Reliability and Availability Engineering: Modeling, Analysis, Applications

Chapter 13 - Non-Homogeneous CTMC

Kishor Trivedi and Andrea Bobbio

Department of Electrical and Computer Engineering
Duke University
Dipartimento di Scienze e Innovazione Tecnologica
Università del Piemonte Orientale

Jan 2017

Contents

NHCTMC: An introduction

NHCTMC models satisfy the Markov property but do not satisfy the homogeneity property.

As a consequence, the infinitesimal generator matrix will have one or more rates that are time dependent.

The time dependence will be on the global time, that is, time origin will be the beginning of system operation at $t = t_0$ (usually $t_0 = 0$).

Transition rates must all have the same global clock.

Non-Homogeneous CTMC - NHCTMC

The transient behavior of a NHCTMC is defined by the system of Kolmogorov Ordinary Differential Equations (ODE):

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)\boldsymbol{Q}(t) \tag{1}$$

with the initial probability vector $\boldsymbol{\pi}(t_0)$ at the beginning of NHCTMC operation at time t_0, subject to the normalization condition (for an n-state NHCTMC):

$$\sum_{i=1}^{n} \pi_i(t) = 1$$

The above equation is very similar to the one for a homogeneous CTMC with one difference: generator matrix entries are now assumed to be time dependent.

Non-Homogeneous CTMC - NHCTMC

If $\boldsymbol{Q}(t)$ is integrable, solution to the above equation exists and it is of the form

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(t_0)\,\boldsymbol{P}(t_0, t)$$

where entries $\boldsymbol{P}(t_0, t) = [p_{ij}(t_0, t)]$ are the probabilities that the Markov chain is in state j at time t given that it started in state i at time t_0.

In analogy with the general solution of an HCTMC, we may be tempted to write the solution as

$$P(t_0, t) = e^{\int_{t_0}^{t} \boldsymbol{Q}(\tau) d\tau}$$

But this does not hold in general unless $\boldsymbol{Q}(t)$ and its time integral $\int_{t_0}^{t} \boldsymbol{Q}(\tau) d\tau$ commute for $\forall t$.

Certainly this occurs for time-independent generator matrix of a homogeneous CTMC.

Non-Homogeneous CTMC - NHCTMC

A special case of NHCTMC in which the matrix $\boldsymbol{Q}(t)$ can be factored so that

$$\boldsymbol{Q}(t) = g(t)\,\boldsymbol{W}$$

In this case the solution to the NHCTMC can be written down as:

$$\pi(t) = \pi(t_0)e^{\left(\int_{t_0}^{t} g(\tau)d\tau\right)\boldsymbol{W}} = \pi(t_0)e^{\boldsymbol{W}g^* t}$$

Where $g^* = (\int_{t_0}^{t} g(\tau)d\tau)/t$ and $\boldsymbol{W}g^*$ is the generator matrix of an HCTMC.

Thus by solving an *equivalent* HCTMC, we can obtain the solution of the original NHCTMC in this special case. We refer to this method of solution of an NHCTMC as the equivalent-HCTMC method.

Non-Homogeneous CTMC - NHCTMC

For the special case of acyclic CTMC, convolution integration approach can be recommended:

$$p_{ij}(t_0, t) = \delta_{ij} e^{-\int_{t_0}^{t} q_{ii}(\tau)d\tau} + \int_{t_0}^{t} \sum_{k} p_{ik}(t_0, x) q_{kj}(x)\, e^{-\int_{x}^{t} q_{jj}(\tau)d\tau} dx,$$

where δ_{ij} is the Kronecker delta function defined by $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.
Corresponding equation for unconditional state probabilities, we have:

$$\pi_i(t) = \pi_i(t_0)\, e^{-\int_{t_0}^{t} q_{ii}(\tau)d\tau} + \int_{t_0}^{t} \sum_{k \neq i} \pi_k(t_0, x) q_{ki}(x)\, e^{-\int_{x}^{t} q_{ii}(\tau)d\tau} dx,$$

Illustrative Example

Example - Duplex processor in critical application - 1

Duplex processors are often used in control systems for safety critical applications.

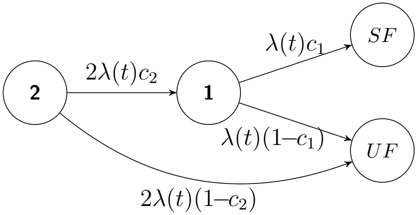
Time to failure is not restricted to be exponential and is allowed to be general.

