

UNIVERSIDADE FEDERAL DE PERNAMBUCO
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
SISTEMAS DE COMUNICAÇÕES ÓPTICAS
PROF. LEONARDO DIDIER COELHO

Projeto - Sistemas de Transmissão por Fibra Óptica

Modelo Numérico para Propagação da Luz em Fibras Ópticas

A equação não-linear de Schrödinger (NLSE) é geralmente difícil de resolver analiticamente. Uma solução numérica, no entanto, pode ser muito eficiente. Na equação (1), $A(z, t) = s(z, t) \cdot \sqrt{A_{\text{eff}}/2Z_F}$ é a amplitude normalizada em W , A_{eff} é a área efetiva, Z_F é a impedância em Ω , $s(z, t)$ é o campo elétrico em V/m , β_2 é o parâmetro de dispersão de velocidade de grupo em s^2/m , β_3 é o parâmetro de dispersão de terceira ordem em s^3/m , γ é o parâmetro não-linear em $1/(W \cdot m)$ and α é o coeficiente de atenuação em m^{-1} .

$$\frac{\partial A(z, t)}{\partial z} - \frac{j}{2}\beta_2 \frac{\partial^2 A(z, t)}{\partial t^2} - \frac{1}{6}\beta_3 \frac{\partial^3 A(z, t)}{\partial t^3} = -j\gamma |A(z, t)|^2 A(z, t) - \frac{\alpha}{2} A(z, t) \quad (1)$$

O parâmetro não-linear γ é dado por

$$\gamma = \frac{2\pi n_2}{\lambda_0 A_{\text{eff}}}, \quad (2)$$

onde n_2 é o índice de refração não-linear e λ é o comprimento de onda de operação.

Método Numérico

Existem basicamente duas categorias de método para resolver (1): métodos pseudo-espectrais e métodos de diferença finita. O *Split-Step Fourier Method* (SSFM) é um método pseudo-espectral que tem sido amplamente utilizado para resolver a propagação de sinais em fibras ópticas [1]. A Figura 1 mostra esquematicamente a implementação do algoritmo para SSFM. Considerando-se uma fibra de comprimento L , o envelope de saída $A(z = L, t)$ pode ser calculado dividindo a fibra em pequenos segmentos de comprimento dz e aplicando o algoritmo SSFM para cada um.

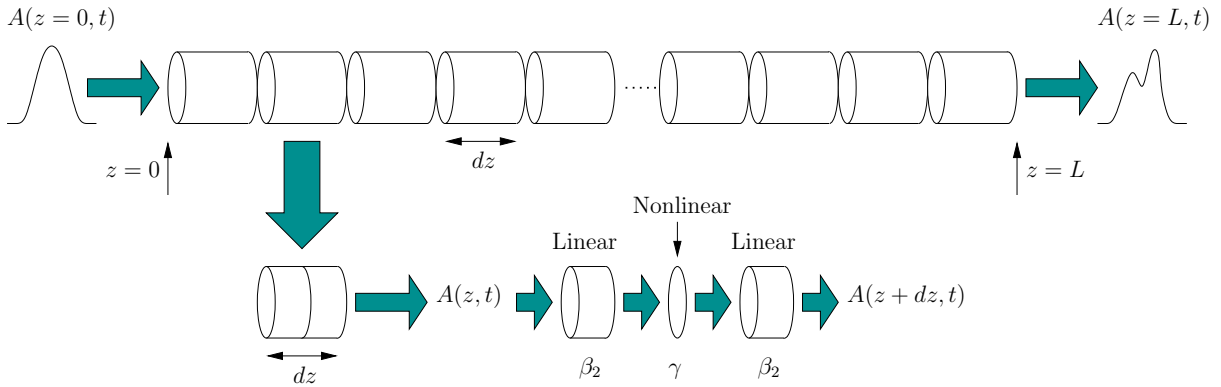


Figura 1: Algoritmo SSFM Simétrico

Portanto, (1) pode ser escrito como:

