MTTF of the simplex system.

Based on these calculations, we might conclude that the TMR system is not as good as the simplex system. This may or may not be a correct conclusion depending on the application and the length of time the system is expected to operate correctly. Figure 4.25 provides a good insight into the reason why the MTTF of the TMR system is less than that of the simplex system. Figure 4.25 shows the reliability of the simplex and the TMR sys-

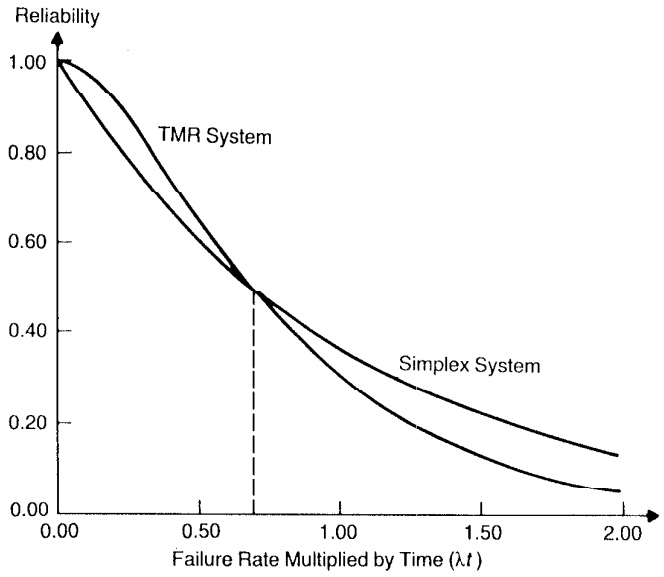

Fig. 4.25 Simplex and TMR reliabilities versus λt—a crossover point occurs where the two reliabilities are equal.

tems as functions of λt. As can be seen, a point in time is reached when the TMR system becomes less reliable than the simplex system. The MTTF is the area under the reliability curve, and that area is larger for the simplex than for the TMR configuration. The point of this discussion is that the MTTF can sometimes misrepresent the quality of a system. For certain values of the product λt, the TMR system's reliability will be superior to that of the simplex system. Regardless of the reliability, if fault tolerance is necessary, the TMR system will be superior to the simplex system.

Sometimes, a single parameter comparison that is better than the MTTF is the **mission time**, denoted as MT[r]. The mission time is the time at which the reliability of a system falls below the level r. For example, a simplex system that obeys the exponential failure law has a reliability of r when

$$r = e^{-\lambda t}$$

The time at which the reliability value of r occurs can be found by taking the natural logarithm of both sides of the preceding equation and solving the resulting equation. The solution yields

$$\text{MT}[r] = \frac{-\ln(r)}{\lambda}$$

A **mission time improvement** can be calculated as the ratio of the mission times of the two systems being compared. For example, suppose that we wish to compare two computer systems: (1) a simplex computer system with a single computer that has a failure rate of 0.01 failures per hour and (2) a TMR system constructed using three of that same computer. The computers are assumed to obey the exponential failure law. We wish to determine the mission time improvement of the TMR system over the simplex system for a reliability of 0.86.

The $MT_{simplex}[0.86]$ is fairly easy to calculate from the exponential reliability function as

$$MT_{simplex}[0.86] = \frac{-\ln(r)}{\lambda} = \frac{-\ln(0.86)}{0.01} = 15.08 \quad \text{hours}$$

The $MT_{TMR}[0.86]$ is found from the solution of the equation

$$R_{TMR}(t) = 3e^{-2\times 0.01t} - 2e^{-3\times 0.01t} = 0.86$$

for the time t, which results in $MT_{TMR}[0.86] = 27$ hours. The mission time improvement of the TMR system over the simplex system is approximately 1.8. In other words, the TMR system, in this example, can operate 1.8 times as long as the simplex system while still maintaining a reliability of greater than 0.86. The graph shown in Fig. 4.26 illustrates the concepts of the mission time and the mission time improvement for the comparison of the simplex and the TMR systems with the failure rate of 0.01 failures per hour.

