POLITECNICO DI TORINO

Collegio di Ingegneria Elettronica, delle Telecomunicazioni e Fisica (ETF)

Master degree course in Nanotechnologies for ICTs

Master Degree Thesis

Analysis of Silicon Photonics Neuromorphic Networks

Supervisors
prof. Paolo Bardella
prof. Andrea Carena

Candidate
Marco Orlandin
matricola: 290237

Academic Year 2022-2023

Contents

Summary

The demand for new artificial intelligence applications is growing exponentially and requires a significant increase in computational power with improved speed while reducing energy consumption. In this scenario, Neuromorphic Computing (NC) is a very active research field in the domain of Artificial Neural Networks (ANNs): elements of a computer are modeled after the human brain and nervous system. For NC applications, Silicon photonics, i.e. the integration of photonic waveguides and components on a Silicon substrate, can represent a valid solution due to its high speed, compatibility with CMOS technology, low power consumption, and reduced footprint. Circuits made with photonic devices, such as optical modulators, can be used in layers of NC as matrices for the calculation of the weights, and they can be implemented next to standard CMOS circuits to work assemble in the neural network.

Device description
The study carried out in this thesis is focused on the analysis of a photonic circuit designed by a research group at of the Electrical and Photonics Engineering High-Speed Optical Communications Department of the Technical University of Denmark (DTU). The circuit was originally conceived as a 7×7 optical fiber switch; a part of it was later used for neuromorphic computation creating a system with three optical inputs and three optical outputs. A mask-level representation of the 3×3 neuromorphic subcircuit is shown in Fig. 1, with the indication of the input and output ports and the assigned MZI labels. The output optical powers are linear combinations of the input optical signals, with weights that can be modified according to the voltages applied to one arm the Mach-Zehnder interferometers (MZIs) available in the circuit and causing therefore a phase shift of the optical beam.

Experimental characterizations of the network have been carried out at DTU, injecting a signal at one port and measuring the powers at three output ports, while changing between 0 V and 2 V the voltage of each MZI, one at the time. The experimental evidence allowed us to identify a main problem: the thermo-optic modulators present in the

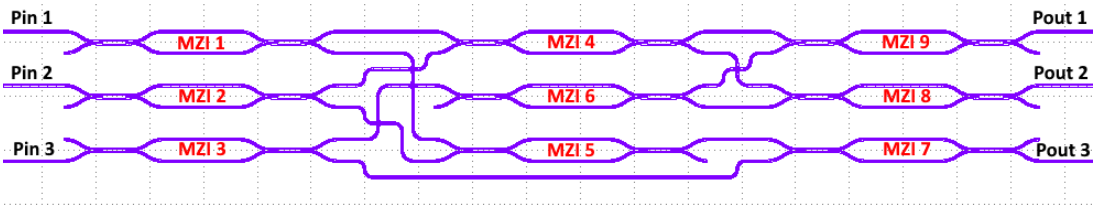


Figure 1: Mask-level representation of the 3×3 circuit used in the experiment.

circuit are strongly affected by thermal crosstalk: the voltage applied to a single MZI causes an unwanted phase shift also in the adjacent ones, resulting in a non-accurate combination of the input signals. This problem can be partially solved increasing the distances between MZIs, but this solution increases the footprint and limits the number of photonic devices that can be integrated into a single chip. To maintain the large-scale integration typical of photonic components, a more effective approach is to develop a model that effectively includes the thermal crosstalk in the control strategy: this is indeed the goal of this thesis.

Device analysis

In order to develop a model capable of accurately predicting output power as a function of the nine applied voltages, I operated at different levels.

First, I studied the input-output connections and wrote a MATLAB code to calculate the output power as a function of the input and nominal phase shift in the MZI. This allowed us to estimate the power at the output ports in the case of ideal components.

