POLITECNICO DI MILANO

Performance Evaluation and Applications

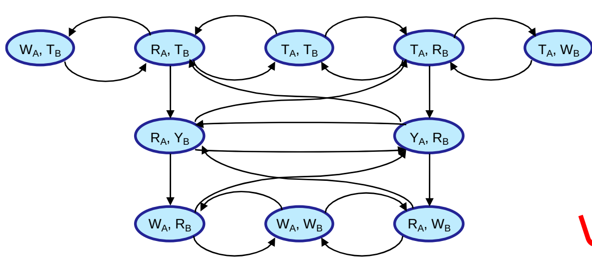
Stochastic Processes and Markov Chains

Motivating example

Discrete Event Simulation and State Machine with exponentially distributed transition times, are a tool with which we can model and analyze many interesting systems.

Proposed solutions are based however on random numbers generation: results are never guaranteed to be accurate, and the best we can do to cope with this situation is presenting confidence intervals instead of precise results.

Can’t we do anything better?



Utilization is between 0.512 and 1.368 with 90% probability!

?

Stochastic Processes

A *Stochastic Process* is a set of random variables

$$X_1, X_2, \dots X_n$$

that operates over the same set.

Since the variables might be correlated, the process must be described with the probability of obtaining a given outcome for a variable *i*, conditioned on the values of the previous outcomes:

$$P(X_i = a_i \mid X_1 = a_1, \dots X_{i-1} = a_{i-1})$$

Things however can be simplified a lot by considering smaller levels of correlations among the random variables.

Stochastic Processes

The index *i* of the random variables, can be either discrete or continuous.

$$X_1, X_2, \ldots X_n \quad or \quad X_t$$

If the index is continuous, it usually corresponds to the time.

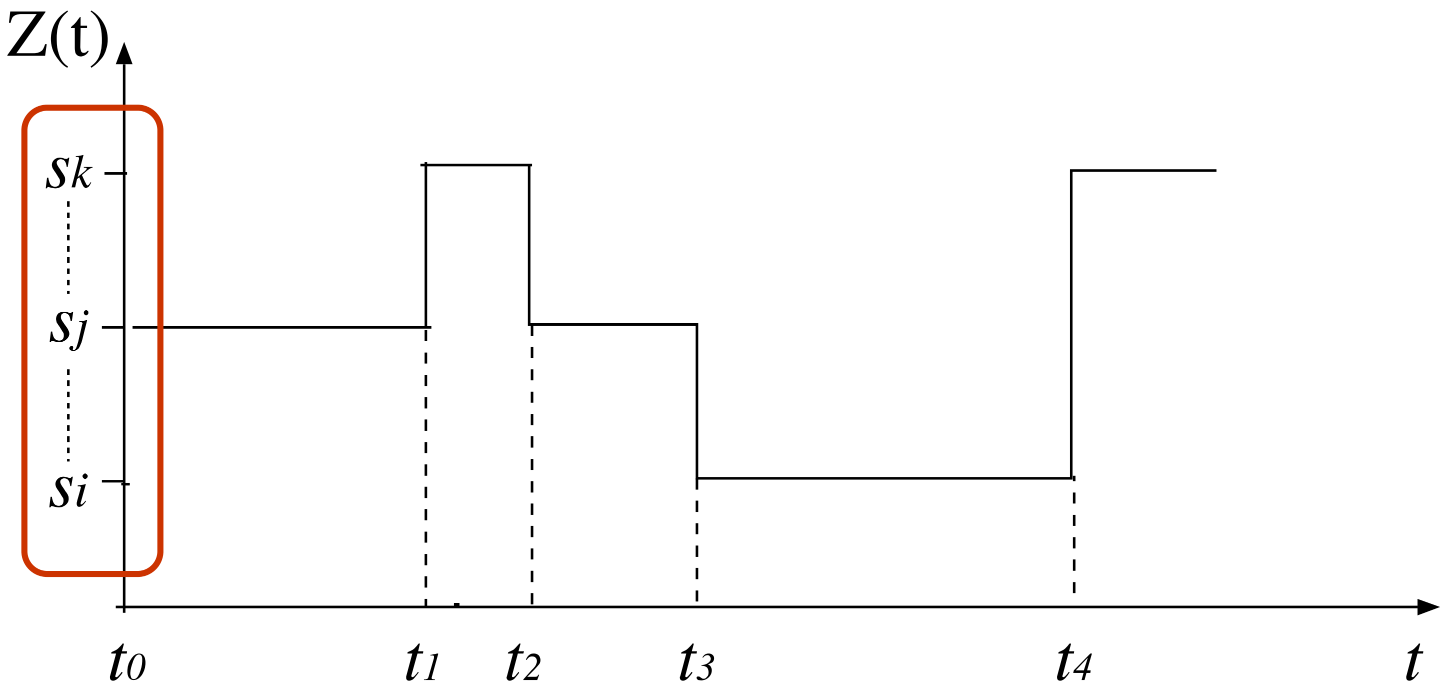
The set of outcomes of the random variables can also be discrete or continuous.

In performance studies, the index is usually *continuous*, denoting the *time*, and the set of outcomes is *discrete*, corresponding to the *states* of a state machine.

Continuous Time Markov Chains (1)

