Assignment 3

Rev 22/10/2024

D. Kong, F. Da Ros, A. T. Clausen

DTU Electro

Department of Electrical and Photonics Engineering, Technical University of Denmark

Learning objectives:

With this assignment, you should be able to do the following tasks in Matlab/Python.

1. Generate additive Gaussian white noise (AWGN) to an amplitude modulated optical signal according to a designated optical-to-noise ratio (OSNR).
2. Construct a Gaussian-shaped optical filter and apply it to the noisy optical signal.
3. Investigate the impact of optical noise and filter on the performance of the optical signal in terms of bit error rate (BER).
4. Generate a wavelength-division multiplexed (WDM) signal.
5. Design an optical receiver for a noisy WDM signal and investigate the performance.

<u>For the hand-in procedure, please refer to Appendix III.</u>

Exercise 1 – Basics of optical filter

The exercise follows the material discussed during the lectures on Multichannel systems.

<u>Question 1-1: Adding noise to the optical signal.</u>

In this question, you will generate an on-off keying (OOK) modulated optical signal with non-return-to-zero (NRZ) pulse shaping, add AWGN to the optical signal, and investigate the performance of the noisy signal using a direct-detection optical receiver.

This question is an extension of Exercises in Point-to-Point Simulations (FDRO).

Before we implement the noise to our optical signal, we need to know the definition of optical signal-to-noise ratio (OSNR). The concept of OSNR will be introduced in detail in chapter 7 – Loss Management (4). But we will give a brief introduction here. The OSNR is defined as

$$\mathrm{ONSR} = \frac{\text{Power of the optial signal}}{\text{Power of the noise in the reference bandwidth}} = \frac{P_s}{2N_{\mathrm{ASE}}B_{\mathrm{ref}}}$$

The definition is historic. P_s is the <u>total average signal power summed over the *two states of polarization*</u>; N_{ASE} is the <u>power spectral density of the amplified spontaneous emission (ASE) noise in *one polarization*</u>; B_{ref} is the reference noise bandwidth. <u>The factor 2 in the definition represents both polarizations of the ASE noise</u>. Here, we only need to know that the noise power is measured in the reference bandwidth, which is defined to be 0.1 nm. Around 1550 nm, this reference bandwidth is ~12.5 GHz. For a more accurate bandwidth in Hz,

$$B_{\text{ref\_Hz}} = \frac{c}{\lambda_0 - B_{\text{ref}} * 0.5} - \frac{c}{\lambda_0 + B_{\text{ref}} * 0.5}$$

where $B_{\text{ref\_Hz}}$ is the noise reference bandwidth in Hz, and B_{ref} is the noise reference bandwidth in meter; c is the speed of light in vacuum; and λ_0 is the central wavelength of the signal (in meters). The following figure shows the definitions of noise bandwidth for SNR and OSNR, respectively.

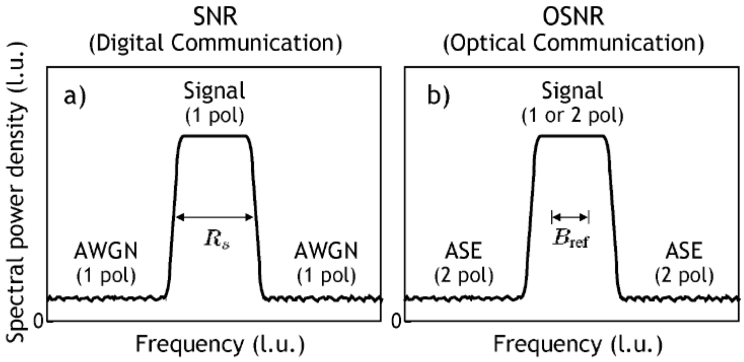


Fig. Ref: R. -J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini and B. Goebel, "Capacity Limits of Optical Fiber Networks," in Journal of Lightwave Technology, vol. 28, no. 4, pp. 662-701, Feb.15, 2010, doi: 10.1109/JLT.2009.2039464.

Here we will explore the source of optical noise but to emulate it using an additive white Gaussian noise (AWGN). AWGN is a basic noise model to emulate the effect of many nature's random processes. Recall curriculum "Point-to-point" about the definition of AWGN:

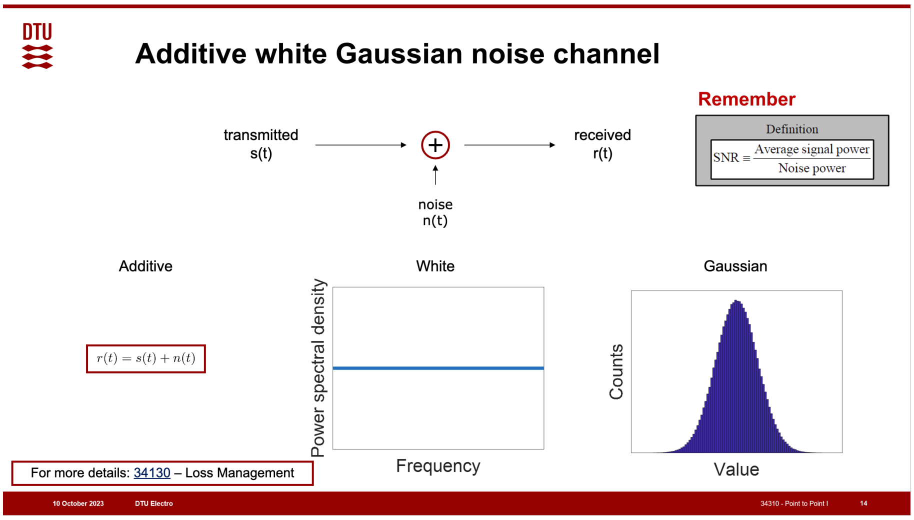


Fig. Ref: AWGN channel in curriculum "point-to-point"

"Additive" means the noise is introduced to the signal via addition. "White" means the spectral density of the noise is constant for all frequencies, i.e., we treat N_{ASE} as a constant in the frequency domain. "Gaussian" means the noise complies with a normal distribution in time domain and the strength of the noise can be solely determined by the variance of the noise.