Upon one failure, system is able to cope with it with the coverage probability c_2. With the complementary probability the failure is not covered leading to an unsafe system State UF.

From State 1, a failure can lead to a safe shut down SF (with probability c_1) or lead to the unsafe state UF (with probability $(1 - c_1)$).

Both the SF and UF are absorbing states.

Duplex processor in critical application - 2



With the assumption that two components are in hot standby mode, age of the both the components is measured from the beginning of system operation, i.e., with the global clock.

Hence the model is a NHCTMC.

Assuming that the TTF is Weibull distributed with the shape parameter β and the scale parameter η, we solve for the probabilities using the convolution integration approach.

Duplex processor in critical application - 3

$$
\begin{aligned}
\pi_2(t) &= e^{-2\int_0^t \lambda(x)dx} = e^{-2\int_0^t \eta\beta(\eta x)^{\beta-1}dx} \\
&= e^{-2(\eta t)^\beta}
\end{aligned}
$$

$$
\begin{aligned}
\pi_1(t) &= \int_0^t \pi_2(x) 2\lambda(x) c_2 e^{-\int_x^t \lambda(y)dy} dx \\
&= \int_0^t e^{-2(\eta x)^\beta} 2\eta\beta(\eta x)^{\beta-1} c_2 e^{-(\eta t)^\beta + (\eta x)^\beta} dx \\
&= 2\eta\beta c_2 e^{-(\eta t)^\beta} \int_0^t e^{-(\eta x)^\beta} (\eta x)^{\beta-1} dx \\
&= 2c_2 e^{-(\eta t)^\beta} \int_0^t e^{-(\eta x)^\beta} d((\eta x)^\beta) \\
&= 2c_2 e^{-(\eta t)^\beta} \frac{1}{\eta}(1 - e^{-(\eta t)^\beta}) = 2c_2(e^{-(\eta t)^\beta} - e^{-2(\eta t)^\beta})
\end{aligned}
$$

Duplex processor in critical application - 4

The probability of the fail safe state SF at time t is given by:

$$\begin{aligned}
\pi_{SF}(t) &= \int_0^t \pi_1(x)\lambda(x)c_1 dx \\
&= \int_0^t 2\eta\beta c_2(e^{-(\eta x)^\beta} - e^{-2(\eta x)^\beta})(\beta x)^{\beta-1}c_1 dx \\
&= 2c_1c_2 \int_0^t (e^{-(\eta x)^\beta} - e^{-2(\eta x)^\beta})d((\eta x)^\beta) \\
&= c_1c_2(1 - 2e^{-(\eta t)^\beta} + e^{-2(\eta t)^\beta})
\end{aligned}$$

Duplex processor in critical application - 5

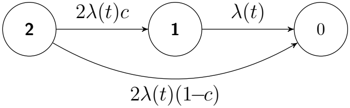
Finally, the system unsafety at time t is given by:

$$
\begin{aligned}
\pi_{UF}(t) &= \int_0^t \pi_1(x)\lambda(x)(1-c_1)dx + \int_0^t \pi_2(x)2\lambda(x)(1-c_2)dx \\
&= \int_0^t 2c_2(e^{-(\eta x)^\beta} - e^{-2(\eta x)^\beta})(\eta\beta)(\eta x)^{\beta-1}(1-c_1)dx \\
&+ \int_0^t e^{-2(\eta t)^\beta} 2\eta\beta(\eta x)^{\beta-1}(1-c_2)dx \\
&= c_2(1-c_1)(1-2e^{-(\eta t)^\beta} + e^{-2(\eta t)^\beta}) + (1-c_2)(1-e^{-2(\eta t)^\beta})
\end{aligned}
$$

From the above, we get the eventual absorption probability to the unsafe state as

$$\pi_{UF}(\infty) = c_2(1-c_1) + (1-c_2) = 1 - c_1 c_2$$

Reliability of a duplex processor - 1



To compute the duplex processor reliability we merge the two states *SF* and *UF* into a single state labelled 0.

