
# Monte Carlo Simulations: Exercise Session

Ibrahim Ahmed & Stefano Marchetti

April 19th 2024

# EXERCISE 1

# Exercise 1



Consider the Weibull distribution:

$$f_T(t) = \frac{\beta}{\tau}\left(\frac{t}{\tau}\right)^{\beta-1} e^{-\left(\frac{t}{\tau}\right)^{\beta}} \qquad F_T(t) = 1 - e^{-\left(\frac{t}{\tau}\right)^{\beta}}$$

with $\beta = 1{,}5$ and $\tau = 1{,}0$

1. Sample *N*=400 values from $f_T(t)$
2. Verify whether the obtained distribution provides a good approximation of the Weibull distribution. To this aim, you are required to:
   A. find the empirical probability density function (pdf) of the sampled values in 1
   B. compare the empirical pdf found in 2A. with the analytical Weibull distribution.

# Point 1: sampling



```
%% Exercise 1
clear all
close all
clc
% Weibull parameters
tau = 1;
beta = 1.5;

% Sample N values from the Weibull distribution
N = 400; % number of samples
r = rand(N,1);
t = tau*(-log(1-r)).^(1/beta); % inverse transform method
```

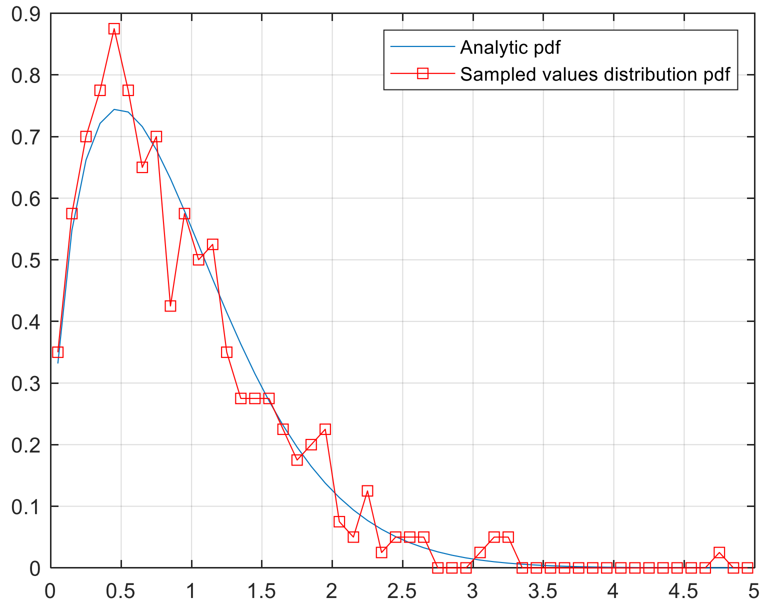
# Point 2: verify the distribution



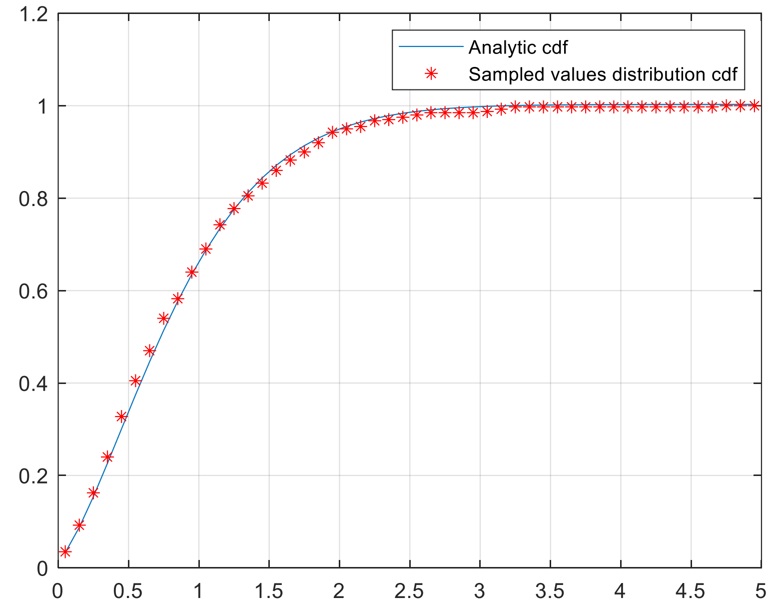
```
% Verify the distribution
% Estimated pdf
delta_t = 0.1;
edges=0:delta_t:5;
interval_centers=delta_t/2:delta_t:5-delta_t/2;
counts = histcounts(t,edges);
pdf_est = counts./(N*delta_t);

% Compute the analytic pdf
analytic_weibull = (beta/tau*(interval_centers/tau).^(beta-1)).*exp(-(interval_centers/tau).^beta);

% plot the pdf
figure % pdf
plot(interval_centers, analytic_weibull); hold on; plot(interval_centers, pdf_est,'-sr'); grid
legend('Analytic pdf','Sampled values distribution pdf')

% plot the cdf
figure
cdf_est = cumsum(pdf_est)*delta_t;
plot(interval_centers, cumsum(analytic_weibull)*delta_t);hold on; plot(interval_centers, cdf_est,'*r'); grid
legend('Analytic cdf','Sampled values distribution cdf')
```

