POLITECNICO
MILANO 1863

The Mach-Zenhder Modulator

Simulations, comparisons and discussion

Roberto Scardia

June 2023

Table of Content

The Mach-Zehnder Modulator

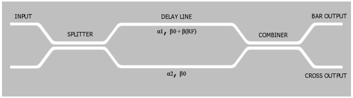


Figure: A schematic of a Mach-Zenhder Modulator. The names of all the components and outputs used in this document are indicated here

The Mach-Zehnder modulator is an optical modulator based on the functioning of the Mach-Zehnder interferometer, in which the incoming light beam is divided into two different paths with a relative phase shift and then recombined.

This recombination results in an interference pattern, which can be leveraged for modulation purposes. The device itself is constructed using two pair of coupled optical guides (the splitter and the combiner) and two independent optical waveguides.

The Pockels effect is exploited to introduce a phase shift in one of the guides. The guides are typically made of lithium niobate with a length in the order of a few centimeters.

In the following analyses, it is considered an MZ modulator in lithium niobate with an $r33 = 30.8 pm/V$, $no = 2.210$ and a length of 5 centimeters.

Scripts' Structure

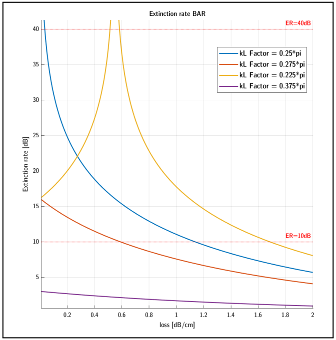

```matlab
phi0 = ne*(Length).*(2*pi/wavelength);

RF_pi = (wavelength*gap)/((ne^3)*confinment_factor*r33*Length);
E_i_laser = 1*V;

%% SETTINGS : fixed coupling coefficient, varying alpha

% Set to 1, 0 or -1 to set the value of the attenuation constant of the
% second line.
% 1  - the second line has the same attenuation constant as the first
% 0  - the second line has an attenuation constant of 0
% -1 - the second line has a fixed attenuation constant
% Default = 0
same_loss_second_line  = -1;


% Fixed attenuation constant for the second line in [dB/cm].
% Irrelevant if 'a_same_loss_second_line' is other than -1.
% Default = 0 [dB/cm]
fixed_loss_second_line = 0.5*(dB/cm);

% Coupling coefficient splitter
kL_factor_s = [pi/4, (pi/4)*1.10, (pi/4)*0.90, (pi/4)*1.5];

% Coupling coefficient combiner
kL_factor_c = [pi/4, (pi/4)*1.10, (pi/4)*0.90, (pi/4)*1.5];

assert(numel(kL_factor_c) >= numel(kL_factor_s), "kL_factor c has less element than kL_factor_s ")

%% EXTINTION RATE SIMULATION : fixed coupling coefficient, varying alpha
```

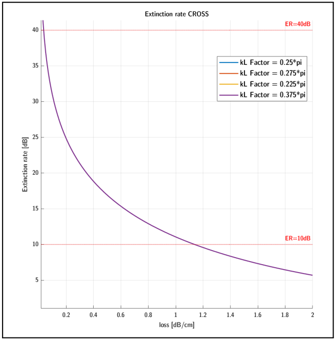
Figure: An example of the setting section of the *MZS_ER_alpha.m* script

MZS_ER_alpha.m, *MZS_ER_kL.m*, *MZS_ER_RF.m* : These scripts calculate the extinction rate of a Mach-Zehnder modulator sweeping over three groups of parameters: the coupling factor of the splitter and combiner, the attenuation constant of the two guidelines and the value of the RF digital input signal corresponding to a "high" logical value.

Each script has its own setting section where the parameters of the computation can be changed.

