Scilab Manual for Random Signal Analysis by Mrs Ashwini S. Sawant Electronics Engineering VESIT[1]

Solutions provided by
Mrs Ashwini S. Sawant,Assistant Professor
Electronics Engineering
Mumbai/V.E.S.I.T.

March 28, 2026

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

Simulation of Discrete Random Variable and Estimation of its PDF and CDF.

Scilab code Solution 1.1 1

```
//Generate a Cumulative Distibution function and Probability Density Function
//plot of discrete random variables where x=1,2,3 and their respective
//probabilities are given by 3/15,7/15,5/15

x=[1,2,3];.................................................//Random Variable

px=[3/15 7/15 5/15];..........................................//Probability of RV

y=cumsum(px);................................................//Cumulative sum of probabilities.
```

```
title("Simulation of Discrete RV and Estimation of
    CDF and PDF");//Title

subplot(121)

plot2d2(x,y,1,"111",rect=[0,0,4,1.25])
    ;........................//plot step function

xlabel("RANDOM VARIABLE X","fontsize",5);

ylabel("CDF F(x)","fontsize",5);

title("Cumulative Distribution Function","fontsize"
    ,5);

subplot(122)

plot2d3(x,px,rect=[0,0,4,1])
    ;..................................//plot stem
    function

xlabel("Random Variable X","fontsize",5);

ylabel("PDF f(x)","fontsize",5);

title("Probability Density Function","fontsize",5);
```

Experiment: 2

Relation between Distribution(CDF) and Density Function(PDF).

Scilab code Solution 2.1 2

```
//Relation between Cumulative Distribution Function and Probability Density Function.

//Generate a set of 100 uniform random variable in the range of -2 to 2.
//Plot Histogram for the same.

stacksize("max");.....................//Maximizing Size

x=grand(1,100,'unf',-2,2);...........//Generate random numbers

w=unique(x);.........................//Arrange in ascending order

p=100^(-1)*ones(1,100);..............//Calculate
```

```
probability

subplot(2,2,1.5);

histplot(10,w);......................//To plot Histogram

xlabel("Random Numbers");

ylabel("Amplitude");

title("Histogram","fontsize",4);

z=cumsum(p);........................//Take cumulative sum of probabilities

subplot(2,2,3);

plot2d2(w,z);.......................//Plot CDF

xlabel("Random Numbers");

ylabel("Amplitude");

title("CDF","fontsize",4);

subplot(2,2,4);

plot2d3(w,z,p);.....................//Plot PDF

title("PDF","fontsize",4);

xlabel("Random Numbers");

ylabel("Amplitude");
```

Experiment: 3

Study of Gaussian Distribution.

Scilab code Solution 3.1 3

```
//GAUSSIAN DISTRIBUTION

stacksize("max");

x=[-2:0.1:2];....................................//defining X-axis

s1=0.5^0.5;......................................//defining variance_1

s2=0.8^0.8;......................................//defining variance_2

gd1=(1/((2*%pi)^0.5*s1)*%e^-((x/s1)^2/2));......//Expression for Gaussian Distribution

plot2d1(x,gd1);

gd2=(1/((2*%pi)^0.5*s2)*%e^-((x/s2)^2/2));.....//Expression for Gaussian Distribution

```

```
plot2d1(x,gd2);

xlabel("X");

ylabel("f(x)");

title("Gaussian Distribution","fontsize",5);

e=gce();....................................//Changing Line style

e1=e.children(1);...........................//Call the Function

e1.thickness=1;.............................//Thickness of the Line

e1.line_style=3;............................//Line style selection

h1=legend(['f1=var=0.5','f2=var=0.8']);.....//Plot for 2 variances.
```

Experiment: 4

Study of Power Spectral Density.

Scilab code Solution 4.1 4

```
//POWER SPECTRAL DENSITY

sample_rate
    =1000;.........................................
    //No. of bit intervals.

t=0:1/sample_rate
    :10;.....................................//time
    axis for pulse shape

N=size(t,'*');

x=cos(2*%pi*47*t)+cos((2*%pi*219*t)+(%pi/4));

f=sample_rate*(0:(N/2))/N;

n=size(f,'*');

sm=pspect(N/2,N,'hn',x)
```

```
;................................// two sided cross-spectral estimate
                                                                      // between 2 discrete time signals

plot2d("gnl",f,sm(1:n));

xlabel('Frequency','fontsize',4);

title('POWER SPECTRAL DENSITY','fontsize',5);

ylabel('Power Content of the given signal','fontsize',4);
```

Experiment: 5

Study of Gaussian Random Process.

Scilab code Solution 5.1 5

```
//GAUSSIAN RANDOM PROCESS

stacksize('max');

u=[50:50];

sig=[100,30;30,140];..............................//
    defining covariance matrix....Cx(ti,tj)

a=1./((sqrt(det(sig)))*(sqrt(2*%pi)));

for(i=1:100)
    for(j=1:100)
        x=[i;j]
        xn=x-u;
        b(i,j)=exp(-(xn'*inv(sig)*xn*0.5));
    end
end

```

```
f=a*b;.........................................//PDF
    =fx1,x2,....xn=[1/(sqrt(2*pi)*sigma)^n)*(exp^(-(
    xi-m)^2/(2*(sigma^2)))]

surf(f);

xlabel('x random variable','fontsize',2);

ylabel('y random variable','fontsize',2);

zlabel('PDF','fontsize',2);

title('Gaussian Random Process','fontsize',5);
```

Experiment: 6

M/M/1 queueing system

Scilab code Solution 6.1 6

```
//M/M/1 Queueing System,Average Arrival rate lambd
    =20customers per hour and avg. service time=1/u=2
    minutes.

a=20;                                                   //
    Arrival rate(customers/hour)

b=30;                                                   //
    Departure Rate(1/b=avg.service time)

N=25;                                                   //
    Number of arrivals in simulation

x=-log(rand(1,N))/a;                        //Random
    interarrival time

x=cumsum(x);
```

```
//Random arrival time

y=-log(rand(1,N))/b;   //Service times for each customer

ser_start=x(1);   //First customer starts service
   //immediately upon arrival.

z(1)=ser_start+y(1);   //Departure time of first customer

for k=2:N   //kth customer
    ser_start=max([z(k-1),x(k)]);   //beginning of service time
    z(k)=ser_start+y(k);   //end of service time.
end

//Construct date to plot graph of queue size vs. time.

xaxis=[0, x(1)];
```

```
//Vector of points for the M/M/1

yaxis=[0, 0];  //Vector of queue sizes at points in preceding vector.

qs=1;  //Current queue size.

x=x(2:length(x));

while length(x)>0
    if x(1)<z(1)  //Next point is arrival.
        qs=qs+1;  //Increase queue size
        xaxis=[xaxis xaxis(length(xaxis)) x(1)];
        yaxis=[yaxis qs qs];
        x=x(2:length(x));
    else  //Next point is departure.
        qs=qs-1;  //Decrease queue size
        xaxis=[xaxis xaxis(length(xaxis)) z(1)];
        yaxis=[yaxis qs qs];
        z=z(2:length(z));
    end
end

plot(xaxis,yaxis)  //Plot realization of birth-death process.
```

```
xlabel('time(hours)',fontsize',4);

ylabel('queue size','fontsize',4)
```