4.6 Availability Models

Thus far, we have considered only the modeling of the reliability of a system. However, we have seen in the discussions of Chapters 1 and 2 that parameters such as availability and maintainability are also important in the analysis of fault-tolerant systems. Many computer companies are concerned more with the probability of their systems being available when their customers want to use them (availability) rather than with the length of time the system can operate without failure (related to reliability). As a result, the rate at which a system can be repaired becomes a critical part of the design. The repair rate can dramatically affect the availability of a system.

Recall that the availability $A(t)$ of a system is defined as the probability that a system will be available to perform its tasks at the instant of time t. Intuitively, we can see that the availability can be approximated as the total time that a system has been operational divided by the total time elapsed since the system was initially placed into operation. In other words, the availability is the percentage of time that the system is available to per-

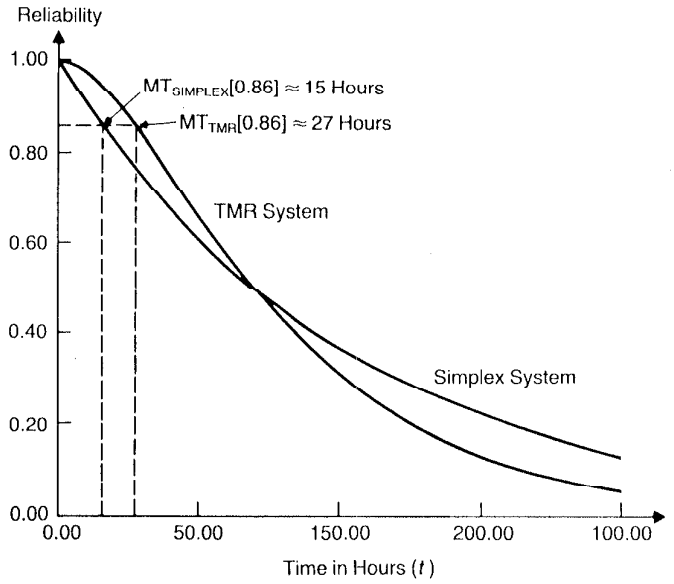

Fig. 4.26 Mission time improvement comparison between simplex and TMR.

form its expected tasks. Suppose that we place a system into operation at time $t = 0$. As time moves along, the system performs its functions, perhaps fails, and is repaired. At some time $t = t_{\text{current}}$, suppose that the system has operated correctly for a total of t_{op} hours and has been in the process of repair or waiting for repair to begin for a total of t_{repair} hours. The time t_{current} is then the sum of t_{op} and t_{repair}. The availability can be determined as

$$A(t_{\text{current}}) = \frac{t_{\text{op}}}{t_{\text{op}} + t_{\text{repair}}}$$

where $A(t_{\text{current}})$ is the availability at time t_{current}.

The preceding expression lends itself well to the experimental evaluation of the availability of a system; we can simply place the system into operation and measure the appropriate times required to calculate the availability of the system at a number of points in time. Unfortunately, the experimental evaluation of the availability is often not possible because of the time and expense involved. Also, we would like to have some means of estimating the availability before we actually build the system so that availability considerations can be factored into the design process. We will consider two approaches. The first is based on the single parameter measures such as MTTF and MTTR and yields what is typically called the

steady-state availability A_{ss}. The second approach uses the failure rates and the repair rates in a Markov model to calculate the availability as a function of time.

We have seen that availability is basically the percentage of time that a system is operational. Using knowledge of the statistical interpretation of the MTTF and the MTTR, we expect that, on the average, a system will operate for MTTF hours and then encounter its first failure. Once the failure has occurred, the system will then, again on the average, require MTTR hours to be repaired and placed into operation once again. The system will then operate for another MTTF hours before encountering its second failure. This concept has been illustrated in Fig. 4.2.