$$\begin{aligned} \frac{\partial A(z, t)}{\partial z} &= \underbrace{-\frac{\alpha}{2}A(z, t) + \frac{j}{2}\beta_2 \frac{\partial^2 A(z, t)}{\partial t^2} + \frac{1}{6}\beta_3 \frac{\partial^3 A(z, t)}{\partial t^3}}_{\text{Linear}} + \underbrace{-j\gamma|A(z, t)|^2 A(z, t)}_{\text{Nonlinear}} \\ &= (\hat{D} + \hat{N})A(z, t), \end{aligned}$$

onde \hat{D} é um operador diferencial, empregado no *domínio da frequência*, que calcula a atenuação e a dispersão na fibra, enquanto \hat{N} é um operador não-linear que calcula as não-linearidades na fibra.

$$\hat{D}(\omega) = -\frac{\alpha}{2} - \frac{j}{2}\beta_2\omega^2 - \frac{j}{6}\beta_3\omega^3 \quad (3)$$

$$\hat{N}(z, t) = -j\gamma|A(z, t)|^2 \quad (4)$$

Em cada segmento dz , pode-se assumir que os termos lineares e não-lineares atuam de forma independente.

$$\begin{aligned} \frac{\partial A}{\partial z} &= \hat{D}A + \hat{N}A \\ \frac{\partial A}{A} &= (\hat{D} + \hat{N})\partial z \\ \int \frac{\partial A}{A} &= \int (\hat{D} + \hat{N}) \partial z \\ \ln(A) &= \int \left(\frac{1}{2}\hat{D} + \hat{N} + \frac{1}{2}\hat{D} \right) \partial z \\ A(z + dz, t) &\approx \exp\left(\frac{dz}{2}\hat{D}(\omega)\right) \exp\left(\int_z^{z+dz} \hat{N}(z', t)dz'\right) \exp\left(\frac{dz}{2}\hat{D}(\omega)\right) A(z, t) \end{aligned}$$

A integral na equação anterior pode ser avaliada utilizando a fórmula trapezoidal:

$$\int_z^{z+dz} \hat{N}(z', t)dz' = \frac{dz}{2}(\hat{N}(z + dz, t) + \hat{N}(z, t)),$$

o que resulta em

$$\begin{aligned} A(z + dz, t) &\approx \exp\left(\frac{dz}{2}\hat{D}(\omega)\right) \exp\left(\frac{dz}{2}(\hat{N}(z + dz, t) + \hat{N}(z, t))\right) \\ &\quad \cdot \exp\left(\frac{dz}{2}\hat{D}(\omega)\right) A(z, t) \end{aligned} \quad (5)$$

Como $\hat{N}(z + dz, t)$ depende de $A(z + dz, t)$, a solução correta deve ser obtida usando um procedimento iterativo para determinar $A(z + dz, t)$ a partir de $A(z, t)$.

O comprimento do passo dz desempenha um papel importante na precisão do algoritmo. Encontrar uma distribuição de comprimento do passo ótimo depende das características do sistema de comunicação óptica. Existem vários critérios para a escolha do comprimento do passo dz no método SSMF [2]. O método de rotação de fase não-linear é um método onde o comprimento do passo é variável e pode dar resultados precisos quando as não-linearidades da fibra são a principal fonte de distorção. Portanto, para um passo de tamanho dz , o efeito do operador não linear $\hat{N}(z, t)$ é incrementar a fase de $A(z, t)$ por uma quantidade $d\phi_{NL} = dz\gamma P_0$,

onde γ é o parâmetro não-linear e P_0 representa a potência de pico na entrada do segmento. Um limite superior ao comprimento do passo pode ser obtido através da limitação do incremento de fase não-linear a um valor máximo $\phi_{\text{NL}}^{\text{Max}}$. Portanto, dz deve ser escolhido tal que

$$dz \leq \frac{\phi_{\text{NL}}^{\text{Max}}}{\gamma P_0}. \quad (6)$$

Um valor típico para $\phi_{\text{NL}}^{\text{Max}}$ é 0.01 rad. Usando (6), o número de seções de comprimento dz na Figura 1 é dado por $N_s = \left\lceil \frac{L \cdot \gamma \cdot P_0}{\phi_{\text{NL}}^{\text{Max}}} \right\rceil$, onde L é o comprimento da fibra e a função $f(x) = \lceil x \rceil$ aproxima “x” para o número inteiro mais próximo que seja maior ou igual a “x” (função `ceil(x)` no MATLAB). Portanto, o comprimento do passo pode ser escrito como

$$dz = L \cdot \left\lceil \frac{L \cdot \gamma \cdot P_0}{\phi_{\text{NL}}^{\text{Max}}} \right\rceil^{-1} \quad (7)$$