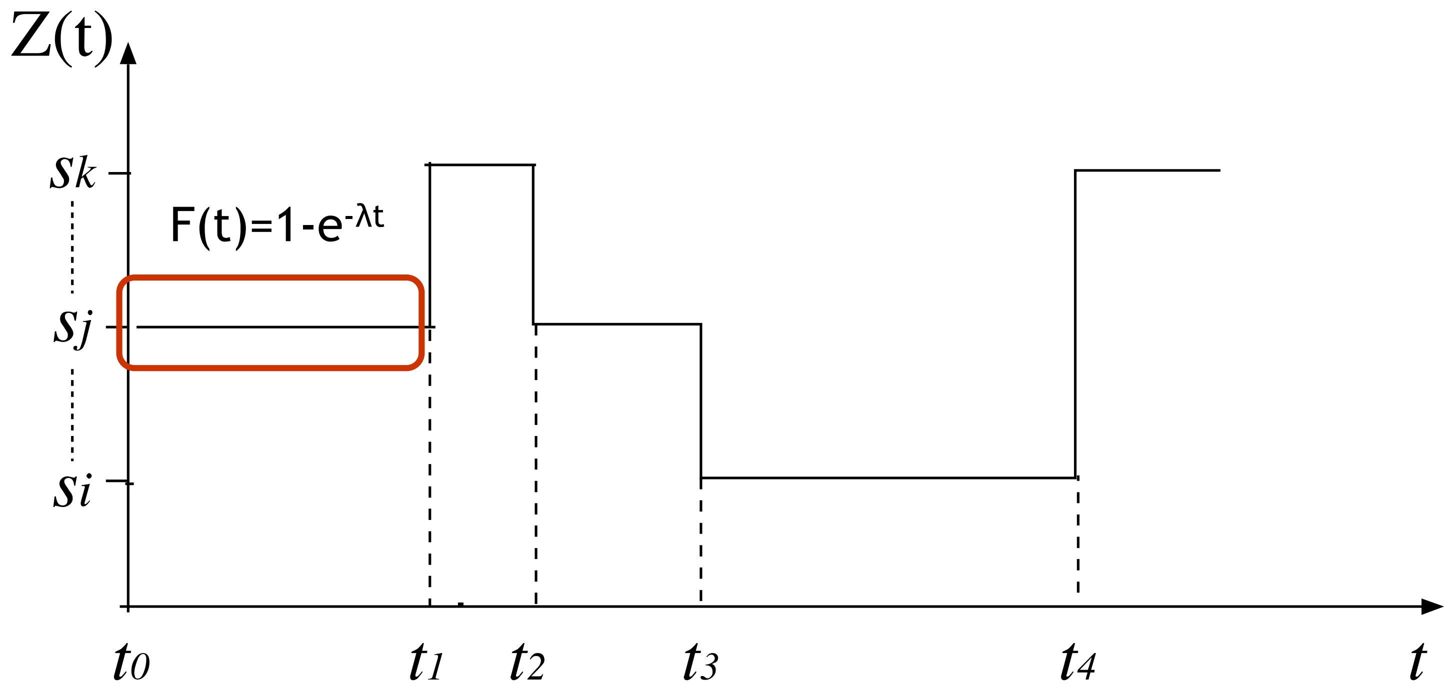
The special type of stochastic processes with discrete state and continuous time used in performance modelling are called *Continuous Time Markov Chains*.

The (finite or infinite) set of outcomes for the random variables, are called *states* $\{s_1, \ldots, s_N\}$.



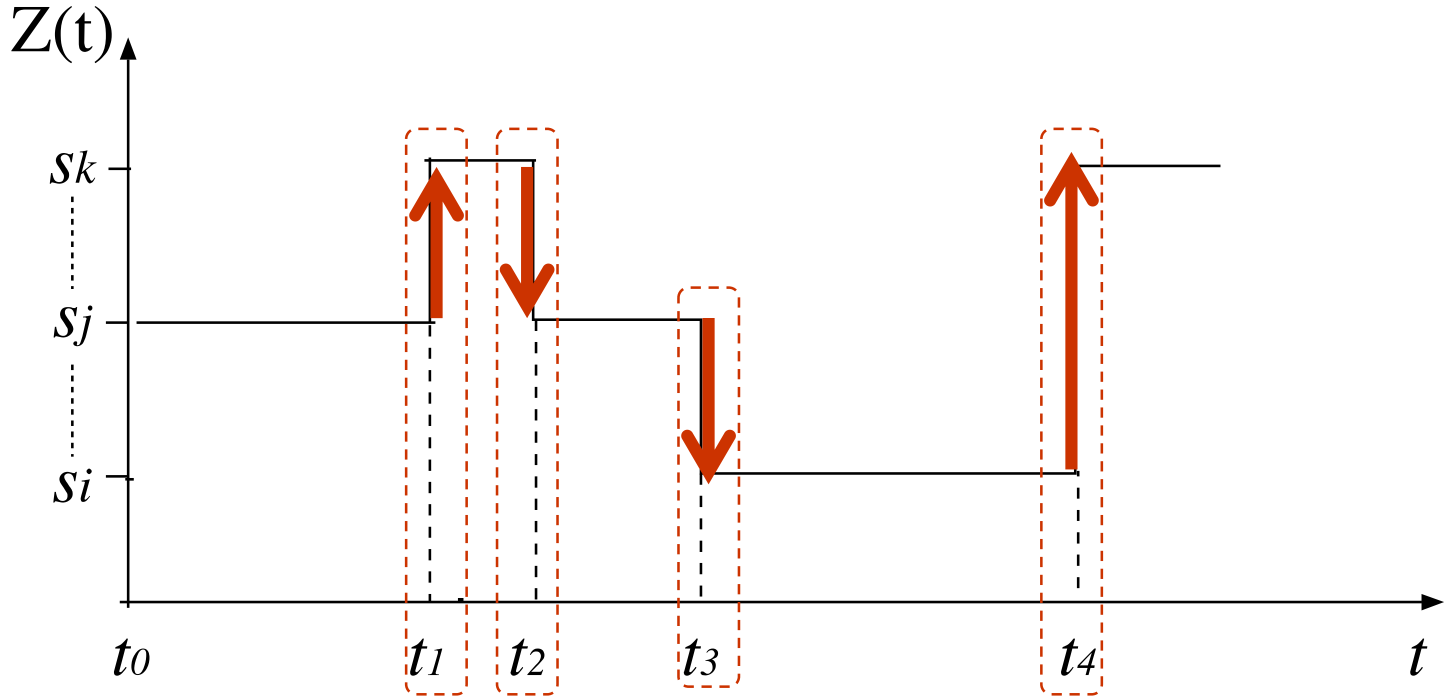
Continuous Time Markov Chains (2)

The system remains in a state s_j for a random exponentially distributed amount of time.



Continuous Time Markov Chains (3)

After that, it jumps to another state, which is also randomly choosen.



