Scilab Textbook Companion for Optical Fiber Communications - Principles And Practice by J. M. Senior[1]

Created by
Mohammad Faisal Siddiqi
B.TECH (ELECTRONICS & COMMUNICATION)
Electronics Engineering
JAMIA MILLIA ISLAMIA, NEW DELHI
College Teacher
Not Decided
Cross-Checked by

July 31, 2019

[1]Funded 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: Optical Fiber Communications - Principles And Practice

Author: J. M. Senior

Publisher: Pearson Education, New Delhi

Edition: 2

Year: 2007

ISBN: 81-317-0309-6

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

List of Figures

Chapter 2

OPTICAL FIBER WAVEGUIDES

Scilab code Exa 2.1 Determination of Critical Angle NA and Acceptance Angle

```
//Example 2.1
//Program to determine the following:
//(a) Critical angle at the core-cladding interface
//(b) NA for the fiber
//(c) Acceptance angle in air for the fiber

clear;
clc ;
close ;

//Given data
n1=1.50;            //CORE REFRACTIVE INDEX
n2=1.47;            //CLADDING REFRACTIVE INDEX

//(a) Critical angle at the core-cladding interface
    in degrees
PHIc=asin(n2/n1)*180/%pi;

//(b) NA for the fiber
```

```
NA=sqrt(n1*n1-n2*n2);

//(c) Acceptance angle in air for the fiber in degrees
THEETAa=asin(NA)*180/%pi;

//Displaying The Results in Command Window
printf("\n\n\t Critical angle at the core-cladding interface is %0.1f degrees.",PHIc);
printf("\n\n\t NA for the fiber is %0.2f.",NA);
printf("\n\n\t Acceptance angle in air for the fiber is %0.1f degrees.",THEETAa);
```

Scilab code Exa 2.2 Determination of NA Solid Acceptance Angle and the Critical An

```
//Example 2.2
//Program to calculate
//(a) NA
//(b) Solid Acceptance Angle
//(c) Critical Angle at the core-cladding interface

clear;
clc ;
close ;

//Given data
n1=1.46;                    //CORE REFRACTIVE INDEX
delta=0.01;                 //RELATIVE REFRACTIVE INDEX DIFFERENCE

//Numerical Aperture
NA=n1*sqrt(2*delta);

//Solid Acceptance Angle in radians
zeta=%pi*(NA)^2;
```

```
//Critical Angle at the core-cladding interface in
   degrees
n2=n1*(1-delta);
PHI_c=asin(n2/n1)*180/%pi;

//Displaying the Results in Command Window
printf("\n\n\t The Numerical Aperture for the fiber
   is %0.2f.",NA);
printf("\n\n\t The Solid Acceptance Angle for the
   fiber is %0.2f radians.",zeta);
printf("\n\n\t The Critical Angle at the core-
   cladding interface for the fiber is %0.1f degrees
   .",PHI_c);
```

Scilab code Exa 2.3 Comparision of Acceptance Angle for Meridional Rays and Skew R

```
//Example 2.3
//Program to Compare the acceptance angle for
   meridional rays and
//skew rays which change direction by 100 degrees at
    each reflection

clear;
clc ;
close ;

//Given data
NA=0.4;                    //NUMERICAL APERTURE
GAMMA=100/2;               //degrees - SKEW RAYS CHANGE
   DIRECTION BY 100 degrees

//Acceptance angle for Meridional rays in degrees
THEETA_a=asin(NA)*180/%pi;

```

```
//Acceptance angle for Skew rays in degrees
THEETA_as=asin(NA/cos(GAMMA*%pi/180))*180/%pi;

//Displaying the Results in Command Window
printf("\n\n\t Acceptance angle for Meridional rays is %0.1f degrees.",THEETA_a);
printf("\n\n\t Acceptance angle for Skew rays is %0.1f degrees.",THEETA_as);
printf("\n\n\t Acceptance angle for Skew rays is about %1.0f degrees greater than Meridional rays.",THEETA_as-THEETA_a);
```

Scilab code Exa 2.4 Estimation of Normalized Frequency and Number of Guided Modes

```
//Example 2.4
//Program to estimate
//(a) Normalized frequency for the fiber
//(b) The Number of guided modes

clear;
clc ;
close ;

//Given data
n1=1.48;                    //CORE REFRACTIVE INDEX
delta=0.015                 //RELATIVE REFRACTIVE INDEX DIFFERENCE
d=80*10^(-6);               //metre - CORE DIAMETER
lambda=0.85*10^(-6);        //metre - OPERATING WAVELENGTH
a=d/2;                      //CORE RADIUS

//(a) Normalized frequency for the fiber
V=2*%pi/lambda*a*n1*sqrt(2*delta);

```

```
//(b) The Number of guided modes
Ms=(V^2)/2;

//Displaying the Results in Command Window
printf("\n\n\t The Normalized frequency for the fiber is %0.1f.",V);
printf("\n\n\t The Number of guided modes of the fiber is %d.",ceil(Ms));
```

Scilab code Exa 2.5 Estimation of total number of Guided Modes propagating in the

```
//Example 2.5
//Program to estimate total number of guided modes propagating in the fiber

clear;
clc ;
close ;

//Given data
NA=0.2;                        //NUMERICAL APERTURE
d=50*10^(-6);                  //metre - CORE DIAMETER
lambda=1*10^(-6);              //metre - OPERATING WAVELENGTH
a=d/2;                         //CORE RADIUS

//Normalized Frequency for the fiber
V=2*%pi/lambda*a*NA;

//Mode Volume for parabolic profile
M=(V^2)/4;

//Displaying the Results in Command Window
printf("\n\n\t The number of modes supported by fiber is %1.0f.",M);
```

Scilab code Exa 2.6 Estimation of maximum and new core diameter for given relative

```
//Example 2.6
//Program to estimate
//(a) The maximum core diameter of an optical fiber
    for Example 2.4
//(b) The new core diameter for single mode
    operation when the
//relative refractive index difference is reduced by
     a factor of 10

clear;
clc ;
close ;

//Given data
V=2.4;                        //Normalized Frequency
lambda=0.85*10^(-6);          //metre - OPERATING
    WAVELENGTH
n1=1.48;                      //CORE REFRACTIVE INDEX
delta=0.015;                  //RELATIVE REFRACTIVE INDEX
    DIFFERENCE

//(a) The maximum core radius of the optical fiber
    with delta=1.5%
a1=V*lambda/(2*%pi*n1*sqrt(2*delta));

//(b) The new core radius for single mode operation
    when the
//relative refractive index difference is reduced by
     a factor of 10
delta=delta/10;
a2=V*lambda/(2*%pi*n1*sqrt(2*delta));

```

```
//Displaying the Results in Command Window
printf("\n\n\t The maximum core diameter of the
    optical fiber with delta 1.5 percent is %0.1f
    micrometre.",2*a1*10^6);
printf("\n\n\t The new core diameter for single mode
     operation when the relative refractive index
    difference is reduced by a factor of 10 is %0.1f
    micrometre.",2*a2*10^6);
```

Scilab code Exa 2.7 Estimation of maximum core diameter of an optical fiber which

```
//Example 2.7
//Program to estimate the maximum core diameter of
    an optical fiber
//which allows single mode operation

clear;
clc ;
close ;

//Given data
alpha=2;                    //Parabolic Profile
lambda=1.3*10^(-6);         //metre - OPERATING
    WAVELENGTH
n1=1.5;                     //CORE REFRACTIVE INDEX
delta=0.01;                 //RELATIVE REFRACTIVE INDEX
    DIFFERENCE

//Normalized Frequency for single mode operation
V=2.4*sqrt(1+2/alpha);

//The maximum core radius for single mode operation
a=V*lambda/(2*%pi*n1*sqrt(2*delta));

//Displaying the Results in Command Window
```

```
printf("\n\n\t The maximum core diameter of the optical fiber which allows single mode operation is %0.1f micrometre.",2*a*10^6);
```

Scilab code Exa 2.8 Estimation of cutoff wavelength for a step index fiber to exhi

```
//Example 2.8
//Program to estimate cutoff wavelength for a step index fiber to
//exhibit single mode operation

clear;
clc ;
close ;

//Given data
a=4.5*10^(-6);              //metre - CORE RADIUS
n1=1.46;                    //CORE REFRACTIVE INDEX
delta=0.0025;               //RELATIVE REFRACTIVE INDEX DIFFERENCE

// The cutoff wavelength for a step index fiber
lambda_c=2*%pi*a*n1*sqrt(2*delta)/2.405;

//Displaying The Results in Command Window
printf("\n\n\t The cutoff wavelength for a step index fiber to exhibit single mode operation is %1.0f nm.",lambda_c*10^9);
```

Scilab code Exa 2.9 Deduction of an approximation for the normalized propagation c

```
//Example 2.9
```

```
//Note: MAXIMA SCILAB TOOLBOX REQUIRED FOR THIS
   PROGRAM
//Program to deduce an approximation for the
   normalized propagation
//constant

clear;
clc ;
close ;

syms W b V;

//Given data
//Eigen Value of the single mode step index fiber
   cladding
W =1.1428*V-0.9960;

//Normalized propagation constant b(V)
b= W^2/V^2;

//Display the result in command window
disp (b,"The normalized propagation constant b(V) is
   given by");
```

Scilab code Exa 2.10 Estimation of fiber core diameter for a single mode step inde

```
//Example 2.10
//Program to estimate the fiber core diameter for a
   single mode
//step index fiber

clear;
clc ;
close ;

```

```
//Given data
V=2.2;                      //NORMALIZED FREQUENCY
MFD=11.6*10^(-6);           //metre - MODE FIELD
    DIAMETER
W0=5.8*10^(-6);

// The fiber core radius
a=W0/(0.65+1.619*V^(-1.5)+2.879*V^(-6));

//Displaying the Result in Command Window
printf("\n\n\t The fiber core diameter for a single
    mode step index fiber is %0.1f um.",2*a*10^6);
```

Scilab code Exa 2.11 Determination of spot size at the operating wavelength using

```
//Example 2.11
//Program to determine spot size at the operating
    wavelength using ESI
//technique

clear;
clc ;
close ;

//Given data
lambda=1.30*10^(-6);          //metre - OPERATING
    WAVELENGTH
lambda_c=1.08*10^(-6);        //metre - CUTOFF
    WAVELENGTH
THEETA_min=12;                //degree

// The effective core radius
a_eff=3.832*lambda/(2*%pi*sin(THEETA_min*%pi/180));

// The effective normalized frequency
```

```
V_eff=2.405*lambda_c/lambda;

// The spot size
w0=3.81*10^(-6)*(0.6043+1.755*V_eff^(-1.5)+2.78*V_eff^(-6));

//Displaying the Results in Command Window
printf("\n\n\t The effective core radius is %0.2f um.",a_eff*10^6);
printf("\n\n\t The effective normalized frequency is %0.2f.",V_eff);
printf("\n\n\t The spot size at the operating wavelength is %0.2f um.",w0*10^6);
```

Scilab code Exa 2.12 Determination of relative refractive index difference using E

```
//Example 2.12
//Program to determine relative refractive index difference using ESI
//technique

clear;
clc ;
close ;

//Given data
lambda_c=1.19*10^(-6);        //metre - CUTOFF WAVELENGTH
w0=5.2*10^(-6);               //metre - SPOT SIZE
n1=1.485;                     //MAXIMUM REFRACTIVE INDEX OF THE CORE

// The ESI core diameter
d_ESI=1.820*w0;

```

```
// The ESI relative index difference
delta_ESI=(0.293/n1^2)*(lambda_c/d_ESI)^2;

//Displaying the Result in Command Window
printf("\n\n\t The relative refractive index
    difference using ESI technique is %0.2f percent."
    ,delta_ESI*10^2);
```

Chapter 3

TRANSMISSION CHARACTERISTICS OF OPTICAL FIBERS

Scilab code Exa 3.1 Determination of signal attenuation under different cases and

```
//Example 3.1
//Program to Determine
//(a)Overall signal attenuation
//(b)Signal attenuation per kilometer
//(c)Overall signal attenuation for 10 km optical
    link with splices
//(d)Numerical Input/Output power ratio

clear;
clc ;
close ;

//Given data
Pi=120;           //uW - INPUT OPTICAL POWER
Po=3;             //uW - OUTPUT OPTICAL POWER
L=8;              //km - FIBER LENGTH

```

```
//(a)Overall signal attenuation
Alpha_dB_L=10*log10(Pi/Po);

//(b)Signal attenuation per kilometer
Alpha_dB=Alpha_dB_L/L;

//(c)Overall signal attenuation for 10 km optical
   link with splices
A=Alpha_dB*10+9;

//(d)Numerical Input/Output power ratio
Pi_by_Po=10^(round(A)/10);

//Displaying the Results in Command Window
printf("\n\n\t (a)Overall signal attenuation is %1.0
   f dB.",Alpha_dB_L);
printf("\n\n\t (b)Signal attenuation per kilometer
   is %1.0f dB/km.",Alpha_dB);
printf("\n\n\t (c)Overall signal attenuation for 10
   km optical link with splices is %1.0f dB.",A);
printf("\n\n\t (d)Numerical Input/Output power ratio
    is %0.1f.",Pi_by_Po);
```

Scilab code Exa 3.2 Determination of theoretical attenuation per kilometer due to

```
//Example 3.2
//Program to Determine Theoretical attenuation in dB
   /km due to fundamental rayleigh scattering at
   optical wavelengths:
//(a)0.63um
//(b)1.00um
//(c)1.30um

clear;
clc ;
```

```
close ;

//Given data
n=1.46;              //REFRACTIVE INDEX
p=0.286;             //PHOTOELASTIC COEFFICIENT
Bc=7*10^(-11);       //m^2/N - ISOTHERMAL COMPRESSIBILITY
K=1.381*10^(-23);    //J/K - BOLTZMANN's CONSTANT
Tf=1400;             //Kelvin - FICTIVE TEMPERATURE
l=1000;              //metre - FIBER LENGTH

//(a)Attenuation in dB/km due to fundamental rayleigh scattering at 0.63um
lambda=0.63*10^(-6);                //metre - WAVELENGTH
Gamma_R=8*(%pi)^3*n^8*p^2*Bc*K*Tf/(3*lambda^4);
L_km1=exp(-Gamma_R*l)
A1=10*log10(1/L_km1);

//(b)Attenuation in dB/km due to fundamental rayleigh scattering at 1.00um
lambda=1.00*10^(-6);                //metre - WAVELENGTH
Gamma_R=8*(%pi)^3*n^8*p^2*Bc*K*Tf/(3*lambda^4);
L_km2=exp(-Gamma_R*l)
A2=10*log10(1/L_km2);

//(c)Attenuation in dB/km due to fundamental rayleigh scattering at 1.30um
lambda=1.30*10^(-6);                //metre - WAVELENGTH
Gamma_R=8*(%pi)^3*n^8*p^2*Bc*K*Tf/(3*lambda^4);
L_km3=exp(-Gamma_R*l)
A3=10*log10(1/L_km3);

//Displaying the Results in Command Window
printf("\n\n\t (a)Attenuation in dB/km due to fundamental rayleigh scattering at 0.63um = %0.1f
```

```
dB/km.",A1);
printf("\n\n\t (b)Attenuation in dB/km due to fundamental rayleigh scattering at 1.00um = %0.1f dB/km.",A2);
printf("\n\n\t (c)Attenuation in dB/km due to fundamental rayleigh scattering at 1.30um = %0.1f dB/km.",A3);
```

Scilab code Exa 3.3 Comparision of threshold optical powers for SBS and SRS

```
//Example 3.3
//Program to compare the threshold optical powers for stimulated
//Brillouin and Raman Scattering

clear;
clc ;
close ;

//Given data
alpha_dB=0.5;          //dB/km - ATTENUATION
lambda=1.3;            //micrometre - OPERATING WAVELENGTH
d=6;                   //micrometre - FIBER CORE DIAMETER
nu=0.6;                //GHz     - LASER SOURCE BANDWIDTH

//Threshold optical power for SBS
Pb=4.4*10^(-3)*(d^2)*(lambda^2)*alpha_dB*nu;

//Threshold optical power for SRS
Pr=5.9*10^(-2)*d^2*lambda*alpha_dB;

//Displaying the Results in Command Window
```

```
printf("\n\n\t The threshold optical power for SBS is %0.1f mW.",Pb*10^3);
printf("\n\n\t The threshold optical power for SRS is %0.2f W.",Pr);
```

Scilab code Exa 3.4 Estimation of critical radius of curvature

```
//Example 3.4
//Program to estimate critical radius of curvature at which large
//bending loss occur

clear;
clc ;
close ;

//Given data for part (a)
n1=1.500;          //metre - LENGTH
delta=0.03;        //*100 percent - RELATIVE REFRACTIVE INDEX DIFFERENCE
lambda=0.82*10^(-6); //metre - OPERATING WAVELENGTH

//Calculation of the radius of curvature of Multi Mode fiber
n2=sqrt(n1^2-2*delta*n1^2);
Rc=3*n1^2*lambda/(4*%pi*(n1^2-n2^2)^(3/2));

//Given data for part (b)
n1=1.500;          //metre - LENGTH
delta=0.003;       //*100 percent - RELATIVE REFRACTIVE INDEX DIFFERENCE
lambda=1.55*10^(-6); //metre - OPERATING WAVELENGTH
d=8*10^(-6);       //metre - CORE DIAMETER

//Calculation of the radius of curvature of Single
```

```
Mode fiber
n2=sqrt(n1^2-2*delta*n1^2);
a=d/2;
lambda_c=2*%pi*a*n1*sqrt(2*delta)/2.405;
Rcs=20*lambda*(2.748-0.996*lambda/lambda_c)^(-3)/(n1-n2)^(3/2);

//Displaying the Results in Command Window
printf("\n\n\t (a)The radius of curvature of Multi Mode fiber is %1.0f um.",Rc/10^(-6));
printf("\n\n\t (b)The radius of curvature of Single Mode fiber is %1.0f mm.",Rcs/10^(-3));
```