Referências

- [1] G. P. Agrawal, *Nonlinear Fiber Optics*, 4th ed. San Diego, USA: Academic Press, 2006.
- [2] O. V. Sinkin, R. Holzlohner, J. Zweck, and C. R. Menyuk, “Optimization of the split-step Fourier method in modeling optical-fiber communications systems,” *J. Lightw. Technol.*, vol. 21, no. 1, pp. 61–68, Jan 2003.

Problema 1 (20 pt)

Verifique por substituição direta que $A(z, t) = \sqrt{P_0} \operatorname{sech}\left(\frac{t}{T_q}\right) e^{-j\frac{z}{2L_D}}$ satisfaz (1), se $\alpha = 0$ e $\beta_3 = 0$. Assuma que $P_0 \cdot L_D \cdot \gamma = 1$ (soliton fundamental), onde P_0 é a potência do pulso, $L_D = \frac{T_q^2}{|\beta_2|}$ é o comprimento dispersão, $\beta_2 < 0$ (dispersão anômala) e T_q é uma medida da largura de pulso. O uso da seguinte relação e derivadas depende da sua derivação. Você também pode usar outras relações e derivadas.

$$\operatorname{sech}^2(x) + \tanh^2(x) = 1$$

$$\frac{d(\operatorname{sech}(u))}{dx} = -\operatorname{sech}(u) \cdot \tanh(u) \cdot \frac{du}{dx}$$

$$\frac{d(\tanh(u))}{dx} = \operatorname{sech}^2(u) \cdot \frac{du}{dx}$$

Problema 2 (30 pt)

1. Resolva a equação não-linear de Schrödinger (1) usando o *Split-Step Fourier Method*. Você deve implementar uma função no MATLAB chamada *splitstep* da seguinte forma:

```
function Aout=splitstep(Ain,L,stepsize,attenuation,beta2,omega,gama),
```

onde “Ain” e “Aout” são as amplitudes normalizadas da entrada e da saída, “L” é o comprimento da fibra, “attenuation” é α , “beta2” é β_2 , “omega” é a frequência angular e “gama” é γ . Aqui assumimos $\beta_3 = 0$ e aproximamos “ $\exp\left(\frac{dz}{2}(\hat{N}(z + dz, t) + \hat{N}(z, t))\right)$ ” por “ $\exp(dz\hat{N}(z, t))$ ”, evitando, portanto, o processo de iteração.

2. A fim de analisar a propagação de solitons em fibras ópticas, use Template 1 e a função “splitstep.m” para esboçar a potência de saída de cada um dos quatro pulsos iniciais (total de oito gráficos). Comente seus resultados. A relação entre a largura do soliton T_q e a taxa de bits é dada por $R_b = 1/T_b = 1/(2q_0 T_q)$, onde T_b é a duração do bit e $2q_0 = T_b/T_q$ é a separação entre solitons vizinhos em unidades normalizadas.

Problema 3 (50 pt)

Geralmente nos sistemas de comunicação óptica de longa distância, o caminho de transmissão consiste de uma cascata de seções idênticas. Cada seção é composta por fibras SSMF e DCF, e EDFA. Com base no esquema de compensação de dispersão, são identificados três tipos de seções, como mostrado nas Figuras 2, 3 e 4. Dentro da seção, três parâmetros podem ser variados a fim de identificar a configuração ótima. Estes parâmetros são as potências de entrada das fibras SSMF e DCF, P_{SSMF} e P_{DCF} , e a relação de compensação (CR – *compensation ratio*). O ganho dos amplificadores para todas as configurações variam de acordo com P_{SSMF} e P_{DCF} .