Continuous Time Markov Chains (4)

Thanks to the memory-less property of the exponential distribution, Markov Chains are stochastic processes where the probability of the state at time t_m depends only on the state at which the system was in a previous time t_{m-1} (and the total time passed t_m-t_{m-1}) :

$$(0 < t_1 < t_2 < \ldots < t_{m-1} < t_m)$$

$$Pr\{Z(t_m) = s_{j_m} | Z(t_{m-1}) = s_{j_{m-1}}, \ldots, Z(t_1) = s_{j_1}\} =$$

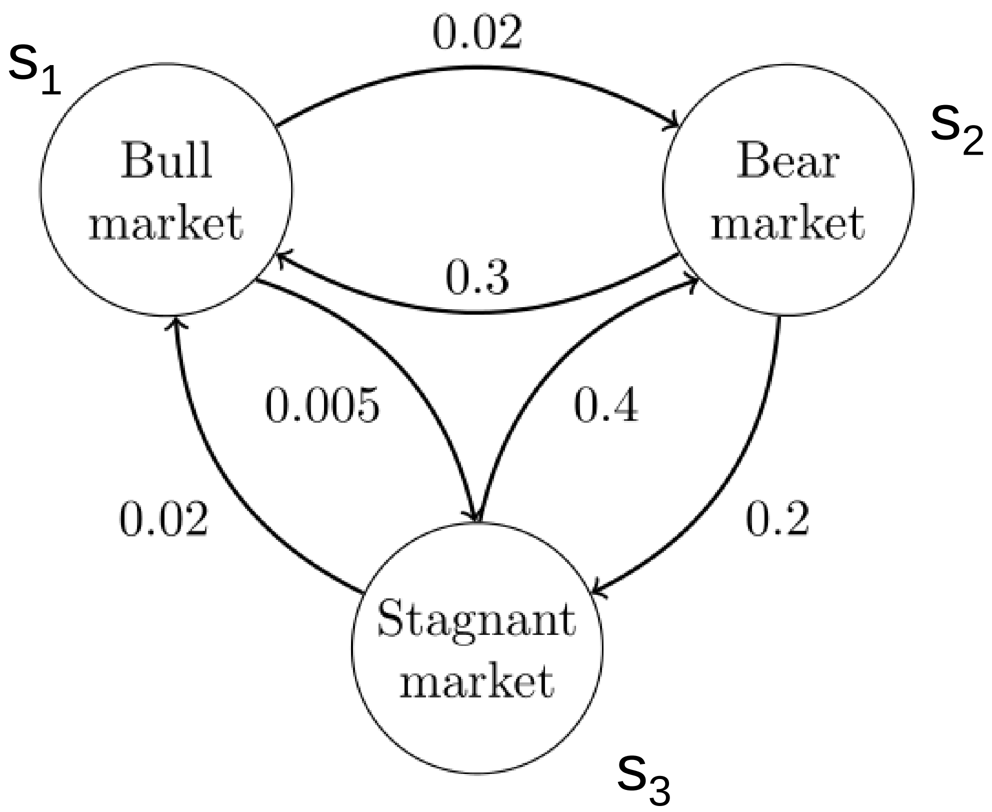
$$= Pr\{Z(t_m) = s_{j_m} | Z(t_{m-1}) = s_{j_{m-1}}\}$$

To simplify the definition, we will introduce the following notation:

$$\pi_i(t) = Pr\{Z(t) = s_i\}$$

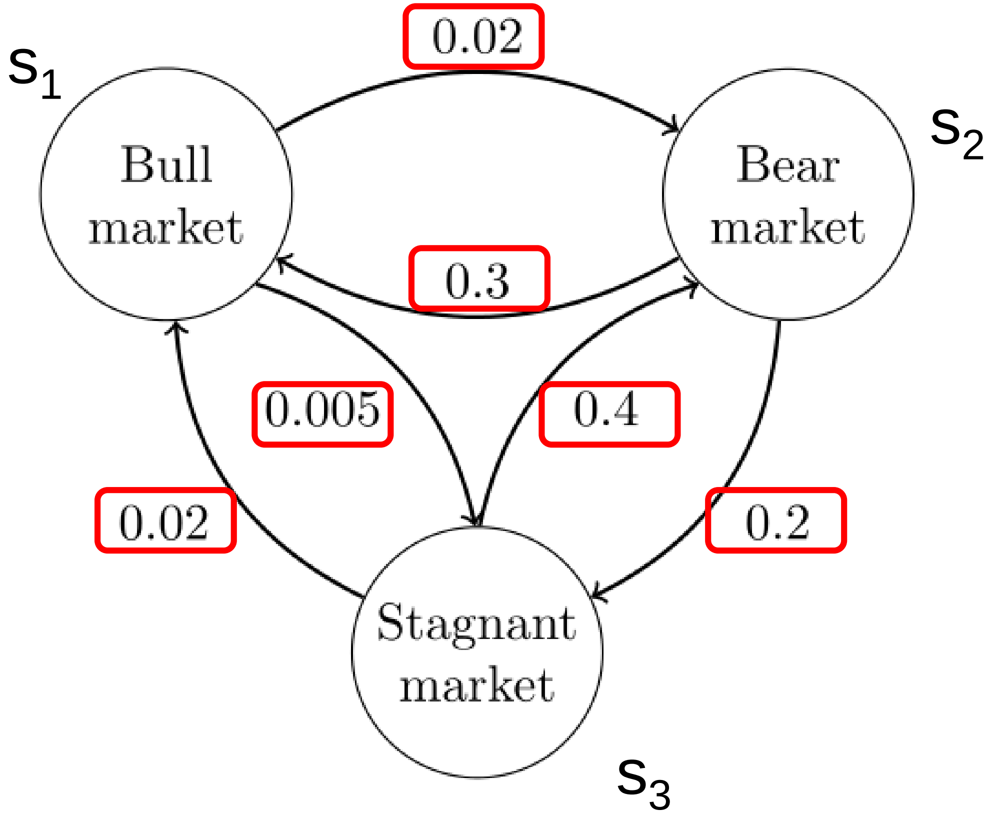
Continuous Time Markov Chains (5)

Usually CTMC are drawn as graphs, where nodes represent the states, and edges the possible transitions among the states.



