

Scilab Textbook Companion for
Wireless Communications and Networking
by V. Garg¹

Created by
Vishal Changlani
Electronics and telecommunication engineering
Electronics Engineering
VESIT
College Teacher
None
Cross-Checked by
Spandana

July 31, 2019

¹Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website <http://scilab.in>

Book Description

Title: Wireless Communications and Networking

Author: V. Garg

Publisher: Morgan Kaufmann, San Francisco.

Edition: 2

Year: 2007

ISBN: 978-0-12-373580-5

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes	4
2 Multiple Access Techniques	5
3 Radio Propagation and Propagation PathLoss Models	15
4 An Overview of Digital Communication and Transmission	24
5 Fundamentals of Cellular Communications	31
6 Multiple Access Techniques	41
8 Speech Coding and Channel Coding	50
9 Modulation Schemes	54
10 Antennas Diversity and Link Analysis	63
11 Spread Spectrum and CDMA Systems	70
12 Mobility Management in Wireless Networks	81
13 Security in Wireless Systems	84
14 Mobile Network and Transport Layer	86

17 Planning and Design of a Wireless Network	88
19 Wireless Personal Area Network Bluetooth	104
21 Wireless Local Area Network	106
24 Spreading Codes used in CDMA	115

List of Scilab Codes

Exa 2.1	To get Gos during Busy Hour	5
Exa 2.2	To find usage in CCS and Erlangs	5
Exa 2.3	To find offered load	6
Exa 2.4	To find traffic intensity	7
Exa 2.5	To find traffic intensity	7
Exa 2.6	To find traffic intensity in Erlangs and CCS during eight hour period and busy hour . . .	8
Exa 2.7	To find traffic per subscriber per BH in Erlangs	9
Exa 2.8	To find average number of BHCAs	9
Exa 2.9	To calculate total traffic and number of MSCs required	10
Exa 2.10	To find the designed cell capacity for the switch and design Erlangs	11
Exa 2.11	To find number of service channels required to handle the load	13
Exa 2.12	To find the number of supported mobile sub- scribers	13
Exa 3.1	To find free space and reflected surface atten- uations	15
Exa 3.2	To find received signal power and SNR	16
Exa 3.3	To find the allowable path loss	17
Exa 3.4	To find the distance between transmitter and receiver	17
Exa 3.5	To determine type of fading and symbol dis- tortion	18
Exa 3.6	To determine number of fades per second and maximum velocity of mobile	19
Exa 3.7	To determine L50 pathloss	20

Exa 3.8	To determine coverage radius of an access point	21
Exa 3.9	To calculate probability of exceeding signal strength beyond receiver sensitivity	22
Exa 3.10	To find required transmitter power in watts	23
Exa 4.1	To determine the sampling rates	24
Exa 4.2	To determine the SNR	25
Exa 4.3	To calculate spacing between successive pulses of multiplexed signal	26
Exa 4.4	To calculate bits per PCM word sampling rate resultant and symbol transmission rate . . .	27
Exa 4.5	To determine choice of modulation scheme .	28
Exa 4.6	To find choice of modulation scheme without error correcting coding	29
Exa 5.1	To find system capacity	31
Exa 5.2	To find reuse factor for AMPS and GSM . .	32
Exa 5.3	To calculate call capacity of cell and Mean S by I for N as 4 and 7 and 12	33
Exa 5.4	To calculate the number of calls per hour per cellsite	35
Exa 5.5	To calculate calls per hour per cellsite and mean S by I and spectral efficiency	36
Exa 6.1	To calculate spectral efficiency of modulation	41
Exa 6.2	To find multiple access spectral efficiency for FDMA	42
Exa 6.3	To find multiple access spectral efficiency of the TDMA system	42
Exa 6.4	To calculate the capacity and spectral efficiency of TDMA system	43
Exa 6.5	To calculate the frame efficiency and the number of channels per frame	44
Exa 6.6	To calculate capacity and spectral efficiency of the DSCDMA system	45
Exa 6.7	Compare DS CDMA and TDMA omnidirectional cell	46
Exa 6.8	To calculate minimum number of PN chips per frequency word	47
Exa 6.9	To find normalized throughput for different CSMA protocols	47

Exa 6.10	To find data link protocol efficiency with different protocols	48
Exa 8.1	To calculate the gain in the link budget in dB	50
Exa 8.2	To calculate output of convolution encoder .	51
Exa 8.3	To demonstrate operations of converting burst errors into bit errors	52
Exa 9.1	To find E_b by N_0 in dB	54
Exa 9.2	To calculate amplitude A of the carrier signal	55
Exa 9.3	To calculate final phase for given bitstream for given modulation method	55
Exa 9.4	To calculate frequency shift along with transmitted frequencies and bandwidth efficiency	57
Exa 9.5	To calculate bit error probability for GMSK	58
Exa 9.6	To calculate bit rate of modulator	59
Exa 9.7	To determine the modulation scheme to be used along with required E_b by N_0	59
Exa 9.8	Compare the performance of 16PSK with 16QAM	60
Exa 9.9	To find total bandwidth and its efficiency along with required E_b by N_0 and carried bits per symbol	61
Exa 10.1	To find received signal power and SNR of antenna	63
Exa 10.2	To find received signal power and SNR of antenna	64
Exa 10.3	To calculate gain of antenna	65
Exa 10.4	To determine 3dB beam width of an antenna	66
Exa 10.5	To calculate various parameters of helical antenna	66
Exa 10.6	To find probability that SNR will drop below 10dB and compare result with a case of without diversity	67
Exa 10.7	To calculate minimum delay difference to successfully resolve multipath components	68
Exa 11.1	To calculate processing gain and improvement in information rate achieved	70
Exa 11.2	Show that the transmitted signals to mobiles 1 2 and 3 are recovered at the mobile receivers by despreading the resultant signal z	71

Exa 11.3	Minimum number of PN chips required for each frequency symbol	73
Exa 11.4	To calculate data symbol transmitted per hop and number of non overlapping hop frequencies	73
Exa 11.5	To consider various parameters of FHSS systems	74
Exa 11.6	To show how mobile1 will detect its information using a two path rake receiver	75
Exa 11.7	To calculate the time required by mobile to make the required change	78
Exa 11.8	To determine the possible pair of soft slope and add intercept values	79
Exa 12.1	To evaluate the impact of LUs on the radio resource and calculate MSC or VLR transaction load	81
Exa 13.1	To generate public and private keys for RSA algorithm	84
Exa 13.2	To determine the secret encrypting key K using DH key exchange algorithm	85
Exa 14.1	To determine minimum possible latency and window size to achieve this latency	86
Exa 14.2	To calculate upper bound of the throughput for TP connection and throughput with retransmission due to errors	87
Exa 17.1	To calculate various parameters for GSM 1800 network	88
Exa 17.2	To estimate the data and voice traffic per subscriber and per cell	89
Exa 17.3	To calculate data Erlangs along with TS utilization and capacity	91
Exa 17.4	To develop downlink and uplink cell budget and calculate cell radius	92
Exa 17.5	To calculate Uplink cell load factor and pole capacity	93
Exa 17.6	To calculate uplink throughput for WCDMA data service	94
Exa 17.7	To calculate downlink cell load factor and number of voice users per cell	94

Exa 17.8	To determine minimum signal power and maximum allowable path loss	95
Exa 17.9	To develop a radio link budget for uplink and downlink	96
Exa 17.10	To find the number of users supported on the downlink of a WCDMA Network	97
Exa 17.11	find the average throughput for various cases	98
Exa 17.12	TO find the allowable throughput of a reverse link of a CDMA 2000	100
Exa 17.13	To estimate average SINR of a HSDPA . . .	100
Exa 17.14	To find bandwidth of a Iub interface	101
Exa 17.15	To find the required RNCs	102
Exa 19.1	To find hopping rate and various other parameters of Bluetooth	104
Exa 19.2	To find the associated data rate of a symmetric 1soot DM1	105
Exa 21.1	To find number of users supported by WLAN and its bandwidth efficiency	106
Exa 21.2	To find hopping bandwidth and various other parameters for FH MFSK WLAN system .	107
Exa 21.3	To find bandwidth of a subchannel and various other parameters of a OFDM WLAN system	108
Exa 21.4	To find coverage of the AP	109
Exa 21.5	To calculate coded symbol and bit transmission rate per subscriber of the two modes . .	110
Exa 21.6	To find user data rate for HIPERLAN 2 . .	111
Exa 21.7	To determine the collision probability of a FH packet	111
Exa 21.8	To calculate min SIR and using this SIR and other data calculate Rmax	112
Exa 21.9	To calculate Rmax for given interference scenarios	113
Exa 21.10	To calculate Rmax for IEEE 802 11 FH and DS devices	114
Exa 24.1	To generate m sequence and demonstrate its properties	115
Exa 24.2	To generate second m sequence	116

Chapter 2

Multiple Access Techniques

Scilab code Exa 2.1 To get Gos during Busy Hour

```
1 // Exa 2.1
2 // TO get Gos during BH(Busy Hour).
3
4 clc;
5 clear all;
6
7 LC=10; //Lost calls
8 CC=380; //Carried calls
9
10 // soution
11 OC=LC+CC; //Total offered calls
12 //Gos=Blocking probability=(number of Lost calls/
    Total number of offered calls)
13 Gos=LC/OC;
14 printf('The Gos during busy hour is %f \n ', Gos);
```

Scilab code Exa 2.2 To find usage in CCS and Erlangs

```

1 // Exa 2.2
2 // To find usage in call-seconds,CCS(centrum call
   seconds) and Erlangs).
3
4 clc;
5 clear all;
6
7 Ht=5; //Average holding time in seconds
8 PC=450; // Peg count for one hour period
9 OC=0; // Overflow count
10
11 // solution
12 // usage(Erlangs)=(peg count- overflow count)*
   Average holding time(in hrs)
13 U=(PC-OC)*(5/3600);
14 printf('Usage = %.3f Erlangs = ',U);
15
16 // IN CCS
17 Uccs=U*36; // usage in CCS
18 printf('%.1f CCS = ',Uccs);
19
20 Ucs=Uccs *100; //usage in call-seconds
21 printf('%d call-seconds\n',Ucs);

```

Scilab code Exa 2.3 To find offered load

```

1 // Exa 2.3
2 //TO find Offered Load.
3
4 clc;
5 clear all;
6
7 B=0.05; //Blocking (5%)
8 CL=3000; //Carried Load in CCS
9

```

```

10 //Solution
11 //Offered Load=Carried Load/(1-Blocking);
12 OL=CL/(1-B); //Offered Load in CCS
13 printf('Offered load is %d CCS \n',round(OL));
14 printf(' Overflow is %d CCS \n',round(OL)-CL);

```

Scilab code Exa 2.4 To find traffic intensity

```

1 // Exa 2.4
2 //To find traffic intensity.
3
4 clc;
5 clear all;
6
7 Ht=120; //Holding time in seconds
8 CR=2; //call rate per hour
9
10 //solution
11 //Traffic Intensity(Erlangs) = call rate * Holding
    time(in hrs));
12 I=CR*Ht/3600; //Traffic Intensity
13 printf('Traffic Intensity is %.4f Erlangs = %.1f CCS
    \n',I,I*36);

```

Scilab code Exa 2.5 To find traffic intensity

```

1 //Exa 2.5
2 //To find traffic intensity in Erlangs and CCS.
3 //Refer-Table 2.1(page No 28): Traffic data used to
    estimate traffic intensity
4
5 clc;
6 clear all;

```

```

7
8 time=90; //in minutes
9 calls=10; //no of calls in 90mins
10
11 //solution
12 CR=calls/(time/60); //call arrival rate in calls/
    hour
13 tavg=(60+74+80+90+92+70+96+48+64+126)/10; //
    average call holding time in sec per call
14 I= CR*(tavg/3600); //traffic intensity in
    Erlangs
15 printf('Traffic Intensity is %.3f Erlangs = %.2f CCS
    \n ',I,I*36);

```

Scilab code Exa 2.6 To find traffic intensity in Erlangs and CCS during eight hour

```

1 // Exa 2.6
2 //To find traffic intensity in Erlangs and CCS
    during eight hour period and busy hour(4:00 PM to
    5:00 PM).
3 //Refer-Table 2.2(page no 28): Traffic on customer
    line between 9:00PM and 5:00PM
4
5 clc;
6 clear all;
7
8 time=8; //in hours
9 calls=11; //no of calls in 90mins period
10
11 //solution
12 CR=calls/time; //call arrival rate in calls/hour
13 tavg=(3+10+7+10+5+5+1+5+15+34+5)/11; //average
    call holding time in mins per call
14 I= CR*(tavg/60); //traffic intensity in Erlangs
15 printf('Traffic Intensity during eight hour period

```

```

    is %.3f Erlangs = %.1f CCS \n',I,I*36);
16
17 //For Busy Hour i.e between 4:00PM and 5:00PM
18 CRB=2; //call arrival rate during busy hour in
    calls/hour(from table 2.2)
19 tavgB=(34+5)/2; //average holding time during Busy
    Hour in mins/call (from table 2.2)
20 IB=CRB*(tavgB/60); //Traffic Intensity during Busy
    Hour
21 printf(' Traffic Intensity during busy hour is %.2f
    Erlangs = %.1f CCS',IB,IB*36);

```

Scilab code Exa 2.7 To find traffic per subscriber per BH in Erlangs

```

1 //Exa 2.8
2 // To find traffic per user per BH in Erlangs.
3
4 clc;
5 clear all;
6
7 minutes=500; //No of minutes used per month per user
8 Twork=0.9; //Traffic During Work day
9 TBH=0.1; //Traffic during busy hour
10 Days=20; //No of workdays in a month
11
12 //solution
13 //Avg BH usage per subscriber (in minutes) = minutes
    *Twork(TBH/Days);
14 Traffic=minutes*Twork*(TBH/Days);
15 printf('Traffic per user per BH is %.4f Erlangs \n',
    Traffic/60);

```

Scilab code Exa 2.8 To find average number of BHCAs

```

1 //Exa:2.8
2 // To find average busy hour call attempts(BHCA's).
3
4 clc;
5 clear all;
6
7 minutes=500; //No of minutes used per month per user
8 Twork=0.9; //Traffic During Work day
9 TBH=0.1; //Traffic during busy hour
10 Days=20; //No of workdays in a month
11 MeanHT=100; //Mean holding time(in secs)
12
13 //solution
14 //Avg BH usage per subscriber (in minutes) = minutes
    *Twork(TBH/Days);
15 //BHCA= t r a f f c (in Erlangs) *3600/(meanHT)
16
17 Traffic=minutes*Twork*(TBH/Days);
18 BHCA=(Traffic/60)*(3600/MeanHT);
19
20 printf('Average Busy Hour call attempts are %.2f ',
    BHCA);

```

Scilab code Exa 2.9 To calculate total traffic and number of MSCs required

```

1 //Exa 2.9
2 // To find total traffic in Erlangs and no of MSCs
    required to handle it.
3
4 clc;
5 clear all;
6
7 Tpopu=200000; //Total population
8 SP=0.25; //subscriber penetration
9 HT1=100; //holding time for Mobile to Land line and

```



```

        vicecersa
10 c1=3; //Avg calls/hr for Mobile to Land line and
    vicecersa
11 HT2=80; //For mobile to mobile
12 c2=4; //For mobile to mobile
13 TMSC=1800; //traffic one msc can hold
14 TrafDist=0.9 //Traffic distribution for Mobile to
    Land line and vicecersa
15
16 //solution
17 aM_L=c1*HT1/3600; //Traffic Generated by Subscriber
    (M-L and L-M).
18 aM_M=c2*HT2/3600; //Traffic Generated by Subscriber
    (M-M).
19 WlessSub=SP*Tpopu; //total wireless subscribers
20 TotalTraffic=WlessSub*TrafDist*aM_L+WlessSub*(1-
    TrafDist)*aM_M;
21 MSCreqd=TotalTraffic/TMSC;
22 if(MSCreqd-int(MSCreqd)>0)//for rounding of to next
    integer ef 2.33 to 3
23     MSCreqd=MSCreqd+1;
24     end
25 printf('Total Traffic is %.1f Erlangs \n',
    TotalTraffic);
26 printf(' NO of MSCs Required are %d \n',int(MSCreqd)
    );

```

Scilab code Exa 2.10 To find the designed cell capacity for the switch and design

```

1 //Exa_2.10
2 // TO find ABS/BH(average busy season per busy hour)
    calling rates , design cell capacity for the
    switch and design Erlangs.
3
4 clc;

```

```

5 clear all;
6
7 Rlines=15000; //Residential lines
8 Blines=80000; //Business lines
9 PWElines=5000; //PBX, WATS, and Foreign Exchange (FX)
   lines
10 CR_R=2; //Call rates for Rlines
11 CR_B=3; // call rates for Blines
12 CR_PWE=10; //call rates for PWElines
13 HT_R=140; //average holding time for Rlines(sec)
14 HT_B=160; //average holding time for Blines(sec)
15 HT_PWE=200; //average holding time for PWE lines(sec)
16 Slines=100000; // no of lines carried by switch
17 HD_ABS=1.5; // HD/ABS for the switch
18
19 //solution
20 percentR_lines=Rlines/Slines;
21 percentB_lines=Blines/Slines;
22 percentPWE_lines=PWElines/Slines;
23 CCSR=CR_R*HT_R/100;
24 CCSB=CR_B*HT_B/100;
25 CCSPWE=CR_PWE*HT_PWE/100;
26
27 CR=CR_R*percentR_lines+CR_B*percentB_lines+CR_PWE*
   percentPWE_lines;
28 printf('The call rate is %.1f calls per line \n ',CR
   );
29 CCS=CCSR*percentR_lines+CCSB*percentB_lines+CCSPWE*
   percentPWE_lines;
30 AvgHTperline=CCS*100/CR;
31 ABS_BH_calls=CR*Slines;
32 ABS_BH_usage=CCS/36*Slines;
33 printf('Design cell capacity based on HD is %d calls
   \n ',HD_ABS*ABS_BH_calls);
34 printf(' DESIGN Erlangs based on HD is %d \n ',round(
   HD_ABS*ABS_BH_usage));