1. Calcule os ganhos do amplificador G_1 e G_2 em função das potências de entrada P_{SSMF} and P_{DCF} , atenuação e comprimento das fibras SSMF e DCF.

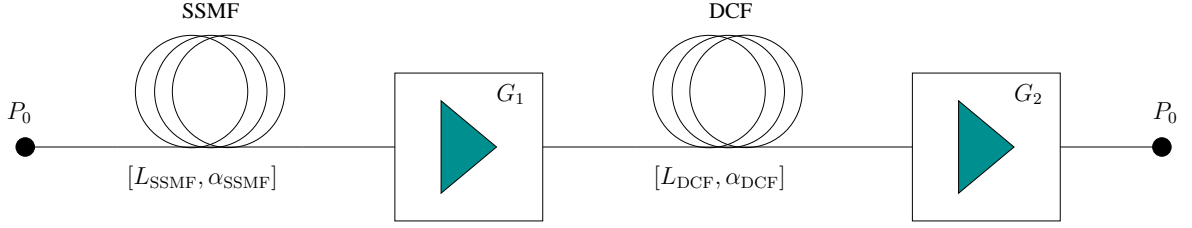


Figura 2: Post-compensação

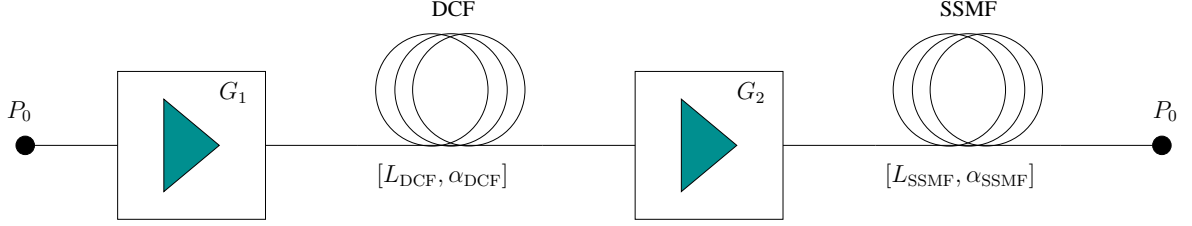


Figura 3: Pre-compensação

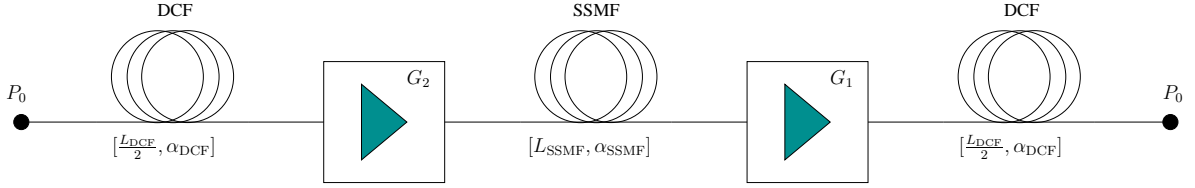


Figura 4: Compensação híbrida

2. Calcule para cada esquema de compensação da dispersão a potência de ruído acumulada após N_s seções em função de P_{SSMF} , P_{DCF} , P_{n_1} e P_{n_2} . Defina os parâmetros necessários para calcular as potências de ruído P_{n_1} e P_{n_2} geradas pelos amplificadores 1 e 2, respectivamente.
3. Mostre que a OSNR é a mesma para os três esquemas de compensação da dispersão após N_s seções. A OSNR é definida aqui como $OSNR = P_0/P_{ASE}$, onde P_{ASE} é a potência do ruído acumulada para cada esquema de compensação da dispersão. Observe nas figuras 2, 3 e 4 que P_0 é diferente para cada caso.

Uma vez que os três esquemas de compensação da dispersão tem a mesma OSNR no final do caminho de transmissão, eles devem ter o mesmo desempenho. No entanto, como a fibra óptica é um meio não-linear, cada esquema de compensação da dispersão produz sinais de saída diferentes. Portanto, em sistemas OOK, a *Eye Opening Penalty* (EOP) é uma boa medida para avaliar a interação entre dispersão e não-linearidades em fibras ópticas.

