

Scilab Textbook Companion for
Satellite Communication
by A. K. Maini And V. Agrawal¹

Created by
Laxman Ghanasham Sole
B.Tech
Electronics Engineering
Vishwakarma Institute of Technology
College Teacher
Prof. Vijay Mane
Cross-Checked by
K. V. P. Pradeep

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: Satellite Communication

Author: A. K. Maini And V. Agrawal

Publisher: Wiley India Pvt. Ltd. New Delhi

Edition: 1

Year: 2010

ISBN: 978-81-265-2071-8

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 Satellite orbits and Trajectories	5
3 Satellite Launch and In orbit Operations	15
4 Satellite Hardware	27
5 Communication Techniques	35
6 Multiple Access Techniques	44
7 Satellite Link Design Fundamentals	49

List of Scilab Codes

Exa 2.1	velocity of satellite	5
Exa 2.2	Orbit parameters	5
Exa 2.3	Orbit parameters	6
Exa 2.4	Orbit parameters	7
Exa 2.5	Orbit eccentricity	7
Exa 2.6	Satellite velocity at particular point	8
Exa 2.7	Satellite velocity at particular point	8
Exa 2.8	Satellite velocity at particular point	9
Exa 2.9	Orbital time period	9
Exa 2.10	Ratio of orbital time period	10
Exa 2.11	Orbit parameters	10
Exa 2.12	Time required to move between two points	11
Exa 2.13	velocity at apogee	11
Exa 2.14	velocity of satellite at particular point	12
Exa 2.16	Apogee distance	12
Exa 2.17	velocity of satellite	13
Exa 2.18	Apogee Distance	13
Exa 3.1	Inclination Angle	15
Exa 3.2	Velocity thrust	15
Exa 3.3	Velocity thrust	16
Exa 3.4	Velocity thrust	16
Exa 3.5	Velocity thrust	17
Exa 3.6	Velocity thrust	17
Exa 3.7	Velocity thrust	18
Exa 3.8	Maximum line of sight distance	19
Exa 3.9	line of sight distance	19
Exa 3.10	Inter satellite distance	20
Exa 3.11	Azimuth and elevation angle	20

Exa 3.12	Delay calculation	21
Exa 3.13	Angular spacing and intersatellite distance .	23
Exa 3.14	covered surface area	24
Exa 3.15	Area swept by ground track of satellite . . .	25
Exa 4.1	Ejection velocity	27
Exa 4.2	Mass of propellant to be burnt	27
Exa 4.3	Mass of propellant to be burnt	28
Exa 4.5	Required no of solar cells	28
Exa 4.6	Mass of battery system	29
Exa 4.7	Antenna Gain	30
Exa 4.8	Aperture efficiency and effective aperture . .	30
Exa 4.9	Directivity	31
Exa 4.10	null to null beam width	31
Exa 4.11	received signal strength	32
Exa 4.12	gain and beam width	33
Exa 4.13	beam width	33
Exa 4.14	phase angle	34
Exa 4.15	Earth station EIRP	34
Exa 5.1	Power Saving	35
Exa 5.2	Total Power in the modulated signal	36
Exa 5.3	Percentage power saving	36
Exa 5.4	Carrier Frequency	37
Exa 5.5	modulation parameters	37
Exa 5.6	maximum phase and frequency deviation . .	38
Exa 5.7	frequency deviation and Bandwidth	38
Exa 5.8	modulation index and Bandwidth	39
Exa 5.9	modulation index and Bandwidth	39
Exa 5.10	Daviation ratio and Bandwidth	40
Exa 5.11	Sampling level quantizing levels and no of bits	40
Exa 5.12	Nyquist rate	41
Exa 5.13	bits per sample and time duration for one bit	41
Exa 5.15	Sampling rate and sampling interval	42
Exa 5.16	Bit duration and Transmission Rate	42
Exa 6.1	TDMA frame	44
Exa 6.2	Frame efficiency	45
Exa 6.3	TDMA Frame	45
Exa 6.4	Doppler Shift	46
Exa 6.5	Chip Duration and chip rate	47

Exa 6.6	Maximum Permissible Doppler Effect	48
Exa 6.7	Noise reduction	48
Exa 7.1	Power received by the receiving antenna	49
Exa 7.2	free space path loss	49
Exa 7.3	Attenuation	50
Exa 7.4	Noise temperature and noise figure	51
Exa 7.5	overall noise figure	51
Exa 7.6	noise figure	52
Exa 7.7	Loss factor	53
Exa 7.8	System noise temperature	53
Exa 7.9	carrier to interface ratio	54
Exa 7.10	carrier to interface ratio	54
Exa 7.11	carrier to interface ratio	55
Exa 7.12	Longitudinal separation between two satellites	56
Exa 7.13	Gain per degree Kelvin	56
Exa 7.14	Gain per degree Kelvin	57
Exa 7.15	Link Margin	57

Chapter 2

Satellite orbits and Trajectories

Scilab code Exa 2.1 velocity of satellite

```
1 // Example 2.1, page no-36
2 clear
3 clc
4
5 r1=6370 //Earth's Orbit in km
6 r2=630 //Height of satellite from surface in
   km
7 G=6.67*10^-11 // Gravitational constant inNm^2/kg^2
8 M=5.98*10^24 //Mass of earth in kg
9
10 R=r1+r2
11
12 v=sqrt(G*M/(R*10^3))
13 printf("The velocity of sattelite %.2fkm/s",floor(v
   /10)*10^-2)
```

Scilab code Exa 2.2 Orbit parameters


```

1 // Example 2.2, page no-37
2 clear
3 clc
4
5 A=45000 //Apogee in km
6 P=7000 //Perigee in km
7 //(a)
8 a=(A+P)/2
9 //(b)
10 e=(A-P)/(2*a)
11 //(c)
12 e=(floor(e*100))/100
13 d=a*e
14
15 printf("(a)\nSemi-major axis of elliptical orbit is
        %d km",a)
16 printf("\n(b)\nEccentricity = %.2f",e)
17 printf("\n(c)\nThe distance between centre of earth
        and centre of ellipse is %d km ",d)

```

Scilab code Exa 2.3 Orbit parameters

```

1 // Example 2.3, page no-37
2 clear
3 clc
4
5 ma=42000 //Major axis distance in Km
6 P=8000 //Perigee distance in Km
7
8 A=ma-P
9 e=(A-P)/ma
10
11 printf("Apogee=%dkm\n Eccentricity=%.2f",A,e)

```

Scilab code Exa 2.4 Orbit parameters

```
1 // Example 2.4, page no-37
2 clear
3 clc
4
5 e=0.6 //Eccentricity
6 d=18000 //distance between earth's centre and centre
      of ellipse
7
8 a=d/e
9 A=a*(1+e)
10 P=a*(1-e)
11
12 printf("Semi-major axis of elliptical orbit is %d km
      \n Apogee distance=%dkm\n Perigee distance=%dkm",
      a,A,P)
```

Scilab code Exa 2.5 Orbit eccentricity

```
1 // Example 2.5, page no-38
2 clear
3 clc
4 AP_diff=30000 //difference between apogee and
      perigee in km
5 AP_sum=62800 //Apogee+perigee
6
7 E=AP_diff/AP_sum
8 printf("Orbit Eccentricity= %.3f",E)
```

Scilab code Exa 2.6 Satellite velocity at particular point

```
1 // Example 2.6, page no-38
2 clear
3 clc
4
5 R=7000*10^3 //sattelite orbit in m
6 mu=39.8*10^13 //constant G*M in Nm^2/kg
7 A=47000*10^3 //appogee distance in m
8 P=7000*10^3 //perigee distance in m
9 v=sqrt(mu/R)
10 a=(A+P)/2
11 v1=sqrt(mu*((2/R)-(1/a)))
12 printf("Velocity of satellite A at point X is v=%0.2
    fkm/s\n Velocity of satellite B at point X is V=%0.
    .3fkm/s",v/1000,v1/1000)//value in book is
    different at 3rd decimal place.
```