If the average system experiences N failures during its lifetime, the total time that the system is operational is N(MTTF) hours. Likewise, the total time that the system is "down" for repairs is N(MTTR) hours. In other words, the operational time t_{op} is N(MTTF) hours and the down-time t_{repair} is N(MTTR) hours. The average, or steady-state, availability is

$$A_{ss} = \frac{N(\text{MTTF})}{N(\text{MTTF}) + N(\text{MTTR})} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

We know, however, that the MTTF and the MTTR of a simplex system are related to the failure rate and the repair rate, respectively, as

$$\text{MTTF} = \frac{1}{\lambda}$$

$$\text{MTTR} = \frac{1}{\mu}$$

Therefore, the steady-state availability of a simplex system is given by

$$A_{ss} = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} = \frac{1}{1 + \frac{\lambda}{\mu}}$$

Recall that the repair rate μ is expressed in repairs per hour, whereas the failure rate λ is in failures per hour. We would expect that if the failure rate goes to 0, implying that the system never fails, or the repair rate goes to infinity, implying that no time is required to repair the system, the availability will go to 1. Looking at the expression for the steady-state availability, we can see that this is true.

As an example calculation, consider a computer system that has a failure rate of one failure every 100 hours and a repair rate of one repair every 10 hours. The failure rate of this system is $\lambda = 0.01$ failures per hour and the

repair rate is $\mu = 0.1$ repairs per hour. The steady-state availability is calculated as

$$A_{ss} = \frac{1}{1 + \dfrac{0.01}{0.1}} = 0.90909$$

This implies that the system is available for use an average of slightly more than 90% of the time.

Now suppose we investigate the use of the Markov model as a means of determining the availability of a system. We already have the necessary tools to accomplish this task. The Markov model of a system with repair is in fact the model required to calculate the availability of a system. Recall the two-state model of a simple system with repair and having a failure rate of λ and a repair rate of μ. The state diagram of this model is repeated in Fig. 4.27 for convenience. State O represents the state in which the system is completely operational, whereas state F is the state in which the system has failed and is in the process of being repaired. The probability of the system failing during the time interval Δt is given by $\lambda \Delta t$, whereas the probability of the system being repaired during the time interval Δt is $\mu \Delta t$.

The equations of the Markov model are given by

$$\begin{bmatrix} p_O(t + \Delta t) \\ p_F(t + \Delta t) \end{bmatrix} = \begin{bmatrix} 1 - \lambda \Delta t & \mu \Delta t \\ \lambda \Delta t & 1 - \mu \Delta t \end{bmatrix} \begin{bmatrix} p_O(t) \\ p_f(t) \end{bmatrix}$$

where $p_O(t)$ is the probability, at time t, that the system is in the operational state and is, therefore, available to perform its tasks. Consequently, $p_O(t)$ is the availability of the system.

As an example, the Markov model shown in Fig. 4.27 has been solved for the failure rate of $\lambda = 0.01$ failures per hour and the repair rate of $\mu = 0.1$ repairs per hour. The plot of the resulting availability is shown in Fig. 4.28. Note that the availability approaches the value of 0.90909 that was previ-

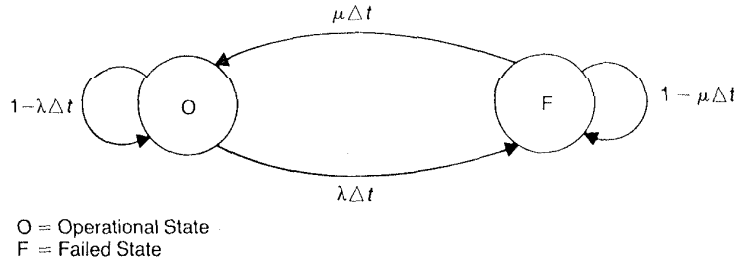

Fig. 4.27 Markov model of a simple nonredundant system with repair.