Scilab code Exa 3.5 Estimation of Maximum Bandwidth Pulse dispersion per unit leng

```
//Example 3.5
//Program to estimate
//(a)The maximum possible bandwidth on the link assuming no ISI
//(b)The pulse dispersion per unit length
//(c)The bandwidth-length product for the fiber

clear;
clc ;
close ;

//Given data
tau=0.1*10^(-6); //second - TOTAL PULSE BROADENING
L=15;             //km - DISTANCE

//(a)The maximum possible bandwidth on the link assuming no ISI
B_opt=1/(2*tau);

//(b)The pulse dispersion per unit length
```

```
Dispersion=tau/L;

//(c)The bandwidth-length product for the fiber
B_optXL=B_opt*L;

//Displaying the Results in Command Window
printf("\n\n\t (a)The maximum possible bandwidth on the link assuming no ISI is %1.0f MHz.",B_opt/10^6);
printf("\n\n\t (b)The pulse dispersion per unit length is %0.2f ns/km.",Dispersion/10^(-9));
printf("\n\n\t (c)The bandwidth-length product for the fiber is %1.0f MHz km.",B_optXL/10^6);
```

Scilab code Exa 3.6 Determination of Material Dispersion Parameter and RMS Pulse B

```
//Example 3.6
//Program to estimate Material dispersion parameter and rms pulse
//broadening per kilometer
clear;
clc ;
close ;

//Given data
lambda=0.85*10^(-6);      //metre - WAVELENGTH
L=1;                      //km - DISTANCE
MD=0.025;//MATERIAL DISPERSION = mod(lamda^2*[del^2(n1)/del(lamda)^2)
c=2.998*10^8;             //m/s - VELOCITY OF LIGHT IN VACCUM
sigma_lambda=20*10^(-9);//metre - RMS SPECTRAL WIDTH

//Material Dispersion Parameter
M=MD/(lambda*c);
```

```
//R.M.S. pulse broadening per kilometer
sigma_m=sigma_lambda*L*M;

//Displaying the Results in Command Window
printf("\n\n\t Material Dispersion Parameter is %0.1
   f ps/nm/km.",M*10^6);
printf("\n\n\t R.M.S. pulse broadening per kilometer
    is %0.2f ns/km.",sigma_m/10^(-12));
```

Scilab code Exa 3.7 Estimation of RMS Pulse Broadening per kilometer for the fiber

```
//Example 3.7
//Program to estimate rms pulse broadening per
   kilometer for the fiber

clear;
clc ;
close ;

//Given data
lambda=0.85*10^(-6);       //metre - WAVELENGTH
L=1;                       //km - DISTANCE
MD=0.025;//MATERIAL DISPERSION = mod(lamda^2*[del^2(
   n1)/del(lamda)^2)
c=2.998*10^8;              //m/s - VELOCITY OF LIGHT IN
    VACCUM
sigma_lambda_by_lambda=0.0012;// sigma_lambda/lambda

//Material Dispersion Parameter
M=MD/(lambda*c);

//R.M.S. Spectral Width
sigma_lambda=sigma_lambda_by_lambda*lambda;

```

```
//R.M.S. pulse broadening per kilometer
sigma_m=sigma_lambda*L*M;

//Displaying the Result in Command Window
printf("\n\n\t R.M.S. pulse broadening per kilometer is %0.2f ns/km.",sigma_m/10^(-12));
```

Scilab code Exa 3.8 Estimation of Delay Difference RMS Pulse Broadening Maximum Bi

```
//Example 3.8
//Program to estimate
//(a)The delay difference between the slowest and fastest modes at the fiber output
//(b)The rms pulse broadening due to intermodal dispersion on the link
//(c)The maximum bit rate
//(d)Bandwidth−length product corresponding to (c)

clear;
clc ;
close ;

//Given data
delta=0.01;        //*100 percent − RELATIVE REFRACTIVE INDEX DIFFERENCE
L=6;               //km − LENGTH OF OPTICAL LINK
n1=1.5;            //CORE REFRACTIVE INDEX
c=2.998*10^8;      //m/s − VELOCITY OF LIGHT IN VACCUM

//(a)The delay difference between the slowest and fastest modes at the fiber output
del_Ts=L*n1*delta/c;

//(b)The rms pulse broadening due to intermodal dispersion on the link
```

```
sigma_s=L*n1*delta/(2*sqrt(3)*c);

//(c)The maximum bit rate
Bt=1/(2*del_Ts);
//Improved maximum bit rate
Bti=0.2/sigma_s;

//(d)Bandwidth-length product corresponding to (c)
BoptXL=Bti*L;

//Displaying the Results in Command Window
printf("\n\n\t (a)The delay difference between the slowest and fastest modes at the fiber output is %1.0f ns.",del_Ts/10^(-12));
printf("\n\n\t (b)The rms pulse broadening due to intermodal dispersion on the link is %0.1f ns.",sigma_s/10^(-12));
printf("\n\n\t (c)The maximum bit rate is %0.1f Mbit/s and improved bit rate is %0.1f Mbit/s.",Bt/10^(9),Bti/10^(9));
printf("\n\n\t (d)Bandwidth-length product is %0.1f MHz km.",BoptXL/10^(9));
```

Scilab code Exa 3.9 Comparision of RMS Pulse Broadening per Kilometer for two case

```
//Example 3.9
//Program to compare rms pulse broadening per kilometer due to
//intermodal dispersion for multimode step index fiber with that of
//near parabolic graded index fiber

clear;
clc ;
close ;
```

```

//Given data
delta=0.01;        //*100 percent - RELATIVE
   REFRACTIVE INDEX DIFFERENCE
L=1;               //km - LENGTH OF OPTICAL LINK
n1=1.5;            //CORE REFRACTIVE INDEX
c=2.998*10^8;      //m/s - VELOCITY OF LIGHT IN VACCUM

//RMS pulse broadening /km due to intermodal
   dispersion for MMSI Fiber
sigma_s=L*n1*delta/(2*sqrt(3)*c);

//RMS pulse broadening /km for near parabolic graded
   index fiber
sigma_g=L*n1*delta^2/(20*sqrt(3)*c);

//Displaying the Results in Command Window
printf("\n\n\t RMS pulse broadening per kilometer
   due to intermodal dispersion for MMSI Fiber is %0
   .1f ns/km.",sigma_s/10^(-12));
printf("\n\n\t RMS pulse broadening per kilometer
   for near parabolic graded index fiber is %0.1f ps
   /km.",sigma_g/10^(-15));
```

Scilab code Exa 3.10 Estimation of total RMS pulse broadening and BW Length produc

```
//Example 3.10
//Program to estimate
//(a)RMS pulse broadening per kilometer
//(b)Bandwidth-Length product for the fiber

clear;
clc ;
close ;

```

```
//Given data
NA=0.3 ;            //NUMERICAL APERTURE
n1=1.45;            //CORE REFRACTIVE INDEX
M=250*10^(-6);      //s/km^2 - MATERIAL DISPERSION
    PARAMETER
sigma_lambda=50*10^(-9); //metre - RMS SPECTRAL
    WIDTH
L=1;                //km - LENGTH OF OPTICAL LINK
c=2.998*10^8;       //m/s - VELOCITY OF LIGHT IN VACCUM

//RMS pulse broadening /km due to material
    dispersion
sigma_m=sigma_lambda*L*M;

//RMS pulse broadening /km due to intermodal
    dispersion
sigma_s=L*NA^2/(4*sqrt(3)*n1*c);

//(a)Total RMS pulse broadening /km
sigma_t=sqrt(sigma_m^2+sigma_s^2);

//(b)Bandwidth-Length product
BoptXL=0.2/sigma_t;

//Displaying the Results in Command Window
printf("\n\n\t Total RMS pulse broadening per
    kilometer is %0.1f ns/km.",sigma_t/10^(-12));
printf("\n\n\t Bandwidth-Length product is %0.1f MHz
    km.",BoptXL/10^(9));
```

Scilab code Exa 3.11 Comparision of total first order dispersion for the fiber

```
//Example 3.11
//Program to compare the total first order
    dispersion and determine
```

```
//waveguide dispersion

clear;
clc ;
close ;

//Given data
lambda0=1310;                //nm − ZERO DISPERSION WAVELENGTH
So=0.09*10^(-12);            //s/nm^2/km − DISPERSION SLOPE

//Dt at 1280nm
lambda1=1280;                //nm − OPERATING WAVELENGTH
Dt1=lambda1*So/4*(1-(lambda0/lambda1)^4);

//Dt at 1550nm
lambda2=1550;                //nm − OPERATING WAVELENGTH
Dt2=lambda2*So/4*(1-(lambda0/lambda2)^4);

//Waveguide Dispersion at 1550nm
Dm=13.5*10^(-12);            //s/nm/km − MATERIAL DISPERSION
Dp=0.4*10^(-12);             //s/nm/km − PROFILE DISPERSION
Dw=Dt2-(Dm+Dp);

//Displaying the Results in Command Window
printf("\n\n\t Dt(1280nm) = %0.1f ps/nm/km.",Dt1/10^(-12));
printf("\n\n\t Dt(1550nm) = %0.1f ps/nm/km.",Dt2/10^(-12));
printf("\n\n\t Dw = %0.1f ps/nm/km.",Dw/10^(-12));
```

Scilab code Exa 3.12 Determination of Modal birefringence coherence length and dif

```
//Example 3.12
//Program to determine modal birefringence, coherence length and difference between propagation constants for the two orthogonal modes

clear;
clc ;
close ;

//Given data
lambda=0.9*10^(-6);    //metre - PEAK WAVELENGTH
Lb=9*10^(-2);          //metre - BEAT LENGTH
del_lambda=1*10^(-9); //metre - SPECTRAL LINE WIDTH

//Modal Birefringence
Bf=lambda/Lb;

//Coherence Length
Lbc=lambda^2/(Bf*del_lambda);

//Difference between propagation constants for the two orthogonal
//modes
Bx_minus_By=2*%pi/Lb;

//Displaying the Results in Command Window
printf("\n\n\t The Modal birefringence  is %1.0f X 10^(-5) .",Bf/10^(-5));
printf("\n\n\t The Coherence Length is %d m.",round(Lbc));
printf("\n\n\t The difference between propagation constants for the two orthogonal modes is %0.1f .",Bx_minus_By);
```

Scilab code Exa 3.13 Determination of fiber birefringence for two given cases

```
//Example 3.13
//Program to determine fiber birefringence for given
    beat lengths
//(1)Lb = 0.7 mm
//(2)Lb = 80 m

clear;
clc ;
close ;

//Given data
lambda=1.3*10^(-6); //metre - OPERATING WAVELENGTH

//Part (1)
Lb1=0.7*10^(-3);     //metre - BEAT LENGTH
Bf1=lambda/Lb1;

//Part (2)
Lb2=80;              //metre - BEAT LENGTH
Bf2=lambda/Lb2;

//Displaying the Results in Command Window
printf("\n\n\t The fiber birefringence for Lb = 0.7
   mm is %0.2f X 10^(-3) which is high.",Bf1/10^(-3)
   );
printf("\n\n\t The fiber birefringence for Lb = 80 m
    is %0.2f X 10^(-8) which is low.",Bf2/10^(-8));
```

Scilab code Exa 3.14 Determination of mode coupling parameter for the fiber

```
//Example 3.14
//Program to determine the mode coupling parameter
   for the fiber
```

```
clear;
clc ;
close ;

//Given data
L=3.5*10^3;          //metre - LENGTH
CT=-27;              //dB - POLARIZATION CROSSTALK

//Mode coupling parameter for the fiber
h=(10^(CT/10))/L; //as tan(h*L)=h*L for small values

//Displaying the Result in Command Window
printf("\n\n\t The mode coupling parameter for the
    fiber is %0.1f X 10^(-7)/m.",h/10^(-7));
```

Chapter 4

OPTICAL FIBERS AND CABLES

Scilab code Exa 4.1 Estimation of fracture stress for the fiber and percentage str

```
//Example 4.1
//Program to determine the following:
//(a) Fracture Stress in psi for the fiber
//(b) Percentage Strain at the break

clear;
clc ;
close ;

//Given data
St=2.6*10^6;            //psi    - THEORETICAL
    COHESIVE STRENGTH
la=0.16*10^-9;          //metres - BOND DISTANCE
C=10*10^-9;             //metres - DEPTH OF CRACK
E= 9*10^10  ;           //N/m^2  - YOUNG'S MODULUS OF
    SILICA

Gamma_p=(4*la*St^2)/E;

```

```
//Fracture Stress for an Elliptical Crack
Sf_psi=sqrt((2*E*Gamma_p)/(%pi*C));

//Fracture Stress in psi units
Sf=Sf_psi*6894.76;

//Strain Calculation
strain=Sf/E;

//Displaying the Results in Command Window
printf("\n\n\t Fracture Stress for the fiber is %0.2f X 10^9 N/m or %0.2f X 10^5 psi.",Sf/10^9,Sf_psi/10^5);
printf("\n\n\t Percentage Strain at the break is %d percent.",strain*100);
```

Chapter 5

OPTICAL FIBER CONNECTIONS JOINTS COUPLERS AND ISOLATORS

Scilab code Exa 5.1 Calculation of the optical loss in decibels at the joint

```
//Example 5.1
//Program to calculate the optical loss in decibels
   at the joint

clear;
clc ;
close ;

//Given data
n1=1.5;            //CORE REFRACTIVE INDEX
n=1.0;

//Magnitude of Frensel reflection at the fiber-air
   interface
r=((n1-n)/(n1+n))^2;
```

```
//Optical Loss
Loss_fres=-10*log10(1-r);

//Displaying the Results in Command Window
printf("\n\n\t Optical Loss is %0.2f dB .",Loss_fres
   );
printf("\n\n\t Total loss due to Frensel Reflection
   at the fiber joint is %0.2f dB .",Loss_fres*2);
```

Scilab code Exa 5.2 Estimation of the insertion loss in two given cases

```
//Example 5.2
//Program to estimate the insertion loss when:
//(a)there is a small air gap at the joint
//(b)the joint is considered index matched

clear;
clc ;
close ;

//Given data
n1=1.5;            //CORE REFRACTIVE INDEX
n=1.0;
y=5*10^(-6);       //metre - LATERAL MISALIGNMENT
a=25*10^(-6);      //metre - CORE RADIUS

//(a)Coupling efficiency
eeta_lat1=16*(n1/n)^2/(1+(n1/n))^4*1/%pi*(2*acos(y
   /(2*a))-(y/a)*sqrt(1-(y/(2*a))^2));
//Insertion Loss
Loss_lat1=-10*log10(eeta_lat1);

//(b)Coupling efficiency
eeta_lat2=1/%pi*(2*acos(y/(2*a))-(y/a)*sqrt(1-(y/(2*
```

```
a))^2));
//Insertion Loss
Loss_lat2=-10*log10(eeta_lat2);

//Displaying the Results in Command Window
printf("\n\n\t (a)Insertion Loss (there is a small air gap at the joint) is %0.2f dB .",Loss_lat1);
printf("\n\n\t (b)Insertion Loss (the joint is considered index matched) is %0.2f dB .",Loss_lat2);
```

Scilab code Exa 5.3 Estimation of the insertion loss in two given cases

```
//Example 5.3
//Program to estimate the insertion loss when:
//(a)there is uniform illumination of all guided modes only
//(b)there is uniform illumination of all guided and leaky modes

clear;
clc ;
close ;

//Given data
y=3*10^(-6);      //metre - LATERAL MISALIGNMENT
a=25*10^(-6);     //metre - CORE RADIUS

//(a)Misalignment Loss
Lt1=0.85*(y/a);
//Coupling efficiency
eeta_lat1=1-Lt1;
//Insertion Loss
Loss_lat1=-10*log10(eeta_lat1);

```

```
//(b)Misalignment Loss
Lt2=0.75*(y/a);
//Coupling efficiency
eeta_lat2=1-Lt2;
//Insertion Loss
Loss_lat2=-10*log10(eeta_lat2);

//Displaying the Results in Command Window
printf("\n\n\t (a)Insertion Loss (there is uniform
    illumination of all guided modes only) is %0.2f
    dB .",Loss_lat1);
printf("\n\n\t (b)Insertion Loss (there is uniform
    illumination of all guided and leaky modes) is %0
    .2f dB .",Loss_lat2);
```

