01. AM

```
clc;
clear all;
close all;
Vm = 10;
Vc = 5;
fm = 4;
fc = 40;
phi_m = 0;
phi_c = 0;
Ma = 2;
t = 0:0.0001:1;
% Modulating Signal
modulatingSignal = Vm*sin(2*pi*fm*t + phi_m)
subplot(3,1,1);
plot(t, modulatingSignal);
axis([0 1 -Vm-1.5 Vm+1.5]);
xlabel('Time');
ylabel('Amplitude');
title('Modulating Signal');
 % Carrier Signal
carrierSignal = Vc*cos(2*pi*fc*t + phi_c);
subplot(3,1,2);
plot(t,carrierSignal);
axis([0 1 -Vc-1 Vc+1]);
xlabel('Time');
ylabel('Amplitude');
title('Carrier Signal');
```

```
% Modulating Signal
modulatedSignal = Vc.*cos(2*pi.*fc.*t).*(1 + Ma.*cos(2*pi.*fm.*t))
subplot(3,1,3);
plot(t, modulatedSignal);
xlabel('Time');
ylabel('Amplitude');
title('Modulated Signal');
```

02.FM

```
clc;
clear all;
close all;
Vm = 10;
Vc = 5;
fm = 4;
fc = 40;
phi_m = 0;
phi_c = 0;
Mf = 5;
t = 0:0.0001:1;
modulatingSignal = Vm*sin(2*pi*fm*t + phi_m);
subplot(3,1,1);
plot(t, modulatingSignal);
axis([0 1 -Vm-1.5 Vm+1.5]);
carrierSignal = Vc*cos(2*pi*fc*t + phi_c);
subplot(3,1,2);
plot(t, carrierSignal);
```

```
axis([0 1 -Vc-1 Vc+1]);
modulatedSignal = Vc.*sin(2*pi*fc*t + Mf.*sin(2*pi*fm*t));
subplot(3,1,3);
plot(t,modulatedSignal);
axis([0 1 -Vc-1.0 Vc+1.0]);
```

03.PM

```
clc;
clear all;
close all;
Vm = 10;
Vc = 5;
fm = 4;
fc = 40;
phi_m = 0;
phi_c = 0;
Mp = 10;
t = 0:0.0001:1;
modulatingSignal = Vm*sin(2*pi*fm*t + phi_m);
subplot(3,1,1);
plot(t, modulatingSignal);
axis([0 1 -Vm-1.5 Vm+1.5]);
carrierSignal = Vc*cos(2*pi*fc*t + phi_c);
subplot(3,1,2);
plot(t, carrierSignal,'black');
axis([0 1 -Vc-1 Vc+1]);
modulatedSignal = Vc.*sin(2*pi*fc*t + Mp.*sin(2*pi*fm*t));
subplot(3,1,3);
```

```
plot(t,modulatedSignal, 'm');
axis([0 1 -Vc-1.0 Vc+1.0]);
```

04. BASK & OOK

```
clc;
clear all;
close all;
bitstream = [0 1 0 0 1 1 0 1];
n = length(bitstream);
sampleperbit = 100;
highlogicvoltlevel = +5;
lowlogicvoltlevel = -5;
t = 0:1/sampleperbit:n;
for i = 1:1:n
    if bitstream(i) == 1
        digitalsignalvoltage(i) = highlogicvoltlevel;
    else
        digitalsignalvoltage(i) = lowlogicvoltlevel;
    end
end
totalsample = 1:1:(n + 1) * sampleperbit;
for i = 1:1:n
    for j = i:1/sampleperbit:i+1
        digitalsignal(totalsample(i*sampleperbit:(i+1)*sampleperbit)) = digitalsignalvoltage(i);
    end
end
digitalsignal = digitalsignal(sampleperbit:end);
subplot(3,1,1);
```

```
plot(t, digitalsignal,'green', 'linewidth', 2);
axis([-0.05 n+0.05 lowlogicvoltlevel-1 highlogicvoltlevel+1]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Digital Input Signal');
grid on;
Vc = 5;
fc = 5;
thetac = 0;
carriersignal = Vc .* sin(2 * pi * fc * t + thetac);
% Plot the carrier signal
subplot(3,1,2);
plot(t, carriersignal);
axis([-0.05 n+0.05 lowlogicvoltlevel-1 highlogicvoltlevel+1]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Carrier Signal');
grid on;
amplitudeDeviation = 2.0;
amplitudecomponentforsymbolone = Vc + amplitudeDeviation;
amplitudecomponentforsymbolzero = Vc - amplitudeDeviation;
% for OOK: amplitudecomponentforsymbolzero = 0;
phasecomponent = 0;
for i = 1:1:(n * sampleperbit + 1)
    % Check the digital signal and modulate accordingly
    if digitalsignal(i) == highlogicvoltlevel
        amplitudecomponent(i) = amplitudecomponentforsymbolone;
    else
```

```
        amplitudecomponent(i) = amplitudecomponentforsymbolzero;
    end
end
modulatedsignal = amplitudecomponent .* sin(2 * pi * fc * t + phasecomponent);
subplot(3,1,3);
plot(t, modulatedsignal);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Modulated Signal');
grid on;
```