```

Scilab code Exa 2.11 To find number of service channels required to handle the load

```
1 // Exa 2.11
2 // To find offered load A and number of service
   channels required to handle the load
3
4 clc;
5 clear all;
6
7 maxcalls_hour=4000; //maximum call per hour
8 avgHT=160; //average holding time in sec
9 Gos=0.02;
10
11 //solution
12 A=maxcalls_hour*avgHT/3600; //offered load
13 printf('Offered load A = %.2f Erlangs \n ',A);
14 //Referring Appendix A i.e Erlang B table
15 disp("For calculated Offered load and referring
   Erlang B table we get Service channels as 182
   giving 168.3 Erlangs at 2% blocking");
```

Scilab code Exa 2.12 To find the number of supported mobile subscribers

```
1 // Exa 2.12
2 // To calculate number of mobile subscribers
   supported for the given system.
3
4 clc;
5 clear all;
6
7 channels=50;
8 blocking=0.02;
```

```
9 HT=120; //average holding time inm sec
10 BHcall=1.2; // in calls per hour
11
12 //solution
13 //Refering Erlang B table in appendix A, For 50
    channels at 2% blocking, the offered load=40.26
    Erlangs.
14 A=40.26;
15 B=A*(1-0.02); //carried load
16 Avgtraffic_user=BHcall*HT/3600;
17 No_users=B/Avgtraffic_user;
18 printf('NO of mobile subscribers supported are %d \n
    ',round(No_users));
```

Chapter 3

Radio Propagation and Propagation PathLoss Models

Scilab code Exa 3.1 To find free space and reflected surface attenuations

```
1 // Exa 3.1
2 // To determine free space and reflected surface
   attenuations.
3
4 clc;
5 clear all;
6
7 hb=100; //in feets (height of BS antenna)
8 hm=5; // in feets (height of mobile antenna)
9 f=881.52; //in MHz
10 lamda=1.116; //in feet
11 d=5000; //in feet
12 Gb=100.8; //8dB (BS antenna gain)
13 Gm=100; // 0dB (Mobile antenna gain)
14
15 //solution
16 free_atten=(4*%pi*d/lamda)2*(Gb*Gm)-1;
17 y=round(10*log10(free_atten));
18 printf('Free space attenuation is %d dB \n',y);
```

```

19 reflect_atten= (d^4/(hb*hm)^2)*(Gb*Gm)^-1;
20 x=round(10*log10(reflect_atten));
21 printf(' Reflecting surface attenuation is %d dB \n
        ',x);

```

Scilab code Exa 3.2 To find received signal power and SNR

```

1 //Exa 3.2
2 //To determine received signal power and SNR ratio.
3
4 clc;
5 clear all;
6
7 d=8000; //Distance between base station and mobile
        station
8 f=1.5*10^9; //in Hz
9 lamda=0.2; //in metres
10 Pt=10; //BS transmitted power in watts
11 Lo=8; //Total system losses in dB
12 Nf=5; //Mobile receiver noise figure in dB
13 T=290; //temperature in degree kelvin
14 BW=1.25*10^6; //in Hz
15 Gb=8; //in dB
16 Gm=0; //in dB
17 Hb=30; //in metres
18 Hm=3; //in metres
19 B=1.38*10^-23; //Boltzmann's constant
20
21 //solution
22 Free_Lp=20*log10(Hm*Hb/d^2);
23 Pr=Free_Lp-Lo+Gm+Gb+Pt; //in dBW
24 Te=T*(3.162-1);
25 Pn=B*(Te+T)*BW;
26 printf('Received signal power is %d dBW \n',10*log10
        (Pn));

```

```
27 SNR=Pr-10*log10(Pn);
28 printf(' SNR ratio is %d dB \n',round(SNR));
```

Scilab code Exa 3.3 To find the allowable path loss

```
1 //Exa 3.3
2 //To determine received power and allowable Path
  loss.
3
4 clc;
5 clear all;
6
7 d=3*1000;//in metres
8 Y=4;// path loss exponent
9 Pt=4; //Transmitted power in watts
10 f=1800*10^6;//in Hz
11 Shadow=10.5; //in dB
12 d0=100;//in metres
13 P0=-32; //in dBm
14
15 //solution
16 disp("Using equation 3.11 and including shadow
  effect we get")
17 Pr=P0+10*Y*log10(d0/d)+Shadow;
18 printf(' Received power is %.1f dBm \n',Pr);
19 path_loss=10*log10(Pt*1000)-Pr;
20 printf(' Allowable path loss is %.1f dB \n ',
  path_loss);
```

Scilab code Exa 3.4 To find the distance between transmitter and receiver

```
1 //Exa 3.4
```

```

2 //To determine distance between transmitter and
   receiver.
3
4 clc;
5 clear all;
6
7 shadow=10; //in dB
8 Lp=150; //in dB
9
10 //solution
11 disp(" Using equation given in Problem i.e Lp
      =133.2+40*log(d) we get ,");
12 d=10^((Lp-10-133.2)/40);
13 printf(" Separation between transmitter and
      receiver as %.2f km',d);

```

Scilab code Exa 3.5 To determine type of fading and symbol distortion

```

1 // Exa 3.5
2 // To calculate coherence time, coherence bandwidth,
   type of Symbol distortion and type of fading.
3
4 clc;
5 clear all;
6
7 v=60*0.44704; //.. mph to mps
8 fc=860*10^6; //in Hz
9 td=2*10^-6; //RMS delay spread in sec
10 c=3*10^8; // speed of light in m/sec
11 Rs=19200; //Coded symbol rate in bps
12
13 //solution
14 lamda=c/fc;
15 fm=v/lamda; //Maximum doppler shift
16 tc=1/(2*%pi*fm); //Channel coherence time

```



```

17 printf('Channel coherence time is %.4f sec \n',tc);
18 ts=1/Rs; //symbol interval
19 printf(' Symbol interval is %d microsec \n',ts*10^6)
    ;
20 disp(" As the symbol interval is much smaller
        compared to the channel coherence time. So,
        Symbol distortion is minimal and fading is slow."
        );
21 disp("");
22 Bc=1/(2*pi*td);
23 printf(' Coherence Bandwidth is %.2f kHz \n',Bc
        /1000)

```

Scilab code Exa 3.6 To determine number of fades per second and maximum velocity of

```

1 // Exa 3.6
2 // TO determine NO of fades per second, average fade
   duration and maximum velocity of mobile.
3
4 clc;
5 clear all;
6
7 p=1; // reflection coefficient of ground
8 c=3*10^8; // velocity of light in free space(m/sec)
9 e=2.71828; //Euler's number
10 fm=20; //in Hz
11 fc=900*10^6; //carrier frequency in Hz
12
13 //solution
14 Nr=sqrt(2*pi)*fm*p*e^-(p^2);
15 printf('NO of fades per second are %.2f \n',Nr);
16 Afd=e^-(p^2)/(p*fm*sqrt(2*pi));
17 printf(' Average fade duration is %.4f sec \n ',Afd)
    ;
18 v=fm*c/fc;

```

```

19 printf('Maximum velocity of mobile is %.2f m/sec =
    %d Km/hour \n',v,v*18/5);

```

Scilab code Exa 3.7 To determine L50 pathloss

```

1 // Exa 3.7
2 // To calculate L50 path loss for a PCS system using
    Okumura and COST231 models.
3
4 clc;
5 clear all;
6
7 d=[1 2 3 4 5]; //in km
8 hb=30; //Height of BS antenna in metres
9 hm=2; // height of mobile antenna in metres
10 fc=900; //carrier frequency in MHz
11 W=15; //street width(m)
12 b=30; // distance between building along radio path
    (m)
13 phi=90; // incident angle relative to the street
14 hr=30; //in m
15
16 //solution
17 dellhm=hr-hm;
18 //L50=Lf+Lrts+Lms
19 // By COST 231 model
20 Lf=32.4+20*log10(d)+20*log10(fc);
21 L0=4-0.114*(phi-55);
22 Lrts=-16.9-10*log10(W)+10*log10(fc)+20*log10(dellhm)
    +L0;
23 Lbsh=-18*log10(11);
24 ka=54-0.8*hb;
25 dellhb=hb-hr;
26 kd=18-15*dellhb/dellhm;
27 kf=4+0.7*(fc/925-1);

```

```

28 Lms=Lbsh+ka+kd*log10(d)+kf*log10(fc)-9*log10(b);
29 L50=[0 0 0 0 0];
30 L50=Lf+Lrts+Lms;
31 //Okumura/Hata model
32 ahm=(1.1*log10(fc)-0.7)*hm-(1.56*log10(fc)-0.8);
33 L_50=69.55+26.16*log10(fc)+(44.9-6.55*log10(hb))*
      log10(d)-13.82*log10(hb)-ahm;
34 xlabel("DISTANCE FROM TRANSMITTER IN KM");
35 ylabel("PATH LOSS in dB");
36 plot2d(d,[L50',L_50'],[1,2]);
37 legends(['Cost 231 Model';'Okumura/Hata Model'],[1,2
      ],opt=2)
38 xgrid();
39 disp("L50 values by Cost 231 model");
40 printf('%0.2f %0.2f %0.2f %0.2f %0.2f \n ',L50(1),L50
      (2),L50(3),L50(4),L50(5));
41 disp("L50 values bu Okumura/Hata model");
42 printf('%0.2f %0.2f %0.2f %0.2f %0.2f \n ',L_50(1),
      L_50(2),L_50(3),L_50(4),L_50(5));
43 disp("The results from the plot of two models shows
      that the calculated path loss with the COST 231
      model is higher than the value obtained by the
      Okumura/Hata model.");

```

Scilab code Exa 3.8 To determine coverage radius of an access point

```

1 // Exa 3.8
2 // TO find coverage radius of an access point.
3
4 clc;
5 clear all;
6
7 SNRmin=12; //in dB
8 n=3; //No of floors
9 Backgroundnoise=-115; //dBm

```

```

10 pt=100 //in dBm
11
12 //solution
13 pt_db=10*log10(pt);
14 Sr=Backgroundnoise+SNRmin; //receiver sensitivity
15 Lpmax=pt_db-Sr;
16 //Referring table 3.4
17 Lp_d0=38; //ref path loss at the first meter(dB)
18 Lf=15+4*(n-1); //signal attenuation through n floors
19 y=3; //path loss exponent
20 X=10; //Shadowing effect(dB)
21 d=10^((Lpmax-Lp_d0-Lf-X)/30); //max allowable path
    loss
22 printf('Coverage radius of an access point = %d m \n
    ',round(d));

```

Scilab code Exa 3.9 To calculate probability of exceeding signal strength beyond r

```

1 //Exa 3.9
2 // To calculate probability of exceeding signal
    beyond the receiver sensitivity.
3
4 clc;
5 clear all;
6
7 SSmean=-100; //signal strength(dBm)
8 Sr=-110; //receiver sensitivity(dBm)
9 sd=10; //standard deviation(dB)
10
11 //solution
12 P_Smin=(0.5-0.5*erf((Sr-SSmean)/(sqrt(2)*sd)));
13 printf('probability of exceeding signal beyond the
    receiver sensitivity is %.2f \n',P_Smin);

```

Scilab code Exa 3.10 To find required transmitter power in watts

```
1 //Exa 3.10
2 //To find required total transmit power in Watts.
3
4 clc;
5 clear all;
6
7 Lp=140; // path losses in dB
8 k=1.38*10^-23; // Boltzmanns constant (W/Kelvin-
    Hz)
9 k_db=10*log10(k);
10 f=900; //in MHz
11 Gt=8; //transmitting antenna gain(dB)
12 Gr=0; //receiver antenna gain(dB)
13 Ag=24; //gain of receiver amplifier in dB
14 Fmargin=8; //Fade margin(dB)
15 Nf=6; //Noise figure(dB)
16 L0=20; // other losses in dB
17 Lf=12; // antenna feed line loss in dB
18 T=24.6; //Temperature expressed in dB
19 R=39.8; // data rate in dB
20 M=8; //overall link margin(dB)
21 Eb_No=10; //dB
22
23 //solution
24 //From equation (3.54)
25 pt_db=M-Gt-Gr-Ag+ Nf + T+ k_db+ Lp+ Lf+ L0 + Fmargin
    + R+ Eb_No;
26
27 Pt=10^(pt_db/10); //dB into normal number
28 printf('Total transmitted power is %d Watts \n',Pt);
```

Chapter 4

An Overview of Digital Communication and Transmission

Scilab code Exa 4.1 To determine the sampling rates

```
1 // Exa 4.1
2 // To calculate the sampling rate.
3
4 clc;
5 clear all;
6
7 Fm=20; // in KHz
8
9 //solution
10 disp(" An Engineering version of the Nyquist
      sampling rate : fs >=2.2*fm.");
11 printf('Therefore sampling rate of >= %d ksp/s should
      be used ',(2.2*Fm));
12 disp("The sampling rate for a compact disc digital
      audio player = 44.1 ksp/s and for a studio quality
      audio player = 48 ksp/s are used.")
```

Scilab code Exa 4.2 To determine the SNR

```
1 // Exa 4.2
2 // To calculate SNR for L=32, 64, 128, and 256.
3
4 clc;
5 clear all;
6
7 Rt=1; //Resistance(ohm)
8 //L= Number of quantization values
9 L1=32;
10 L2=64;
11 L3=128;
12 L4=256;
13
14 // solution
15 //  $L=2^R$  i.e  $R=\log_2(L)$ ;
16 R1=log2(L1);
17 R2=log2(L2);
18 R3=log2(L3);
19 R4=log2(L4);
20
21 //P=A^2/2; //average power of signal
22 // sig^2=0.333*A^2*2^(-2*Rt); //Avg quantization
    noise power
23 //SNR=P/sig^2;
24 // SNR(dB)=1.8+ 6R;
25
26 SNR1=1.8+6*R1;
27 SNR2=1.8+6*R2;
28 SNR3=1.8+6*R3;
29 SNR4=1.8+6*R4;
30
31 printf('For L=32, SNR is %.1f dB\n ',SNR1);
```

```

32 printf('For L=64, SNR is %.1f dB\n ',SNR2);
33 printf('For L=128, SNR is %.1f dB\n ',SNR3);
34 printf('For L=256, SNR is %.1f dB\n ',SNR4);

```

Scilab code Exa 4.3 To calculate spacing between successive pulses of multiplexed

```

1 // Exa 4.3
2 // To calculate the spacing between successive
   pulses of the multiplexed signal.
3
4 clc;
5 clear all;
6
7 Fs=8*10^3; //in Hz
8 Fm=3.4*10^3; // in Hz
9 VCH=24; //voice channels
10 SCH=1; //synchronization channel
11 PDur=1; //extra pulse duration in microsec
12
13 //solution
14 Ts=1/(Fs);
15 TimeCH=Ts/(VCH+SCH)*10^6; // in microsec
16 printf('Time between the pulses is %d microsec\n',(
   TimeCH-PDur));
17 //Now by using the engineering version of Nyquist
   rate sampling
18 NyquistRate=2.2*Fm;
19 Ts1_microsec=1/NyquistRate*10^6;
20 Tc=round(Ts1_microsec)/(VCH+SCH);
21 printf(" Time between the pulses by using
   engineering version of Nyquist rate sampling is
   %.2f microsec\n", (Tc-PDur));

```

Scilab code Exa 4.4 To calculate bits per PCM word sampling rate resultant and sym

```
1 // Exa 4.4
2 // TO calculate:
3 // A)The minimum number of bits/sample or bits/PCM
   word that should be used.
4 // B)The minimum sampling rate, and what is the
   resulting transmission rate.
5 // C)The PCM pulse or symbol transmission rate.
6
7 clc;
8 clear all;
9
10 Fm=3000; //highest modulating frequency in signal(Hz
   )
11 M=32; // number of pulse levels
12 b=5; //bits per symbol
13 p=0.01; //Quantization distortion
14
15 //solution
16 //2^R = L >= 1/2P
17 // where R is the number of bits required to
   represent quantization levels L
18 R=log10(1/(2*p))/log10(2);
19 Fs=2*Fm; // Nyquist sampling criteria (samples per
   second)
20 fs=round(R)*Fs;
21 Rs=fs/b;
22 printf('The minimum number of bits/sample or bits/
   PCM word that should be used are %d',round(R));
23 printf('\\n The minimum sampling rate is %d samples
   per second\\n ',Fs);
24 printf('The resulting transmission rate is %d bps\\n
   ',fs);
25 printf('The PCM pulse or symbol transmission rate is
   %d symbols/sec\\n',Rs);
```

Scilab code Exa 4.5 To determine choice of modulation scheme

```
1 // Exa 4.5
2 // To determine choice of modulation scheme if no-
   error correction coding is used.
3
4 clc;
5 clear all;
6
7 S_No=53; //dB-Hz
8 R=9.6*10^3; //bps
9 BW=4.8*10^3; //Khz
10 Pb=10^-5; //BER<=10^-5;
11
12 //solution
13 disp("Since the required data rate of 9.6 kbps is
   more than the available bandwidth of 4.8 kHz, the
   channel is bandwidth-limited.");
14 Eb_No=S_No-10*log10(R); //dB
15 // Try for 8-PSK modulation scheme
16 M=8;
17 Ps=log2(M)*Pb; //Max ps
18 Es_No=log2(M)*10^(0.1*Eb_No);
19 //Ps(8)=2*Q(sqrt(2*Es_No)*sin(%pi/8));
20 //2*Q(sqrt(2*Eb_No))=erfc(sqrt(Eb_No)); //Refer EQn
   C(7) from appendix C
21
22 Ps8=erfc(sqrt(Es_No)*sin(%pi/8));
23 disp("");
24 printf('Symbol error rate is given as %.5f \n ',Ps)
   ;
25 printf('The ratio of signal energy to noise is %.2f
   \n ',Es_No);
26 printf('Symbol error rate for 8-PSK is %.5f \n ',
```

```

    Ps8);
27 disp("As symbol error rate for 8-PSK modulation is
    lower than threshold value. so, We can use 8-PSK
    modulation.")