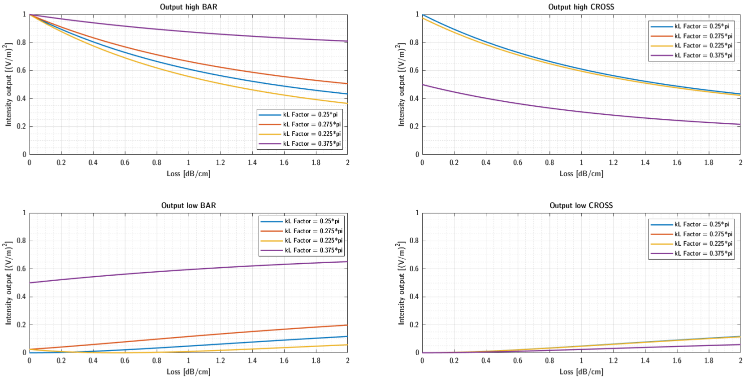
Extinction rate over α: symmetrical combiner/splitter



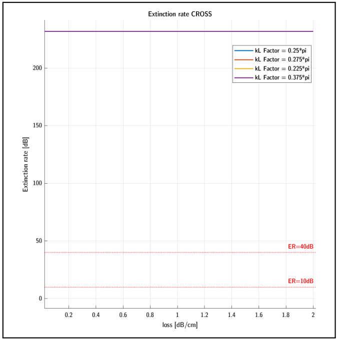


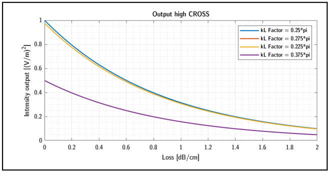
- Here splitter and combiner have the same coupling factor. The second line has an attenuation constant of 0 dB/cm.
- The values of coupling factor used are $\frac{\pi}{4}$, 90% of $\frac{\pi}{4}$, 110% of $\frac{\pi}{4}$ and 150% of $\frac{\pi}{4}$.
- The extinction rate depends only on the α constant of the delay line.

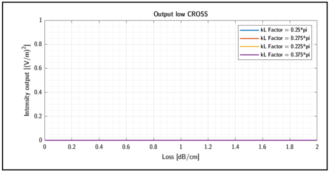
Intensity over α: symmetrical combiner/splitter



Same attenuation constant for both lines

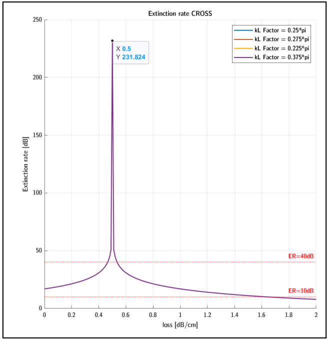


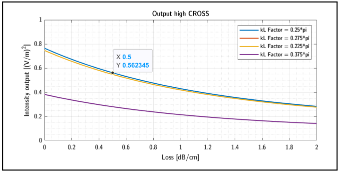


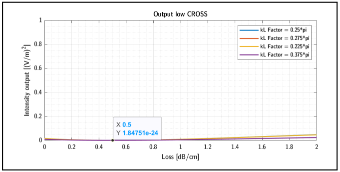


In this scenario, both lines have the same attenuation constant.
It can be seen that the Extinction Rate of the Cross output port tends towards infinity and the Intensity output *low* remains consistently 0 for all values of the α constant.

Extinction rate over α: fixed α for second line

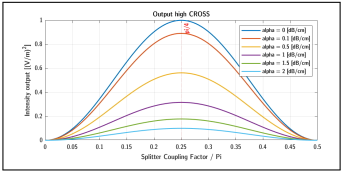


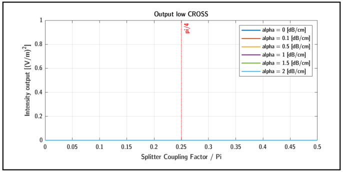




Here the second line has a fixed attenuation constant of 0.5$[dB/cm]$. In order to maximize the extinction rate the α constant of the first line needs to follow the value of the second line. However, the attenuation constant of the first line negatively affects the value of the intensity from the output.