Scilab code Exa 5.4 Estimation of the insertion loss in two given cases

```
//Example 5.4
//Program to estimate the insertion loss for
//NA = 0.2
//NA = 0.4

clear;
clc ;
close ;

//Given data
n1=1.48;            //CORE REFRACTIVE INDEX
n=1.0;
theeta=5;           //degree - ANGULAR MISALIGNMENT

//Calculation for NA = 0.2
NA=0.2
eeta_ang1=16*(n1/n)^2/(1+n1/n)^4*(1-n*theeta*%pi
    /180/(%pi*NA));
```

```
//Insertion Loss
Loss_ang1=-10*log10(eeta_ang1);

//Calculation for NA = 0.4
NA=0.4
eeta_ang2=16*(n1/n)^2/(1+n1/n)^4*(1-n*theeta*%pi/180/(%pi*NA));
//Insertion Loss
Loss_ang2=-10*log10(eeta_ang2);

//Displaying the Results in Command Window
printf("\n\n\t Insertion Loss (NA=0.2) is %0.2f dB.",Loss_ang1);
printf("\n\n\t Insertion Loss (NA=0.4) is %0.2f dB.",Loss_ang2);
```

Scilab code Exa 5.5 Estimation of the total insertion loss of the fiber joint with

```
//Example 5.5
//Program to estimate the total insertion loss of the fiber joint
//with a lateral misalignment and angular misalignment

clear;
clc ;
close ;

//Given data
V=2.40;          //NORMALIZED FREQUENCY
n1=1.46;         //CORE REFRACTIVE INDEX
d=8*10^(-6);     //metre - CORE DIAMETER
NA=0.1;          //NUMERICAL APERTURE
y=1*10^(-6);     //metre - LATERAL MISALIGNMENT
theeta=1;        //degree - ANGULAR MISALIGNMENT
```

```

//Normalized Spot Size
a=d/2;
omega=a*(0.65+1.62*V^(-3/2)+2.88*V^(-6))/sqrt(2);

//Loss due to lateral offset
Tl=2.17*(y/omega)^2;

//Loss due to angular misalignment
Ta=2.17*((theeta*%pi/180)*omega*n1*V/(a*NA))^2;

//Total insertion loss
Tt=Tl+Ta;

//Displaying the Result in Command Window
printf("\n\n\t Total Insertion Loss is %0.2f dB .",
   Tt);
```

Scilab code Exa 5.6 Calculation of the loss at the connection due to mode field di

```
//Example 5.6
//Program to calculate the loss at the connection
   due to mode field
//diameter mismatch

clear;
clc ;
close ;

//Given data
MFD01=11.2;            //um - MODE FIELD DIAMETER
MFD02=8.4;             //um - MODE FIELD DIAMETER

//Calculation of Intrinsic Loss
omega_01=MFD01/2;
```

```
omega_02=MFD02/2;
Loss_int=-10*log10(4*(omega_02/omega_01+omega_01/omega_02)^(-2))

//Displaying the Result in Command Window
printf("\n\n\t Intrinsic Loss is %0.2f dB .",Loss_int);
```

Scilab code Exa 5.7 Determination of excess loss insertion losses crosstalk and sp

```
//Example 5.7
//Program to determine the excess loss, insertion losses, crosstalk
//and split ratio

clear;
clc ;
close ;

//Given data
P1=60*10^(-6);      //Watts - INPUT POWER AT PORT 1
P2=0.004*10^(-6);   //Watts - OUTPUT POWER AT PORT 2
P3=26*10^(-6);      //Watts - OUTPUT POWER AT PORT 3
P4=27.5*10^(-6);    //Watts - OUTPUT POWER AT PORT 4

//Calculation of Excess Loss
Excess_loss=10*log10(P1/(P3+P4));

//Calculation of Insertion Loss (ports 1 to 3)
Insertion_loss3=10*log10(P1/P3);

//Calculation of Insertion Loss (ports 1 to 4)
Insertion_loss4=10*log10(P1/P4);

//Calculation of Crosstalk
```

```
Crosstalk=10*log10(P2/P1);

//Calculation of Split Ratio
Split_ratio=P3/(P3+P4)*100;

//Displaying the Results in Command Window
printf("\n\n\t Excess Loss is %0.2f dB .",Excess_loss);
printf("\n\n\t Intrinsic Loss (ports 1 to 3) is %0.2f dB .",Insertion_loss3);
printf("\n\n\t Intrinsic Loss (ports 1 to 4) is %0.2f dB .",Insertion_loss4);
printf("\n\n\t Crosstalk is %0.1f dB .",Crosstalk);
printf("\n\n\t Split Ratio is %0.1f percent .",Split_ratio);
```

Scilab code Exa 5.8 Determination of excess loss insertion losses crosstalk and sp

```
//Example 5.8
//Program to determine the total loss incurred by the star coupler
//and average insertion loss

clear;
clc ;
close ;

//Given data
Pi=1*10^(-3);        //Watts - INPUT POWER AT PORT 1
Po=14*10^(-6);       //Watts - OUTPUT POWER AT OTHER PORTS
N=32;                //Ports

//Calculation of Splitting Loss
Splitting_loss=10*log10(N);
```

```

//Calculation of Excess Loss
Excess_loss=10*log10(Pi/(Po*N));

//Calculation of Total loss
Total_loss=Splitting_loss+Excess_loss;

//Calculation of Average Insertion Loss
Insertion_loss=10*log10(Pi/Po);

//Displaying the Results in Command Window
printf("\n\n\t Total loss is %0.2f dB .",Total_loss)
    ;
printf("\n\n\t Average Insertion Loss is %0.2f dB ."
    ,Insertion_loss);
```

Scilab code Exa 5.9 Determination of the insertion loss associated with one typica

```
//Example 5.9
//Program to determine the insertion loss associated
    with one typical
//path

clear;
clc ;
close ;

//Given data
Excess_loss=0.2;    //dB - EXCESS LOSS OF EACH PORT
Split_ratio=0.5;    //*100 percent - SPLIT RATIO
N=16;               //PORTS
M=4;                //For N=16 ports
Splice_loss=0.1;    //dB - SPLICE LOSS

//Calculation of Total Excess Loss
```

```
Total_Excess_loss=M*Excess_loss+3*Splice_loss;

//Calculation of Splitting Loss
Splitting_loss=10*log10(N);

//Calculation of Insertion Loss
Insertion_loss=Splitting_loss+Total_Excess_loss;

//Displaying the Result in Command Window
printf("\n\n\t Insertion Loss is %0.2f dB .",
    Insertion_loss);
```

Scilab code Exa 5.10 Calculation of the grating period for reflection

```
//Example 5.10
//Program to find the grating period for reflection

clear;
clc ;
close ;

//Given data
n=1.46;         //CORE REFRACTIVE INDEX
lambda_b=1.55;  //um - WAVELENGTH

//Grating Period
lambda=lambda_b/(2*n);

//Displaying the Result in Command Window
printf("\n\n\t Grating Period is %0.2f um .",lambda)
    ;
```

Chapter 6

OPTICAL SOURCES 1 THE LASER

Scilab code Exa 6.1 `Calculation of the ratio of stimulated emission rate to the sp`

```
//Example 6.1
//Program to calculate the ratio of stimulated emission rate to the
//spontaneous emission rate

clear;
clc ;
close ;

//Given data
Lambda=0.5*10^-6;       //metres - OPERATING WAVELENGTH
k=1.381*10^(-23);       //m^2 kg/s - BOLTZMANN's CONSTANT
c= 2.998*10^8;          //m/s - SPEED OF LIGHT
h=6.626*10^(-34);       //J/K - PLANK's CONSTANT
T=1000;                 //Kelvin - TEMPERATURE

//Average operating frequency
```

```
f=c/Lambda;

//Stimulated Emission Rate/Spontaneous Emission Rate
Ratio=1/(exp(h*f/(k*T))-1);

//Displaying the Result in Command Window
printf("\n\n\t Stimulated Emission Rate/Spontaneous
   Emission Rate =  %0.1f X 10^(-13).",Ratio
   /10^(-13));
```

Scilab code Exa 6.2 Determination of the number of longitudinal modes and their fr

```
//Example 6.2
//Program to determine the number of longitudinal
   modes and their
//frequency separation in a ruby laser

clear;
clc ;
close ;

//Given data
Lambda=0.55*10^-6;     //metres - PEAK EMISSION
   WAVELENGTH
n=1.78;                //REFRACTIVE INDEX
c= 2.998*10^8;         //m/s - SPEED OF LIGHT
L=4*10^(-2);           //metres - CRYSTAL LENGTH

//Number of Longitudinal modes
q=2*n*L/Lambda;

//Frequency separation of the modes
del_f=c/(2*n*L);

//Displaying the Results in Command Window
```

```
printf("\n\n\t Number of Longitudinal modes is %0.1f X 10^5.",q/10^5);
printf("\n\n\t Frequency separation of the modes is %0.1f GHz.",del_f/10^9);
```

Scilab code Exa 6.3 Calculation of laser gain coefficient for the cavity

```
//Example 6.3
//Program to calculate laser gain coefficient for the cavity

clear;
clc ;
close ;

//Given data
L=600*10^-4;            //cm - CAVITY LENGTH
r=0.3;                  //*100 percent - REFLECTIVITY
alpha_bar= 30;          //per cm - LOSSES

//Laser Gain Coefficient
gth_bar=alpha_bar+1/L*log(1/r);

//Displaying the Result in Command Window
printf("\n\n\t Laser Gain Coefficient is %1.0f per cm.",gth_bar);
```

Scilab code Exa 6.4 Comparision of the approximate radiative minority carrier life

```
//Example 6.4
//Program to compare the approximate radiative minority carrier
//lifetimes in gallium arsenide and silicon
```

```

clear;
clc ;
close ;

//Given data
N=10^18;                    //per cm^3 - HOLE CONCENTRATION
Br1=7.21*10^(-10);      //cm^3 / s - RECOMBINATION COEFFICIENT FOR GaAs
Br2=1.79*10^(-15);      //cm^3 / s - RECOMBINATION COEFFICIENT FOR Si

//Radiative minority carrier lifetime for GaAs
tau_r1=1/(Br1*N);

//Radiative minority carrier lifetime for Si
tau_r2=1/(Br2*N);

//Displaying the Results in Command Window
printf("\n\n\t Radiative minority carrier lifetime for GaAs is %0.2f ns.",tau_r1/10^(-9));
printf("\n\n\t Radiative minority carrier lifetime for Si is %0.2f ms.",tau_r2/10^(-3));
```

Scilab code Exa 6.5 Determination of the threshold current density and the thresho

```
//Example 6.5
//Program to determine the threshold current density and the
//threshold current for the device

clear;
clc ;
close ;
```

```

//Given data
n=3.6;                          //REFRACTIVE INDEX OF GaAs
beeta_bar=21*10^(-3);           //A/cm^3 - GAIN FACTOR
alpha_bar=10;                   //per cm - LOSS
    COEFFICIENT
L=250*10^(-4);                  //cm - LENGTH OF OPTICAL
    CAVITY
W=100*10^(-4);                  //cm - WIDTH OF OPTICAL
    CAVITY

//Reflectivity for normal incidence
r=((n-1)/(n+1))^2;

//Threshold current density
Jth=1/beeta_bar*(alpha_bar+1/L*log(1/r));

//Threshold current
Ith=Jth*W*L;

//Displaying the Results in Command Window
printf("\n\n\t Threshold current density is %0.2f X
    10^3 A/cm^2.",Jth/10^3);
printf("\n\n\t Threshold current is %0.1f mA.",Ith
    /10^(-3));
```

Scilab code Exa 6.6 Calculation of external power efficiency of the device

```
//Example 6.6
//Program to calculate the external power efficiency
     of the device

clear;
clc ;
close ;
```

```
//Given data
eeta_t=0.18;          //*100 percent - TOTAL
   EFFICIENCY
Eg=1.43;              //eV - ENERGY BAND GAP OF GaAs
V=2.5;                //Volts - APPLIED VOLTAGE

//External power efficiency of the device
eeta_ep=eeta_t*Eg/V;

//Displaying the Result in Command Window
printf("\n\n\t External power efficiency of GaAs
    device is %1.0f percent.",eeta_ep*100);
```

Scilab code Exa 6.7 Comparision of the ratio of threshold current densities at 20

```
//Example 6.7
//Program to compare the ratio of threshold current
    densities at 20 C
//and 80 C for AlGaAs and InGaAsP

clear;
clc ;
close ;

//Given data
T1=293;               //degree C
T2=352;               //degree C

//For AlGaAs
T0=170;               //degree C
Jth_20=exp(T1/T0);
Jth_80=exp(T2/T0);
Ratio=Jth_80/Jth_20;

```

```
//Displaying the Result in Command Window
printf("\n\n\t Ratio of current densities for AlGaAs
    is %0.2f .",Ratio);

//For InGaAsP
T0=55;                    //degree C
Jth_20=exp(T1/T0);
Jth_80=exp(T2/T0);
Ratio=Jth_80/Jth_20;

//Displaying the Result in Command Window
printf("\n\n\t Ratio of current densities for
   InGaAsP is %0.2f .",Ratio);
```

Scilab code Exa 6.8 Determination of RMS value of the power fluctuation and RMS no

```
//Example 6.8
//Determine the
//(a)The RMS value of the power fluctuation
//(b)The RMS noise current at the output of the
   detector

clear;
clc ;
close ;

//Given data
B=100*10^6;                          //Hz - BANDWIDTH
S_rinf_by_Pebarsquare=10^(-15);   //per Hz - RIN
   VALUE
e=1.602*10^(-19);                    //Coulumbs - CHARGE
    OF AN ELECTRON
eeta=0.6;                            //*100 percent -
   QUANTUM EFFICIENCY
lambda=1.55*10^(-6);                 //metre -
```

```
WAVELENGTH
h= 6.626*10^(-34);                    //J/K - PLANK's CONSTANT
c=2.998*10^8;                         //m/s - VELOCITY OF LIGHT IN VACCUM
Pe_bar=2*10^(-3);                     //Watt - INCIDENT POWER

//(a)The RMS value of the power fluctuation
RMS_value=sqrt(S_rinf_by_Pebarsquare*B);

//(b)The RMS noise current at the output of the detector
RMS_noise_current=e*eeta*lambda/(h*c)*RMS_value*Pe_bar;

//Displaying the Results in Command Window
printf("\n\n\t (a)The RMS value of the power fluctuation is %0.2f X 10^(-4) W.",RMS_value/10^(-4));
printf("\n\n\t (b)The RMS noise current at the output of the detector is %0.2f X 10^(-7) A.",RMS_noise_current/10^(-7));
```

Chapter 7

OPTICAL SOURCES 2 THE LIGHT EMITTING DIODE

Scilab code Exa 7.1 Determination of total carrier recombination lifetime and the

```
//Example 7.1
//Program to determine the total carrier recombination lifetime and
//the power internally generated within the device

clear;
clc ;
close ;

//Given data
Tau_r=60;            //ns - RADIATIVE RECOMBINATION LIFETIME
Tau_nr=100;          //ns - NON RADIATIVE RECOMBINATION LIFETIME
Lambda=0.87*10^-6;   //metres - PEAK EMISSION WAVELENGTH
c= 2.998*10^8;       //m/s - SPEED OF LIGHT
h= 6.626*10^(-34);   //J/K - PLANK's CONSTANT
e=1.602*10^(-19);    //Coulumbs - CHARGE OF AN
```

```
ELECTRON
i=40*10^(-3);        //A − DRIVE CURRENT

//Total carrier recombination lifetime
Tau=Tau_r*Tau_nr/(Tau_r+Tau_nr);

//Internal quantum efficiency
eeta_int=Tau/Tau_r;

//Power internally generated within the device
P_int=eeta_int*h*c*i/(Lambda*e);

//Displaying the Results in Command Window
printf("\n\n\t Total carrier recombination lifetime is %0.1f ns.",Tau);
printf("\n\n\t Power internally generated within the device is %0.1f mW .",P_int/10^(-3));
```

Scilab code Exa 7.2 Calculation of optical power emitted into air as a percentage

```
//Example 7.2
//Program to :
//(a)Calculate the optical power emitted into air as a percentage of
//internal optical power
//(b)Determine the external power efficiency

clear;
clc ;
close ;

//Given data
F=0.68;                     //TRANSMISSION FACTOR
n=1;
nx=3.6;                     //REFRACTIVE INDEX OF GaAs
```

```
Pint_by_P=0.5;                //*100 percent - Pe/P

//Percentage optical power emitted
Pe_by_Pint=F*n^2/(4*nx^2);

//External power efficiency
eeta_ep=Pe_by_Pint*Pint_by_P;

//Displaying the Results in Command Window
printf("\n\n\t (a)Percentage optical power emitted
   is %0.1f percent of generated optical power.",
   Pe_by_Pint*100);
printf("\n\n\t (b)External power efficiency is %0.2f
    percent.",eeta_ep*100);
```

Scilab code Exa 7.3 Calculation of Coupling Efficieny and Optical loss in decibels

```
//Example 7.3
//Program to calculate the:
//(a)Coupling Efficieny
//(b)Optical loss in decibels relative to Pe
//(c)Optical loss in decibels relative to Pint

clear;
clc ;
close ;

//Given data
NA=0.2;                       //NUMERICAL APERTURE
F=0.68;                       //TRANSMISSION FACTOR
n=1;
nx=3.6;                       //REFRACTIVE INDEX OF GaAs

//(a)Coupling Efficieny
eeta_c=(NA)^2;
```

```
//(b)Optical loss in decibels relative to Pe
Loss1=-10*log10(eeta_c);

//Percentage optical power emitted
Pint_by_P=F*n^2/(4*nx^2);

//(c)Optical loss in decibels relative to Pint
Loss2=-10*log10(eeta_c*Pint_by_P);

//Displaying the Results in Command Window
printf("\n\n\t (a)Coupling Efficieny is %1.0f
    percent.",eeta_c*100);
printf("\n\n\t (b)Optical loss in decibels relative
    to Pe is %0.1f dB.",Loss1);
printf("\n\n\t (c)Optical loss in decibels relative
    to Pint is %0.1f dB.",Loss2);
```