Then, to improve the accuracy of the results, I used the Beam propagation Method (BPM) simulations, carried out in Synopsys RSoft, to precisely estimate the behavior of the optical components present in the circuit, in particular MZIs and Multi-Mode Interferometers (MMIs), estimating losses correlating the optical phase shift to voltage applied to one branch. In addition, waveguide cross-connections were also simulated to estimate the introduced optical losses. Examples of these simulations are shown in Fig. 2.These analysis allowed us to insert the effect of such losses into the MATLAB model. A proper tuning of each

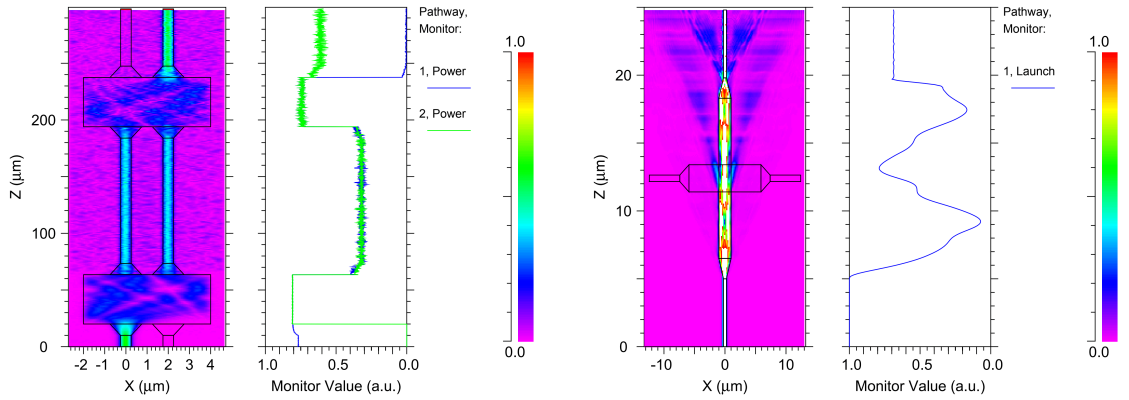
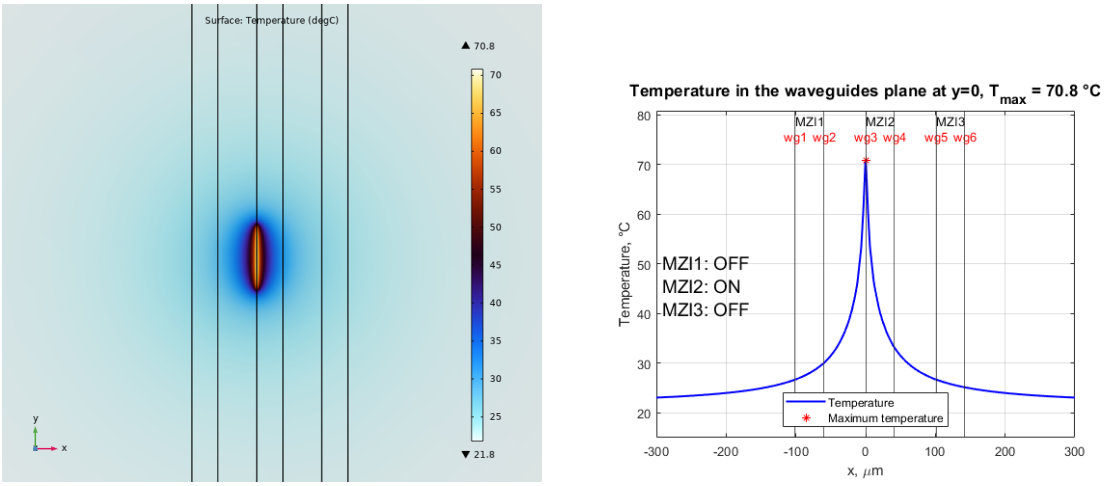


Figure 2: Example of Synopsys RSoft BPM simulations: (a) cascade of two MMIs, (b) cross-connections.

MZIs' properties was required to improve the fitting of the experimental data.

Additionally, I focused on the thermal analysis of the MZIs, to identify the causes of the crosstalk and quantify its amount. I used the Finite Elements Method to thermally simulate a column of three MZIs in COMSOL Multiphysics, applying voltages to the heaters and observing the resulting heat distribution in a 3D representation. Examples of two thermal simulations are presented in Fig. 3. The thermal profiles obtained were used to map the voltage into a temperature change and finally into a refractive index and optical phase variation, allowing us to use the real voltage instead of the phases as a control signal. Moreover, crosstalk was properly modeled estimating the temperature change induced by voltage not only on the driven MZI but also in the adjacent ones.

