STK3405 – Lecture 16

A. B. Huseby

Department of Mathematics
University of Oslo, Norway



Section 7.2

Pure jump processes



Pure jump processes

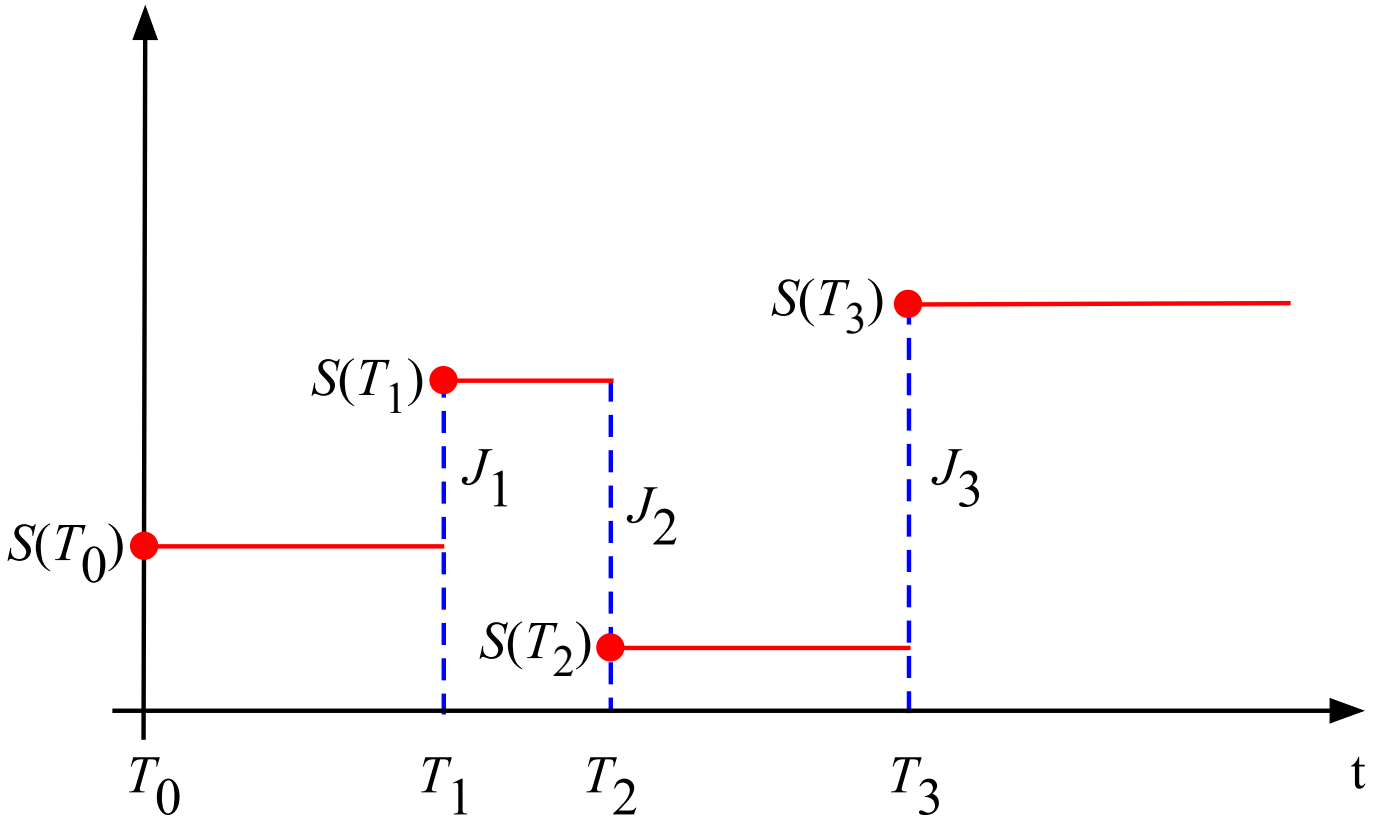


Figure: A pure jump process $S(t)$



Pure jump processes (cont.)