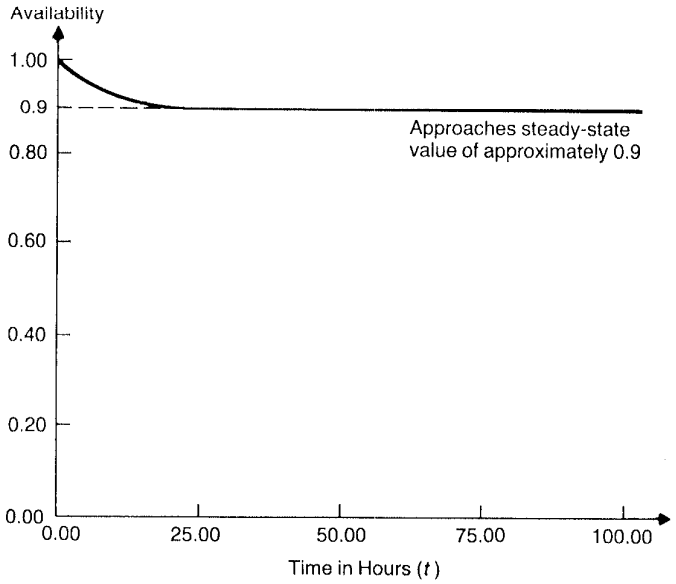

Fig. 4.28 Availability versus time for a simple nonredundant system with $\lambda = 0.01$ failures per hour and $\mu = 0.1$ repairs per hour.

ously determined as the steady-state availability of a system with this particular failure rate and repair rate.

4.7 Maintainability Models

As defined in Chapter 1, the maintainability is the probability that a failed system will be restored to working order within a specified time. We will use the notation that $M(t)$ is the maintainability for time t. In other words, $M(t)$ is the probability that a system will be repaired in a time less than or equal to t.

An important parameter in maintainability modeling is the repair rate μ. The repair rate is the average number of repairs that can be performed per time unit. The inverse of the repair rate is the MTTR, which is the average time required to perform a single repair. Mathematically, the relationship between the repair rate and the MTTR is given by

$$\text{MTTR} = \frac{1}{\mu}$$

In industry, the MTTR, and as a result μ, is usually derived in an experimental fashion. A system can be constructed and faults injected; the average time required to repair the system is measured and recorded as the MTTR. A good estimate of the MTTR can be obtained only if a sufficient number of different faults are injected and repair personnel with a variety of skill levels are used.

An expression for the maintainability of a system can be derived in a manner similar to that used to develop the exponential failure law for the reliability function. Suppose that we have N systems. We inject one unique fault into each of the systems, and we allow one maintenance person to repair each system. We begin this experiment by injecting the faults into the systems at time $t = 0$. Later, at some time t, we determine that $N_r(t)$ of the systems have been repaired and $N_{nr}(t)$ have not been repaired. Since the maintainability of a system at time t is the probability that the system can be repaired by time t, an estimate of the maintainability can be computed as

$$M(t) = \frac{N_r(t)}{N} = \frac{N_r(t)}{N_r(t) + N_{nr}(t)}$$

If we differentiate $M(t)$ with respect to time, we obtain

$$\frac{dM(t)}{dt} = \frac{1}{N}\frac{dN_r(t)}{dt}$$

which can also be written as

$$\frac{dN_r(t)}{dt} = N\frac{dM(t)}{dt}$$

The derivative of $N_r(t)$ is simply the rate at which components are repaired at the instant of time t.

At time t, we have $N_{nr}(t)$ systems that have not been repaired. If we divide $dN_r(t)/dt$ by $N_{nr}(t)$, we obtain

$$\frac{1}{N_{nr}(t)}\frac{dN_r(t)}{dt}$$

which is called the *repair rate function* and is assumed to have a constant value of μ, the repair rate; μ has the units of repairs per unit of time.

Using the expression for the repair rate and the expression for the derivative of $N_r(t)$, we can write

$$\mu = \frac{1}{N_{nr}(t)}\frac{dN_r(t)}{dt} = \frac{N}{N_{nr}(t)}\frac{dM(t)}{dt}$$

which yields a differential equation of the form

$$\frac{dM(t)}{dt} = \mu\frac{N_{nr}(t)}{N}$$

We know, however, that $N_{nr}(t)/N$ is $1 - M(t)$, so we can write

$$\frac{dM(t)}{dt} = \mu(1 - M(t))$$

The solution to the differential equation is well known and is given by

$$M(t) = 1 - e^{-\mu t}$$

The relationship developed for $M(t)$ has the desired characteristics. First, if the repair rate is zero, the maintainability is also zero since the system cannot be repaired in any length of time. Second, if the repair rate is infinite, the maintainability is one since repair can be performed in zero time. A final interesting feature of the maintainability function is its value at a time corresponding to the MTTR. At t = MTTR, the maintainability function will be

$$M(t{=}\text{MTTR}) = 1 - e^{-\mu 1/\mu} = 1 - e^{-1} = .632$$

which implies that there is a probability of 0.632 that a system will be repaired in a time less than or equal to its MTTR.