Scilab code Exa 7.4 Estimation of the optical power coupled into the fiber

```
//Example 7.4
//Program to estimate the optical power coupled into
     the fiber

clear;
clc ;
close ;

//Given data
d=50*10^(-4);              //cm - EMISSION AREA
    DIAMETER
R_D=30;                    //W/sr/cm^2
NA=0.15;                   //NUMERICAL APERTURE
r=0.01;                    //REFLECTION COEFFICIENT

```

```
//Optical power coupled into the fiber
a=d/2;                          //RADIUS
A=%pi*a^2;                      //EMISSION AREA
Pc=%pi*(1-r)*A*R_D*NA^2;

//Displaying the Result in Command Window
printf("\n\n\t Optical power coupled into the fiber
   is %0.1f uW.",Pc/10^(-6));
```

Scilab code Exa 7.5 Determination of the overall power conversion efficiency

```
//Example 7.5
//Program to determine the overall power conversion
   efficiency

clear;
clc ;
close ;

//Given data
Pc=190*10^(-6);        //Watts - INPUT OPTICAL POWER
I=25*10^(-3);          //A - FORWARD CURRENT
V=1.5;                 //V - FORWARD VOLTAGE

//Overall power conversion efficiency
P=I*V;
eeta_pc=Pc/P;

//Displaying the Result in Command Window
printf("\n\n\t Overall power conversion efficiency
   is %0.1f percent.",eeta_pc*100);
```

Scilab code Exa 7.6 Comparision of electrical and optical bandwidth for an optical

```
//Example 7.6
//Compare the electrical and optical bandwidth for an optical fiber
//commuication system and develop a relationship between the two

clear;
clc ;
close ;

//Given data
Re_dB=3;                        //dB - ELECTRICAL 3 dB POINTS
Ro_dB=3;                        //dB - OPTICAL 3 dB POINTS

//Electrical Bandwidth
Iout_by_Iin=sqrt(10^(-Re_dB/10));
printf("\n\n\t For Electrical Bandwidth, Iout/Iin = %0.3f .",Iout_by_Iin);

//Optical Bandwidth
Iout_by_Iin=10^(-Ro_dB/10);
printf("\n\n\t For Optical Bandwidth, Iout/Iin = %0.1f .",Iout_by_Iin);
```

Scilab code Exa 7.7 Determination of optical output power modulated at frequencies

```
//Example 7.7
//Determine the optical output power modulated at frequencies
//(a)20 MHz
//(b)100 MHz
//Also determine electrical and optical bandwidths

```

```
clear;
clc ;
close ;

//Given data
P_dc=300*10^(-6);            //Watt - OPTICAL OUTPUT POWER
tau_i=5*10^(-9);             //s - CARRIER RECOMBINATION LIFETIME

//(a)Optical output power at 20 MHz
f=20*10^6;                   //Hz - OPERATING FREQUENCY
Pe=P_dc/sqrt(1+(2*%pi*f*tau_i)^2);
printf("\n\n\t (a)Optical output power at %1.0f MHz, Pe(%1.0f MHz) = %0.2f uW.",f/10^6,f/10^6,Pe/10^(-6));

//(b)Optical output power at 100 MHz
f=100*10^6;                  //Hz - OPERATING FREQUENCY
Pe=P_dc/sqrt(1+(2*%pi*f*tau_i)^2);
printf("\n\n\t (b)Optical output power at %1.0f MHz, Pe(%1.0f MHz) = %0.2f uW.",f/10^6,f/10^6,Pe/10^(-6));

//Optical Bandwidth
Bopt=sqrt(3)/(2*%pi*tau_i);
printf("\n\n\t Optical Bandwidth, Bopt = %0.1f MHz.",Bopt/10^6);

//Electrical Bandwidth
B=Bopt/sqrt(2);
printf("\n\n\t Electrical Bandwidth, B = %0.1f MHz.",B/10^6);
```

Scilab code Exa 7.8 Estimation of the CW operating lifetime for the given LED

```
//Example 7.8
//Program to estimate the CW operating lifetime for
   the given LED

clear;
clc ;
close ;

//Given data
Ea=1*1.602*10^(-19);   //Joules - ACTIVATION ENERGY
k=1.38*10^(-23);       //m^2 kg/s - BOLTZMANN's
   CONSTANT
T=290;                 //Kelvin - JUNCTION
   TEMPERATURE
Pe_by_Pout=0.67;       //Pe/Pout RATIO
Beeta_o=1.84*10^7;     //per  h - CONSTANT OF
   PROPORTIONALITY

//Degradation Rate
Beeta_r=Beeta_o*exp(-Ea/(k*T));

//CW operating lifetime for the given LED
t=log(Pe_by_Pout)/-Beeta_r;

//Displaying the Result in Command Window
printf("\n\n\t CW operating lifetime for the given
   LED is %0.1f X 10^9 h.",t/10^9);
```

Chapter 8

OPTICAL DETECTORS

Scilab code Exa 8.1 Determination of the quantum efficiency and responsivity of th

```
//Example 8.1
//Program to determine the Quantum efficiency and Responsivity of
//the photodiode

clear;
clc ;
close ;

//Given data
Lambda=0.85*10^-6;                //metres - WAVELENGTH
e=1.602*10^(-19);                 //Coulumbs - CHARGE OF AN ELECTRON
h= 6.626*10^(-34);                //J/K - PLANK's CONSTANT
c=2.998*10^8;                     //m/s - VELOCITY OF LIGHT IN VACCUM
Ne=1.2*10^11;                     //NUMBER OF ELECTRONS COLLECTED
Np=3*10^11;                       //NUMBER OF
```

```
    INCIDENT PHOTONS

//Quantum Efficiency
eeta=Ne/Np;

//Responsivity
R=eeta*e*Lambda/(h*c);

//Displaying the Results in Command Window
printf("\n\n\t Quantum Efficiency = %0.1f .",eeta);
printf("\n\n\t Responsivity, R = %0.3f A/W .",R);
```

Scilab code Exa 8.2 Determination of operating wavelength and incident optical pow

```
//Example 8.2
//Program to determine:
//(a)Operating Wavelength
//(b)Incident Optical Power

clear;
clc ;
close ;

//Given data
eeta=0.65;                      //*100 percent - QUANTUM EFFICIENCY
e=1.602*10^(-19);               //Coulumbs - CHARGE OF AN ELECTRON
h=6.626*10^(-34);               //J/K - PLANK's CONSTANT
c=2.998*10^8;                   //m/s - VELOCITY OF LIGHT IN VACCUM
Ip=2.5*10^(-6);                 //A - PHOTOCURRENT
E=1.5*10^(-19);                 //J - ENERGY OF PHOTONS
```

```

//(a)Operating Wavelength
Lambda=h*c/E;

//Responsivity
R=eeta*e*Lambda/(h*c);

//(b)Incident Optical Power
Po=Ip/R;

//Displaying the Results in Command Window
printf("\n\n\t (a)Operating Wavelength = %0.2f um.",
   Lambda/10^(-6));
printf("\n\n\t (b)Incident Optical Power = %0.2f uW.
   ",Po/10^(-6));
```

Scilab code Exa 8.3 Determination of wavelength above which an intrinsic photodete

```
//Example 8.3
//Program to determine the wavelength above which an
    intrinsic
//photodetector will cease to operate

clear;
clc ;
close ;

//Given data
e=1.602*10^(-19);                   //Coulumbs - CHARGE
    OF AN ELECTRON
h=6.626*10^(-34);                   //J/K - PLANK's
   CONSTANT
c=2.998*10^8;                       //m/s - VELOCITY OF
    LIGHT IN VACCUM
Eg=1.43*e;                          //V - BANDGAP
```

```
    ENERGY

//Wavelength determination
Lambda_c=h*c/Eg;

//Displaying the Result in Command Window
printf("\n\n\t The wavelength above which an
   intrinsic photodetector will cease to operate is
   %0.2f um.",Lambda_c/10^(-6));
```

Scilab code Exa 8.4 Determination of drift time of the carriers and junction capac

```
//Example 8.4
//Program to determine:
//(a)Drift time of the carriers
//(b)Junction capacitance of the photodiode

clear;
clc ;
close ;

//Given data
w=20*10^(-6);             //metre - WIDTH OF INTRINSIC
    REGION
r=500*10^(-6);            //metre - RADIUS
epsilon_s=10.5*10^(-11);//F/m - PERMITTIVITY
vd=10^5;                  //m/s - DRIFT VELOCITY OF
   ELECTRONS

//(a)Drift time of the carriers
t_drift=w/vd;

//(b)Junction capacitance of the photodiode
A=%pi*r^2;
Cj=epsilon_s*A/w;
```

```

//Displaying the Results in Command Window
printf("\n\n\t (a)Drift time of the carriers is %1.0
    f ps.",t_drift/10^(-12));
printf("\n\n\t (b)Junction capacitance of the
    photodiode is %1.0f pF.",Cj/10^(-12));
```

Scilab code Exa 8.5 Determination of maximum response time for the device

```
//Example 8.5
//Program to determine maximum response time for the
     device

clear;
clc ;
close ;

//Given data
w=25*10^(-6);               //metre - WIDTH OF DEPLETION
     REGION
vd=3*10^4;                  //m/s - DRIFT VELOCITY OF
    CARRIER

//Maximum 3 dB Bandwidth
Bw=vd/(2*%pi*w);

//Maximum response time
t=1/Bw;

//Displaying the Result in Command Window
printf("\n\n\t Maximum response time for the device
    is %0.1f ns.",t/10^(-9));
```

Scilab code Exa 8.6 Calculation of noise equivalent power and specific detectivity

```
//Example 8.6
//Program to calculate the noise equivalent power and specific
//detectivity for the device

clear;
clc ;
close ;

//Given data
Id=8*10^(-9);               //A - DARK CURRENT
eeta=0.55;                  //*100 - QUANTUM EFFICIENCY
Lambda=1.3*10^(-6);         //metre - OPERATING WAVELENGTH
A=100*50*(10^(-6))^2;       //m^2 - AREA
e=1.602*10^(-19);           //Coulumbs - CHARGE OF AN ELECTRON
h= 6.626*10^(-34);          //J/K - PLANK's CONSTANT
c=2.998*10^8;               //m/s - VELOCITY OF LIGHT IN VACCUM

//Noise equivalent power
NEP=h*c*sqrt(2*e*Id)/(eeta*e*Lambda);

//Specific detectivity
D=sqrt(A)/NEP;

//Displaying the Results in Command Window
printf("\n\n\t Noise equivalent power = %0.2f X 10^(-14) W.",NEP/10^(-14));
printf("\n\n\t Specific detectivity = %0.1f X 10^8 m H^(1/2)/W.",D/10^(8));
```

Scilab code Exa 8.7 Determination of the multiplication factor of the photodiode

```
//Example 8.7
//Program to determine the multiplication factor of the photodiode

clear;
clc ;
close ;

//Given data
eeta=0.80;                          //*100 percent - QUANTUM EFFICIENCY
e=1.602*10^(-19);                   //Coulumbs - CHARGE OF AN ELECTRON
h=6.626*10^(-34);                   //J/K - PLANK's CONSTANT
c=2.998*10^8;                       //m/s - VELOCITY OF LIGHT IN VACCUM
Lambda=0.9*10^(-6);                 //metre - OPERATING WAVELENGTH
I=11*10^(-6);                       //A - OUTPUT CURRENT
Po=0.5*10^(-6);                     //Watt - INCIDENT OPTICAL POWER

//Responsivity
R=eeta*e*Lambda/(h*c);
//Photocurrent
Ip=Po*R;
//Multiplication Factor
M=I/Ip;

//Displaying the Result in Command Window
printf("\n\n\t The multiplication factor of the photodiode is approximately %1.0f.",M);
```

Scilab code Exa 8.8 `Determination of optical gain of the device and common emitter`

```
//Example 8.8
//Program to determine:
//(a)Optical gain of the device
//(b)Common emitter current gain

clear;
clc ;
close ;

//Given data
eeta=0.40;                          //*100 percent - QUANTUM EFFICIENCY
e=1.602*10^(-19);                   //Coulumbs - CHARGE OF AN ELECTRON
h=6.626*10^(-34);                   //J/K - PLANK's CONSTANT
c=2.998*10^8;                       //m/s - VELOCITY OF LIGHT IN VACCUM
Lambda=1.26*10^(-6);                //metre - OPERATING WAVELENGTH
Ic=15*10^(-3);                      //A - COLLECTOR CURRENT
Po=125*10^(-6);                     //Watt - INCIDENT OPTICAL POWER

//(a)Optical Gain
Go=h*c*Ic/(Lambda*e*Po);

//(b)Common emitter current gain
h_FE=Go/eeta;

//Displaying the Results in Command Window
```

```
printf("\n\n\t (a)Optical Gain, Go = %0.1f.",Go);
printf("\n\n\t (b)Common emitter current Gain, h_FE
    = %0.1f.",h_FE);
```

Scilab code Exa 8.9 Determination of the maximum 3 dB bandwidth permitted by the d

```
//Example 8.9
//Program to determine the maximum 3 dB bandwidth
    permitted by the
//device

clear;
clc ;
close ;

//Given data
tf=5*10^(-12);            //second - ELECTRON TRANSIT
    TIME
G=70;                     //PHOTOCONDUCTIVE GAIN

//Maximum 3 dB bandwidth permitted by the MSM
Bm=1/(2*%pi*tf*G);

//Displaying the Result in Command Window
printf("\n\n\t Maximum 3 dB bandwidth permitted by
    the device is %0.1f MHz.",Bm/10^6);
```

Chapter 9

DIRECT DETECTION RECEIVER PERFORMANCE CONSIDERATIONS

Scilab code Exa 9.1 Determination of the theoretical quantum limit at the receiver

```
//Example 9.1
//Note: MAXIMA SCILAB TOOLBOX REQUIRED FOR THIS
   PROGRAM
//Program to determine:
//(a)The theoretical quantum limit at the receiver
   in terms of quantum
//efficiency and energy of incident photon
//(b)The minimum incident optical power

clear;
clc ;
close ;

syms h f eeta;

//(a)The theoretical quantum limit at the receiver
   in terms of quantum
```

```
//efficiency and energy og incident photon
BER=10^(-9);                    //BIT ERROR RATE
z_min=-log(BER)
E_min=z_min*h*f/eeta;
disp(E_min,"(a)The theoretical quantum limit at the receiver in terms of quantum efficiency and energy of incident photon is =");
printf(" which is equivalent to %0.1f h*f/eeta.",z_min);

//(b)The minimum incident optical power
h1= 6.626*10^(-34);                //J/K - PLANK's CONSTANT
f1=2.998*10^14;                    //Hz - FREQUENCY
Bt=10*10^6;                        //bit/s - SIGNALING RATE
eeta1=1;                           //*100 percent - QUANTUM EFFICIENCY
Po_binary=z_min*h1*f1*Bt/(2*eeta1);
Po=10*log10(Po_binary/10^(-3));
printf("\n\n (b)The minimum incident optical power is %0.1f pW or %0.1f dBm.",Po_binary/10^(-12),Po);
```

Scilab code Exa 9.2 Calculation of incident optical power to achieve given SNR

```
//Example 9.2
//Program to calculate incident optical power to achieve given SNR

clear;
clc ;
close ;

//Given data
```

```
SNR=50;                              //dB - SIGNAL TO NOISE RATIO GIVEN
h= 6.626*10^(-34);                   //J/K - PLANK's CONSTANT
Lambda=1*10^(-6);                    //metre - OPERATING WAVELENGTH
c=2.998*10^8;                        //m/s - VELOCITY OF LIGHT IN VACCUM
B=5*10^6;                            //MHz - POST DETECTION BANDWIDTH
eeta=1;                              //*100 percent - QUANTUM EFFICIENCY

//Incident optical power to achieve given SNR
Po=2*h*c*B*10^(SNR/10)/(eeta*Lambda);

//Displaying the Result in Command Window
printf("\n\n The incident optical power is %0.1f nW or %0.1f dBm.",Po/10^(-9),10*log10(Po/10^(-3)));
```

Scilab code Exa 9.3 Comparision of the shot noise generated in the photodetector w

```
//Example 9.3
//Program to compare the shot noise generated in the photodetector
//with the thermal noise in the load resistor

clear;
clc ;
close ;

//Given data
Id=3*10^(-9);                        //A - DARK CURRENT
e=1.602*10^(-19);                    //Coulumbs - CHARGE OF AN ELECTRON
```

```
h= 6.626*10^(-34);                    //J/K - PLANK's CONSTANT
Lambda=0.9*10^(-6);                   //metre - OPERATING WAVELENGTH
c=2.998*10^8;                         //m/s - VELOCITY OF LIGHT IN VACCUM
eeta=0.6;                             //*100 percent - QUANTUM EFFICIENCY
Po=200*10^(-9);                       //Watt- INCIDENT OPTICAL POWER
k=1.381*10^(-23);                     //m^2 kg/s - BOLTZMANN's CONSTANT
T=293;                                //Kelvin - TEMPERATURE
B=5*10^6;                             //Hz - BANDWIDTH OF RECEIVER
Rl=4*10^3;                            //Ohms - LOAD RESISTANCE

//RMS shot noise current
Ip=eeta*Po*e*Lambda/(h*c);
Shot_noise_current=sqrt(2*e*B*(Id+Ip));

//RMS thermal noise current
Thermal_noise_current=sqrt(4*k*T*B/Rl);

//Displaying the Results in Command Window
printf("\n\n RMS shot noise current = %0.3f X 10^(-10) A.",Shot_noise_current/10^(-10));
printf("\n\n RMS thermal noise current = %0.3f X 10^(-9) A.",Thermal_noise_current/10^(-9));
```