4. Considerando em primeiro lugar o transmissor, a sua amplitude de saída normalizada pode ser escrita como

$$A(z=0, t) = \sqrt{2P_p} \cdot \cos\left(\frac{\pi}{2} \cos^2\left(\frac{\pi t}{T_b}\right)\right) \cdot \tilde{a}_0(t),$$

onde P_p é a potência do pulso, $\tilde{a}_0(t) \in \{0, 1\}$ dependendo da sequência de bits de entrada

e T_b é o intervalo do bit. Para o intervalo $[0, T_b]$ e $\tilde{a}_0(t) = 1$, calcular a potência média e de pico do pulso. (Dica: use $P_{avg} = \frac{1}{T_b} \int_0^{T_b} [A(z=0, t)]^2 dt$).

5. A otimização do sistema foi realizado utilizando as seções das Figuras 2, 3 and 4. Portanto, as potências de entrada ótima P_{SSMF} e P_{DCF} foram determinadas para cada esquema de compensação, bem como as larguras de filtros ópticos e elétricos ótimas, como mostra a Tabela 1. Para identificar o melhor esquema de compensação da dispersão você deve implementar uma função que calcula *Eye Opening Penalty* depois de cada seção. Use o Template 2 e a função implementada anteriormente "splitstep.m" para executar esta tarefa. A função deve ser implementada da seguinte forma:

```
function y=eyeopeningpen(U,a0,Nb,samples,f,BoptRX,Be),
```

onde "a0" é a sequência de bits dada, "Nb" é o número de bits na sequência de bits, "samples" é o número de amostras por bit, "f" é o vetor de frequência, "BoptRX" e "Be" são as larguras de banda dos filtro ópticos e elétricos, respectivamente. "U" é uma matriz, onde as linhas são as amplitudes de saída depois de cada seção. A primeira linha deve ser o sinal de entrada utilizado como referência para o cálculo da *Eye Opening Penalty*. Note-se que a EOP é calculada usando a corrente elétrica após detecção e filtragem. Comente e inclua o gráfico produzido pelo Template 2 em sua solução.

6. Usando as funções anteriormente implementadas "splitstep.m" e "eyeopeningpen.m" e Template 2, você deve identificar o melhor esquema de compensação da dispersão para 10 Gbit/s (Design "10e9" para a variável "Rb" no Template 2). Comente os seus resultados.

Tabela 1: Potências de entrada ótimas para cada esquema de compensação e larguras de banda de filtros ópticos e elétricos ótimos. Potências estão em dBm.

Esquema de Compensação	OOK			
	B_e/R_b	B_{opt}/R_b	P_{SSMF}	P_{DCF}
Pre-	0.8	1.35	-1.5	-5.5
Hybrid	0.8	1.35	-1	-6
Post-	0.8	1.35	-0.5	-7

Template 1: Soliton Propagation

```

clear all;
tempo=clock;                                % For simulation time
                                              % calculation

%-----General Parameters-----
Rb=10e9;                                     % Bit rate [bit/s]
Tb=1/Rb;                                    % Bit interval [s]
q0=3.5;                                     % Separation between solitons
Tq=1/(2*q0*Rb);                             % Soliton width [s]
comp=4;                                     % Number of bit slots
samples=64;                                 % Samples per slot
N=comp*samples;                             % Total number of samples
time=Tb/samples*(-N/2:(N/2-1));             % Time vector for all slots
f=1/(comp*Tb)*( [1:N] - N/2 - 1);           % Simulation Bandwidth [Hz]
omega=2*pi*f;                               % Angular frequency
%-----

%-----Fiber Parameters-----
k=0.01;                                     % Stepsize constant
beta2 = -1e-27;                             % Parameter for GVD [s^2/m]
gama = 1.52e-3;                             % Nonlinear parameter[1/(W*m)]
attenuation = (0.0*1e-3)/(10*log10(exp(1))); % Attenuation [dB/m]
Ld=Tq^2/abs(beta2);                         % Dispersion length [m]
L = 50*Ld;                                  % Fiber Length [m]
L_step=Ld;                                  % Step for the 3D-plot [m]
P0=abs(beta2)/(gama*Tq^2);                  % Pulse peak power [W]
stepsize = L/ceil(L*gama*P0/k);             % Stepsize in [m]
index=find(stepsize>L);                     % The stepsize should not
stepsize(index)=L;                          %exceed the fiber length
%-----

%-----Initial Condition-----
% 0 - One soliton pulse
% 1 - One Gaussian pulse
% 2 - Soliton interaction (Two pulses)
% 3 - Soliton interaction (Two pulses with different amplitudes)
initial_pulse=0;
switch initial_pulse
    case 0
        u0=sqrt(P0)*sech(time/Tq);
    case 1
        u0=sqrt(P0)*exp(-time.^2/(2*Tq^2));
    case 2
        u0=sqrt(P0)*( sech((time-Tb/2)/Tq) + sech((time+Tb/2)/Tq));
    case 3
        u0=sqrt(P0)*( sech((time-Tb/2)/Tq) + 1.1*sech((time+Tb/2)/Tq));
end
%-----
MrkSz=16;

