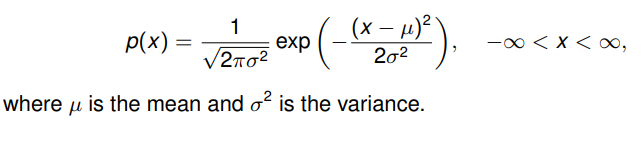
**INNOVATIVE EXPERIMENT – 1**

**Aim :** To generate PDF and CDF of the Gaussian random variable using MATLAB.

**Software Used :** MATLAB Online

**Theory :** Gaussian random variable is a continuous random variable with probability density function (PDF) is of the form :



In general, the Gaussian PDF is centered about the point x = µ and has a width that is proportional to σ.

Gaussian random variables are commonly used in such a wide variety of applications, including the thermal noise encountered in electronic circuits and is hence deemed of the most importance among random variables.

It is standard practice to introduce a shorthand notation to describe a Gaussian random variable, X ∼ N (µ, σ2). This is read “X is distributed normally (or Gaussian) with mean, µ and variance, σ2.”

In probability theory, a normal (or Gaussian or Gauss or Laplace–Gauss) distribution is a type of continuous probability distribution for a real-valued random variable.

A random variable with a Gaussian distribution is said to be normally distributed, and is called a normal deviate.

The first goal to be addressed in the study of Gaussian random variables is to find its CDF. The CDF is required whenever we want to find the probability that a Gaussian random variable lies above or below some threshold or in some interval.

**Source Code :**

clc;

clearvars;

close all;

%input configurations

N=1e6;

x=-5:0.01:5;

%plot configurations

FONTSIZE\_LABEL=14;

FONTSIZE\_LEGEND=12;

%main program

f=@(x) 1/sqrt(2\*pi) \* exp(-x.^2/2);

F=@(x) 1-qfunc(x);

X=randn(1,N);

%pdf plots

figure(1);

subplot(1,2,1);

%plot histogram

histogram(X,200,'Normalization','pdf');

hold on;

%plot theoretical pdf

plot(x,f(x),'r-','LineWidth',3);

legend('Simulated','Theoretical','Fontsize',FONTSIZE\_LEGEND, ...

'Location','northeast');

xlabel('x','Fontsize',FONTSIZE\_LABEL);

ylabel('f\_X(x)','Fontsize',FONTSIZE\_LABEL);

title('PDF of standard gaussian random variable.');

grid on; axis tight; grid minor; hold off;

%cdf plots

figure(1); subplot(1,2,2);

%plot histogram

histogram(X,200,'Normalization','cdf');

hold on;

%plot theoretical pdf

plot(x,F(x),'r-','LineWidth',3);

legend('Simulated','Theoretical','Fontsize',FONTSIZE\_LEGEND, ...

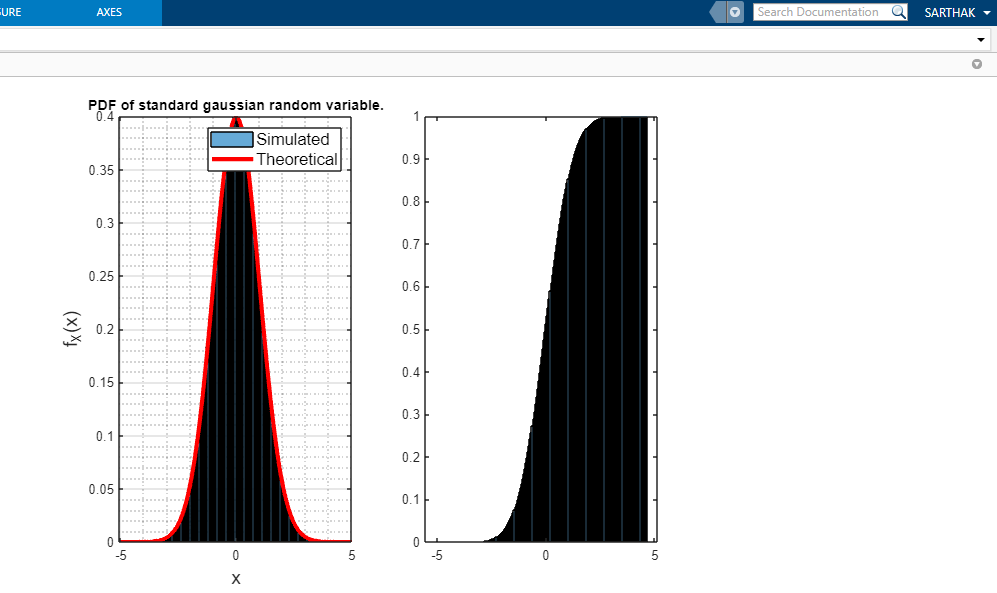
'Location','northwest');

xlabel('x','Fontsize',FONTSIZE\_LABEL);

ylabel('F\_X(x)','Fontsize',FONTSIZE\_LABEL);

title('CDF of standard gaussian random variable.');