The infinitesimal generator matrix in this case, is

$$\boldsymbol{Q}(t) = \begin{bmatrix} -2\lambda(t) & 2\lambda(t)c & 2\lambda(t)(1-c) \\ 0 & -\lambda(t) & \lambda(t) \\ 0 & 0 & 0 \end{bmatrix},$$

Since the solution equation can be factored into $\boldsymbol{Q}(t) = \lambda(t)\boldsymbol{W}$ where $\boldsymbol{W}$ is an HCTMC matrix, we can use the equivalent-HCTMC solution method.

$$\boldsymbol{Q}(t) = \lambda(t) \begin{bmatrix} -2 & 2c & 2(1-c) \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \lambda(t)\,\boldsymbol{W}$$

Reliability of a duplex processor - 2

The solution to the NHCTMC can be written down as:

$$\pi(t) = \pi(t_0)e^{(\int_{t_0}^{t} \lambda(\tau)d\tau)\boldsymbol{W}} = \pi(t_0)e^{\boldsymbol{W}\lambda^* t}$$

Where $\lambda^* = (\int_{t_0}^{t} \lambda(\tau)d\tau)/t$ and $\boldsymbol{W}\lambda^*$ is the generator matrix of an HCTMC.

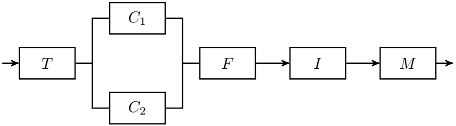
The final solution consists in solving a HCTMC with infinitesimal generator:

$$\boldsymbol{Q}^* = \boldsymbol{W}\lambda^*$$

To go on we need to define the functional expression for $\lambda(t)$.

Reliability of a multivoltage high speed train - 1

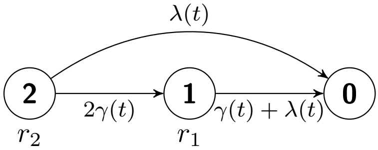
The RDB of a single module is in Figure.



The RDB of a single module can be modeled as a 3-state reward model.

Although in the original paper constant failure rates are assumed, we generalize to time dependent failure rates.

$\lambda(t)$ is the sum of the failure rates of the Transformer, the Filters, the Inverter and the Motors.
$\gamma(t)$ is the failure rate of an individual Four Quadrant Converter.



Reliability of a multivoltage high speed train - 2

We first derive the state probability expressions for the three states using the the convolution integration approach.

$$
\begin{aligned}
\pi_2(t) &= e^{-\int_0^t \left(\lambda(x)+2\gamma(x)\right)dx} \\
&= e^{-\int_0^t \lambda(x)dx} \cdot e^{-\int_0^t 2\gamma(x)dx} = e^{-\left(\Lambda(t)+2\Gamma(t)\right)}
\end{aligned}
$$

where: $\Lambda(t) = \int_0^t \lambda(x)dx$, $\Gamma(t) = \int_0^t \gamma(x)dx$

$$
\begin{aligned}
\pi_1(t) &= \int_0^t \pi_2(x) \cdot 2\gamma(x) \cdot e^{-\int_x^t \left(\lambda(y)+\gamma(y)\right)dy} dx \\
&= \int_0^t e^{-\left(\Lambda(t)+2\Gamma(t)\right)} \cdot 2\gamma(x) \cdot e^{-\int_x^t \left(\lambda(y)+\gamma(y)\right)dy} dx \\
&= 2\int_0^t e^{-\int_0^t \left(\lambda(y)+\gamma(y)\right)dy} \cdot \gamma(x) \cdot e^{-\Gamma(x)} dx \\
&= 2e^{-\left(\Lambda(t)+\Gamma(t)\right)} \cdot (1-e^{-\Gamma(t)}) = 2e^{-\left(\Lambda(t)+\Gamma(t)\right)} - 2e^{-\left(\Lambda(t)+2\Gamma(t)\right)}
\end{aligned}
$$

Reliability of a multivoltage high speed train - 3

$$\pi_0(t) = 1 - \pi_2(t) - \pi_1(t)$$

From the state probability expressions, we can write down the expected power available at time t as an example of expected reward rate at time t: The expected power $E[X(t)]$ available at time t is:

$$E[X(t)] \;=\; \sum_{i=0}^{2} r_i\, \pi_i(t) \;=\; r_2\, \pi_2(t) \,+\, r_1\, \pi_1(t)$$

and the expected accumulated energy delivered in the interval $(0, t]$ as an example of expected accumulated reward as:

$$E[Y(t)] \;=\; \sum_{i=0}^{2} \int_0^{\infty} \; r_i\, \pi_i(t)\, dt \;=\; r_2 \int_0^{\infty} \; \pi_2(t)\, dt + r_1 \int_0^{\infty} \; \pi_1(t)\, dt$$

Piecewise Constant Approximation

Piecewise Constant Approximation - Example

Piecewise Constant Approximation (PCA) is based on approximating a continuous time-variant function by a stair-case function whose value remains constant in certain intervals.

