

## 1 Introduction

In this simulation project, we want to investigate one of the mathematical models used to analyze the optical fiber non-linear effect through approximately solving the NLSE.

As we know, NLSE is not solvable in its most general form (i.e. when all fiber impairments are active and effective), hence approximate mathematical models come to play. All of these models accept span and link parameters (such as  $\alpha$ ,  $\beta_2$ ,  $\beta_3$ ,  $\gamma$ , span amplifier noise figure, a set of occupied WDM channels and transmitted power which will be clarified, and more parameters which will be discussed when they matter) and obtain a real value representing non-linear noise PSD <sup>1</sup> in one of the WDM channels of our interest, to which we sometimes refer as COI <sup>2</sup> or CUT <sup>3</sup>.

Based on previous discussions about WDM (which is essentially the same FDM) in the OptNet course, we know that the total spectrum of an optical fiber (over which the mid-band signals are transmitted) can be divided to distinct channels, each of which possessing a portion of the spectrum, which is referred to as **channel bandwidth** and a center frequency which is used to convert a baseband signal to an equivalent midband signal to make it ready for transmission over the fiber.

WDM grid may be elastic, in which the channel bandwidths are not or loosely restricted to any rule; they can take on every value. This kind of WDM grid is also known as *flexible grid*. In contrast, fixed WDM grid or just simply fixed grid are terms used to refer that all channel bandwidths must be equal to some fixed constant. It is now clear that WDM grid can support multiple concurrent transmission in time, with each WDM channel is allocated to a single user, while preserving degrees of freedom in choosing launch power (the power consumed by a user for transmission over own channel) and modulation format (such as PSK or QAM for converting bit-stream to symbol-stream).

## 2 Roadmap

With possibly multiple users having active concurrent transmission on their own channels through an optical fiber, non-linear impairment may occur due to *SPM*, *XPM* and *FWM* which we care about when they occur in COI. SPM is the non-linear effect (crosstalk) of COI on itself, XPM is the non-linear effect of COI and two other channels on COI, and finally, FWM is the non-linear crosstalk of three different channels (other than COI) which happens because of the interaction between them. All of the non-linear effects become

---

<sup>1</sup> Power Spectral Density

<sup>2</sup> Channel Of Interest

<sup>3</sup> Channel Under Test

more significant as the injected power into the fiber increases (because of the non-linear 3rd-order term,  $\gamma P.E$ , in the Manakov's equation).

The non-linear effects described above translate to noise commonly referred to as *NLI*<sup>4</sup> *noise power*, which, alongside the ASE noise of lumped elements and amplifiers, is added to main transmitted signal and degrade *Signal to Noise and Interference Ratio* accordingly. This simulation project dedicates to account for both the NLI+ASE and approximately calculate the NLI+ASE power spectral density and consequently, the total noise power.

### 3 NLI Power Spectral Density

In this section, we technically focus on all the parameters we talked about. The *NLI Power Spectral Density in COI* is calculated through  $G_{\text{NLI}}^{\text{Rx}}(f_{\text{CUT}})$  which is given by:

$$G_{\text{NLI}}^{\text{Rx}}(f_{\text{CUT}}) = \sum_{n=1}^{N_{\text{span}}} \left( G_{\text{NLI}}^{(n)}(f_{\text{CUT}}) \prod_{k=n+1}^{N_{\text{span}}} \Gamma^{(k)}(f_{\text{CUT}}) \cdot e^{-2 \cdot \alpha^{(k)}(f_{\text{CUT}}) \cdot L_{\text{span}}^{(k)}} \right) \quad (1)$$

$$G_{\text{NLI}}^{(n)}(f_{\text{CUT}}) = \frac{16}{27} (\gamma^{(n)})^2 \Gamma^{(n)}(f_{\text{CUT}}) \cdot e^{-2 \alpha^{(n)}(f_{\text{CUT}}) \cdot L_{\text{span}}^{(n)}} \cdot \bar{G}_{\text{CUT}}^{(n)} \cdot \left( \rho_{\text{CUT}}^{(n)} \cdot [\bar{G}_{\text{CUT}}^{(n)}]^2 I_{\text{CUT}}^{(n)} + \sum_{n_{\text{ch}}=1, n_{\text{ch}} \neq n_{\text{CUT}}^{(n)}}^{N_{\text{ch}}} 2 \rho_{n_{\text{ch}}}^{(n)} \cdot [\bar{G}_{n_{\text{ch}}}^{(n)}]^2 I_{n_{\text{ch}}}^{(n)} \right) \quad (2)$$