Therefore, we only need to calculate the power of the optical noise P_N, and use it as the variance of a zero-mean Gaussian distribution to generate the time-domain noise sequence.

The power of the optical noise is integrated using the power spectral density N_{ASE} over the bandwidth of interest, i.e., the observation bandwidth, which is equal to the sampling rate, F_s

$$P_N = N_{\text{ASE}} \cdot F_s$$

And we know from the definition of OSNR

$$N_{\text{ASE}} = \frac{P_s}{2B_{\text{ref}} \cdot \text{ONSR}}$$

We can calculate the power of the optical signal by summing the absolute squares of its time-domain samples divided by the signal length N (number of samples), or equivalently, the square of its root-mean-square (RMS) level.

$$P_s = \frac{\sum_{n=1}^{N} \left|E_{\text{sig},n}\right|^2}{N}$$

Using these equations, we have

$$P_N = \frac{\sum_{n=1}^{N} \left|E_{\text{sig},n}\right|^2}{2B_{\text{ref}} \cdot \text{ONSR} \cdot N} \cdot F_s$$

There's only one missing piece before we can conduct the simulation. If we remember from curriculum "Chapter 3 Optical Transmitters (3) - Laser Characteristics final" that the noise can be complex (see the figure below), we know that the noise power or the variance σ^2 must be devided by 2 and be applied to a complex sequence (electrical field)

$$C_n = \sqrt{P_N/2} * \left(C_{n,r} + 1i \cdot C_{n,i}\right)$$

$C_{n,r}$ and $C_{n,i}$ are real sequences with a Gaussian distribution N(0, 1), and C_n is the complex noise sequence (electrical field) to be added to the optical signal (electrical field).

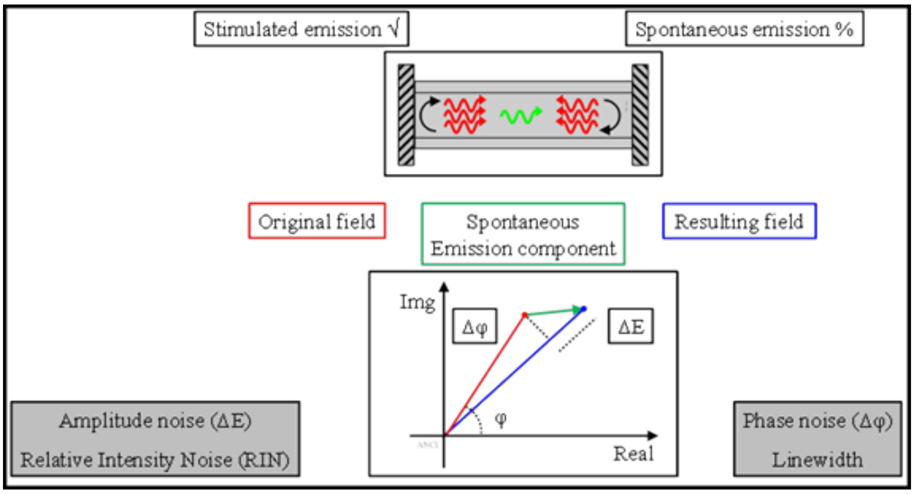


Fig. Ref: Complex noise in Chapter 3 Optical Transmitters (3) - Laser Characteristics final

In summary, for the implementation of AWGN, we will define an OSNR value, generate the electrical field of the noise in time domain according to the previous description, and finally add the noise to the optical signal.

Tips:

1. For a recap of time and frequency grids, as well as Fourier transform, refer to Appendix II.
2. Remember the OSNR and SNR in the above equation are both in linear scale.
3. Remember the time-domain sequences for the noise and the optical signal must be represented in electrical fields.

In this question

1. Set up a time-frequency simulation grid using these parameters:
   a. Number of samples per symbol, $N_{ss} = 64$
   b. Signal baud rate, $R_s = 10e9$ baud, i.e., 10 Gbaud
   c. Number of symbols, $N_{\text{sym}} = 1e3$, i.e, 1000 symbols
2. Initialize the random number generator in Matlab with a seed equal to 34310: rng(34310) and generate an OOK signal with NRZ pulse shaping using a Mach-Zehnder modulator.
3. Plot and observe the power spectrum density (PSD) of the OOK signal, hold on the plot.
4. Add optical noise to the optical signal so that the OSNR is 20 dB.
5. Plot and observe the PSD of the noisy OOK signal. Mark roughly the noise level and the signal level. Is this what you expected?

Hint
Ignore the large residual carrier of the optical signal when marking the signal level.

6. (Optional) Change the Number of samples per symbol, N_{ss}, to 32. Repeat step 2 – 5, observe the PSD plots, and answer: Is the out-of-band noise level dependent on the N_{ss}? Does the OSNR mean in-band signal-to-noise ratio or simply P_s/P_N like the definition of SNR?