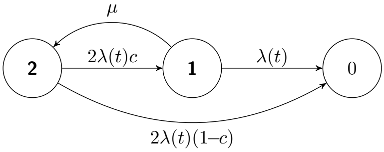
Given an interval $(0, t]$ over which the original continuous function should be evaluated, we divide the interval into $n > 1$ smaller intervals, usually of equal length, such that the function can be considered approximately constant over any of the n small intervals.

In this way, computing the value of the original function in the midpoint of each small interval, we can generate the approximating stair-case function.

The approximation improves as n increases and the length of each small interval decreases.

Duplex processor with repair - 1

In the 3-state model, consider adding a repair transition from State 1 back to State 2 with a time-independent rate μ.



With no repair the model was an NHCTMC.
When repair is introduced, we need two assumptions for the model to remain an NHCTMC:

- repair time is negligible compared to the time to failure;
- repair is minimal, i.e., the repaired processor is in a state (age) equal to the one just before its failure *(as bad as old)*.

Duplex processor with repair - 2

The infinitesimal generator matrix of this approximate NHCTMC is:

$$\boldsymbol{Q}(t) = \begin{bmatrix} -2\lambda(t) & 2\lambda(t)c & 2\lambda(t)(1-c) \\ \mu & -(\lambda(t)+\mu) & \lambda(t) \\ 0 & 0 & 0 \end{bmatrix}$$

The failure rate $\lambda(t)$ is assumed to be the hazard rate of a two-parameter Weibull distribution with an increasing failure rate:

$$\lambda(t) = \eta\beta(t\beta)^{\beta-1}$$

The table shows the values of the parameters.

Table: Parameter values

Parameter	Value
β	2.1
$1/\eta$	1.02
μ	120
c	0.9
t_0	0

Duplex processor with repair - 3

Such a model requires only a global clock to describe all time dependent transition rates, since in every state, each component is as old as the system.

The model can be studied as an approximate NHCTMC.

Matrix $\boldsymbol{Q}(t)$ cannot be factored and the convolution integration approach will not be easy to use since the transition graph is not acyclic.

To compute the reliability of this model, Piecewise Constant Approximation (PCA) can be used.

PCA consists in approximating the time continuous transition rate functions with a piecewise staircase function.

Duplex processor with repair - 4

The PCA method is based on the construction of a piecewise constant approximation of the time continuous failure rate $\lambda(t)$. The overall time interval $(t_0 = 0, t_1]$ is divided into $n+1$ shorter intervals of length δ:

$$t \in (0, t_1] \rightarrow t \in (i\delta, (i+1)\delta], \;\; i = 0, 1, \ldots, n$$

wherein the function is assumed to have the constant value $\lambda(i\delta)$:

$$\lambda(t) = \begin{cases} \lambda(\frac{\delta}{2}) & 0 < t \leq \delta \\ \lambda(\frac{\delta}{2} + \delta) & \delta < t \leq 2\delta \\ \vdots & \vdots \\ \lambda(\frac{\delta}{2} + i\delta) & i\delta < t \leq (i+1)\delta \\ \vdots & \vdots \end{cases} \quad ; \;\; i = 0, 1, \ldots, n$$

Duplex processor with repair - 5

The infinitesimal generator matrix $\boldsymbol{Q}(t)$ also makes changes at discrete epochs:

$$\boldsymbol{Q}(t) = \begin{cases} \boldsymbol{Q}(\frac{\delta}{2}) & 0 \leq t < \delta \\ \boldsymbol{Q}(\frac{\delta}{2}+\delta) & \delta < t \leq 2\delta \\ \vdots & \vdots \\ \boldsymbol{Q}(\frac{\delta}{2}+i\delta) & i\delta < t \leq (i+1)\delta \\ \vdots & \vdots \end{cases} \quad ; \quad i = 0, 1, \ldots, n$$