Let $\{S(t)\}$ be a stochastic process where $S(t)$ denotes the state of the process at time $t \geq 0$. $\{S(t)\}$ is said to be a *pure jump process* if $S(t)$ can be written as:

$$S(t) = S(0) + \sum_{j=1}^{\infty} I(T_j \leq t) J_j, \qquad t \geq 0,$$

where $0 = T_0 < T_1 < T_2 < \cdots$ is a sequence of random points of time, and $J_1, J_2, \ldots$ is a sequence of random (positive or negative) *jumps*.

In particular, for $k = 0, 1, \ldots$, we have:

$$S(t) = S(0) + \sum_{j=1}^{k} J_j = S(T_k), \qquad \text{for all } t \in [T_k, T_{k+1}).$$

From this it follows that the state function $S(t)$ is *piecewise constant* and *right-continuous* in t, with jumps at $T_1 < T_2 < \cdots$.

In order to keep track of how the process evolves only the *event points* need to be considered.



Regular pure jump processes

Let $\{S(t)\}$ be pure jump process, and let:

$$N(t) = \sum_{j=1}^{\infty} I(T_j \leq t)$$
$$= \text{The number of jumps in [0, t].}$$

We say that $\{S(t)\}$ is *regular* if $P(N(t) < \infty) = 1$ for all $t > 0$.

When running Monte Carlo simulations of pure jump processes over a given finite time frame $[0, t)$, regularity ensures that the simulations can be completed within a finite loop.



Regular pure jump processes (cont.)

Proposition (7.2.4)

Let $\{S_1(t)\}, \ldots, \{S_n(t)\}$ *be n regular pure jump processes, and let* $H(t) = H(\boldsymbol{S}(t))$, *where* $\boldsymbol{S}(t) = (S_1(t) \ldots, S_n(t))$, $t \geq 0$.

Then $\{H(t)\}$ *is a regular pure jump process as well.*

That is, $H(t) = H(\boldsymbol{S}(t))$ *is:*

- *Piecewise constant*
- *Right-continuous in* t,
- *The number of jumps in* $[0, t]$ *is finite with probability one for all* $t > 0$



Regular pure jump processes (cont.)

PROOF: Let $\mathcal{T}_i$ be the set of jump points of $\{S_i(t)\}$, $i = 1, \ldots, n$, and let $\mathcal{T}$ be the set of jump points of the process $\{H(t)\}$.

Since the state value of H cannot change unless there is a change in the state value of at least one of the elementary processes, it follows that $\mathcal{T} \subseteq (\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n)$.

Thus, $H(t)$ is piecewise constant and right-continuous in t. Moreover, for any finite interval $[0, t]$ we also have:

$$\mathcal{T} \cap [0, t] \subseteq [(\mathcal{T}_1 \cap [0, t]) \cup \cdots \cup (\mathcal{T}_n \cap [0, t])].$$

By regularity $(\mathcal{T}_i \cap [0, t])$ is finite almost surely for $i = 1, \ldots, n$. Hence, $\mathcal{T} \cap [0, t]$ is finite almost surely as well, implying that $\{H(t)\}$ is regular.



Section 7.3

Repairable binary monotone systems



Repairable binary monotone systems

Consider (C, ϕ), a binary monotone system of *n repairable* components.

