1. Frequency Shift Keying .

%Frequency Shift Keying matlab code

clear all;

close all;

clc;

f1=1;

f2=3;

fs=100;

t=0: 1/fs: 1;

bit=[1 1 0 0 1 0 1 1];

time=[];

FSK\_Signal=[];

for i=1:1: length(bit)

FSK\_Signal=[FSK\_Signal (bit(i)==0)\*sin(2\*pi\*f1\*t)...

+ (bit(i)==1)\*sin(2\*pi\*f2\*t)];

time=[time t];

t=t +1;

end

plot(time,FSK\_Signal,'green','LineWidth',2.5);

axis([0 time(end) -1.5 1.5]);

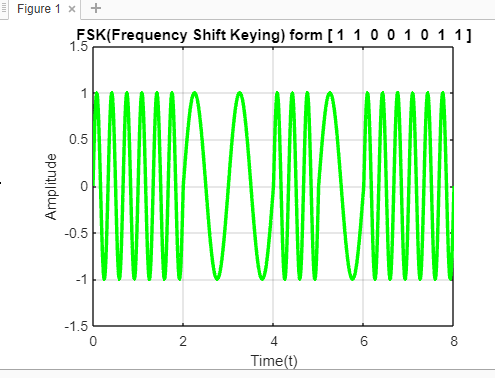
xlabel('Time(t)');

title(['FSK(Frequency Shift Keying) form [ ' num2str(bit) ' ]']);

ylabel('Amplitude');

grid on;

output:



1. Phase Shift Keying .

code:

clear all;

close all;

clc;

f1=2;

fs=100;

t=0: 1/fs: 1;

bit=[1 1 0 0 1 0 1 1];

time=[];

psk\_Signal=[];

for i=1:1: length(bit);

psk\_Signal=[psk\_Signal (bit(i)==0)\*-sin(2\*pi\*f1\*t)...

+ (bit(i)==1)\*sin(2\*pi\*f1\*t)];

time=[time t];

t=t +1;

end

plot(time,psk\_Signal,'magenta','LineWidth',2.5);

axis([0 time(end) -1.5 1.5]);

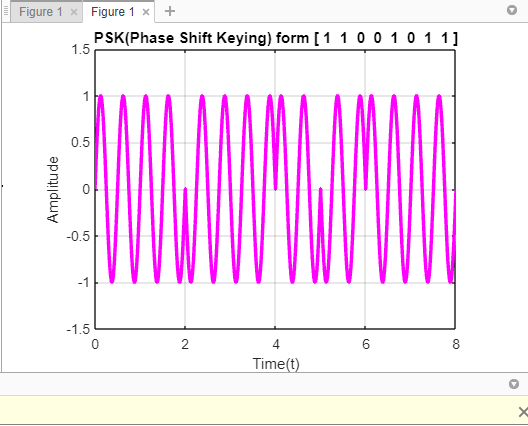
xlabel('Time(t)');

title(['PSK(Phase Shift Keying) form [ ' num2str(bit) ' ]']);

ylabel('Amplitude');

grid on;

output:



3. Amplitude Shift Keying.

code:

clear all;

close all;

clc;

f1=2;

fs=100;

t=0: 1/fs: 1;

bit=[1 1 0 0 1 0 1 1];

time=[];

ask\_Signal=[];

for i=1:1: length(bit);

ask\_Signal=[ask\_Signal (bit(i)==0)\*...

zeros(1,length(t))+ (bit(i)==1)\*sin(2\*pi\*f1\*t)];

time=[time t];

t=t +1;

end

plot(time,ask\_Signal,'cyan','LineWidth',2.5);

axis([0 time(end) -1.5 1.5]);

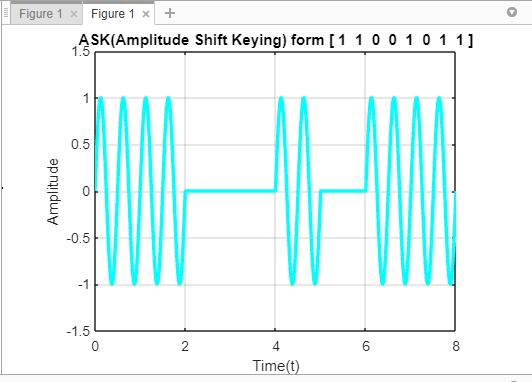
xlabel('Time(t)');

title(['ASK(Amplitude Shift Keying) form [ ' num2str(bit) ' ]']);

ylabel('Amplitude');

grid on;

output:



4. Unipolar Not return to zero(NRZ)

code:

%UnipolarNRZ matab code

clear all;

close all;

clc;

fs=100;

t=0:1/fs:1;

bit=[1 1 0 1 0 0 1];

time = [];

Digital\_Signal = [];

for i = 1: 1 : length(bit)

Digital\_Signal = [Digital\_Signal (bit(i)==0)\*zeros(1,length(t))+(bit(i)==1)\*ones(1,length(t))];

time = [time t];

t = t+1;

end

subplot(1,1,1);

plot(time,Digital\_Signal,'magenta','linewidth',2.5);

title(['Unipolar NRZ [ ' num2str(bit) ' ]']);

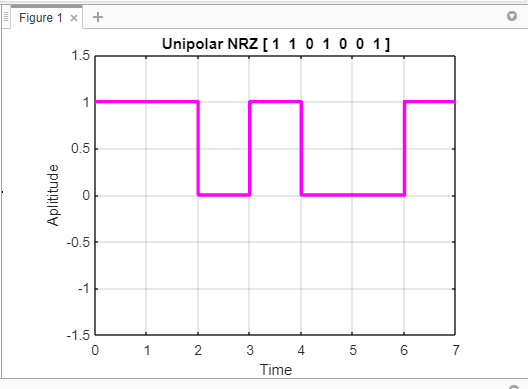
xlabel('Time');

ylabel('Aplititude');

axis([0 time(end) -1.5 1.5]);

grid on;

output:



5.Manchester line coding .

code:

clear all;

close all;

clc;

fs=100;

t=0: 1/fs:1;

bit=[1 1 0 1 0 0 1];

nbit=[];

Time=[];

Dig\_Sig=[];

k=1;

for i=1: 1 : length(bit)

if(bit(i)==1)

nbit(k)=1;

nbit(k+1)=-1;

end

if(bit(i)==0)

nbit(k)=-1;

nbit(k+1)=1;

end

k=k+2;

end

for i=1: 1 : length(nbit)

Dig\_Sig=[Dig\_Sig (nbit(i)==1)\*...

ones(1,length(t)) + (nbit(i)==-1)\*-ones(1,length(t))];

Time=[Time t]; t=t+1;

end

subplot(1,1,1);

plot(Time,Dig\_Sig, 'black','LineWidth',2.5);

axis([ 0 Time(end) -1.5 1.5]);

grid on;

xlabel('Time');

ylabel('Amplitude');

title(['Manchester [ ' num2str(bit) ' ]']);

code:

