Ferry maintenance Report

Ali Ghasemi
aghasemi@kth.se

Filippo Romano
froma@kth.se

December 4, 2024

1 Conceptual Framework

In this project, the ferry's operation is modeled as a continuous-time Markov process. And the velocity of the ferry is a stochastic process V_t with Markovian property.

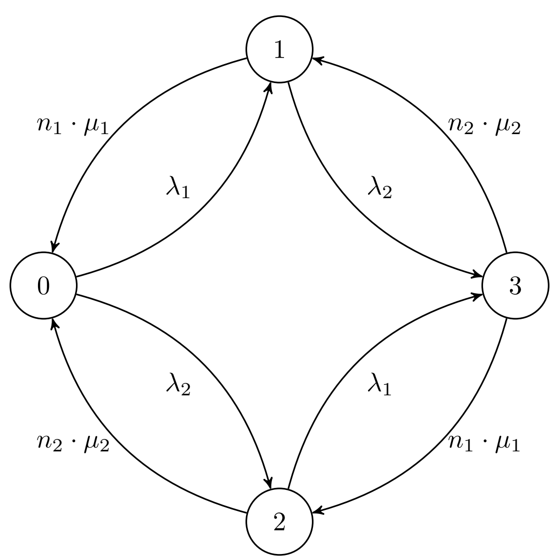
1. Defining The States of Markov Process

The ferry has two engines (with different powers), each drives a separate propeller. Based on the condition of engines and assumptions of the problem, the stochastic process can be in four states:

- Stage 0: Both engines are fully functional and the system operates at velocity v
- State 1: Only engine 2 is operational and the system operates at velocity v_2(failure of engine 1)
- State 2: Only engine 1 is operational and the system operates at velocity v_1 (failure of engine 2)
- State 3: System is Static $V_t = 0$ (failure of both engines)

2. Intensity Matrix

Based on the formulation of the problem and parameters the graphical representation of is as follows:



As seen in the graph, the model assumes that failure or completion of repair cannot happen at the same time (states 0 and 3 are non-adjacent). Furthermore, the completion of the repair of one engine and the failure of the other one cannot happen at the same time(states 1 and 2 are non-adjacent). Based on this model, the general shape of the intensity matrix is as follow:

$$Q = \begin{bmatrix} -(\lambda_1+\lambda_2) & \lambda_1 & \lambda_2 & 0 \\ 3\mu_1 & -(3\mu_1+\lambda_2) & 0 & \lambda_2 \\ 3\mu_2 & 0 & -(3\mu_2+\lambda_1) & \lambda_1 \\ 0 & n_2\mu_2 & n_1\mu_1 & -(n_1\mu_1+n_2\mu_2) \end{bmatrix} \tag{1}$$

Note that, based on the assumption of the problem if only one engine is non-operational, all three repairers should work on repairing that engine, in the first three rows the n factor for all the μ are 3. based on the each of four different repair strategies, we can have a unique intensity matrix. And they only differ from each other with respect to the last row:

- Repair Strategy 1 $\rightarrow Q_{\mathrm{I}}$ with $n_1 = 3$ and $n_2 = 0$
- Repair Strategy 2 $\rightarrow Q_{\mathrm{II}}$ with $n_1 = 2$ and $n_2 = 1$
- Repair Strategy 3 $\rightarrow Q_{\mathrm{III}}$ with $n_1 = 1$ and $n_2 = 2$
- Repair Strategy 4 $\rightarrow Q_{\mathrm{IV}}$ with $n_1 = 0$ and $n_2 = 3$

3. Existence of Stationary Distribution

(I) Since each state has multiple paths to return to itself with varying lengths, this system is aperiodic because the gcd of the return times for each state is 1.

(II) The system is recurrent since the expected time it takes for a process to return to each state is finite.

(III) As all states communicate with each other, the system is irreducible.

(IV) As a result of I and II, the system is ergodic.

(V) As a result of III and IV, there exists a stationary distribution for the system.

4. Stationary Distribution

The stationary distribution can be found by solving the steady-state equations:

$$\pi Q = 0 \quad \sum_{j=0}^{3} \pi_j = 1 \tag{2}$$

For each repair strategy, an intensity matrix is constructed Q that represents the transition rates between states. Then the steady-state distribution has been found by solving the linear system $\pi Q = 0$ using `null(Q')` to find the eigenvector corresponding to eigenvalue 0. The resulting vector was then normalized so that its entries sum to 1, giving the steady-state probabilities for each state. This steady-state distribution was computed for each strategy and used to analyze long-term system behavior.

5. Ferry's Average Speed

The expected value of the stochastic process (average speed) can be calculated using the multiplication of the stationary probability vector with the velocity vector.

$$V_{\mathrm{avg}} = nV_{\mathrm{mtrx}} \tag{3}$$

Where V_{mtrx} is the vector containing the values of each state (speed of the ferry).

$$V_{\mathrm{mtrx}} = \begin{pmatrix} v \\ v_2 \\ v_1 \\ 0 \end{pmatrix} \tag{4}$$

2 Continuous Time Approach

concerning this approach, only the first strategy is taken into account (corresponding to Q_I).