**Output :**

****

**Conclusion :** Successfully generated Gaussian random variable using MATLAB.

**INNOVATIVE EXPERIMENT – 2**

**Aim :** Generation of phase demodulation using MATLAB.

**Software Used :** MATLAB Online

**Theory :**

In phase modulation, the information is encoded as variations in the phase of the carrier signal. In its generic form, a phase modulated signal expressed as an information-bearing sinusoidal signal modulating another sinusoidal carrier signal is expressed as

x(t) = Acos[2πfct + ß + asin(2πfmt + θ)]

where, m(t) = asin(2πfmt + θ) represents the information-bearing modulating signal, with the following parameters

## Demodulating a phase modulated signal :

The phase modulated signal shown in equation (1), can be simply expressed as

x(t) = Acos[Ø(t)]

Here, Ø(t) is the instantaneous phase that varies according to the information signal m(t).

A phase modulated signal of form x(t) can be demodulated by forming an analytic signal by applying Hilbert transform and then extracting the instantaneous phase.

We note that the instantaneous phase :

Ø(t) = 2πfct + ß + asin(2πfmt + θ)

is linear in time, that is proportional to 2πfct. This linear offset needs to be subtracted from the instantaneous phase to obtain the information-bearing modulated signal. If the carrier frequency is known at the receiver, this can be done easily. If not, the carrier frequency term 2πfct needs to be estimated using a linear fit of the unwrapped instantaneous phase.

**Source Code :**

clc;

close all;

clear all;

t= [0:0.001:1];

fm = 5;

m= cos(2\*pi\*fm\*t);

wm=2\*pi\*fm;

subplot(4,1,1);

plot(t,m);

title('Message Signal');

fc= 100;

c= sin(2\*pi\*fc\*t);

wc=2\*pi\*fc;

subplot(4,1,2);

plot(t,c);

title('Carrier Signal');

kp=input('Enter the value of phase sensitivity');

s=cos(wc\*t + m.\*kp);

subplot(4,1,3);

plot(t,s);

title('Phase Modulated signal');

z=hilbert(s);

inst\_phase = unwrap(angle(z));

p = polyfit(t,inst\_phase,1);

offsetTerm = polyval(p,t);

d= inst\_phase - offsetTerm;

subplot(4,1,4);

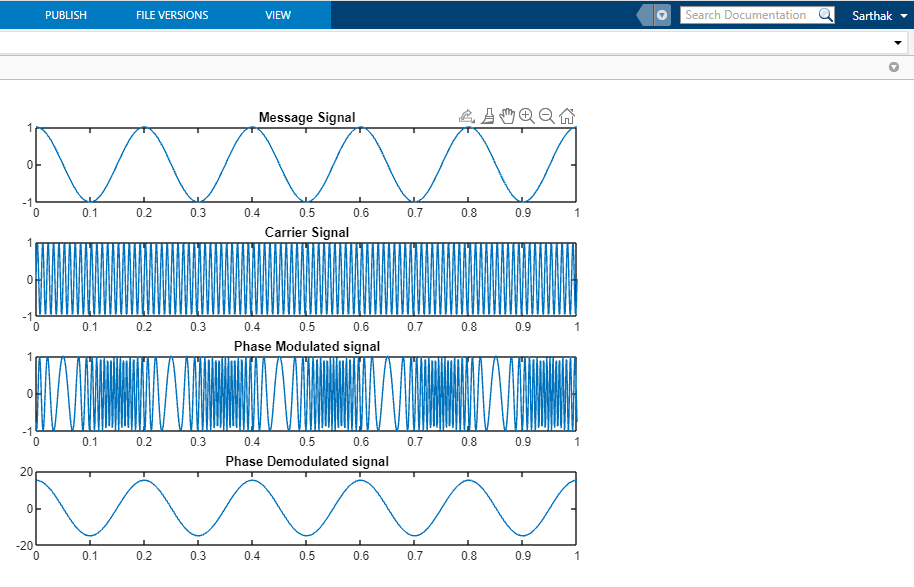
plot(t,d);

title('Phase Demodulated signal');

**Input :**

Phase Sensitivity = 15

**Output :**

****

**Conclusion :** Successfully generated phase demodulation using MATLAB