$$I_{\text{CUT}}^{(n)} = \frac{1}{2\pi |\bar{\beta}_{2,\text{CUT}}^{(n)}| \cdot 2\alpha^{(n)}(f_{\text{CUT}})} \cdot \text{asinh} \left( \frac{\pi^2}{2} \left| \frac{\bar{\beta}_{2,\text{CUT}}^{(n)}}{2\alpha^{(n)}(f_{\text{CUT}})} \right| R_{\text{CUT}}^2 \right) \quad (3)$$

$$I_{n_{\text{ch}}}^{(n)} = \frac{\text{asinh} \left( \pi^2 \left| \frac{\bar{\beta}_{2,n_{\text{ch}}}^{(n)}}{2\alpha^{(n)}(f_{n_{\text{ch}}})} \right| \left[ f_{n_{\text{ch}}}^{(n)} - f_{\text{CUT}} + \frac{R_{n_{\text{ch}}}^{(n)}}{2} \right] R_{\text{CUT}} \right) - \text{asinh} \left( \pi^2 \left| \frac{\bar{\beta}_{2,n_{\text{ch}}}^{(n)}}{2\alpha^{(n)}(f_{n_{\text{ch}}})} \right| \left[ f_{n_{\text{ch}}}^{(n)} - f_{\text{CUT}} - \frac{R_{n_{\text{ch}}}^{(n)}}{2} \right] R_{\text{CUT}} \right)}{4\pi |\bar{\beta}_{2,n_{\text{ch}}}^{(n)}| \cdot 2\alpha^{(n)}(f_{n_{\text{ch}}})} \quad (4)$$

$$\bar{\beta}_{2,\text{CUT}}^{(n)} = \beta_2^{(n)} + \pi \beta_3^{(n)} [2f_{\text{CUT}} - 2f_c^{(n)}] \quad , \quad \bar{\beta}_{2,n_{\text{ch}}}^{(n)} = \beta_2^{(n)} + \pi \beta_3^{(n)} [f_{n_{\text{ch}}}^{(n)} + f_{\text{CUT}} - 2f_c^{(n)}] \quad (5)$$

The latter equations are referred to as CFM equations which may seem more forbidding than what they should, but you will not need to delve into all the details.

The Python code provided with the attachment, actually simulates equation 1, based on equations 2,3,4 and 5, and ASE noise power spectral density. The function headers and parameters are defined below. The term **active channel** is used to represent a channel over which a single user is transmitting.

### 4 Where to start coding

In the Python script given to you, the CFM code is provided however only the transmitter and receiver code blocks for SSFM are given. You are supposed to write the NLSE solver on your own by taking help from an attached .m file that simulates the NLSE solver in MATLAB (actually, you must complete the code body of the function `NLSE_Solver_Link`

---

<sup>4</sup> Non-Linear and Interference

within the lines 1073 to 1081 and use it in the line 1175 of the main code block). Also you must pass suitable parameters (link parameters provided in the code and the input signal). After completing the code of SSFM, you can start simulating both CFM and SSFM to obtain SNR values by passing the input signal to `NLSE_Solver_Link` and returning the impaired link output signal. The modulation format is QPSK.

## 5 Function Headers and Parameters

##### ASE\_PSD Function Header #####

ASE\_PSD ( NumSpan , AmplifierGain , AmplifierNF )

##### Parameters #####

1. NumSpan (int) : Number of spans in link
2. AmplifierGain (float) : Amplifier gain calculated through  $e^{\alpha L}$
3. AmplifierNF (float) : Amplifier noise figure in dB scale

(Single Channel Transmission)

Assume only a single user transmits data over a channel with center frequency of 32.5G and a channel bandwidth of 32.5G. The transmission takes place within a link consisting  $N_{\text{Span}}$  spans, each of which containing an optical fiber followed by an EDFA. The optical fibers and EDFAs have the following typical parameters:

1.  $\alpha_{\text{dB}} = 0.22 \text{ dB/km}$
2.  $D = 17 \text{ ps/(nm.km)}$
3.  $\gamma = 1.3W^{-1} \cdot \text{km}^{-1}$
4.  $L_{\text{Span}} = 140 \text{ km}$
5.  $\text{NF}_{\text{EDFA}} = 5.5 \text{ dB}$

In the following questions, you are assigned to run both SSFM and CFM functions using proper inputs to obtain SNR values.