Question 1-2: Apply optical filtering to the noisy signal

Next, we are going to implement a Gaussian-shaped optical filter to the noisy signal and observe the noise suppression effect by the filter. In this assignment, we will implement optical filters in the frequency domain.

The Gaussian shape for an optical filter is usually described in the power domain, not the electrical field. And the power transfer function of a Gaussian-shaped filter is

$$H_p(f) = \exp\left[-\frac{1}{2}\left(\frac{f-f_0}{F_0}\right)^{2\cdot\text{ord}}\right]$$

where f is the frequency grid; f_0 is the filter center frequency. If the simulation frequency grid centered at 0 (base-band simulations), f_0 is then the frequency offset; ord is the order of the Gaussian filter. F_0 is the 1/e width of the filter. Usually, we refer to the bandwidth of an optical filter as the 3-dB bandwidth, i.e., the full-width half-maximum (FWHM) in linear scale of the power transfer function, $B_{3\text{dB}}$. It is easy to derive that F_0 is related to the 3-dB bandwidth, $B_{3\text{dB}}$, of the filter by

$$F_0 = \frac{B_{3\mathrm{dB}}}{2 \cdot \sqrt[2\cdot\mathrm{ord}]{2\ln 2}}$$

Note again that $H_p(f)$ here denotes the power transfer function of the optical filter. In order to apply the filter to the optical signal represented in electrical field, we will have to write down the field transfer function of the Gaussian-shaped filter

$$H(f) = \sqrt{H_p(f)} = \exp\left[-\frac{1}{4}\left(\frac{f-f_0}{F_0}\right)^{2\cdot\mathrm{ord}}\right]$$

We will also implement the out-of-band rejection ratio for the optical filter. This can be easily done by setting a minimum value for the transfer function. The rejection ratio usually reads in decibel scale, i.e., dB. If the rejection ratio is $R_{\mathrm{P\_dB}} = 20$ dB (power transfer function), the minimum value for the field transfer function is

$$H(f)_{\min} = \frac{1}{10^{\frac{R_{\mathrm{P\_dB}}}{20}}}$$

In this question

1. Construct a first-order Gaussian-shaped optical filter with a 3-dB bandwidth of $4 \cdot R_s$, and an out-of-band rejection ratio of 20 dB.

   Hint

   We can emulate the out-of-band rejection by

```
RejectionRatio_dB = 20;                 % in power transfer function, dB
Hf_min = 1/(10^(RejectionRatio_dB/20)); % Hf_min used for field transfer function, linear.
Hf_field (Hf_field <= Hf_min) = Hf_min;
```

2. Plot the power transfer function of the Gaussian-shaped optical filter in dB scale in the frequency domain using the code template for the plot (replace corresponding variable names to the ones fit your program). Is the result what you expected? Is the 3-dB bandwidth of the filter correct (within limited error margin?)

   Code template

```
% Frequency grid F in Hz, shown in GHz in the figure
% Hf_field is the field transfer function of the optical filter, in linear scale
% Hf_power is the power transfer function of the optical filter, in decibel scale
% Fs is the sampling rate
% filterBtw is the 3-dB bandwidth of the filter in Hz.
% RejectionRatio_dB is the out-of-band rejection ratio, in decibel scale
figure
Hf_power = -20*log10(Hf_field);
plot(F/1e9, Hf_power)
set(gca, 'YDir','reverse')
axis([-Fs/2/1e9 Fs/2/1e9 min(Hf_power)-5 max(Hf_power)+5])
text(-filterBtw/2/1e9, 3, 'FWHM point, left \rightarrow', 'HorizontalAlignment', 'right')
text(filterBtw/2/1e9, 3, '\leftarrow FWHM point, right', 'HorizontalAlignment', 'left')
axis([-Fs/2/1e9 Fs/2/1e9 0 RejectionRatio_dB])
xlabel('Frequency (GHz)')
ylabel('Rejection (dB)')
```

3. Apply the optical filter to the noisy OOK signal.

> ***Hint***
>
> Remember the FFT practice carried out in assignment 1. To apply the filter, you have to
> 1. Use fft() function to convert the optical signal (electrical field) from the time domain into the frequency domain,
> 2. use fftshift() function to align the results to the frequency grid,
> 3. multiply the filter's transfer function (electrical field),
> 4. use ifftshift() function to align the frequency sequence to IFFT algorithm grid, and
> 5. use ifft() function to convert the signal from frequency domain back to the time domain.

4. Detect the noisy signal using a photodetector without filtering and plot:

   > Hint:
   > Remember photodetector based optical receiver in curriculum "optical receiver 3(B), and section 10.7.4 in Agrawal," and the exercises in "point-to-point."

   a. the eyediagrams (plot using 20,000 samples to speed up),
   b. the waveforms (plot only 1/100 section of the time grid), e.g., `axis([Tw/100*67 Tw/100*68 … …])`
   c. the power spectral densities
5. Detect the filtered signal using a photodetector and plot:
   a. the eyediagrams (plot using 20,000 samples to speed up),
   b. the waveforms (plot only 1/100 section of the time grid), e.g., `axis([Tw/100*67 Tw/100*68 … …])`
   c. the power spectral densities
6. Compare results from step 5 and step 6. Explain qualitatively the filtering effect on the optical signal.

<u>Question 1-3: Performance evaluation</u>

In this question, we will investigate quantitatively the filtering effect on the performance of the optical signal in terms of the bit error rate (BER) in a point-to-point transmission system. Recall curriculum "point-to-point" for optical signal generation, detection, downsampling, and BER calculation.