Scilab code Exa 2.7 Satellite velocity at particular point

```
1 // Example 2.7, page no-39
2 clear
3 clc
4
5 R=42000*10^3 //sattelite orbit in m
6 mu=39.8*10^13 //constant G*M in Nm^2/kg
7 A=42000*10^3 //appogee distance in m
8 P=7000*10^3 //perigee distance in m
9 v=sqrt(mu/R)
10 a=(A+P)/2
11 v1=sqrt(mu*((2/R)-(1/a)))
12 printf("Velocity of satellite A at point X is v=%0.3
    fkm/s\n Velocity of satellite B at point X is V=%0.
    .3fkm/s",v/1000,v1/1000)
```

Scilab code Exa 2.8 Satellite velocity at particular point

```
1 // Example 2.8, page no-40
2 clear
3 clc
4
5 R=25000*10^3 //sattelite orbit in m
6 mu=39.8*10^13 //constant G*M in Nm^2/kg
7 A=43000*10^3 //appogee distance in m
8 P=7000*10^3 //perigee distance in m
9 v=sqrt(mu/R)
10 a=(A+P)/2
11 v1=sqrt(mu*((2/R)-(1/a)))
12 printf("Velocity of satellite A at point X is v=%0.3
    fkm/s\n Velocity of satellite B at point X is V=%0
    .3fkm/s",v/1000,v1/1000)//value in book is
    different at 3rd decimal place.
```

Scilab code Exa 2.9 Orbital time period

```
1 // Example 2.9, page no-40
2 clear
3 clc
4
5 a=(50000/2)*10^3 //Semi-major axis in m
6 mu=39.8*10^13 //constant G*M in Nm^2/kg
7
8 T=2*3.14*sqrt((a^3)/mu)
9 h=T/(60*60)
10 x=modulo(T,3600)
11 m=x/60
12 s=modulo(x,60)
```

```

13 printf("Orbital time period is given by, T = %dsec\n
    \t\t\t\t\t = %dh %dm %ds",T,floor(h),floor(m),
    floor(s))//value in book is different for seconds
.

```

Scilab code Exa 2.10 Ratio of orbital time period

```

1 // Example 2.10, page no-42
2 clear
3 clc
4
5 a1=18000*10^3           //Semi-major axis for first
    satellite in m
6 a2=24000*10^3           //Semi-major axis for 2nd
    satellite in m
7
8 T2_by_T1=(a2/a1)^(3/2)
9 printf("Orbital time period of satellite 2 is %.2f
    times that of satellite 1",T2_by_T1)//value in
    book is different for seconds.

```

Scilab code Exa 2.11 Orbit parameters

```

1 // Example 2.11, page no-42
2 clear
3 clc
4
5 a=25000*10^3           //apogee distance in m
6 b=18330*10^3           //perigee distance in m
7 e=(sqrt(a^2-b^2)/a)
8 printf("Apogee distance = a(1+e)= %dkm\n Perigee
    distance = a(1-e)= %dkm\n",a*(1+e)/1000,ceil(a
    *(1-e)/1000))

```

Scilab code Exa 2.12 Time required to move between two points

```
1 // Example 2.12, page no-43
2 clear
3 clc
4
5 e=0.6 // eccentricity of elliptical
   orbit
6 a=0.97 // area of shaded region
7 b=2.17 //Area of non-shaded region
8 t=3 // time taken by satellite to
   move from pt B to A
9
10 x=b/a
11 y=x*t
12 printf("Time taken by satellite to move from A to B
   is %.3f hours ",y)
```

Scilab code Exa 2.13 velocity at apogee

```
1 // Example 2.13, page no-44
2 clear
3 clc
4
5 A=42000 //Apogee in km
6 P=8000 //Perigee in km
7 v_p=9.142 // velocity at perigee point
8 v_a=v_p*P/A
9 printf("Velocity at apogee = %.3f km/s",v_a)
```

Scilab code Exa 2.14 velocity of satellite at particular point

```
1 // Example 2.14, page no-44
2 clear
3 clc
4 theta=56.245 //angle made by direction of
   satellite with local horizontal
5 d=16000 //distance of particular point
6 P=8000 //Perigee in m
7 v_p=9.142 // velocity at perigee point
8
9 v=(P*v_p)/(d*floor(cos(theta*%pi/180)*1000)/1000)
10 printf("The velocity of satellite at that particular
   point is %.3f km/s",v)
```

Scilab code Exa 2.16 Apogee distance

```
1 // Example 2.16, page no-49
2 clear
3 clc
4
5 A1=12000 //first Apogee distance
6 P=8000 // Perigee distance
7 v1=1 // assume v1 as 1
8 v2=1.2*v1 //20% higher than v1
9
10 x=(v2/v1)^2
11 k=((1+(P/A1))/x)-1)
12 k=floor(k*10^4)/10^4
13 A2=P/k
14
15 printf("A2 = %.0fkm",ceil(A2))
```

Scilab code Exa 2.17 velocity of satellite

```
1 // Example 2.17, page no-50
2 clear
3 clc
4 vp=8 //horizontal velocity of satellite in
   km/s
5 r=1620 // distance from earth's surface in km
6 R=6380 // Earth's radius in km
7 d=10000 // distance of point at which velocity
   to be calculated
8 theta=30 // angle made by satellite with local
   horzon at that point
9
10 P=r+R
11 v=(vp*P)/(d*cos(theta*pi/180))
12 printf("v = %.2f km/s",v)
```

Scilab code Exa 2.18 Apogee Distance

```
1 // Example 2.18, page no-50
2 clear
3 clc
4
5 r=620 // distance from earth's surface in km
6 vp=8 //horizontal velocity of satellite at
   9000km height in km/s
7 R=6380 // Earth's radius in km
8 d=9000 // distance of point at which velocity
   to be calculated
9 theta=30 // angle made by satellite with local
   horzon at that point
10 mu=39.8*10^13 // Nm^2/kg
11
12 P=r+R
```



```
13 m=vp*d*cos(theta*pi/180)/P //m=sqrt((2mu/P)-[2mu
    /(A+P)])
14 m=(m*10^3)^2
15 x=(2*mu/(P*10^3))-m //x=[2mu/(A+P)]
16 x=floor(x/10^4)*10^4
17 k=(2*mu)/x //k=A+P
18 k=ceil(k/10^4)*10^4
19 A=k-(P*10^3)
20 printf("A = %.0f km",A/1000)
```

Chapter 3

Satellite Launch and In orbit Operations

Scilab code Exa 3.1 Inclination Angle

```
1 // Example 3.1, page no-72
2 clear
3 clc
4
5 Az=85 // Azimuth angle of injection point
6 l=5.2 // latitude of launch site
7 cosi=sin(Az*%pi/180)*cos(l*%pi/180)
8 i=acos(cosi)
9 i=i*180/%pi
10 printf("Inclination angle attained , i=%0.1 f ",i)
```

Scilab code Exa 3.2 Velocity thrust

```
1 // Example 3.2, page no-73
2 clear
3 clc
```

```

4
5 delta_i=7 //orbital plane inclination
6 V=3000 //velocity of satellite in
    circularized orbit
7 vp=2*V*sin(delta_i*pi/(2*180))
8 printf("Velocity thrust to make the inclination 0
    = %.0f m/s",vp)