Extinction rate over coupling factor: Cross

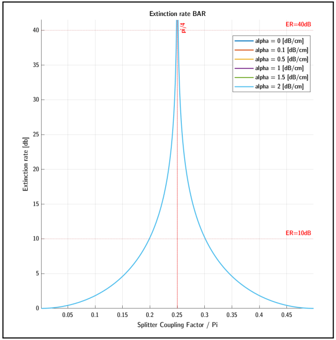


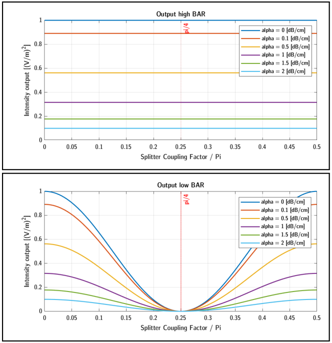


- in this scenario splitter and combiner share the **same coupling factor** and both lines have the **same α constant**, the ***low*** **Intensity output drops to near 0 regardless of the value of α and the coupling factor**, thus bringing up to infinity the ext. rate.

- It must be noted that the *high* Intensity output has **its maximums in the neighbourhood of $\pi/4$ and for lower values of α**.

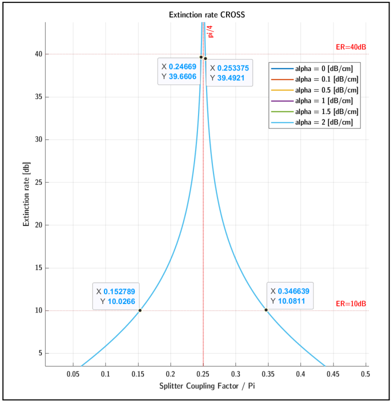
Extinction rate over coupling factor: Bar



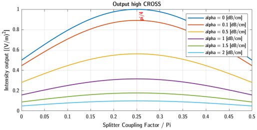


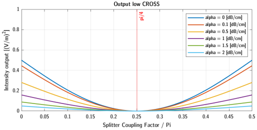
Instead, the extinction rate of the Bar port heavily depends on the coupling factor even when the splitter and combiner have the same value.

Extinction rate over coupling factor



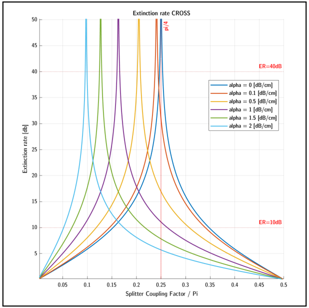
For this plot the script's parameters have been set such that the **combiner's coupling factor is fixed at $\frac{\pi}{4}$**. Both lines still have **the same α constant**. We can observe a behavior of the extinction rate similar to that of the previous graph. The modulator can tolerate an error of 1.3% in the coupling factor of the splitter to have an extinction rate of $> 40dB$, and an error of 38% to be up $10dB$.

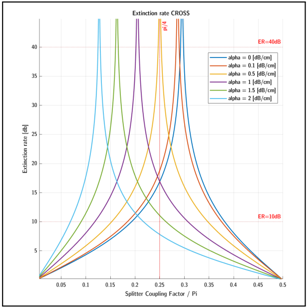




Extinction rate over coupling factor: fixed α second line

Setting fixed both the α constant of the second line and the combiner's coupling factor gives instead the following plots:

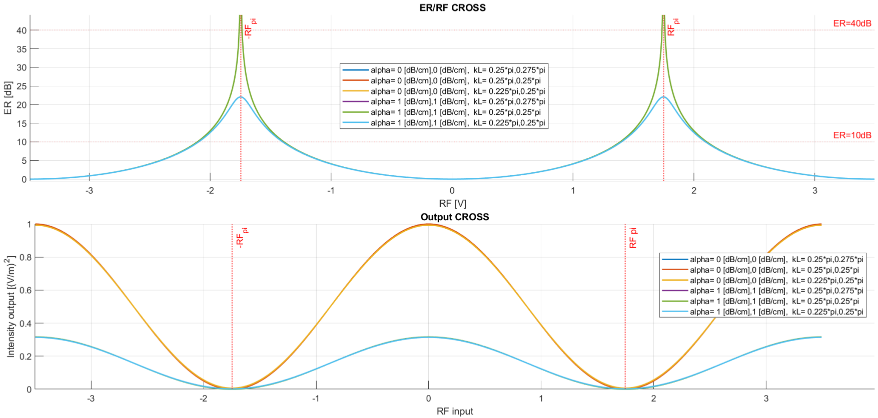




In this graph, the second line has an α of $0dB$ and the combiner coupling factor is $\frac{\pi}{4}$.
In the legend the value of the delay line attenuation constant.