As we have seen, the repair rate plays a crucial role in the maintainability of a system. The repair rate can differ depending on the type of repair that must be performed. For example, a banking system that can be repaired on location by a local maintenance person will have a better repair rate than one that must be returned to the factory or some third party for repair. In addition, certain types of faults can be easily repaired on location, whereas others require facilities that are not practical to bring to the location of the electronic system. For example, the replacement of a memory card can be performed easily at the site of the system, but the replacement of the power supply and cooling system can be much more difficult.

Because of the preceding issues, the repair rate is typically specified for several levels of repair. The most common partitioning is to provide three levels of repair. The first is called the *organizational level* and consists of all repairs that can be performed at the site where the system is located. Organizational repairs typically include all faults that can be located to specific circuit cards such that the cards can simply be replaced and the system made operational once again. For example, if an aircraft can be repaired without bringing it off the runway, it is considered an organizational level repair. The key to organizational repairs is the ability to locate the fault. It is seldom feasible to bring sophisticated fault detection and location equipment to the site of the system. Repairs at the organizational level must often depend on the built-in test provided by the system to locate the specific problem.

The second level of repair is called the *intermediate level*. Intermediate level repairs cannot be performed at the organizational level, but they can be performed in the immediate vicinity of the system. For example, a com-

puter firm can have a local repair facility to which the faulty pieces of equipment are taken for repair. Intermediate level repair is not as good as being able to perform the repair on site, but it is better than having to return a piece of equipment to the factory. In the case of an airplane, for example, an intermediate level repair might be made in the hangar as opposed to on the runway.

The final level of repair is called the *depot level* or the *factory level*. In depot level repairs, the equipment must be returned to a major facility for the repair process. For example, if a calculator cannot be repaired at home (organizational level), it is taken to the store from which it was purchased (intermediate level). If the store is unable to perform the repair, they send the calculator to a site designated by the manufacturer as a major repair facility (depot level). The length of time required to perform the repair depends on the level at which it is performed. It may take less than an hour to repair a device at the organizational level, several hours or perhaps days at the intermediate level, and as much as several weeks or months at the depot level.

As an example, assume that the MTTR for a computer system is 2.0 hours at the organizational level, 8.0 hours at the intermediate level, and one week (168 hours) at the depot level. The resulting maintainability functions are plotted versus time in Fig. 4.29. Note the tremendous difference that exists between the maintainability of the system for the different levels of repair.

4.8 Redundancy Ratios

A system that is more reliable or more available than another is better with regards to that one attribute. However, to achieve the improved reliability, availability, or maintainability, the system may contain an excessive amount of redundancy. The cost of the extra redundancy will appear in the weight, size, power consumption, volume, and financial costs of the improved system. In many applications, the improvements in the reliability, for example, may not be worth the extra weight that the system contains.

One good measure of the impact that improvements in reliability, availability, and maintainability have on a system is the **redundancy ratio**. The redundancy ratio is defined simply as the amount of hardware, information, time, or software that the redundant system requires divided by the amount required in a nonredundant system that performs the same function. The redundancy ratio can be specified for each type of hardware component; for example, the processors, memory, buses, interface units, power supplies, and displays. The redundancy ratio gives a measure of the extra resources required for a given application.

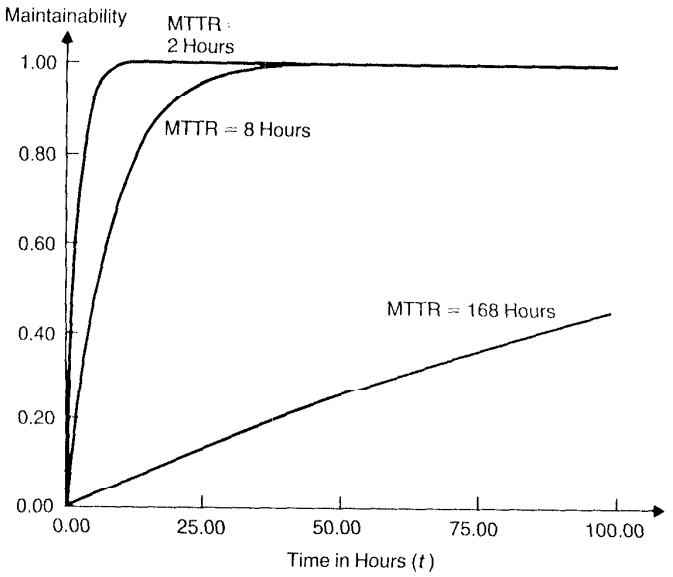

Fig. 4.29 Maintainability $M(t)$ versus time for example values of the MTTR.