```

Scilab code Exa 3.3 Velocity thrust

```

1 // Example 3.3, page no-73
2 clear
3 clc
4
5 mu=39.8*10^13 // Nm^2/kg
6 P=7000*10^3 // Perigee distance in m
7 e=0.69 // eccentricity of elliptical
    orbit
8 w=60/2 // angle made by line joing
    centre of earth and perigee e
    with the line of nodes
9 k=(e/sqrt(1+e))
10 k=floor(k*100)/100
11 v=2*(sqrt(mu/P))*k*sin(w*pi/180)
12 printf("The velocity thrust required to rotate the
    perigee point\n by desired amount is given by, v=
    %.1f m/s = %.3fkm/s",v,v/1000)

```

Scilab code Exa 3.4 Velocity thrust

```

1 // Example 3.4, page no-74
2 clear
3 clc

```

```

4
5 A=15000*10^3           // Original apogee distance
6 A1=25000*10^3         // Raised opogee distance
7 P=7000*10^3           // Perigee Distance
8 mu=39.8*10^13         //Nm^2/kg
9
10 A_d=A1-A
11 v=sqrt((2*mu/P)-(2*mu/(A+P)))
12 del_v=A_d*mu/(v*(A+P)^2)
13 printf("required Thrust velocity Delta_v = %.1f m/s"
        ,del_v)

```

Scilab code Exa 3.5 Velocity thrust

```

1 // Example 3.5, page no-75
2 clear
3 clc
4
5 A=15000*10^3           // Original apogee distance
6 A1=7000*10^3          // Raised opogee distance
7 P=7000*10^3           // Perigee Distance
8 mu=39.8*10^13         //Nm^2/kg
9
10 A_d=A-A1
11 v=sqrt((2*mu/P)-(2*mu/(A+P)))
12 del_v=A_d*mu/(v*(A+P)^2)
13 printf("required Thrust velocity Delta_v = %.1f m/s"
        ,del_v)

```

Scilab code Exa 3.6 Velocity thrust

```

1 // Example 3.6, page no-76
2 clear

```

```

3  clc
4
5  A=15000*10^3           //Original apogee distance
6  A1=16000*10^3         // Raised opogee distance
7  P=7000*10^3           // Perigee Distance
8  mu=39.8*10^13         //Nm^2/kg
9
10 A_d=A1-A
11 v=sqrt((2*mu/P)-(2*mu/(A+P)))
12 v=v*P/A
13 del_v=A_d*mu/(v*(A+P)^2)
14 printf("required Thrust velocity Delta_v = %.1f m/s"
        ,del_v)

```

Scilab code Exa 3.7 Velocity thrust

```

1  // Example 3.7, page no-77
2  clear
3  clc
4
5  R=6378*10^3           //Radius of earth
6  mu=39.8*10^13         //Nm^2/kg
7  r1=500*10^3           // original orbit from
        earths surface
8  r2=800*10^3           // orbit to be raised to
        thisdistance
9  R1=R+r1
10 R2=R+r2
11 delta_v=sqrt(2*mu*R2/(R1*(R1+R2)))-sqrt(mu/R1)
12 delta_v_dash=sqrt(mu/R2)-sqrt(2*mu*R1/(R2*(R1+R2)))
13
14 printf("Two thrusts to be applied are,\n Delta_v = %
        .2f m/s \n Delta_v_dash = %.2f m/s",delta_v,
        delta_v_dash)

```

Scilab code Exa 3.8 Maximum line of sight distance

```
1 // Example 3.8, page no-97
2 clear
3 clc
4
5 H=36000 //Height of geostationary satellite
   from the surface of earth
6 R=6370 // Radius of earth in km
7 k=acos(R/(R+H))
8 //k=k*180/%pi
9 k=sin(k)
10 k=ceil(k*1000)/1000
11 d=2*(H+R)*k
12 printf("Maximum line-of-sight distance is %.2f km",d
   )
```

Scilab code Exa 3.9 line of sight distance

```
1 // Example 3.9, page no-98
2 clear
3 clc
4
5 H=36000 //Height of geostationary satellite
   from the surface of earth
6 R=6370 // Radius of earth in km
7 theta=20 // angular separation between two
   satellites
8
9 D=(H+R)
10 k=ceil(cos(theta*%pi/180)*100)/100
11 d=sqrt(2*D^2*(1-k))
```

```
12
13 printf("The line-of-sight distance is %.4f km",d)
```

Scilab code Exa 3.10 Inter satellite distance

```
1 // Example 3.9, page no-98
2 clear
3 clc
4
5 //IntelSat-VI location= 37 W
6 // IntelSat-VII location=74 E
7 theta=37+74 // angular separation between two
   satellites
8 D=42164 //circular equilateral
   geostationary orbit in km
9 k=cosd(theta)
10 //printf("%f\n",k)
11 k=-0.357952
12 d=sqrt(2*D^2*(1-k))
13
14 printf("Inter-satellite distance is %.2f km",d)
```

Scilab code Exa 3.11 Azimuth and elevation angle

```
1 // Example 3.11, page no-99
2 clear
3 clc
4
5 theta_l=30 //earth station's location 30 W
   longitude
6 theta_s=50 //satellite's location 50 W
   longitude
```

```

7 theta_L=60           //earth station 's location 60 N
   latitude
8 r=42164             // orbital radius of the
   satellite in km
9 R=6378              //Earth 's radius in km
10
11 A_dash=atan((tand(theta_s-theta_l))/sind(60))
12 A_dash=A_dash*180/%pi
13 A=180+A_dash       //Azimuth angle
14
15 x=(180/%pi)*acos(cosd(theta_s-theta_l)*cosd(theta_L)
   )
16 y=r-ceil(R*(cosd(theta_s-theta_l)*cosd(theta_L)))
17 z=R*sind(x)
18 E=(atan(y/z)*180/%pi)-x
19 printf("Azimuth angle =%.1 f \n Elevation angle =%.
   .1 f ",A,E)

```

Scilab code Exa 3.12 Delay calculation

```

1 // Example 3.12, page no-100
2 clear
3 clc
4
5 theta_l=60           //earth station 's location 60 W
   longitude
6 theta_s=105          //satellite 's location 105 W
   longitude
7 theta_L=30           //earth station 's location 30 N
   latitude
8
9 theta_l1=90          //earth station 's location 90
   W longitude
10 theta_s1=105        //satellite 's location 105 W
   longitude

```



```

11 theta_L1=45           //earth station 's location 45
    N latitude
12
13 c=3*10^8             //speed of light
14 r=42164              // orbital radius of the
    satellite in km
15 R=6378               //Earth's radius in km
16
17 x=(180/%pi)*acos(cosd(theta_s-theta_l)*cosd(theta_L)
    )
18 y=r-ceil(R*(cosd(theta_s-theta_l)*cosd(theta_L)))
19 z=R*sind(x)
20 E=(atan(y/z)*180/%pi)-x
21
22 x1=(180/%pi)*acos(cosd(theta_s1-theta_l1)*cosd(
    theta_L1))
23 y1=r-ceil(R*(cosd(theta_s1-theta_l1)*cosd(theta_L1))
    )
24 z1=R*sind(x1)
25 E1=(atan(y1/z1)*180/%pi)-x1
26 E1=floor(E1)
27
28 //calculation of slant range dx
29 k=(R/r)*cosd(E)
30 k=(180/%pi)*asin(k)
31 k=k+E
32 k=sind(k)
33 k=ceil(k*1000)/1000
34 //k=k+E
35 //k=sin(k)
36 dx=(R)^2+(r)^2-(2*r*R*k)
37 dx=sqrt(dx)
38
39
40 //calculation of slant range dy
41 k1=(R/r)*cosd(E1)
42 k1=(180/%pi)*asin(k1)
43 k1=k1+E1

