Matlab Program for Digital Communication

#1 ASK

% MATLAB Code FOR ASK (Amplitude Shift Keying) :

% Enter the freq of Sine Wave carrier:50

% Enter the freq of Periodic Binary pulse (Message):5

% Enter the amplitude (For Carrier & Binary Pulse Message):5

clc %for clearing the command window

close all %for closing all the window except command window

clear all %for deleting all the variables from the memory

fc=input('Enter the freq of Sine Wave carrier:');

fp=input('Enter the freq of Periodic Binary pulse (Message):');

amp=input('Enter the amplitude (For Carrier & Binary Pulse Message):');

t=0:0.001:1; % For setting the sampling interval

c=amp.\*sin(2\*pi\*fc\*t);% For Generating Carrier Sine wave

subplot(3,1,1) %For Plotting The Carrier wave

plot(t,c)

xlabel('Time')

ylabel('Amplitude')

title('Carrier Wave')

m=amp/2.\*square(2\*pi\*fp\*t)+(amp/2);%For Generating Square wave message

subplot(3,1,2) %For Plotting The Square Binary Pulse (Message)

plot(t,m)

xlabel('Time')

ylabel('Amplitude')

title('Binary Message Pulses')

w=c.\*m; % The Shift Keyed Wave

subplot(3,1,3) %For Plotting The Amplitude Shift Keyed Wave

plot(t,w)

xlabel('Time')

ylabel('Amplitude')

title('Amplitide Shift Keyed Signal')

#2 FSK or BFSK

% MATLAB Code FOR FSK (Frequency Shift Keying) BFSK in this case:

% Enter the freq of 1st Sine Wave carrier:50

% Enter the freq of 2nd Sine Wave carrier:10

% Enter the freq of Periodic Binary pulse (Message):5

% Enter the amplitude (For Both Carrier & Binary Pulse Message):5

clc %for clearing the command window

close all %for closing all the window except command window

clear all %for deleting all the variables from the memory

fc1=input('Enter the freq of 1st Sine Wave carrier:');

fc2=input('Enter the freq of 2nd Sine Wave carrier:');

fp=input('Enter the freq of Periodic Binary pulse (Message):');

amp=input('Enter the amplitude (For Both Carrier & Binary Pulse Message):');

amp=amp/2;

t=0:0.001:1; % For setting the sampling interval

c1=amp.\*sin(2\*pi\*fc1\*t);% For Generating 1st Carrier Sine wave

c2=amp.\*sin(2\*pi\*fc2\*t);% For Generating 2nd Carrier Sine wave

subplot(4,1,1); %For Plotting The Carrier wave

plot(t,c1)

xlabel('Time')

ylabel('Amplitude')

title('Carrier 1 Wave')

subplot(4,1,2) %For Plotting The Carrier wave

plot(t,c2)

xlabel('Time')

ylabel('Amplitude')

title('Carrier 2 Wave')

m=amp.\*square(2\*pi\*fp\*t)+amp;%For Generating Square wave message

subplot(4,1,3) %For Plotting The Square Binary Pulse (Message)

plot(t,m)

xlabel('Time')

ylabel('Amplitude')

title('Binary Message Pulses')

for i=0:1000 %here we are generating the modulated wave

if m(i+1)==0

mm(i+1)=c2(i+1);

else

mm(i+1)=c1(i+1);

end

end

subplot(4,1,4) %For Plotting The Modulated wave

plot(t,mm)

xlabel('Time')

ylabel('Amplitude')

title('Modulated Wave')

#3 PAM

% MATLAB Code For PAM (Pulse-amplitude modulation)

% Enter the amplitude = 4

% Enter the frequency = 3

clc;

close all;

clear all;

a = input('Enter the amplitude = ');

f = input('Enter the frequency = ');

t = 0:0.02:2; % for a total of 16 samples

x1 = 1; %generation of an impulse signal

x2 = a\*sin(2\*pi\*f\*t); %generation of sine wave

y = x1.\*x2; %modulation step

subplot(3,1,1); %for impulse signal plot

stem(x1);

title('Impulse Signal');

xlabel('Time');

ylabel('Amplitude');

grid on

subplot(3,1,2) %for sine wave plot

plot(t,x2);

title('Sine Wave');

xlabel('Time ');

ylabel('Amplitude ');

grid on

subplot(3,1,3) %for PAM wave plot

stem(t,y);

title('PAM Wave');

xlabel('Time');

ylabel('Amplitude');

grid on

#4 PCM

% MATLAB Code For PCM (Pulse Code Modulation)