Scilab code Exa 9.4 Determination of SNR at the output of the receiver

```
//Example 9.4
```

```
//Program to determine SNR at the output of the receiver

clear;
clc ;
close ;

//Given data
Id=3*10^(-9);                    //A - DARK CURRENT
e=1.602*10^(-19);                //Coulumbs - CHARGE OF AN ELECTRON
h= 6.626*10^(-34);               //J/K - PLANK's CONSTANT
Lambda=0.9*10^(-6);              //metre - OPERATING WAVELENGTH
c=2.998*10^8;                    //m/s - VELOCITY OF LIGHT IN VACCUM
eeta=0.6;                        //*100 percent - QUANTUM EFFICIENCY
Po=200*10^(-9);                  //Watt- INCIDENT OPTICAL POWER
k=1.381*10^(-23);                //m^2 kg/s - BOLTZMANN's CONSTANT
T=293;                           //Kelvin - TEMPERATURE
B=5*10^6;                        //Hz - BANDWIDTH OF RECEIVER
Rl=4*10^3;                       //Ohms - LOAD RESISTANCE
Fn=3;                            //dB - AMPLIFIER NOISE FIGURE

//RMS shot noise current
Ip=eeta*Po*e*Lambda/(h*c);
Shot_noise_current=sqrt(2*e*B*(Id+Ip));
//RMS thermal noise current
Thermal_noise_current=sqrt(4*k*T*B/Rl);

```

```
//SNR Calculation
SNR=Ip^2/(Shot_noise_current^2+Thermal_noise_current
    ^2*10^(Fn/10));

//Displaying the Result in Command Window
printf("\n\n SNR = %0.2f dB.",10*log10(SNR));
```

Scilab code Exa 9.5 Calculation of maximum load resistance and bandwidth penalty c

```
//Example 9.5
//Program to:
//(i)Calculate Maximum Load Resistance
//(ii)Determine Bandwidth Penalty considering
    amplifier capacitance

clear;
clc ;
close ;

//Given data
Cd=6*10^(-12);              //Farad - PHOTODIODE
    CAPACITANCE
Ca=6*10^(-12);              //Farad - AMPLIFIER INPUT
    CAPACITANCE
B=8*10^6;                   //Hz - POST DETECTION
    BANDWIDTH

//(i)Maximum Load Resistance
Rl=1/(2*%pi*Cd*B);

//(ii)Maximum Bandwidth considering amplifier
    capacitance
Bm=1/(2*%pi*Rl*(Cd+Ca));

//Displaying the Results in Command Window
```

```
printf("\n\n\t (i)Maximum Load Resistance, Rl(max) = %0.2f kiloOhms.",Rl/10^3);
printf("\n\n\t(ii)Maximum Bandwidth considering amplifier capacitance, B = %1.0f MHz.",Bm/10^6);
```

Scilab code Exa 9.6 Determination of the maximum SNR improvement

```
//Example 9.6
//Program to determine the maximum SNR improvement between
//M=1 and M=Mop

clear;
clc ;
close ;

//Given data
Cd=5*10^(-12);                    //Farad - APD CAPACITANCE
B=50*10^6;                        //Hz - POST DETECTION BANDWIDTH
T=291;                            //Kelvin - TEMPERATURE
k=1.381*10^(-23);                 //m^2 kg/s - BOLTZMANN's CONSTANT
Id=0;                             //A - DARK CURRENT
x=0.3;
Fn=1;                             //dB - AMPLIFIER NOISE FIGURE
e=1.602*10^(-19);                 //Coulumbs - CHARGE OF AN ELECTRON
Ip=10^(-7);                       //A - PHOTOCURRENT

//Maximum Load Resistance
Rl=1/(2*%pi*Cd*B);
```

```
//For M=1
M=1
SNR1=Ip^2*M^2/(2*e*B*(Ip+Id)*M^(2+x)+4*k*T*B*Fn/Rl);
//Displaying the Result in Command Window
printf("\n\n For M = 1, SNR = %0.2f dB.",10*log10(
    SNR1));

//For M=Mop
Mop=(4*k*T/(x*e*Rl*Ip))^(1/(2+x));
M=Mop;
SNR2=Ip^2*M^2/(2*e*B*(Ip+Id)*M^(2+x)+4*k*T*B*Fn/Rl);
//Displaying the Result in Command Window
printf("\n\n For M = Mopt, SNR = %0.2f dB.",10*log10
    (SNR2));
printf("\n\n SNR Improvement = %0.2f dB.",10*log10(
    SNR2)-10*log10(SNR1));
```

Scilab code Exa 9.7 Determination of the optimum avalanche multiplication factor

```
//Example 9.7
//Program to determine the optimum avalanche
    multiplication factor

clear;
clc ;
close ;

//Given data
Rl=10*10^3;                          //Ohms - LOAD
    RESISTANCE
T=120;                               //Kelvin -
    TEMPERATURE
SNR=35;                              //dB - SIGNAL TO
    NOISE RATIO
```

```
Fn=1;                              //dB - AMPLIFIER NOISE FIGURE
B=10*10^6;                         //Hz - POST DETECTION BANDWIDTH
x=1;
k=1.381*10^(-23);                  //m^2 kg/s - BOLTZMANN's CONSTANT
e=1.602*10^(-19);                  //Coulumbs - CHARGE OF AN ELECTRON

//As Ip=10*Id, Minimum Photo Current
Ip=(10^(SNR/10)*(12*k*T*B*10^(Fn/10)/Rl)/(4*k*T*10^(Fn/10)/(1.1*e*Rl))^(2/(2+x)))^(3/4);

//Optimum avalanche multiplication factor
Mop=(4*k*T*10^(Fn/10)/(e*Rl/10*1.1*Ip))^(1/(2+x));

//Displaying the Result in Command Window
printf("\n\n Optimum avalanche multiplication factor, Mop = %0.2f.",Mop);
```

Scilab code Exa 9.8 Determination of Maximum bandwidth Mean square thermal noise c

```
//Example 9.8
//Program to determine:
//(a)Maximum bandwidth without equilization
//(b)Mean square thermal noise current per unit bandwidth
//(c)(Compare (a) and (b) for transimpedance amplifier

clear;
clc ;
close ;

```

```
//Given data
Ra=4*10^6;                          //Ohms - INPUT
    RESISTANCE
Rb=4*10^6;                          //Ohms - DETECTOR
    BIAS RESISTANCE
Ct=6*10^(-12);                      //Farad - TOTAL
    CAPACITANCE
k=1.381*10^(-23);                   //m^2 kg/s -
    BOLTZMANN's CONSTANT
T=300;                              //Kelvin -
    TEMPERATURE
Rf=100*10^3;                        //Ohms - LOAD
    RESISTANCE
G=400;                              //OPEN LOOP GAIN OF
    TRANSIMPEDANCE AMP.

//Total effective load resistance
Rtl=Rb*Ra/(Rb+Ra);

//(a)Maximum bandwidth without equilization
B=1/(2*%pi*Rtl*Ct)

//(b)Mean square thermal noise current per unit
    bandwidth
it_sq_bar=4*k*T/Rtl;

//(c)(Compare (a) and (b) for transimpedance
    amplifier
B1=G/(2*%pi*Rf*Ct)
it_sq_bar1=4*k*T/Rf;

//Displaying the Results in Command Window
printf("For High Gain Transimpedance Amplifier:")
printf("\n\n (a)Maximum bandwidth without
    equilization, B = %0.2f X 10^4 Hz.",B/10^4);
printf("\n\n (b)Mean square thermal noise current
    per unit bandwidth, it_sq_bar = %0.2f X 10^(-27)
    A^2/Hz.",it_sq_bar/10^(-27));
```

```
printf("\n\n (c)For High Gain Transimpedance
    Amplifier:")
printf("\n\n     Maximum bandwidth without
    equilization , B = %0.2f X 10^8 Hz.",B1/10^8);
printf("\n\n     Mean square thermal noise current
    per unit bandwidth, it_sq_bar = %0.2f X 10^(-25)
    A^2/Hz.",it_sq_bar1/10^(-25));
printf("\n\n Mean square thermal noise current for
    transimpedance amplifier is %1.0f times or %1.0f
    dB greater.",it_sq_bar1/it_sq_bar,10*log10(
    it_sq_bar1/it_sq_bar));
```

Chapter 10

OPTICAL AMPLIFICATION WAVELENGTH CONVERSION AND REGENERATION

Scilab code Exa 10.1 Determination of Refractive Index of active medium and 3dB Sp

```
//Example 10.1
//Program to determine the Refractive Index of the Active Medium and
//the 3dB spectral bandwidth of the device

clear;
clc ;
close ;

//Given data
L=300*10^-6;            //metres - ACTIVE REGION LENGTH
Lambda=1.5*10^-6;       //metres - PEAK GAIN WAVELENGTH
Delta_Lambda=1*10^-9; //metres - MODE SPACING
```

```
c= 2.998*10^8;          //m/s - SPEED OF LIGHT
Gs_dB=4.8;              //dB     - SINGLE PASS GAIN
R1=0.3;                 //INPUT FACET REFRACTIVITY
R2=0.3;                 //OUTPUT FACET REFRACTIVITY

//Refractive Index of the active medium at the peak gain wavelength
n=(Lambda^2)/(2*Delta_Lambda*L);

//Gain Gs from Gs_dB by taking antilog with base 10
Gs=10^((1/10)*Gs_dB);

//3dB spectral Bandwidth
B_fpa=(c/(%pi*n*L))*asin((1-sqrt(R1*R2)*Gs)/(2*sqrt(sqrt(R1*R2)*Gs)));

//Displaying the Results in Command Window
printf("\n\n\t Refractive Index of the active medium at the peak gain wavelength is %0.2f .",n);
printf("\n\n\t 3dB spectral Bandwidth is %0.1f GHz .",B_fpa/10^9);
```

Scilab code Exa 10.2 Derivation of an approximate equation for the cavity gain of

```
//Example 10.2
//Note: MAXIMA SCILAB TOOLBOX REQUIRED FOR THIS PROGRAM
//Program to derive an approximate equation for the cavity gain
//of an SOA

clear;
clc ;
close ;

```

```
syms R1 R2;

//For 3 dB peak through ratio
//Let A=sqrt(R1*R2)*Gs
A=(1-sqrt(0.5))/(1+sqrt(0.5));

//Cavity gain
G=A/(1-A)^2/sqrt(R1*R2);;

//Displaying the Result in Command Window
disp(G,"The approximate equation of cavity gain is, G = ")
```

Scilab code Exa 10.3 Determination of the length of the device and the ASE noise s

```
//Example 10.3
//Program to determine:
//(a)The length of the device
//(b)The ASE noise signal power at the output of the amplifier

clear;
clc ;
close ;

//Given data
Gs_dB=30;             //dB - SINGLE PASS GAIN
g_bar=200;            //NET GAIN COEFFICIENT
m=2.2;                //MODE FACTOR
n_sp=4;               //SPONTANEOUS EMISSION FACTOR
h= 6.626*10^(-34);    //J/K - PLANK's CONSTANT
c=2.998*10^8;         //m/s - VELOCITY OF LIGHT IN VACCUM
B=1*10^(12);          //Hz - OPTICAL BANDWIDTH
Lambda=1.55*10^(-6);  //metre - OPERATING WAVELENGTH
```

```

//(a)The length of the device
L=Gs_dB/(10*g_bar*log10(%e));

//(b)The ASE noise signal power at the output of the amplifier
Gs=10^(Gs_dB/10);
f=c/Lambda;
P_ASE=m*n_sp*(Gs-1)*h*f*B;

//Displaying the Results in Command Window
printf("\n\n\t (a)The length of the SOA is %0.2f X 10^(-3) m.",L/10^(-3));
printf("\n\n\t (b)The ASE noise signal power at the output of the amplifier, P_ASE = %0.2f mW.",P_ASE/10^(-3));
```

Scilab code Exa 10.4 Determination of the fiber non linear coefficient and the par

```
//Example 10.4
//Program to determine:
//(a)The fiber non-linear coefficient
//(b)The parametric gain in dB when it is reduced to quadratic gain

clear;
clc ;
close ;

//Given data
L=500;                  //metre - LENGTH
Lambda=1.55*10^(-6);    //metre - OPERATING WAVELENGTH
Pp= 1.4;                //W - SIGNAL POWER
Gp_dB=62.2;             //dB - PEAK GAIN

```

```
//(a)The fiber non-linear coefficient
gaamma=(Gp_dB-10*log10(1/4))/(Pp*L)*1/(10*log10((%e)^2));

//(b)The parametric gain in dB when it is reduced to quadratic gain
Gp_dB1=10*log10((gaamma*Pp*L)^2);

//Displaying the Results in Command Window
printf("\n\n\t (a)The fiber non-linear coefficient is %0.2f X 10^(-3) per W per km.",gaamma/10^(-3));
printf("\n\n\t (b)The parametric gain in dB when it is reduced to quadratic gain is %0.1f dB.",Gp_dB1);
```

Scilab code Exa 10.5 Calculation of the frequency chirp variation at the output si

```
//Example 10.5
//Program to calculate:
//(a)The frequency chirp variation at the output signal
//(b)The differential gain required

clear;
clc ;
close ;

//Given data
Lambda=1.55*10^(-6);      //metre - OPERATING WAVELENGTH
alpha=-1;                 //ENHANCEMENT FACTOR
Pin=0.5*10^(-3);          //Watt - INPUT SIGNAL POWER
dPin_by_dt=0.01*10^(-6);  //metre - INPUT SIGNAL POWER VARIATION
```

```
dnr_by_dn=-1.2*10^(-26); //m^3 - DIFFERENTIAL
    REFRATIVE INDEX

//(a)The frequency chirp variation at the output
    signal
del_f=-alpha/(4*%pi)*1/Pin*dPin_by_dt;

//(b)The differential gain required
dg_by_dn=4*%pi/Lambda*dnr_by_dn/alpha;

//Displaying the Results in Command Window
printf("\n\n\t (a)The frequency chirp variation at
    the output signal is %0.2f X 10^(-6)Hz.",del_f
    /10^(-6));
printf("\n\n\t (b)The differential gain required is
    %0.3f X 10^(-20) m^2.",dg_by_dn/10^(-20));
```

Chapter 11

INTEGRATED OPTICS AND PHOTONICS

Scilab code Exa 11.1 `Determination of Voltage required to provide pi radians phase`

```
//Example 11.1
//Program to determine the Voltage required for a phase change of
//pi radians

clear;
clc ;
close ;

//Given data
L=2*10^-2;              //metres - LENGTH OF THE WAVEGUIDE
Lambda=1.3*10^-6;       //metres - WAVELENGTH
d=25*10^-6;             //metres - DISTANCE BETWEEN THE ELECTRODES
r=30.8*10^-12;          //m/V    - ELECTRO-OPTIC COEFFICIENT
n1=2.1;                 //REFRACTIVE INDEX AT 1.3um WAVELENGTH
```

```
//Calculation of the Voltage required for a phase
   change of pi radians
V_pi=(Lambda*d)/((n1^3)*r*L);

//Displaying the Result in Command Window
printf("\n\n\t Voltage required for a phase change
   of pi radians is %0.1f V.",V_pi);
```

Scilab code Exa 11.2 Determination of Corrugation Period and Filter 3dB Bandwidth

```
//Example 11.2
//Program to determine Corrugation Period and Filter
   's 3dB Bandwidth

clear;
clc ;
close ;

//Given data
L=1*10^-2;              //metres - LENGTH OF THE
   DEVICE
Lambda_B=1.52*10^-6;    //metres - CENTRE WAVELENGTH
Theeta=1;               //Degree - INCIDENT ANGLE
n1=3.1;                 //REFRACTIVE INDEX of InGaAsP

//Calculation of Effective Refractive Index of the
   Waveguide
ne=n1*sin(2*Theeta*%pi/180);

//Calculation of the Corrugation Period
D=(Lambda_B)/(2*ne);

//Calculation of the Filter's 3dB Bandwidth
delta_Lambda=(D*Lambda_B)/L;
```

```
//Displaying the Results in Command Window
printf("\n\n\t Corrugation Period of the First Order Grating is %0.1f um.",D/10^-6);
printf("\n\n\t Filters 3dB Bandwidth is %0.1f Armstrong.",delta_Lambda/10^-10);
```