### Question 1)

Assume  $\gamma = 0$  for all optical fibers in spans (i.e. the optical fiber has no non-linear impairment). Each span consists an optical fiber followed by an EDFA with  $\text{NF} = 5.5\text{dB}$ . The other parameters have their typical values.

a- Find  $\alpha$  and  $\beta_2$  from  $\alpha_{\text{dB}}$  and  $D$  by using the following formulas:

$$\alpha_{\text{dB}} = 4.343e3 \times \alpha$$

$$D = -\frac{2\pi c}{\lambda^2} \beta_2$$

- b- Calculate the ASE noise variance from the following formula:

$$\sigma_{\text{ASE}}^2 = h\nu \cdot N_{\text{Span}} \cdot \frac{(F \cdot G - 1)}{2} \cdot \text{BW}$$

for four values of  $N_{\text{Span}} = 1, 7, 15, 25$ . The parameters are  $h$  as Planck's constant,  $\nu = 196.07840\text{THz}$ ,  $F$  as a single EDFA's noise factor (the linear scale of noise figure) and  $G$  as the EDFA power amplification gain, which is equal to  $e^{\alpha L}$  with  $\alpha$  and  $L$  being the optical fiber attenuation and total length, respectively. Also, BW is the receiver's channel bandwidth which is equal to 33.15G.

- c- Run the function `ASE_PSD` passing proper parameters for  $N_{\text{Span}}$  being the same as the previous part. Refer to the **ASE\_PSD Function Header** and **Parameters** for hints. Do not forget to multiply the output of the function in  $\text{BW}$ . Compare the result with that of part a.
- d- Sketch a plot for SNR in dB scale with launch power varying from -5dBm to 10dBm with steps 0.5dBm and  $N_{\text{Span}} = 1, 5, 10, 25$  for both CFM and SSFM. The CFM SNR is calculated automatically given proper parameters, but you should embed `NLSE_Solver_Link` function in the main code as described before and in the main code script. Use `numpy.arange` function for this. Also for plotting, you can use `matplotlib.pyplot.plot` which is the same `plot` function in MATLAB; you can pass two vectors `x` and `y` to it, e.g. the following code block

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x=np.arange(-1,1,0.001)
plt.plot(x,x**2)
```

plots the  $x^2$  curve in  $[-1, 1]$  with step=0.001.

(Remark: for plotting multiple figures, you can add new figures using

```
plt.figure()
```

```
)
```

## Question 2)

Now let  $\gamma = 1.3 \times 10^{-3}$  and all the parameters have their typical values.

- a- Sketch the SNR plot in dB scale with launch power varying from -5dBm to 10dBm with steps 0.5dBm and  $N_{\text{Span}} = 1, 5, 10, 25$  for both SSFM and CFM. Compare the plot with that you obtained in part d of question 1 and mention the major differences with reasons.
- b- Sketch scatter plots of the transmitted and received modulation symbols for  $N_{\text{Span}} = 25$  through the SSFM simulation for launch power = -5, 0, 5, 10 dBm. How does the scatter plot change versus launch power?

(Hint: to obtain the scatter plot of a quantity, namely `var`, use the command

```
plt.plot(numpy.real(var),numpy.imag(var),'.')
```

```
)
```

### Question 3)

As accumulated dispersion appears in the received signal, an EDC block is considered to remove it totally, yielding a dispersion-free signal. The received symbols can then be obtained by regularly resampling the signal and the QPSK constellation can be observed in the receiver. By removing EDC, we expect the received constellation to be messy with undistinguishable symbols. To test this theory, obtain scatter plots removing the EDC block from detection process (by passing  $\beta_2=0$  to its function) for  $N_{\text{span}} = 25$  and launch power = 5 dBm. Compare the received constellation in this case with that obtained from part b of question 2 and mention the major differences.