Component state processes: $\{X_1(t)\}, \ldots, \{X_n(t)\}$, where:

$$X_i(t) = \text{the state of component } i \text{ at time } t \geq 0, \quad i \in C.$$

Component state vector: $\boldsymbol{X}(t) = (X_1(t), \ldots, X_n(t))$

System state process: $\{\phi(t)\}$, where:

$$\phi(t) = \phi(\boldsymbol{X}(t)) = \text{the state of the system at time } t \geq 0$$



Repairable components

We introduce the following random variables for all $i \in C$

- U_{ij} = The jth lifetime of the ith component, $\quad j = 1, 2, \ldots$
- D_{ij} = The jth repair time of the ith component, $\quad j = 1, 2, \ldots$

For all $i \in C$ we assume that:

- $U_{i1}, U_{i2}, \ldots$ are i.i.d. with mean value $0 < \mu_i < \infty$
- $D_{i1}, D_{i2}, \ldots$ are i.i.d with mean value $0 < \nu_i < \infty$

All lifetimes and repair times are assumed to be independent. Thus, in particular the component processes $\{X_1(t)\}, \ldots, \{X_n(t)\}$ are independent of each other.

NOTE: The U_{ij}'s and the D_{ij}'s are waiting times between state changes for the components.



Repairable components

We now focus on the jump points of the state process of component i.

- The first jump happens after the first lifetime of the component, i.e., at time U_{i1}. This is a jump downwards from state 1 to state zero, so $J_1 = -1$.
- The second jump happens after the first repair time of the component, i.e., at time $U_{i1} + D_{i1}$. This is a jump upwards from state zero to state 1, so $J_2 = +1$.
- The third jump happens after the second lifetime of the component, i.e., at time $U_{i1} + D_{i1} + U_{i2}$. This is another jump downwards from state 1 to state zero, so $J_3 = -1$.
- This pattern is then repeated over and over again.



Repairable components

We denote the jump points by $T_{i1}, T_{i2}, \ldots$. These jump points can be calculated from the lifetimes and the repair times as follows:

$$
\begin{aligned}
T_{i,1} &= U_{i1}, \\
T_{i,2} &= U_{i1} + D_{i1} \\
T_{i,3} &= U_{i1} + D_{i1} + U_{i2}, \\
T_{i,4} &= U_{i1} + D_{i1} + U_{i2} + D_{i2} \\
&\ldots
\end{aligned}
$$

We recall that the jumps of the component state process is an alternate sequence of (-1)'s and $(+1)$'s. Thus, it follows that the jumps, $J_1, J_2, \ldots$, can be expressed as:

$$J_j = (-1)^j, \quad j = 1, 2, \ldots$$

We may then write:

$$X_i(t) = X_i(0) + \sum_{j=1}^{\infty} I(T_{ij} \leq t) J_j, \quad i = 1, \ldots, n.$$



Availability

Let $A_i(t)$ be the availability of the ith component at time t. That is, for $i = 1, \ldots, n$ we have:

$$A_i(t) = \Pr(X_i(t) = 1) = E[X_i(t)].$$

By renewal theory the corresponding stationary availabilities are given by:

$$A_i = \lim_{t \to \infty} A_i(t) = \frac{\mu_i}{\mu_i + \nu_i}, \qquad i = 1, \ldots, n.$$

Introduce $\boldsymbol{A}(t) = (A_1(t), \ldots, A_n(t))$ and $\boldsymbol{A} = (A_1, \ldots, A_n)$. The system availability at time t is given by:

$$A_\phi(t) = \Pr(\phi(\boldsymbol{X}(t)) = 1) = E[\phi(\boldsymbol{X}(t))] = h(\boldsymbol{A}(t)),$$

where h is the system's reliability function. The corresponding stationary availability is given by:

$$A_\phi = \lim_{t \to \infty} A_\phi(t) = h(\boldsymbol{A})$$



Criticality

The component i is said to be *critical* at time t if

$$\psi_i(\boldsymbol{X}(t)) = \phi(1_i, \boldsymbol{X}(t)) - \phi(0_i, \boldsymbol{X}(t)) = 1.$$

$\psi_i(\boldsymbol{X}(t))$ is the *criticality state* of component i at time t.