1. Set up a point-to-point optical transmission system using the following params.
   a. Modulation format: OOK; Detection scheme: direct detection using a single photodetector.
   b. Signal baud rate, $R_s = 10e9$ baud, i.e., 10 Gbaud
   c. Number of samples per symbol, $N_{ss} = 16$
   d. Number of symbols, $N_{\text{sym}} = 1e5$ symbols
2. Investigate the signal performance with optical filtering.
   a. Iterate OSNR from 16 dB down to 6 dB, with an interval of 1 dB.
   b. Generate AWGN and apply to the signal according to current OSNR.
   c. Construct a first-order Gaussian-shaped optical filter centered with the optical signal with a 3-dB bandwidth of $2 \cdot R_s$, and an out-of-band rejection ratio of 20 dB.
   d. Apply the optical filter to the signal.
   e. Detect the filtered OOK signal, do symbol decision, and calculate BER.
   f. Repeat b to e until the OSNR values are fully iterated. You can put the OSNR iteration and BER calculations in a "for" loop to speed up.
   g. Plot the OSNR-BER curve. You should use a line plot with markers to show the data points.

IMPORTANT:
The detected signal should be normalized to its power to facilitate the decision, i.e.,
`detectedSignal = detectedSignal / sqrt( sum( detectedSignal.^2 )/L );`
The reason why normalizing the detected signal to its power is beyond the scope of this assignment. But we can explain the need for a better decision recalling curriculum “Optical Receivers (3C) – Receiver Design.”

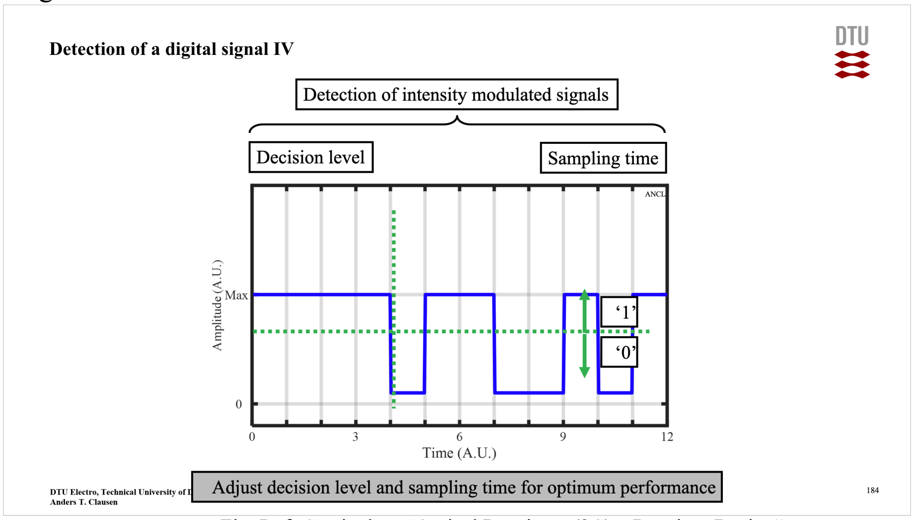


Fig. Ref: Curriculum “Optical Receivers (3C) – Receiver Design”

Answer these questions:

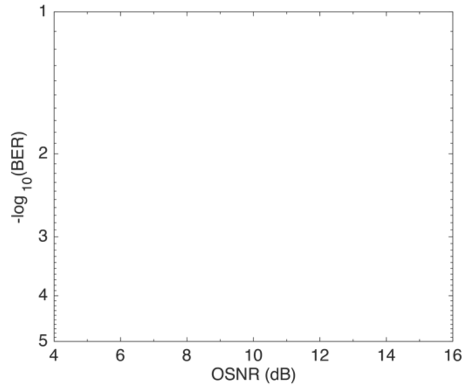
- What are the two factors in the detection process to adjust for optimum performance?
- Look at the eyediagrams of the detected signal before downsampling it for symbol decision at, e.g., OSNR=14 dB. Is the noise distribution on the signal levels (0s and 1s) even? Which level does the noise concentrate more on?
- Why do you think the noise is distributed differently for the two signal levels? Does this have anything to do with the filtering? Explain this in your words.
- The normalization process for the signal to its power is to facilitate _______ ?

Code template:
Refer to the following example code to plot the OSNR-BER curve so the OSNR and BER are nicely shown.

```
% Plot the OSNR-BER curve
plot(OSNR_iter_dB, -log10(BER), '-*')
% set common styles
h = gca;
limX                    = [4, 16];
tickDivX                = 2;
minorTickDivX           = 1;
limY                    = [1, 5];
tickDivY                = 1;
minorTickDivY           = 0.1;
h.YDir                  = 'reverse';
h.FontName              = 'Helvetica';
h.FontSize              = 14;
h.XLabel.String         = 'OSNR (dB)';
% Font automatically adjusted according to global h.FontName and h.FontSize
h.YLabel.String         = '-log_{10}(BER)';
% But the FontSize is 1.1 times larger by default.
h.XLabel.FontSize       = 14;
h.YLabel.FontSize       = 14;
%
% set specific x and y axis styles, such as TickLabelRotation
h.XAxis.Scale           = 'linear';
h.XAxis.Limits          = limX;
```

```
h.XAxis.TickValues      = (limX(1):tickDivX:limX(2));
h.XAxis.MinorTick       = 'on';
h.XAxis.MinorTickValues = limX(1):minorTickDivX:limX(2);
h.XAxis.TickLabelFormat = '%g';
%
h.YAxis.Scale           = 'log';
h.YAxis.Limits          = limY;
h.YAxis.TickValues      = (ceil(limY(1)):tickDivY:limY(2));
h.YAxis.MinorTick       = 'on';
h.YAxis.MinorTickValues = limY(1):minorTickDivY:limY(2);
h.YAxis.TickLabelFormat = '%g';
```

The appearance of your plot should look like the following.