Scilab code Exa 11.3 Design of a wavelength channel plan for a dense WDM Interleav

```
//Example 11.3
//Program to design a wavelength channel plan for an 8 band, 32
//channel dense WDM Interleaver Waveband Filter. Also to determine:
//(a)Total No. of channel required for each interleaver band filter
//(b)The overall bandwidth of the filter in each case

clear;
clc ;
close ;

//Given data
number_of_bands=8;
M=4;                    //TOTAL NUMBER OF CHANNELS IN EACH BAND

//(a)Total No. of channel required for each interleaver band filter
//(i)N=0
N=0;
Cskip0=(number_of_bands-1)*N;
Ctotal0=number_of_bands*M+Cskip0;
//Displaying the Result in Command Window
```

```
printf("\n\n\t (a)(i)For 4-skip-0, Ctotal = %d.", Ctotal0);

//(ii)N=1
N=1;
Cskip1=(number_of_bands-1)*N;
Ctotal1=number_of_bands*M+Cskip1;
//Displaying the Result in Command Window
printf("\n\n\t   (ii)For 4-skip-1, Ctotal = %d.", Ctotal1);

//(iii)N=2
N=2;
Cskip2=(number_of_bands-1)*N;
Ctotal2=number_of_bands*M+Cskip2;
//Displaying the Result in Command Window
printf("\n\n\t  (iii)For 4-skip-2, Ctotal = %d.", Ctotal2);

//Generation of Table 11.1
printf("\n\n\n\t\t TABLE 11.1:WAVELENFTH CHANNEL PLAN");
W1=1552.52;           //nm - WAVELENGTH FOR 1 CHANNEL
printf("\n\n\n\t\t (i)4-skip-0");
for i = 0:Ctotal0-1
printf("\n\t Number of Channels = %d, Wavelength = %0.2f nm.",i+1,W1+0.8*i);
end
printf("\n\n\n\t\t (ii)4-skip-1");
for i = 0:Ctotal1-1
printf("\n\t Number of Channels = %d, Wavelength = %0.2f nm.",i+1,W1+0.8*i);
end
printf("\n\n\n\t\t (iii)4-skip-2");
for i = 0:Ctotal2-1
printf("\n\t Number of Channels = %d, Wavelength = %0.2f nm.",i+1,W1+0.8*i);
end
```

```

//(b)The overall bandwidth of the filter in each
    case taking values
//from Table 11.1
//(i)N=0
W2=1577.32;          //nm - WAVELENGTH FOR 32
    CHANNELS
BW=W2-W1;
//Displaying the Result in Command Window
printf("\n\n\n\t (b)(i)For 4-skip-0, Filter
    Bandwidth = %0.1f nm.",BW);

//(ii)N=1
W2=1582.92;          //nm - WAVELENGTH FOR 39
    CHANNELS
BW=W2-W1;
//Displaying the Result in Command Window
printf("\n\n\t   (ii)For 4-skip-1, Filter Bandwidth
    = %0.1f nm.",BW);

//(iii)N=2
W2=1588.52;          //nm - WAVELENGTH FOR 46
    CHANNELS
BW=W2-W1;
//Displaying the Result in Command Window
printf("\n\n\t  (iii)For 4-skip-2, Filter Bandwidth
    = %0.1f nm.",BW);
```

Chapter 12

OPTICAL FIBER SYSTEMS 1 INTENSITY MODULATION AND DIRECT DETECTION

Scilab code Exa 12.1 Determination of bit rate and duration of a Time slot Frame a

```
//Example 12.1
//Program to determine:
//(a)Bit rate for the system
//(b)The duration of a time slot
//(c)The duration of a frame and multiframe

clear;
clc ;
close ;

//Given data
f=8*10^3;                //Hz - SAMPLING RATE
b=8;                     //bits - SAMPLE SIZE
T=32;                    //NUMBER OF TIME SLOTS

//(a)Bit rate for the system
Number_of_bits=T*b;
```

```
Bit_rate=f*Number_of_bits
//(b)The duration of a time slot
Bit_duration=1/Bit_rate;
Slot_duration=b*Bit_duration;
//(c)The duration of a frame and multiframe
Duration_of_frame=T*Slot_duration;
Duration_of_multiframe=T/2*Duration_of_frame;

//Displaying The Results in Command Window
printf("\n\n\t  (a)Bit rate for the system is %0.3f Mbit/s.",Bit_rate/10^6);
printf("\n\n\t  (b)The duration of a time slot is %0.1f us.",Slot_duration/10^(-6));
printf("\n\n\t  (c)The duration of a frame is %1.0f us and multiframe is %1.0f ms.",Duration_of_frame/10^(-6),Duration_of_multiframe/10^(-3));
```

Scilab code Exa 12.2 Determination of required electrical and optical SNR

```
//Example 12.2
//Program to determine the required electrical and optical SNR

clear;
clc ;
close ;

//Given data
BER=10^(-9);                    //BIT ERROR RATE

//Optical SNR
SNR_op=(erfinv(1-2*BER))*2*sqrt(2); //erfc(x)=1-erf(x)

//Electrical SNR
```

```
SNR_el=((erfinv(1-2*BER))*2*sqrt(2))^2; //erfc(x)=1-erf(x)

//Displaying the Results in Command Window
printf("\n\n\t Optical SNR is %1.0f or %0.1f dB.",SNR_op,10*log10(SNR_op));
printf("\n\n\t Electrical SNR is %1.0f or %0.1f dB.",SNR_el,10*log10(SNR_el));
```

Scilab code Exa 12.3 Estimation of the average number of photons which must be inc

```
//Example 12.3
//Program to estimate the average number of photons which must be
//incident on the APD to register a binary one with a BER of 10^(-9)

clear;
clc ;
close ;

//Given data
k=0.02;                 //CARRIER IONIZATION RATE
M=100;                  //MULTIPLICATION FACTOR
SNR=144;                //SIGNAL TO NOISE RATIO
Bt=0.6;                 //FOR RAISED COSINE PULSE SPECTRUM
n=0.8;                  //(*100) percent - QUANTUM EFFICIENCY

//Excess avalanche noise factor F(M)
F=k*M+(2-1/M)*(1-k);

//Average number of photons
z=2*Bt*ceil(F)/n*SNR;
```

```

//Displaying the Result in Command Window
printf("\n\n\t  The average number of photons which must be incident on the APD is %1.0f photons.",z);
```

Scilab code Exa 12.4 Estimation of incident optical power to register binary 1 at

```
//Example 12.4
//Program to estimate incident optical power to register binary 1
//at bit rates of 10 Mbit/s and 140 Mbit/s

clear;
clc ;
close ;

//Given data
BER=10^(-9);                          //BIT ERROR RATE
e=1.602*10^(-19);                     //Coulumbs - CHARGE OF AN ELECTRON
Lambda=1*10^(-6);                     //metre - WAVELENGTH
h= 6.626*10^(-34);                    //J/K - PLANK's CONSTANT
c=2.998*10^8;                         //m/s - VELOCITY OF LIGHT IN VACCUM
zm=864;                               //photons - FROM EXAMPLE 12.3

//For 10 Mbit/s
Bt=10*10^6;                           //bps - BIT RATES
Po=zm*h*c*Bt/(2*Lambda);
//Displaying the Result in Command Window
printf("\n\n\t  Incident optical power for %1.0f
```

```
    Mbit/s is %0.1f pW or %0.1f dBm.",Bt/10^6,Po
    /10^(-12),10*log10(Po/10^(-3)));

//For 140 Mbit/s
Bt=140*10^6;                    //bps - BIT RATES
Po=zm*h*c*Bt/(2*Lambda);
//Displaying the Result in Command Window
printf("\n\n\t Incident optical power for %1.0f
    Mbit/s is %0.3f nW or %0.1f dBm.",Bt/10^6,Po
    /10^(-9),10*log10(Po/10^(-3)));
```

Scilab code Exa 12.5 Determination of the total channel loss ignoring dispersion

```
//Example 12.5
//Program to determine the total channel loss
    ignoring dispersion

clear;
clc ;
close ;

//Given data
alpha_fc=5;              //dB/km - FIBER CABLE
    ATTENUATION
alpha_j=2;               //dB/km - SPLICE LOSS
alpha_s=3.5;             //dB - SOURCE CONNECTOR
    LOSS
alpha_d=2.5;             //dB - DETECTOR CONNECTOR
     LOSS
L=4;                     //km - LENGTH OF OPTICAL
    FIBER LINK

//Total channel loss
alpha_cr=alpha_s+alpha_d
C_L=(alpha_fc+alpha_j)*L+alpha_cr;
```

```
//Displaying The Result in Command Window
printf("\n\n\tTotal channel loss , C_L = %1.0f dB", C_L)
```

Scilab code Exa 12.6 Estimation of the dispersion equalization penalty for bit giv

```
//Example 12.6
//Program to estimate the dispersion-equalization penalty for bit
//rates:
//(a)25 Mbit/s
//(b)150 Mbit/s

clear;
clc ;
close ;

//Given data
L=8;                        //km - LENGTH OF FIBER LINK
sigma=0.6*10^(-9);          //s/km - RMS PULSE BROADENING


//(a)For 25 Mbit/s
Bt=25*10^6;                  //bit/sec - BIT RATE
//Without mode coupling
sigma_T=sigma*L;
D_L=2*(2*sigma_T*Bt*sqrt(2))^4;
printf("\n\n\t (a)For Bt = %1.0f Mbit/s, Without mode coupling , D_L = %0.2f dB",Bt/10^6,D_L);
//With mode coupling
sigma_T=sigma*sqrt(L);
D_L=2*(2*sigma_T*Bt*sqrt(2))^4;
printf("\n\n\t     For Bt = %1.0f Mbit/s, With mode
```

```
coupling , D_L = %0.2f X 10^(-4) dB",Bt/10^6,D_L/10^(-4));

//(b)150 Mbit/s
Bt=150*10^6;                    //bit/sec - BIT RATE
//Without mode coupling
sigma_T=sigma*L;
D_L=2*(2*sigma_T*Bt*sqrt(2))^4;
printf("\n\n\t (b)For Bt = %1.0f Mbit/s, Without mode coupling, D_L = %0.2f dB",Bt/10^6,D_L);
//With mode coupling
sigma_T=sigma*sqrt(L);
D_L=2*(2*sigma_T*Bt*sqrt(2))^4;
printf("\n\n\t     For Bt = %1.0f Mbit/s, With  mode coupling, D_L = %0.2f dB",Bt/10^6,D_L);
```

Scilab code Exa 12.7 Estimation of the maximum bit rate that may be achieved on th

```
//Example 12.7
//Program to estimate the maximum bit rate that may be achieved on
//the link when using NRZ format

clear;
clc ;
close ;

//Given data
L=8;                              //km - LENGTH OF FIBER LINK
Ts=8*10^(-9);                     //s - SOURCE RISE TIME
Dn=5*10^(-9);                     //s/km - INTERMODAL RISE TIME
Dc=1*10^(-9);                     //s/km - INTRAMODAL
```

```
    RISE TIME
Td=6*10^(-9);                          //s - DETECTOR RISE
    TIME
Tn=Dn*L;
Tc=Dc*L;

//Total Rise Time
Tsyst=1.1*sqrt(Ts^2+Tn^2+Tc^2+Td^2);

//Maximum bit rate
Bt= 0.7/Tsyst;

//Displaying the Result in Command Window
printf("\n\n\t Maximum bit rate, Bt(max) is %0.1f
    Mbit/s which for NRZ is equivalent to a 3 dB
    optical bandwidth of %0.1f Mbit/s.",Bt/10^6,Bt
    /10^6/2);
```

Scilab code Exa 12.8 Estimation of maximum possible link length without repeaters

```
//Example 12.8
//Program to estimate:
//(a)Maximum possible link length without repeaters
    when operating at 35 Mbit/s
//(b)Maximum possible link length without repeaters
    when operating at 400 Mbit/s
//(c)Reduction in maximum possible link length
    considering dispersion-equalization penalty

clear;
clc ;
close ;

//Given data
Pi=-3;                    //dBm - POWER LAUNCHED
```

```
alpha_fc=0.4;                  //dB/km - CABLE FIBER LOSS
alpha_j=0.1;                   //dB/km - SPLICE LOSS
alpha_cr=2;                    //dB - TOTAL CONNECTOR LOSS
Ma=7;                          //dB - REQUIRED SAFETY MARGIN
Dl=1.5;                        //dB - DISPERSION-EQUALIZATION PENALTY

//(a)Maximum possible link length without repeaters when operating at 35 Mbit/s
Po=-55;                        //dBm - REQUIRED POWER BY APD
//Optical budget: Pi-Po=(alpha_fc+alpha_j)L+alpha_cr+Ma
L1=(Pi-Po-alpha_cr-Ma)/(alpha_fc+alpha_j);
printf("\n\n\t (a)Maximum possible link length without repeaters when operating at 35 Mbit/s is %1.0f km.",L1);

//(b)Maximum possible link length without repeaters when operating at 400 Mbit/s
Po=-44;                        //dBm - REQUIRED POWER BY APD
//Optical budget: Pi-Po=(alpha_fc+alpha_j)L+alpha_cr+Ma
L2=(Pi-Po-alpha_cr-Ma)/(alpha_fc+alpha_j);
printf("\n\n\t (b)Maximum possible link length without repeaters when operating at 400 Mbit/s is %1.0f km.",L2);

//(c)Reduction in maximum possible link length considering dispersion-equalization penalty
//Optical budget considering dispersion-equalization penalty:
//Pi-Po=(alpha_fc+alpha_j)L+alpha_cr+Ma
L3=(Pi-Po-alpha_cr-Dl-Ma)/(alpha_fc+alpha_j);
printf("\n\n\t (c)Reduction in maximum possible link
```

```
length considering dispersion-equalization
penalty is %1.0f km.",L2-L3);
```

Scilab code Exa 12.9 Determination of the viability of optical power budget

```
//Example 12.9
//Program to determine the viability of optical
    power budget

clear;
clc ;
close ;

//Given data
L=7;                        //km - OPTICAL FIBER LINK
    LENGTH
alpha_fc=2.6;               //dB/km - CABLE FIBER LOSS
alpha_j=0.5;                //dB/km - SPLICE LOSS
alpha_cr=1.5;               //dB - TOTAL CONNECTOR
    LOSS
Ma=6;                       //dB - REQUIRED SAFETY
    MARGIN
Pr_dBm=-41;                 //dBm - RECEIVER
    SENSITIVITY
Pi=100*10^(-6);             //Watt - POWER LAUNCHED
Pi_dBm=10*log10(Pi/10^(-3));

//Total System Margin
Total_system_margin=Pi_dBm-Pr_dBm;
printf("\n\n\t Total System Margin is %0.1f dB.",
    Total_system_margin);

//Total System Loss
Total_system_loss=L*alpha_fc+(L-1)*alpha_j+alpha_cr+
    Ma;
```

```
printf("\n\n\t Total System Loss is %0.1f dB.",Total_system_loss);

//Excess Power margin
Excess_power_margin=Total_system_margin-Total_system_loss;
printf("\n\n\t Excess Power margin is %0.1f dB.",Excess_power_margin);

//Testing Viability
if Excess_power_margin >=0 then
printf("\n\n\t The system is viable.");
else
printf("\n\n\t The system is not viable.");
end
```

Scilab code Exa 12.10 Estimation of ratio of SNR of the coaxial system to the SNR

```
//Example 12.10
//Program to estimate ratio of SNR of the coaxial system to the SNR
//of the fiber system

clear;
clc ;
close ;

//Given data
V=5;                  //volts - TRANSMITTER PEAK OUTPUT VOLTAGE
Zo=100;               //ohms - CABLE IMPEDANCE
T=290;                //Kelvin - OPERATING TEMPERATURE
lambda=0.85*10^(-6);  //metre - WAVELENGTH
K=1.38*10^(-23);      //J/K - BOLTZMANN's CONSTANT
```

```
n=0.7;                    //(*100) percent - QUANTUM EFFICIENCY
Pi=1*10^(-3);             //Watts - OPTICAL POWER
h=6.626*10^(-34);         //(m^2)Kg/s - PLANK's CONSTANT
c=2.998*10^8;             //m/s - SPEED OF LIGHT

//Ratio SNR(coax)/SNR(fiber)
Ratio=V^2*h*c/(2*K*T*Zo*n*Pi*lambda);

//Displaying the Result in Command Window
printf("\n\n\t SNR(coax)/SNR(fiber) = %d dB.",10*log10(Ratio));
```

Scilab code Exa 12.11 Determination of the average incident optical power required

```
//Example 12.11
//Program to determine the average incident optical power required at
//the receiver

clear;
clc ;
close ;

//Given data
k=1.38*10^(-23);            //J/K - BOLTZMANN's CONSTANT
e=1.602*10^(-19);           //Coulumbs - CHARGE OF AN ELECTRON
SNR_dB=55;                  //dB - SIGNAL POWER TO RMS NOISE RATIO
ma=0.8;                     //MODULATION INDEX
Id=0;                       //A - DARK CURRENT
T=293;                      //K - OPERATING TEMPERATURE
B=5*10^6;                   //Hz - BANDWIDTH
```

```
Fn_dB=1.5;                  //dB - NOISE FIGURE
Rl=1*10^6;                  //Ohms - EFFECTIVE INPUT IMPEDANCE
R=0.5;                      //A/W - RESPONSIVITIY
b=0.7;                      //RATIO OF LUMINANCE TO COMPOSITE VIDEO
SNR=10^(SNR_dB/10);
Fn=10^(Fn_dB/10);

//Photo-current, Ip=R*Po Ip=Po*R;
//(SNR)p-p=(2*ma*Ip*b)^2/(2*e*B*(Ip+Id)+(4*k*T*B*Fn/Rl));
//Rearranging and solving the quadratic equation, Incident Power
Po=((SNR*2*e*B*R)+sqrt((SNR*2*e*B*R)^2-4*(2*ma*R*b)^2*(SNR*(-4*k*T*B*Fn/Rl))))/(2*(2*ma*R*b)^2);

//Displaying the Result in Command Window
printf("\n\n\t The average incident optical power required at the receiver is %0.2f uW or %0.1f dBm.",Po/10^(-6),10*log10(Po/10^(-3)));
```