6a. Determining Time To Next Jump & Jump Direction

The time to the next jump can be modeled using dwell time (the time that the process remains in the current state until the next transition). Firstly, the transition time from state i to state j, modeled as a random variable with exponential distribution with the mean value inversely proportional to the transition intensity q_{ij}, is computed for the two possible transition directions. Once the transition times are computed, their values are compared, and the lower of the two becomes the dwell time. Consequently, the lower of the two transition times also indicates where the process jumps (e.g., if $T_{01} < T_{02}$, then the state switches from 0 to 1, whereas if $T_{01} > T_{02}$, then the process goes to state 2). The relation between the time to transition from i to j and dwell times discussed above can be summarized by:

$$E[T_{ij}] = \frac{1}{q_{ij}} \quad , \quad T_i = \min_{j \neq i} T_{ij} \tag{5}$$

7a. Simulation

The continuous-time Markov process was simulated by repeatedly determining the next state transition time using exponential distributions based on the transition rates. Starting from the initial state, the time spent in each state was recorded until the total simulated time reached a predefined threshold. By accumulating time in each state over numerous transitions, the state probabilities were estimated and subsequently compared to the analytic steady-state solutions. This approach enabled verification of the simulation results against the theoretical steady-state distribution.

8a. Expected Time To Failure

The expected time to failure was estimated by performing a simulation that began with the system in a fully functioning state and progressed until a failure state was reached, defined as the point at which all engines were non-operational. For each simulation iteration, the time required to reach the failure state was recorded.

This process was repeated 10,000 times to ensure statistical reliability. The recorded times to failure from each iteration were then averaged to provide an empirical estimate of the expected time to failure. This estimation approach utilized Monte Carlo simulation techniques, providing a robust approximation of the expected time to failure.

3 Discretization Approach

In this approach, we exclusively consider the first repair strategy, which corresponds to the intensity matrix Q_{I}.

6b. Determining Transition Matrix

The transition matrix for a Discrete-Time Markov chain, which effectively approximates the continuous-time process, can be derived using two different techniques.

First, by utilizing linear approximative relations, the transition matrix function P can be determined as follows:

$$P(h) = I + Qh$$

where h is the time step and I represents the identity matrix.
Alternatively, the matrix exponential function in MATLAB can be used to achieve the same result:

$$P(h) = e^{Qh}$$

Both methods should provide consistent results. It is crucial to note that the continuous-time process is accurately approximated only when h is sufficiently small.

7b./8b. Simulation & Expected Speed

The next step involves implementing the simulation for the discrete-time model. The simulation starts at time $t = 0$ in state 0. The process is observed at times $t = kh$ for $k = 1, 2, \ldots, N$, with N and h chosen to be sufficiently large and small, respectively, to ensure convergence and accurate approximation.

At each step, a random variable drawn from a uniform distribution between 0 and 1 is generated and compared against the probabilities in the approximated transition matrix. Specifically, the jump probability from state i to state j is denoted as p_{ij}, with the condition that $\sum_{j \neq i} p_{ij} = 1$ (i.e., all rows sum to 1). The probability space for a random variable is divided according to the cumulative sum of these probabilities: $p_{i1}, p_{i1} + p_{i2}, p_{i1} + p_{i2} + p_{i3}$, and so on, determining which state the process jumps to depending on the part of the space where the random variable falls.

Throughout each time step, the algorithm tracks the number of times each state is visited. At the end of the simulation, these counts are averaged with respect to the total number of visits to estimate the stationary state probabilities.

The expected speed is calculated similarly to the average speed calculation in Subsection 5. This involves multiplying the estimated stationary distribution vector by the velocity vector.

9b. Probability Questions

The first goal is to determine the probability that the process will reach state 3 (where no engines are operational) after 10 steps, starting from state 0 (where both engines are operational). To accomplish this, we use the ten-step transition matrix, which is obtained by raising the transition matrix P (used for continuous-time approximation) to the power of 10. According to the lecture theory, this is expressed as:

$$P^{(n)} = P^n$$

In this case, we compute P^{10}. The probability we are looking for, $p_{03}^{(10)}$, is located in the top right corner of the ten-step transition matrix.
The second goal is to determine the expected time to failure of the engines, the MATLAB code follows a structured approach using the theory of expected passage times.

First, the transition matrix P is constructed to approximate the continuous-time process. This is achieved through the linear approximation:

$$P = I + Q_i \cdot h$$

where Q_i is the intensity matrix, h is the time step, and I is the identity matrix. The matrix P is then reduced to consider only the first three states, excluding the final state.

Next, an identity matrix I of the same dimension as P is created. Using this identity matrix, the inverse of $I - P$ is calculated to obtain the matrix N:

$$N = (I - P)^{-1}$$

Finally, the expected time to failure is determined by summing the elements of the first row of N up to the third column. This result, represented by the formula:

$$ETtTF = \sum(N(1, 1:3))$$

provides the expected number of steps required to transition from the initial state to the final state. This calculation offers a precise estimation of the expected time to failure, using a methodical and robust approach based on stochastic process theory.

Figure 1: Matlab Grader Results