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

Solutions provided by
Prof Mrs. Ashwini S. Sawant
Electronics Engineering
V.E.S.I.T,Chembur

April 19, 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 Scilab Solutions 3

1 Estimate the Joint PDF of a pair of random variables uniformly distributed over the unit circle. 5

2 To study the Convergence of Gaussian and Arcsine Random Variables. 8

3 To study the Mean Ergodic Random Process. 10

4 To study the Poisson Arrival Process. 12

5 Realization of Random Telegraph Signal. 14

List of Experiments

Solution 1.1 1 5
Solution 2.1 1 8
Solution 3.1 1 10
Solution 4.1 1 12
Solution 5.1 1 14

List of Figures

Experiment: 1

Estimate the Joint PDF of a pair of random variables uniformly distributed over the unit circle.

Scilab code Solution 1.1 1

```
//ESTIMETE THE JOINT PDF OF A PAIR OF RANDOM VARIABLES UNIFORMLY DISTRIBUTED
//OVER THE UNIT CIRCLE.
//Two random variables are generated independently x=rand(1) and y=rand(2).
//This would prodece3 a pair of random variables uniformly distributed over
//the square 0<x<1 and 0<y<1.

clc;
clear all;
N=1000;                              //Number of samples per iteration
bw=0.1;                              //Bin width for histogram
```

```
xb=[-1.4:bw:1.4];
yb=[-1.4:bw:1.4];                                   //Histogram bins
iterations=100;                                     //Number of ierations
M=length(xb);
Nsamples=zeros(M,M);

count=0;                                            //Initialize matrix for storing
                                                    //data
for  ii=1:iterations
    x=2*rand(1,N)-1;
    y=2*rand(1,N)-1;                                //Generate vatiables over square

   //Keep only those within the unit circle
    X=[];Y=[];
    for k=1:N
        if  (x(k)^2+y(k)^2)<1
            X=[X x(k)];
            Y=[Y y(k)];

        end
    end
    count=count+length(X);                          //Count random samples generated

    //Compute number of samples that fall within each bin.
    for m=1:length(xb)
        for n=1:length(yb)
    temp1=(abs(X-xb(m))<bw/2)
    temp2=(abs(Y-yb(n))<bw/2)
    Nsamples(m,n)=Nsamples(m,n)+sum(temp1.*temp2)

    end
end
```

```
end
PDFest=Nsamples/(count*bw^2);                    //
    Convert to probability
                                                 //
                                                    densities
mesh(xb,yb,PDFest)                               //Plot
    estimate of joint PDF
xlabel('x');ylabel('y');                         //Label
    plot axes
zlabel('Joint PDF');
```

Experiment: 2

To study the Convergence of Gaussian and Arcsine Random Variables.

Scilab code Solution 2.1 1

```
//THIS EXAMPLE SHOWS HOW THE SAMPLE MEAN AND SAMPLE
    VARIANCE CONVERGE TO THE
//TRUE MEAN FOR GAUSSIAN AND ARCSINE RANDOM
    VARIABLES.
//MEAN=3 AND THE VARIANCE OF EACH SAMPLE=1

clc();

N=100;

//Create Gaussian random variables

mu1=3;

sigma1=1;
```

```
X1=sigmal*rand(1,N)+mu1;
mu_hat1=cumsum(X1)./[1:N];                //sample means
//Create Arcsine random variables
mu2=3;
b=sqrt(2);
sigma2=b^2/2;
X2=b*cos(2*%pi*rand(1,N))+mu2;
mu_hat2=cumsum(X2)./[1:N];                //sample means
subplot(2,1,1);
plot([1:N],mu_hat1,'-',[1:N],mu1,'-')
xlabel('Samples,n','fontsize',2);
ylabel('Sample mean','fontsize',2);
title('Gaussian','fontsize',3)
mtlb_axis([0,N,0,2*mu1])
subplot(2,1,2);
plot([1:N],mu_hat2,'-',[1:N],mu2,'-')
xlabel('Samples,n','fontsize',2);
ylabel('Sample mean','fontsize',2);
title('Arcsine','fontsize',3)
mtlb_axis([0,N,0,2*mu2])
```

Experiment: 3

To study the Mean Ergodic Random Process.