The Birnbaum measure of importance of component i at time t, $I_B^{(i)}(t)$, is the probability that i is critical at time t:

$$\begin{aligned} I_B^{(i)}(t) &= \Pr(\psi_i(\boldsymbol{X}(t)) = 1) = E[\psi_i(\boldsymbol{X}(t))] \\ &= h(1_i, \boldsymbol{A}(t)) - h(0_i, \boldsymbol{A}(t)). \end{aligned}$$

The corresponding stationary measure is given by:

$$I_B^{(i)} = \lim_{t\to\infty} I_B^{(i)}(t) = h(1_i, \boldsymbol{A}) - h(0_i, \boldsymbol{A}).$$



Section 7.5

Exercises



Exercise 7.1

Exercise 7.1: Assume that T is Weibull distributed with parameters $\alpha > 0$ and $\beta > 0$.

a) Show that:

$$E[T] = \beta \cdot \Gamma(\frac{1}{\alpha} + 1),$$

where:

$$\Gamma(a) = \int_0^\infty x^{a-1} e^{-x} dx, \quad a > 0.$$



Exercise 7.1 (cont.)

Hint: Use that:

$$a \cdot \Gamma(a) = \Gamma(a+1)$$

and the following lemma:

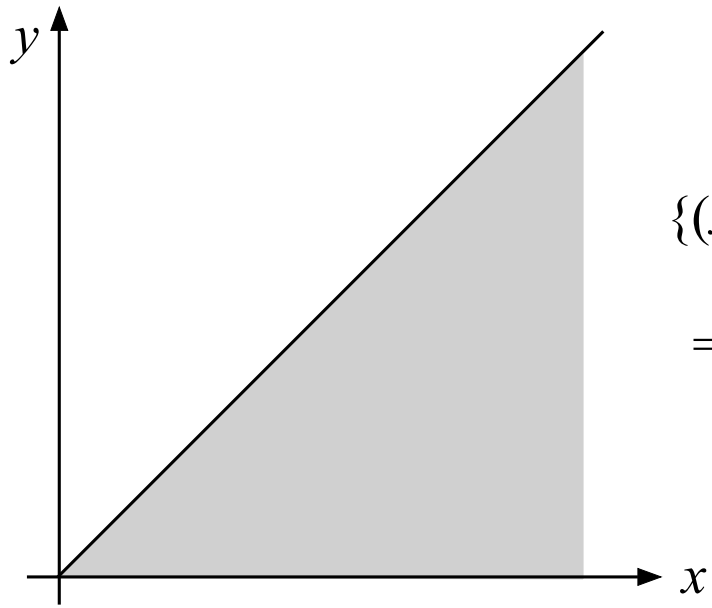
Lemma (Expected value of a non-negative variable)

Let X be a non-negative absolutely continuously distributed random variable with density $f(x)$. Then the expected value of X is given by:

$$E[X] = \int_0^\infty P(X > y)\, dy.$$



Exercise 7.1 (cont.)



$$\{(x,y):\ 0 \le x < \infty,\ 0 \le y \le x\}$$

$$= \{(x,y):\ 0 \le y < \infty,\ y \le x < \infty\}$$

PROOF: We start out by noting that:

$$E[X] = \int_0^\infty x f(x) dx = \int_0^\infty [\int_0^x dy\,]\, f(x)\, dx$$

We then interchange the integrals and get:

$$E[X] = \int_0^\infty [\int_y^\infty f(x) dx\,]\, dy = \int_0^\infty P(X > y)\, dy$$



Exercise 7.1 (cont.)

We recall that if T is Weibull distributed with parameters α and β, then:

$$P(T > t) = e^{-(t/\beta)^{\alpha}}, \quad t \geq 0.$$

Hence, by using the lemma we get:

$$E[T] = \int_0^{\infty} P(T > t)dt = \int_0^{\infty} e^{-(t/\beta)^{\alpha}} dt$$