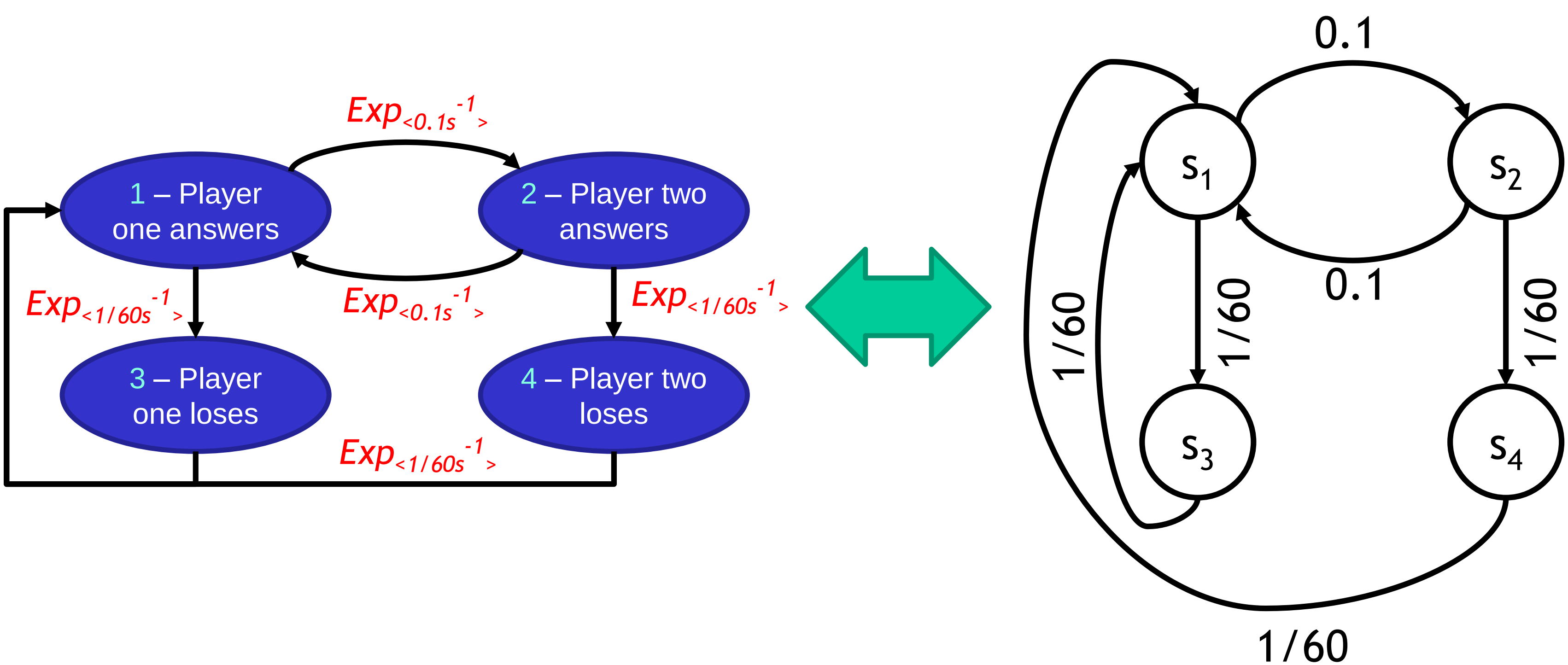
Continuous Time Markov Chains (5)

Arcs are labelled with the rate of an exponential distribution.



CTMC for dependability (1)

There is a one-to-one correspondence between a State Machine with exponential transition times, and a Continuous Time Markov Chain.



Continuous Time Markov Chain – transition rates

Each transition from state s_i to state s_j has associated a rate q_{ij} which corresponds to the rate of an exponential random variable.

The system in state s_i jumps to state s_j after an exponentially distributed random amount of time with rate q_{ij} .

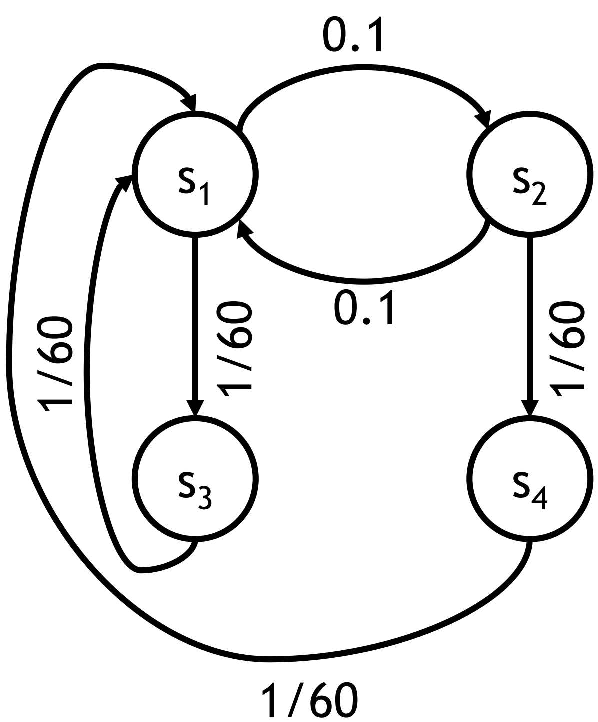
If there is no arc connecting state s_i to state s_j, then $q_{ij} = 0$.

If there are more than one arc exiting from state s_i , the system follows the evolution along the path of the event that happens first (*race policy*).

State Space (2)

The set $\Omega = \{s_i , \ldots\}$ of all the possible states the system can assume, is called the *state space* of the model.

For example:



$\Omega = \{s_1, s_2, s_3, s_4\}$ State space

$q_{12} = 0.1$
$q_{21} = 0.1$
$q_{13} = 1/60$
$q_{24} = 1/60$
$q_{31} = 1/60$
$q_{41} = 1/60$
$q_{14} = q_{23} = q_{32} = q_{34} = q_{42} = q_{43} = 0$

Transition rates

Transition rate

The transition rate q_{ij} can be seen as the limit of the probability that the system performs a jump in a small time Δt , (divided by Δt):

$$q_{ij} = \lim_{\Delta t \to 0} \frac{prob(\text{"System jumps from } s_i \text{ to } s_j \text{ in } \Delta t\text{"})}{\Delta t}$$

If Δt is small enough, the previous definition can be inverted.

$$prob\left(\text{“System jumps from } s_i \text{ to } s_j \text{ in } \Delta t\text{”}\right) = q_{ij} \cdot \Delta t$$

The Chapman-Kolmogorov equation (1)

CTMC analysis allows to compute the probability $\pi_i(t)$ that the system is in each state s_i of Ω at time t.

More formally, from $\pi_i(t)$ we can compute the probability $\pi_i(t+\Delta t)$ at time $t+\Delta t$ as the sum of two probabilities:

1. Being in state s_i at time t and not leaving state s_i in Δt
2. Being in another state s_j at time t (with $j \neq i$) and jumping from state s_j to state s_i in Δt (for every possible state s_j)

Being instate s_i at t

$$\pi_i(t+\Delta t) = \pi_i(t) \cdot \underbrace{\left(1 - \sum_{j\neq i}^{\dots} q_{ij} \cdot \Delta t\right)}_{\text{Not leaving state } s_i \text{ in } \Delta t \ (= 1 - \text{leaving state } s_i \text{ to another state } s_j)} + \sum_{j\neq i}^{\dots} \pi_j(t) \cdot \overbrace{q_{ji}}^{\text{Jumping from } s_j \text{ to } s_i \text{ in } \Delta t} \cdot \Delta t$$

Being instate s_j at t

The Chapman-Kolmogorov equation (2)