05.BFSK

```
clc;
clear all;
close all;
bitstream = [0 1 0 0 1 1 0 1];
n = length(bitstream);
sampleperbit = 100;
highlogicvoltlevel = +5;
lowlogicvoltlevel = -5;
t = 0:1/sampleperbit:n;
for i = 1:1:n
    if bitstream(i) == 1
        digitalsignalvoltage(i) = highlogicvoltlevel;
    else
        digitalsignalvoltage(i) = lowlogicvoltlevel;
    end
end
```

```
 totalsample = 1:1:(n + 1) * sampleperbit;
for i = 1:1:n
   for j = i:1/sampleperbit:i+1
       digitalsignal(totalsample(i*sampleperbit:(i+1)*sampleperbit)) = digitalsignalvoltage(i);
   end
end
digitalsignal = digitalsignal(sampleperbit:end);
subplot(3,1,1);
plot(t, digitalsignal,'green', 'linewidth', 2);
axis([-0.05 n+0.05 lowlogicvoltlevel-2 highlogicvoltlevel+2]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Digital Input Signal');
grid on;
Vc = 5;
fc = 4;
thetac = 0;
carriersignal = Vc .* sin(2 * pi * fc * t + thetac);
subplot(3,1,2);
plot(t, carriersignal);
axis([-0.05 n+0.05 lowlogicvoltlevel-2 highlogicvoltlevel+2]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Carrier Signal');
grid on;
frequencySeparation = 2.0;
phasecomponentforsymbolone = 2.*pi.*(frequencySeparation/2);
phasecomponentforsymbolzero = - 2.*pi.*(frequencySeparation/2);
```

```
amplitudecomponent = 1;

for i = 1:1:(n * sampleperbit + 1)
    % Check the digital signal and modulate accordingly
    if digitalsignal(i) == highlogicvoltlevel
        phasecomponent(i) = phasecomponentforsymbolone;
    else
        phasecomponent(i) = phasecomponentforsymbolzero;
    end
end
modulatedsignal = amplitudecomponent*Vc .*sin((2*pi*fc + phasecomponent).*t);
subplot(3,1,3);
plot(t, modulatedsignal);
axis([-0.05 n+0.05 -Vc-1.5 Vc+1.5]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Modulated Signal');
grid on;
```

06. BPSK

```
clc;
clear all;
close all;
bitstream = [0 1 0 0 1 1 0 1];
n = length(bitstream);
sampleperbit = 100;
highlogicvoltlevel = +5;
lowlogicvoltlevel = -5;
```

```
t = 0:1/sampleperbit:n;

for i = 1:1:n
    if bitstream(i) == 1
        digitalsignalvoltage(i) = highlogicvoltlevel;
    else
        digitalsignalvoltage(i) = lowlogicvoltlevel;
    end
end
 totalsample = 1:1:(n + 1) * sampleperbit;
for i = 1:1:n
    for j = i:1/sampleperbit:i+1
        digitalsignal(totalsample(i*sampleperbit:(i+1)*sampleperbit)) = digitalsignalvoltage(i);
    end
end
digitalsignal = digitalsignal(sampleperbit:end);

subplot(3,1,1);
plot(t, digitalsignal,'green', 'linewidth', 2);
axis([-0.09 n+0.09 lowlogicvoltlevel-2 highlogicvoltlevel+2]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Digital Input Signal');
grid on;
Vc = 5;
fc = 4;
thetac = 0;
carriersignal = Vc .* sin(2 * pi * fc * t + thetac);
```

```
subplot(3,1,2);
plot(t, carriersignal);
axis([-0.05 n+0.05 lowlogicvoltlevel-2 highlogicvoltlevel+2]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Carrier Signal');
grid on;
frequencySeparation = 2.0;
phasecomponentforsymbolone = 0;
phasecomponentforsymbolzero = pi;
amplitudecomponent = 1;

for i = 1:1:(n * sampleperbit + 1)
   if digitalsignal(i) == highlogicvoltlevel
      phasecomponent(i) = phasecomponentforsymbolone;
   else
      phasecomponent(i) = phasecomponentforsymbolzero;
   end
end
modulatedsignal = amplitudecomponent*Vc .*sin((2*pi*fc*t + phasecomponent));
subplot(3,1,3);
plot(t, modulatedsignal, 'color', 'm');
axis([-0.09 n+0.09 -Vc-1.5 Vc+1.5]);
xlabel('Time(sec)');
ylabel('Amplitude(volt)');
title('Modulated Signal');
grid on;
```