# Point 2: verify the distribution

## PDF




## CDF



# EXERCISE 1
# part 2

# Exercise 1

Consider the Weibull distribution:

$$f_T(t) = \frac{\beta}{\tau}\left(\frac{t}{\tau}\right)^{\beta-1} e^{-\left(\frac{t}{\tau}\right)^{\beta}} \qquad F_T(t) = 1 - e^{-\left(\frac{t}{\tau}\right)^{\beta}}$$

with $\beta = 1{,}5$ and $\tau = 1{,}0$

1. Sample *N*=400 values from $f_T(t)$
2. Verify whether the obtained distribution provides a good approximation of the Weibull distribution. To this aim, you are required to:
   A. find the empirical probability density function (pdf) of the sampled values in 1
   B. compare the empirical pdf found in 2A. with the analytical Weibull distribution.
3. Provide an estimate $G_N$ of $\int_0^{+\infty} t f_T(t) dt$
4. Estimate the variance of $G_N$

# Point 3 and point 4: estimator

## MATLAB

```
% Estimating the mean of the distribution (definite integral)
GN = mean(t); % Sample mean
G=tau/beta*gamma(1/beta); %real value
disp(['G_N = ',num2str(GN)])


% Estimating the variance of the distribution (definite integral)
S2GN = var(t)/N;
disp(['G_N variance = ', num2str(S2GN)])
disp(['Estimate = ',num2str(GN),' +- ',num2str(sqrt(S2GN))]);
disp([' '])
disp(['True Value = ', num2str(G)])
% mean_weibull=tau/beta*gamma(1/beta)
disp([' '])
disp(['Error = ', num2str(abs(G-GN))])
```

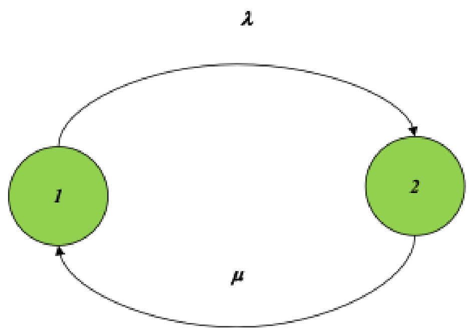
# EXERCISE 2

# Exercise 2



Consider a continuously monitored component with constant failure ($\lambda$) and repair ($\mu$) rates in the table. Assuming a mission time $T = 1000\ hours$, write the MC code for the estimation of:

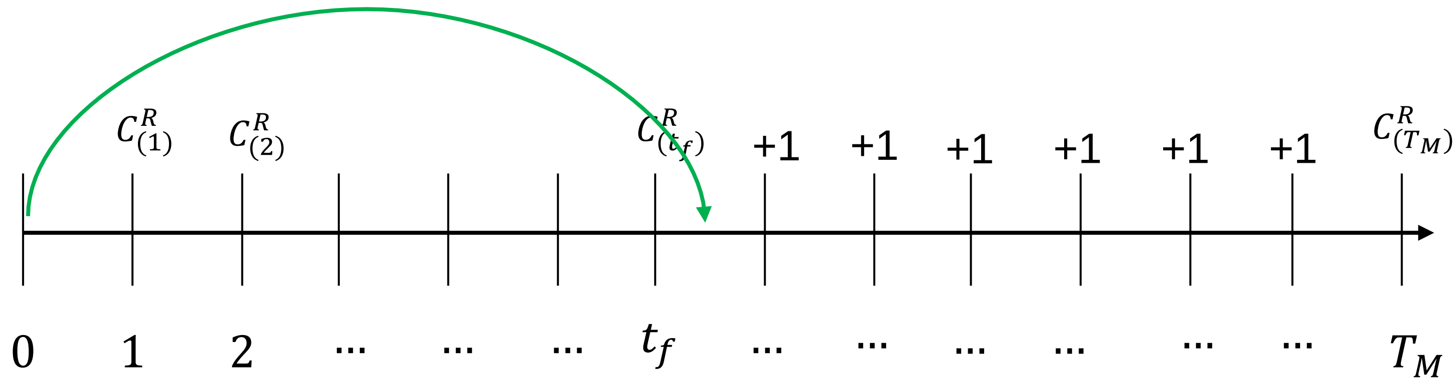
1. The instantaneous availability
2. The time dependent reliability



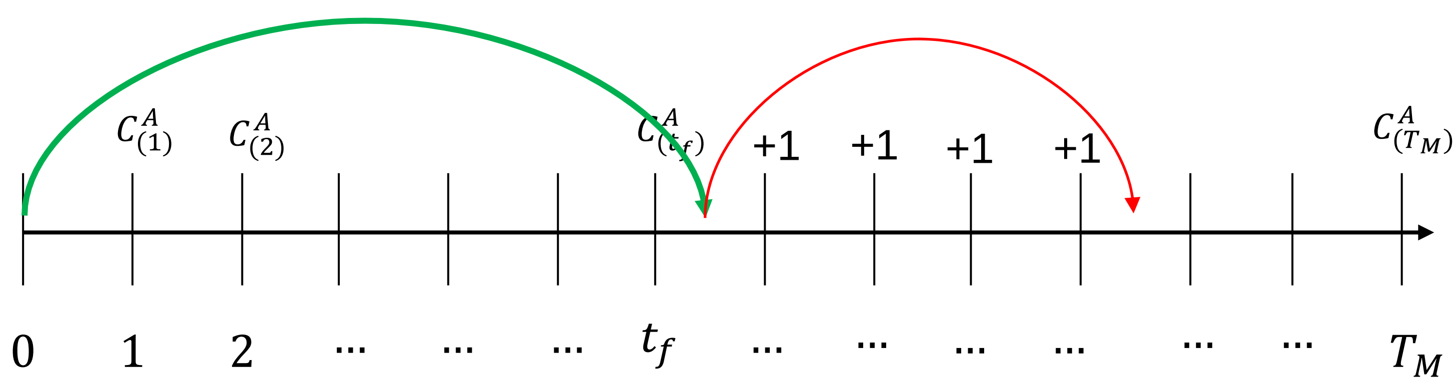

| | values |
|---|---|
| λ | $3 \cdot 10^{-3}\ h^{-1}$ |
| μ | $25 \cdot 10^{-3}\ h^{-1}$ |

# Exercise 2 – How to update the counters

## Estimation of the System Reliability




## Estimation of the System Availability



# MATLAB

```
clear all
close all
clc

%Initialize parameters
Tm=10^3; %mission time;
M=10^5; %number of trials;
lambda=3e-3;
mu=25e-3;
Dt=1; %bin length;
Time_axis=0:Dt:Tm;
counter_q=zeros(1,length(Time_axis));
```

# Exercise 2: Availability: MC



```
for i=1:M %simulation of M MC trials
    %parameter initialization for each trial
    t=0;
    state=0; %state=0 = working;
    %state=1 = failed;
    while t<Tm
        if state == 0 %system is working ? sampling of failure time
            t=t-[log(1-rand)]/lambda; %failure time
            failure_time=t;
            state=1; % new state = failed
            lower_b=find(Time_axis >= failure_time,1,'first'); % first unavailability counter to be increased
        else %system is failed then sampling of repair time
            t=t-[log(1-rand)]/mu; %repair_time
            state=0;
            repair_time=t;
        if t<Tm
            upper_b=max(find(Time_axis < repair_time)); %last unavailability counter to be increased
        else %repair ends after mission time
            upper_b=length(Time_axis);
        end
            counter_q(lower_b:upper_b)= counter_q(lower_b:upper_b)+1; %increase all unavailability counter between lower_b and
            %upper_b
        end
    end
end

Ava_MC=1-counter_q/M;
```

# Exercise 2: Reliability - initialize

## MATLAB

```
clear all
close all
clc

%Initialize parameters
Tm=10^3; %mission time;
M=10^5; %number of trials;
lambda=3e-3;
mu=25e-3;
Dt=1; %bin length;
Time_axis=0:Dt:Tm;
counter_f=zeros(1,length(Time_axis));
```

# Exercise 2: Reliability: MC



```
%Initialize parameters
Tm=10^3; %mission time;
M=10^5; %number of trials;
lambda=3e-3;
mu=25e-3;
Dt=1; %bin length;
Time_axis=0:Dt:Tm;
counter_f=zeros(1,length(Time_axis));


for i=1:M
        t=0;
        t=t-[log(1-rand)]/lambda;
        counter_f(ceil(t):end)=counter_f(ceil(t):end)+1;
end

%Monte Carlo Reliability;
Rel_MC=1-counter_f/M;

%Real Reliability
Rel_true=exp(-lambda*Time_axis);
```