To simplify the equations we define q_{ii} as:

$$q_{ii} = -\sum_{j \neq i} q_{ij}$$

$$\pi_i(t + \Delta t) = \pi_i(t) \cdot \left(1 \overbrace{- \sum_{j \neq i} q_{ij}}^{+q_{ii}} \cdot \Delta t \right) + \sum_{j \neq i} q_{ji} \cdot \pi_j(t) \cdot \Delta t$$

$$p_i(t + \mathrm{D}t) = p_i(t) + p_i(t) \cdot q_{ii} \cdot \mathrm{D}t + \sum_{j \neq i} q_{ji} \cdot p_j(t) \cdot \mathrm{D}t$$

The Chapman-Kolmogorov equation (3)

Now, q_{ii} can be included in the summation. The equation becomes:

$$p_i(t+\mathrm{D}t) = p_i(t) + p_i(t)\cdot q_{ii}\cdot \mathrm{D}t + \sum_{j\neq i} q_{ji}\cdot p_j(t)\cdot \mathrm{D}t$$

$$\pi_i(t+\Delta t) = \pi_i(t) + \sum_j q_{ji}\cdot \pi_j(t)\cdot \Delta t$$

With some computation we can find:

$$\frac{\pi_i(t+\Delta t) - \pi_i(t)}{\Delta t} = \sum_j q_{ji}\cdot \pi_j(t)$$

Taking the limit of Δt to 0, we have:

$$\frac{d\pi_i(t)}{dt} = \sum_j q_{ji}\cdot \pi_j(t)$$

This equation is known as the *Chapman-Kolmogorov master equation*

Infinitesimal generator

The terms q_{ij} can be collected in a matrix Q :

$$Q = \begin{vmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{vmatrix}$$

Q is called the **Infinitesimal generator**.

Due to the definition of q_{ii}, the elements of **all the rows** of matrix Q must **sum up to *0***.

Chapman-Kolmogorov equation in matrix form

If we collect all the probabilities in a row vector $\pi(t)$, the Chapman-Kolmogorov equation in matrix form becomes:

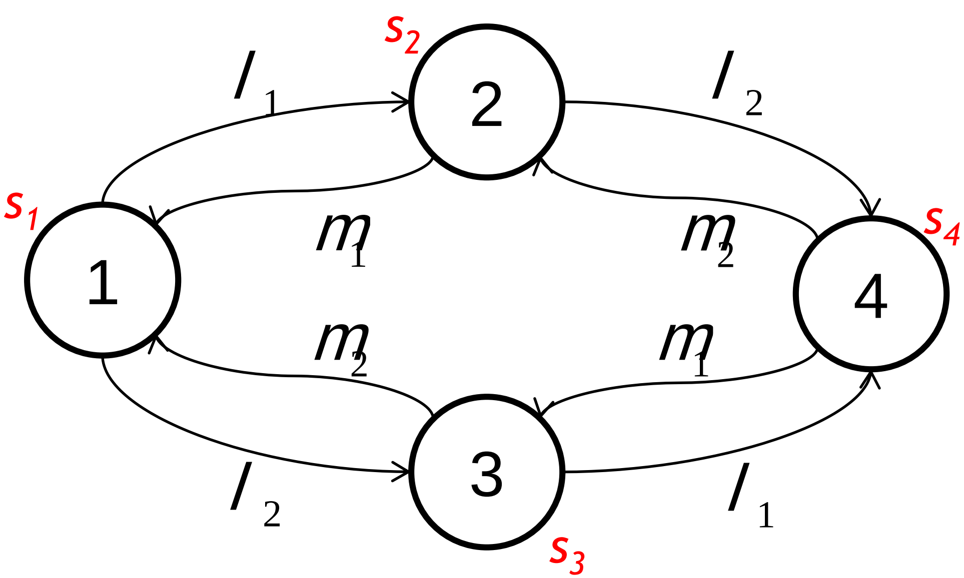
$$\frac{d\pi(t)}{dt} = \pi(t) \cdot Q$$

If we know the initial state distribution $\pi(0)$, we can compute the probability distribution at time t of the system being in each of its states, by solving the differential equation using $\pi(0)$ as the initial condition.

This is usually addressed as the *transient solution* of the model.

Defining the infinitesimal generator (1)

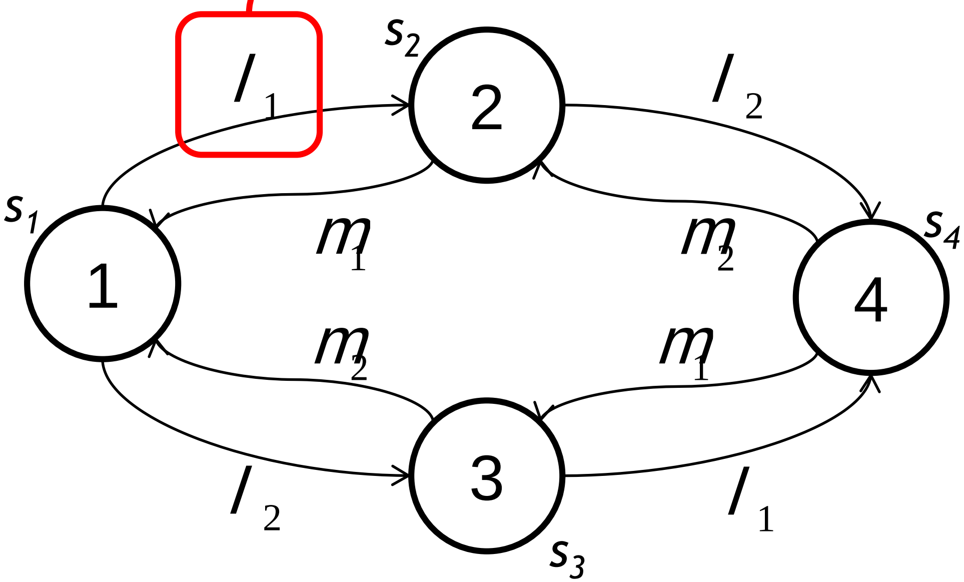
In general, we can create the Q matrix from its graphical representation, by first enumerating the states. Each state corresponds to a row and the a column of matrix Q.



$$
Q = \begin{array}{c|c:c:c:c|c}
 & s_1 & s_2 & s_3 & s_4 & \\
\hline
 & -\square & & & & s_1 \\
\hdashline
 & & -\square & & & s_2 \\
\hdashline
 & & & -\square & & s_3 \\
\hdashline
 & & & & -\square & s_4 \\
\end{array}
$$

Defining the infinitesimal generator (2)