```

```

44 k1=floor(k1)
45 k1=sind(k1)
46 k1=ceil(k1*1000)/1000
47 dy=(R)^2+(r)^2-(2*r*R*k1)
48 dy=sqrt(dy)
49
50 tr=dy+dx
51 delay=tr*10^6/c
52 x=50
53 td=delay+x
54
55 printf(" Elevation angle , Ex =%.1 f " ,E)
56 printf("\n Elevation angle , Ey =%.1 f " ,floor(E1))
57 printf("\n Slant range dx of the earth station X is
    dx=%.2 fkm" ,dx)
58 printf("\n Slant range dy of the earth station Y is
    dy=%.1 fkm" ,dy)
59 printf("\n Therefore , total range to be covered is %
    .2 fkm" ,tr)
60 printf("\n propagation delay=%.2 fms" ,delay)
61 printf("\n\n Time required too transmit 500 kbs of
    information at \n a transmisssion speed of 10Mbps
    is given by 500000/10^7=%.0 fms" ,500000000/10^7)
62 printf("\n\n Total Delay= %.2 fms" ,td)

```

Scilab code Exa 3.13 Angular spacing and intersatellite distance

```

1 // Example 3.13 , page no-102
2 clear
3 clc
4
5
6 da=38000 // slant range of satellite
    A
7 db=36000 // slant range of satellite

```

```

      B
8  beeta=60          // difference between
   longitudes of two satellites
9  R=42164          // radius of the orbit of
   satellites
10
11
12  theta=(da^2+db^2-2*(R^2)*(1-cosd(beeta)))/(2*da*db)
13  theta=(180/%pi)*acos(theta)
14
15  d=sqrt(2*(R^2)*(1-cosd(beeta)))
16  printf("Angular spacing between two satellites
   viewed by earth station is,\n theta= %.1 f ",
   theta)
17  printf("\nInter-satellite distance , d=%.0 fkm",d)

```

Scilab code Exa 3.14 covered surface area

```

1  // Example 3.14, page no-107
2  clear
3  clc
4
5  r=42164          // orbital radius of the
   satellite in km
6  R=6378          //Earth's radius in km
7
8  //refer to Figure 3.53
9  //for E=0
10  alfa=asin(R/r)*(180/%pi)
11  alfa=floor(alfa*10)/10
12  theta=90-alfa
13  //in the right angle triangle OAC,
14  k=sind(alfa)
15  k=floor(k*1000)/1000
16  oc=R*k

```

```

17 oc=ceil(oc*10)/10
18 A=2*%pi*R*(R-oc)
19
20
21 //for E=10
22 E=10
23 alfa1=asin((R/r)*cosd(E))*(180/%pi)
24 //alfa1=ceil(alfa1*100)/100
25 theta1=90-alfa1-E
26 //in the right angle triangle OAC,
27 k1=sind(alfa1+E)
28 k1=floor(k1*1000)/1000
29 oc1=R*k1
30 oc1=floor(oc1*10)/10
31 A1=2*%pi*R*(R-oc1)
32
33 printf("for E=0 ,\n covered surface area is %.1f km
      ^2",A)
34 printf("\n\n for E=10 ,\n covered surface area is %
      .1f km^2",A1)

```

Scilab code Exa 3.15 Area swept by ground track of satellite

```

1 // Example 3.15, page no-108
2 clear
3 clc
4
5
6 theta=30 //satellite inclination to
      the equitorial plan
7 //the extreme latitudes covered in northern and
      southern hemisphere are the same as orbit
      inclination
8
9 printf("Extreme Northern latitude covered = %.0f N

```

```
    ",theta)
10 printf("\n Extreme Southern latitude covered = %.0
    f S",theta)
11 printf("\n\n In fact , the ground track would sweep\n
    all latitudes between %d N and %d S",theta,
    theta)
```

Chapter 4

Satellite Hardware

Scilab code Exa 4.1 Ejection velocity

```
1 // Example 4.1, page no-122
2 clear
3 clc
4
5
6 I=250 //specific impulse of a propellant
7 g=9.807 // acceleration due to gravity
8
9 v=I*g
10 printf("Ejection velocity of the propellant mass is ,
    v= %.2f m/s",v)
```

Scilab code Exa 4.2 Mass of propellant to be burnt

```
1 // Example 4.2, page no-122
2 clear
3 clc
4
```

```

5 m=4330          //initial mass of the satellite
6 i=290           //specific impulse of a propellant
7 del_v=-100      //velocity increment
8 g=9.807         // acceleration due to gravity
9
10 m1=m*(1-exp(del_v/(g*i)))
11 printf("Mass of propellant necessary to be burnt is ,
      m= %.0 fkg", ceil(m1))

```

Scilab code Exa 4.3 Mass of propellant to be burnt

```

1 // Example 4.3, page no-123
2 clear
3 clc
4
5 m=2950          //initial mass of the satellite
6 F=450           //required thrust
7 T=10            // thrust for time period
8 i=300           //specific impulse of a propellant
9 g=9.807         // acceleration due to gravity
10
11 mi=F*T/(i*g)
12 printf("Mass of propellant that would be consumed is
      , m=%.2 fkg", mi)

```

Scilab code Exa 4.5 Required no of solar cells

```

1 // Example 4.5, page no-134
2 clear
3 clc
4
5 p=2000          //electrical energy to be generated from
      solar panel in Watt

```

```

6 fi=1250      // solar flux falling normally to the
   solar cell in worst case
7 s=4*10^-4    // Area of each solar cell
8 e=0.15       //conversion efficiency of solar cell
   including the losses
9 theta=10     // angle made by rays of sun with
   normal
10
11 n=p/(fi*s*e)
12 n1=ceil(n)*%pi
13 n2=ceil(n1)/cosd(theta)
14 printf("Required no of solar cells , n = %.0f cells",
   ceil(n1))
15 printf("\n No of cells when sunrays are making an
   angle of 10   are %.0f",ceil(n2))

```

Scilab code Exa 4.6 Mass of battery system

```

1 // Example 4.6 , page no-134
2 clear
3 clc
4
5 p=3600        //Power required
6 t=1.2         //worst case eclipse period
7 c=90         //capacity of each cell in Ah
8 v=1.3        //voltage of each cell in V
9 d=0.8        // Depth of discharge
10 e=0.95      //Discharge efficiency
11 E_sp=60     //specific energy specification of the
   battery
12
13 energy=p*t
14 n=energy/(c*v*d*e)
15 E_b=energy/(d*e)
16 m=E_b/E_sp

```



```

17 printf("No of cells , n= %.0f cells\n Energy required
    to be stored in the battery system is %.1f Wh\n
    Mass of battery system = %.2f kg",n,E_b,m)

```

Scilab code Exa 4.7 Antenna Gain

```

1 // Example 4.7, page no-153
2 clear
3 clc
4
5 theta=0.5 //azimuth beam width=Elevation beam
    width
6 f=6*10^9 // operating frequency 6 Ghz
7 c=3*10^8 //speed of light in cm/s
8 theta_r=theta*pi/180
9 theta_r=ceil(theta_r*10^5)/10^5
10 A=4*pi/(theta_r^2)
11 A=ceil(A*100)/100
12 A_dB=10*log10(A)
13 lambda=c/f
14 Ag=(A*lambda^2)/(4*pi)
15
16 printf("\nGain in dB = %.2f dB \nAntenna gain
    expressed in terms of \nantenna aperture(A) is
    given by G = %.2f m^2",A_dB,Ag)

```

Scilab code Exa 4.8 Aperture efficiency and effective aperture

```

1 // Example 4.8, page no-153
2 clear
3 clc
4
5 la=0.5 //length efficiency in azimuth direction

```

```

6 le=0.7      //length efficiency in elevation
  direction
7 A=10        // Actual projected area of an antenna
8
9 Ae=1a*le
10 Aee=Ae*A
11 printf(" Aperture efficiency = %.2f \n Effective
  Aperture = %.1f m^2",Ae,Aee)