A good example of the redundancy ratio can be found in a TMR processor system. If the nonredundant system requires only a single processor, the TMR system will clearly require three and will have a redundancy ratio for the processors of three. Redundancy ratios are equally applicable to software as well as hardware. If a nonredundant system requires 30,000 lines of code and the redundant system requires 40,000 lines, the software redundancy ratio is 1.33. In other words, one-third more software is required to implement the redundant system than is required in the nonredundant system.

4.9 Qualitative Methods

Thus far, we have discussed methods of evaluation that generate specific numbers to use to compare two or more systems. For example, one system may have a higher reliability, another may be less expensive, a third may weigh less, and still another may consume less power. Often, we find that certain attributes of a system that enter into the design process are extremely difficult to quantify. We may anticipate a drop in the prices of some hardware at a future date, or we may feel that the production plan on a

given system architecture will fit more easily into the production resources. In practice, the major decisions are often made using more qualitative information than quantitative. Consequently, it is important to understand the many qualitative comparisons that can be made.

4.9.1 Flexibility

The **flexibility** of a system is an important parameter that is extremely difficult to quantify. The basic flexibility issue in a fault-tolerant system is the ability to expand and improve as customer needs change and technological advances occur. When the expansion or the technology upgrades occur, the system's fault tolerance, reliability, availability, and maintainability must be maintained without the need to completely redesign the system. The military, for example, does not want to redevelop a flight control system simply because they wish to upgrade to a faster processor. Nor does a bank wish to lose some of its availability just to add another terminal to its transactions processing system.

Incorporating flexibility into a design is a very difficult task; it is often hard enough to design a system considering only the known environment and technology. To consider possibly unknown circumstances and capabilities to achieve a desired level of flexibility makes the design process even more difficult.

4.9.2 Technology Dependence

The dependence of a system on the technology employed is closely related to the flexibility of the system, but it must often be considered as a separate entity. Many systems are closely tied to the technology. For example, many aircraft flight control systems must use *flight-qualified* technology. In many cases, the technology is five to ten years old before it can meet the requirements necessary to become flight-qualified. Therefore, the designs developed using flight-qualified technology may lack the capability that can be obtained with other technology. **Technology dependence** must be considered when evaluating fault-tolerant systems. A system that is significantly more reliable, lighter, less expensive, and consumes less power may not be selected because it cannot be developed using currently qualified technology. The system's developers may not be willing to take a risk on the required technology becoming qualified before the system must be placed into operation.

4.9.3 Transparency to the User

A factor that is often overlooked in the design of a system is the impact that the system's characteristics have on the user. A system that is difficult to use will be doomed by end users' complaints. The attribute of fault toler-

ance often affects the end user of a system. For example, if the user's programs on a fault-tolerant system are impacted by the fact that there are redundant memories in the system, the inclusion of fault tolerance has had a tremendous effect on the user. For example, users of the system typically do not want to have to learn much about the system's fault tolerance characteristics in order to effectively use the system. Instead, the user wants to, for example, develop programs, send electronic mail, and perform database searches without any knowledge of the fault detection, voting, or error correction that is occurring within the system.

4.9.4 Testability

Testing is sometimes a painful process because of the lack of preparation for the test procedure. Experience over the past two decades has shown that the test procedure and the design process cannot be independent; instead, the two procedures must be coupled tightly to allow the design to be developed with **testability** in mind.