Scilab code Exa 12.12 Determination of the average incident optical power required

```
//Example 12.12
//Program to determine the average incident optical power required to
//maintain given SNR

clear;
clc ;
close ;

//Given data
Lambda=1*10^(-6);                       //metre -
```

```
WAVELENGTH
h= 6.626*10^(-34);                //J/K - PLANK's CONSTANT
c=2.998*10^8;                     //m/s - VELOCITY OF LIGHT IN VACCUM
k=1.38*10^(-23);                  //J/K - BOLTZMANN's CONSTANT
e=1.602*10^(-19);                 //Coulumbs - CHARGE OF AN ELECTRON
eeta=0.6;                         //*100 percent - QUANTUM EFFICIENCY
SNR_dB=45;                        //dB - CURRENT SNR
Rl=50*10^3;                       //Ohms - EFFECTIVE LOAD IMPEDANCE
T=300;                            //K - OPERATING TEMPERATURE
ma=0.5;                           //MODULATION INDEX
Fn_dB=6;                          //dB - NOISE FIGURE
B=10*10^6;                        //Hz - POST DETECTION BANDWIDTH

SNR=10^(SNR_dB/10);
Fn=10^(Fn_dB/10);

//Average incident optical power required to maintain given SNR
Po=h*c/(e*eeta*ma^2*Lambda)*sqrt(8*k*T*Fn/Rl)*sqrt(SNR*B);

//Displaying the Result in Command Window
printf("\n\n\t The average incident optical power required at the receiver is %0.2f uW or %0.1f dBm.",Po/10^(-6),10*log10(Po/10^(-3)));
```

Scilab code Exa 12.13 Determination of the viability of optical power budget and e

```
//Example 12.13
//Program to:
//(a)Determine the viability of optical power budget
//(b)Estimate any possible increase in link length

clear;
clc ;
close ;

//Given data
L=2;                           //km - OPTICAL FIBER LINK
   LENGTH
alpha_fc=3.5;                  //dB/km - CABLE FIBER LOSS
alpha_j=0.7;                   //dB/km - SPLICE LOSS
alpha_cr=1.6;                  //dB - CONNECTOR LOSS AT
   RECEIVER
Ma=4;                          //dB - REQUIRED SAFETY
   MARGIN
Pr_dBm=-25;                    //dBm - RECEIVER
   SENSITIVITY
Pi_dBm=-10;                    //dBm - POWER LAUNCHED

//Total System Margin
Total_system_margin=Pi_dBm-Pr_dBm;
printf("\n\n\t (a)Total System Margin is %0.1f dB.",
   Total_system_margin);

//Total System Loss
Total_system_loss=L*alpha_fc+L*alpha_j+alpha_cr+Ma;
printf("\n\n\t     Total System Loss is %0.1f dB.",
   Total_system_loss);

//Excess Power margin
Excess_power_margin=Total_system_margin-
   Total_system_loss;
printf("\n\n\t     Excess Power margin is %0.1f dB.",
   Excess_power_margin);

```

```
//(a)Testing Viability
if Excess_power_margin >=0 then
printf("\n\n\t    The system is viable.");
else
printf("\n\n\t    The system is not viable.");
end

//(b)Maximum possible link length
Pi=0;                        //dBm - LAUNCHED POWER
Po=-25;                      //dBm - REQUIRED POWER BY APD
Ma=7;                        //dB - SAFETY MARGIN
//Optical budget: Pi-Po=(alpha_fc+alpha_j)L+alpha_cr+Ma
L1=(Pi-Po-alpha_cr-Ma)/(alpha_fc+alpha_j);
printf("\n\n\t (b)Maximum possible increase in link length is %0.1f km.",L1-L);
```

Scilab code Exa 12.14 Determination of whether the combination of components gives

```
//Example 12.14
//Program to determine whether the combination of components gives
//an adequate temporal response

clear;
clc ;
close ;

//Given data
L=5;                         //km - LENGTH OF FIBER LINK
Ts=10*10^(-9);               //s - SOURCE RISE TIME
Dn=9*10^(-9);                //s/km - INTERMODAL
```

```
    RISE TIME
Dc=2*10^(-9);                    //s/km - CHROMATIC
    RISE TIME
Td=3*10^(-9);                    //s - DETECTOR RISE
    TIME
Bopt=6*10^6;                     //Hz - REQUIRED
    OPTICAL BANDWIDTH

Tn=Dn*L;
Tc=Dc*L;

//Maximum permitted rise time
Tsyst_max=0.35/Bopt;

//Total system rise time
Tsyst=1.1*sqrt(Ts^2+Tn^2+Tc^2+Td^2);

//Displaying the Results in Command Window
printf("\n\n\t Maximum permitted rise time, Tsyst(
    max) = %0.1f ns.",Tsyst_max/10^(-9));
printf("\n\n\t Total system rise time, Tsyst = %0.1f
    ns.",Tsyst/10^(-9));
printf("\n\n\t Hence system gives adequate temporal
    response.");
```

Scilab code Exa 12.15 Derivation of an expression for the improvement in post dete

```
//Example 12.15
//Note: MAXIMA SCILAB TOOLBOX REQUIRED FOR THIS
    PROGRAM
//Program to:
//(a)Derive an expression for the improvement in
    post detection SNR
//(b)Determine the improvement in post detection SNR
    and Bandwidth
```

```
clear;
clc ;
close ;

//(a)Derive an expression for the improvement in post detection SNR
//Symbolic Representation
syms Pa R Po Ba No Df
//D-IM OUTPUT SNR
SNR_DIM=(R*Po)^2*Pa/(2*Ba*No);
//FM OUTPUT SNR
SNR_FM=3*Df^2*(R*Po)^2*Pa/(4*Ba*No);
//SNR IMPROVEMENT
SNR_imp=SNR_FM/SNR_DIM;
//SNR IMPROVEMENT IN dB
SNR_imp_dB=10*log10(SNR_imp);
disp(SNR_imp,"SNR IMPROVEMENT = ");
disp(SNR_imp_dB,"SNR IMPROVEMENT IN dB = ");
printf("\n\n\t The above expression is equivalent to 1.76+20*log10(Df)");

//(b)Determine the improvement in post detection SNR and Bandwidth
//Given data
fd1=400*10^3;                    //Hz - PEAK FREQUENCY DEVIATION
Ba1=4*10^3;                      //Hz- BANDWIDTH
//Frequency Deviation Ratio
Df1=fd1/Ba1;
//SNR Improvement expression from part(a)
SNR_imp_dB1=1.76+20*log10(Df1);
//Bandwidth
Bm=2*(Df1+1)*Ba1;
printf("\n\n\t The SNR Improvement = %0.2f dB.",SNR_imp_dB1);
printf("\n\n\t The Bandwidth of FM-IM, Bm = %1.0f kHz.",Bm/10^3);
```

Scilab code Exa 12.16 Program to determine the ratio of SNRs of FM IM and PM IM sy

```
//Example 12.16
//Note: MAXIMA SCILAB TOOLBOX REQUIRED FOR THIS
   PROGRAM
//Program to determine the ratio of SNRs of FM-IM
   and PM-IM systems

clear;
clc ;
close ;

//Symbolic representation
syms fd Pa R Po Ac Ba No

//FOR FM-IM
Df=fd/Ba;    //Frequency Deviation
SNR_FM=3*Df^2*Pa*(R*Po)^2*Ac^2/2/(2*Ba*No);

//FOR PM-IM
Dp=fd/Ba;    //Frequency Deviation
SNR_PM=Df^2*Pa*(R*Po)^2*Ac^2/2/(2*Ba*No);

//Determining Ratio
Ratio=SNR_FM/SNR_PM;

//Displaying the Result in Command Window
disp(Ratio," SNR_FM/SNR_PM = ");
```

Scilab code Exa 12.17 Calculation of the optimium receiver bandwidth and the peak

```
//Example 12.17
//Program to calculate:
//(a)The optimium receiver bandwidth
//(b)The peak to peak signal power to rms noise ratio

clear;
clc ;
close ;

//Given data
Tr=12*10^(-9);                //s - SYSTEM RISE TIME
fo=20*10^6;                   //Hz - NOMINAL PULSE RATE
fd=5*10^6;                    //Hz - PEAK TO PEAK FREQUECY DEVIATION
M=60;                         //APD MULTIPLICATION FACTOR
R=0.7;                        //APD RESPONSIVITY
B=6*10^6;                     //Hz - BASEBAND NOISE BANDWIDTH
Ppo=10^(-7);                  //Watt - PEAK OPTICAL POWER
in_sq_bar=10^(-17);           //A^2 - RECEIVER MEAN SQUARE NOISE CURRENT

//(a)The optimium receiver bandwidth
Bopt=1/Tr;
To=1/fo;

//(b)The peak to peak signal power to rms noise ratio
SNR=3*(To*fd*M*R*Ppo)^2/((2*%pi*Tr*B)^2*in_sq_bar);

//Displaying the Results in Command Window
printf("\n\n\t (a)The optimium receiver bandwidth is %0.1f MHz.",Bopt/10^6);
printf("\n\n\t (b)The peak to peak signal power to
```

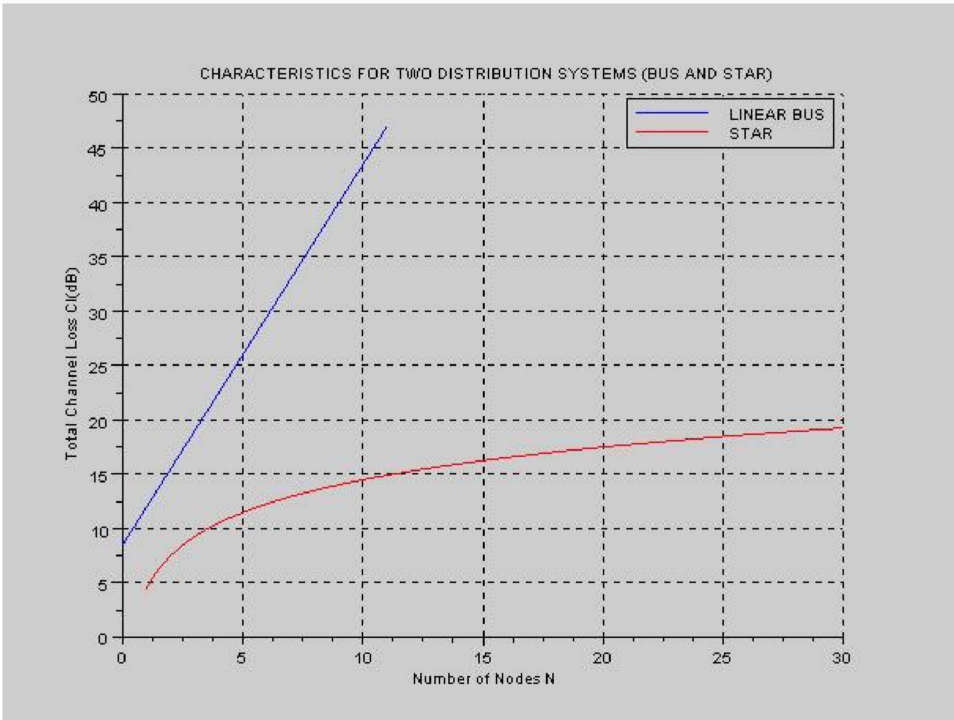

Figure 12.1: Formation of comparision showing total channel loss against number of nodes for Bus and Star Distribution Systems

```
rms noise ratio is %0.1f dB.",10*log10(SNR));
```

Scilab code Exa 12.18 Formation of comparision showing total channel loss against

```
//Example 12.18
//Program to form  comparision showing total channel
    loss against
//number of nodes for:
//(i)Bus Distribution System
//(ii)Star Distribution System
```

```
clear;
clc ;
close ;

//Given data
alpha_cr=1;                         //dB - CONNECTOR LOSS
alpha_fc=5;                         //dB/km - FIBER CABLE LOSS
L_bu=0.1                            //m - FIBER LENGTH
L_tr=10;                            //dB - ACCESS COUPLER TAP RATIO
L_sp=3;                             //dB - SPLITTER LOSS
L_ac=1;                             //dB - ACCESS COUPLER INSERTION LOSS
L_st=0.1;                           //m - TOTAL FIBER LENGTH IN STAR ARMS
L_ex=0;                             //dB - STAR COUPLER EXCESS LOSS

//Bus Distribution System
N=0:0.01:11;
Cl_bus=2*alpha_cr+(N-1)*alpha_fc*L_bu+(2*alpha_cr+
    L_ac)*(N-3)+(2*alpha_cr+L_tr)+L_sp+alpha_cr;
Hm=abs(Cl_bus);
figure;
plot2d(N,Hm,2);

//Star Distribution System
N=1:0.01:30;
Cl_star=4*alpha_cr+alpha_fc*L_st+10*log10(N)+L_ex;
Hm=abs(Cl_star);
plot2d(N,Hm,5);
xlabel('Number of Nodes N');
ylabel('Total Channel Loss Cl(dB)');
title('CHARACTERISTICS FOR TWO DISTRIBUTION SYSTEMS
    (BUS AND STAR)');
```

```
xgrid (1);
h=legend(['LINEAR BUS';'STAR']);
```

Scilab code Exa 12.19 Estimation of the maximum system length for satisfactory per

```
//Example 12.19
//Program to estimate the maximum system length for satisfactory
//performance

clear;
clc ;
close ;

//Given data
SNR_dB=17;                //dB - REQUIRED SNR
L=100*10^3;               //metre - INTERVAL SPACING
K=4;                      //FOR AMPLIFIER
h= 6.626*10^(-34);        //J/K - PLANK's CONSTANT
c=2.998*10^8;             //m/s - VELOCITY OF LIGHT IN VACCUM
B=1.2*10^(9);             //bit/s - TRANSMISSION RATE
Pi_dBm=0;                 //dBm - INPUT POWER
Lambda=1.55*10^(-6);      //metre - OPERATING WAVELENGTH
alpha_fc=0.22;            //dB/km - FIBER CABLE ATTENUATION
alpha_j=0.03;             //dB/km - SPLICE LOSS

//Calculation of SNR and Pi
SNR=10^(SNR_dB/10);
Pi=10^(Pi_dBm/10)*10^(-3);

//Maximum system length
Lto=(Pi*Lambda*10^(-(alpha_fc+alpha_j)*L/10/10^3)/(K
```

```
*h*c*B))/SNR*L;

//Displaying the Result in Command Window
printf("\n\n\t Maximum system length for satisfactory performance is %1.0f X 10^4 km.",Lto/10^7);
```

Scilab code Exa 12.20 `Obtain an expression for the total noise figure for the syst`

```
//Example 12.20
//Note: MAXIMA SCILAB TOOLBOX REQUIRED FOR THIS PROGRAM
//Program to obtain an expression for the total noise figure for the
//system

clear;
clc ;
close ;

//Symbolic representation
syms F G k M;

//Given data
//F_to = F1*G1 + F2*G2 + F3*G3 +......+ FM*GM
//For Identical Repeaters :
//F1*G1 = F2*G2 = F3*G3 =......= FM*GM = F*G(say)
x=F*G;
F_to = symsum(x,k,1,M);

//Displaying The Results in Command Window
disp (F_to,"TOTAL NOISE FIGURE: F_to = ");
disp ("At the output from the first amplifier repeater, a degradation in SNR of F*G occurs followed by a decrease of 1/M");
```

Scilab code Exa 12.21 Calculation of second order dispersion coefficient for L1 an

```
//Example 12.21
//Program to :
//(a)Calculate second order dispersion coefficient for L1
//(b)Determine the dispersion slope for L2
//(c)Verify that periodic dispersion management map will provide
//sufficient coincidence to facilitate reliable DWDM transmission

clear;
clc ;
close ;

//Given data
L1=160;                //km - PATH LENGTH
L2=20;                 //km - PATH LENGTH

//(a)To calculate second order dispersion coefficient for L1
Beeta22=17;            //ps/nm/km - 2nd ORDER DISPERSION COEFF. FOR L2
Beeta21=-Beeta22*L2/L1;
printf("\n\n\t(a)The second order dispersion coefficient for L1 is %0.3f ps/nm/km",Beeta21);

//(b)To determine the dispersion slope for L2
S1=0.075;              //ps/nm^2/km - DISPERSION SLOPE FOR L1
S2=-S1*L1/L2;
printf("\n\n\t(b)The dispersion slope for L2 is %0.1f ps/nm^2/km",S2);
```

```
//(c)To verify that periodic dispersion management map will provide
//sufficient coincidence to facilitate reliable DWDM transmission
OP=S1*(L1/L2)+S1*(Beeta22/Beeta21);
if OP==0 then
printf("\n\n\t(c)Periodic dispersion management map will provide sufficient coincidence to facilitate reliable DWDM transmission as S1(L1/L2)+S1(Beeta22/Beeta21)=0");
else
printf("\n\n\t(c)Periodic dispersion management map will not provide sufficient coincidence to facilitate reliable DWDM transmission as S1(L1/L2)+S1(Beeta22/Beeta21)!=0");
end
```