The transient state probabilities are computed successively in each sub-interval

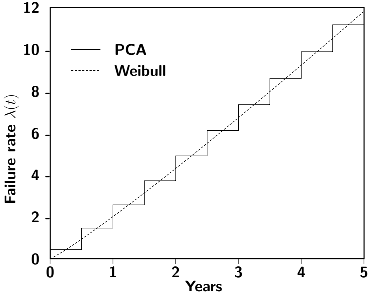
$$\pi(\delta) = \pi(0) e^{\boldsymbol{Q}(\frac{\delta}{2})(\delta - 0)}$$

$$\pi(2\delta) = \pi(\delta) e^{\boldsymbol{Q}(\frac{\delta}{2}+\delta)(2\delta - \delta)}$$

$$\ldots$$

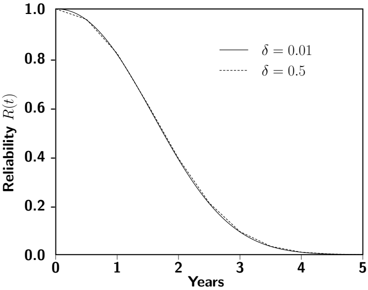
$$\pi(i\delta) = \pi((i-1)\delta) e^{\boldsymbol{Q}(\frac{\delta}{2}(i-1)+\delta)\delta}$$

Duplex processor with repair - 6



The Figure shows both the Weibull failure rate and its piecewise constant approximation with $\delta = 0.5$.

The Figure shows the reliability of the system for two different values of δ. We note that the two approximations are very similar.



Queueing Examples

M/M/m queue for dynamic memory allocation - 1

Consider a variation of the M/M/m queue where an arriving job will request a certain amount of memory dynamically.

Most jobs will release the acquired resource once they finished using it but occasionally this may not occur and the memory is “lost" or depleted as far as the Operating System (OS) is concerned.

This phenomena of resource leak/depletion is ubiquitous in most all computer operating systems and has been called software aging.

We aim to study the time to resource exhaustion due to such leaks.

Y. Bao, X. Sun, and K. S. Trivedi, "A workload-based analysis of software aging, and rejuvenation", *IEEE Transactions on Reliability*, vol. 54, no. 3, pp. 541-548, 2005.

M/M/m queue for dynamic memory allocation - 2

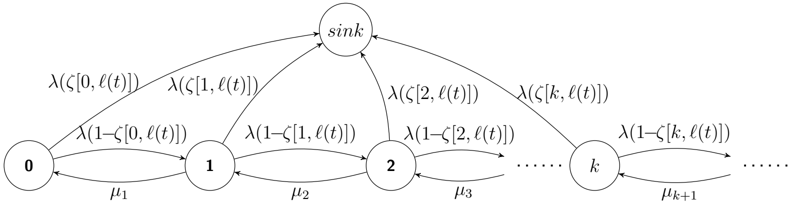
Each incoming request can cause the system to transit to the sink state if the amount of requested resource is more than the current available amount of the resource in the system.

We define first the variables necessary for modeling depletion caused by resource leakage.

- total amount of memory initially available: M
- resource requests arrival rate: λ
- amount of resource requested at each arrival: X with density $g(x)$
- number of processes in the system: k
- resource release rate: μ_k
- accumulated resource leak at time t: $\ell(t)$
- conditional probability of the system failure or crash in state k upon the arrival of a new request: $\zeta[k, \ell(t)]$

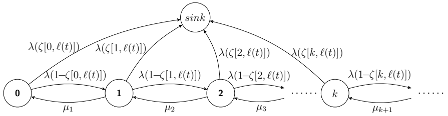
M/M/m queue for dynamic memory allocation - 3

The degradation model



M/M/m queue for dynamic memory allocation - 4

All the requests are i.i.d. r.v. X with density function $g(x)$. The total amount of resource requests after k requests is



$$S_k = \sum_{i=0}^{k} X$$

The density of S_k and its cdf are, respectively:

$$g^{[k]}(x) \qquad \text{and} \qquad G^{[k]}(x) = \int_0^t g^{[k]}(u)\,d\,u$$

where $g^{[k]}(x)$ is the k-fold convolution of $g(x)$.

M/M/m queue for dynamic memory allocation - 5

Given that $\ell(t)$ is accumulated resource leak at time t, the resource available is $(M - \ell(t))$ (where $\ell(t)$ will likely be an increasing function of the time).

The conditional probability of the system failure or crash in state k upon the arrival of a new request is:

$$\begin{aligned}
\zeta[k, \ell(t)] &= P\{S_{k+1} > (M - \ell(t)) | S_k \leq (M - \ell(t))\} \\
&= \frac{G^{[k]}(M - \ell(t)) - G^{[k+1]}(M - \ell(t))}{G^{[k]}(M - \ell(t))} \\
&= 1 - \frac{G^{[k+1]}(M - \ell(t))}{G^{[k]}(M - \ell(t))}
\end{aligned}$$

In the leak-free case ($\ell(t) = 0$), the model is a homogeneous CTMC since all transition rates are independent of the global time variable t.