We then substitute $u = (t/\beta)^{\alpha}$. Thus, $du = \frac{\alpha}{\beta}(t/\beta)^{\alpha-1}dt = \frac{\alpha}{\beta}u^{1-\frac{1}{\alpha}}dt$, and hence, by solving the last equation with respect to dt:

$$dt = \frac{\beta}{\alpha}u^{\frac{1}{\alpha}-1}du$$

$$E[T] = \frac{\beta}{\alpha}\int_0^{\infty} u^{\frac{1}{\alpha}-1}e^{-u}du = \beta \cdot \frac{1}{\alpha} \cdot \Gamma(\frac{1}{\alpha}) = \beta \cdot \Gamma(\frac{1}{\alpha}+1)$$



Exercise 7.1 (cont.)

b) Show that:

$$Med[T] = \beta \cdot [\ln(2)]^{1/\alpha}$$

Let $\bar{F}(t)$ denote the survival function of the Weibull distribution with parameters α and β. $Med[T]$ is the solution to the equation:

$$\bar{F}(Med[T]) = e^{-(Med[T]/\beta)^{\alpha}} = \frac{1}{2}$$

This equation is equivalent to:

$$-(Med[T]/\beta)^{\alpha} = -\ln(2)$$

which has the solution:

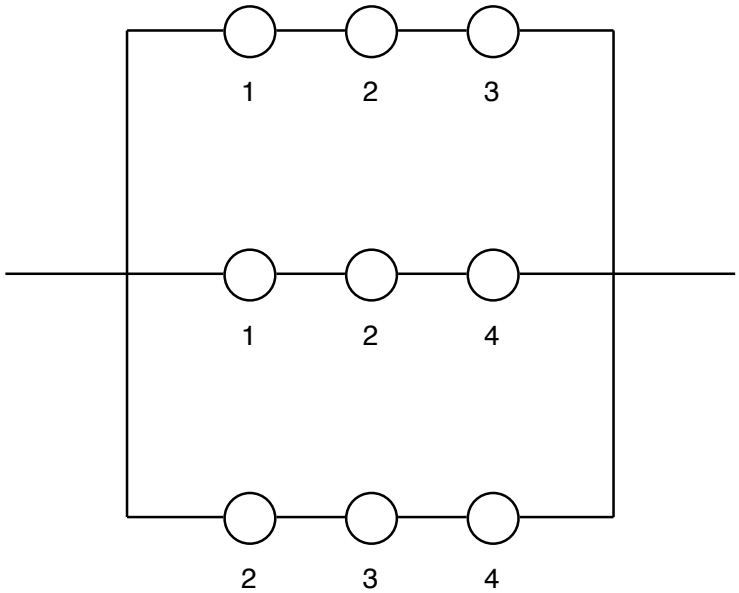
$$Med[T] = \beta \cdot [\ln(2)]^{1/\alpha}$$



Exercise 7.2

Exercise 7.2: Let (C, ϕ) be the binary monotone system shown in the figure. The lifetime of component i, T_i is Weibull-distributed with parameters a_i and b_i, for all $i \in C$ where:

$$a_1 = 2.0, \quad a_2 = 2.5, \quad a_3 = 3.0, \quad a_4 = 1.0,$$
$$b_1 = 50.0, \quad b_2 = 60.0, \quad b_3 = 70.0, \quad b_4 = 40.0.$$





Exercise 7.2 (cont.)

a) Find the structure function of the system.

SOLUTION: We observe that component 2 occurs in all three branches in the reliability block diagram. Thus, this component is in series with the rest of the system.

Given that component 2 is functioning, the remaining system is a 2-out-of-3 system consisting of the components 1, 3 and 4.

Hence, the structure function of the system is:

$$\phi(\boldsymbol{x}) = x_2 \cdot (x_1 x_3 + x_1 x_4 + x_3 x_4 - 2x_1 x_3 x_4)$$



Exercise 7.2 (cont.)

b) Consider the python script B.3.1. Modify this script so that you can use it to calculate $h(\boldsymbol{p}(t))$ for the system shown in the figure.

Save the resulting plot to a file.