07. PAM

```
clc;
clear all;
close all;
Am = 1;
fm = 0.9;
phim = 0;
minTime = 0;
maxTime = 2;
samplingRate = 1000;
t = minTime : 1/samplingRate : maxTime;
modulatingsignal = Am * cos(2 * pi * fm * t + phim);
subplot(2,2,1);
plot(t, modulatingsignal);
axis([minTime maxTime -Am-0.4 Am+0.4])
xlabel('Time (s)');
ylabel('Amplitude');
title('Modulating Signal');
grid on;
fc = 10;
Ac = 1;
carrierpulses = Ac * square(2 * pi * fc * t) + Ac;
subplot(2,2,3);
plot(t, carrierpulses);
axis([minTime maxTime Ac-1.4 2*Ac+0.4]);
xlabel('Time (s)');
ylabel('Amplitude');
title('Carrier Pulses');
```

```
grid on;
doublesidebandmodulatingsignal = modulatingsignal .* carrierpulses;
subplot(2,2,2);
plot(t, doublesidebandmodulatingsignal);
axis([minTime maxTime -2*Ac-0.4 2*Ac+0.4]);
xlabel('Time (s)');
ylabel('Amplitude');
title('Double-Sideband Modulated Signal');
grid on;
singlesidebandmodulatingsignal  = [];
for i = 1:1:length(doublesidebandmodulatingsignal)
    if doublesidebandmodulatingsignal(i) == 0
        singlesidebandmodulatingsignal = [singlesidebandmodulatingsignal, doublesidebandmodulatingsignal(i)];
    else
        singlesidebandmodulatingsignal = [singlesidebandmodulatingsignal, doublesidebandmodulatingsignal(i) + 3 * Ac * Am];
    end
end

subplot(2,2,4);
plot(t, singlesidebandmodulatingsignal);
axis([minTime maxTime Ac-1.8 5+0.8]);
xlabel('Time (s)');
ylabel('Amplitude');
title('Single-Sideband Modulated Signal');
grid on;
```

08. PWM /PDM

```
clc;
clear all;
close all;
Vm = 5;
fm = 1;
thetam = 0;
minTime = 0;
maxTime = 1;
samplePerTime = 1000;
t = minTime : 1/samplePerTime : maxTime;
t = t(1:end-1);
modulatingSignal = Vm * sin(2*pi*fm*t + thetam);
subplot(3,1,1);
plot(t, modulatingSignal);
hold on;
axis([-minTime-0.01 maxTime+0.01 -2*Vm +2*Vm]);
xlabel('Time (sec)');
ylabel('Amplitude (V)');
title('Modulating signal');
pulseTrain = [];
Vc = 5;
fc = 10;
n = fc * maxTime;
m = (samplePerTime * maxTime) / fc;
duty = 20;
on_t = duty * m / 100;
off_t = m - on_t;
for i = 1:1:n
```

```
    pulseTrain(m*(i-1)+1 : m*(i-1)+on_t) = Vc;          % for On period
    pulseTrain(m*(i-1)+on_t+1 : m*(i-1)+on_t+off_t) = 0;
end
subplot(3,1,2);
plot(t, pulseTrain);
grid on;
axis([-minTime-0.01 maxTime+0.01 -1 Vc+1]);
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
title('Carrier pulse train')
sawtoothSignal = 1.8 * Vm * sawtooth(2*pi*fc*t);
subplot(3,1,1);
plot(t, sawtoothSignal);
grid on;
modulatedSignal = zeros(size(t));
for i = 1:1:length(sawtoothSignal)
    if (modulatingSignal(i) >= sawtoothSignal(i))
        modulatedSignal(i) = Vc;
    else
        modulatedSignal(i) = 0;
    end
end
subplot(3,1,3);
plot(t, modulatedSignal);
grid on;
axis([-minTime-0.01 maxTime+0.01 -1 Vc+1]);
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
```

```
title('Modulated signal');
```

