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.
- Used to describe the system at any time.

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

**Ex: TMR**

**System state representation:**

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

<table>
<thead>
<tr>
<th>System state</th>
<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>
<td></td>
</tr>
<tr>
<td>(1,1,0)</td>
<td>(0,0,1)</td>
<td></td>
</tr>
<tr>
<td>(0,1,1)</td>
<td>(0,1,0)</td>
<td></td>
</tr>
<tr>
<td>(1,0,1)</td>
<td>(1,0,0)</td>
<td></td>
</tr>
</tbody>
</table>

\[2^n\]  

\(n\) is # of components in the state representation
State transition

When module 1 fails

\[(1,1,1) \xrightarrow{\text{when module 1 fails}} (0,1,1)\]

In a Markov model, 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

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

\[\therefore\] Assume only one failure at a time. Then the state diagram of TMR is as follows:

The prob. that a transition will occur is determined by the prob. of failure, fault coverage, prob. of repair, etc.
The Markov model can be simplified by combining states having the same # of non-failed modules, i.e.,

\[ \sum \text{prob\{system was in state } i \text{ at } t \} \times \text{prob\{a single transition from } i \text{ to } j \text{ occurs within } \Delta t} \]

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

- System is in state 3 at time t + \Delta t
- Prob. of 3→3 occurs within time \Delta t at state 3

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

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

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

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

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

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 3\lambda \rightarrow 2 2\lambda \rightarrow F
```

\( \kappa \) 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. 

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

Time domain \( t^n \)
Laplace domain \( \frac{1}{s^n} \)

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 \( P_3(t) \)

\[ P_3(s) = \frac{1}{s + 3\lambda} \]
\[ P_2(s) = \frac{3\lambda}{(s + 2\lambda)(s + 3\lambda)} = \frac{3}{s + 2\lambda} + \frac{-3}{s + 3\lambda} \]
\[ & P_F(s) = \frac{6\lambda^2}{s(s + 2\lambda)(s + 3\lambda)} = \frac{1}{s} + \frac{-3}{s + 2\lambda} + \frac{2}{s + 3\lambda} \]

Apply the inverse LT

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

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

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 $1/720$

lambdam $1/(2 \times 720)$

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

* 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)
+value(t;2P3m,22)
+…..\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 using Sharpe

Case 1: all failed components can be repaired at the same time

unavailability

$$U_p(t) = \frac{\lambda_p}{\lambda_p + \mu_p} - \frac{\lambda_p}{\lambda_p + \mu_p} e^{-(\lambda_p + \mu_p)t}$$

$$U_m(t) = \frac{\lambda_m}{\lambda_m + \mu_m} - \frac{\lambda_m}{\lambda_m + \mu_m} e^{-(\lambda_m + \mu_m)t}$$
When defining a component, use unreliability $F(t)$ for reliability modeling, and use unavailability $U(t)$ or $\bar{A}(t)$ 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, i.e., no further component failure in a failure state

No, because processor repair takes priority over memory repair
\begin{align*}
\text{bind} & \\
\text{lamdamp} & \frac{1}{720} \\
\text{lambdam} & \frac{1}{2(2*720)} \\
\text{mup} & \frac{1}{4} \\
\text{mum} & \frac{1}{2} \\
\end{align*}

\text{markov} M
\begin{align*}
* \text{ memory failure} \\
32 & 22 & 3*\text{lambdam} \\
* \text{ processor failure} \\
* \text{ processor repair} \\
30 & 31 & \text{mup} \\
31 & 32 & \text{mup} \\
20 & 21 & \text{mup} \\
21 & 22 & \text{mup} \\
10 & 11 & \text{mup} \\
11 & 12 & \text{mup} \\
01 & 02 & \text{mup} \\
* \text{ memory repair} \\
22 & 32 & \text{mum} \\
12 & 22 & \text{mum} \\
02 & 12 & \text{mum} \\
\end{align*}
end

* steady state unavailability
\begin{align*}
\text{expr} & \text{ prob}(M,30) + \text{prob}(M,20) + \\
& \text{prob}(M,10) + \text{prob}(M,01) + \text{prob}(M,02) \\
\text{for unavailability at time } t = 1 \text{ hr} \\
\text{expr} & \text{tvalue}(1; M, 30) \\
& + \text{tvalue}(1; M, 20) \\
& + \text{tvalue}(1; M, 10) \\
& + \text{tvalue}(1; M, 02) \\
& + \text{tvalue}(1; M, 01)
\end{align*}
end

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. The system cannot cope with such a near-coincident fault, i.e., the system fails if such a near-coincident fault occurs in a 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:

\( (T_2) \)

\( \alpha \)

\( \Rightarrow \)

\( \text{fault} \rightarrow \text{recovered} \)

\( (T_1) \)

\( (n-1)\lambda \)

\( F \)

Probability of fault recovered

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

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

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

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

\( \therefore \) when \( t = \infty \)
In general (even for non-exponential distribution)

\[
\text{prob of fault} \xrightarrow{\text{recovered}} \text{prob } \{ T_2 < T_1 \} \cdot \text{pdf of } T_2 \\
= \int_0^\infty \text{prob } \{ t < T_1 \} \cdot f_{T_2}(t) \, dt \\
= \int_0^\infty e^{-(n-1)\lambda t} \cdot \alpha e^{-\alpha t} \, dt \\
= \frac{\alpha}{(n-1)\lambda + \alpha}
\]

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:

```plaintext
func c(n, \lambda)\n  alpha/(alpha+(n-1)*\lambda)
bind
  alpha 360
end
markov sift(\lambda.p, \lambda.m)
  43 33 4*\lambda.p*c(4,\lambda.p)
  33 23 3*\lambda.p*c(3,\lambda.p)
  42 32 4*\lambda.p*c(4,\lambda.p)
  32 22 3*\lambda.p*c(3,\lambda.p)
  43 42 3*\lambda.m*c(3,\lambda.m)
  33 32 3*\lambda.m*c(3,\lambda.m)
  23 22 3*\lambda.m*c(3,\lambda.m)
* to failure state
  43 F 4*\lambda.p*(1-c(4,\lambda.p))+3* \lambda.m*(1-c(3,\lambda.m))
  33 F 3*\lambda.p*(1-c(3,\lambda.p))+3* \lambda.m*(1-c(3,\lambda.m))
  23 F 2*\lambda.p+3* \lambda.m*(1-c(3,\lambda.m))
  42 F 4*\lambda.p*(1-c(4,\lambda.p))+2* \lambda.m
  32 F 3*\lambda.p*(1-c(3,\lambda.p))+2* \lambda.m
  22 F 2*\lambda.p+2* \lambda.m
end
43 1.0
end
expr mean(sift, F; 0.0001, 0.00001)
expr 1-value(10;sift;0.0001,0.00001)
end
```