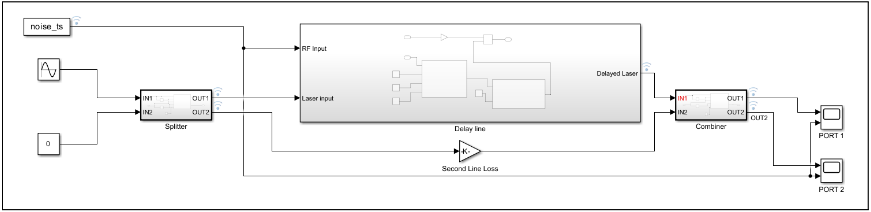
In this graph, the second line has an α of $0.5dB$ and the combiner coupling factor is $\frac{\pi}{4}$.
In the legend the value of the delay line attenuation constant.

Extinction rate over RF



These graphs illustrate the trend of the extinction rate and output intensity in the *Cross* output port spanning over the RF *high* digital voltage value.

Simulink Model

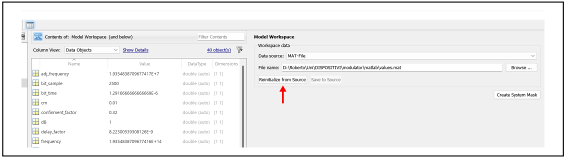


This Simulink model emulates the real-time functionality of a Mach-Zehnder. It comprises three sub-systems: the ***delay line***, the ***splitter*** and the ***combiner***. The model makes use of the additional external Matlab function *time_delay.m* to computer the delay of the laser input signal given a phase and the *generate_input.m* to generate a random RF digital input signal both with and without white Gaussian noise. This generation is dependent on parameters such as the simulation's **sample time**, the number of **samples per bit**, the **length of the input in bits**, the ***high* digital input voltage** and **noise's the standard deviation**. In order to operate, the model retrieves the input and its operational parameters from the Simulink model workspace. The script *initial_value.m* is provided to generate a random input and configure the model's variables accordingly.

Configuration


```
%% LASER VALUES
wavelength  = 1550*nm;                  % LASER WAVELENGTH
frequency   = (3e8)/wavelength;         % LASER FREQUENCY
time_period = 1/frequency;              % LASER PERIOD

%% SIMULATION VALUES (DO NOT MODIFY)
prop_factor         = 10000000;                 % FREQUENCY/SIMULATION FREQUENCY FACTOR
adj_frequency       = frequency/prop_factor;    % SIMULATION FREQUENCY
sample_frequency    = adj_frequency*100;        % SAMPLE FREQUENCY
sample_time         = 1/sample_frequency;       % SAMPLE TIME
delay_factor        = 1/(2*pi*adj_frequency);   % COMMON USED VALUE

%% WAVEGUIDE VALUES
kL_factor_s = pi/4; % COUPLING FACTOR SPLITTER
kL_factor_c = pi/4; % COUPLING FACTOR COMBINER

loss_1  = 0*dB/m; % ATTENUATION COEFFICIENT FIRST LINE
loss_2  = 0*dB/m; % ATTENUATION COEFFICIENT FIRST LINE

Length  = 5*cm; % LINES LENGHT
gap     = 6*um; % GAP BETWEEN DELAY LINE'S ELECTRODE

r33                 = 30.8e-12;     % LiNbO3 r33
n0                  = 2.2111;       % LiNbO3 ordinary refractive index
confinment_factor   = 0.32;         % GUIDE'S CONFINMENT FACTOR

V_pi = (wavelength*gap)/((n0^3)*confinment_factor*r33*Length); % PI VOLTAGE

loss_factor_1 = exp(-(loss_1/8.6860000037)*Length);
loss_factor_2 = exp(-(loss_2/8.6860000037)*Length);
```