```

Scilab code Exa 4.9 Directivity

```

1 // Example 4.9, page no-154
2 clear
3 clc
4
5 p=100       //Antenna power in W
6 pd=10       //Power Density in mW/m^2
7 d=1000      //distance in m
8 p2=10000    // New antenna power
9 directivity=10*log10(p2/p)
10 printf(" Directivity (in dB)= %d dB",directivity)

```

Scilab code Exa 4.10 null to null beam width

```

1 // Example 4.10, page no-154
2 clear
3 clc
4
5 beam_w=0.4  //antenna's 3dB beam width
6 Ae=5        //Effective Aperture of Antenna
7
8 printf("The null-to-null beam width of a paraboloid
  reflector is twice its 3dB beam width. \n

```

Therefore , Null-to-null beam width = $\% .1 f \quad , 2 * \text{beam_w}$

Scilab code Exa 4.11 received signal strength

```
1 // Example 4.11 , page no-154
2 clear
3 clc
4 d=20 //received signal strength in dB
5 loss=3 //incident polarization is circular and
        antenna is circularly polarized
6 theta=60 //received wave making angle with
           horizontal
7 total=d+loss
8 los=d*log10(1/cosd(theta))
9
10 printf("(a)\n When received polarization is same as
        antenna \n polarization , the polarization loss is
        zero.\n Therefore , received signal strength = %ddB"
        ,total)
11 printf("\n\n(b)\n When the incident wave is
        vertically polarized ,\n the angle between
        incident polarization and antenna polarization is
        90 \n Hence , Polarization loss = infinity\n
        received signal strength = 0")
12 printf("\n\n(c)\n When incident wave is left-hand
        circularly polarized\n and antenna polarization
        is linear ,\n then there is polarization loss of
        %ddB and\n received signal strength is %ddB" ,loss
        ,d)
13 printf("\n\n(d)\n Polarization loss = %ddB \n
        Received signal strength = %ddB" ,los ,ceil(total-
        los))
```

Scilab code Exa 4.12 gain and beam width

```
1 // Example 4.12, page no-155
2 clear
3 clc
4 Ea=1 //effective aperture
5 f=11.95*10^9 //downlink operating frequency
6 c=3*10^8 //speed of light
7
8 Ae=floor((%pi*1000*Ea^2)/4)/1000
9 lamda=floor(c*1000/f)/1000
10 ag=floor(100*4*%pi*Ae/lamda^2)/100
11 adb=floor(100*10*log10(ag))/100
12 width=70*lamda/Ea
13 printf("Operating wavelength = %.3fm\n Antenna Gain
    = %.2f\n Antenna Gain in dB = %.2fdB\n 3dB beam
    width = %.2 f ",lamda,ag,adb,width)
```

Scilab code Exa 4.13 beam width

```
1 // Example 4.13, page no-155
2 clear
3 clc
4 f=2.0 //reflector focal length
5 d=2.0 // reflector diameter
6 l=90/100 //90% of the angle
7 theta=4*(atand(1/(4*f/d)))
8 theta=4*atand(0.25007) // this value gives exact
    answer as in book
9 dbw=l*theta
10 printf("The angle subtended by the focal point feed\
    n at the edges of the reflector is , theeta = %.2
```

```
f \n\n 3dB beam width = %.2 f \n null-to-null
beam width = % .2 f ",theta,dbw,floor(200*dbw)
/100)
```

Scilab code Exa 4.14 phase angle

```
1 // Example 4.14, page no-155
2 clear
3 clc
4 c=3*10^8 //speed of light
5 f=2.5*10^9 //operating frequency
6 s=0.1 //inter element spacing
7 theta =10 //10 right towards array axis
8 l=c/f
9 fi=(360*s/l)*ceil(10000*sind(theta))/10000
10 fi=ceil(10*fi)/10
11 printf("The phase angle for elements 1,2,3,4 and 5 \
n are respectively 0 ,%.1 f ,%.1 f ,%.1 f and %
.1 f ",fi,2*fi,3*fi,4*fi)
```

Scilab code Exa 4.15 Earth station EIRP

```
1 // Example 4.15, page no-156
2 clear
3 clc
4
5 p=10000 //power fed to the antenna in W
6 ag=60 //Antenna gain
7 loss=2 //Power lossin feed system
8 adb=10*log10(p)
9 EIRP=adb+ag-loss
10 printf("Earth station EIRP = %ddb",EIRP)
```

Chapter 5

Communication Techniques

Scilab code Exa 5.1 Power Saving

```
1 // Example 5.1, page no-174
2 clear
3 clc
4 //for case (a)
5 m=0.5 //modulation index
6 //for AM
7 pt1=(1+(m^2)/2)
8
9 //for SSBSC
10 pt2=(m^2)/4
11
12 //% power saving
13 p=(pt1-pt2)*100/pt1
14 p=floor(p*10)/10
15 printf("Percentage power saving is %.1f%%",p)
16
17 //for case (b)
18 m=1 //modulation index
19 //for AM
20 pt1=(1+(m^2)/2)
21
```

```

22 //for SSBSC
23 pt2=(m^2)/4
24
25 //% power saving
26 p=(pt1-pt2)*100/pt1
27 p=floor(p*10)/10
28 printf("\n Percentage power saving is %.1f%%",p)

```

Scilab code Exa 5.2 Total Power in the modulated signal

```

1 // Example 5.2, page no-174
2 clear
3 clc
4
5 pc=500 //energy of carrier signal
6 m=0.6 //AM modulation index
7 //for (a)
8 pt=pc*(1+(m^2)/2)
9
10 //for (b)
11 pt2=pc*(m^2)/4
12
13 printf("(a)\n A3E is the double side band AM with
    full carrier.\n Therefore, Pt= %.0f W\n\n (b)\n
    J3E is an SSBSC system.\n Therefore, Pt= %.0f W",
    pt,pt2)

```

Scilab code Exa 5.3 Percentage power saving

```

1 // Example 5.3, page no-175
2 clear
3 clc
4

```

```

5 m=0.6           //60% modulation
6 //for A3E
7 pt1=(1+(m^2)/2)
8
9 //for J3E
10 pt2=(m^2)/4
11
12 //% power saving
13 p=(pt1-pt2)*100/pt1
14 p=ceil(p*10)/10
15 printf("Percentage power saving is %.2f%%",p)

```

Scilab code Exa 5.4 Carrier Frequency

```

1 // Example 5.4, page no-175
2 clear
3 clc
4
5 //multiplication of two signals gives AM with
   frequency component( $w_c-w_m$ ) and ( $w_c+w_m$ ) and its BW
   is  $2w_m$ 
6 bw=0.5/100      //bw is 0.5% of carrier freq.
7 wc=2/bw
8 printf("Wc = %.0fWm",wc)

```

Scilab code Exa 5.5 modulation parameters

```

1 // Example 5.5, page no-190
2 clear
3 clc
4
5 //comparing given equation with stanard equation
6 m=6           //Modulation Index

```



```

7  wc=7.8*10^8           //unmodulated carrier frequency
8  wm=1450              //Modulating frequency
9
10 fc=wc/(2*pi)
11 fm=wm/(2*pi)
12 printf("Unmodulated carrier frequency , fc = %.2f MHz
    \n The modulation index m = %d \n Modulating
    frequency , fm = %.2f Hz",fc/10^6,m,fm)

```

Scilab code Exa 5.6 maximum phase and frequency deviation

```

1 // Example 5.6 , page no-190
2 clear
3 clc
4
5 printf("comparing given equation with stanard
    equation , we have,\n Maximum phase deviation = 6
    radian\n Maximum frequency deviation =
    12*3.14*10^3 radian/s = 6 KHz")