Finally, the results obtained with the modified MATLAB code were compared with the experiments, shown a good agreement between prediction and measured results. As an example, Fig.4 shows a comparison between predicted output power at port 2 with the input at port 1, as a function of the voltage applied individually to the nine MZIs. Each vertical band highlights a single MZI, with a voltage tuned between 0 V and 2 V.



(a) (b)

Figure 3: Example of COMSOL Multiphysics simulation for the thermal analysis: (a) 2 V applied in the central MZI, (b) Temperature profile at the plane of the waveguides.

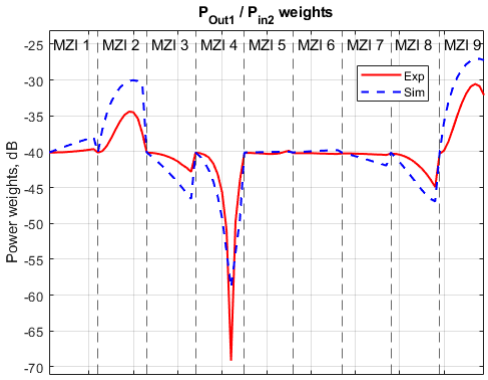


Figure 4: Comparison between predicted and measured output power as a function of the applied voltage

.

Chapter 1

Introduction

1.1 An overview

In the last decade, artificial intelligence (AI) has started to permeate everyday life to complete tasks emulating the human brain. Examples are everywhere from our smartphones to cars, most of the computing tasks and sensoring are analyzed by AI: biometrics for personal recognition, language recognition in Amazon's Alexa, Google AI, and chatGPT, sensors in cars, and also present in computer electronics to optimize processes in CPUs and GPUs [1].

Actually, in the present and near future, AI will be massively implemented in areas such as transportation, will be important for a continuous analysis of the sensors in electrical cars; Health care is one of the most promising work areas, which, can improve thanks to AI, in fact, a fast diagnosis recognition for health issues, such as heart attack or in the media services, where the AI, already now, helps costumers in choices. These are just a few examples among the huge number of AI applications at the moment and will be probably used in the future [1].

1.2 Human brain functionality

Our brain can solve complex tasks with remarkable efficiency, much higher than modern computers. The human brain processes information in parallel thanks to billions of neurons connected via axons and

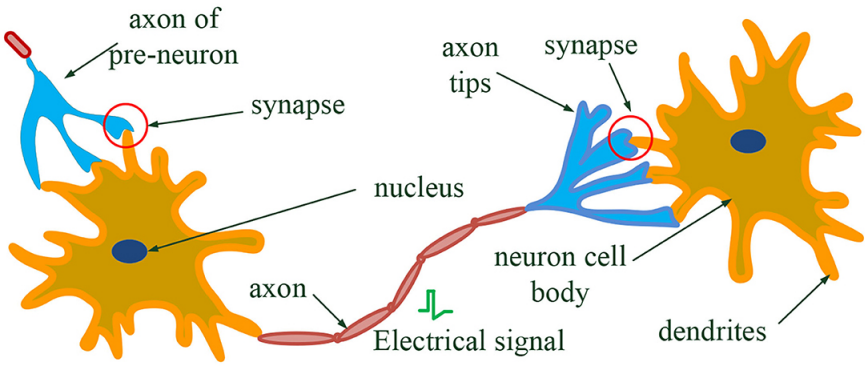


Figure 1.1: Description of the building blocks of the human brain taken from [2]

synapses [2]. The synapse varies the strength of the electrical signal that is transmitted between two neurons.

This process is mediated by excitatory and inhibitory hormones, which allow a weight variation in the synapse. The phenomenon of weight modification is called spike-timing-dependent-plasticity and two mechanisms are involved: Long-Term Potentiation (LTP) and Long-Term Depression (LTD) which change the strengths of the neurons' connections [2].