Exercise 7.2 (cont.)

```
from math import *
import matplotlib.pyplot as plt
import numpy as np

save_plots = True

# Time parameters
max_t = 120
steps = 120

# Weibull-parameters for the components
alpha = [2.0, 2.5, 3.0, 1.0]
beta = [50.0, 60.0, 70.0, 40.0]

# Component reliability vector
num_comps = 4
p = np.zeros(num_comps)

# System reliabilities
h = np.zeros(steps)


def reliability(pp):
    return pp[1] * (pp[0] * pp[2] + pp[0] * pp[3] + pp[2] * pp[3] - 2 * pp[0] * pp[2] * pp[3])

# See: https://en.wikipedia.org/wiki/Weibull_distribution
def weibull(aa, bb, t):
    if t >= 0:
        return exp(- (t / bb) ** aa)
    else:
        return 1.0
```



Exercise 7.2 (cont.)

```
# See: https://en.wikipedia.org/wiki/Weibull_distribution
def weibull(aa, bb, t):
    if t >= 0:
        return exp(- (t / bb) ** aa)
    else:
        return 1.0

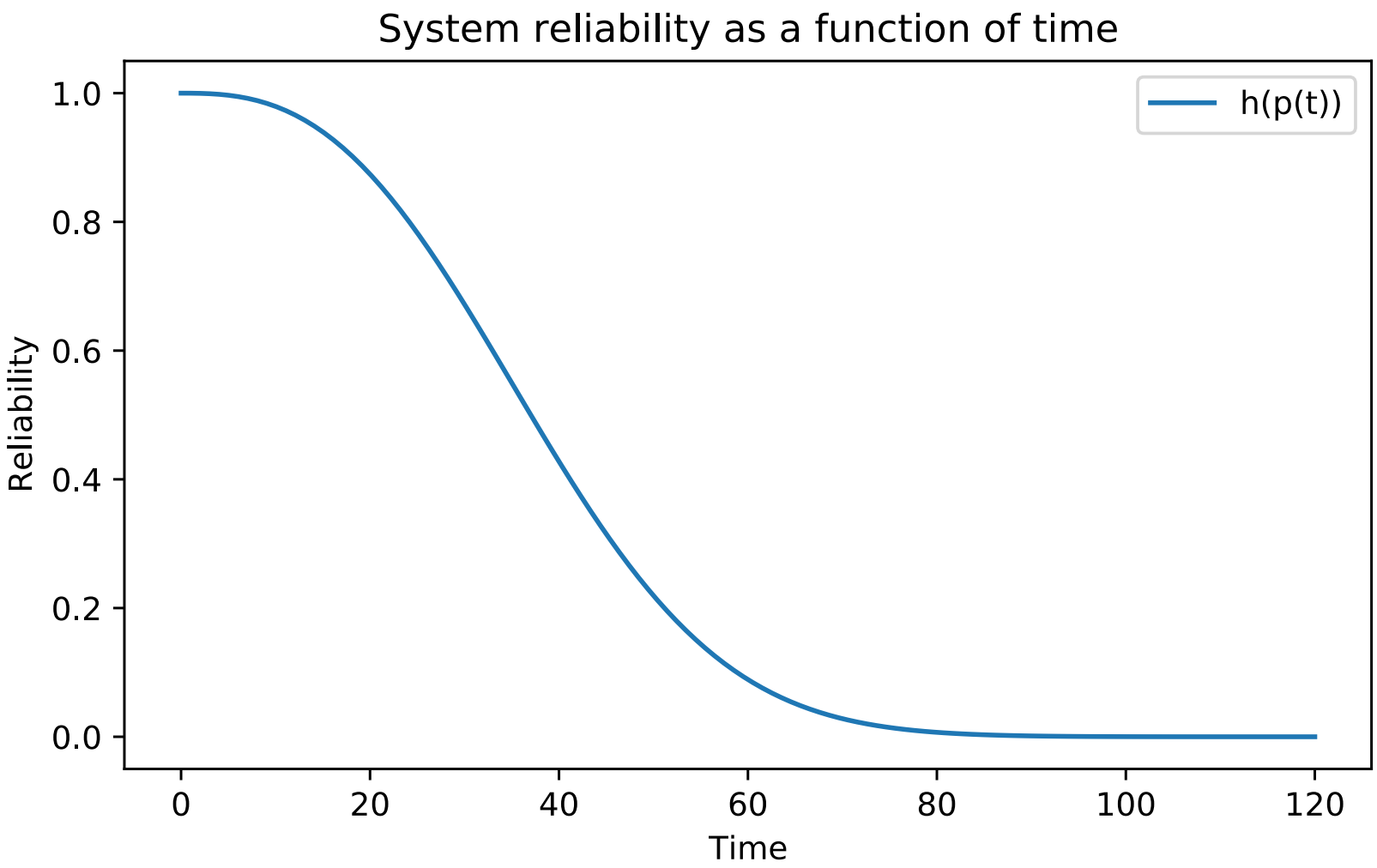

time = np.linspace(0.0, max_t, steps)

for i in range(steps):
    for j in range(num_comps):
        p[j] = weibull(alpha[j], beta[j], time[i])
    h[i] = reliability(p)

fig = plt.figure(figsize = (7, 4))
plt.plot(time, h, label='h(p(t))')
plt.xlabel('Time')
plt.ylabel('Reliability')
plt.title("System reliability as a function of time")
plt.legend()
if save_plots:
    plt.savefig("dynsys_01b/reliability_01b.pdf")
plt.show()
```