```

Scilab code Exa 5.7 frequency deviation and Bandwidth

```

1 // Example 5.7 , page no-191
2 clear
3 clc
4
5 //comparing given equation with stanard equation
6 mf=150                //modulation index
7 fm=1                  // modulating frequency in KHz
8 fd=mf*fm
9 bw=2*(mf+1)*fm
10 printf("frequency deviation = %.0f kHz\n Bandwidth =
    %.0f kHz \n\n Expression for instantaneous

```

frequency is given by, \n $f = 10^8 - 150 \cdot (10^3) \cdot \sin(2 \cdot 3.14 \cdot 10^3 \cdot t)$ ”, fd, bw)

Scilab code Exa 5.8 modulation index and Bandwidth

```
1 // Example 5.8, page no-191
2 clear
3 clc
4
5
6 fd=50           //frequency deviation in kHz
7 fm=1           //modulating frequency in kHz for
   case 1
8 fm2=100        //modulating frequency in kHz for
   case 2
9
10 //for casse 1
11 m=fd/fm
12 bw=2*(m+1)*fm
13 //for case 2
14 m2=fd/fm2
15 bw2=2*(m2+1)*fm2
16
17 printf("For first case\n Modulation index = %.0f \n
   Bandwidth = %.0f kHz \n\n For second case\n
   Modulation index = %.1f \n Bandwidth = %.0f kHz",
   m, bw, m2, bw2)
```

Scilab code Exa 5.9 modulation index and Bandwidth

```
1 // Example 5.9, page no-192
2 clear
3 clc
```

```

4
5 bw=20*10^3      //bandwidth in Hz
6 fm=1* 10^3     //modulating frequency in Hz
7 mf=(bw/(2*fm))-1
8 new_mf=mf*6
9 new_fm=0.5     //kHz
10 new_bw=2*(new_mf+1)*new_fm
11 printf("mf=%0.0f\n New modulation index = %0.0f\n New
    bandwidth = %0.0f kHz",mf,new_mf,new_bw)

```

Scilab code Exa 5.10 Daviation ratio and Bandwidth

```

1 // Example 5.10, page no-192
2 clear
3 clc
4
5 fd=75          //Maximum allowed frequency deviation in
    kHz
6 fm=15         //Highest modulating frequency in kHz
7
8 D=fd/fm
9 bw=2*(D+1)*fm
10 printf("Deviation Ratio , D = %0.0f\n Bandwidth = %0.0f
    kHz",D,bw)

```

Scilab code Exa 5.11 Sampling level quantizing levels and no of bits

```

1 // Example 5.11, page no-199
2 clear
3 clc
4
5 fm=3200        //highest frequency component in
    message signal

```

```

6 k=48000           //channel capacity in b/s
7 fs=2*fm
8 n=k/fs
9 n=floor(n)
10 printf("n = %.0f\n L = 2^7 = %.0f\n fs = %.3f kHz",n
        ,2^7,(k/7)/1000)

```

Scilab code Exa 5.12 Nyquist rate

```

1 // Example 5.12, page no-199
2 clear
3 clc
4
5 //rearranging equation and comparing it with standard
   equation we have,
6 f=2500           //Highest frequency component in
   the signal in Hz
7 printf("Nyquist rate = 2 x f\n\t      = %.0f Hz = %
   .0f kHz",2*f,2*f/1000)

```

Scilab code Exa 5.13 bits per sample and time duration for one bit

```

1 // Example 5.13, page no-199
2 clear
3 clc
4
5 l=128           //no of Quantizing levels
6 fs=10000       //sampling frequency in Hz
7 n= log2(l)
8 t=1/(n*fs)
9 printf("Number of bits per sample (n) = %.0f\n Time
   duration of one bit of binary encoded signal is
   %.3f micro second",n,t*10^6)

```

Scilab code Exa 5.15 Sampling rate and sampling interval

```
1 //Example 5.15, page no-208
2 clear
3 clc
4
5 f1=2.4           //first signal frequency
6 f2=3.2           //2nd signal frequency
7 f3=3.4           //3rd signal frequency
8
9 //minimum sampling rate for each of the signals
   would be twice the highest frequency component
10
11
12 sr=3*(f3*2)
13 st=10^6/(sr*10^3)
14 printf("Sampling rate of the composite signal = %.1f
   kHz \n Sampling interval of the composite signal
   = %.0f micro second",sr,st)
```

Scilab code Exa 5.16 Bit duration and Transmission Rate

```
1 //Example 5.16, page no-209
2 clear
3 clc
4
5 bw=3.2           // voice channel band limited
   frequency in kHz
6 r=1.2           //1.2 times the Nyquist rate
7 n=24           //no of voice channel
8 b=8           // 8-bit PCM
```

```
9 sr=2*bw*r
10 p=10^6/(sr*10^3)
11 N=(n*b)+1
12 bit_d=p/N
13 bit_d=ceil(bit_d*1000)/1000
14 tr=1/bit_d
15
16 printf("Number of bits in each frame = %.0f \n Bit
duration = %.3f micro second \n Transmission rate
= %.3f Mbps",N,bit_d,ceil(tr*1000)/1000)
```

Chapter 6

Multiple Access Techniques

Scilab code Exa 6.1 TDMA frame

```
1 // Example 6.1, page no-230
2 clear
3 clc
4 t=20 //TDMA frame length in ms
5 lc=352 //length of carrier and clock recovery
      frequency in bits
6 lu1=48 //length of unique word in bits
7 lo=510 //length of order wire channel in bits
8 lm= 256 //length of management channel in bits
9 lt=320 // length of transmit timing channel in
      bits
10 ls1=24 // length of service channel in bits
11 gt=64 // Guard time in bits
12 rb=2 // reference burst
13
14 lr=lc+lu1+lo+lm+lt
15 tb=lc+lu1+lo+ls1
16 tob=(lr*rb)+(tb*t)+((t+rb)*gt)
17 printf("(a)\nThe length of reference burst(from
      given data) is %d bits\n\n(b)\nThe length of
      traffic burst preamble(from given data)is %d bits
```

```
\n\n(c)\nTotal number of overhead bits is %d bits
",lr,tb,tob)
```

Scilab code Exa 6.2 Frame efficiency

```
1 // Example 6.2, page no-230
2 clear
3 clc
4 t=20 //TDMA frame length in ms
5 lc=352 //length of carrier and clock recovery
    frequency in bits
6 lu1=48 //length of unique word in bits
7 lo=510 //length of order wire channel in bits
8 lm= 256 //length of management channel in bits
9 lt=320 // length of transmit timing channel in
    bits
10 ls1=24 // length of service channel in bits
11 gt=64 // Guard time in bits
12 rb=2 // reference burst
13 br=90*10^6 //burst bit rate 90Mbps
14
15 bfr=br*t*10^-3
16 lr=lc+lu1+lo+lm+lt
17 tb=lc+lu1+lo+ls1
18 tob=(lr*rb)+(tb*t)+((t+rb)*gt)
19 feff=(bfr-tob)*100/bfr
20 feff=ceil(feff*100)/100
21 printf("Frame efficiency = %.2f%%",feff)
```