```

```

LnWdth=2;
figure(1);
plot(time/1e-12,(abs(u0).^2)/1e-3,'LineWidth',LnWdth,'MarkerSize',MrkSz);
hold on;

% Propagating through the fiber
u=splitstep(u0,L,stepsize,attenuation,beta2,omega,gama);

figure(1)
plot(time/1e-12,(abs(u).^2)/1e-3,'rx','LineWidth',LnWdth,'MarkerSize',MrkSz);
set(gca,'linewidth',3,'fontsize',26);
xlabel('Time [ps]','FontSize',30);
ylabel('Power [mW]','FontSize',30);

% 3D-plot
u=u0;
U=u0;
J=L/L_step;
for j=1:J
    u=splitstep(u,L_step,stepsize,attenuation,beta2,omega,gama);
    U=[U;u];
end

figure(2);
h=waterfall(time/1e-12,(0:size(U,1)-1)*L_step/1e3,abs(U).^2/1e-3);
set(gca,'linewidth',3,'fontsize',26);
set(h,'LineWidth',LnWdth);
xlabel('Time [ps]','FontSize',30);
ylabel('Distance [Km]','FontSize',30);
zlabel('Power [mW]','FontSize',30);
axis([time(1)/1e-12 time(end)/1e-12 0 J*L_step/1e3 ...
    0 max(max((abs(U).^2)./1e-3)) 15 16]);
set(gca,'YDir','reverse');

etime(clock,tempo)

```


Template 2: System Optimization

```

clear all;
tempo=clock;                                     % For simulation time
                                                % calculation

%-----General Parameters-----
Rb=40e9;                                         % Bit-Rate
Tb=1/Rb;                                       % Bit interval
samples=32;                                   % Samples per bit
Nb=64;                                         % Number of bits
t=Tb/samples*(0:samples-1);
time=Tb/samples*(0:Nb*samples-1);
f=1/(Nb*Tb)*([1:(samples*Nb)]-samples*Nb/2-1); % Simulation Bandwidth
omega=2*pi*f;
max_sections=20;                             % Number of sections
BoptRX=1.35*Rb;                              % Optical filter RX [Hz]
Be=0.8*Rb;                                   % Electrical filter [Hz]
%-----

%-----Fiber Parameters-----
k=0.01;                                       % Stepsize constant
CR=1;                                        % Compesation ratio

% Standard Single Mode Faser (SSMF)
beta2 = -21.17e-27;                         % Parameter for GVD [s^2/m]
gama = 1.52e-3;                             % Nonlinear parameter[1/(W*m)]
attenuation = (0.2*1e-3)/(10*log10(exp(1))); % Attenuation [dB/m]
L = 80e3;                                   % Fiber Length [m]

% Dispersion Compensating Fiber (DCF)
beta2_dcf = 131.88e-27;                    % GVD Parameter [s^2/m]
gama_dcf = 5.27e-3;                       % Nonlinear parameter[1/(W*m)]
attenuation_dcf = (0.50*1e-3)/(10*log10(exp(1))); % Attenuation [dB/m]
L_dcf = CR * L * abs(beta2) / abs(beta2_dcf); % fiber length [m]
%-----

%-----Transmitter-----
% Pulse 50% RZ
Pp=0.001;                                     % Pulse power [W]
p=sqrt(2*Pp)*cos(cos(pi*t/(Tb)).^2*pi/2);

% Precoding
a0=[0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 0 0 0 1 1 ...
    1 0 0 1 0 0 1 0 1 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 1 1 1 1 1];

% Sequence of pulses
u0=zeros(1,Nb*samples);
for j=1:Nb
    u0((j-1)*samples+1:j*samples)=p*a0(j);
end