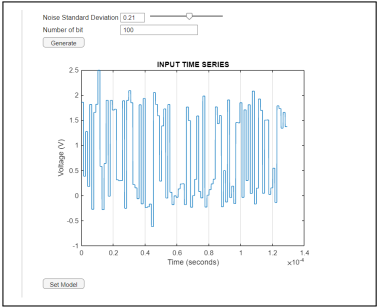
```
%% INPUT, SAMPLE AND SIMULTATION TIME
period      = 1/adj_frequency;                  % ADJUSTED LASER PERIOD
n_camp      = 25*round(period/sample_time);     % NUMBER OF SAMPLE

bit_sample  = n_camp;                       % SAMPLE NUMBER PER BIT
input_dim   = 40;                           % INPUT'S BITS NUMBER
bit_time    = sample_time*bit_sample;       % BIT TIME
total_time  = input_dim*bit_time;           % TOTAL SIMULATION TIME

std_dev = 0.1;                              % NOISE STANDARD DEVIATION

set_param('MZexample', 'StopTime', string(total_time));        GENERATE INPUT

[noise_ts, unalt_ts] = generate_input(sample_time, bit_sample, input_dim, V_pi, std_dev);

save("values")  ← SAVING VALUES
```



1. Set the desired parameters inside the *initial_value.m* script
2. Run the script: a *values.mat* file will be generated
3. In Simulink: Explore→ MZexample→ Model Workspace→ Reinitialize from Source

Generate Input

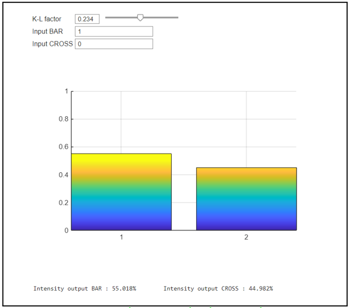
To simplify the analysis and the use of the model, three Matlab LiveScripts are provided.
Generate_input_script.mlx can be used to visualize and directly set the model's input.



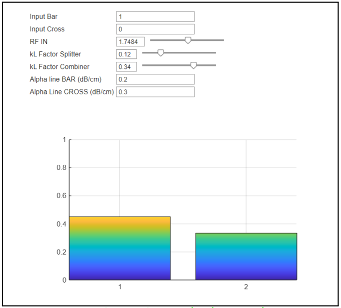
A new input can be generated by clicking on the ***Generate*** button and loaded into the model's workspace with the ***Set Model*** button

Intensity output

The *splitter_calculator.mlx* calculates and visually represents the percentage ratio of output intensities from a splitter/combiner compared to the total intensity of the input, given the coupling factor and the inputs' intensity in V^2/m^2. Likewise, the *intensity_calculator.mlx* script provides similar functionality, allowing the user to visualize the ratio of output intensities for the entire modulator, considering appropriate parameters.