Scilab code Exa 6.3 TDMA Frame

```
1 // Example 6.2, page no-230
2 clear
```



```

3  clc
4  t=20          //TDMA frame length in ms
5  lc=352        //length of carrier and clock recovery
                frequency in bits
6  lu1=48        //length of unique word in bits
7  lo=510        //length of order wire channel in bits
8  lm= 256       //length of management channel in bits
9  lt=320        // length of transmit timing channel in
                bits
10 ls1=24        // length of service channel in bits
11 gt=64         // Guard time in bits
12 rb=2          // reference burst
13 br=90*10^6    //burst bit rate 90Mbps
14 dr= 64*10^3   //data rate 64 kbps
15 bfr=br*t*10^-3
16 lr=lc+lu1+lo+lm+lt
17 tb=lc+lu1+lo+ls1
18 tob=(lr*rb)+(tb*t)+((t+rb)*gt)
19 feff=(bfr-tob)*100/bfr
20 feff=ceil(feff*100)/100
21 vsb=dr*t*10^-3
22 x=bfr*feff/100
23 printf("The number of bits in a frame for a voice
        sub-burst is %d\n\n The total no of bits
        available in a frame for carrying traffic is %d\n
        \n Maximum no of PCM voice channels in a frame is
        %d channels",vsb,x,x/vsb)

```

Scilab code Exa 6.4 Doppler Shift

```

1  // Example 6.4, page no-231
2  clear
3  clc
4
5  R=42150        //orbital radius of satellite

```

```

6 oi=0.25/100 // orbit inclination
7 acc=0.3 //error of 0.3 degree
8 c=3*10^8 // speed of light
9 x=oi*R
10 x=ceil(x*10)/10
11 y=R*2*pi*acc/360
12 y=ceil(y*10)/10
13 z=sqrt(x^2+y^2)
14 z=ceil(z*10)/10
15 delay=z*10^6/c
16 delay=floor(delay*1000)/1000
17 pd=2*delay
18 printf("variation in altitude caused by orbit
    inclination = %.1fkm\n variation due to station-
    keeping error of 0.3 = %.1fkm",x,y)
19 printf("\n Both these errors will introduce a
    maximum range variation of %.1fkm\n This cause a
    one-way propagation delay of %.3fms\n Round trip
    propagation delay =%.2fms\n Dopler Shift = %.2f
    ms in 8h=56.25 ns/s",z,delay,delay*2,pd)

```

Scilab code Exa 6.5 Chip Duration and chip rate

```

1 // Example 6.5, page no-238
2 clear
3 clc
4
5 de=40 //Doppler effect variation due to
    station-keeping errors in ns/s
6 d=280 //Sttelite round trip delay in ms
7 c=20/100 // DS-CDMA signals should not exceed 20
    % of the chip duration
8
9 te=de*10^-9*d*10^-3
10 tc=te/c

```

```
11 printf("Chip Duration , Tc = %.0f ns \n This gives
    maximum chip rate as (1/56)Gbps = 1000/56 Mbps =
    %.3f Mbps",tc*10^9,1000/56)
```

Scilab code Exa 6.6 Maximum Permissible Doppler Effect

```
1 // Example 6.6, page no-238
2 clear
3 clc
4
5 cr=25 //Chip rate is 25 Mbps
6 c=20/100 // DS-CDMA signals should not exceed
    20% of the chip duration
7 d=1000/cr //chip duration in ns
8 tr=c*d
9 x=tr/(280*10^-3)
10 printf("The maximum allowable timing error per
    satellite round trip is %.0f ns\n This %.0f ns
    error is to occur in 280 ms.\n Therefore, maximum
    permissible Doppler effect variation is %.2f ns/s
    ",tr,cr,x)
```

Scilab code Exa 6.7 Noise reduction

```
1 // Example 6.7, page no-238
2 clear
3 clc
4 cr=20*10^6 //chip rate in Mbps
5 ir= 20*10^3 //information bit rate
6 g=10*log10((cr)/(ir))
7 printf("Noise reduction achhievable = Processing
    gain = %.0f dB",g)
```

Chapter 7

Satellite Link Design Fundamentals

Scilab code Exa 7.1 Power received by the receiving antenna

```
1 // Example 7.1, page no-249
2 clear
3 clc
4
5 d=36000 *10^3 //distance of geostationary satellite
   from earth's surface
6 Gt=100 // Antenna gain of 20dB
7 Pt=10 // Power radiated by earth station
8
9 Prd=Pt*Gt/(4*%pi*d^2)
10 printf("Prd = %.4f * 10 ^-12 W/m^2\n Power received
   by the receiving antenna is given by Pr = %.3f
   pW", Prd*10^12, Prd*10^13)
```

Scilab code Exa 7.2 free space path loss

```

1 // Example 7.2 , page no-262
2 clear
3 clc
4
5 c=3*10^8 //speed of light
6 R=10000 //path length
7 f=4 // operating frequency in GHz
8 EIRP=50 //in dB
9 gr=20 //antenna gain in dB
10 rp=-120 // received power in dB
11 //(a)
12 lamda=c/(f*10^9)
13 pl=20*log10(4*pi*R/lamda)
14
15 //(b)
16 Lp=EIRP+gr-rp
17 printf("(a)\n Operating wavelength = %.3f m\n Path
    loss (in dB) = %.2f dB",lamda,pl)
18 printf("\n\n (b)\n Path loss = %.0fdB",Lp)

```

Scilab code Exa 7.3 Attenuation

```

1 // Example 7.3 , page no-262
2 clear
3 clc
4
5 p=75 // rotation of plane of polarization
6
7 //Polarization rotation is inversely proportional to
    square of the operating frequency
8 f= 5 //frequency increased by factor
9 x=f^2 //rotation angle will decrease by a
    factor of 25
10 p_ex=p/x
11 Apr=-20*log10(cosd(p))

```

```

12 Apr2=-20*log10(cosd(p_ex))
13 printf("For polarization mismatch angle = 75 \n
    Attenuation = %.2f dB",Apr)
14 printf("\n\n For polarization mismatch angle = 3
    \n Attenuation = %.3f dB",Apr2)

```

Scilab code Exa 7.4 Noise temperature and noise figure

```

1 // Example 7.4, page no-270
2 clear
3 clc
4
5 g1=30 //gain of RF stage in dB
6 t1=20 //Noise temperature in K
7 g2=10 //down converter gain in dB
8 t2=360 //noise temperature in K
9 g3=15 //gain of IF stage in dB
10 t3=1000 //noise temperature in K
11 t=290 //reference temperature in K
12
13 G1=1000 //30 dB equivalent gain
14 Te=t1+(t2/G1)+t3/(G1*g2)
15 F=1+Te/t
16 printf("Effective noise temperature, Te = %.2fK",Te)
17 printf("\n\n System Noise Figure, F = %.2f",F)

```

Scilab code Exa 7.5 overall noise figure

```

1 // Example 7.5, page no-271
2 clear
3 clc
4
5 g1=30 //gain of RF stage in dB

```

```

6 t1=20           //Noise temperature in K
7 g2=10           //down converter gain in dB
8 t2=360          //noise temperature in K
9 g3=15           //gain of IF stage in dB
10 t3=1000        //noise temperature in K
11 t=290          //reference temperature in K
12
13 G1=1000        //30 dB equivalent gain
14 //Te=t1+(t2/G1)+t3/(G1*g2)
15 F1=1+t1/t
16 F2=1+t2/t
17 F3=1+t3/t
18
19 F=F1+((F2-1)/G1)+(F3-1)/(G1*g2)
20 printf("Noise Figure specifications of the three
        stages are as follow ,\n\n F1 = %.3f\n F2 = %.2f\n
        F3 = %.2f",F1,F2,F3)
21 printf("\n\n The overall noise figure is , F = %.2f",
        F)

```

Scilab code Exa 7.6 noise figure

```

1 // Example 7.6 , page no-272
2 clear
3 clc
4
5 L=1.778          //Loss factor of the feeder 2.5dB
        equivalent
6 ts=30           //Noise temperature of satellite
        receiver in K
7 t=50            //Noise temperature in K
8 ti=290          // reference temperature in K
9 x=t/L
10 y=ti*(L-1)/L
11 Te=x+y+ts

