Non-State-Space Models
1) Reliability block diagrams
2) Fault trees
3) Reliability graphs

Can be analyzed based on the individual components & info. about the system structure; the assumption is that the failure or repair of a component is not affected by other components.

State-Space Models
1) Markov — the “sojourn” time, i.e., the amount of time in a state, is exponentially distributed.
   Chap. 4
2) Semi-Markov — the “sojourn” time, i.e., the amount of time in a state can be any distribution.
   Chap. 8
   When we associate “rewards” with states of Markov or Semi-Markov models, we have so called Markov reward models.
   Chap. 6
3) Stochastic Petri Net Models — a concise & more intuitive representation for the Markov model.
Chap. 7
   When we associate “rewards” to the markings of the net, we have stochastic reward nets.
**Markov Models (continuous-time)**

Two main concepts in the Markov model are “system state” and “state transition”.

Representing the change of state due to the occurrence of an event, e.g., failures, repairs, etc.

For reliability models, we frequently use faulty & non-faulty modules in the system.

Ex: TMR

System state representation:

\[(S_1, S_2, S_3)\] where \(S_i = \begin{cases} 1 & \text{if module } i \text{ is fault free} \\ 0 & \text{if module } i \text{ is faulty} \end{cases}\)

<table>
<thead>
<tr>
<th>States in which the system is operational</th>
<th>States in which the system has failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1,1)</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>(1,1,0)</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>(0,1,1)</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>(1,0,1)</td>
<td>(1,0,0)</td>
</tr>
</tbody>
</table>

How many of these?

\[2^n\] n is # of components in the state representation
State transition

(1,1,1) \quad \text{when module 1 fails} \quad (0,1,1)

Assume that each module obeys the exponential failure law and has a constant failure rate $\lambda$. The prob. of module 1 being failed at time $t+\Delta t$, given that it was operational at time $t$, is given by

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

\[ \therefore \text{ Assume only one failure at a time. Then the state diagram of TMR is as follows:} \]
The Markov model can be simplified by combining states having the same # of non-failed modules, i.e.,

\[ \sum \left( 1 - \lambda \Delta t \right) \left( 1 - 2\lambda \Delta t \right) \left( 1 - \lambda \Delta t \right) = 1.0 \]

The aggregate transition rate is from the perspective of source state; there is only a single component in the state representation.

\[ \text{prob\{system in state } j \text{ at } t + \Delta t\} = \sum \text{prob\{system was in state } i \text{ at } t\} \]

* prob\{a single transition from \( i \) to \( j \) occurs within \( \Delta t \)\}

\[ \text{prob\{system in state } j \text{ at } t + \Delta t\} = \sum \text{prob\{system was in state } i \text{ at } t\} \]

* prob\{a single transition from \( i \) to \( j \) occurs within \( \Delta t \)\}

\[ P_3(t + \Delta t) = (1 - 3\lambda \Delta t)P_3(t) \]

\[ \text{System is in state } 3 \quad \text{Prob. of } 3 \rightarrow 3 \text{ occurs at time } t+\Delta t \]

\[ \text{System was at state } 3 \quad \text{within time } \Delta t \]

\[ P_2(t + \Delta t) = 3\lambda \Delta tP_3(t) + (1 - 2\lambda \Delta t)P_2(t) \]

\[ P_F(t + \Delta t) = 2\lambda \Delta tP_2(t) + P_F(t) \]
Rewriting the above three expressions, we have:

\[
\frac{dP_3(t)}{dt} = \lim_{\Delta t \to 0} \frac{P_3(t + \Delta t) - P_3(t)}{\Delta t} = -3\lambda P_3(t) \quad \text{(1)}
\]

\[
\frac{dP_2(t)}{dt} = \lim_{\Delta t \to 0} \frac{P_2(t + \Delta t) - P_2(t)}{\Delta t} = 3\lambda P_3(t) - 2\lambda P_2(t) \quad \text{(2)}
\]

\[
\frac{dP_3(t)}{dt} = \lim_{\Delta t \to 0} \frac{P_F(t + \Delta t) - P_F(t)}{\Delta t} = 2\lambda P_2(t) \quad \text{(3)}
\]

Or in matrix form as

\[
\begin{bmatrix}
    P_3'(t) \\
    P_2'(t) \\
    P_F'(t)
\end{bmatrix}
= \begin{bmatrix}
    -3\lambda & 0 & 0 \\
    3\lambda & -2\lambda & 0 \\
    0 & 2\lambda & 0
\end{bmatrix}
\begin{bmatrix}
    P_3(t) \\
    P_2(t) \\
    P_F(t)
\end{bmatrix}
\]

or

\[
P'(t) = AP(t)
\]
* this can be derived directly from the following state-transition-rate diagram

```
3 3λ 2 2λ  F
```

negative: out
positive: in

\[ \therefore P_3'(t) = -3\lambda P_3(t) \]
\[ P_2'(t) = 3\lambda P_3(t) - 2\lambda P_2(t) \]
\[ P_F'(t) = 2\lambda P_2(t) \]