```

Scilab code Exa 4.6 To find choice of modulation scheme without error correcting c

```

1 // Exa 4.6
2 // To determine design choice of modulation scheme
  without an error correction coding.
3
4 clc;
5 clear all;
6
7 SNR=48; //dB-Hz
8 BW=45*10^3; //in Hz
9 R=9.6*10^3; //bps
10 Pb=10^-5; //Bit error rate
11 e=2.71828; //Natural exponent e
12
13 //solution
14 disp(" since the available bandwidth of 45 kHz is
    more than adequate to support the required data
    rate of 9.6 kbps.");
15 disp("So, the channel is not bandwidth limited ");
16 Eb_No=SNR-10*log10(R);
17 //We try the 16-FSK modulation scheme
18 M=16;
19
20 Es_No=log2(M)*Eb_No;
21 Ps=(M-1)/2*e^(-Es_No/2);
22 //For orthogonal signalling
23 Ps16=(2^M-1)/(2^(M-1))*Pb;
24 disp("");
25 printf(' The maximum symbol error probability is %0

```

```
    .5f \n ',Ps16);  
26 printf('The symbol error probability achieved by 16-  
    PSK is %.9f \n ',Ps);  
27 disp("As achieved symbol error probability is far  
    less than maximum tolerable value");  
28 disp("So, we can meet the given specifications for  
    this power-limited channel with a 16-FSK  
    modulation scheme without any error-correction  
    coding")
```

Chapter 5

Fundamentals of Cellular Communications

Scilab code Exa 5.1 To find system capacity

```
1 // Exa 5.1
2 // To Calculate
3 // A) The system capacity if the cluster size , N (
   reuse factor), is 4 and
4 // B) The system capacity if the cluster size is 7.
5 // C) How many times would a cluster of size 4 have
   to be replicated to cover the entire cellular
   area?
6 // D) Does decreasing the reuse factor N increase
   the system capacity?
7
8 clc;
9 clear all;
10
11 ToCH=960; // Total available channels
12 Cellarea=6; //in km2
13 Covarea=2000; //in km2
14 N1=4; // Cluster Size
15 N2=7; //Cluster Size
```

```

16
17 //solution
18 Area1=N1*Cellarea;//for N=4
19 Area2=N2*Cellarea;//For N=7
20 No_of_clusters1=round(Covarea/Area1);
21 No_of_clusters2=round(Covarea/Area2);
22 No_of_CH1=ToCH/N1;    // No of channels with cluster
    size 4
23 No_of_CH2=ToCH/N2;    // No of channels with cluster
    size 7
24 SysCap1=No_of_clusters1*ToCH;
25 SysCap2=No_of_clusters2*ToCH;
26 printf(' System Capacity with cluster size 4 is %d
    channels \n ',SysCap1);
27 printf(' Number of clusters for covering total area
    with N equals 4 are %d \n ',No_of_clusters1);
28 printf(' System Capacity with cluster size 7 is %d
    channels \n ',SysCap2);
29 disp("      It is evident when we decrease the value
    of N from 7 to 4, we increase the system
    capacity from 46080 to 79680 channels. Thus,
    decreasing the reuse factor (N) increases the
    system capacity.")

```

Scilab code Exa 5.2 To find reuse factor for AMPS and GSM

```

1 // Exa 5.2
2 // To calculate reuse factor for AMP and GSM systems
3
4 clc;
5 clear all;
6
7 S_IAMP=18;// S/I ratio in dB
8 S_IGSM=12;// S/I ratio in dB

```

```

9 PPL=4; // propogation path loss coefficient
10
11 //solution
12 // Using Equation 5.16 on page no 132, we get
13 N_AMP=(1/3)*((6*10^(0.1*S_IAMP))^(2/PPL)); //reuse
    factor for AMPS
14
15 N_GSM=(1/3)*((6*10^(0.1*S_IGSM))^(2/PPL)); //reuse
    factor for GSM
16
17 printf('Reuse Factor for AMP system is N = %f =
    approx %d \n',N_AMP,N_AMP+1);
18 printf(' Reuse Factor for GSM system is N = %f =
    approx %d \n',N_GSM,N_GSM+1);

```

Scilab code Exa 5.3 To calculate call capacity of cell and Mean S by I for N as 4

```

1 // Exa 5.3
2 // To calculate
3 // A) The number of calls per cell site per hour (i.
    e., call capacity of cell).
4 // B) Mean S/I ratio for cell reuse factor equal to
    4, 7 and 12.
5
6 clc;
7 clear all;
8
9 VCH=395; //Total voice channels
10 CallHT=120; //average call holding time in sec
11 Blocking=0.02; // 2%
12 PPL=4; //propogation path loss coefficient
13 N1=4 //reuse factor
14 N2=7; //reuse factor
15 N3=12; //reuse factor
16

```

```

17 //solution
18 No_of_VCH1=VCH/N1; //for reuse factor N1
19 No_of_VCH2=VCH/N2; //for reuse factor N2
20 No_of_VCH3=VCH/N3; //for reuse factor N3
21 printf('\nNO of voice channels for N=4 are %d',round
    (No_of_VCH1));
22 printf('\nNO of voice channels for N=7 are %d',round
    (No_of_VCH2));
23 printf('\nNO of voice channels for N=12 are %d\n',
    round(No_of_VCH3));
24 disp("Using the Erlang-B traffic table (see
    Appendix A) for 99 channels with 2% blocking, we
    nd a traffic load of 87 Erlangs.");
25 TrafLoad1=87.004;
26 Carryload1=(1-Blocking)*TrafLoad1;
27 disp("Using the Erlang-B traffic table (see
    Appendix A) for 56 channels with 2% blocking, we
    nd a traffic load of 45.88 Erlangs.");
28 TrafLoad2=45.877;
29 Carryload2=(1-Blocking)*TrafLoad2;
30 disp("Using the Erlang-B traffic table (see
    Appendix A) for 33 channels with 2% blocking, we
    nd a traffic load of 24.6 Erlangs.");
31 TrafLoad3=24.629;
32 Carryload3=(1-Blocking)*TrafLoad3;
33 // To find cell capacity
34 Ncall1=Carryload1*3600/CallHT;//Calls per hour per
    cell
35 Ncall2=Carryload2*3600/CallHT;
36 Ncall3=Carryload3*3600/CallHT;
37 printf('\ncalls per hour per cell for N=4 are %d',
    round(Ncall1));
38 printf('\ncalls per hour per cell for N=7 are %d',
    round(Ncall2));
39 printf('\ncalls per hour per cell for N=12 are %d \
    n',Ncall3);
40 // To find S BY I
41 //  $N=(1/3)[6*(S/I)]^{(2/PPL)}$ 

```



```

42 S_I1=10*(PPL/2)*(log10(N1)-log10(1/3)-(2/PPL)*log10
    (6)); //Mean S/I (dB)
43
44 S_I2=10*(PPL/2)*(log10(N2)-log10(1/3)-(2/PPL)*log10
    (6));
45 S_I3=10*(PPL/2)*(log10(N3)-log10(1/3)-(2/PPL)*log10
    (6));
46
47 printf('\nMean S/I(dB) for N=4 is %.1f',S_I1);
48 printf('\nMean S/I(dB) for N=7 is %.1f',S_I2);
49 printf('\nMean S/I(dB) for N=12 is %.1f',S_I3);

```

Scilab code Exa 5.4 To calculate the number of calls per hour per cellsite

```

1 // Exa 5.4
2 // To find the number of calls per hour per cell
  site.
3
4 clc;
5 clear all;
6
7 spectrum=12.5*10^6; //in Hz
8 CHBW=200*10^3; //in Hz
9 N=4; //reuse factor
10 Blocking=0.02; // 2%
11 callHT=120; //average call holding time in sec
12 PPL=4; //propogation path loss coefficient
13 CntrlCH=3; //No of control channels
14 Ts=8; // No of voice channels per RF channel
15
16 //solution
17 No_ofVCH=((spectrum*Ts)/(CHBW*N))-CntrlCH;
18 printf('\n No of voice channels for N=4 are %d',
    No_ofVCH);
19 disp("");

```

```

20 disp("Using the Erlang-B traffic table for 122
    channels with 2% blocking, we find a traffic
    load of 110 Erlangs. ");
21 TrafLoad=110;
22 CarryLoad=(1-Blocking)*TrafLoad;
23 Ncall=CarryLoad*3600/callHT;
24 printf('\n Calls per hour per cell for N=4 are %d
    calls/hour/cell \n ',round(Ncall));
25 S_I=10*(PPL/2)*(log10(N)-log10(1/3)-(2/PPL)*log10(6)
    );
26 printf('\n Mean S/I(dB) for N=4 is %.1f dB \n ',S_I)
    ;

```

Scilab code Exa 5.5 To calculate calls per hour per cellsite and mean S by I and s

```

1 // Exa 5.5
2 // To Calculate:
3 // a) The calls per hour per cell site
4 // b) The mean S/I ratio
5 // c) The spectral efficiency in Erlang/km2/MHz
6 // for Reuse ratio =4,7,12 and for omnidirectional,
    120 degree and 60 degree antenna systems.
7
8 clc;
9 clear all;
10
11 VCH=395; //Total allocated voice channels
12 CHBW=30; // in kHz
13 Spectrum=12.5; // in MHz
14 CallHT=120; //Average call holding time in sec
15 Blocking=0.02; // 2%
16 PL=40; //slope of path loss in dBperdecade
17
18 //solution
19 disp("We consider only the first tier interferers

```

```

        and neglect the effects of cochannel interference
        from the second and other higher tiers.”);
20 //FOR 120degree sectorization
21 //N=4
22 VCH11=(VCH/(4*3));
23 OffLoad11=24.629; // Offered t r a f c load per
        sector from Erlang-B table(Appendix A)
24 Load_site11=3*OffLoad11;
25 CarLoad11=(1-Blocking)*Load_site11;
26 Calls_hr_site11=CarLoad11*3600/CallHT;
27 R11=sqrt(CarLoad11/0.52);
28 Seff11=CarLoad11/(2.6*Spectrum*R11^2);
29 S_I11=PL*log10(sqrt(3*4))-10*log10(2);
30 //N=7
31 VCH12=(VCH/(3*7));
32 OffLoad12=12.341; // Offered t r a f c load per
        sector from Erlang-B table(Appendix A)
33 Load_site12=3*OffLoad12;
34 CarLoad12=(1-Blocking)*Load_site12;
35 Calls_hr_site12=CarLoad12*3600/CallHT;
36 R12=sqrt(CarLoad12/0.52);
37 Seff12=CarLoad12/(2.6*Spectrum*R12^2);
38 S_I12=PL*log10(sqrt(3*7))-10*log10(2);
39 //N=12
40 VCH13=VCH/(3*12);
41 OffLoad13=5.842; // Offered t r a f c load per
        sector from Erlang-B table(Appendix A)
42 Load_site13=3*OffLoad13;
43 CarLoad13=(1-Blocking)*Load_site13;
44 Calls_hr_site13=CarLoad13*3600/CallHT;
45 R13=sqrt(CarLoad13/0.52);
46 Seff13=CarLoad13/(2.6*Spectrum*R13^2);
47 S_I13=PL*log10(sqrt(3*12))-10*log10(2);
48 //For omnidirectional
49 //N=4
50 VCH21=VCH/(4);
51 OffLoad21=87.004; // Offered t r a f c load per
        sector from Erlang-B table(Appendix A)

```

```

52 Load_site21=OffLoad21;
53 CarLoad21=(1-Blocking)*Load_site21;
54 Calls_hr_site21=CarLoad21*3600/CallHT;
55 R21=sqrt(CarLoad21/0.52);
56 Seff21=CarLoad21/(2.6*Spectrum*R21^2);
57 S_I21=PL*log10(sqrt(3*4))-10*log10(6);
58 //N=7
59 VCH22=VCH/(7);
60 OffLoad22=46.817; // Offered traffic load per
    sector from Erlang-B table(Appendix A)
61 Load_site22=OffLoad22;
62 CarLoad22=(1-Blocking)*Load_site22;
63 Calls_hr_site22=CarLoad22*3600/CallHT;
64 R22=sqrt(CarLoad22/0.52);
65 Seff22=CarLoad22/(2.6*Spectrum*R22^2);
66 S_I22=PL*log10(sqrt(3*7))-10*log10(6);
67 //N=12
68 VCH23=VCH/(12);
69 OffLoad23=24.629; // Offered traffic load per
    sector from Erlang-B table(Appendix A)
70 Load_site23=OffLoad23;
71 CarLoad23=(1-Blocking)*Load_site23;
72 Calls_hr_site23=CarLoad23*3600/CallHT;
73 R23=sqrt(CarLoad23/0.52);
74 Seff23=CarLoad23/(2.6*Spectrum*R23^2);
75 S_I23=PL*log10(sqrt(3*12))-10*log10(6);
76 // For 60degree Sectorization
77 //N=3
78 VCH31=VCH/(6*3);
79 OffLoad31=14.902; // Offered traffic load per
    sector from Erlang-B table(Appendix A)
80 Load_site31=6*OffLoad31;
81 CarLoad31=(1-Blocking)*Load_site31;
82 Calls_hr_site31=CarLoad31*3600/CallHT;
83 R31=sqrt(CarLoad31/0.52);
84 Seff31=CarLoad31/(2.6*Spectrum*R31^2);
85 S_I31=PL*log10(sqrt(3*3))-10*log10(1);
86 //N=4

```

```

87 VCH32=VCH/(6*4);
88 OffLoad32=10.656; // Offered traffic load per
    sector from Erlang-B table(Appendix A)
89 Load_site32=6*OffLoad32;
90 CarLoad32=(1-Blocking)*Load_site32;
91 Calls_hr_site32=CarLoad32*3600/CallHT;
92 R32=sqrt(CarLoad32/0.52);
93 Seff32=CarLoad32/(2.6*Spectrum*R32^2);
94 S_I32=PL*log10(sqrt(3*4))-10*log10(1);
95 //N=7
96 VCH33=VCH/(6*7);
97 OffLoad33=5.084; // Offered traffic load per
    sector from Erlang-B table(Appendix A)
98 Load_site33=6*OffLoad33;
99 CarLoad33=(1-Blocking)*Load_site33;
100 Calls_hr_site33=CarLoad33*3600/CallHT;
101 R33=sqrt(CarLoad33/0.52);
102 Seff33=CarLoad33/(2.6*Spectrum*R33^2);
103 S_I33=PL*log10(sqrt(3*7))-10*log10(1);
104 //N=12
105 VCH34=VCH/(6*12);
106 OffLoad34=2.227; // Offered traffic load per
    sector from Erlang-B table(Appendix A)
107 Load_site34=6*OffLoad34;
108 CarLoad34=(1-Blocking)*Load_site34;
109 Calls_hr_site34=CarLoad34*3600/CallHT;
110 R34=sqrt(CarLoad34/0.52);
111 Seff34=CarLoad34/(2.6*Spectrum*R34^2);
112 S_I34=PL*log10(sqrt(3*12))-10*log10(1);
113
114 printf('For Omnidirectional
    Calls_per_hour_per_cellsite      Mean S-I ratio
    SpectralEfficiency\n')
115 printf('For N=4
    %.1f      %d
    Calls_hr_site21,S_I21,Seff21);
116 printf('For N=7
    %.1f      %d
    Calls_hr_site21,S_I21,Seff21);

```

```

    Calls_hr_site22 ,S_I22 ,Seff22);
117 printf('For N=12                                %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site23 ,S_I23 ,Seff23);
118
119 printf('For 120deg sector
    Calls_per_hour_per_cellsite      Mean S_I ratio
    SpectralEfficiency\n')
120 printf('For N=4                                %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site11 ,S_I11 ,Seff11);
121 printf('For N=7                                %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site12 ,S_I12 ,Seff12);
122 printf('For N=12                               %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site13 ,S_I13 ,Seff13);
123
124 printf('For 60 deg Sector
    Calls_per_hour_per_cellsite      Mean S_I ratio
    SpectralEfficiency\n')
125 printf('For N=3                                %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site31 ,S_I31 ,Seff31);
126 printf('For N=4                                %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site32 ,S_I32 ,Seff32);
127 printf('For N=7                                %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site33 ,S_I33 ,Seff33);
128 printf('For N=12                               %d
                                                %.1 f      %.3 f\n',
    Calls_hr_site34 ,S_I34 ,Seff34);

```

Chapter 6

Multiple Access Techniques

Scilab code Exa 6.1 To calculate spectral efficiency of modulation

```
1 // Exa 6.1
2 // To calculate spectral efficiency .
3
4 clc;
5 clear all;
6
7 Area=8; //in km^2
8 Cover=4000; // in km^2
9 CallBH=1.2; //Avg calls during BH
10 HT=100; // Avg holding time in sec
11 Block=0.02; //Blocking=2%
12 N=4; //Frequency reuse factor
13 Spectrum=12.5; // in MHz
14 CHBW=200; // in kHz
15 User_CH=8; //No of users per RF channel
16
17 //solution
18 RFCH=Spectrum*1000/CHBW;
19 TCH=int(RFCH)*User_CH;
20 SigCH=3; //No of signalling channels per cell
21 TCH_cell=TCH/N-SigCH;
```

```

22 Cells=Cover/Area;
23 OffLoad=108.4; // in Erlangs
24 printf('Using Erlang-B Tables, Total traffic offered
        by %d channels at 0.02 blocking = %.1f Erlangs/
        cell \n ',TCH_cell,OffLoad*(1-Block));
25 CarLoad=OffLoad*(1-Block);
26 Calls_hr_cell=CarLoad*3600/HT;
27 MaxUser_hr_cell=Calls_hr_cell/CallBH;
28 Seff=CarLoad*Cells/(Spectrum*Cover);
29 printf('Spectral Efficiency is %.2f Erlangs/MHz/km
        ^2\n',Seff);

```

Scilab code Exa 6.2 To find multiple access spectral efficiency for FDMA

```

1 // Exa 6.2
2 // To calculate spectral efficiency of FDMA.
3
4 clc;
5 clear all;
6
7 TCH=395; // Traffic Channels
8 SysBW=12.5; //in MHz
9 CHspace=30; // in kHz
10
11 //solution
12 Eff=TCH*CHspace/(SysBW*1000);
13 printf('Multiple access spectral efficiency of FDMA
        System is %.3f\n ',Eff);

```

Scilab code Exa 6.3 To find multiple access spectral efficiency of the TDMA system

```

1 // Exa 6.3
2 // To calculate spectral efficiency of TDMA.