In a fault-tolerant system, testing is particularly important. Many fault tolerance techniques are designed to hide the occurrence of faults or errors. The test process, however, has completely the opposite goal. Testing attempts to make errors appear at an observable output of a system. In a fault-tolerant design, some means of easily testing the system must be developed. The solutions can be very simple in nature. For example, fault masking makes the TMR approach to achieve fault tolerance difficult to test. Even if one fault occurs, the TMR system still continues to produce the correct output because of the fault masking that is inherent in the majority voting. Attempts to test a TMR system by simply looking at the output as a function of the input can be extremely complicated or impossible. One approach to overcome the testing problem of the TMR system is to simply provide the inputs to the majority voter as primary outputs of the system. As a result, each of the three modules of the TMR system can be tested independently. The inputs to the voter should also be controllable by an external source to allow the voter to be tested independently of the remainder of the system.

4.10 Tradeoff Analysis Example

As with any new material, it is important to see the material applied before a true understanding can be obtained. To accomplish this goal, we examine a practical example that uses some of the evaluation tools that have been presented in this chapter [Johnson and Aylor 1986]. The example involves determining the benefits of including redundancy in the design of an electronic controller. We are interested in obtaining reliability, availability, and safety improvements. We will look at two aspects of the problem. In the

first, we use combinatorial modeling techniques to estimate the reliability and availability improvements of including redundancy in the system. In the second, we use Markov models to explore the safety aspects of two architectures for designing the redundant system. We begin our discussions with a brief description of the system.

The electronic controller that is to be designed controls the velocity of a direct current (dc) motor. The structure of the system is similar to that found in electric vehicle control systems and other applications that have a human providing commands to a system and a digital controller modifying those commands to guarantee the stability and acceptable response of the system. Example applications include electric wheelchairs, process control systems, robotic systems, remotely controlled vehicles, and certain aircraft control systems. The digital electronics are critical to the performance of the system, and a failure of the electronics can result in harm to either the system under control or the human.

The control system consists of five essential components: the user command sensors, controller electronics, motor drive electronics, motor feedback sensors, and the electric motors. The command sensors are typically potentiometers contained within joysticks. The velocity feedback sensors can be tachometers, back electromagnetomotive force (EMF) sensors, or other motor speed detection devices. The controller electronics use the driver commands and the feedback signals to develop input signals for the motor drive electronics. The motor drives are typically transistor or relay bridges that control the magnitude and direction of current flow to the direct current (dc) motors. A block diagram of the basic control system is shown in Fig. 4.30.

The core of the control system is the electronic controller. Any signal processing required within the system is normally performed by the controller electronics. For example, the feedback or command signals often require filtering to remove unwanted, and possibly detrimental, noise. Also, any velocity feedback that is implemented will be a part of the controller electronics. All of the control applications for this system use some form of velocity or position feedback to improve their response.

In the design of the electronic controller, safety, reliability, and availability are the three issues of primary importance. High availability is required to ensure that the system is operational for a very high percentage of the time. The user of the control system wants it to be functional and *ready to run* when its services are needed.

The electronic controller must also be safe. A failure of the controller must not result in the user, or the system itself, being hurt in any way. Faults that result in system failures have the potential to compromise the safety of the system, the user, or both. If a system is nonredundant, all failures have the potential to compromise the safety of the system. A system that contains fault detection capability, however, may be able to maintai