Scilab code Exa 12.22 Determination of the separation for the soliton pulses to av

```
//Example 12.22
//Program to determine
//(a)The separation for the soliton pulses to avoid interaction
//(b)The transmission bit rate of the optical soliton communication
//system

clear;
clc ;
close ;

//Given data
To=70*10^(-12);                    //s - BIT PERIOD
```

```
tau=6*10^(-12);                              //s - PULSE WIDTH
Beeta2=-0.5*10^(-12)*10^(-12)*10^(-3);//s^2/km - 2nd ORDER DISPERSION
                                             //COEFFICIENT
La=50*10^3;                                  //AMPLIFIER SPACING

//(a)The separation for the soliton pulses to avoid interaction
qo=1/2*(To/tau);
//(b)The transmission bit rate of the optical soliton comm. system
Bt=1/(2*qo)*1/sqrt(abs(Beeta2)*La);

//Displaying the Results in Command Window
printf("\n\n\t(a)The separation for the soliton pulses to avoid interaction is %0.1f .",qo);
printf("\n\n\t(b)The maximum bit rate of the optical soliton communication system is much less than %0.2f Gbit/s .",Bt/10^9);
```

Scilab code Exa 12.23 Determination of the maximum transmission bit rate for the s

```
//Example 12.23
//Program to determine the maximum transmission bit rate for the
//system

clear;
clc ;
close ;

//Given data
```

```
To=40*10^(-12);                        //s - BIT PERIOD
tau=4*10^(-12);                        //s - PULSE WIDTH
Beeta2=-1.25*10^(-12)*10^(-12)*10^(-3); //s^2/km - 2 nd ORDER
                                       //DISPERSION COEFFICIENT
alpha=0.2*10^(-3);                     //dB/m - ATTENUATION CONSTANT

//The separation for the soliton pulses to avoid interaction
qo=1/2*(To/tau);

//Maximum transmission bit rate for the system
Bt=1/(2*qo)*sqrt(alpha/abs(Beeta2));

//Displaying the Result in Command Window
printf("\n\n\t The maximum bit rate of the ultrashort pulse optical soliton system is significantly greater than %1.0f Gbit/s .",Bt/10^9);
```

Chapter 13

OPTICAL FIBER SYSTEMS 2 COHERENT AND PHASE MODULATED

Scilab code Exa 13.1 Estimation of the maximum temperature change that could be al

```
//Example 13.1
//Program to estimate the maximum temperature change that could
//be allowed for the local oscillator laser

clear;
clc ;
close ;

//Given data
IF=1.5*10^6;            //Hz - NOMINAL IF
del_f=19*10^6;          //Hz/C - OUTPUT FREQUENCY CHANGE

//Maximum temperature change that could be allowed
f=0.1*IF;
Max_temp_change=f/del_f;
```

```
//Displaying the Result in Command Window
printf("\n\n\t Maximum temperature change that could
    be allowed for the local oscillator laser is %0
   .3f C .",Max_temp_change);
```

Scilab code Exa 13.2 Determination of the operating bandwidth of the receiver

```
//Example 13.2
//Program to determine the operating bandwidth of
   the receiver

clear;
clc ;
close ;

//Given data
SNL=-85.45;            //dBm - SHOT NOISE LIMIT
eeta=0.86;             //*100 percent - EFFICIENCY FOR
    IDEAL RECEIVER
Lambda=1.54*10^(-6);  //metre - OPERATING WAVELENGTH
SNR=12;                //dB - SIGNAL TO NOISE RATIO
h= 6.626*10^(-34);    //J/K - PLANK's CONSTANT
c=2.998*10^8;          //m/s - VELOCITY OF LIGHT IN
   VACCUM

//Incoming Signal Power
Ps=10^(SNL/10);

//Operating bandwidth of the receiver
B=eeta*Ps*Lambda/(h*c*10^(SNR/10));

//Displaying the Result in Command Window
printf("\n\n\t Operating bandwidth of the receiver ,
   B = %0.1f GHz.",B/10^9);
```

Scilab code Exa 13.3 Calculation of the number of received photons per bit for dif

```
//Example 13.3
//Program to calculate the number of received
  photons per bit for:
//(a)ASK heterodyne synchronous detection
//(b)ASK heterodyne asynchronous detection
//(c)PSK homodyne detection

clear;
clc ;
close ;

//Given data
eeta=1;                //*100 percent - EFFICIENCY FOR
    IDEAL RECEIVER
BER=10^(-9);           //BIT ERROR RATE

//Number of received photons per bit for:
printf("\n\n\t Number of received photons per bit
  for:");
//(a)ASK heterodyne synchronous detection
Np=(erfinv(1-2*BER))^2*4/eeta; //erfc(x)=1-erf(x)

//Displaying the Result in Command Window
printf("\n\n\t (a)ASK heterodyne synchronous
  detection = %1.0f.",Np/2);

//(b)ASK heterodyne asynchronous detection
Np=-log(2*BER)*4/eeta;

//Displaying the Result in Command Window
printf("\n\n\t (b)ASK heterodyne asynchronous
  detection = %1.0f.",Np/2);
```

```
//(c)PSK homodyne detection
Np=(erfinv(1-2*BER))^2/2; //erfc(x)=1-erf(x)

//Displaying the Result in Command Window
printf("\n\n\t (c)PSK homodyne detection = %1.0f.",
   Np);
```

Scilab code Exa 13.4 Calculation of the minimum incoming power level

```
//Example 13.4
//Program to calculate the minimum incoming power
   level

clear;
clc ;
close ;

//Given data
K=1;                    //CONSTANT FOR HETERODYNE
   DETECTION
Z=1;                    //CONSTANT FOR FSK MODULAION
   SCHEME
eeta=1;                 //*100 percent - QUANTUM
   EFFICIENCY
Bt=400*10^6;            //bps - TRANSMISSION RATE
BER=10^(-9);            //BIT ERROR RATE
h= 6.626*10^(-34);      //J/K - PLANK's CONSTANT
c=2.998*10^8;           //m/s - VELOCITY OF LIGHT IN
   VACCUM
Lambda=1.55*10^(-6); //metre - OPERATING WAVELENGTH

//Minimum incoming peak power level
Ps=(erfinv(1-2*BER))^2*2*h*c*Bt/Lambda; //erfc(x)=1-
   erf(x)
```

```

//Displaying the Result in Command Window
printf("\n\n\t Minimum incoming peak power level is %0.1f nW or %0.1f dBm.",Ps/10^(-9),10*log10(Ps/(1*10^(-3))));
```

Scilab code Exa 13.5 Calculation of the absolute maximum repeater spacing for diff

```
//Example 13.5
//Program to calculate the absolute maximum repeater spacing for the
//following ideal receiver types:
//(a)ASK heterodyne synchronous detection
//(b)PSK homodyne detection

clear;
clc ;
close ;

//Given data
Np=36;                 //Average photons per bit - FROM EXAMPLE 13.3
h= 6.626*10^(-34);     //J/K - PLANK's CONSTANT
c=2.998*10^8;          //m/s - VELOCITY OF LIGHT IN VACCUM
Lambda=1.55*10^(-6);   //metre - OPERATING WAVELENGTH

//(a)ASK heterodyne synchronous detection
Np=36;                 //Average photons per bit - FROM EXAMPLE 13.3
//For 50 Mbit/s Transmission Rate
Bt=50*10^6;            //bit/sec - GIVEN TRANSMISSION RATE
Ps=Np*h*c*Bt/Lambda;
Max_system_margin=4-10*log10(Ps/(1*10^(-3)));
```

```
Max_repeater_spacing=Max_system_margin/0.2;
//Displaying the Result in Command Window
printf("\n\n\t (a)ASK : Maximum repeater spacing for %1.0f Mbit/s transmission rate is %1.0f km.",Bt/10^6,Max_repeater_spacing);

//For 1 Gbit/s Transmission Rate
Bt=1*10^9;          //bit/sec - GIVEN TRANSMISSION RATE
Ps=Np*h*c*Bt/Lambda;
Max_system_margin=4-10*log10(Ps/(1*10^(-3)));
Max_repeater_spacing=Max_system_margin/0.2;
//Displaying the Result in Command Window
printf("\n\n\t          Maximum repeater spacing for %1.0f Gbit/s transmission rate is %1.0f km.",Bt/10^9,Max_repeater_spacing);

//(b)PSK homodyne detection
Np=9;          //Average photons per bit - FROM EXAMPLE 13.3
//For 50 Mbit/s Transmission Rate
Bt=50*10^6;          //bit/sec - GIVEN TRANSMISSION RATE
Ps=Np*h*c*Bt/Lambda;
Max_system_margin=4-10*log10(Ps/(1*10^(-3)));
Max_repeater_spacing=Max_system_margin/0.2;
//Displaying the Result in Command Window
printf("\n\n\t (b)PSK : Maximum repeater spacing for %1.0f Mbit/s transmission rate is %1.0f km.",Bt/10^6,Max_repeater_spacing);

//For 1 Gbit/s Transmission Rate
Bt=1*10^9;          //bit/sec - GIVEN TRANSMISSION RATE
Ps=Np*h*c*Bt/Lambda;
Max_system_margin=4-10*log10(Ps/(1*10^(-3)));
Max_repeater_spacing=Max_system_margin/0.2;
//Displaying the Result in Command Window
```

```
printf("\n\n\t             Maximum repeater spacing for %1.0f Gbit/s transmission rate is %1.0f km.",Bt/10^9,Max_repeater_spacing);
```

Scilab code Exa 13.6 Estimation of the minimum transmitter power requirement for a

```
//Example 13.6
//Program to estimate the minimum transmitter power requirement for
//an optical coherent WDM

clear;
clc ;
close ;

//Given data
Np=150;                  //photons per bit - RECEPTION
h= 6.626*10^(-34);       //J/K - PLANK's CONSTANT
c=2.998*10^8;            //m/s - VELOCITY OF LIGHT IN VACCUM
B_fib=20*10^12;          //Hz - OPTICAL BANDWIDTH
Lambda=1.3*10^(-6);      //metre - SHORTEST WAVELENGTH

//Minimum transmitter power requirement for an optical coherent WDM
Ptx=Np*h*c*B_fib/Lambda;

//Displaying the Result in Command Window
printf("\n\n\t Minimum transmitter power requirement for an optical coherent WDM is %0.1f mW or %1.0f dBm .",Ptx/10^(-3), 10*log10(Ptx/(1*10^(-3))));
```

Chapter 14

OPTICAL FIBER MEASUREMENTS

Scilab code Exa 14.1 Determination of the attenuation for the fiber and estimation

```
//Example 14.1
//Program to determine the attenuation per kilometer for the fiber
//and estimate the accuracy of the result

clear;
clc ;
close ;

//Given data
L1=2*10^3;            //metres - INITIAL LENGTH
L2=2;                 //metres - FINAL LENGTH
V1=2.1;               //volts - INITIAL OUTPUT VOLTAGE
V2=10.7;              //volts - FINAL OUTPUT VOLTAGE

//Attenuation per Kilometer
alpha_dB=10/(L1-L2)*log10(V2/V1);

//Uncertainity in measured attenuation
```

```
Uncertainity=0.2/(L1-L2);

//Displaying the Results in Command Window
printf("\n\n\t Attenuation is %0.1f dB/km.",alpha_dB*10^3);
printf("\n\n\t Uncertainity in measured attenuation is +-%0.1f dB.",Uncertainity*10^3);
```

Scilab code Exa 14.2 Determination of the absorption loss for the fiber under test

```
//Example 14.2
//Program to determine the absorption loss for the fiber under test

clear;
clc ;
close ;

//Given data
t1=10;                  //s - INITIAL TIME
t2=100;                 //s - FINAL TIME
Tinf_minus_Tt1=0.525;//From Figure 14.6
Tinf_minus_Tt2=0.021;//From Figure 14.6
C=1.64*10^4;            //J/degree C - THERMAL CAPACITY PER KILOMETER
Tinf=4.3*10^(-4);       //degree C - MAXIMUM THERMAL TEMPERATURE RISE
Popt=98*10^(-3);        //Watt - OPTICAL POWER

//Time constant for the calorimeter
tc=(t2-t1)/(log(Tinf_minus_Tt1)-log(Tinf_minus_Tt2));

//Absorption loss of the test fiber
alpha_abs=C*Tinf/(Popt*tc);
```

```

//Displaying the Results in Command Window
printf("\n\n\t Time constant for the calorimeter is %0.1f s.",tc);
printf("\n\n\t Absorption loss of the test fiber is %0.1f dB/km.",alpha_abs);
```

Scilab code Exa 14.3 Determination of the loss due to scattering for the fiber

```
//Example 14.3
//Program to determine the loss due to scattering for the fiber

clear;
clc ;
close ;

//Given data
Vsc=6.14*10^(-9);                  //V - OPTICAL OUTPUT POWER
Vopt=153.38*10^(-6);               //V - OPTICAL POWER WITHOUT SCATTERING
l=2.92;                            //cm - LENGTH OF THE FIBER

//Loss due to scattering for the fiber
alpha_sc=4.343*10^5/l*Vsc/Vopt;

//Displaying the Result in Command Window
printf("\n\n\t Loss due to scattering for the fiber is %0.1f dB/km.",alpha_sc);
```

Scilab code Exa 14.4 Calculation of 3 dB Pulse Broadening and Fiber Bandwidth Leng

```
//Example 14.4
//Program to calculate:
//(a)3 dB Pulse Broadening in ns/km
//(b)Fiber Bandwidth-Length product

clear;
clc ;
close ;

//Given data
tau_o=12.6;           //ns - 3 dB width of Output Pulse
tau_i=0.3;            //ns - 3 dB width of Input Pulse
L=1.2;                //km - LENGTH

//(a)3 dB Pulse Broadening in ns/km
tau=sqrt(tau_o^2-tau_i^2)/L;

//(b)Fiber Bandwidth-Length product
Bopt=0.44/tau;

//Displaying the Results in Command Window
printf("\n\n\t (a)3 dB Pulse Broadening is %0.1f ns/km.",tau);
printf("\n\n\t (b)Fiber Bandwidth-Length product is %0.1f MHz km.",Bopt*10^3);
```

Scilab code Exa 14.5 Calculation of the Numerical Aperture of the fiber

```
//Example 14.5
//Program to calculate the Numerical Aperture(NA) of the fiber

clear;
```

```
clc ;
close ;

//Given data
D=10;                    //cm - SCREEN POSITION
A=6.2;                   //cm - OUTPUT PATTERN SIZE

// Numerical Aperture(NA) of the fiber
NA=A/sqrt(A^2+4*D^2);

//Displaying The Results in Command Window
printf("\n\n\t  The Numerical Aperture(NA) of the
   fiber is %0.2f .",NA);
```

Scilab code Exa 14.6 Determination of the outer diameter of the optical fiber in m

```
//Example 14.6
//Program to determine outer diameter of the optical
    fiber in micrometer

clear;
clc ;
close ;

//Given data
l=0.1;                   //m - MIRROR POSITION
d_PHI_by_dt=4;           //rad/s - ANGULAR VELOCITY
We=300*10^(-6);          //us - WIDTH OF SHADOW PULSE

//Outer diameter of the optical fiber
d0=We*l*d_PHI_by_dt;

//Displaying the Result in Command Window
printf("\n\n\t  The Outer diameter of the optical
   fiber is %1.0f um.",d0*10^6);
```

Scilab code Exa 14.7 Conversion of optical signal powers to dBm and dBu

```
//Example 14.7
//Program to:
//(a) Convert optical signal powers to dBm
//(b) Convert optical signal powers to dBu

clear;
clc ;
close ;

//(a)Convert optical signal powers to dBm
Po=5*10^(-3);       //Watt - GIVEN OPTICAL POWER
dBm=10*log10(Po/1*10^3);
printf("\n\n\t  (a)The %1.0f mW of optical power is
    equivalent to %0.2f dBm.",Po/10^(-3), dBm);

Po=20*10^(-6);      //Watt - GIVEN OPTICAL POWER
dBm=10*log10(Po/1*10^3);
printf("\n\n\t     The %1.0f uW of optical power is
    equivalent to %0.2f dBm.",Po/10^(-6), dBm);

//(b)Convert optical signal powers to dBu
Po=0.03*10^(-3);    //Watt - GIVEN OPTICAL POWER
dBm=10*log10(Po/1*10^6);
printf("\n\n\t  (b)The %0.2f mW of optical power is
    equivalent to %0.2f dBu.",Po/10^(-3), dBm);

Po=800*10^(-9);     //Watt - GIVEN OPTICAL POWER
dBm=10*log10(Po/1*10^6);
printf("\n\n\t     The %1.0f nW of optical power is
    equivalent to %0.2f dBu.",Po/10^(-9), dBm);
```

Scilab code Exa 14.8 Calculation of the ratio of back scattered optical power to t

```
//Example 14.8
//Program to calculate the ratio in dB of back
   scattered optical
//power to the forward optical power at the fiber
   input

clear;
clc ;
close ;

//Given data
NA=0.2;             //NUMERICAL APERTURE
gamma_r=0.7*10^-3;  //per m - RAYLEIGH SCATTERING
   COEFFICIENT
Wo=50*10^(-9);      //s - PULSE DURATION
c=2.998*10^8;       //m/s - VELOCITY OF LIGHT IN
   VACCUM
n1=1.5;             //CORE REFRACTIVE INDEX

//Calculated Ratio Pra(0)/Pi
Pra0_by_Pi=0.5*NA^2*gamma_r*Wo*c/(4*n1^3);

//Displaying the Result in command window
printf("\n\n\t  Pra(0)/Pi = %0.1f dB.",10*log10(
   Pra0_by_Pi));
```