```



```

3
4 clc;
5 clear all;
6
7 Tf=40; //Frame duration in msec
8 Mt=6; // Frames per slot
9 Bu=30; //bandwidth(KHz) of an individual user during
    his or her time slot
10 Nu=395; // number of users sharing the same time
    slot in the system, but having access to
    different frequency sub-bands
11 Bw=12.5; // in MHz
12 DR=16.2; //Data rate in kbps
13 FDur=40; // Frame duration in msec
14 slots=6; //No of slots per time frame
15 IndiRate=16.2; //Individual data rate in kbps
16 Srate=13; //Speech rate in kbps
17
18 //solution
19 TimeSlot=(Srate/IndiRate)*(FDur/slots);
20 Seff=TimeSlot*slots*Bu*Nu/(FDur*Bw*1000);
21 printf('Multiple access spectral efficiency of TDMA
    is %.2f\n ',Seff);
22 printf('The overhead portion of the frame is %d
    percent \n ',round((1-Seff)*100));

```

Scilab code Exa 6.4 To calculate the capacity and spectral efficiency of TDMA system

```

1 // Exa 6.4
2 // To calculate capacity and spectral efficiency of
    a TDMA system.
3
4 clc;
5 clear all;
6

```

```

7 nb=0.9; //BW efficiency factor
8 u=2; // Bit Efficiency with QPSK
9 Vf=1; // Voice activity factor
10 BW=12.5; //in MHz
11 IR=16.2; // in kbps
12 N=19; //frequency reuse factor
13
14 //solution
15 Nu=nb*u*BW*1000/(Vf*IR*N); // number of channels (
    mobile users) per cell
16 Seff=int(Nu)*IR/(BW*1000);
17 printf('Capacity of system is %d mobile users per
    cell\n ',Nu);
18 printf('Spectral efficiency of TDMA system is %.3f
    bit/sec/Hz\n',Seff);

```

Scilab code Exa 6.5 To calculate the frame efficiency and the number of channels p

```

1 // Exa 6.6
2 // To calculate frame efficiency and the number of
    channels per frame.
3
4 clc;
5 clear all;
6
7 Nr=2; // number of reference bursts per frame
8 Nt=24; // number of traffic bursts (slots) per
    frame(120msec)
9 FL=120; //Frame length in msec
10 Br=148; // number of overhead bits per reference
    burst
11 Bp=34; // number of overhead bits per preamble per
    slot
12 Bg=8.25; //number of equivalent bits in each guard
    time interval

```

```

13 Tf=120; // frame duration in msec
14 Rrf=270.83333333; // bit rate of the RF channel in
    kbps
15 R=22.8; //bit rate of each channel in kbps
16
17 //solution
18 B0=Nr*(8*Br)+Nt*(8*Bp)+(Nt+Nr)*(8*Bg); //The number
    of overhead bits per frame
19 Bt=FL*10^-3*Rrf*10^3; //The total number of bits per
    frame
20 Eff=(1-B0/Bt)*100;
21 CH_Frame=(Eff/100)*Rrf/R; //No of channels/frame
22 printf('The frame efficiency is %.2f percent\n ',Eff
    );
23 printf('Number of channels/frame are %d\n',round(
    CH_Frame));

```

Scilab code Exa 6.6 To calculate capacity and spectral efficiency of the DSCDMA sy

```

1 // Exa 6.6
2 // To calculate capacity and spectral e f ciency of
    the DS-CDMA system.
3
4 clc;
5 clear all;
6
7 nb=0.9; //bandwidth e f ciency
8 nf=0.45; //frequency reuse e f ciency
9 Cd=0.8; //capacity degradation factor
10 Vf=0.4; //voice activity factor
11 Eb_I0=7; // desired energy-to-interference ratio in
    dB
12 L=1; // e f ciency of sector-antenna in cell
13 BW=12.5; //One way system BW in MHz
14 R=16.2; //Information rate in kbps

```

```

15
16 //solution
17 Eb_I=10^(Eb_I0*0.1); //To convert from dB to a normal
    value
18 Nu=(nf*nb*Cd*L/Vf)*(BW*1000/(Eb_I*R)); //Capacity of
    system
19 Seff=round(Nu)*R/(12.5*10^3);
20 printf('Capacity of system is %d mobile users per
    cell\n ',round(Nu));
21 printf('Spectral efficiency of TDMA system is %.3f
    bits/sec/Hz\n',Seff);
22
23 disp("In these calculations , an omnidirectional
    antenna is assumed. If a three sector antenna (i
    .e., G=3) is used at a cell site with lamda(
    efficiency of sector-antenna in a cell)= 2.6, the
    capacity will be increased to 325 mobile users
    per cell , and spectral e f ciency will be 0.421
    bits/sec/Hz.")

```

Scilab code Exa 6.7 Compare DS CDMA and TDMA omnidirectional cell

```

1 // Exa 6.7
2 // Using the data given in Exa 6.4 and 6.6, compare
    the capacity of the DS-CDMA and TDMA
    omnidirectional cell.
3
4 clc;
5 clear all;
6
7 //Given Data from Exa 6.4 and Exa 6.6
8 Cd=0.8; //capacity degradation factor
9 R=16.2; //Data rate in kbps
10 Eb_I0=7; //in dB
11 Eb_I=10^(Eb_I0*0.1); //To convert from dB to a normal

```

```

        value
12 Vf=0.4; //voice activity factor
13 u=2; // Bit Efficiency
14 IR=16.2; // in kbps
15 N=19; //frequency reuse factor
16 nf=0.45; //frequency reuse efficiency
17
18 //solution
19 Ncdma_by_Ntdma=Cd*N*nf*IR/(Eb_I*Vf*u*R);
20 printf('The ratio of capacity of DS-CDMA to TDMA is
        %.3f\n',Ncdma_by_Ntdma);

```

Scilab code Exa 6.8 To calculate minimum number of PN chips per frequency word

```

1 // Exa 6.8
2 // To calculate the minimum number of PN chips that
   are required for each frequency word.
3
4 clc;
5 clear all;
6
7 Bss=600; //Hopping bandwidth in MHz
8 stepsize=400; // in Hz
9
10 //solution
11 No_of_Tones=Bss*10^6/stepsize;
12 Min_chips_required=log2(No_of_Tones);
13 printf('Minimum number of chips required are %d
        chips \n ',Min_chips_required);

```

Scilab code Exa 6.9 To find normalized throughput for different CSMA protocols

```

1 // Exa 6.9

```

```

2 // To calculate the normalized throughput with:
3 //(a) an unslotted nonpersistent ,
4 //(b) a slotted persistent , and
5 //(c) a slotted 1-persistent CSMA protocol.
6
7 clc;
8 clear all;
9
10 e=2.71828; //Euler's number
11 Tprop=0.4; //Max propagation delay in sec
12 R=10; //data rate in Mbps
13 PackLen=400; //packet length in bits
14
15 //solution
16 Tp=PackLen/R; //packet transmission time in microsec
17 a=Tprop/Tp;
18 G=Tp*10^-6*R*10^6/PackLen; //normalized offered
   t r a f f c   l o a d
19 //Slotted nonpersistent
20 S0=a*G*e^(-a*G)/(1-e^(-a*G)+a); //normalized
   t h r o u g h p u t
21 //Unslotted nonpersistent
22 S1=G*e^(-a*G)/(1+(2*a)+e^(-a*G)); //normalized
   t h r o u g h p u t
23 //Slotted 1-persistent
24 S2=G*e^(-G*(1+a))*(1+a-e^(-a*G))/((1+a)*(1-e^(-a*G))
   +a*e^(-G*(1+a))); //normalized throughput
25 printf('The Normalized throughput with an unslotted
   non persistent , a slotted persistent and a
   slotted 1-persistent CSMA protocol are \n %.3f,%
   .3f and %.3f respectively \n',S0,S1,S2);

```

Scilab code Exa 6.10 To find data link protocol efficiency with different protocol

```

1 // Exa 6.10

```

```

2 // To calculate the data link protocol efficiency
  with
3 //(1) Stop and Wait protocol full duplex,
4 //(2) SRP with window size W=8, and
5 //(3) Go-Back-N protocol with window size W=8.
6
7 clc;
8 clear all;
9
10 Tprop=4; //maximum propagation delay in sec
11 R=10; // data rate in Mbps
12 PackLen=400; //data packet length in bits
13 ACK=20; //length of ACK packet in bits
14 Tproc=1; //processing time(sec)
15 p=0.01; //probability that a data packet or its ACK
    can be corrupted during transmission
16
17 //solution
18 Tp=PackLen/R; //packet transmission time in microsec
19 Ta=ACK/R; // transmission time for an ACK in
    microsec
20 T=Tp+2*Tprop+2*Tproc+Ta; // total time for
    transmission time
21 // Stop and wait ARQ
22 Eff0=(1-p)*Tp/((1-p)*T+p*Tp);
23 //SRP with window size W=8
24 W=8;
25 Eff1=(2+p*(W-1))/(2+p*(3*W-1));
26 //Go-Back-N protocol with window size W=8
27 Eff2=1/(1+W*(p/(1-p)));
28 printf('The data link protocol efficiency with Stop
    and Wait protocol, SRP and GBN are \n %.3f, %.3f
    abd %.3f respectively\n',Eff0,Eff1,Eff2);

```

Chapter 8

Speech Coding and Channel Coding

Scilab code Exa 8.1 To calculate the gain in the link budget in dB

```
1 // Exa 8.1
2 // To calculate coverage gain in dB.
3
4 clc;
5 clear all;
6
7 Pdiff=-3; //in dB
8 AMR1=12.2; //in kbps
9 AMR2=7.95; //in kbps
10 AMR3=4.75; //in kbps
11
12 //solution
13 //CG(dB)=10log {(DPDCH(kbps)+DPCCH)/(DPDCH(AMR bit
    rate (kbps))+ DPCCH)}
14 CG1=10*log10((AMR1+AMR1*10^(Pdiff/10))/(AMR2+AMR1
    *10^(Pdiff/10)));
15 CG2=10*log10((AMR1+AMR1*10^(Pdiff/10))/(AMR3+AMR1
    *10^(Pdiff/10)));
16 printf('By reducing the AMR bit rate from 12.2 to
```



```

    7.95 kbps coverage gain becomes %.2f dB \n ',CG1)
;
17 printf('By reducing the AMR bit rate from 7.95 to
    4.75 kbps coverage gain becomes %.2f dB \n ',CG2)
;

```

Scilab code Exa 8.2 To calculate output of convolution encoder

```

1 // Exa 8.2
2 // To calculate the output of the encoder.
3
4 clc;
5 clear all;
6
7 K=4; //constraint length
8 r=1/2; //code rate(n/k)
9 x=poly(0,"x");//Defining x as a ploynomial variable
10 G1=1+x^2+x^3;
11 G2=1+x+x^2+x^3;
12 in=[1 0 1 1 1];//input(first bit first)
13
14 //solution
15 //with reference to Fig 8.9 on page no 239
16 g1=[1 0 1 1]; //converting from G1 polynomial to bit
    form
17 g2=[1 1 1 1];/////converting from G2 polynomial to
    bit form
18 x1=round(convol(g1,in));
19 x2=round(convol(g2,in));
20 V1=modulo(x1,2);
21 V2=modulo(x2,2);
22 disp("Multiplexing the V1 and V2 to get required
    output sequence as ");
23 a=5;
24 for i= 1:5

```

```

25     printf( '%d%d',V2(a),V1(a));
26     a=a-1;
27
28 end

```

Scilab code Exa 8.3 To demonstrate operations of converting burst errors into bit

```

1 // Exa 8.3
2 // To demonstrate 4X4 Bit interleaving/de-interleaving
3
4 clc;
5 clear all;
6
7 BitStream= [0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1]; //Last
   bit to first bit
8
9 //solution
10 disp("Interleaving is performed by storing the data
   in a table containing rows and columns at the
   transmitter. The data is written in rows and
   transmitted in a vertical direction (according to
   columns). At the receiver, the data is written
   and read in the opposite manner. ")
11
12 // Interleaver
13         Input1=[1 0 0 0           //Writing data
   row wise
14                 1 0 0 0
15                 1 1 1 0
16                 0 0 0 0];
17 disp("GIven Bit stream is")
18 disp(BitStream);
19 disp("Input to interleaver is")
20 disp(Input1);

```

```

21
22 Output1=[0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1];    //
    Reading data column wise
23 disp("Output of interleaver is");
24 disp(Output1);
25 //De-interleaver
26         Input2=[1 1 1 0 //Writing o/p data
    row wise
27                 0 0 1 0
28                 0 0 1 0
29                 0 0 0 0];
30 // Let From 6th to 9th bits have Burst Error
31 disp("Input to de-interleaver is");
32 disp(Input2);
33 //Output of deinterleaver
34
35 Output2= [0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1];
36 disp("Output of de-interleaver is")
37 disp(Output2);
38 disp( "Bits with Burst error were from 6th to 9th.
    But in output of de-interleaver , they relocated
    to positions 3rd , 6th , 10th and 14th.");

```

Chapter 9

Modulation Schemes

Scilab code Exa 9.1 To find Eb by No in dB

```
1 // Exa 9.1
2 // To calculate Eb/No in dB for BPSK and Coherent
  FSK.
3
4 clc;
5 clear all;
6
7 Pe=10^-6; //Probability of error
8 e=2.71828; //Euler's Number
9
10 //solution
11 // For BPSK
12 //Pe(=10^-6)=e^(-x)/(2*sqrt(%pi*x)); where x=Eb/No
13
14 deff('y=f(x)', 'y=2.71828^(-x)/(2*sqrt(%pi*x))-10^-6'
  );
15 [x,v,info]=fsolve(0.1,f);
16
17 printf('Eb/No For BPSK is %.2f dB\n ',10*log10(x));
18 printf('FSK requires 3 dB more in terms of Eb/No to
  give the same Pe as BPSK so it comes out to be %
```

```
.2 f dB',10*log10(x)+3);
```

Scilab code Exa 9.2 To calculate amplitude A of the carrier signal

```
1 // Exa 9.2
2 // To calculate amplitude A of a carrier signal.
3
4 clc;
5 clear all;
6
7 Pe=10^-6; //Probability of error
8 No=10^-10; // PSD in W/Hz
9 R=100*10^3; //data rate in bps
10
11 //solution
12 disp("From Example 9.1, Eb/N0= 10.54dB (11.32) for
      Pe=10^-6 ");
13 //Therefore
14 Eb_No=11.32; //From Exa. 9.1
15 // Eb/No = A^2/(2*No*R);
16 A=sqrt(2*No*(Eb_No)*R);
17 printf(' Amplitude of a carrier signal is %.3f mV',A
      *1000);
```

Scilab code Exa 9.3 To calculate final phase for given bitstream for given modulation

```
1 // Exa 9.3
2 // To calculate final phase for the pi/4-DQPSK
  modulation method.
3
4 clc;
5 clear all;
6
```

```

7 B=['00 ','10 ','01 ','11 ','01 ','00 ','11 ','10 ','10 ','01 '
    , '01 ','00 ']; //Given Bit stream
8
9 //solution
10 disp("Phase transition table for pi/4-DQPSK
    Modulation is given as ")
11 disp(" By Referring Table 9.1 on page No 266 i.e");
12 disp("Symbol      Phase transition")
13 disp("00          =>    45  ");
14 disp("01          =>   135  ");
15 disp("10          =>   -45  ");
16 disp("11          =>  -135  ");
17 disp("");
18 disp("sym          Dell phi(k)   Phi(k)")
19 //BitStream='001001110100111010010100';
20
21 phase=0; //Taking initial phase as zero
22 for i=1:12
23
24
25     if(B(i)=='00 ')
26         phase=phase+45;
27         printf(' %s          45          %d \n ',B(i),
                phase);
28     end
29
30     if(B(i)=='01 ')
31         phase=phase+135;
32         printf(' %s          135          %d \n ',B(i),
                phase);
33     end
34     if(B(i)=='10 ')
35         phase=phase-45;
36         printf(' %s          -45          %d \n ',B(i),
                phase);
37     end
38     if(B(i)=='11 ')
39         phase=phase-135;

```

```

40         printf(' %s          -135          %d \n',B(i),
                phase);
41
42 end
43 end
44 disp("");
45 printf('final phase for the pi/4-DQPSK modulation
        method for given bitstream is %d degree\n',phase)
        ;

```

Scilab code Exa 9.4 To calculate frequency shift along with transmitted frequencies

```

1 // Exa 9.4
2 // To calculate
3 //(a) the frequency shift between binary 1 and
    binary 0,
4 //(b) the transmitted frequencies if the carrier
    frequency is 900 MHz, and
5 //(c) the bandwidth efficiency in bps/Hz.
6
7 clc;
8 clear all;
9
10 CHBW=200; //Channel BW in KHz
11 R=270.83; //Data rate in kbps
12 Fc=900; //carrier frequency in MHz
13
14 //solution
15 FreqShift=0.5*R;
16 //Transmitted Frequencies
17 Fh=Fc*1000+0.25*R; //Max
18 Fl=Fc*1000-0.25*R; //Min
19 BWEff=R/CHBW;
20 printf('The frequency shift between binary 1 and
        binary 0 is %.3f kHz\n ',FreqShift);

```

```

21 printf('Maximum and Minimum value of transmitted
    frequencies are %.4f mHz and %.4f mHz
    respectively\n ',Fh/1000,F1/1000);
22 printf('Bandwidth efficiency is %.2f bps/Hz',BWEff);