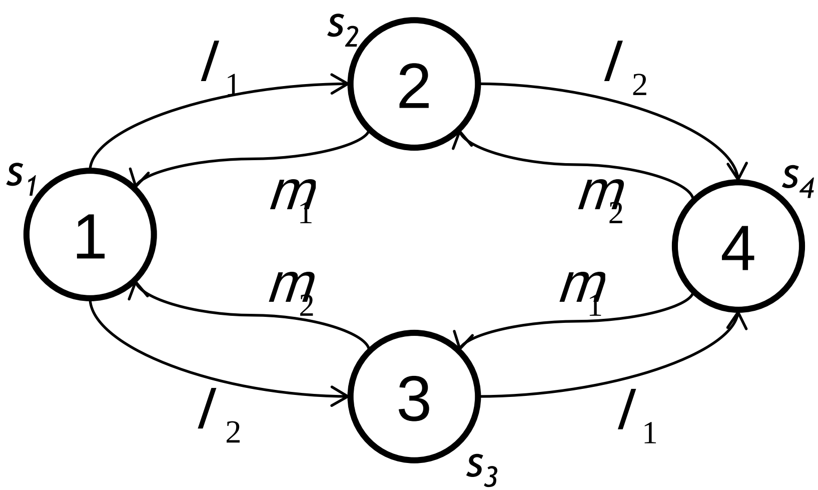
Then we put the transition rates associated to the arcs in the corresponding rows and columns.



$$Q = \begin{array}{c|cccc|c} & s_1 & s_2 & s_3 & s_4 & \\ \hline & -\square & \boxed{l_1} & l_2 & & s_1 \\ & m_1 & -\square & & l_2 & s_2 \\ & m_2 & & -\square & l_1 & s_3 \\ & & m_2 & m_1 & -\square & s_4 \\ \hline \end{array}$$

Defining the infinitesimal generator (3)

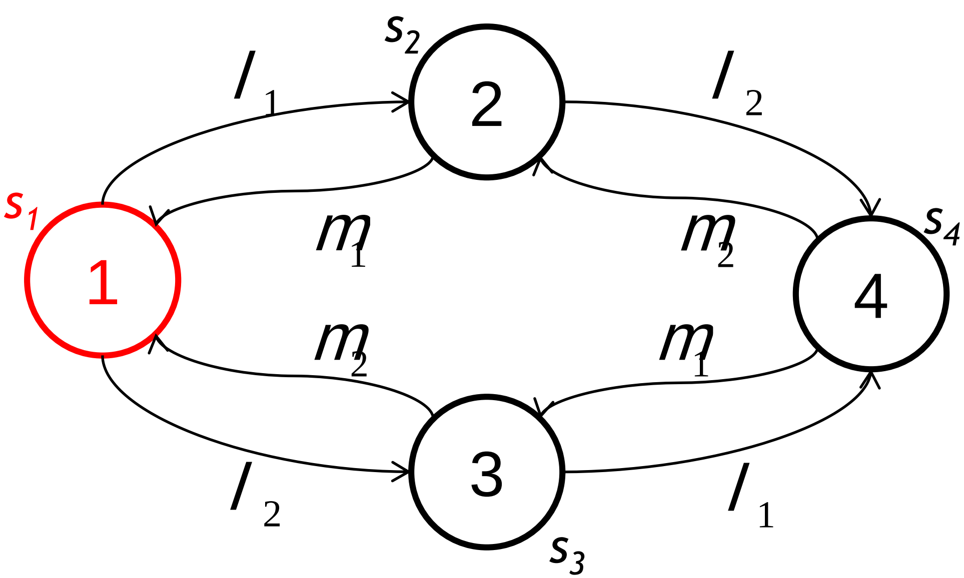
Finally we compute the diagonal element by summing the other elements of the rows, and changing their sign.



$$
Q = \begin{array}{c|cccc|c}
 & s_1 & s_2 & s_3 & s_4 & \\
\hline
 & \boxed{-\lambda_1-\lambda_2} & \lambda_1 & \lambda_2 & & s_1 \\
 & \mu_1 & -\mu_1-\lambda_2 & & \lambda_2 & s_2 \\
 & \mu_2 & & -\mu_2-\lambda_1 & \lambda_1 & s_3 \\
 & & \mu_2 & \mu_1 & -\mu_2-\mu_1 & s_4 \\
\hline
\end{array}
$$

Defining the infinitesimal generator (4)

Then we select one of the states of the system as its initial state (e.g. s_1). We can define the $\pi(0)$ as a vector with all zeros, except a value of 1 in the element corresponding to the initial state:



$$
\begin{array}{cc}
 & \begin{array}{cccc} s_1 & s_2 & s_3 & s_4 \end{array} \\
p(0) = & \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right|
\end{array}
$$

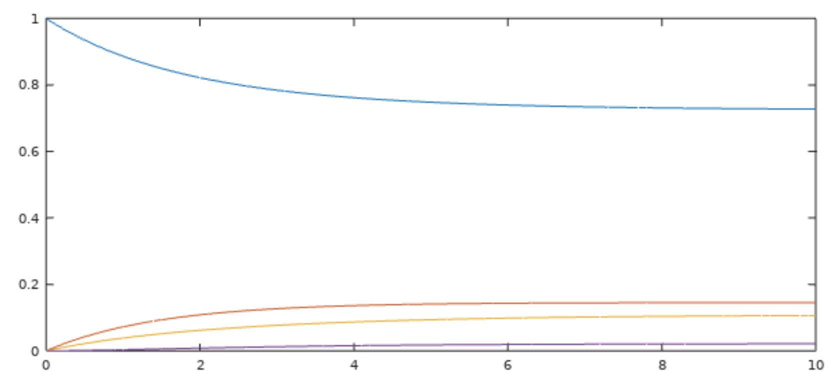
Solution of the ODE

The Chapman-Kolmogorov equation can then be solved numerically. For example, in Matlab, the code for the previously presented model, is:

```
MTTF1 = 10;
MTTF2 = 20;
MTTR1 = 2;
MTTR2 = 3;

l1 = 1/MTTF1;
l2 = 1/MTTF2;
m1 = 1/MTTR1;
m2 = 1/MTTR2;

Q = [-l1-l2,  l1   ,  l2   ,  0;
        m1  ,-m1-l2,   0   , l2;
        m2  ,   0  ,-m2-l1, l1;
         0  ,  m2  ,  m1   ,-m2-m1];

p0 = [1, 0, 0, 0];

[t, Sol]=ode45(@(t,x) Q'*x, [0 10], p0');

plot(t, Sol, "-");
```