09. PPM

```
clc;
clear all;
close all;
Vm = 5;
fm = 1;
thetam = 0;
minTime = 0;
maxTime = 1;
samplePerTime = 1000;
t = minTime : 1/samplePerTime : maxTime;
t = t(1:end-1);
modulatingSignal = Vm .* sin(2*pi*fm*t + thetam);
subplot(3,1,1);
plot(t, modulatingSignal);
grid on;
hold on;
axis([-minTime-0.01 maxTime+0.01 -2*Vm +2*Vm]);
xlabel('Time (sec)');
ylabel('Amplitude (V)');
title('Modulating signal');
pulseTrain = [];
Vc = 5;
fc = 10;
n = fc * maxTime;
m = (samplePerTime * maxTime) / fc;
```

```
duty = 30;
on_t = duty * m / 100;
off_t = m - on_t;
for i = 1:1:n
    pulseTrain(m*(i-1)+1 : m*(i-1)+on_t) = Vc;
    pulseTrain(m*(i-1)+on_t+1 : m*(i-1)+on_t+off_t) = 0;
end
subplot(3,1,2);
plot(t, pulseTrain);
grid on;
axis([-minTime-0.01 maxTime+0.01 -1 Vc+1]);
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
title('Carrier pulse train');
sawtoothSignal = 1.8 * Vm * sawtooth(2*pi*fc*t);
subplot(3,1,1);
plot(t, sawtoothSignal);
modulatedSignal = zeros(1, length(pulseTrain));
for i = 1:1:n
    for j = 1:1:i*m
        if (modulatingSignal(m*(i-1)+j) < sawtoothSignal(m*(i-1)+j))
            modulatedSignal(m*(i-1)+j : m*(i-1)+j+on_t-1) = Vc;
            break
        end
    end
end
subplot(3,1,3);
plot(t, modulatedSignal);
```

```
grid on;
axis([-minTime-0.01 maxTime+0.01 -1 Vc+1]);
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
title('Modulated signal');
```

10. PCM

```
clc;
clear all;
close all;
A = 5;
f = 1;
minTime = 0;
maxTime = 1;
theta = 0;
samplePerTime = 100;
t = minTime : 1/samplePerTime : maxTime;
analogSignal = A*sin(2*pi*f*t + theta);
subplot(2,2,1);
plot(t , analogSignal);
axis([minTime maxTime -A-1 A+1]);
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
title('Analog Signal');
grid on;
samplingRate = 30*f;
totalSample = minTime : 1/samplingRate : maxTime;
sampledSignal = A*sin(2*pi*f*totalSample + theta);
```

```
subplot(2,2,2);
stem(totalSample ,sampledSignal );
axis([minTime maxTime -A-1 A+1]);
xlabel('Time (sec)');
title('Sampled Signal');
grid on;
n = 8;
L = 2^n;
minV = -A;
maxV = +A;
delta = (maxV - minV)/L;
partition = minV : delta : maxV;
codebook = minV - (delta/2) : delta : maxV + (delta/2);
[index,quants] = quantiz(sampledSignal,partition,codebook);
subplot(2,2,3);
stem(totalSample ,sampledSignal );
axis([minTime maxTime -A-1 A+1]);
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
title('Quantized Signal');
grid on;

codematrix = de2bi(index,'left-msb');
k = 1;
for i=1:1:length(index)
    for j=1:1:n
        codevector(k)=codematrix(i,j);
        k=k+1;
    end
end
disp(codevector);
subplot(2,2,4);
stairs(codevector);
title('PCM Signal');
```

11. DM

```
clc;
clear all;
close all;
A = 5;
f = 1;
theta = 0;
minTime = 0;
maxTime = 2;
samplePerTime = 100;
t = minTime : 1/samplePerTime : maxTime;
analogSignal = A*sin(2*pi*f*t + theta);
figure;
subplot(2,1,1);
plot(t , analogSignal);
axis([minTime maxTime -A-1 A+1]);
%xlabel('Time (sec)');
%ylabel('Amplitude (volts)');
%title('Analog Signal');
%grid on;
hold on;
```

```
samplingRate = 30*f;
sampleTime = minTime : 1/samplingRate : maxTime;
sampledSignal = A*sin(2*pi*f*sampleTime + theta);
digitalData = [];
staircaseSignal = 0;
delta = 1;
for i = 1:1:(samplingRate*maxTime)
    if sampledSignal(i)> staircaseSignal(i)
        digitalData(i+1) = 1;
        staircaseSignal(i+1) =  staircaseSignal(i) + delta;
    else
        digitalData(i+1) = 0;
        staircaseSignal(i+1) = staircaseSignal(i) - delta;
    end
end

disp(digitalData);
disp(staircaseSignal);
stairs(sampleTime, staircaseSignal);
ax = gca;
ax.XTick = [0:1/samplingRate:maxTime];
ax.YTick = [-(A+0.1):delta:(A+0.1)];
xlabel('Time (sec)');
ylabel('Amplitude (volts)');
title('Digital Signal');
grid on;
```

```
digitaldatalength = length(digitalData);
t1 = 0 : maxTime/(digitaldatalength-1) : maxTime;
subplot(2,1,2);

stairs(t1, digitalData);
axis([minTime-0.05 maxTime+0.05 -0.1 +1.1]);
title('Digital Output Signal');
```