```

```

P0=u0*u0'/length(u0);
%-----

%-----General System Parameters-----

%-----Post-Compensation-----
Pssmf_pos=10.^(-0.05)*1e-3; % Optimum power for 40Gbit/s
stepsize_pos = L/ceil(L*gama*2*Pssmf_pos/k); % Stepsize in [m]
index=find(stepsize_pos>L); % The stepsize should not
stepsize_pos(index)=L; %exceed the fiber length

Pdcf_pos=10.^(-0.7)*1e-3; % Optimum power for 40Gbit/s
stepsize_dcf_pos = L_dcf/ ...
    ceil(L_dcf*gama_dcf*2*Pdcf_pos/k); % Stepsize in [m]
index=find(stepsize_dcf_pos>L_dcf); % The stepsize should not
stepsize_dcf_pos(index)=L_dcf; %exceed the fiber length
u_pos=u0*sqrt(Pssmf_pos/P0); % Input power
%-----

%-----Pre-Compensation-----
Pssmf_pre=10.^(-0.15)*1e-3; % Optimum power for 40Gbit/s
stepsize_pre = L/ceil(L*gama*2*Pssmf_pre/k); % Stepsize in [m]
index=find(stepsize_pre>L); % The stepsize should not
stepsize_pre(index)=L; %exceed the fiber length

Pdcf_pre=10.^(-0.55)*1e-3; % Optimum power for 40Gbit/s
stepsize_dcf_pre = L_dcf/ ...
    ceil(L_dcf*gama_dcf*2*Pdcf_pre/k); % Stepsize in [m]
index=find(stepsize_dcf_pre>L_dcf); % The stepsize should not
stepsize_dcf_pre(index)=L_dcf; %exceed the fiber length
u_pre=u0*sqrt(Pssmf_pre/(exp(attenuation*L)*P0)); % Input power
%-----

%-----Hybrid-Compensation-----
Pssmf_hyb=10.^(-0.1)*1e-3; % Optimum power for 40Gbit/s
stepsize_hyb = L/ceil(L*gama*2*Pssmf_hyb/k); % Stepsize in [m]
index=find(stepsize_hyb>L); % The stepsize should not
stepsize_hyb(index)=L; %exceed the fiber length

Pdcf_hyb=10.^(-0.6)*1e-3; % Optimum power for 40Gbit/s
stepsize_dcf_hyb = L_dcf/ ...
    ceil(L_dcf*gama_dcf*2*Pdcf_hyb/k); % Stepsize in [m]
index=find(stepsize_dcf_hyb>L_dcf); % The stepsize should not
stepsize_dcf_hyb(index)=L_dcf; %exceed the fiber length
u_hyb=u0*sqrt(Pdcf_hyb/(exp(attenuation_dcf*L_dcf/2)*P0)); % Input power
%-----

U_pos=[u_pos];
U_pre=[u_pre];
U_hyb=[u_hyb];