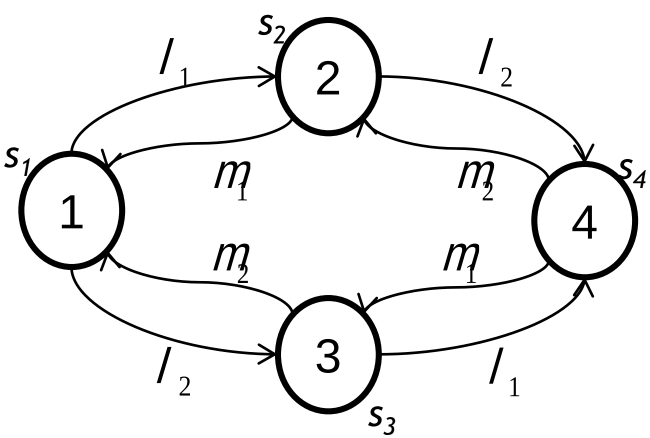
$$Q = \left| \begin{array}{cccc} -\lambda_1-\lambda_2 & \lambda_1 & \lambda_2 & \\ \mu_1 & -\mu_1-\lambda_2 & & \lambda_2 \\ \mu_2 & & -\mu_2-\lambda_1 & \lambda_1 \\ & \mu_2 & \mu_1 & -\mu_2-\mu_1 \end{array} \right|$$

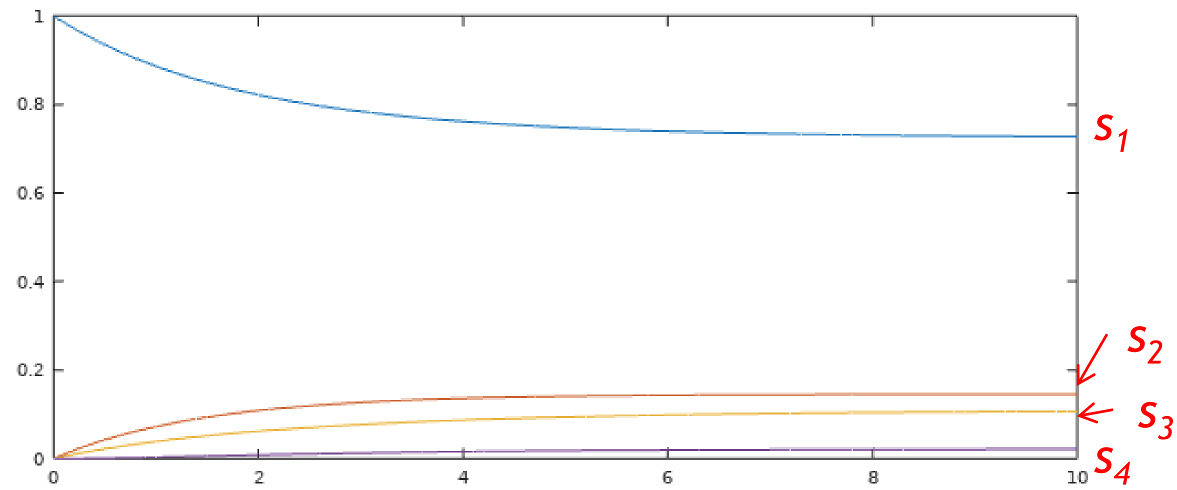
$$p(0) = \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right| \qquad \frac{d\pi(t)}{dt} = \pi(t) \cdot Q$$



Solution of the ODE (2)

The solution of the ODE is a time dependent vector $\pi(t)$, that tell us the probability of each state at each time instant t.



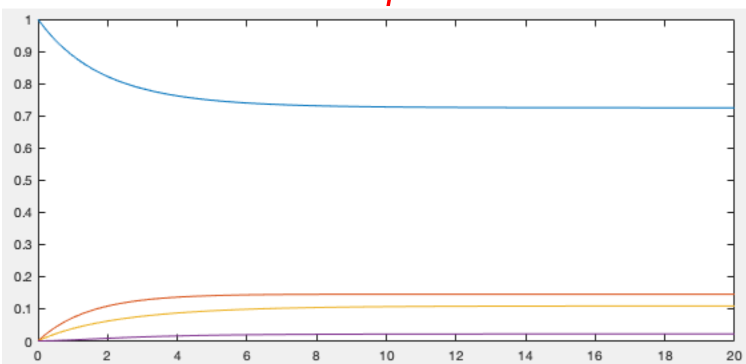


This vector is then used as the basis for computing other performance indices: we will return later on how this can be used.

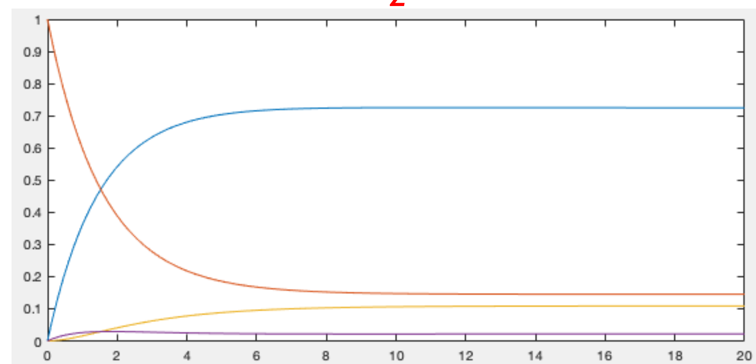
Solution of the ODE

Please note that the results greatly depend on the initial state: however, all models tends to the same limiting values.