Exercise 7.2 (cont.)





Exercise 7.2 (cont.)

c) Find the minimal path and cut sets of the system.

SOLUTION: The minimal path sets are:

$$P_1 = \{1, 2, 3\}, \quad P_2 = \{1, 2, 4\}, \quad P_3 = \{2, 3, 4\}$$

The minimal cut sets are:

$$K_1 = \{2\}, \quad K_2 = \{1, 3\}, \quad K_3 = \{1, 4\}, \quad K_4 = \{3, 4\}$$



Exercise 7.2 (cont.)

d) Consider the python script B.3.2. Modify this script so that you can use it to estimate $h(\boldsymbol{p}(t))$ for the system shown in the figure.

Save the resulting plot to a file.

Compare the resulting plot to the one you found in (b).



Exercise 7.2 (cont.)

```
from math import *
from random import *
import matplotlib.pyplot as plt
import numpy as np

save_plots = True

# Number of simulations
num_sims = 100000

# Weibull-parameters for the components
alpha = [2.0, 2.5, 3.0, 1.0]
beta = [50.0, 60.0, 70.0, 40.0]

# Number of components and minimal paths
num_comps = 4
num_paths = 3


# Minimal path sets
paths = [{1,2,3}, {1,2,4}, {2,3,4}]


# Calculate the system lifetime given the component lifetimes
def sys_lifetime(tt):
    sys_life = 0.0
    for j in range(num_paths):
        path_life = np.inf
        for i in paths[j]:
            if tt[i-1] < path_life:
                path_life = tt[i-1]
        if path_life > sys_life:
            sys_life = path_life
    return sys_life
```