3. Investigate the signal performance under noise without filtering.
   Repeat step 2 but do not use the optical filter.

4. Compare the OSNR-BER curves from step 2 and step 3. Have you observed a performance improvement using optical filtering? Explain your findings. What can you propose to further improve the performance of the signal?
5. Repeat step 2 but change the filter bandwidth to $0.75 \cdot R_s$, plot the OSNR-BER curve.
6. Repeat step 2 but change the filter bandwidth to $0.5 \cdot R_s$, plot the OSNR-BER curve.
7. Comparing the results from step 2, 5, and 6, will the performance keep increasing if the filter bandwidth kept reducing to suppress more noise? Why? Explain this in your words.

Exercise 2 – WDM system

The exercise follows the material discussed during the lectures on Multichannel systems.

<u>Question 2-1: Generation of a WDM signal</u>

In the curriculum “Multichannel system (2)” we have discussed the basic concepts of WDM systems. In contrast to single channel optical transmission systems, WDM system utilizes multiple optical carriers in different wavelengths carried with independent bit streams. This could increase the overall transmission capacity and spectral efficiency of the system (see the figure from the curriculum below). We will deal with the generation of WDM signals in this question.

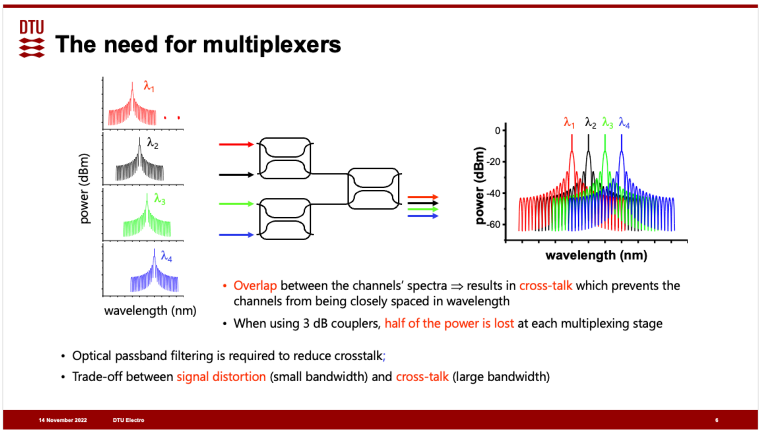


Fig. Ref: curriculum "Multichannel systems"

Remember that we are doing a "equivalent baseband simulation" instead of "optical passband simulation," as introduced in previous exercises:

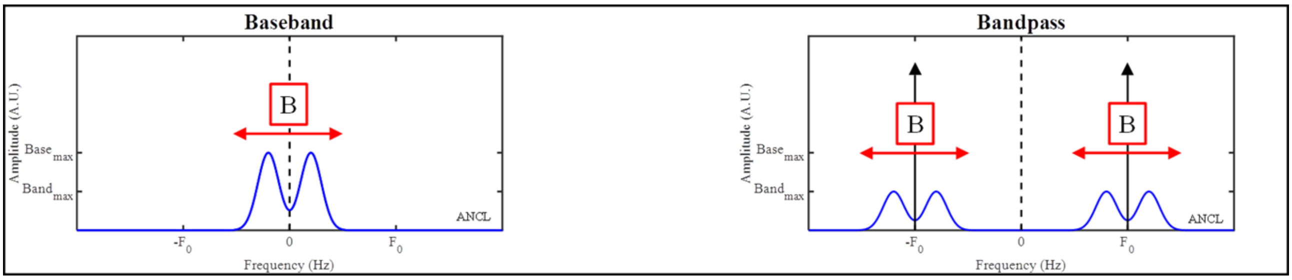


Fig. Baseband system vs Bandpass system. Courtesy: ANCL

Therefore, we will generate the multichannel (WDM) signal at baseband. And the zero frequency will be the center of all signal channels (like the one shown in the above-left figure).

This can be realized by generating multiple optical signal channels and frequency-shifting them according to the grid of the WDM system. The frequency shifting can be realized by multiplying the optical signal with a virtual carrier of frequency f_n,

$$E_{\mathrm{WDM\_ch\_n}}(t) = E(t) \cdot \exp(j \cdot 2\pi \cdot f_n \cdot t)$$

where n is the channel index; $E(t)$ is the optical field before frequency shifting; t is the time grid; and $E_{\mathrm{WDM\_ch\_n}}(t)$ is the n-th frequency-shifted channel to assemble the WDM signal. If the WDM has an even grid, we can define f_n by

$$f_n = \left(n - \frac{N_{\mathrm{ch}} + 1}{2}\right) \cdot \Delta f$$

where N_{ch} is the number of channels in the WDM signal; Δf is the channel spacing. Note that the frequencies and WDM grid are defined in Hz.