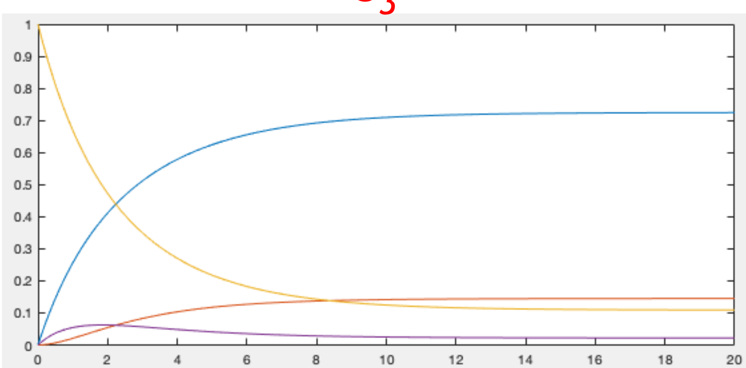
s_1



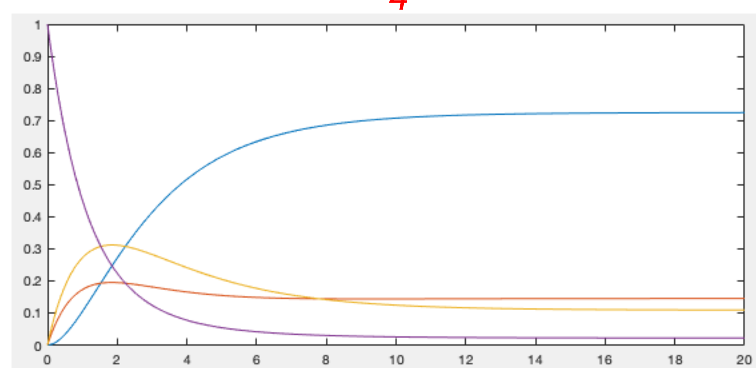
s_2



s_3



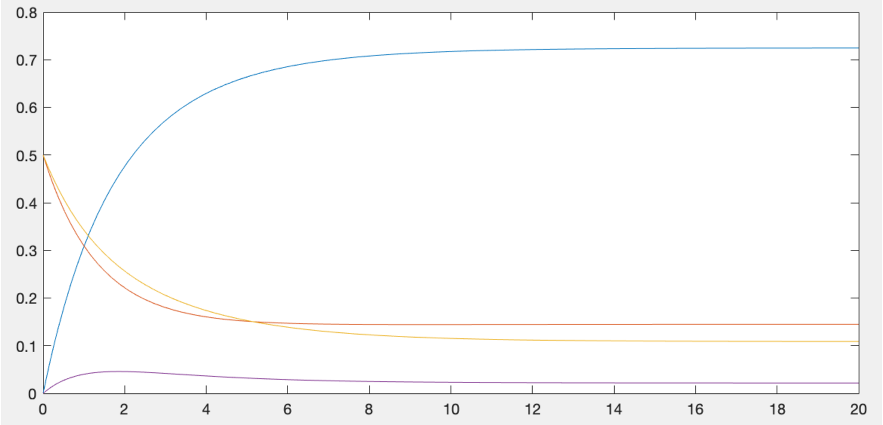
s_4



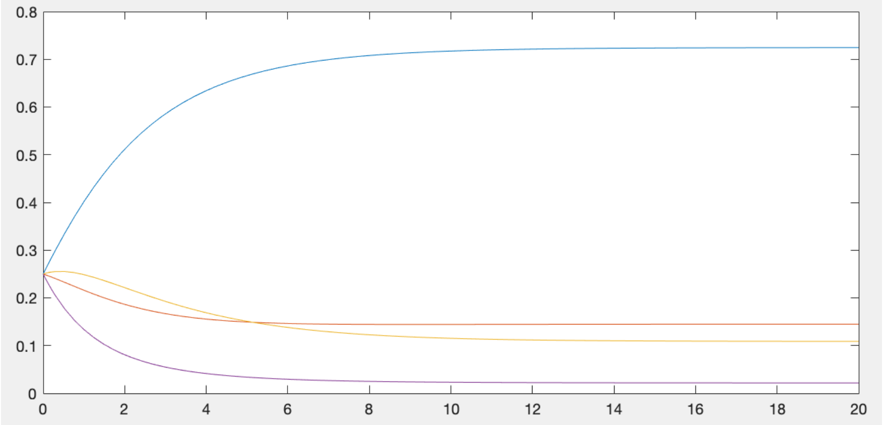
Defining the infinitesimal generator (4)

Note also that $\pi(0)$ could be any arbitrary vector where all components sums up to one. This is used to model systems where the initial state is not known.

$$
\begin{array}{cccccc}
 & s_1 & s_2 & s_3 & s_4 \\
p(0) = & \Big| \; 0 & 0.5 & 0.5 & 0 \; \Big|
\end{array}
$$

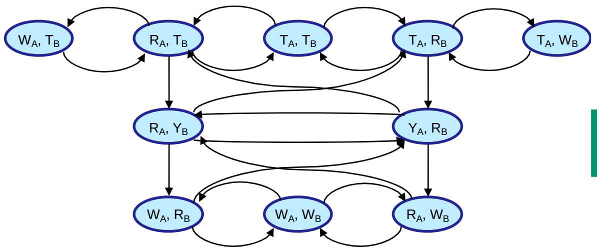


$$
\begin{array}{cccccc}
 & s_1 & s_2 & s_3 & s_4 \\
p(0) = & \Big| \; 0.25 & 0.25 & 0.25 & 0.25 \; \Big|
\end{array}
$$



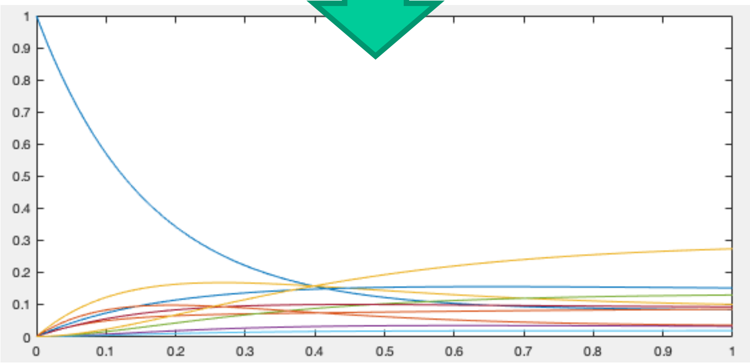
Analysis of Motivating Example

We can transform the State Machine into a Markov Chain, and analytically evaluate the desired performance indices.



Q =

-3.3425	0.9697	0.7242	0	0	0	0	0.6710	0	0.9776
0.8134	-4.8188	0	0	1.0366	0	0	1.8737	0	1.0950
0	1.7647	-3.1657	0	0	0	0	0	1.4010	0
1.5787	0	0	-5.1333	0.8966	0	1.7852	0	0.8729	0
1.0573	1.2793	0	0	-4.8189	0.8015	0.7579	0.9229	0	0
0	0	0	0.6858	1.6031	-6.4463	0.7865	1.7145	1.6564	0
0.9704	0.8440	0	0.9943	1.2748	1.0071	-7.1267	0	1.1396	0.8965
0.7354	0	1.6962	1.2422	0.8092	1.3857	0	-6.6503	0	0.7816
0	0.9024	1.0478	0.7753	1.0973	0	0	0	-4.5101	0.6874
0.7658	1.2104	0	1.3580	0	1.0180	0	0	0.8905	-5.2427



Utilization is 0.831