Scilab code Solution 3.1 1

```
//ERGODIC RANDOM PROCESS.
//COMPARISON OF THE SAMPLE MEAN AND ENSEMBLE MEAN
   FOR THE SINUSOID WITH RANDOM
//FREQUENCY.
//The solid line represents sample mean and the
   dashed line is ensemble mean.


clc();

f=4;                                          //Maximum
   frequency

N=1000;                                       //Number
   of realizations

t=[-4.995:0.01:4.995];                        //Time
```

```
    axis

F=f*rand(N,1);                              //Uniform frequencies

x=cos(2*%pi*F*t);                           //Each row is a ealization of process

z=sum(x,1)
sample_mean=z/N;                            //Compute sample mean

true_mean=(sin(2*%pi*f*t))./(2*%pi*f*t);    //Compute ensemble mean

plot(t,sample_mean,'–');                    //Plot results
plot(t,true_mean,'—');

xlabel('t (seconds)');

ylabel('mu(t)');
```

Experiment: 4

To study the Poisson Arrival Process.

Scilab code Solution 4.1 1

```
//HISTOGRAM OF THE PMF OF THE QUEUE LENGTH FOR THE
   TAXI STAND.
//POISSON ARRIVAL PROCESS WITH AN AVERAGE ARRIVAL
   RATE OF 0.85 ARRIVALS PER
//TIME UNIT.


clc();

N=10000;
   //Length of simulation

a=0.85;
   //Arrival rate

k=[0:10];

Poisson=zeros(size(k));
   //Calculate Poisson PMF
```

```
for m=k
    Poisson(m+1)=a.^m*exp(-a)./factorial(m);
end
queue(1)=0; //Initial queue size
for n=1:N
    x=rand(1);
    arrivals=sum(x>cumsum(Poisson)); //Poisson RV
    departures=queue(n)>0;
    queue(n+1)=queue(n)+arrivals-departures; //Current queue length
end
mean_queue_length=sum(queue)/length(queue) //Compute average queue length
bins=[0:25]
y=histplot(bins,queue);
PMF=y/N; //Estimate PMF
bar(bins,PMF) //Plot results
plot([min(bins)-1 max(bins)+1 0 1.1*max(PMF)])
ylabel('Probability Density Function','fontsize',2);
xlabel('Queue Length','fontsize',2);
```

Experiment: 5

Realization of Random Telegraph Signal.

Scilab code Solution 5.1 1

```
//REALIZATION OF RANDOM TELEGRAPH SIGNAL.
//LET T1,T2,T3,...BE A SEQUENCE OF IID RANDOM VARIABLES,EACH WITH AN EXPONENTIAL
//DISTRIBUTION.AT ANY TIME INSTANTS X(t) TAKES 2 POSSIBLE STATES,X(t)=0 OR X(t)=1

clc();

N=10;                        //number of swithces in realization

Fs=100;                      //sample rate (samples per second)

lambda=1/2;                  //switching rate(switches per second)

X=[];

```

```
S=rand(1,N);            //uniform random variables

T=(-log(S))/lambda;       // transform to exponential RVs

V=cumsum(T);            //switching times

state=0;

Nsold=1;

for k=1:N
    Nsnew=ceil(V(k)*Fs);            //new switchim=ng time
    Ns=Nsnew - Nsold;               //Number of samples in current switching interval
    X=[X state*ones(1,Ns)];
    state=1-state;                  //switch state
    Nsold=Nsnew;
end

t=[1:length(X)]/Fs;                 //time axis

plot2d(t,X);                          //plot results

xlabel('time, t');

ylabel('X(t)');
```