```

```

12 F1=1+(ts/ti)
13 F2=1+(Te/ti)
14 printf("contribution of antenna noise temperature
    when\n referred to the input of the receiver is %
    .1f K",x)
15 printf("\n\n Contribution of feeder noise when
    referred to the\n input of the receiver is %.1f",
    y)
16 printf("\n\n1. Noise figure in first case = %.3f = %
    .3f dB",F1,10*log10(F1))//answer in book is
    different 0.426dB
17 printf("\n\n2. Noise figure in second case = %.3f =
    %.2f dB",F2,10*log10(F2))

```

Scilab code Exa 7.7 Loss factor

```

1 // Example 7.7, page no-272
2 clear
3 clc
4
5 Ta= 40 //Antenna Noise temperature
6 Ti=290 //Reference temperature in K
7 T=50 // Effecitve input noise temperatuire
8 Tf=Ti
9 L=(Ta-Tf)/(T-Tf)
10 L=ceil(L*10^4)/10^4
11 printf("Loss factor = %.4f = %.3f dB",L,10*log10(L))

```

Scilab code Exa 7.8 System noise temperature

```

1 // Example 7.8, page no-273
2 clear
3 clc

```



```

4
5 Ta=50           //Antenna Noise temperature
6 Tf=300         //Thermodynamic temperature of the
   feeder
7 Te=50          // Effecitve input noise temperatuire
8
9 //(a)
10 Lf=1
11 T=(Ta/Lf)+(Tf*(Lf-1)/Lf)+Te
12 printf("(a)\n System noise temperature = %.0fK",T)
13
14 //(b)
15 Lf=1.413
16 T=(Ta/Lf)+(Tf*(Lf-1)/Lf)+Te
17 printf("\n\n (b)\n System noise temperature = %.3fK"
   ,ceil(T*10^3)/10^3)

```

Scilab code Exa 7.9 carrier to interface ratio

```

1 // Example 7.9, page no-278
2 clear
3 clc
4 e=35           //EIRP radiated by satellite in dBW
5 g=50           //receiver antenna gain in dB
6 e1=30          //EIRP of interfacing satellite in dBW
7 theeta=4       //line-of-sight between earth station and
   interfacing sattelite
8
9 x=(e-e1)+(g-32+25*log10(theeta))
10 printf("carrier-to-interface (C/I) = %.2f dB",x)

```

Scilab code Exa 7.10 carrier to interface ratio

```

1 // Example 7.10, page no-279
2 clear
3 clc
4
5 ea=80 //EIRP value of earth station A in dBW
6 eb=75 //EIRP value of earth station B in dBW
7 g=50 //transmit antenna gain in dB
8 gra=20 //receiver antenna gain for earth station
   A in dB
9 grb=15 //receiver antenna gain for earth station
   B in dB
10 theeta=4 //viewing angle of the sattelite from two
   earth station
11 eirp_d=eb-g+32-25*log10(theeta)
12 c_by_i=ea-eirp_d+(gra-grb)
13 printf("carrier-to-interference ratio at the
   satellite due to\n inteference caused by Eart
   station B is, (C/I) = %.0f dB ",c_by_i)

```

Scilab code Exa 7.11 carrier to interface ratio

```

1 // Example 7.11, page no-279
2 clear
3 clc
4
5 //carrier sinal strength at sattelite by uplink
6 u=10000 // equivalent to 40dB
7
8 //carrier sinal strength at eart station by downlink
9 d=3162.28 //equivalent to 35dB
10
11 x=1/((1/u)+(1/d))
12 printf("Total carrier-to-interference ratio is %.2f
   = %.1f dB",x,10*log10(x))

```

Scilab code Exa 7.12 Longitudinal separation between two satellites

```
1 //Example 7.12, Page no.280
2 clear
3 clc
4
5 theeta=5 //Angle form by slant ranges of two
           satellites
6 dA=42100*10^3 //Slant range of satellite A
7 dB=42000*10^3 //Slant range of satellite B
8 r=42164*10^3 //radius of geostationary orbit
9 beeta=((dA^2+dB^2-cosd(theeta)*2*dA*dB)/(2*r^2))
10 beeta=ceil(beeta*10^3)/10^3
11 beeta=(180/%pi)*acos(1-beeta)
12 printf("Longitudinal separation between two
           satellites is %.3 f ",beeta)
```

Scilab code Exa 7.13 Gain per degree Kelvin

```
1 //Example 7.13, Page no.281
2 clear
3 clc
4 Ga=60 //Antenna Gain in dB
5 Ta= 60 //Noise teperature of Antenna
6 L1=1.12 //Feeder Loss equivalent to dB
7 T1=290 //Noise teperature of stage 1
8 G2=10^6 //Gain of stage 2 in dB
9 T2=140 //Noise teperature of stage 2
10 T3=10000 //Noise teperature of stage 3
11 G=Ga-0.5 // input of low noise amplifier
12 Ts=(Ta/L1)+(T1*(L1-1)/L1)+T2+(T3/G2)
13 Ts=floor(Ts*100)/100
```

```

14 x=G-10*log10(Ts)
15 printf("Tsi = %.2fK\n\n G/T(in dB/K)= %.0f dB/K",Ts,
        x)

```

Scilab code Exa 7.14 Gain per degree Kelvin

```

1 //Example 7.14, Page no.282
2 clear
3 clc
4
5 Ga=60 //Amplifier Gain in dB
6 Ta= 60 //Noise teperature of Antenna
7 L1=1.12 //Feeder Loss equivalent to dB
8 T1=290 //Noise teperature of stage 1
9 G2=10^6 //Gain of stage 2 in dB
10 T2=140 //Noise teperature of stage 2
11 T3=10000 //Noise teperature of stage 3
12 G=Ga-0.5 // input of low noise amplifier
13
14 T=Ta+T1*(L1-1)+L1*(T2+(T3/G2))
15 x=G-10*log10(T)
16 printf("T = %.1fK\n\n G/T = %.0f dB/k",T,ceil(x))
17 printf("\n\n It is evident from the solutions of the
        problems 13 and 14\n that G/T ratio is invarient
        regardless of the reference point in agreement \
        n with a statement made earlier in the text.")

```

Scilab code Exa 7.15 Link Margin

```

1 //Example 7.15, Page no.286
2 clear
3 clc
4

```

```

5 f=6*10^9 //uplink frequency
6 eirp= 80 //Earth station EIRP in dBW
7 r=35780 //Earth station satellite distance
8 l=2 //attenuation due to atomospheric
  factors in dB
9 e=0.8 // satellite antenna's aperture
  efficiency
10 a=0.5 // satellite antenna's aperture area
11 T=190 // Satellite receiver's effective noise
  temperature
12 bw=20 *10^6 //Satellite receiver's bandwidth
13 cn=25 // received carrier-to-noise ratio in dB
14 c=3*10^8 //speed of light
15
16 k=1.38*10^-23
17 lamda=c/f
18 G=e*4*%pi*a/lamda^2
19 G=ceil(G*100)/100
20 Gd=10*log10(G)
21 p=10*log10(k*T*bw)
22 pl=20*log10(4*%pi*r*10^3/lamda)
23 rp=eirp-l-pl+Gd
24 rp=floor(rp*100)/100
25 rc=floor((rp-p)*100)/100
26 lm=rc-cn
27 printf(" Satellite Antenna gain , G = %.2f = %.2f dB \
  n Receivers Noise Power = %.1f dB\n free-space
  path loss = %.2f dB \n received power at
  satellite = %.2f dB \n receiver carrier = %f is
  stronger than noise.\n It is %.2f dB more than
  the required threshold value.\n Hence, link
  margin = %.2f dB" ,G,Gd,p,pl,rp,rc,lm,lm)

```