```

Scilab code Exa 9.5 To calculate bit error probability for GMSK

```

1 // Exa 9.5
2 // To calculate –
3 // a) 3–dB bandwidth for a Gaussian low–pass filter
4 // b) 99.99% power bandwidth in the RF channel, and
5 // c) bit error probability for GMSK.
6
7 clc;
8 clear all;
9
10 R=270; //data rate in kbps
11 Eb_No=6; // in dB
12 GMSK=0.3; //Gaussian minimum shift keying
13
14 //solution
15 Tb=1/R *10^3; //in microsec
16 B=GMSK/Tb;
17 printf('3–dB BW for a gaussian low pass filter is %
    .3f kHz\n',B*1000);
18 disp("The 3–dB bandwidth is 81.08 kHz. to determine
    the 99.99% power bandwidth, we use Table 9.3 to
    find that 1.41*Rb is the required value and
    degradation factor(beta)= 0.89");
19 PowerBW=1.41*R;
20 DegradFac=0.89;
21 Pe=erfc(sqrt(2*DegradFac*10^(0.1*Eb_No)));
22 printf(' Power bandwidth in the RF channel is %.1f
    kHz\n ',PowerBW);

```



```
23 printf('Bit error probability for GMSK is %f *  
    10^-5\n',Pe*10^5);
```

Scilab code Exa 9.6 To calculate bit rate of modulator

```
1 // Exa 9.6  
2 // To calculate bit rate.  
3  
4 clc;  
5 clear all;  
6  
7 Rs=19200; //symbols per second  
8 states=64;  
9  
10 //solution  
11 Bits_symbol=log2(states);  
12 BitRate=Bits_symbol*Rs;  
13 printf('Bit Rate of the modulator is %.1f kbps',  
    BitRate/1000 );
```

Scilab code Exa 9.7 To determine the modulation scheme to be used along with requi

```
1 // Exa 9.7  
2 // To determine modulation scheme to be used and Eb/  
    No.  
3  
4 clc;  
5 clear all;  
6  
7 Rb=144; //data rate in kbps  
8 BW=36; //in MHz  
9 Pb=3*10^-5; //probability of bit error  
10
```

```

11 //solution
12 Seff=Rb/BW; //spectral efficiency in bps/Hz
13
14 M=2^(Rb/BW); //since the channel is band limited
15 disp("16-QAM (refer Equation 9.66) should be used
      as it is more efficient than 16-PSK (refer
      Equation 9.50)");
16 disp("");
17
18 //since  $Q[\sqrt{2*Eb\_No}]=(1/2)*erfc[\sqrt{Eb\_No}]$ 
      // refer page no 257 equ 9.35
19 deff('y=f(x)', 'y=(3/8)*erfc(sqrt((2/5)*x))-Pb');
      //from eqn 9.66 and 9.35
20
21 [x,v,info]=fsolve(0.1,f); //x=Eb_No
22
23 printf('For a rectangular constellation (refer
      Figure 9.12), with a Gaussian channel and matched
      lter reception, the calculated Eb/No value is
      %.1f dB\n ',10*log10(x));

```

Scilab code Exa 9.8 Compare the performance of 16PSK with 16QAM

```

1 // Exa 9.8
2 // To compare the performance of 16-PSK with 16-QAM.
3
4 clc;
5 clear all;
6
7 Pb=10^-8; //BER probability
8
9 //solution
10 disp(" For 16-PSK");
11 // Pb=0.5*Q(0.552*sqrt(Eb_No));
12 //since  $Q[\sqrt{2*Eb\_No}]=(1/2)*erfc[\sqrt{Eb\_No}]$ 

```

```

// refer page no 257 equ 9.35
13 deff('y=f(x)', 'y=0.25*erfc(sqrt(0.5*0.552^2*x))-Pb')
    ;
14 [x,v,info]=fsolve(0.1,f); //x=Eb_No
15
16 printf('Using equation 9.50 we get Eb/No as %d dB (
    approx) \n ',round(10*log10(x)));
17 disp('For 16-QAM');
18 //Pb=0.75*Q(sqrt(0.8*Eb_No));
19 deff('y=f1(x1)', 'y=(3/8)*erfc(sqrt(0.4*x1))-Pb'); //
    x=Eb_No
20 //since Q(sqrt(2*Eb_No))=(1/2)*erfc(sqrt(Eb_No)) //
    refer page no 257 equ 9.35
21 [x1,v1,info1]=fsolve(0.1,f1); //x=Eb_No
22 printf('Using equation 9.66 we get Eb/No as %d dB (
    approx)\n ',round(10*log10(x1)));
23 disp('');
24 printf('Thus 16-QAM has an advantage of about %d dB
    compared to 16-PSK \n ',10*log10(x)-10*log10(x1))
    ;

```

Scilab code Exa 9.9 To find total bandwidth and its efficiency along with require

```

1 // Exa 9.9
2 // To calculate -
3 // a) The total bandwidth required ,
4 // b) The bandwidth efficiency ,
5 // c) The Eb/No required , and
6 // d) No of carried bits per symbol.
7
8 clc;
9 clear all;
10
11 M=8; //number of different signal elements
12 Fc=250; //carrier frequency in kHz

```

```

13 DelF=25; //kHz
14 Pe=10^-6; //probability of error
15
16 //solution
17 TotalBW=2*M*DelF;
18 nb=2*log2(M)/(M+3);
19 //Pe=7*Q(z) and z=approx(5.08)
20 z=5.08;
21 Eb_No=(z)^2/log2(M);
22 bits_sym=log2(M);
23 printf('Total bandwidth required is %d kHz \n ',
        TotalBW);
24 printf('The bandwidth efficiency is %.4f \n ',nb);
25 printf('The required Eb/No is %.3f dB \n ',10*log10(
        Eb_No));
26 printf('Carried bits per symbol are %d \n ',bits_sym
        );

```

Chapter 10

Antennas Diversity and Link Analysis

Scilab code Exa 10.1 To find received signal power and SNR of antenna

```
1 // Exa 10.1
2 // To calculate the received signal power at the
  receiver antenna and the SNR of the received
  signal.
3
4 clc;
5 clear all;
6
7 D=10000; //in metres
8 TxEIRP=30; //Effective Isotropic Radiated Power(
  EIRP)dBW
9 lamda=0.2; //in metres
10 Pt=10; //Transmitted power in dBW
11 Gt=20; //transmitter gain in dBi
12 Gr=3; //receiver gain in dBi
13 Lo=6; //total system losses in dB
14 Nf=5; //noise figure in dB
15 BW=1.25; //mHz
16 k=1.38*10^-23; //Boltzmann constant
```

```

17 T=290; //temperature in degree kelvin
18
19 //solution
20 Lp=20*log10(lamda/(4*pi*D)); //free space loss
21 Pr=Lp+Pt+Gt+Gr-Lo; // received power in dBW
22
23 No=10*log10(k*T); //Noise density in dBW
24 NO=No+30; //factor of '30' to convert from dBW to
    dBm
25 Pn=Nf+10*log10(BW*10^6)+NO; // noise signal power in
    dBm
26 SNR=(Pr+30)-Pn;
27 printf('The received signal power is %d dBm\n ',
    round(Pr+30)); //factor of '30' to convert from
    dBW to dBm
28 printf('SNR is %d dB\n',SNR);

```

Scilab code Exa 10.2 To find received signal power and SNR of antenna

```

1 // Exa 10.2
2 // To calculate the received signal power at the
    receiver antenna and the SNR of the received
    signal.
3
4 clc;
5 clear all;
6
7 //As we have to use data from Eg 10.1,
8 D=10000; // in metres
9 TxEIRP=30; //Effective Isotropic Radiated Power(
    EIRP)dBW
10 lamda=0.2; //in metres
11 Pt=10; //trasmitted power in dBW
12 Gt=20; //transmitter gain in dBi
13 Gr=3; //receiver gain in dBi

```

```

14 Lo=6; //total system losses in dB
15 Nf=5; //noise figure in dB
16 BW=1.25; //mHz
17 k=1.38*10^-23; //Boltzmann constant
18 T=290; //temperature in degree kelvin
19 //additional data given in this eg
20 hr=40; //height of receiver in metre
21 ht=2; //trasmittter antenna height in metres
22
23 //solution
24 Lp=20*log10(hr*ht/D^2);
25 Pr=Lp+Pt+Gt+Gr-Lo; // received power in dBW
26 No=10*log10(k*T); //Noise density in dBW
27 NO=No+30; //factor of '30' to convert from dBW to
    dBm
28 Pn=Nf+10*log10(BW*10^6)+NO; // noise signal power in
    dBm
29 SNR=(Pr+30)-Pn;
30 printf('The received signal power is %d dBm\n ',
    round(Pr+30)); //factor of '30' to convert from
    dBW to dBm
31 printf('SNR is %d dB\n',SNR);

```

Scilab code Exa 10.3 To calculate gain of antenna

```

1 // Exa 10.3
2 // To calculate gain of antenna.
3
4 clc;
5 clear all;
6
7 Pin=12; //Input power in watts
8 Ploss=3; //resistive losses in Watts
9 D=5; //Directivity
10

```

```

11 //solution
12 Eff=(Pin-Ploss)/Pin;
13 G=Eff*D;
14 printf('Gain of the antenna is %.2f dB = %.2f \n'
        ,10*log10(G),G);

```

Scilab code Exa 10.4 To determine 3dB beam width of an antenna

```

1 // Exa 10.4
2 // To calculate the 3-dB beam width of a linear
  element antenna.
3
4 clc;
5 clear all;
6
7 G=12; //Gain of antenna in dBi
8
9 //solution
10 Theta=101.5/10^(G/10);
11 printf('The 3-dB beam width of a linear element
        antenna is %.1f degrees ',Theta);

```

Scilab code Exa 10.5 To calculate various parameters of helical antenna

```

1 // Exa 10.5
2 // To calculate the optimum diameter (DH), spacing (
  S) for the antenna and total length of the
  antenna,
3 // To calculate the antenna gain,
4 // To calculate the beam width of the antenna.
5
6 clc;
7 clear all;

```



```

8
9 N=12; //number of turns
10 fr=1.8; //frequency in GHz
11
12 //solution
13 lamda=3*10^8/(fr*10^9);
14 DH=lamda/%pi; // diameter of helix in milli-meters
15 S=lamda/4; //turn spacing in millimetres
16 L=N*S;
17 G=15*N*S*(DH*%pi)^2/lamda^3;
18 Theta=52*lamda/(%pi*DH)*sqrt(lamda/(N*S));
19 printf('The optimum diameter is %d mm\n ',DH*1000);
20 printf('Spacing is %.1f mm\n ',S*1000);
21 printf('Total Length of antenna is %d mm\n ',L*1000)
    ;
22 printf('The antenna gain is %.1f dBi\n ',10*log10(G)
    );
23 printf('The BeamWidth of antenna is %d degrees \n ',
    Theta);

```

Scilab code Exa 10.6 To find probability that SNR will drop below 10dB and compare

```

1 // Exa 10.6
2 // To find probability that SNR will fall below 10dB
   (= 10).
3
4 clc;
5 clear all;
6
7 E0=1000; //average SNR
8 Eg=10; //threshold value for SNR
9 M=3; //3-Branch Combiner
10 e=2.71828; //Euler's number
11
12 //solution

```

```

13 x=Eg/E0;
14 P3=(1-e^(-x))^M; //Considering 3-branch selection
    combiner
15 printf('By considering 3-branch selection combiner
    technique, probability comes to be %d * 10^-6\n ',
    ,round(P3*10^6));
16
17 disp("When diversity is not used, M =1");
18 P1=(1-e^(-x)); //M=1;
19 printf(' BY not considering diversity technique,
    probability comes to be %d * 10^-2 \n',round(P1
    *10^2));

```

Scilab code Exa 10.7 To calculate minimum delay difference to successfully resolve

```

1 // Exa 10.7
2 // To determine the minimum delay difference to
    successfully resolve the multipath components and
    operate the Rake receiver
3
4 clc;
5 clear all;
6
7 SR=3.84; //spreading rate in Mcps
8
9 //solution
10 disp("In order to resolve multipath components, the
    chip duration should be equal to or greater
    than T(tau), where T is defined as ratio of
    delay distance to speed to electromagnetic wave")
    ;
11 ChipDur=1/(SR*10^6);
12 Speed=3*10^8;
13 Dd=ChipDur*Speed;
14 disp("");

```

```
15 printf('Minimum delay distance to successfully  
    resolve the multipath components and operate the  
    Rake receiver is %d m \n',Dd);
```

Chapter 11

Spread Spectrum and CDMA Systems

Scilab code Exa 11.1 To calculate processing gain and improvement in information rate

```
1 // Exa 11.1
2 // To calculate the processing gain and improvement
  in information rate.
3
4 clc;
5 clear all;
6
7 CR1=1.2288; //Mcps(Clock rate 1)
8 CR2=5; //Mcps(Clock rate 2)
9 R1=9.6; //Information rate in Kbps for CR1
10 PG2=256; //Processing Gain for CR2
11
12 //solution
13 PG1=10*log10(CR1*10^3/9.6); //Processing Gain for CR1
14 R2=CR2*10^3/PG2; //information rate in Kbps for CR2
15 printf('The processing gain for clock rate 1.2288
  Mcps is %d dB\n ',PG1);
16 printf('Improvement in information rate is %.2f Kbps
  \n ',R2-R1);
```

Scilab code Exa 11.2 Show that the transmitted signals to mobiles 1 2 and 3 are re

```
1 // Exa 11.2
2 // To show that the transmitted signals to mobiles
   1, 2, and 3 are recovered at the mobile receivers
   by despreding the resultant signal z(t).
3
4 clc;
5 clear all;
6
7 //solution
8 disp("From figure 11.4, we note that transmitted
   data for mobile1 as [0 1 1 0 0 ], for mobile2 as
   [0 0 1 0 0 ] and for mobile3 as [0 1 0 0 1] ");
9 disp("From figure 11.5 we get resultant demodulated
   signal at a mobile ");
10 Rx={[1 1 1 1 -3 1];[1 -3 1 1 1 1 ];[1 -3 1 1 1
   -3];[1 -3 1 1 1 1];[-1 3 3 -1 3 -1]}; //Resultant
   demodulated signal at mobile
11 disp(Rx);
12 //from Figure 11.4
13 c1={[-1 -1 -1 -1 1 1];[1 -1 1 1 -1 -1];[1 -1 1 -1 -1
   -1];[-1 1 1 1 -1 1];[1 -1 -1 1 -1 1]};
14 c2={[1 1 -1 1 1 -1];[-1 1 -1 1 -1 -1];[-1 -1 1 1 1
   -1];[1 1 -1 -1 1 -1];[1 -1 -1 -1 -1 -1]};
15 c3={[-1 -1 1 -1 1 -1];[-1 -1 -1 1 1 1];[-1 1 1 -1 -1
   1];[-1 1 -1 -1 -1 -1];[1 1 1 -1 1 1]};
16 //t={ [1 2 3 4 5 6];[7 8 9 10 11 12];[13 14 15 16 17
   18];[19 20 21 22 23 24];[25 26 27 28 29 30]};
17 //for Mobile 1
18 for i= 1:5
19
20 Demod1(i)=c1(i,1)*Rx(i,1)+c1(i,2)*Rx(i,2)+c1(i,3)*Rx
   (i,3)+c1(i,4)*Rx(i,4)+c1(i,5)*Rx(i,5)+c1(i,6)*Rx(
```

```

        i,6);
21 if(Demod1(i)<0)
22     B1(i)=1;
23 else
24     B1(i)=0;
25     end
26 end
27 //for mobile 2
28 for i= 1:5
29
30 Demod2(i)=c2(i,1)*Rx(i,1)+c2(i,2)*Rx(i,2)+c2(i,3)*Rx
        (i,3)+c2(i,4)*Rx(i,4)+c2(i,5)*Rx(i,5)+c2(i,6)*Rx(
        i,6);
31 if(Demod2(i)<0)
32     B2(i)=1;
33 else
34     B2(i)=0;
35     end
36 end
37 //for mobile 3
38 for i= 1:5
39
40 Demod3(i)=c3(i,1)*Rx(i,1)+c3(i,2)*Rx(i,2)+c3(i,3)*Rx
        (i,3)+c3(i,4)*Rx(i,4)+c3(i,5)*Rx(i,5)+c3(i,6)*Rx(
        i,6);
41 if(Demod3(i)<0)
42     B3(i)=1;
43 else
44     B3(i)=0;
45     end
46 end
47 disp("Value of integration at end of bit period for
        mobile1");
48 disp(Demod1');
49 disp("Value of integration at end of bit period for
        mobile2");
50 disp(Demod2');
51 disp("Value of integration at end of bit period for

```

```

        mobile3");
52 disp(Demod3');
53 disp("The recovered signal at mobile 1 is ");
54 disp(B1');
55 disp("The recovered signal at mobile 2 is ");
56 disp(B2');
57 disp("The recovered signal at mobile 3 is ");
58 disp(B3');
59 disp("In all cases , Recovered signal is negated
        value of transmitted signal")

```

Scilab code Exa 11.3 Minimum number of PN chips required for each frequency symbol

```

1 // Exa 11.3
2 // To find the minimum number of PN chips.
3
4 clc;
5 clear all;
6
7 BW=100; //in MHz
8 Fspac=10; //frequency spacing in kHz
9
10 //solution
11 FreqTones=BW*103/Fspac;
12 Chips=log2(FreqTones);
13 printf('Minimum number of chips required are %d
        chips \n ',Chips);

```

Scilab code Exa 11.4 To calculate data symbol transmitted per hop and number of no

```

1 // Exa 11.4
2 // To calculate –
3 //(a) data symbol transmitted per hop, and

```

```

4 // (b) the number of nonoverlapping hop frequencies.
5
6 clc;
7 clear all;
8
9 R=120; //transmission rate in kbps
10 Hop=2000; //per second
11 Spectrum=10; //in MHz
12
13 //solution
14 //For 32-FSK
15 Bits_sym=log2(32);
16 SR=R/Bits_sym;
17 printf('Bits per symbol are %d \n ',Bits_sym);
18 printf('Hops per second are 2000 and Symbol rate is
    %d kbps\n ',SR);
19 disp("Since the symbol rate is higher than the hop
    rate, the system is a slow FHSS system.");
20 Sym_hop=SR*10^3/Hop;
21 Min_BW=Sym_hop*SR;
22 Nonoverlap_hop=Spectrum*10^3/Min_BW;
23 disp("");
24 printf(' Symbols transmitted per hop are %d \n ',
    Sym_hop);
25 printf('Number of non-Overlapping hop frequencies
    are %d \n ',round(Nonoverlap_hop));