## MATLAB

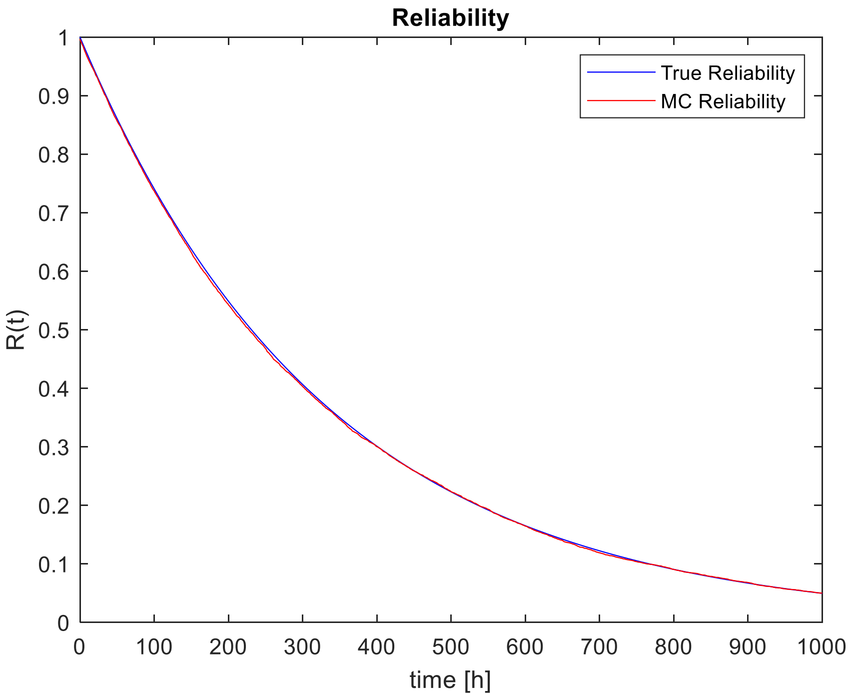
```
Ava_MC=1-counter_q/M;
Ava_true=(mu)/(lambda+mu)+(lambda/(lambda+mu))*exp(-(lambda+mu).*Time_axis);


figure
plot(Time_axis,Ava_true,'blue')
hold on
plot(Time_axis,Ava_MC,'red')

title("Availability")
legend("True Availability","MC availability")

xlabel("time [h]")
ylabel("A(t)")
```

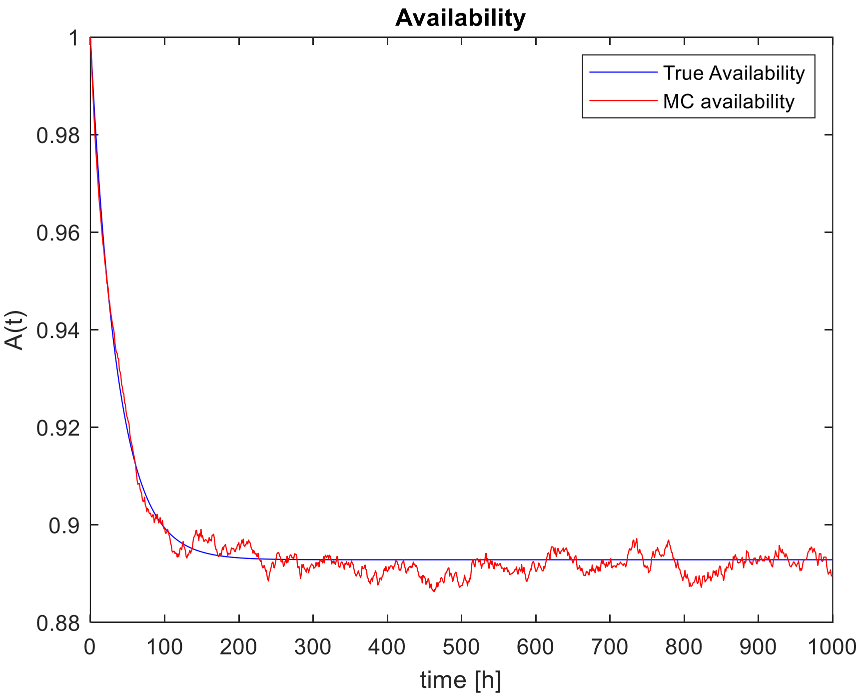
## MATLAB

```
Rel_MC=1-counter_f/M;
Rel_true=exp(-lambda*Time_axis);


figure
plot(Time_axis,Rel_true,'blue')
hold on
plot(Time_axis,Rel_MC,'red')
title("Reliability")
legend("True Reliability","MC Reliability")
xlabel("time [h]")
ylabel("R(t)")
```

