DIGITAL COMMUNICATION LAB COMPLETE SOLUTIONS

1. BPSK TRANSMITTER & RECEIVER

MATLAB CODE:

```
clc; clear all; close all;
N=10^5; x=randi([0 1],1,N);
tx=2*x-1;
SNR=0:2:20;
for i=1:length(SNR)
    y=awgn(tx,SNR(i),'measured');
    rx=y>0;
    BER(i)=sum(x~=rx)/N;
end
semilogy(SNR,BER,'r*-');grid on;
xlabel('SNR(dB)');ylabel('BER');title('BPSK BER Performance');
```

2. QPSK TRANSMITTER & RECEIVER

```
clc; clear; close all;
N=1e5; data=randi([0 3],1,N);
tx=pskmod(data,4,pi/4);
SNR=0:2:20;
for i=1:length(SNR)
    rx=awgn(tx,SNR(i),'measured');
    demod=pskdemod(rx,4,pi/4);
    BER(i)=sum(data~=demod)/N;
end
semilogy(SNR,BER,'bo-');grid on;
xlabel('SNR(dB)');ylabel('BER');title('QPSK BER Performance');
```

3. DSSS TRANSMITTER & RECEIVER

```
clc; clear; close all;
data=[1 0 1 1 0];
pn=[1 0 1 0 1 0 0];
spread=[]; for i=1:length(data)
    if data(i)==1; spread=[spread pn]; else spread=[spread ~pn]; end
end
tx=2*spread-1; rx=tx; despread=[];
for i=1:length(data)
    seg=rx((i-1)*7+1:i*7); temp=sum(seg.*(2*pn-1));
    if temp>0 despread=[despread 1]; else despread=[despread 0]; end
end
disp('Recovered data:');disp(despread);
```

4. FHSS TRANSMITTER & RECEIVER

```
clc; clear; close all;
data=[1 0 1 1 0];
fc=[2 4 6 8 10]; fs=100; t=0:1/fs:1;
for i=1:length(data)
    if data(i)==1
        y=cos(2*pi*fc(i)*t);
    else
        y=sin(2*pi*fc(i)*t);
    end
    subplot(length(data),1,i); plot(t,y);
end
sgtitle('FHSS Transmitted Signals');
```

5. M-ARY PSK & QAM SIMULATION

```
clc; clear; close all;
M=16; N=1e5; data=randi([0 M-1],N,1);
```

```
psk_tx=pskmod(data,M,pi/M,'gray');
qam_tx=qammod(data,M,'gray');
figure(1); scatterplot(psk_tx); title('16-PSK Gray Constellation');
figure(2); scatterplot(qam_tx); title('16-QAM Gray Constellation');
```

6. PROBABILITY OF ERROR (16-PSK, 16-QASK, 16-FSK)

```
clc; clear;
M=16; k=log2(M); Eb=5e-8; N0=1e-9;
SNR=Eb/N0; Pe=(1/k)*erfc(sqrt(k*SNR)*sin(pi/M));
disp(['Probability of error = ',num2str(Pe)]);
```

7. OFDM TRANSMITTER & RECEIVER

```
clc; clear; close all;
N=64; M=4; data=randi([0 M-1],1,N);
modData=pskmod(data,M);
ifftData=ifft(modData);
cpLen=8; tx=[ifftData(end-cpLen+1:end) ifftData];
rx=tx; rx=rx(cpLen+1:end);
demod=pskdemod(fft(rx),M);
disp('Transmitted vs Received:'); disp([data.' demod.']);
```

8. HUFFMAN CODING

```
clc; clear; close all;
symbols=[1 2 3 4 5];
p=[0.4 0.19 0.16 0.15 0.10];
[dict,avglen]=huffmandict(symbols,p);
hcode=huffmanenco(symbols,dict);
dsig=huffmandeco(hcode,dict);
H=-sum(p.*log2(p));
eff=(H/avglen)*100;
```

```
disp('Efficiency='),disp(eff);
```

9. ENTROPY & MUTUAL INFORMATION

```
clc; clear; close all;
P=[0.3 0 0; 0 0.2 0; 0 0 0.5];
Px=sum(P,2); Py=sum(P,1);
Hx=-sum(Px.*log2(Px+eps));
Hy=-sum(Py.*log2(Py+eps));
Hxy=-sum(P(:).*log2(P(:)+eps));
I=Hx+Hy-Hxy;
disp(['H(X)=',num2str(Hx),'  H(Y)=',num2str(Hy),'  MI=',num2str(I)]);
```

10. LINEAR BLOCK CODE ENCODING/DECODING

```
clc; clear; close all;
H=[1 1 1 0 0 0; 0 1 1 1 0 0; 1 1 0 1 0 0];
G=[1 0 0 1 1 0; 0 1 0 1 0 1; 0 0 1 0 1 1];
msg=[1 0 1];
code=mod(msg*G,2);
disp('Codeword:'); disp(code);
r=code; r(3)=~r(3);
syndrome=mod(r*H',2);
disp('Syndrome:'); disp(syndrome);
```