```

```

% Post-Compensation-----
j=1;
while j<=max_sections
    disp(sprintf('Post-Comp. Span: %g Pdcf: %g dBm Pssmf: %g dBm',...
        j,10*log10(Pdcf_pos/1e-3),(10*log10(Pssmf_pos/1e-3))));

    % Fiber SSMF
    u_pos=splitstep(u_pos,L,stepsize_pos,attenuation,beta2,omega,gama);

    % Optical Amplifier
    Gain=Pdcf_pos*exp(attenuation*L)/Pssmf_pos;
    u_pos=u_pos*sqrt(Gain);

    % Fiber DCF
    u_pos=splitstep(u_pos,L_dcf,stepsize_dcf_pos,attenuation_dcf, ...
        beta2_dcf,omega,gama_dcf);

    % Optical Amplifier
    Gain=Pssmf_pos*exp(attenuation_dcf*L_dcf)/Pdcf_pos;
    u_pos=u_pos*sqrt(Gain);

    U_pos=[U_pos;u_pos];
    j=j+1;
end
%-----

% Pre-Compensation-----
j=1;
while j<=max_sections

    disp(sprintf('Pre-Comp. Span: %g Pdcf: %g dBm Pssmf: %g dBm', ...
        j,10*log10(Pdcf_pre/1e-3),(10*log10(Pssmf_pre/1e-3))));

    % Optical Amplifier
    Gain=Pdcf_pre*exp(attenuation*L)/Pssmf_pre;
    u_pre=u_pre*sqrt(Gain);

    % Fiber DCF
    u_pre=splitstep(u_pre,L_dcf,stepsize_dcf_pre,attenuation_dcf, ...
        beta2_dcf,omega,gama_dcf);

    % Optical Amplifier
    Gain=Pssmf_pre*exp(attenuation_dcf*L_dcf)/Pdcf_pre;
    u_pre=u_pre*sqrt(Gain);

    % Fiber SSMF
    u_pre=splitstep(u_pre,L,stepsize_pre,attenuation,beta2,omega,gama);

    U_pre=[U_pre;u_pre];
    j=j+1;
end

```

```

%-----

% Hybrid-Compensation-----
j=1;
while j<=max_sections

    disp(sprintf('Hybrid-C. Span: %g Pdcf: %g dBm Pssmf: %g dBm', ...
        j,10*log10(Pdcf_hyb/1e-3),(10*log10(Pssmf_hyb/1e-3))));

    % Fiber DCF/2
    u_hyb=splitstep(u_hyb,L_dcf/2,stepsize_dcf_hyb/2, ...
        attenuation_dcf,beta2_dcf,omega,gama_dcf);

    % Optical Amplifier
    Gain=Pssmf_hyb*exp(attenuation_dcf*L_dcf)/Pdcf_hyb;
    u_hyb=u_hyb*sqrt(Gain);

    % Fiber SSMF
    u_hyb=splitstep(u_hyb,L,stepsize_hyb,attenuation,beta2,omega,gama);

    % Optical Amplifier
    Gain=Pdcf_hyb*exp(attenuation*L)/Pssmf_hyb;
    u_hyb=u_hyb*sqrt(Gain);

    % Fiber DCF/2
    u_hyb=splitstep(u_hyb,L_dcf/2,stepsize_dcf_hyb/2, ...
        attenuation_dcf,beta2_dcf,omega,gama_dcf);

    U_hyb=[U_hyb;u_hyb];
    j=j+1;
end
%-----
MrkSz=16;
LnWdth=2;
plot(0:max_sections,eyeopeningpen(U_hyb,a0,Nb,samples,f,BoptRX,Be),...
    '-o','LineWidth',LnWdth,'MarkerSize',MrkSz)
hold on
plot(0:max_sections,eyeopeningpen(U_pos,a0,Nb,samples,f,BoptRX,Be),...
    ':sr','LineWidth',LnWdth,'MarkerSize',MrkSz)
plot(0:max_sections,eyeopeningpen(U_pre,a0,Nb,samples,f,BoptRX,Be),...
    '--*k','LineWidth',LnWdth,'MarkerSize',MrkSz)
legend('Hybrid Compensation','Post-Compensation','Pre-Compensation',2);
set(gca,'linewidth',3,'fontsize',26);
xlabel('Sections','FontSize',30);
set(gca,'XTick',0:2:max_sections)
ylabel('Eye Opening Penalty [dB]','FontSize',30);
title(sprintf('RZ-OOK %gGbit/s',Rb/1e9),'FontSize',30);
grid on;

etime(clock,tempo)

```