M/M/m queue for dynamic memory allocation - 6

In the leak present case, the CTMC is non-homogeneous because $\zeta[k, \ell(t)]$ is a function of the global time variable t.

t time since the last reboot, and after failure the system is rebooted.

For the non-homogeneous CTMC model, with generator matrix $\boldsymbol{Q}(t)$, we solve the equation:

$$\frac{d\,\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)\,\boldsymbol{Q}(t) \quad , \qquad \text{with intial condition} \qquad \pi_0 = 1$$

and $$\pi_{sink}(t) = 1 - \sum_k \pi_k(t)$$

M/M/m queue for dynamic memory allocation - 7

Given T is the time to absorption, its cdf is $\pi_{sink}(t)$ and its hazard rate (the system failure rate):

$$h(t) = \frac{\frac{d\,\pi_{sink}(t)\,dt}{dt}}{1 - \pi_{sink}(t)} = \frac{\sum_k \lambda \zeta[k, \ell(t)] \pi_k(t)}{\sum_k \pi_k(t)}$$

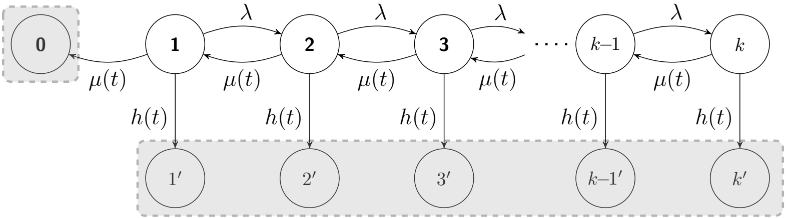
- In the homogeneous case $h(t)$ increases up to an asymptotic value;
- in the non-homogeneous case $h(t)$ increases monotonically as the leaked resource accumulates.

This can be said to be an analytic demonstration of the phenomena of software aging.

Effect of aging in program execution

We modify the behavior of an M/M/1/K queue so as to add two deleterious effects of (software) aging: performance degradation and crash/hang failures.

t is the global time since the last reboot.



S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, "Analysis of preventive maintenance in transactions based software systems," IEEE Trans. Comput., vol. 47, no. 1, pp. 96-107, Jan. 1998.

Effect of aging in program execution

Model parameters:

- Requests for processing arrival rate λ.
- Service rate $\mu(t)$
  - $\mu(t) = \mu$ is time-independent in the absence of aging,
  - μ is a decreasing function of time since the last reboot in case of slow performance degradation due to software aging
- $h(t)$ is the crash / hang failure rate and is an increasing function of time since the last reboot.
  $h(t)$ can be the failure rate obtained as an output in the previous example.

The NHCTMC can be solved using the PCA method.

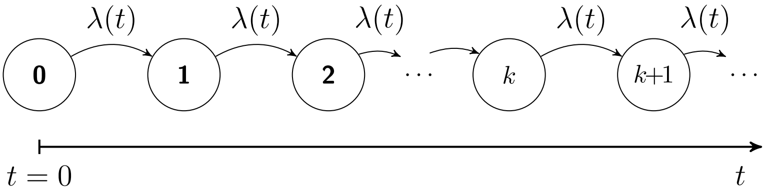
Reliability growth models

Reliability growth models

We consider a generalization of the homogeneous Poisson process to the case where the failure intensity λ is allowed to be time (age) dependent.

The time-dependence of the failure intensity function $\lambda(t)$ is measured using a global clock.

State diagram of this NHPP (non-homogeneous Poisson process) is shown in Figure



Reliability growth models

The NHPP models the number of detected failures $N(t)$ in the interval $(0, t]$.

Its pmf $P\{N(t) = k\} = \pi_k(t)$ can be derived using the convolution integration method as :

$$\pi_k(t) = e^{-m(t)} \frac{[m(t)]^k}{k!} \qquad k \geq 0$$

where the mean value function $m(t) = E[N(t)]$ is the expected number of failures detected by time t.

The derivative $\lambda(t) = \frac{dm(t)}{dt}$ is the failure intensity function.

The label on each arc emanating from state $k \geq 0$ is $\lambda(t)$ where t is measured with the global clock.