The set of differential equations can be solved numerically or analytically. To solve it analytically, one approach is to use Laplace Transform.

\[ F(t) \xrightarrow{\text{LT}} L(F(t)) = f(s) \]

\[ \text{Laplace domain} \]

\[ \text{Time domain} \]
\[ 1 \]
\[ t \]
\[ t^n \]
\[ e^{at} \]

\[ \frac{1}{s} \]
\[ \frac{1}{s^2} \]
\[ \frac{1}{n!} \]
\[ \frac{1}{s^{n+1}} \]
\[ \frac{1}{s-a} \]

Laplace transform of derivatives:

if \( L(F(t)) = f(s) \), then \( L(F'(t)) = sf(s) - F(0) \)

e.g., if \( L(P_3(t)) = P_3(s) \), then \( L(P_3'(t)) = sP_3(s) - P_3(0) \)
Applying LT, we have

\[ sP_3(s) - P_3(0) = -3\lambda P_3(s) \]
\[ sP_2(s) - P_2(0) = 3\lambda P_3(s) - 2\lambda P_2(s) \]
\[ sP_F(s) - P_F(0) = 2\lambda P_F(s) \]

Where \( P_3(s) \) is the LF of \( P_3(t) \)

\[ \therefore P_3(s) = \frac{1}{s + 3\lambda} \]

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

\[ & P_F(s) = \frac{6\lambda^2}{s(s + 2\lambda)(s + 3\lambda)} = \frac{1}{s} + \frac{-3}{s + 2\lambda} + \frac{2}{s + 3\lambda} \]

Apply the inverse LT

\[ \therefore P_3(t) = e^{-3\lambda t} \]
\[ P_2(t) = 3e^{-2\lambda t} - 3e^{-3\lambda t} \]
\[ P_F(t) = 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t} \]
For the TMR system, the system reliability is the sum of

\[ P_3(t) + P_2(t), \text{ i.e., } 1 - P_F(t) \]

\[
R_{\text{system}} = e^{-3\lambda t} + 3e^{-2\lambda t} - 3e^{-3\lambda t} = 3e^{-2\lambda t} - 2e^{-3\lambda t}
\]

Same expression as we obtained earlier using a reliability block diagram or a fault tree model.

In sharpe:

```sharpe
markov main(lambda)
3 2 3*lambda
2 F 2*lambda
end
3 1.0
end
* print cdf=F(t) in symbolic form
cdf(main;0.000001) * same as cdf(main,F;0.000001)
* print F(t) at t = 0.2, 0.4, 0.6, 0.8, 1.0
eval(main,F;0.000001) 0.2 1.0 0.2
end
```
Example: the 2P3m system

Modeling 1-out-of-3 memory & 1-out-of-2 CPU: the system is alive when at least one memory and one CPU are alive

\[ R_{\text{system}}(t) = P_{32}(t) + P_{31}(t) + P_{22}(t) + P_{21}(t) + P_{12}(t) + P_{11}(t) \]
bind
lambdap $1/720$  * MTTF of a processor
lambdam $1/(2\times720)$  * is 720 hrs
* MTTF of a memory
unit is $2\times720$ hrs
end

markov 2P3m
* memory failure
32 22 3*lambdam
22 12 2* lambdam
12 02 lambdam
31 21 3* lambdam
21 11 2* lambdam
11 01 lambdam
* processor failure
32 31 2*lambdap
31 30 lambdap
22 21 2 * lambdap
21 20 lambdap
12 11 2 * lambdap
11 10 lambdap
end
32 1.0
end

* Q(t)
echo Q(t) is as follows:
cdf (2P3m)
* R(t) can be found by
* “expr 1-value(t;2P3m)”;
* it can also be found by
* defining my own function
* called gp(t) below
func gp(t) value(t;2P3m,32)\n +value(t;2P3m,22)\n +.....\n +value(t;2P3m,11)
* R(1 hr)
* print reliability(t=1 hr)
expr 1-value(1;2P3m)
* use loop to print R(t) at
* different values
loop t, 0.5, 1, 0.1
expr gp(t)
end
end
Availability Modeling

Case 1: Independent repairman model, i.e., all components have own repair facility and can be repaired independently

unavailability

\[ U_p(t) = \frac{\lambda_p}{\lambda_p + \mu_p} \left( 1 - \frac{\lambda_p}{\lambda_p + \mu_p} e^{-(\lambda_p + \mu_p)t} \right) \]

\[ U_m(t) = \frac{\lambda_m}{\lambda_m + \mu_m} \left( 1 - \frac{\lambda_m}{\lambda_m + \mu_m} e^{-(\lambda_m + \mu_m)t} \right) \]
See p.354 text on a user-defined distribution syntax:

\[ \text{poly } \text{name(param-list) dist.} \]

\[
\begin{align*}
F(t) &= \sum_{j} a_j \cdot t^{k_j} \cdot e^{b_j t} \\
\end{align*}
\]

When defining a component, use unreliability \( F(t) \) for reliability modeling, and use unavailability \( U(t) \) or \( A(t) \) for availability modeling.
Case 2: There is only 1 repair facility capable of repairing one component at a time, with processor repair having a higher priority over memory repair.