# Exercise 2 – Plots

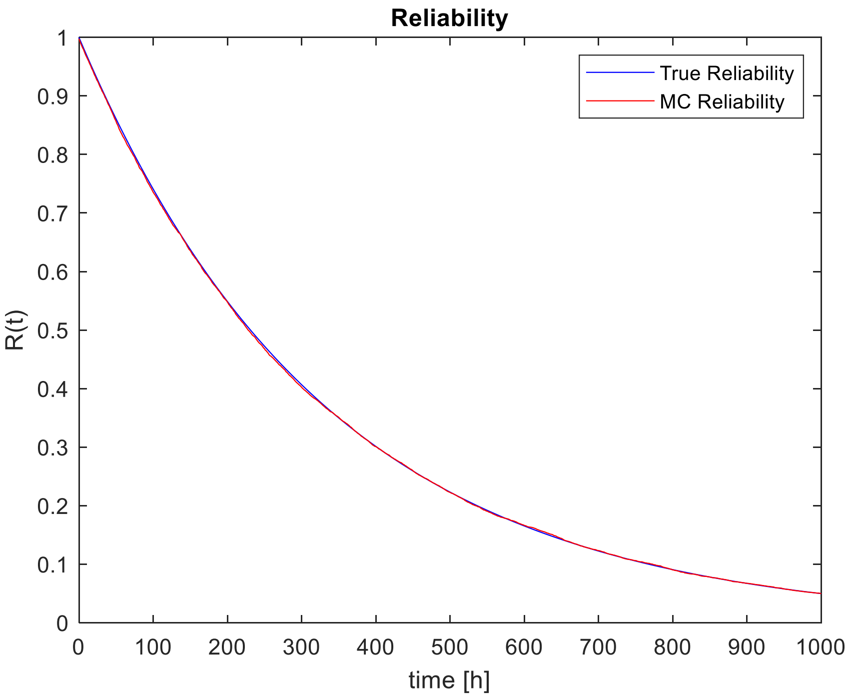
## M = 10000

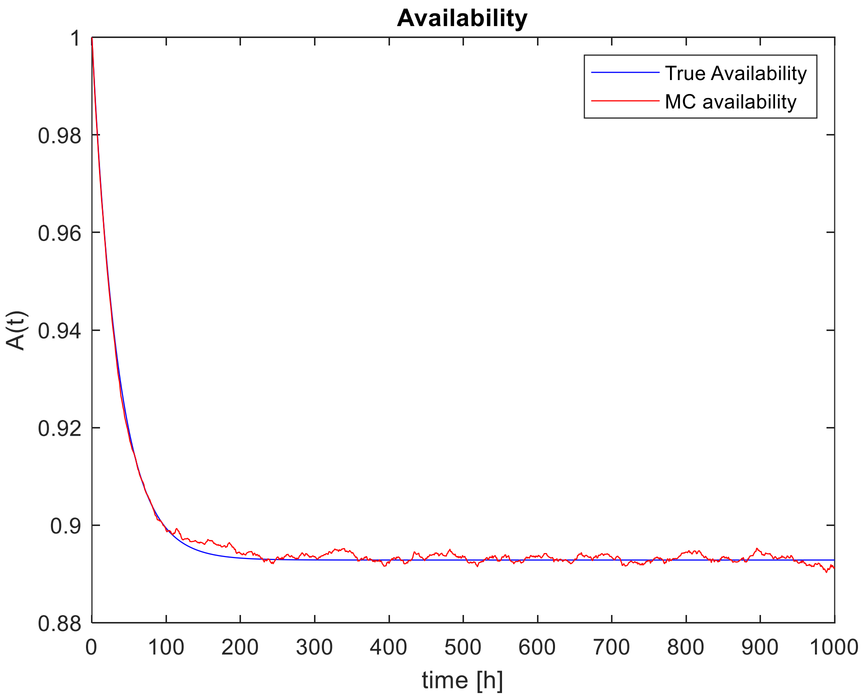


# Exercise 2 – Plots

## M = 100000



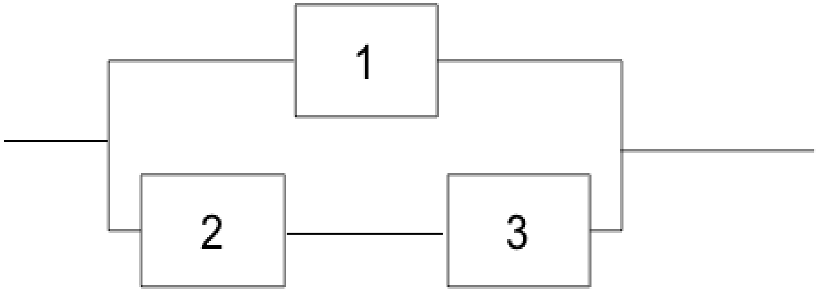
# EXERCISE 3

# Exercise 3



Consider the system in figure composed of three components(A, B, C). Each component can be in two different health states (1-nominal, 2-failed) with exponentially distributed transition times (table) between them. Assuming a mission time $T = 500\ hours$, write the MC code for the estimation of:

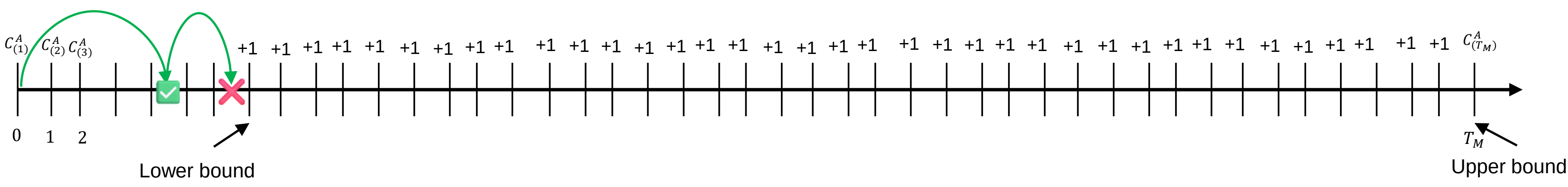
- The time dependent reliability
- The instantaneous availability.
- The estimators uncertainty



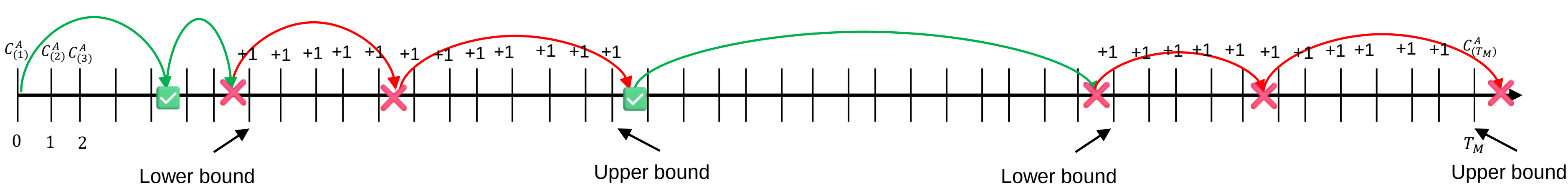

| | 1 | 2 | 3 |
|---|---|---|---|
| $\lambda$ | $1 \cdot 10^{-3}\ h^{-1}$ | $2 \cdot 10^{-2}\ h^{-1}$ | $5 \cdot 10^{-2}\ h^{-1}$ |
| $\mu$ | $3 \cdot 10^{-2}\ h^{-1}$ | $5 \cdot 10^{-2}\ h^{-1}$ | $5 \cdot 10^{-3}\ h^{-1}$ |

# Exercise 3 – How to update the counters

## Estimation of the System Reliability




## Estimation of the System Availability



# Exercise 3: initializing



MATLAB

```
%% Reliability/Availability of Simple Systems
clc
clear all
close all
% choice = input('1 --> Simulation mode 2 --> Step by step mode');
% System parameters
lambda_A= 1e-3; mu_A= 3e-2;
lambda_B= 2e-2; mu_B= 5e-2;
lambda_C= 5e-2; mu_C= 5e-3;

Trans_A =[0 lambda_A; mu_A 0];
Trans_B= [0 lambda_B; mu_B 0];
Trans_C= [0 lambda_C; mu_C 0];

% 2 = Failure; 1 = Working
failed_states=[2 2 1; 2 1 2; 2 2 2];
initial_state=[1 1 1];
% Missiontime
Tm=500;
% MC simulationparameters
N = 1e5;
Dt=1;
Time_axis =0:Dt:Tm;
% MC cycle
unrel_counter = zeros(1,N);
unrel = zeros(N,length(Time_axis));
counter_q=zeros(N,length(Time_axis));
```