Reliability growth models - Duane model

In this model, the NHPP function is the hazard rate of a Weibull distribution

$$\lambda(t) = \eta\beta(\eta t)^{\beta-1} \qquad \text{then:} \qquad m(t) = \int_0^t \lambda(u)\,du = (\eta t)^{\beta}$$

This model is also known as the power law model and AMSAA (Army Materials Systems Analysis Activity) model.

The time to first failure (and occupancy or holding time of state 0) is Weibull distributed.

The sojourn time of any state $k > 0$ will not be Weibull distributed.

J.T. Duane, "Learning curve approach to reliability monitoring," *IEEE Trans. Aerospace*, Vol. 2, pp. 563-566 (1964)

Software reliability growth models (SRGM)

Software testing for bugs and following removal of bugs is expected to enhance the reliability of software systems.

Software reliability growth models are used to quantify and assess the extent of reliability growth achieved by the testing and debugging.

Assume that the number of failures $N(t)$ occurring during the time interval $(0, t]$ of testing follows a NHPP.

In finite failure SRGM the mean value function $m(t)$ reaches a limit, say, a as $t \to \infty$.

Then $m(t)/a$ satisfies properties of a distribution function.

Software reliability growth models (SRGM)

Let $F(t) = m(t)/a$ and let $h(t) = \frac{dF(t)/dt}{1-F(t)}$ be the corresponding hazard rate function. Since the failure intensity is $\lambda(t) = dm(t)/dt$, then

$$h(t) = \frac{\lambda(t)/a}{1 - m(t)/a}$$

and $\lambda(t) = h(t)[a - m(t)]$

- $h(t)$ can be interpreted as the failure occurrence rate per fault;
- $a - m(t)$ as the average number of remaining faults at time t
- a is the expected number of faults to be found after an infinite amount of testing.

Software reliability growth models (SRGM)

Different finite failure NHPP SRGMs can be obtained by specifying different functions $F(t)$ (or equivalently, $h(t)$).

These are listed in the Table

Table: Finite Failure NHPP SRGMs

	$F(t)$	$h(t)$	$m(t)$	$\lambda(t)$
Goel-Okumoto	$1 - e^{-bt}$	b	$a(1 - e^{-bt})$	abe^{-bt}
Gen Goel-Okumoto	$1 - e^{-bt^c}$	bct^{c-1}	$a(1 - e^{-bt^c})$	$abce^{-bt^c} t^{c-1}$
S-shaped	$1-(1+gt)e^{-gt}$	$\frac{g^2 t}{1+gt}$	$a[1-(1+gt)e^{-gt}]$	$ag^2 te^{-gt}$
Log-logistic	$\frac{(\lambda t)^\kappa}{1+(\lambda t)^\kappa}$	$\frac{\lambda\kappa(\lambda t)^{\kappa-1}}{1+(\lambda t)^\kappa}$	$a\frac{(\lambda t)^\kappa}{1+(\lambda t)^\kappa}$	$a\frac{\lambda\kappa(\lambda t)^{\kappa-1}}{[1+(\lambda t)^\kappa]^2}$

Software reliability growth models (SRGM)

A. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures", *IEEE Transactions on Reliability*, vol. 28, no. 3, pp. 206-211, 1979.

A. Goel, "Software reliability models: Assumptions, limitations, and applicability", *IEEE Transactions on Software Engineering*, vol. 11, no. 12, pp. 1411-1423, 1985.

S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection", *IEEE Transactions on Reliability*, vol. 32, no. 5, pp. 475-484, 1983.

S. Gokhale, M. Lyu, and K. Trivedi, "Analysis of software fault removal policies using a non-homogeneous continuous time Markov chain", *Software Quality Journal*, vol. 12, no. 3, pp. 211-230, 2004.

J. D. Musa and K. Okumoto, "A logarithmic poisson execution time model for software reliability measurement", *in Proceedings of the 7th International Conference on Software Engineering*, ICSE '84. Piscataway, NJ, USA: IEEE Press, 1984, pp. 230-238.

Numerical Solution Methods for NHCTMC

ODE Method

Most of the conventional ODE methods perform acceptably for transient analysis of non-stiff Markov models. One of the commonly used methods is the 4^{th} order Runge-Kutta which is an explicit single step method.

However, for stiff Markov chains explicit methods have enormous difficulty to solve the corresponding ODE.