```

Scilab code Exa 11.5 To consider various parameters of FHSS systems

```

1 // Exa 11.5
2 // To Calculate:
3 //(a) minimum separation between frequency tones,
4 //(b) number of frequency tones produced by a
    frequency synthesizer,
5 //(c) processing gain, and

```



```

6 // (d) hopping bandwidth.
7
8 clc;
9 clear all;
10
11 R=200; //input data rate in bps
12 Fhop=200; //per second
13 k=1; //Multiplication_Factor
14
15 //solution
16 // We have 32-FSK modulation scheme
17 Bits_sym=log2(32);
18 Rs=Fhop/Bits_sym;
19 printf('There are 200 hops per second and Symbol
    rate is %d symbols per sec \n',Rs);
20 disp("The hop rate is higher than symbol rate, the
    system is a fast FHSS system.");
21 SDur=1/Rs;
22 L=Fhop/Rs;
23 CDur=SDur/L;
24 Separation=1/CDur;
25 M=2^Bits_sym;
26 Hop_BW=k*M*Fhop*L;
27 Gp=M*k*L;
28 disp("");
29 printf(' Minimum separation between frequency tones
    should be %d Hz\n',Separation);
30 printf(' Number of different frequency tones
    produced by a frequency synthesizer are %d\n',M);
31 printf(' Processing Gain is %d\n ',Gp);
32 printf('Hopping bandwidth is %d kHz\n',Hop_BW/1000);

```

Scilab code Exa 11.6 To show how mobile1 will detect its information using a two p

```

1 // Exa 11.6

```

```

2 // To show how the signal is recovered at mobile by
   using two-rake receivers.
3
4 clc;
5 clear all;
6
7 //solution
8 disp("As we are given that actual bit values for
   mobile are [1 0 0 1 1] ");
9 M1=[1 0 0 1 1];
10
11 Rx1={ [1 1 1 1 -3 1]; [1 -3 1 1 1 1 ]; [1 -3 1 1 1
   -3]; [1 -3 1 1 1 1 ]; [-1 3 3 -1 3 -1]; [1 -1 -1 0 0
   0]}; //Resultant demodulated signal at mobile(Z(t)
   ) from path1
12
13 Rx2={ [-1 -1 1 1 1 1 ]; [-3 1 1 -3 1 1 ]; [1 1 1 -3 1 1
   ]; [1 -3 1 -3 1 1 ]; [1 1 -1 3 3 -1 ]; [3 1 -1 0 0
   0]}; //Resultant demodulated signal at mobile(Z(t
   -2Tc) ) from path2
14
15 Rx=Rx1+Rx2; //since ,Z(t)=z(t)+Z(t-2Tc)
16
17 //from Figure 11.13 (d) & Figure 11.14
18 c1={ [-1 -1 -1 -1 1 1 ]; [1 -1 1 1 -1 -1 ]; [1 -1 1 -1 -1
   -1]; [-1 1 1 1 -1 1 ]; [1 -1 -1 1 -1 1 ]};
19 c2={ [-1 1 -1 -1 -1 -1 ]; [1 1 1 -1 1 1 ]; [-1 -1 1 -1
   1 -1 ]; [-1 -1 -1 1 1 1 ]; [-1 1 1 -1 -1 1 ]; [-1 1
   0 0 0 0]};
20
21 //case -1:Z(t)*C1(t);
22 for i= 1:5
23
24 Demod_1(i)=c1(i,1)*Rx(i,1)+c1(i,2)*Rx(i,2)+c1(i,3)*
   Rx(i,3)+c1(i,4)*Rx(i,4)+c1(i,5)*Rx(i,5)+c1(i,6)*
   Rx(i,6);
25 if(Demod_1(i)<0)
26     B1(i)=1;

```

```

27 else
28     B1(i)=0;
29     end
30 end
31
32 //case -2:Z(t)*C1(t-2Tc);
33 for j=1:5
34
35 Demod_2(j)=c2(j,3)*Rx(j,3)+c2(j,4)*Rx(j,4)+c2(j,5)*
    Rx(j,5)+c2(j,6)*Rx(j,6)+c2(j+1,1)*Rx(j+1,1)+c2(j
    +1,2)*Rx(j+1,2);
36 if(Demod_2(j)<0)
37     B2(j)=1;
38 else
39     B2(j)=0;
40     end
41 end
42     disp("case -1:z(t)*c1(t)");
43 disp("Value of integration at end of bit period for
    mobile(case-1)");
44 disp(Demod_1');
45 disp("The recovered signal at mobile(case-1) is ");
46 disp(B1');
47 disp("Actual bit values are");
48 disp(M1);
49 disp("Recovered and actual values are not matching")
    ;
50 disp("case -2:z(t)*c1(t-2Tc)");
51 disp("Value of integration at end of bit period for
    mobile(case-2)");
52 disp(Demod_2');
53 disp("The recovered signal at mobile(case-2) is ");
54 disp(B2');
55 disp("Actual bit values are");
56 disp(M1);
57 disp("Recovered and actual values are not matching")
    ;
58 //case3-Sum of path1 and path2

```

```

59 disp("case -3:Sum of path1 & path2 integrator");
60 disp("Sum of integrator outputs(rake receiver output
    )");
61 Demod_3=Demod_1+Demod_2;
62 disp(Demod_3');
63 for k=1:5
64
65 if(Demod_3(k)<0)
66     B3(k)=1;
67 else
68     B3(k)=0;
69 end
70 end
71 disp("Detected bit value ");
72 disp(B3');
73 disp("Actual bit values are");
74 disp(M1);
75 disp("Recovered and actual values are matching");

```

Scilab code Exa 11.7 To calculate the time required by mobile to make the required

```

1 // Exa 11.7
2 // To find power value to be set as a first
   approximation ans time required by mobile station
   to make changes as directed by base station.
3
4 clc;
5 clear all;
6
7 Prm=-97;//the signal strength from the base stations
   in dBm
8
9 //The constant ( K ) is the part of the broadcast
   message that is sent to the mobile by the base
   station on the paging channel.

```

```

10 K=-73; //dB
11 P2=18; //power as directed by BS (dBm)
12
13 //solution
14 Ptm=K-Prm;
15 printf('The mobile transmitter power be set as a
    first approximation of %d dBm \n ',Ptm);
16 Pwr_Redu=Ptm-P2;//power reduction
17 printf('Power reduction = %d dBm \n ',Pwr_Redu);
18 disp("Therefore, mobile requires 6 decrements each
    at 1.25 ms (1/800 sec) ");
19 Time=6*1.25;
20 printf(' Time required by mobile station to make
    changes as directed by base station is %.1f msec\
    n',Time);

```

Scilab code Exa 11.8 To determine the possible pair of soft slope and add intercept

```

1 // Exa 11.8
2 // To determine a possible pair of the SOFT-SLOPE
    and ADD-INTERCEPT values that will trigger the
    mobile station to send a PSMM to the base station
    and if the mobile station is IS-95 compliant,
    nd the value of T-COMP that could trigger the
    mobile station to generate a PSMM.
3
4 clc;
5 clear all;
6
7 P1=-95; //pilot1 in dBm
8 P2=-100; //pilot2 in dBm
9 P3=-101; //pilot3 in dBm
10 P4=-105; //pilot4 in dBm
11 P5=-102; //pilot in dBm
12 NoiseP=-107; //Receiver sensitivity (dBm)

```

```

13 Tadd=-13; //dB
14
15 //solution
16 //Pcj = received power of the jth pilot in the
    candidate set
17 // Pai= received power of the ith pilot in the
    active set
18 Pa1=P1-NoiseP;
19 Pa2=P2-NoiseP;
20 Pa3=P3-NoiseP;
21 Pa4=P4-NoiseP;
22 Pc5=P5-NoiseP;
23
24 X=10*log10(10^(0.1*Pa1)+10^(0.1*Pa2)+10^(0.1*Pa3)
    +10^(0.1*Pa4)+10^(0.1*Pc5));
25 disp("Max {14.22 *(SOFT-SLOPE)+ (ADD-INTERCEPT),-13
    }<=5");
26 disp("Thus we have two equations as follows");
27 disp("14.22*SOFT-SLOPE+ADD-INTERCEPT>=-13 and");
28 disp("14.22*SOFT-SLOPE+ADD-INTERCEPT<=5");
29 disp("Solving these equations, we get SOFT-SLOPE=
    0.5 and ADD-INTERCEPT=-4");
30 disp("For an IS-95-compliant mobile station (Pcj-Pai
    )>=0.5*T-COMP");
31 disp("Since P1>P2>P3>P4, we replace P4");
32 T_COMP=(P5-P4)/0.5;
33 disp("");
34 printf('The value of T-COMP that could trigger the
    mobile station to generate a PSMM should be <= %d
    dB (<= %d) .\n ',T_COMP,round(10^(0.1*T_COMP)));

```

Chapter 12

Mobility Management in Wireless Networks

Scilab code Exa 12.1 To evaluate the impact of LUs on the radio resource and calcu

```
1 // Exa 12.1
2 // To evaluate the impact of LUs on the radio
   resource and calculate the MSC/VLR transaction
   load using the uid ow model.
3
4 clc;
5 clear all;
6
7 P=10000; //Mobile density(mobiles/km^2)
8 R=500*10^-3; //km
9 V=10; ..//Average moving velocity of a mobile in
   Kmph
10 Nc=10; //No of cells per LA
11 N_LA=5; //Number of LAs per MSC/VLR
12
13 //Number of transactions and duration of each
   transaction to MSC/VLR per LU for different LU
   types are given in Table 12.1.(page no.374)
14
```

```

15 // solution
16 // L=length (km) of the cell exposed perimeter in an
    LA
17 L=6*R*(1/3+1/(2*sqrt(Nc)-3)); //Km
18 // lamdaLU=number of transactions processed by MSC/
    VLR in an LA perimeter of the jth cell per hour
19 LamdaLu=V*P*L/%pi; //Lus per hour
20
21
22 // case(1)
23 disp("Case-1:In the rst case, the jth cell
    located at the border of two LAs is related to
    the same MSC/VLR, only intra-VLR LUs are
    processed in the cell");
24 R1_LU=LamdaLu/3600*(1*600/1000); //resource
    occupancy from Table 12.1
25 disp("");
26 printf(' The resource occupancy in the jth cell due
    to MS LUs is %.1f Erlangs\n',R1_LU);
27
28 disp("This requires 18 channels at 1% blocking (
    refer to the Erlang-B table, Appendix A) or 18/8
    =2.25 traf c channel (about 1/4 of an RF
    channel, assuming there are 8 traf c channels
    per RF channel). ")
29
30 //case(2)
31 disp("case-2:In this case the jth cell is located at
    the border of two LAs related to two different
    VLRs.In this case, only inter-VLR LUs will be
    processed in the cell. We assume 80% of LUs are
    with TMSI and 20% of LUs are with IMSI");
32 R2_LU=LamdaLu/3600*(0.8*3500/1000+0.2*4000/1000);
    //from Table 12.1
33 disp("");
34 printf(' The resource occupancy in the jth cell due
    to MS LUs is %.2f Erlangs \n',R2_LU);
35 disp("This requires 75 channels at 1% blocking (

```



```

refer to the Erlang-B table, Appendix A) or
75/8=9.38 tr a f c channels (about 1.25 RF
channels).");
36
37
38 disp("MSC/VLR transaction load");
39
40 disp("We assume that one LA is in the center of the
region and the remaining four LAs are on the
border of the region.We also assume that, in the
perimeter cells at the border LAs, only intra-VLR
LUs are generated. For half of the perimeter
cells at the border LAs, only inter-VLR LUs are
generated.")
41
42 Np=6*sqrt(Nc/3)-3;//Number of cells located on
perimeter of an LA
43 disp("");
44 printf(' Number of cells where inter-VLR LUs occur
will be: %d \n',round(0.5*Np*4));
45 disp("");
46 printf(' Number of cells where intra-VLR LUs occur
will be: %d \n',4*Nc-16);
47 disp("");
48 TNLU=LamdaLu*(2*24+16*(0.8*14+0.2*16)); //from
table 12.1
49 printf(' The MSC/VLR transaction load using the
uid ow model is %.2f * 10^6 transactions at
peak hour \n',TNLU/10^6);

```

Chapter 13

Security in Wireless Systems

Scilab code Exa 13.1 To generate public and private keys for RSA algorithm

```
1 // Exa 13.1
2 // To generate public and private keys for RSA
  algorithm.
3
4 clc;
5 clear all;
6
7 //Two prime numbers
8 p=5;
9 q=7;
10
11 //solution
12 n=p*q;
13 z=(p-1)*(q-1);
14 e=input("Choose _e_ such that 1<e<z and e and n are
  coprime= ");
15 d=input("Choose _d_ such that e*d-1 should be
  exactly divisible by z= ");
16 printf('Public keys is (%d, %d)\n',n,e);
17 printf('Private key is (%d, %d)\n',n,d);
18
```

```

19 //Results
20 //Choose _e_ such that  $1 < e < z$  and e and n are coprime=
    5
21 //Choose _d_ such that  $e*d-1$  should be exactly
    divisible by  $z= 29$ 
22 //Public keys is (35, 5)
23 //Private key is (35, 29)

```

Scilab code Exa 13.2 To determine the secret encrypting key K using DH key exchange

```

1 // Exa 13.2
2 // To determine secret encrypting key K using DH key
    exchange algorithm.
3
4 clc;
5 clear all;
6
7 p=23; //prime number that both parties agreed upon
8 g=5; // g is primitive mod p
9 a=6; //party A choosen number
10 b=15; //party B choosen number
11
12 //solution
13 printf('Party A sends to party B as  $(g^a \text{ mod } p) = \%d$ 
    \n',modulo( $g^a, 23$ ));
14 printf(' Party B sends to party A as  $(g^b \text{ mod } p) =$ 
    \%d \n',modulo( $g^b, 23$ ));
15 printf(' Party A computes secret key as  $((g^b \text{ mod } p)^a \text{ mod } p) = \%d$  \n',modulo(modulo( $g^b, 23$ )^a,p));
16 printf(' Party B computes secret key as  $((g^a \text{ mod } p)^b \text{ mod } p) = \%d$  \n',modulo(modulo( $g^a, 23$ )^b,p));
17 disp("Thus both parties uses k=2 as secret key for
    encryption");

```

Chapter 14

Mobile Network and Transport Layer

Scilab code Exa 14.1 To determine minimum possible latency and window size to achieve this latency.

```
1 // Exa 14.1
2 // To determine the minimum possible latency and the
   minimum window size that achieves this latency.
3
4 clc;
5 clear all;
6
7 O=800*1000; //Object size(Bytes)
8 S=536*8; //max Segment Size(in bits)
9 RTT=0.1; //Round trip-time in sec
10 R=1*10^6; //Transmission rate of the link from the
   server to the client in bps
11
12 //solution
13 Lmin=2*RTT+(O/R); //latency(msec)
14 // For minimum latency  $(S/R) + RTT - (W*S/R) = 0$ ;
   Therefore
15 W=1+(RTT)/(S/R);
16 printf('The minimum possible latency is %d sec \n',
```

```

    Lmin);
17 printf(' The minimum window size is %.1f segments \n
    ',W);

```

Scilab code Exa 14.2 To calculate upper bound of the throughput for TP connection

```

1 // Exa 14.2
2 // To determine the upper bound of the throughput
  and the throughput with retransmissions due to
  errors.
3
4 clc;
5 clear all;
6
7 RTT=0.1; //Round trip-time in sec
8 MSS=536*8; //Maximum segment size in bits
9 p=0.01; // packet loss probability for the path
10 RTO=5*RTT; //Retransmission time out(from eqn 14.2
    on page 450)
11
12 //solution
13 R=0.93*MSS/(RTT*sqrt(p));
14 RR=MSS/(RTT*sqrt(1.33*p)+RTO*p*(1+32*p^2)*min(1,3*
    sqrt(0.75*p)));
15 printf('The upper bound of the throughput is %.4f
    Mbps \n',R*10^-6);
16 printf(' The throughput with retransmission due to
    errors is %.4f Mbps \n',RR*10^-6);

```

Chapter 17

Planning and Design of a Wireless Network

Scilab code Exa 17.1 To calculate various parameters for GSM 1800 network

```
1 // Exa 17.1
2 //To calculate –
3 //(a) average busy-hour t r a f c per subscriber ,
4 //(b) t r a f c capacity per cell ,
5 //(c) required number of base stations per zone , and
6 //(d) the hexagonal cell radius for the zone.
7
8 clc;
9 clear all;
10
11 Susage=150;//subscriber usage per month in mins
12 days=24;//days per month
13 busyhrs=6;//in a day
14 BW=4.8*10^3; //in kHz
15 Freqreuse=4/12;//Frequency reuse plan
16 chwidth=200; //in kHz
17 subscriber=50000;//Present subscriber count
18 Sgrowth=0.05;//Growth rate per year
19 Area=500; //in km
```

```

20  BTScapacity=30; //in Erlangs
21  N=4; //Initial installation design years
22
23  //solution
24  Erlangpersub=Susage/(days*busyhrs*60);
25  printf('Average busy-hour traffic per subscriber
        is %.4f Erlangs \n ',Erlangpersub);
26  RFcarriers=BW/chwidth;
27  RFcarrier_percell=RFcarriers/((Frequreuse^-1)*4); //
        freq reuse factor of 4
28
29  //Assuming 2 control channels per cell
30  CC=2; //control channels
31  TC_percell=2*RFcarriers/3-CC;
32  //Referring Erlang-B table in Appendix A
33  disp("Referring Erlang-B table in Appendix A,
        Traffic capacity of a GSM cell at 2% GoS for 14
        channels = 8.2 Erlangs ");
34  Tcapacity=8.2; // in Erlangs
35  disp("There are 3 cells per BTS");
36  BTS=3;
37  Traffic_perBTS=Tcapacity*BTS;
38  printf(' Traffic capacity per BTS is %.1f Erlangs ',
        Traffic_perBTS);
39  disp("Therefore, Traffic per BTS is less than BTS
        capacity(30 Erlangs)")
40  maxsubscriber=Traffic_perBTS/Erlangpersub;
41  initialsub=subscriber*(1+Sgrowth)^N;
42  BTS_perZone=initialsub/maxsubscriber;
43  printf(' The required number of base stations per
        zone are %d \n ',round(BTS_perZone));
44  cellRadius=(Area/(BTS_perZone*2.6))^0.5;
45  printf('The hexagonal cell radius is %.1f km \n ',
        cellRadius);