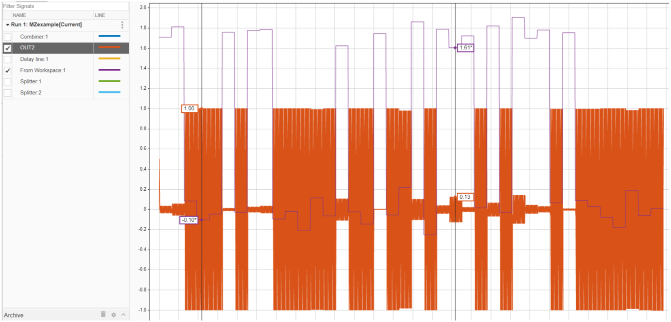


splitter_calculator.mlx



intensity_calculator.mlx

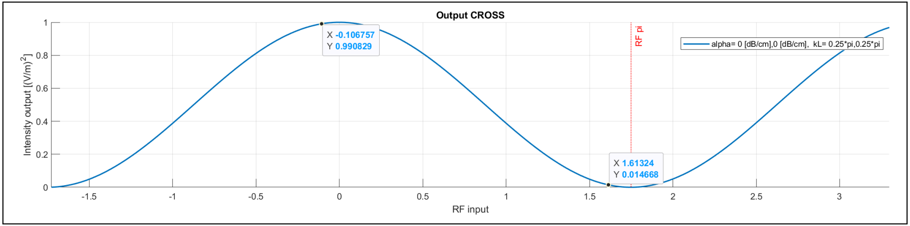
Output Example



This is an example of a run with a random digital input disturbed by white noise with a standard deviation of 0.1.
The violet stroke is the *RF input*, and the orange one is the *module of the electric field* at the cross output.

The effect of the noise

In the previous slide, it can be noted that the noise has a greater effect on the output when the RF input is high than when it is low. This fits the result of the previous Matlab script:



$\rightarrow$

$\sqrt{0.990829} = 0.99540 \sim 1.00!$

$\sqrt{0.014668} = 0.12111$

Output Frequency Analysis

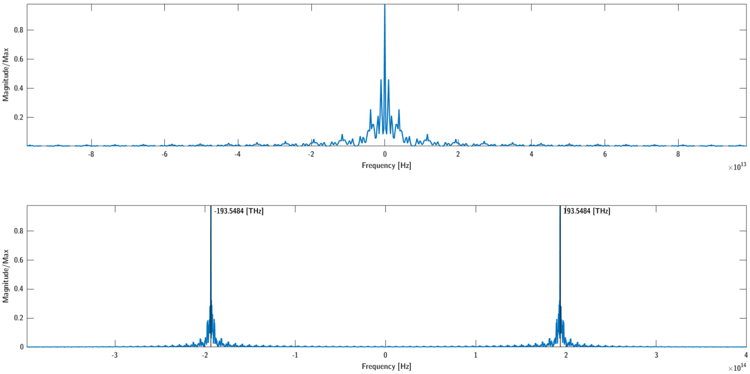
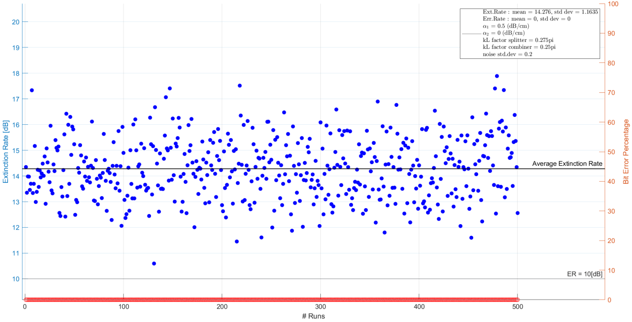


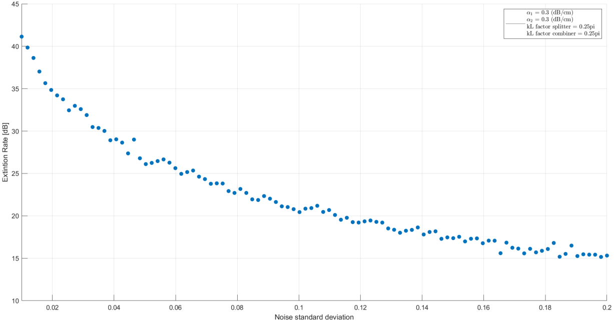
Figure: Fourier transform of a random digital input and its modulated result on the cross output

Mean Extinction Rate



The *LS_base.m* script calculates the mean extinction rate of the simulated modulator over a certain number of runs with a noise-ridden input signal. To use the script it is advised to install the Parallel Computing Toolbox.

Extinction rate over noise



The *LS_noise.m* script calculates the extinction rate sweeping over the noise's standard deviation. In this case for every value of standard deviation, the ext. rate is calculated with an input of 500 bits. To use the script it is advised to install the Parallel Computing Toolbox.