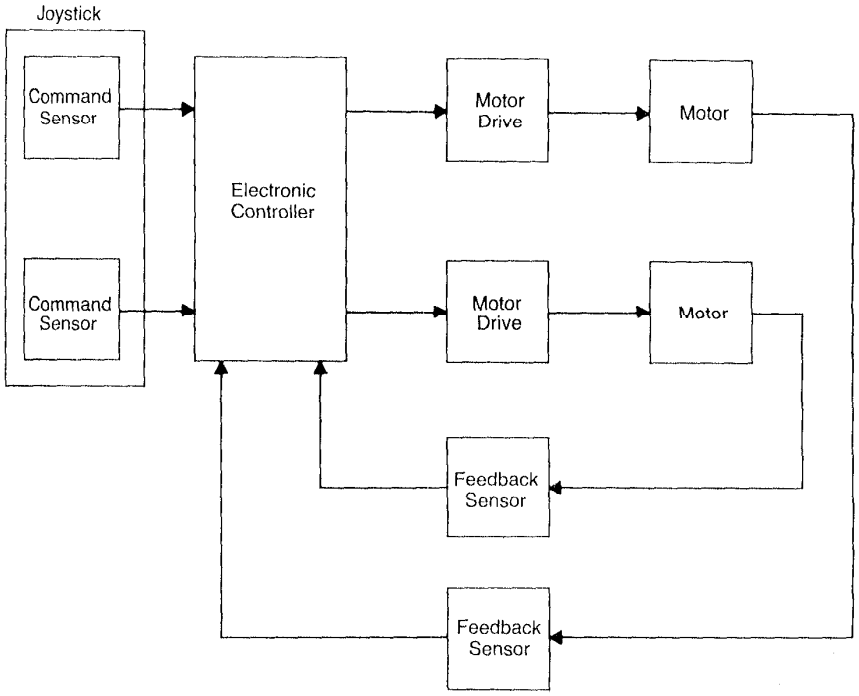

Fig. 4.30 The block diagram of the control system.

the safety of the system even though the fault is not tolerated. In essence, safety is the ability of a system to be fail-safe. Even if a fault results in the controller becoming inoperative, the fault should not cause the user to lose control of the system.

The reliability of the system is important because of the harm that can result if the system performs incorrectly. The manufacturer of the system depends on the analysis for some indication of the probability of system failure.

Specific requirements have not yet been derived for the reliability, availability, and safety of the system. The designer of the system is more interested, at this time, in the difference in these attributes between the redundant and the nonredundant systems.

The issues of reliability and availability are certainly not new ones. As we have seen in this chapter, commercial banking systems, industrial controllers, and military applications have always had stringent availability and reliability requirements. Military aircraft, for example, must have a

high probability of being available for use when the need arises. Likewise, the electronics found on board a military aircraft must be extremely reliable to protect the crew members and the aircraft itself.

We first look at the computation of the reliability and availability of the controller that does not include redundancy. We can use our reliability block diagrams and the exponential failure law to calculate the reliability of the system. The reliability block diagram for the basic control system is shown in Fig. 4.31.

Each component of the system is assigned a failure rate that represents the expected number of failures of that element per a specific time period. The reliability $R(t)$ of that component is then

$$R(t) = e^{-\lambda t}$$

where λ is the component's failure rate. As we have seen in this chapter, the exponential form of the reliability function can be proved mathematically under several fairly loose assumptions.

The reliability of the nonredundant system can be written as

$$R_{\text{system}}(t) = e^{-\lambda_i t} e^{-\lambda_i t} e^{-\lambda_e t} e^{-\lambda_d t} e^{-\lambda_d t} e^{-\lambda_m t} e^{-\lambda_m t} e^{-\lambda_f t} e^{-\lambda_f t} = e^{-\lambda_{\text{system}} t}$$

where λ_i is the failure rate of one potentiometer, λ_e is the failure rate of the electronics of the controller, λ_d is the failure rate of one of the motor drive circuits, λ_f is the failure rate of one feedback sensor, and λ_m is the failure rate of one of the motors. λ_{system} is the equivalent failure rate of the complete system and is given by

$$\lambda_{\text{system}} = 2\lambda_i + \lambda_e + 2\lambda_d + 2\lambda_m + 2\lambda_f$$

To allow specific data to be examined, consider a microprocessor-based controller. Assume that the controller consists of one 16-bit processor (such as an Intel 8086, Zilog Z8000, or Motorola 68000), 6K of random access memory (RAM), 64K of erasable, programmable read-only memory (EPROM), a 16-channel multiplexer and analog-to-digital converter, and two channels of digital-to-analog conversion. The motor drives are assumed to be transistor bridge circuits, and the motors are 24-volt, permanent-magnet, direct current motors. The input sensors are analog potentiometers, and the feedback sensors are analog circuits that measure the back EMF.

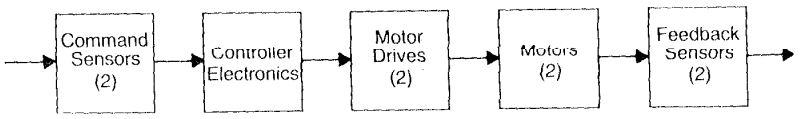

Fig. 4.31 Reliability block diagram for the nonredundant controller of Fig. 4.2