```

Scilab code Exa 17.2 To estimate the data and voice traffic per subscriber and per

```
1 // Exa 17.2
2 // To calculate voice and data traffic per cell.
3
4 clc;
5 clear all;
6
7 usage=150; //subscriber usage per month in mins
8 days=24; //Days in a month
9 BHrs=6; //Busy hours per day
10 BW=4.8; //in MHz
11 RFch=200; //in kHz
12 Psubscribers=50000; //present subscriber count
13 growth=0.05; //subscriber growth per year
14 rollover=4; //network roll over period
15 NPCS=5; //Number of packet calls per session
16 NPP=25; //Number of packets within a packet call
17 Tr=120; //Reading time between packet calls(sec)
18 NBP=480*8; //Packet size(in bits)
19 Tint=0.01; //Time interval between two packets(sec)
20 Ttot=3000; //Total packet service holding time
21 BH_PS=0.15; //Busy hour packet sessions per
    subscriber
22 Penetration=0.25;
23 datarate=48; //in kbps
24 PTT=10; //Packet transmission time(sec)
25 BTS=40; //NO of BTS sites
26
27 //solution
28 Bitstx_duringPTT=NPCS*NPP*NBP/1000;
29 PST=PTT+Tr*(NPCS-1)+Tint*(NPP-1);
30 PT_duringBH=BH_PS*Ttot/PST;
31 Bits_persub_persec=Bitstx_duringPTT*PT_duringBH
    /(60*60);
32 VoiceErlangs=usage/(days*BHrs*60);
33 Initial_subscribers=round(Psubscribers*(1+growth)^
    rollover);
```



```

34 Data_subscribers=Initial_subscribers*Penetration;
35 Totalvoice=Initial_subscribers*VoiceErlangs;
36 Voicetraffic_perBTS=Totalvoice/BTS;
37 printf('Voice Traffic per Cell(sector) is %.2f
    Erlangs \n ',Voicetraffic_perBTS/3);
38 Totaldata=Data_subscribers*Bits_persub_persec;
39 Datathroughput_perBTS=Totaldata/BTS;
40 printf("Data throughput per Cell(sector) is %.2f
    kbps \n ',Datathroughput_perBTS/3);

```

Scilab code Exa 17.3 To calculate data Erlangs along with TS utilization and capacity

```

1 // Exa 17.3
2 //Using traffic data per cell for a GSM/GPRS
   network from Example 17.2
3 // To calculate –
4 // (a) data Erlangs ,
5 // (b) time slot utilization , and
6 // (c) TS capacity .
7
8 clc;
9 clear all;
10
11 Holdtime=120;//Average holding time during Busy
   Hours(in sec)
12 Tx=3;// No of transreceivers
13 Tsig=3;//No of TSS per cell for signaling
14 RLC=0.80;//Radio link control efficiency
15 Radioblocks=9000;//Total numbers of transmitted
   radio blocks
16 TSdata=3;//TSS allocated for data traffic per cell
17 Datarate=15.5; //From eg 17.2
18 Voicetraffic=8.82; //From eg.17.2
19 Duration=0.02;//Duration of block in sec
20

```

```

21 //solution
22 DataEr=Radioblocks*Duration/Holdtime;
23 printf('Data Erlangs = %.1f \n ',DataEr);
24 TSuti=DataEr/TSsig;
25 printf('Time Slot(TS) utilization = %.1f \n ',TSuti)
    ;
26 Throughput=(Datarate/TSdata)*RLC;
27 TScap=Throughput/TSuti;
28 printf("TS capacity is %.2f kbps \n ',TScap);

```

Scilab code Exa 17.4 To develop downlink and uplink cell budget and calculate cell

```

1 // Exa 17.4
2 //To calculate the cell radius.
3
4 clc;
5 clear all;
6
7 Pt=36;//Base station transmitted power in dBm
8 Pms=24;//mobile station transmitted power in dBm
9 Nms=8;//mobile station noise figure in dB
10 Nbs=5;//Base station nise figure in dB
11 Ga=18;//Base station transmit and receive antenna
    gain in dBi
12 Gm=0;//Mobile antenna gain in dBi
13 SNR=12;// in dB
14 Lc_TX=5;//BS transmit antenna cable , connector , and
    lter losses in dB
15 Lc_RX=2;//BS receiver antenna cable , connector , and
    lter losses in dB
16 Bodyloss=3;// Body losses at mobile
17 fading=10.2;// in dB
18 ThermalNoise=-174;// in dBm/Hz
19 Gdiversity=5;//Antenna diversity gain at BS in dB
20 //Assuming standard value of RF channel as

```

```

21 RFch=200*10^3; //in Hz
22
23 //solution
24 N=ThermalNoise+10*log10(RFch)+Nms;
25 Smin=N+SNR;
26 Smean=Smin+fading+Bodyloss;
27 Lp=Pt-Lc_TX+Ga-Smean;
28 N1=ThermalNoise+10*log10(RFch)+Nbs;
29 Smin=N1+SNR-Gdiversity;
30 Smean1=Smin+fading+Bodyloss;
31 Lp1=Pms-Smean1+Ga-Lc_RX;
32 disp("Using uplink path loss and Hata model to
      calculate cell radius");
33 R=10^((Lp1-133.2)/33.8);
34 printf(' Cell radius is %.1f km \n',R);

```

Scilab code Exa 17.5 To calculate Uplink cell load factor and pole capacity

```

1 // Exa 17.5
2 // To calculate uplink cell load factor , number of
   voice users and poll capacity of the cell.
3
4 clc;
5 clear all;
6
7 Ri=12.2*10^3; //Information rate in bps
8 Rc=3.84*10^6; //Chip rate in cps(chips per second)
9 Eb_Nt=4; //in dB
10 Imargin=2; //Interference margin(3 dB)
11 B=0.5; //Interference factor due to other cells
12 Vi=0.65; //Channel activity factor
13
14 //solution
15 Eb_Ntreqd=10^(Eb_Nt/10);
16 LF_peruser=(1+B)*(1/(1+(Rc/Ri)*(1/Eb_Ntreqd)*(1/Vi)))

```

```

    ); //M(no of users=1) in Eq 17.13
17 printf("Cell load factor per voice user is %.5f \n
    ',LF_peruser);
18 CellLoading=(Imargin-1)/Imargin;
19 VoiceUsers=CellLoading/LF_peruser;
20 printf('No of Voice Users are %d per cell \n ',
    VoiceUsers);
21 //From EQ 17.6 assuming Power control efficiency=1
22 Polecap=Rc/(Ri*Vi*(1+B)*Eb_Ntreqd);
23 printf('Pole Capacity is %d \n',Polecap);

```

Scilab code Exa 17.6 To calculate uplink throughput for WCDMA data service

```

1 // Exa 17.6
2 // To calculate Uplink throughput for a WCDMA cell.
3
4 clc;
5 clear all;
6
7 Eb_Nt=1; //in dB
8 cellLoading=0.5;//Required interference margin(3 dB)
9 B=0.5;//Interference factor due to other cells
10 Vi=1;//Channel activity factor
11
12 //solution
13 Eb_Ntreqd=10^(Eb_Nt/10);
14 //Assuming standard value of chip rate as 3.84Mcps
15 Rc=3.84*10^6;//in cps(chips per second)
16 Throughput=(cellLoading*Rc)/(Eb_Ntreqd*(1+B));
17 printf('Uplink Throughput is %d kbps \n ',Throughput
    /1000);

```

Scilab code Exa 17.7 To calculate downlink cell load factor and number of voice users

```

1 // Exa 17.7
2 // To calculate downlink cell load-factor and number
   of voice users per cell for a WCDMA system.
3
4 clc;
5 clear all;
6
7 Ri=12.2*10^3; //Information rate in bps
8 Rc=3.84*10^6; // Chip rate in chips per second
9 Eb_Nt=4; // in dB
10 Eb_Nt=10^(Eb_Nt/10);
11 B=0.5; //Average interference factor due to other
   cells
12 Zeta=0.6; // orthogonality factor
13 Imargin=2; //Interference margin(3 dB)
14 Vi=0.65 //assuming Channel activity factor as 0.65
15
16 //solution
17 Loadfactor_peruser=(Zeta+B)*(1/((Rc/Ri)*(1/Eb_Nt)
   *(1/Vi)))
18 printf('Downlink cell load factor is %.4f \n ',
   Loadfactor_peruser);
19 cellLoading=(Imargin-1)/Imargin;
20 Voiceusers=cellLoading/Loadfactor_peruser;
21 printf('No of voice users per cell are %d \n ',
   Voiceusers);

```

Scilab code Exa 17.8 To determine minimum signal power and maximum allowable path

```

1 // Exa 17.8
2 // To calculate minimum signal power required and
   maximum allowable path loss.
3
4 clc;
5 clear all;

```

```

6
7 N0=-174; //Noise density in dBm/Hz
8 Bc=1.25; //Channel bandwidth in MHz
9 Rc=1.2288; //Chip rate in Mcps
10 Nf=6; //Receiver Noise figure in dB
11 Pt=27; //Effective radiated power from mobile in
    dBm
12 Lct=0.5; //Transmitter cable and connector loss in
    dBm
13 Lbody=1.5; //Body loss in dB
14 Lcr=2; //Receiver cable and connector loss in dB
15 Mint=0; //Interference margin in dB
16 Mfading=2; //fast fading margin in dB
17 Lpent=8; //Penetration loss in dB
18 Gm=0; //Transmitter antenna gain in dBi
19 Gb=12; //Receiver antenna gain in dBi
20 Fm=8; //Fade margin in dB
21 Eb_Nt=7; // in dB
22
23 //solution
24 Nth=N0+Nf;
25 S_Nt=Eb_Nt+10*log10((Rc*10^6)/(Bc*10^6));
26 Smin=S_Nt+10*log10(Rc*10^6)+Nth;
27 Lpmax=(Pt-Smin)+(Gb+Gm)-(Lbody+Lct+Lcr+Fm+Lpent)-
    Mint-Mfading;
28 printf('Minimum signal power required is %.2f dBm \n
    ',Smin);
29 printf('Maximum allowable path loss is %.2f dB \n ',
    Lpmax);

```

Scilab code Exa 17.9 To develop a radio link budget for uplink and downlink

```

1 //Exa 17.9
2 // To calculate Radio link budget for uplink and
    downlink

```

```

3 // Referring Table 17.1 on page no 613
4
5 clc;
6 clear all;
7
8 Rc=3.84; //Chip rate in Mcps
9 Ri=16; //Data rate in kbps
10 UL=0.5; //UL loading factor
11 DL=0.9; //DL loading factor
12 Eb_NtU=4; //in dB
13 Eb_NtD=6; // in dB
14 Gm=0; //Mobile antenna gain in dBi
15 Gb=18; //Base station gain in dBi
16
17 //solution
18 disp("The Okumara-Hata model for an urban macro-cell
      with a base station antenna height of 25m, a
      mobile station height of 1.5m, and a carrier
      frequency of 1950MHz gives  $L_p = 138.5 + 35.7 * \log_{10}(R)$ 
      ) where R is radius of hexagonal cell");
19 disp("From table 17.1,  $L_p$ (Allowable path loss) for
      uplink is 139.65 dB");
20 R=10^((139.65-138.5)/35.7);
21 printf(' Cell Radius is %.3f km \n ',R);
22 Area=round(2.6*R^2);
23 printf('Area covered by hexagonal cell is %d km^2 \n
      ',Area);
24 printf('Number of BTSs required to cover an area of
      2400 Km^2 are %d \n ',2400/Area);

```

Scilab code Exa 17.10 To find the number of users supported on the downlink of a W

```

1 //Exa 17.10
2 //To calculate No of users that can be supported on
  the downlink of the WCDMA network.

```

```

3
4 clc;
5 clear all;
6
7 Rc=3.84; //chip rate in Mcps
8 N=3; //Noise rise in dB
9 OF=0.8; //orthogonality factor
10 B=0.55; //Interference from other cells
11 Eb_NO=4; //in dB
12 Sec_Eff=0.85; //Sector efficiency
13 Pwr_Eff=0.80; //Power control efficiency
14 Y=1.2; //Retransmit rate
15 X=10; //10MB at 384Kbps
16 X1=2; //2MB at 144Kbps
17 X2=1; //1MB at 64Kbps
18
19 //solution
20 //Assuming Voice activity=Vf=1
21 Vf=1;
22 AvgDR=Y*X*106*(1/3600)+Y*X1*106*(1/3600)+Y*X2
    *106*(1/3600);
23 CLoad=(N-1)/(N+1);
24 DLcap=(Rc*106*Pwr_Eff*Sec_Eff)/(((10-(Eb_NO/10)))*(B
    +OF)*Vf));
25 Allowcap=CLoad*DLcap;
26 users=Allowcap/AvgDR;
27 printf('NO of users that can be supported on the
    downlink of the WCDMA network are %d \n ',round(
    users));

```

Scilab code Exa 17.11 find the average throughput for various cases

```

1 // Exa 17.11
2 // To calculate average throughput and compare it
    with equal latency condition.

```



```

3
4 clc;
5 clear all;
6
7 P1=1/2; //relative frequency of packets for user
   class1
8 P2=1/3; //relative frequency of packets for user
   class2
9 P3=1/6; //relative frequency of packets for user
   class3
10 R1=16; //data rate in kbps for P1
11 R2=64; //data rate in kbps for P2
12 R3=1024; //data rate in kbps for P3
13 S1=16; //number of slots assigned to the R1 user
14 S2=8; //number of slots assigned to the R2 user
15 S3=2; //number of slots assigned to the R3 user
16
17 //solution
18 //Using Equation 17.20 from page no 616
19 Ravg=(P1*R1*S1+P2*R2*S2+P3*R3*S3)/(P1*S1+P2*S2+P3*S3
   );
20 // For equal latency , using Eq 17.18
21 Rav=1/(P1/R1+P2/R2+P3/R3);
22 // For Latency ratio=4, using Eq 17.19 from page no
   616
23 PL=4;
24 C=(P1+P2+PL*P3)/(P1/R1+P2/R2+P3/R3);
25 printf('The average throughput for equal access
   condition is %.1f kbps \n ',Ravg);
26 printf('The average throughput by considering equal
   latency is %.1f kbps \n ',Rav);
27 printf('The average throughput by considering
   latency ratio as 4 is %.2f kkbps \n ',C)
28 disp("It is observed that equal access provides the
   highest average output")

```

Scilab code Exa 17.12 To find the allowable throughput of a reverse link of a CDMA

```
1 // Exa 17.12
2 // To calculate allowable throughput of reverse link
   in cdma 2000.
3
4 clc;
5 clear all;
6
7 Ec_Nt=-23; //in dB
8 DRC=-1.5; //DRC gain with respect to pilot in dB
9 Tg=3.75; // Traffic channel gain with respect to
   pilot in dB
10 B=0.85; //Interference factor due to other cells
11
12 //solution
13 Mmax=(1/(1+10^(DRC/10)+10^(Tg/10)))*(1/((10^(Ec_Nt
   /10)*(1+B)))));
14 //The sector loading can be expressed as a fraction
   of the pole capacity Mmax. This is typically 70%
   of the pole capacity.
15 M_allow=int(0.7*Mmax);
16 //From table 17.2 we get Traffic channel rate as 9.6
   kbps since we are given traffic channel gain with
   respect to pilot as 3.75 dB
17 Ri=9.6; //in kbps(see table 17.2)
18 Tput=Ri*M_allow;
19 printf(' Allowable reverse link throughput is %d
   kbps \n ',round(Tput));
```

Scilab code Exa 17.13 To estimate average SINR of a HSDPA

```

1 // Exa 17.13
2 // To calculate average SINR of HSDPA.
3
4 clc;
5 clear all;
6
7 Pmax=5.5; //Maximum transmit power of DSCH in watts
8 Pbs=18; // Total base station power in watts
9 alpha=0.2; //downlink orthogonality factor
10 G=0.363; // geometry factor
11 SF=16; //Spreading Factor for DSCH;fixed at value of
    16
12
13 //solution
14 // Using equation no 17.27 given on page no 623
15 SINR=SF*(Pmax/(Pbs*(1-alpha+(1/G))));
16 // In dB
17 SINR_db=10*log10(SINR);
18
19 printf('The average SINR of HSDPA is %.1f dB = %.4f
    \n ',SINR_db,SINR);

```

Scilab code Exa 17.14 To find bandwidth of a Iub interface

```

1 // Exa 17.15
2 // To calculate bandwidth of Iub interface.
3
4 clc;
5 clear all;
6
7 Users=350; //no of users supported
8 ExpectedTraf=1.8; // From section 17.7 (in Kbps)
9 BHTraf=1.785; //Busy hour traffic in kbps
10 BTS=180;
11

```

```

12 //solution
13 IubBW=(ExpectedTraf*Users*BHTraf)/1000; // in Mbps
14 TotalBW=BTS*IubBW;
15 printf('Required total bandwidth of Iub Interface is
        %.2f Mbps \n ',TotalBW);

```

Scilab code Exa 17.15 To find the required RNCs

```

1 // Exa 17.15
2 // To calculate No of RNC required.
3
4 clc;
5 clear all;
6
7 BTS=800; //No of BTS sites
8 Sec=3; //No of sectors per BTS
9 freq_sec=2; //No of frequencies used per sector
10 cellsRNC=1152; //Maximum capacity of cellRNC
11 btsRNC=384; //One RNC can support btsRNC(BTSs)
12 VE=25; //Voice service(mErl/subscriber)
13 BRV=16; // bitrate Voice
14 CS1=10; //CS data service 1(mErl/subscriber)
15 BRC1=32; //bit rate for CS1 in kbps
16 CS2=5; //CS data service 2(mErl/subscriber)
17 BRC2=64; //bit rate for CS2 in kbps
18 PSdata=0.2; //PS data service(kbps per subscriber)
19 PSoverhead=0.15;
20 SHO=0.4; //40%
21 Totalsub=350000; //Total subscribers
22 Maxcap=196; //Maximum Iub capacity of tpRNC (in Mbps)
23 FR1=0.9; FR2=0.9; FR3=0.9; //Filler rates
24
25 //solution
26 RNCreqd=(BTS*Sec*freq_sec)/(cellsRNC*FR1); //from eqn
        17.28