Before we dive into the simulation, we need to clarify the meaning of ONSR for WDM signals. Remember the definition of OSNR

$$\mathrm{ONSR} = \frac{\text{Power of the optical signal}}{\text{Power of the noise in the reference bandwidth}} = \frac{P_s}{2N_{\mathrm{ASE}}B_{\mathrm{ref}}}$$

where P_s is the power of the optical signal. For WDM signals, P_s should refer to the overall power of the WDM signal, including the power of all WDM channels. Therefore, if every WDM channel has the same optical power, we can rewrite the equation

$$\mathrm{OSNR} = \mathrm{OSNR_{WDM}} = \frac{P_{s,\mathrm{WDM}}}{2N_{\mathrm{ASE}}B_{\mathrm{ref}}} = \frac{N_{\mathrm{ch}}P_{s,\,\mathrm{ch}}}{2N_{\mathrm{ASE}}B_{\mathrm{ref}}} = N_{\mathrm{ch}} \cdot \mathrm{OSNR_{ch}}$$

where $P_{s,\mathrm{WDM}}$ is the power of the WDM signal, $P_{s,\,\mathrm{ch}}$ is the power per WDM channel; and $\mathrm{OSNR_{ch}}$ is called OSNR per channel.

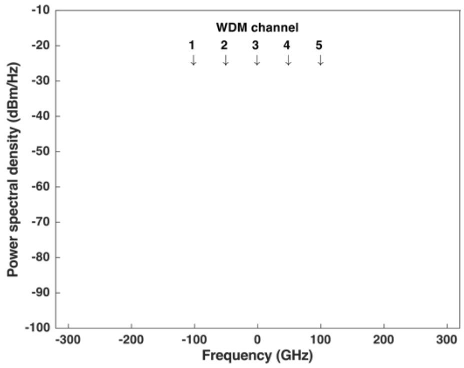
In this question, we will generate a WDM signal with 5 channels using the following params:

- Initialize the random number generator in Matlab with a seed equal to 34310: rng(34310)
- Modulation format: OOK
- Signal baud rate, $R_s = 10e9$ baud, i.e., 10 Gbaud
- Channel spacing: 50 GHz
- Number of samples per symbol, $N_{ss} = 64$
- Number of symbols, $N_{\mathrm{sym}} = 1e5$ symbols
- Detection scheme: direct detection for the target channel using an optical filter and a single photodetector.

1. Question:
   - Why the samples per symbol, N_{ss}, is increased, compared with single-channel simulation in Exercise 1 of this assignment?
   - Calculate the overall bandwidth of the WDM signal and the sampling frequency for the simulation.
   - Explain why we choose $N_{ss} = 64$, not $N_{ss} = 16$.
   - What is the minimum N_{ss} can we use in this simulation?
2. Generate the WDM signal.
3. Plot the power spectral density of the WDM signal using the following code template.
   Check if each channel is supposed to be on the correct grid.

Code template:
```
figure
plot(F/1e9, PSD, 'linewidth',1, 'Color',[0 0 1 0.2]) % x axis in GHz
axis([-Fs/2/1e9 Fs/2/1e9 floor((max(PSD)-70)/10)*10 floor((max(PSD)+20)/10)*10])
maxPSD = max(PSD);
text(0e9/1e9, maxPSD+10, 'WDM channel', 'HorizontalAlignment', 'center',
'VerticalAlignment', 'baseline')
text(-100e9/1e9, maxPSD+5, [sprintf('1 \n'),'\downarrow'], 'HorizontalAlignment',
'center', 'VerticalAlignment', 'baseline')
text(-50e9/1e9, maxPSD+5, [sprintf('2 \n'),'\downarrow'], 'HorizontalAlignment', 'center',
'VerticalAlignment', 'baseline')
text(-0e9/1e9, maxPSD+5, [sprintf('3 \n'),'\downarrow'], 'HorizontalAlignment', 'center',
'VerticalAlignment', 'baseline')
text(50e9/1e9, maxPSD+5, [sprintf('4 \n'),'\downarrow'], 'HorizontalAlignment', 'center',
'VerticalAlignment', 'baseline')
text(100e9/1e9, maxPSD+5, [sprintf('5 \n'),'\downarrow'], 'HorizontalAlignment', 'center',
'VerticalAlignment', 'baseline')
xlabel('Frequency (GHz)')
ylabel('Power spectral density (dBm/Hz)')
```

Your plot should look like the following



where the channels should be right under the arrows.

4. Try detecting the WDM signal directly using a photodetector. Plot the waveform of the electrical signal spanning across several symbols. Comment on what you are seeing.
5. Add AWGN to the WDM signal with 20-dB <u>OSNR per channel</u>.
   What is the OSNR for the WDM signal?
   Plot the power spectral density of the noisy WDM signal. Mark the signal level (ignoring the large carrier at center of each channel), and the average noise level outside signal bands. Is the noise level what you have expected? Why?

<u>Question 2-2: Detection of the WDM signal</u>

In this question, we are going to detect the WDM signal. We are going to build the detection system upon the simulation in the previous question.

1. Build up a Gaussian-shaped optical filter using the following parameters:
   3-dB bandwidth: 2*Rs GHz; Gaussian order: 1; out-of-band rejection: 20 dB.
   Center the optical filter to the first channel of the WDM signal.
2. Apply the optical filter to the WDM signal.
3. Detect the extracted channel.
   a. Plot the optical power spectral density of the extracted channel.
   b. Detect the extracted channel using a single photodiode.
   c. Plot the power spectral density and eyediagrams of the detected electrical signal.
   Questions:
   - Do you see clear eyediagrams? Is this what you expected?
   - What components do you see from the power spectral density of the optical signal and the electrical signal?
   - What is the limiting factor here for performance?
4. We are going to implement a baseband filter and apply it to the detected baseband electrical signal to improve the performance of the detected signal. Note this is also an emulation of the electrical bandwidth of the photodetector.