# Exercise 3: MC (part 1)

**MATLAB**

```
for n = 1:N % Main Monte Carlo cycle
unrel_flag = 0; % 0 if no failures before Tmiss, 1 otherwise
t = 0;
current_state = initial_state;
system_state = 1; %working
while (t <= Tm)
% find the system transition rate (Lambda_sys)
lambda_out(1) = sum(Trans_A(current_state(1),:)); %transition rate of component A
lambda_out(2) = sum(Trans_B(current_state(2),:)); %transition rate of component B
lambda_out(3) = sum(Trans_C(current_state(3),:)); %transition rate of component C

lambda_sys = sum(lambda_out); %transition rate of the system

% Sample transition time (t_trans)
t_trans = -1/lambda_sys*log(rand); %transition time
t = t+t_trans; %current time
if t<Tm % if the transition time falls within Tmiss ? Sample the kind of transition
r = rand;
sum_l = cumsum(lambda_out)/lambda_sys;
comp = find(sum_l>r,1,'first'); % Component that makes the transition
%Change the state of the component that makes the transition
old_st = current_state(comp);
current_state(comp) = 3 - old_st;
end
```

# Exercise 3: MC (part 2)



MATLAB

```
% check if the system is in a failure configuration
for jj = 1:size(failed_states,1) % for each one of the failure states:
if sum(current_state == failed_states(jj,:)) == size(failed_states,2)
failure_flag = 1;
break;
else
failure_flag=0;
end
end
if failure_flag == 1
if unrel_flag==0 % only at the first failure
unrel_flag = 1;
unrel(n,ceil(t/Dt):end)=1; %for reliability estimation
end
if system_state == 1 % if the system was working
failure_time = t;
lower_b=find(Time_axis >= failure_time,1,'first'); %for availability estimation
system_state = 0;
end
else
if system_state == 0 %if the system was failed
repair_time = t;
system_state = 1;
upper_b=min([length(Time_axis),find(Time_axis < repair_time,1,'last')]); %for availability estimation
%increase all unavailability counter between lower_b and upper_b
counter_q(n,lower_b:upper_b)= counter_q(n,lower_b:upper_b)+1;
end
end
end %(of the while loop)
% if the repair ends after the mission time
if (system_state==0 && failure_flag==1)
counter_q(n,lower_b:end)= counter_q(n,lower_b:end)+1;
end
unrel_counter(n) = unrel_flag;
end % end of the N simulation
```