```

```

27 printf('using equation 17.28,Number of RNC required
    are %d \n ',round(RNCreqd));
28 RNC_reqd=BTS/(btsRNC*FR2);//from eqn 17.29
29 printf('using equation 17.29,Number of RNC required
    are %d \n ',round(RNC_reqd));
30 RNCrequired=((VE/1000*BRV+CS1/1000*BRC1+CS2/1000*
    BRC2+(PSdata/(1-PSoverhead)))*(1+SH0)*Totalsub)/(
    Maxcap*1000*FR3);//from eqn 17.30
31 printf('using equation 17.30,Number of RNC required
    are %d \n ',round(RNCrequired));
32
33 printf(' We select first value which is %d RNCs \n '
    ,round(RNCreqd));

```

Chapter 19

Wireless Personal Area Network Bluetooth

Scilab code Exa 19.1 To find hopping rate and various other parameters of Bluetooth

```
1 // Exa 19.1
2 // To calculate the hopping rate of bluetooth , No of
   bits transmitted in one slot , efficiency of
   packet transmission , and No of times a packet is
   sent.
3
4 clc;
5 clear all;
6
7 SS=80; //Frame length of HV3 voice packet
8 R=64*103; //Data rate in bps
9
10 //solution
11 TS=240; //No of bits carried in a slot
12 //From table 19.3:Bluetooth air interface details ,
   we get Frequency hopping rate as 1600hopes/second
13
14 //From table 19.5, we can note that for HV3, No of
   slots are 1 and carry in all(80(user voice sample
```

```

    )+160(parity bits))=240 bits in one slot packet
15 HR=1600; //hopes/second
16 Eff=SS/TS;
17 x=R/SS; //x is number of times a packet is sent
18 printf('Hopping rate is %d hopes/sec \n ',HR);
19 printf('No of bits transmitted in one slot are %d \n
    ',TS);
20 printf('Efficiency of packet transmission is %.4f \n
    ',Eff);
21 printf('Number of times a packet is sent is equal to
    %d \n ',x);

```

Scilab code Exa 19.2 To find the associated data rate of a symmetric 1slot DM1

```

1 // Exa 19.2
2 // To find the associated data rate.
3
4 clc;
5 clear all;
6
7 R=136; // bits per slot
8 SR=800; // no of slots per second
9
10 //solution
11 // A symmetric 1-slot DM1 link is setup between a
    master and a slave
12 AR=R*SR; //Data rate in bps
13
14 printf('Associated data rate is %0.1f kbps \n ',AR
    /1000);

```

Chapter 21

Wireless Local Area Network

Scilab code Exa 21.1 To find number of users supported by WLAN and its bandwidth e

```
1 // Exa 21.1
2 // To find number of users that can be supported by
   the WLAN and the bandwidth e f ciency .
3
4 clc;
5 clear all;
6
7 F1=902; //lower limit frequency MHz
8 Fh=928; //higher limit frequency in MHz
9 Rt=0.5; //symbol transmission rate in Mega symbols
   per sec
10 S=16; //No of symbols
11 BER=10^-5; //Bir error rate
12 SG=2.6; //sector gain
13 B=0.5; //Interference factor
14 a=0.9; //power control efficiency
15
16 //solution
17 BW=Fh-F1;
18 Rb=Rt*log2(S);
19 Gp=BW/Rb;
```



```

20 // BER = 10-5 = 0.5*erfc(sqrt(Eb_No))
21 deff('y=f(x)', 'y=0.5*erfc(sqrt(x))-10-5')
22 [x,v,info]=fsolve(0.1,f); //x=Eb_No
23 M=Gp/x * 1/(1+B) * SG * a;
24 printf('Number of users that can be supported by the
        WLAN are %d \n',M);
25 eff=Rb*int(M)/BW;
26 printf(' The bandwidth efficiency is %.2f bps/Hz \n'
        ,eff);

```

Scilab code Exa 21.2 To find hopping bandwidth and various other parameters for FH

```

1 //Exa 21.2
2 // To find-
3 //a) the hopping bandwidth ,
4 //b) What is the chip-rate ,
5 //c) How many chips are there in each data symbol ,
6 //d) What is the processing gain .
7
8 clc;
9 clear all;
10
11 Stepsize=200; //in Hz
12 Chipsmin=20; //length of linear feedback shift
    register
13 Datarate=1.2*103; //bps
14
15 //solution
16 No_of_tones=2Chipsmin;
17 Bss=No_of_tones*Stepsize;
18 Chiprate=Datarate*Chipsmin;
19 Gp=Bss/Datarate; //processing gain
20 Symbolrate=Datarate/3; //8-ary FSK is used
21 Chips_symbol=Chiprate/Symbolrate;
22 printf('The Hopping Bandwidth is %.3f MHz\n',Bss

```

```

    /10^6);
23 printf(' The chiprate is %d kchip/sec\n',Chiprate
    /10^3);
24 printf(' Chips per symbol are %d \n',Chips_symbol);
25 printf(' The processing gain is %.1f\n',Gp);

```

Scilab code Exa 21.3 To find bandwidth of a subchannel and various other parameter

```

1 //Exa 21.3
2 //To find-
3 // a) The bandwidth of a subchannel ,
4 // b) modulation efficiency ,
5 // c) user symbol rate ,
6 // d) user data rate if the information bits are
    encoded with a rate of 3/4,
7 // e) time utilization efficiency of the system.
8
9 clc;
10 clear all;
11
12 InfoSc=48;//Information subcarriers
13 SyncSc=4;//synchronization subcarriers
14 ReservedSc=12;//Reserved subcarriers
15 Symrate=250; //ksps(kilosymbols per second)
16 BW=20; //in MHz
17 Grdt=800; //Guard time in nsec
18
19 //solution
20 TotalSc=InfoSc+SyncSc+ReservedSc;//Total subcarriers
21 BW_Sch=BW*10^6/TotalSc;//BW of subchannel
22 Mod_eff=Symrate*10^3/(BW_Sch);//Modulation
    efficiency
23 User_txrate=InfoSc*Symrate*10^3;
24 User_bitsymbol=4; //16-QPSK is used
25 disp("From table 21.7 For modulation scheme as 16-

```

```

    QAM and coding rate =3/4 then User data rate will
    be 36Mbps");
26 User_DR=36; //Mbps
27 Sym_Dur=1/(Symrate*10^3);
28 TimeUti=Sym_Dur/(Sym_Dur+(Grdt/10^9));
29
30 printf(' The bandwidth of subchannel is %.1f kHz\n',
    BW_Sch/10^3);
31 printf(' Modulation efficiency is %.1f symbols/sec/
    Hz \n',Mod_eff);
32 printf(' User symbol rate is %d Msps \n',User_txrate
    /10^6);
33 printf(' Time Utilization efficiency is %.2f \n',
    TimeUti);

```

Scilab code Exa 21.4 To find coverage of the AP

```

1 //Exa 21.4
2 // To determine the coverage of AP.
3
4 clc;
5 clear all;
6
7 Eb_No=10; //in dB
8 Noise=-120; //in dBm
9 Pt=20; //in mwatt
10 R=1; //Data rate in Mbps
11 CHBW=0.5; //BW in MHz
12 A=37.7; //path loss at the rst meter in dB
13 Y=3.3; //path loss exponent
14 Lf=19; //function relating power loss with number of
    oors n (in dB)
15 Ls=10; // lognormally distributed random variable
    representing the shadow effect in dB
16

```

```

17 //solution
18 S2Nreqd=Eb_No*R/CHBW;
19 Rx_sensi=Noise+S2Nreqd;
20 Lp=10*log10(20)-Rx_sensi;
21 //Lp=A+10Ylod(d)+Lf+Ls; therefore
22 d=10^((Lp-A-Lf-Ls)/(10*Y));
23 printf('The coverage of AP is %.1f metres \n',d);

```

Scilab code Exa 21.5 To calculate coded symbol and bit transmission rate per subcarrier

```

1 // Exa 21.5
2 // To determine the coded symbol transmission rate
   per subcarrier and bit transmission rate per
   subcarrier for each of the two modes.
3
4 clc;
5 clear all;
6
7 R=3/4; //code rate of convolution encoder
8 M1=9; //payload transmission rate in Mbps for mode 1
9 M2=36; //payload transmission rate in Mbps for mode
   2
10
11 //solution
12 D1=M1*10^6/48; //user data rate in kbps for mode 1
13 D2=M2*10^6/48; //user data rate in kbps for mode 2
14 //Referring to Table 21.11
15 printf('Data transmission rate per carrier with 3/4
   convolution encoder are %.1f Kbps and %d Kbps \n',
   ,D1/10^3,D2/10^3);
16 C1=D1/R;
17 C2=D2/R;
18 printf(' Carrier transmission rate with R=3/4
   convolutional encoder are %d Kbps and %d Kbps\n',
   C1/10^3,C2/10^3);

```

```

19 printf(' Carrier symbol rate with R=3/4
    convolutional encoder are %d ksps and %d Ksps \n'
    ,C1/10^3,C2/4/10^3); //Mode1 as BPSK and MOde2
    as 16-QAM

```

Scilab code Exa 21.6 To find user data rate for HIPERLAN 2

```

1 // Exa 21.6
2 // To determine the user data rate for HIPERLAN/2.
3
4 clc;
5 clear all;
6
7 R=3/4; //code rate for convolution encoder
8
9 //solution
10 //64-QAM modulation is used
11 Sc=250; //Carrier symbol rate(ksps) from Exa 21.5
12 Bits_sym=log2(64); //64-QAM is used
13 User_R=Bits_sym*Sc*10^3*R*48;
14 printf('The user data rate is %d Mbps \n',User_R
    /10^6);

```

Scilab code Exa 21.7 To determine the collision probability of a FH packet

```

1 // Exa 21.7
2 // To determine the PER(Packet error rate) for FH(
    Frequency Hopping packet) and DS(Direct spread
    packet).
3
4 clc;
5 clear all;
6

```

```

7 D=1000*8; //packet size in bits
8 R=2*10^6; //transmission rate in bps
9 L=3; //msec(Dwell time)
10 H=0.625; //msec(Duration of BT packet)
11
12 //solution
13 Tw=10^3*D/R; //the packet duration of IEEE 802.11
    in msec
14 H_L=1;
15 G=(H_L)*L-Tw-H;
16 Gm=abs(G);
17 PER_FH=1-((1-Gm/L)*(78/79)^(H_L)+Gm/L*(78/79)^((H_L)
    -G/Gm));
18 PER_DS=1-((1-Gm/L)*(57/79)^(H_L)+Gm/L*(57/79)^((H_L)
    -G/Gm));
19 printf('The PER for FH packet and PER for DS packet
    are %d percent & %.2f percent respectively ',round
    (PER_FH*100),PER_DS*100);
20 disp("The collision probability with 802.11 DS is
    much higher than with 802.11 FH.")

```

Scilab code Exa 21.8 To calculate min SIR and using this SIR and other data calcul

```

1 // Exa 21.8
2 // To determine SIRmin and r_max.
3
4 clc
5 clear all;
6
7 d=10; // distance between AP and IEEE 802.11 device
    in metres
8 Y=4; //path loss exponent
9 PBt=20; //the transmitted power by the BT in dBm
10 PAp=40; //the transmitted power by the AP in dBm
11 Pe=10^-5; //acceptable error probability

```

```

12
13 //solution
14 //Pe=0.5*e^(-0.5*Eb/No)
15 SIR=log(Pe/0.5)/(-0.5);// signal-to-interference
    ratio
16 rmax=d*(SIR*PBt/PAP)^(1/Y);// range of interference
    between Bluetooth and 802.11 device
17 printf('Minimum SIR is %.2f dB = %.1f \n',10*log10(
    SIR),SIR);
18 printf(' Maximum coverage range is  %.2f metres \n',
    rmax);

```

Scilab code Exa 21.9 To calculate Rmax for given interference scenarios

```

1 // Exa 21.9
2 // To calculate rmax for the interference scenarios
    (see Figure 21.21) using Smin from Example 21.8.
3
4 clc;
5 clear all;
6
7 SIRmin=21.6; //From eg 21.8 i.e(13.36 dB)
8 d=10; //distance between AP and IEEE 802.11 device
    in m
9 PMS=40; // transmitted power of the IEEE 802.11
    device in dBm
10 PBt=20; //the transmitted power by the BT in dBm
11 Y=4 ; //path loss exponent
12
13 //solution
14 rmax=d*(SIRmin*PMS/PBt)^(1/Y);
15 printf('Maximum coverage range is %.1f metres \n',
    rmax);

```

Scilab code Exa 21.10 To calculate Rmax for IEEE 802 11 FH and DS devices

```
1 // Exa 21.10
2 // Repeat Problems 21.8 and 21.9, if the IEEE 802.11
   FH device is replaced by the IEEE 802.11 DS
   device (Gp=11).
3
4 clc
5 clear all;
6
7 Gp=11; //processing gain(given)
8 //Defining variables from Exa 21.8 & 21.9
9 PBt=20; // transmitted power by the BT in dBm
10 PMs=40; // transmitted power of the IEEE 802.11
   device in dBm
11 PAp=40; // transmitted power by the AP in dBm
12 d=10; // distance between AP and IEEE 802.11 device
   in m
13 Y=4; //path loss exponent
14 Pe=10^-5; //Error probability
15
16 //solution
17 //Pe=0.5*e^(-0.5*Eb/No)
18 SIR=log(Pe/0.5)/(-0.5);
19 r1max=d*(SIR*PBt/(PAp*Gp))^(1/Y); // range of
   interference between Bluetooth and 802.11 device
20 printf(' Maximum coverage range for IEEE 802.11 DS
   is %.2f metres \n',r1max);
21 r2max=d*(SIR*PMs/(PBt*Gp))^(1/Y);
22 printf(' Maximum coverage range for IEEE 802.11 FH
   is %.2f metres \n',r2max);
23 disp(" Thus, the interference ranges are smaller for
   the IEEE 802.11 DS device compared to the IEEE
   802.11 FH device.")
```


Chapter 24

Spreading Codes used in CDMA

Scilab code Exa 24.1 To generate m sequence and demonstrate its properties

```
1 // Exa D.1
2 // Using the shift register shown in Figure D.3,
   generate an m-sequence and demonstrate its
   properties.
3
4 clc;
5 clear all;
6
7 //solution
8 //Referring Fig D.3
9 x=[0 0 1]; //Initial stage
10 output=x(3);
11 disp(" First m-sequence using 3-stage shift register
   .");
12 disp("          x1 x2 x3          output");
13 printf(' Initial    %d %d %d          %d \n ',x
   (1),x(2),x(3),output);
14 for i= 1:7
15     printf('Shift %d',i);
```

```

16     x(3)=x(2);
17     if(x(3)==1) //TO get values in range of [-1 1]
           for plot
18         dummy(i)=-1
19 else
20         dummy(i)=1;
21     end
22     x(2)=x(1);
23     if(output== 1& x(3)==1) //As new x(1)=prev stage
           x(3) ored
           prev stage x(2)
24         x(1)=0;
25     else
26         if(output== 0& x(3)==0)
27             x(1)=0;
28         else
29             x(1)=1;
30         end
31     end
32
33     printf('      %d %d %d ',x(1),x(2),x(3));
34     output=x(3);
35     printf('          %d',output);
36     printf('\n ');
37 end
38 bar(dummy,0.2,'green');
39 xlabel("Time","FontSize",5);
40 title("7-chip first m-sequence for one T period","
        FontSize",5);
41 disp("The properties of m-sequence in Figure(0) are -
        ");
42 disp("Number of -1s = 4 , Number of 1s = 3 ");
43 disp("Run length 1 = 2 , Run length 2 = 1");
44 disp("Run length  = 1");

```

Scilab code Exa 24.2 To generate second m sequence

```
1 // Exa D.2
2 // what is the location of the modulo-2 adder for
  the second m-sequence? Generate the second m-
  sequence.
3
4 clc;
5 clear all;
6
7 //solution
8 disp("The location of modulo-2 adder for the second
  m-sequence is shown in Figure D.5(in the book) i.e
  Modulo-2 adder should be between first(x1) and
  second(x2) shift register.");
9 x=[0 0 1]; //Initial stage
10 output=x(3);
11 disp("Second m-sequence using 3-stage register");
12 disp("          x1 x2 x3          output");
13 printf(' Initial      %d %d %d          %d \n ',x
  (1),x(2),x(3),output);
14 for i= 1:7
15     printf(' Shift %d',i);
16     x(3)=x(2);
17     if(x(3)==1) //TO get values in range of [-1 1]
        for plot
18         dummy(i)=-1
19     else
20         dummy(i)=1;
21     end
22     x(2)=x(1);
23     if(output== 1& x(2)==1) //As new x(1)=prev stage
        x(3) ored
        prev stage x(2)
24         x(1)=0;
25     else
26         if(output== 0& x(2)==0)
27             x(1)=0;
```

```

28         else
29             x(1)=1;
30         end
31     end
32
33     printf('      %d  %d  %d ',x(1),x(2),x(3));
34     output=x(3);
35     printf('                %d',output);
36     printf('\n ');
37 end
38 figure(1);
39 bar(dummy,0.2,'green');
40 xlabel("Time","FontSize",5);
41 title("7-chip second m-sequence for one T period","
      FontSize",5);
42 disp("The properties of m-sequence in Figure(1) are -
      ");
43 disp("Number of -1s = 4 , Number of 1s = 3 ");
44 disp("Run length 1 = 2 , Run length 2 = 1");
45 disp("Run length  = 1");

```