Code for the baseband filter

```
BB_Hf_3dB                = 2*Rs;
BB_Hf_order              = 2;
BB_Hf_F0                 = BB_Hf_3dB /(2*nthroot(2*log(2),2*BB_Hf_order));
BB_Hf                    = exp(-1/2*(F./(BB_Hf_F0)).^(2*BB_Hf_order));
```

```
BB_Hf_rejction              = 20;
BB_Hf_min                   = 1/(10^(BB_Hf_rejction/20));
BB_Hf(BB_Hf <= BB_Hf_min) = BB_Hf_min;
% Apply the baseband filter
detectedSignal_filtered    = ...
         ifft(ifftshift(fftshift(fft(detectedSignal)).* BB_Hf));
```

5. Plot the eyediagrams of the baseband filtered signal. Do you see an improvement?
6. Do bit decision and BER calculation. Do you get a zero BER?
7. Questions:
   - Why do we not need to shift back the demultiplexed channel back to central frequency (0) in the simulation?
   - Why do you think electrical filtering (a bandwidth limited photodetector) could improve the performance of the detection?

     > Hint:
     > Think of the photodetection process and frequency components of the detected signal.

   - Suggest some alternative approaches other than baseband filtering to improve the performance of the detected signal.

Question 2-3: WDM system

In this question, we are going to simulate a complete point-to-point WDM transmission system.

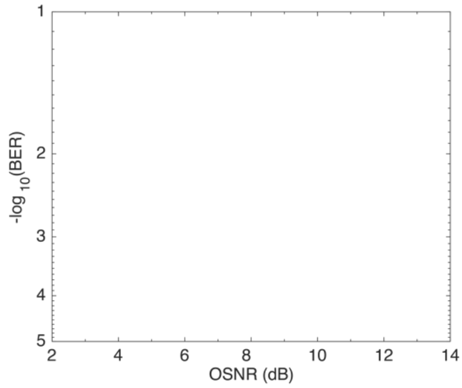
1. Build up a WDM system using the following params.
   a. Initialize the random number generator in Matlab with a seed equal to 34310: rng(34310)
   b. Modulation format: OOK
   c. Signal baud rate, $R_s = 10e9$ baud, i.e., 10 Gbaud
   d. Number of channels: 5; Channel spacing: 50 GHz
   e. Number of samples per symbol, $N_{ss} = 64$
   f. Number of symbols, $N_{\text{sym}} = 1e5$ symbols
   g. Detection scheme: direct detection for the target channel using an optical filter and a single photodetector.
   h. Noise: AWGN, with OSNR per channel iterating from 4 dB to 12 dB, with interval of 1 dB.
   i. Optical filter: 1st order Gaussian; 2*Rs 3-dB bandwidth; 20-dB out-of-band rejection.
   j. Baseband filter: 2nd order Gaussian; 2*Rs 3-dB bandwidth; 20-dB out-of-band rejection.
2. Plot the BER curves for all 5 WDM channels.

   > Code template:
   > Refer to the following example code to plot the OSNR-BER curve so the OSNR and BER are nicely shown.

```
% Plot the OSNR-BER curve
plot(OSNR_iter_dB, -log10(BER), '-*')
% set common styles
h = gca;
limX                    = [2, 14];
tickDivX                = 2;
minorTickDivX           = 1;
limY                    = [1, 5];
tickDivY                = 1;
minorTickDivY           = 0.1;
h.YDir                  = 'reverse';
h.FontName              = 'Helvetica';
h.FontSize              = 14;
h.XLabel.String         = 'OSNR per channel (dB)';
```

```
% Font automatically adjusted according to global h.FontName and h.FontSize
h.YLabel.String         = '-log_{10}(BER)';
% But the FontSize is 1.1 times larger by default.
h.XLabel.FontSize       = 14;
h.YLabel.FontSize       = 14;
%
% set specific x and y axis styles, such as TickLabelRotation
h.XAxis.Scale           = 'linear';
h.XAxis.Limits          = limX;
h.XAxis.TickValues      = (limX(1):tickDivX:limX(2));
h.XAxis.MinorTick       = 'on';
h.XAxis.MinorTickValues = limX(1):minorTickDivX:limX(2);
h.XAxis.TickLabelFormat = '%g';
%
h.YAxis.Scale           = 'log';
h.YAxis.Limits          = limY;
h.YAxis.TickValues      = (ceil(limY(1)):tickDivY:limY(2));
h.YAxis.MinorTick       = 'on';
h.YAxis.MinorTickValues = limY(1):minorTickDivY:limY(2);
h.YAxis.TickLabelFormat = '%g';
```

The appearance of your plot should look like the following.



Please plot all 5 WDM channels in the same figure.

3. Compare the results with the single-channel system in Exercise 1:
   - Why does the WDM system perform better than the single-channel system? Does this make sense to you?
   - What's the difference in the codes? Is there anything missing in the single-channel simulation?
   - Can you get the single-channel system to have a similar if not better performance?