Assume that the system is up when at least 1 processor & 1 memory are up.

When the system is in a failure state, it halts until it is repaired to become operational again, so no further component failure will occur in a failure state.

No, because processor repair takes priority over memory repair.
Same as before in the 2P3m Markov model for reliability modeling

\begin{verbatim}
bind
lambdap 1/720
lambdam 1/(2*720)
mup 1/4
mum 1/2
end
markov M
* memory failure
32 22 3*lambdam
* processor failure
30 31 mup
31 32 mup
20 21 mup
21 22 mup
10 11 mup
11 12 mup
01 02 mup
* memory repair
22 32 mum
12 22 mum
02 12 mum
end
\end{verbatim}

* steady state unavailability
expr prob(M,30)+prob(M,20)+ prob(M,10)+prob(M,01)+prob(M,02)
* for unavailability at time t = 1 hr
expr tvalue(1; M, 30)+
+tvalue(1; M, 20)+
+tvalue(1; M, 10)+
+tvalue(1; M, 02)+
+tvalue(1; M, 01)

Sharpe code for availability modeling of Case 2
Modeling Near-Coincident Fault using a Markov Model

System Description: (Section 9.4.1)

1. 4 CPUs & 3 memories ($\lambda_p$ & $\lambda_m$ are failure rates). The system must have at least 2 CPUs & 2 memories working.

2. When a CPU or memory fails, the system can reconfigure to remove the failed component.

3. Reconfiguration fails iff a second failure of the same component type (as the failed component) occurs during the reconfiguration period. The system cannot cope with such a near-coincident fault, i.e., the system fails if such a near-coincident fault occurs during the reconfiguration period.
4. Reconfiguration rate is $\alpha$

Here $c(n, \lambda)$ means the coverage factor when 1 out of $n$ components (with failure rate $\lambda$) fails: it is the probability that the system can successful perform a reconfiguration using the remaining $n-1$ components.

$$c(n, \lambda) = \frac{\alpha}{\alpha + (n-1)\lambda}$$
Time to occur:

\[ (T_2) \]
\[ \alpha \]

\[ (T_1) \rightarrow (n-1)\lambda \]

\[ F \]

Probability of fault recovered

\[ \frac{\alpha}{(n-1)\lambda + \alpha} \]

\[ P_{\text{fault}}(t) = e^{-(n-1)\lambda + \alpha}t \]

\[ P_{\text{recovered}}(t) = \frac{\alpha}{(n-1)\lambda + \alpha} \left(1 - e^{-(n-1)\lambda + \alpha}t \right) \]

\[ R_F(t) = \frac{(n-1)\lambda}{(n-1)\lambda + \alpha} \left(1 - e^{-(n-1)\lambda + \alpha}t \right) \]

\[ \therefore \text{ when } t = \infty \]
In general (even for non-exponential distribution)

\[
\text{prob of fault} \rightarrow \text{recovered}
\]

\[
= \text{prob } \{ T_2 < T_1 \}
\]

\[
= \int_0^\infty \text{prob} \{ t < T_1 \} f_{T_2}(t) \, dt
\]

\[
= \int_0^\infty e^{-(n-1)\lambda t} \cdot \alpha e^{-\alpha t} \, dt
\]

\[
= \frac{\alpha}{(n - 1) \lambda + \alpha}
\]

Laplace Transform for \( F(t) \) is

\[
f(s) = \int_0^\infty e^{-st} F(t) \, dt
\]

if \( F(t) = e^{-at} \)

then \( f(s) = \int_0^\infty e^{-st} \cdot e^{-at} \, dt = \frac{1}{s + a} \)
Sharpe code:

func c(n, λ)\n   alpha/(alpha+(n-1)*λ)
bind
   alpha 360
end

markov sift(λp,λm)

43 33 4*λp*c(4,λp)
33 23 3*λp*c(3,λp)
42 32 4*λp*c(4,λp)
32 22 3*λp*c(3,λp)
43 42 3*λm*c(3,λm)
33 32 3*λm*c(3,λm)
23 22 3*λm*c(3,λm)

* to failure state

43 F 4*λp*(1-c(4,λp))+3* λm*(1-c(3,λm))
33 F 3*λp*(1-c(3,λp))+3* λm*(1-c(3,λm))
23 F 2*λp+3* λm*(1-c(3,λm))
42 F 4*λp*(1-c(4,λp))+2* λm
32 F 3*λp*(1-c(3,λp))+2* λm
22 F 2*λp+2* λm

end

43 1.0

end

expr mean(sift, F; 0.0001, 0.00001)
expr 1-value(10;sift;0.0001,0.00001)
end