Exercise 7.2 (cont.)

```
# The upper percentile levels
q = np.zeros(num_sims)

# The simulated system lifetimes
s = np.zeros(num_sims)

# The simulated lifetimes of the components
t = np.zeros(num_comps)

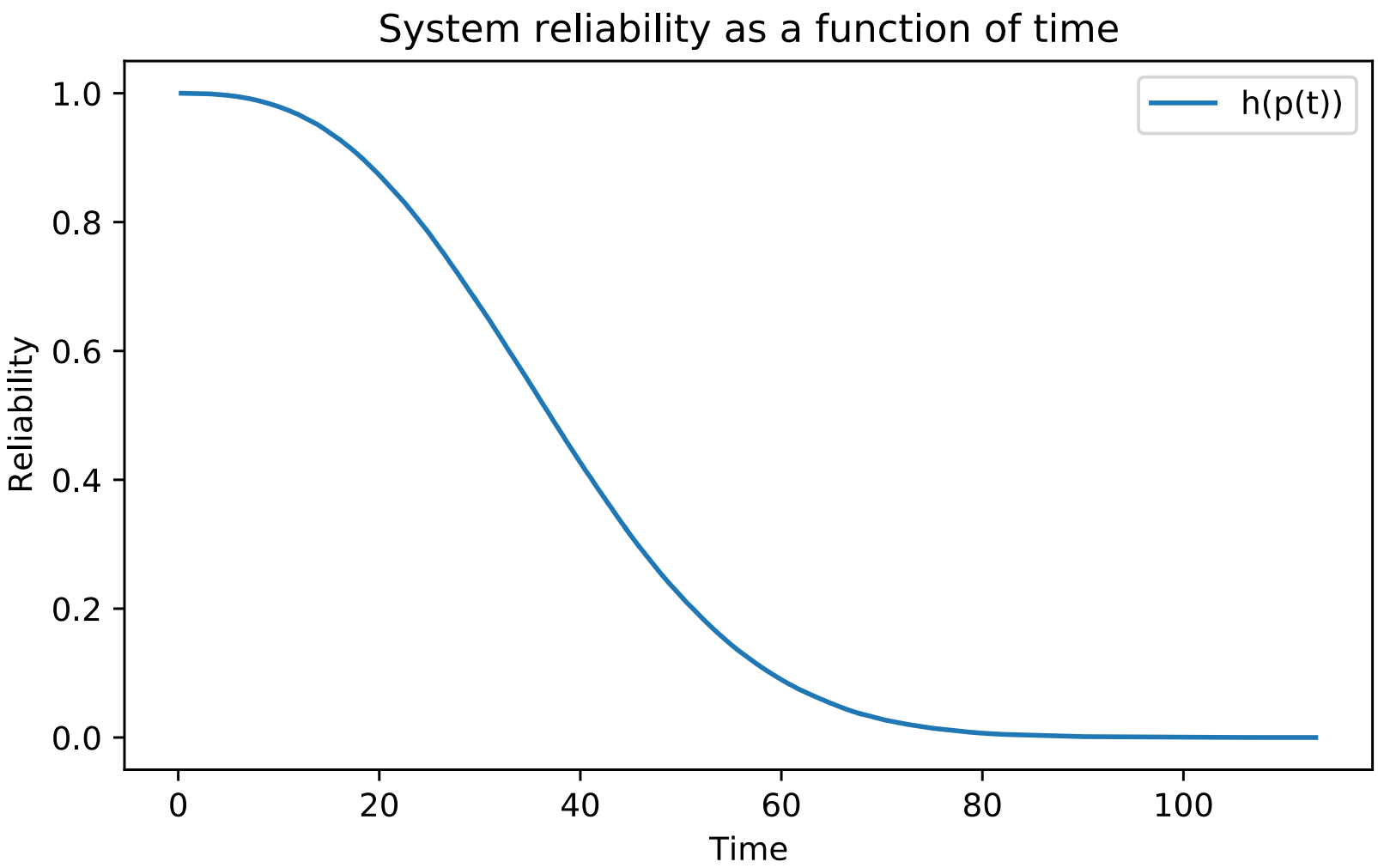
for k in range(num_sims):
    q[k] = 1.0 - k / num_sims
    for i in range(num_comps):
        t[i] = weibullvariate(beta[i], alpha[i])
    s[k] = sys_lifetime(t)

s.sort()

fig = plt.figure(figsize = (7, 4))
plt.plot(s, q, label='h(p(t))')
plt.xlabel('Time')
plt.ylabel('Reliability')
plt.title("System reliability as a function of time")
plt.legend()
if save_plots:
    plt.savefig("dynsys_02b/reliability_02b.pdf")
plt.show()
```



Exercise 7.2 – Analytical calculations





Exercise 7.2 – MC simulations

