

## Home assignment number 1, 2015, in SF2863 Systems Engineering.

*Instructor:* Per Enqvist, [penqvist@math.kth.se](mailto:penqvist@math.kth.se).

This home assignment should be done in groups of 2 persons. The simulation part of the home assignment will be done in two versions, half of the groups should do the discrete simulations (6a,7a,8a) and the other half should do the continuous simulations (6b,7b,8b). Which version you should do is determined by logging in to BILDA, to the event SF2863, and join the first available group HAx or HByy. (With lowest number of xx or yy, so that we fill up the groups from beginning and have the same number of A and B groups if possible.)

One report per group is to be handed in, and you should describe in your own words how the problem was solved. Make the description rich enough so that it is totally clear how you defined your states and modelled the problem and describe how you implemented the simulations. It is not allowed to copy parts of another groups report or computer code! The answers to the questions in the assignment should be given in the main report and numbered as in this text. (In particular, it should not be necessary to look at your programming code to understand how you define and calculate things) Relevant print-outs and plots should be included in the report.

Computer code should be attached.

State your name, and email adress on the front of the report.

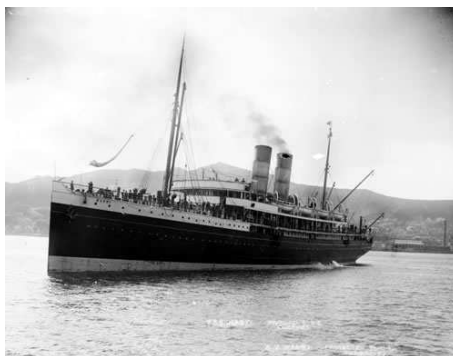
After the deadline has passed, a peer review of the reports will be done. Further instructions will be given at that point.

The report should be sent in to BILDA before 17.00, Wednesday, November 25. Note that this deadline is sharp. (Note that the assignment is completely voluntary, but that a correct report that is handed in in time, and a completed peer review, will grant you bonus points at the final exam.)

This home assignment can grant a maximum of two bonus points on the exam.

---

In this home assignment a simple Markov process in continuous time will be examined. We consider a big (old) ferry with two engines that each drives one propeller. If one engine breaks down the ferry can still move, but it runs a bit slower, and if both breaks down the ferry stops.



Assume that engine number 1 breaks down after some time that is modelled as a stochastic variable with  $\text{Exp}(\lambda_1)$  distribution. Similarly, engine number 2 breaks down after some time that is  $\text{Exp}(\lambda_2)$  distributed. Three repairmen are working on the ferry for repairing

the engines. The repair time is  $\text{Exp}(n_k\mu)$  distributed, where  $n_k$  is the number of repairmen working on engine  $k$ .

Let  $V(t)$  denote the speed of the ferry at time  $t$ . If both the engines are working the ferry is running at a speed which is  $v$ , if only engine 1 is working the speed is  $v_1$  and if only engine 2 is working the speed is  $v_2$ .

**Parameter values**

- $v = 18$ ,  $v_1 = 10$ , and  $v_2 = 14$ .
- $\lambda_1 = 10$ ,  $\lambda_2 = 20$ , and  $\mu = 8$
- $h = 0.001$

Now, the system is a Markov process that will depend on how many workers are assigned to repair the engines if they break down. If only one engine is broken it is obvious that it is optimal to let all three repairmen work on it. If both engines are broken it is not so clear how to distribute the workers. To decide this you should for each different repair strategy determine the average speed of the ferry at steady state and this way choose the best strategy.

You should do this in two different ways.

First, all of you should determine the average speed using the theory of Markov processes. For this approach you need to (for all possible repair strategies)

1. define the states of the Markov process,
2. determine the intensity matrix,
3. motivate why there should exist a stationary distribution
4. determine the stationary distributions (for all repair strategies using the intensities defined above.)
5. determine the average speed from this stationary distribution.

Second you should simulate the system numerically and determine the average speed by ergodic estimates.

This will be done in two different ways; half of the groups will use a time discretization approach, while the second half will directly simulate the continuous time process.

**Discretization approach**

Now a discretization of the continuous time process will be made.

Start the process at time 0 with both engines of the ferry functioning. Then make a discretization of the time axis so that we only consider the process at times  $t_k = kh$  for  $k = 0, 1, 2, \dots, N$ . Then the probability of a jump to another state during the time interval  $[t_k, t_{k+1}]$  can be determined (approximatively) using the intensity matrix for the continuous time Markov process. (Without explicitly using the exponential distributions.) If the time step  $h$  is small this approximation is good.

- 6a. Determine the transition matrix of a discrete time Markov chain that will approximate the continuous time one. Check if the transition matrix you obtain has row sums equal to one.
- 7a. Now use this discretization to simulate the process.  
By simulating the process we mean that you start the process in some state and then you determine one realization of the process by using the transition probabilities to determine which transitions occur. Choose  $N$  large enough so that the ergodic estimates does not vary more than 1%.

**8a.** take averages to determine estimates of the expected speed of the ferry.

If you use Matlab, a stochastic variable with uniform probability in  $[0, 1]$  can be generated using the command `rand`.

*The approach just described has also the purpose of showing the connection between discrete time and continuous time Markov chains.*

### Continuous time approach

It is of course also possible to simulate the continuous time process. Start the process at time 0 with both engines of the ferry functioning. Then you should use that the jump times are exponentially distributed. If you use Matlab an exponentially distributed variable can be generated using the command `expnd`, where you specify the mean of the variable (which is inversely proportional to the intensity). Then you use the exponential distribution to determine how long you remain in a state. You can use several exponential distributions, or one exponential distribution and a uniform probability generated using the command `rand`, to determine how long you remain in a state and which will be the next state.

**6b.** Describe in detail how you determine the time to the next jump and where you jump.

**7b.** Run the simulation for a time interval  $[0, T]$ .

By simulating the process we mean that you start the process in some state and then you determine one realization of the process by using the probability description of the process to determine how long you remain in a state and where you transition.

Choose  $T$  large enough so that the ergodic estimates does not vary more than 1%.

**8b.** take averages to determine estimates of the expected speed of the ferry.

### Hints

Since a large number of time steps have to be made to get reasonable ergodic estimates you might run into memory problems if you store all the old states in the simulations. Please note that it is enough to keep track on how long time the Markov chain has spent so far in each state to determine the estimates for the stationary probabilities and the average speed.

**9.** What is the computation time for achieving the 1% precision?  
(use `tic`, `toc` or `profile` to determine this)

**10.** A last assignment is given in the box below:

If  $\tau$  is the stochastic variable that describes the time to the next jump of the continuous time Markov process  $X(t)$  with intensity  $q$ , then we know that  $\tau$  must be  $\text{Exp}(q)$  distributed. That means that  $\tau$  has the distribution function  $F_\tau(t) = P(\tau \leq t) = 1 - e^{-qt}$ .

#### Show that

if  $\tau$  has the  $\text{Exp}(q)$  distribution, then  $P(\tau \geq 1) = P(\tau \geq 1.5 \mid \tau \geq 0.5)$ .

This is a property any time-homogeneous Markov process should have, why ?