The adaptation of the TR-BDF2 to NHCTMC is as follows. The trapezoid rule applied to interval $(t_i, t_i + \gamma h_i]$ is:

$$\boldsymbol{\pi}_{i+\gamma}\,(\boldsymbol{I} - \frac{\gamma\, h_i}{2}\boldsymbol{Q}(t_i + \gamma\, h_i)) = \boldsymbol{\pi}_i\,(\boldsymbol{I} + \frac{\gamma\, h_i}{2}\boldsymbol{Q}(t_i))$$

After computing $\boldsymbol{\pi}_{i+\gamma}$, we use the 2^{nd} *order backward difference formulae* (BDF2) to step from $t_i + \gamma h_i$ to t_{i+1}; this step requires the solution of the linear system

$$\boldsymbol{\pi}_{i+1}[(2-\gamma)I - (1-\gamma)\,h_i\,\boldsymbol{Q}(t_{i+1})] = \gamma^{-1}\boldsymbol{\pi}_{i+\gamma} - \gamma^{-1}(1-\gamma)^2\boldsymbol{\pi}_i$$

Uniformization Method

The NHCTMC Equation can be solved again by adapting the *uniformization* method to the non-homogeneous case.

For NHCTMC with n states and generator matrix $\boldsymbol{Q}(t) = [q_{ij}(t)]$, assume that there exists a $q < \infty$ such that, for $\forall t < \infty$, $q = \max_j |q_{jj}(t)|$.

We define the Poisson process $\{N(t), t \geq 0\}$ and the embedded DTMC with one-step transition probability matrix

$$\boldsymbol{Q}^*(t) = \boldsymbol{I} + \frac{\boldsymbol{Q}(t)}{q}$$

in the same way as for the homogeneous CTMC. The only difference is that now the transition probability matrix of the embedded DTMC $\boldsymbol{Q}^*(t)$ is time dependent.

Uniformization Method

Given the definitions above, the uniformization series for a NHCTMC for all $0 \leq t \leq \infty$ is:

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0) \sum_{k=0}^{\infty} \frac{(qt)^k}{k!} e^{-qt} \cdot \underset{(t_1 \leq t_2 \leq \ldots \leq t_k)}{\int_0^t \int_0^t \ldots \int_0^t} \boldsymbol{Q}^*(t_1)\boldsymbol{Q}^*(t_2)\ldots\boldsymbol{Q}^*(t_k)\, d\overline{H}(t_1, t_2, \ldots t_k)$$

$$= \boldsymbol{\pi}(0) \cdot \hat{\boldsymbol{U}}(t)$$

where $d\overline{H}(t_1, t_2, \ldots t_k)$ is the joint density of the order statistics $t_1 \leq t_2 \leq \ldots \leq t_k$ of a k-dimensional uniform distribution at $(0, t] \times \ldots \times (0, t] \subset R^k$, and $\boldsymbol{Q}^*(t_1)\boldsymbol{Q}^*(t_2)\ldots\boldsymbol{Q}^*(t_k)$ is the standard matrix product of the DTMC transition probability matrices at times $t_1 \leq t_2 \leq \ldots \leq t_k$.

Uniformization Method

The main complication in computing the non-homogeneous expression is the fact that a continuum of transition matrices is required.

To overcome this complication, a finite-grid approximation is used in [*]. Let $0 < h < q^{-1}$ be the step-size. The discrete-time approximation of the uniformization series, for arbitrary n and $t = nh$, is as follows:

$$\boldsymbol{U}(t) = \boldsymbol{U}(nh)$$

$$= \sum_{k=0}^{\infty} \frac{(qt)^k}{k!} e^{-qt} \cdot \left[\underset{n_1 \le n_2 \le \ldots \le n_k}{\sum_{0}^{n-1} \sum_{0}^{n-1} \ldots \sum_{0}^{n-1}} \boldsymbol{Q}(n_1 h) \boldsymbol{Q}(n_2 h) \ldots \boldsymbol{Q}(n_k h) \overline{H}(n_1, n_2, \ldots n_k) \right]$$

where $\overline{H}(n_1, n_2, \ldots n_k)$ is the pmf of the order statistics $n_1 \le n_2 \le \ldots \le n_k$ of a k-dimensional uniform distribution over $\{0, 1, \ldots, n-1\}^k$.

[*] N. M. van Dijk, "Uniformization for nonhomogeneous markov chains," *Operations Research Letters*, vol. 12, no. 5, pp. 283 – 291, 1992.