1.3 Artificial neural network

As specified above, the human brain is optimized to manage perceptual (such as face recognition), cognitive, and control tasks such as body functions. The human brain exploits a parallel computing system thanks to the billions of interconnected neurons. Artificial Neural Networks (ANN) are mathematical generalizations of this concept. The elements of neural networks are the artificial neurons, or nodes, that can work in parallel to compute a task. The effect of synapses in the ANN is characterized by weighted blocks that modulate the signal and a transfer function exploits the behavior of nonlinear neurons [3]. The output is the weighted sum of the inputs transformed by the transfer function. But the real capability is the ability of the neural network to adjust the weights according to the learning process algorithm [3]. The

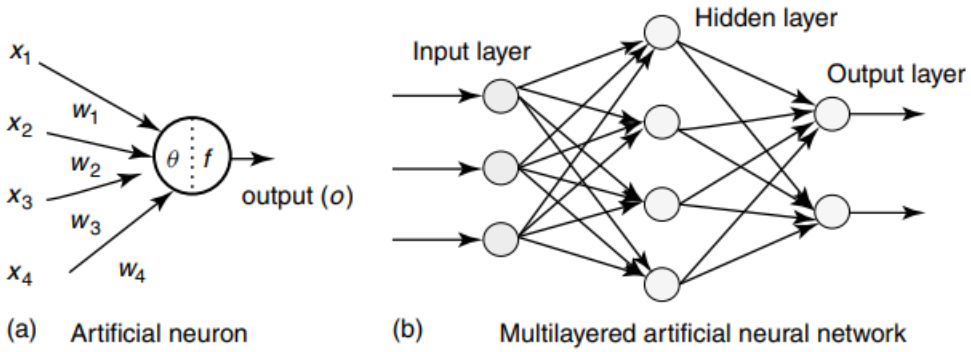


Figure 1.2: Artificial neuron and multilayered artificial neural structure taken from [3]

architecture is based on an input layer, a hidden layer, and an output layer, as in Fig. 1.2. The signals flow from the inputs to the outputs, we can have multiple layers, but the flow stays unidirectional. Some recurrent networks have feedback connections instead. The equation that gives the neuron output signal is

$$O = f(w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \theta) \tag{1.1}$$

A configuration of ANN can be seen as a set of inputs giving a set of outputs. The weights can be changed a priori or the ANN can be trained with teaching patterns and let them change with the learning process. To correctly train a neural network, a wide range of examples is needed, to satisfy the good generalization showing different characteristics of the problem, and a bad generalization leading to unpredictability [3]. Also, the number of neurons must be chosen wisely, because it affects the ability of the network to separate data. A large number of neurons will give good learning and prediction of trained data, but the generalization of new data is compromised. But a low number of neurons compromises the learning process and the prediction error arises [3]. ANN is then widely implemented and makes assist in new technology improvements and research such as neuromorphic computing.

1.4 Neuromorphic computing

Neuromorphic computing is an area of research that aims to create computer systems based on the neural structure of the human brain. It is a field of artificial intelligence that focuses on building machines that can process information in a way that mimics the way the brain works.

In traditional computers, the information is processed using a binary code, while neuromorphic computing uses neurons and synapses to perform complex calculations. The idea behind this concept is to create electronic circuits that can simulate the behavior of biological neurons and synapses, allowing computers to perform tasks that are currently difficult or impossible for traditional computers.

If the computing time for binary hardware may depend on the complexity of the task, neuromorphic computing has ideally no timescale with complexity since all the network works in parallel. In general, the benefits of neuromorphic computing include low power consumption, high-speed processing, and, as said before, the ability to handle large amounts of data in parallel. This solution is particularly useful for all the applications where a large amount of data needs to be processed.

Neuromorphic computing is still a relatively new field and there are many challenges to overcome to create practical neuromorphic devices. However, with ongoing research and development, neuromorphic computing has the potential to be a revolution in the field of artificial intelligence and create new opportunities for computing and data processing.