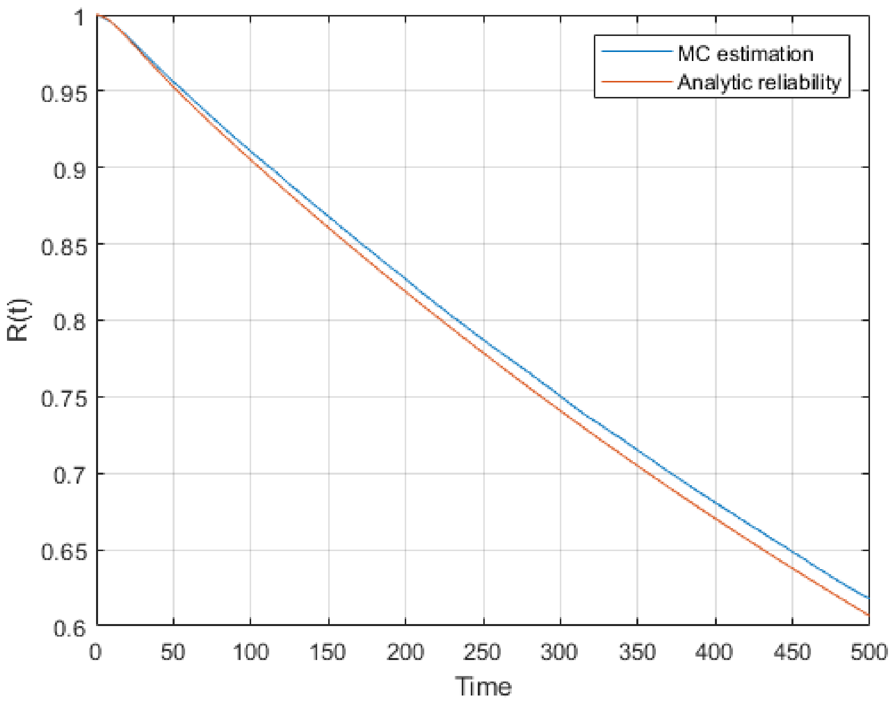
# Exercise 3 – MC and results

## MATLAB

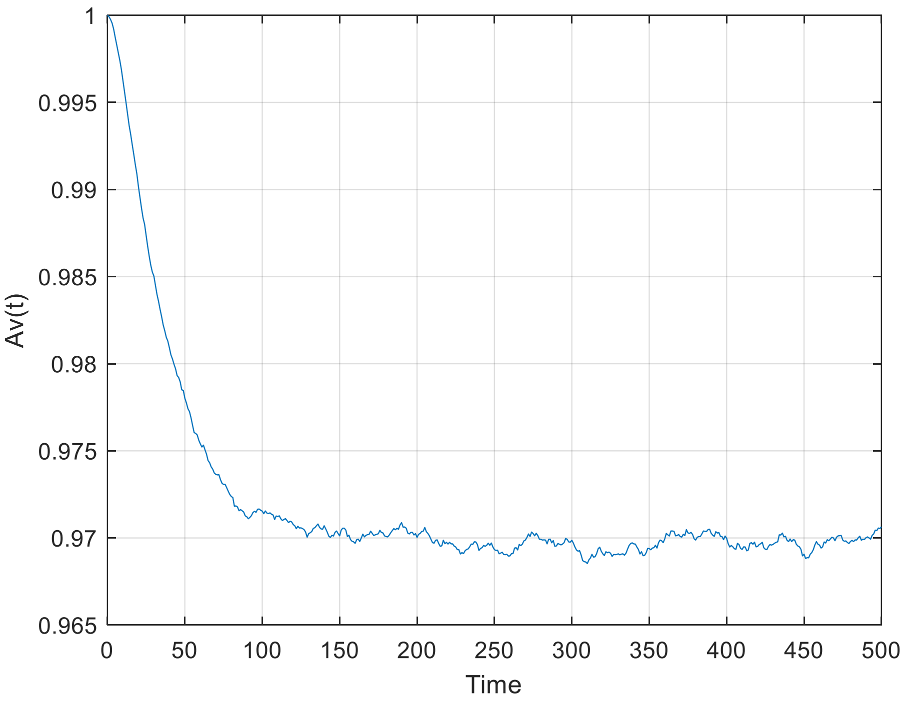
```
% Estimate the reliability by the MC samples
Mean_rel=1-mean(unrel); %time dependent reliability
sig_rel=sqrt(var(1-unrel)/N); %std of the reliability estimator
Av_MC=1-mean(counter_q); %instantaneous availability
sig_av=sqrt(var(1-counter_q)/N); %std of the availability estimator
rel_an= @(tt) exp(-lambda_A*tt)+exp(-(lambda_B+lambda_C)*tt)-exp(-(lambda_A+lambda_B+lambda_C)*tt);

figure(1)
plot(Time_axis,Mean_rel,Time_axis,rel_an(Time_axis)), grid on, xlabel ('Time'), ylabel('R(t)')
legend('MC estimation','Analitical reliability')
figure(2)
plot(Time_axis,sig_rel), grid on, xlabel ('Time'), ylabel('\sigma_R(t)')
figure(3)
plot(Time_axis,Av_MC), grid on
xlabel('Time'); ylabel('Av(t)');
axis([0,Tm,0.97,1.0005])
figure(4)
plot(Time_axis,sig_av), grid on
xlabel('Time'); ylabel('\sigma_A(t)');
```

# Exercise 3 –results

## Reliability

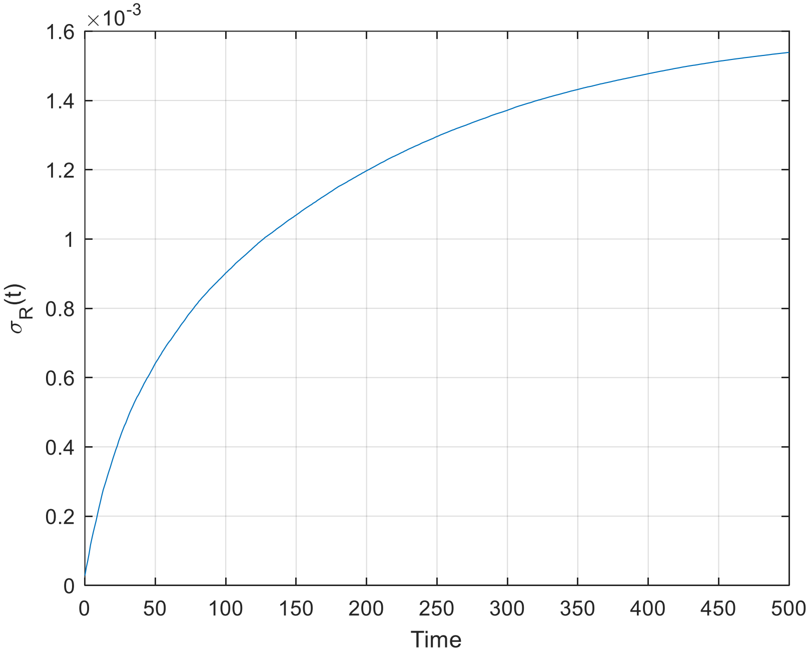



## Availability



# Exercise 3 –results

## Reliability




## Availability