% Enter Bit Depth Of PCM Coding 2 or 4 or 8 or 16

clc

close all

clear all

t = 0:0.0001:20; %sampling at niquist rate

c=input('Enter Bit Depth Of PCM Coding:');

part = -1:0.1:1;%A quantization partition defines several contiguous, nonoverlapping ranges

%of values within the set of real numbers.

codebook = -1:0.1:1.1;%A codebook tells the quantizer which common value to assign to inputs that

%fall into each range of the partition.

msg = sin(t); %or cos(t)

[~,quants] = quantiz(msg,part,codebook);%returns a vector that tells which interval each input is in

subplot(3,1,1);

plot(t,msg);

title('Message Signal');

grid on

subplot(3,1,2);

plot(t,quants);

title('Quantized Signal');

grid on

y = uencode(quants,c);

ybin=dec2bin(y,c); %converting it to final binary form to make it transmit ready

subplot(3,1,3);

plot(t,y);

title('PCM PLOT');

grid on

#5 PPM

% MATLAB Code For PPM (Pulse Position Modulation, PPM)

clc;

clear all;

close all;

fc=1000;

fs=10000;

f=200;

t=0:2/fs:((2/f)-(1/fs));

x=0.4\*cos(2\*pi\*f\*t)+0.5;

y=modulate(x,fc,fs,'ppm');

subplot(3,1,1);

plot(x);

title('modulating signal');

subplot(3,1,2);

plot(y);

title('modulated signal');

x\_recov=demod(y,fc,fs,'ppm');

subplot(3,1,3);

plot(x\_recov);

title('original signal')

#6 PPM

% MATLAB Code FOR PSK (Phase Shift Keying) :

% Enter frequency of Carrier Sine wave: 50

% Enter Message frequency : 10

% Enter Carrier & Message Amplitude(Assuming Both Equal):5

clc %for clearing the command window

close all %for closing all the window except command window

clear all %for deleting all the variables from the memory

t=0:.001:1; % For setting the sampling interval

fc=input('Enter frequency of Carrier Sine wave: ');

fm=input('Enter Message frequency : ');

amp=input('Enter Carrier & Message Amplitude(Assuming Both Equal):');

c=amp.\*sin(2\*pi\*fc\*t);% Generating Carrier Sine

subplot(3,1,1) %For Plotting The Carrier wave

plot(t,c)

xlabel('Time')

ylabel('Amplitude')

title('Carrier')

m=square(2\*pi\*fm\*t);% For Plotting Message signal

subplot(3,1,2)

plot(t,m)

xlabel('time')

ylabel('ampmplitude')

title('Message Signal')% Sine wave multiplied with square wave in order to generate PSK

x=c.\*m;

subplot(3,1,3) % For Plotting PSK (Phase Shift Keyed) signal

plot(t,x)

xlabel('t')

ylabel('y')

title('PSK')

#7 PWM

% MATLAB Code for PWM (Pulse Width Modulation):10 1 5 or 10 5 5

% Comparator Sawtooth frequency:10

% Message frequency(Assuming it to be a sine wave):1 or 5

% Enter Amplitude of Message:5

clc

clear all

close all

fs=input('Comparator Sawtooth frequency:');

fm=input('Message frequency(Assuming it to be a sine wave):');

a=input('Enter Amplitude of Message:');

t=0:0.0001:1; %sampling rate of 10kHz

stooth=1.01\*a.\*sawtooth(2\*pi\*fs\*t); %generating a sawtooth wave

%to make the two non zero lobes of pwm not to overlap the amplitude of

%sawtooth wave must be atleast more than a bit to the message amplitude

subplot(3,1,1);

plot(t,stooth); % plotting the sawtooth wave

title('Comparator Wave');

grid on

msg=a.\*sin(2\*pi\*fm\*t); %generating message wave

subplot(3,1,2);

plot(t,msg); %plotting the sine message wave

title('Message Signal');

grid on

for i=1:length(stooth)

if (msg(i)>=stooth(i))

pwm(i)=1; %is message signal amplitude at i th sample is greater than

%sawtooth wave amplitude at i th sample

else

pwm(i)=0;

end

end

subplot(3,1,3);

plot(t,pwm,'r');

title('PWM');

axis([0 1 0 1.1]); %to keep the pwm visible during plotting.

grid on