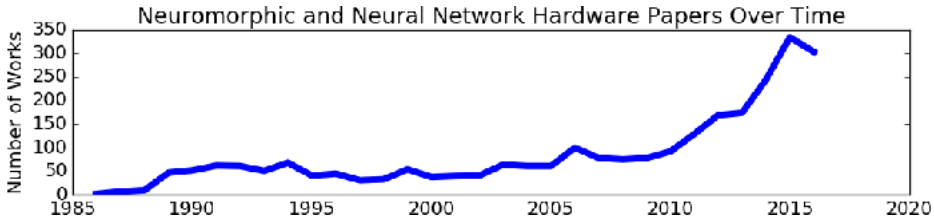


Figure 1.3: It is shown how neuromorphic computing has an increased interest in the last decade taken from [4]

1.4.1 Memristor

The explosion of neuromorphic computing studies favored the research of electrical components that mimic the neurons' and synapses' behavior. One of the most promising devices that can be used is the memristor, or "memory resistor", an electronic storage device theorized in 1971. It is a two-terminal electrical component that can remember the amount of charge that has passed through it in the past[1].

The operation of a memristor is based on the fact that certain materials can change their resistance when an electric field is applied. In particular, some metal oxides, such as Titanium Dioxide, present this behavior, and they are commonly used as active materials in memristors.

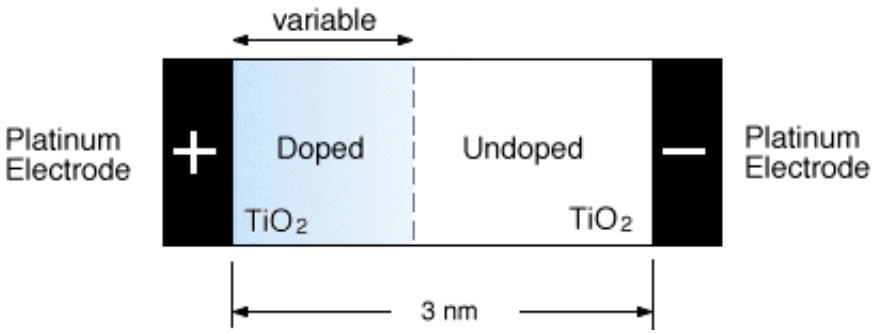


Figure 1.4: sketch of a TiO_2 based memristor taken from [5]

The resistance of a memristor changes in response to the magnitude and direction of the current passing through it. If the current flows in one direction, the resistance increases; if the current flows in the opposite direction, the resistance decreases. Moreover, this resistance change is persistent, meaning that the memristor will "remember" its resistance value even after the current has been turned off.

In addition, the resistance of a memristor can be modeled with a function that describes the current flowing through it to the voltage across it. This function is not linear but rather depends on the history of the current and voltage that has passed through the device, thanks to the hysteresis behavior of this component. This non-linear behavior is what gives the memristor its ability to store information.

They can be used in neural networks and in particular, neuromorphic computing, where their ability to remember past states can be used to perform complex calculations and learn from experience.

Anyway, memristors have also drawbacks since the speed is very limited, it is challenging to build circuits with this electronic device, and, being a passive element, it cannot store or generate power.

1.5 Silicon

In photonics, many materials are used for their structures and properties. We have to differentiate between, single-crystal semiconductors, which are the most active in optoelectronic devices, usually based on a junction, polycrystalline and amorphous semiconductors, used mainly in solar cells and LEDs, other crystals for passive devices, such as waveguides and modulators with ferroelectric, piezoelectric characteristics, glasses for waveguides and, liquid crystals for displays. Semiconductor crystals have essentially two atomic structures: cubic and hexagonal. In the crystal, the structure is regular, atoms have periodic arrangements in 3D space and a point set r defines the Bravais lattice, $r = ka_1 + la_2 + ma_3$, where k, l and m are integer numbers and a_1, a_2 and a_3 are the primitive vectors denoting the primitive cell. Fourteen Bravais lattices exist, but the one we are interested in is cubic. The cubic one has the three primitive vectors equals, and the angles are 90°; cubic crystal structure examples are Silicon, Germanium, and Gallium Arsenide. There are three cubic lattices: simple, face-centered, and body-centered, and, in addition, a cubic structure can be derived from the penetration of two shifted face-centered Bravais lattices, as shown in Fig. 1.5.