Appendix I – Summary of quantities

$\mathscr{A}$:	Symbol alphabet
a_k	symbol at instant k
g(t):	Pulse shape
t:	time
T_0:	Power e^{-1} width of a Gaussian pulse
T_{FWHM}:	Power full width half maximum width of a Gaussian pulse
T_s:	Symbol period
f_{ref}:	Reference frequency for baseband representation of optical fields
$E_{LO}(t)$:	Optical field of local oscillator laser
f_{LO}:	Baseband frequency of the local oscillator (with reference to f_{ref})
$E_S(t)$:	Optical field of signal laser
$d_i(t)$:	Normalized and zero-mean electrical analog signal for driving MZM
V_{pp}:	Peak-to-peak voltage for driving MZM
V_{dc}:	DC (bias) voltage offset for driving MZM
$V_i(t)$:	Analogue voltage signal (rescaled and with offset) used to drive the MZM
$E_{out}^{(i)}$:	Optical field (baseband representation) at the output of MZM i
E_{tx}:	Optical field (baseband representation) at the output of a nested MZM
E_{fiber}:	Optical field (baseband representation) at the output of the optical fiber
i_I:	Current at the output of the in-phase balanced detector in a coherent receiver
i_Q:	Current at the output of the in-quadrature balanced detector in a coherent receiver
I_{rx}:	Complex analog current signal corresponding to the reconstructed optical field after coherent reception
I_{FO}:	Complex analogue current signal after frequency-offset correction
f_1:	Generic frequency value
I_{EDC}:	Complex analogue current signal after electronic dispersion compensation

Appendix II – Time/Frequency grid and DFT

N: Total number of samples

n: Sample number

T_W: Total time window

ΔT: Sampling period

t_{min}: Start time of time window

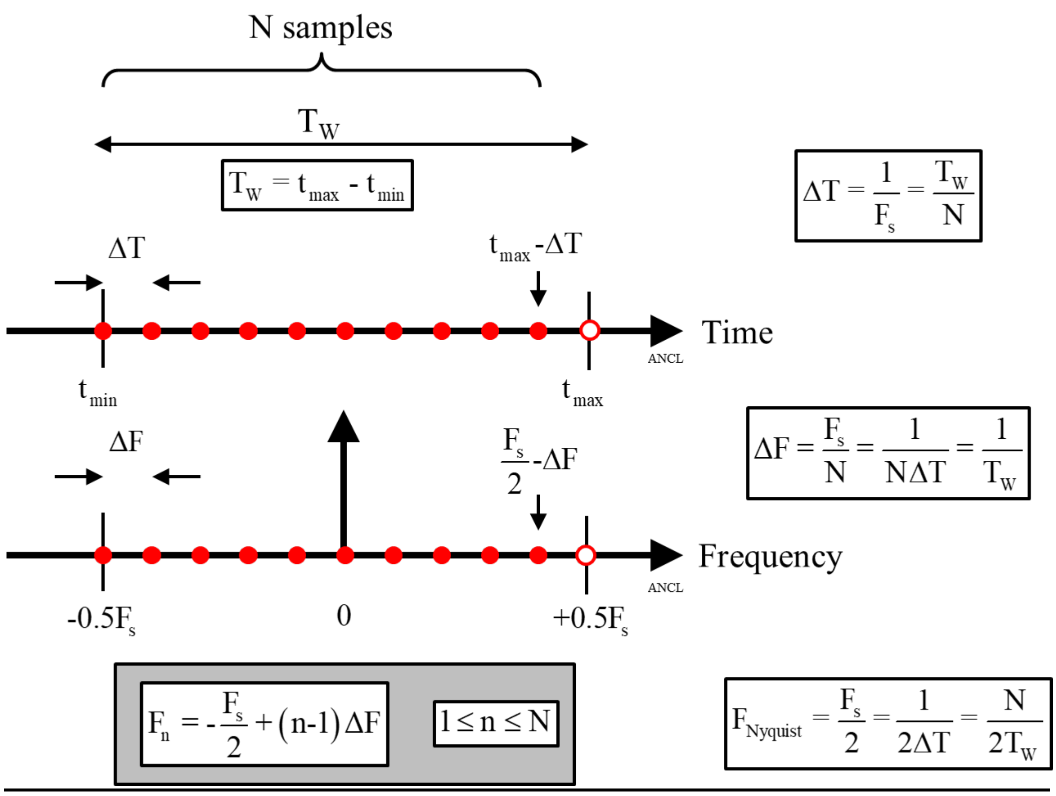
t_{max}: End time of time window

F_{sa}: Sampling frequency
In this assignment, it is denoted as sampling rate, F_s.

ΔF: Sampling distance frequency

$F_{Nyquist}$: Nyquist frequency

F_n: Frequency at sample n



Appendix III – Hand-in procedure

Please hand in a zip file named as "**34130_A3_<firstName>_<lastName>_<studentNumber>.zip**"

My hand-in would be named "34130_A3_Deming_Kong_sxxxxxx.zip"

Within the zip file include at least:

1- A **pdf file** containing your report (named with the same convention explained above). The PDF should be searchable and readable by a screen reader, i.e., "machine readable". The name of the report should include name and assignment number. Furthermore, if several versions are uploaded, name the file with a version number. An example of name could be 34130 E23 Deming Assignment 03 ver01.

2- A **script file** with the code you used to solve the assignment. The code does not need to be clean, nor optimized, but please add a few comments to allow the reader to follow the train of thoughts behind the code.
Your code can be divided into multiple Matlab script files. For example, you can work on Exercise 1, Question 1 and 2 in one script file. And you can work on Exercise 1, Question 3 in another script file. You can name those files accordingly. E.g.,
    E23_Deming_Assignment_03_Exercise_1_Question_1_and_2.m
    E23_Deming_Assignment_03_Exercise_1_Question_3.m
    E23_Deming_Assignment_03_Exercise_2_Question_1_and_2.m
    …
The scripts should run without errors and produce the figures as in your report.