This particular structure is called a "diamond lattice", since it is the carbon atomic arrangement in diamonds and is present also in crystalline silicon which is the main material used in semiconductor electronics and, specifically, in silicon photonics. This diamond lattice has two inter-penetrating face-centered Bravais lattices, displaced in the diagonal of the cell at 1/4 of the diagonal length.

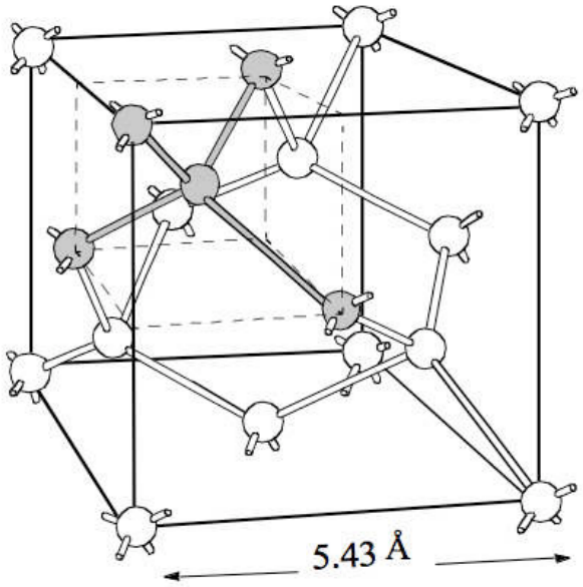


Figure 1.5: crystal structure of Silicon taken from [7]

Silicon properties			
Property	Value	Property	Value
Structure	Cubic	Dielectric constant	11.9
Energy gap	1.12 eV	Minimum direct gap	3.4 eV
Melting point	1414 °C	Boiling point	3265 °C
Specific heat	$0.7\,\mathrm{J\,g^{-1}\,K^{-1}}$	Thermal conductivity	$148\,\mathrm{W\,m^{-1}\,K^{-1}}$
Refractive index	3.42	Thermo-optic coefficient	$1.8 \times 10^{-4}\,\mathrm{K^{-1}}$

Table 1.1: Silicon properties at 300 K, taken from [8]

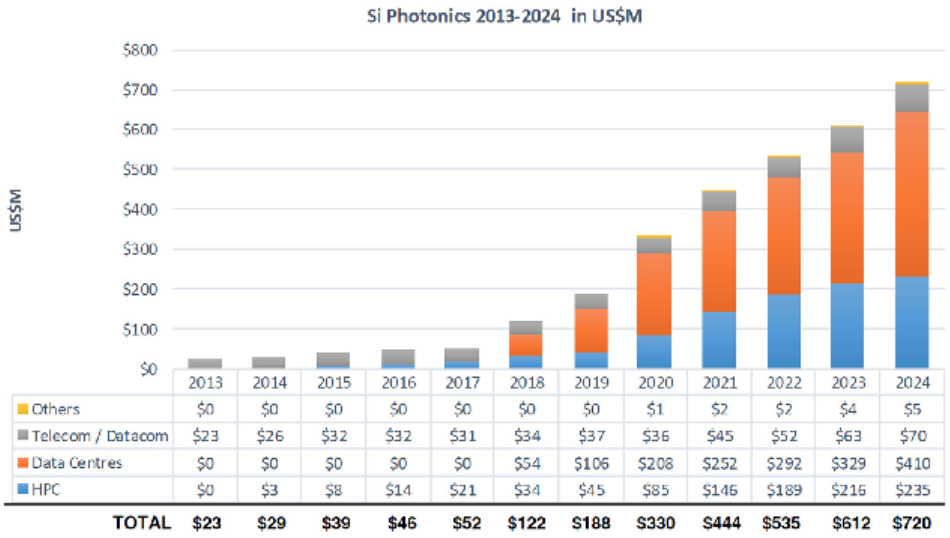
1.6 Silicon photonics

Silicon photonics is a technology that combines silicon-integrated circuits with photonic devices to create optical communication systems. It allows data to be transferred using light, rather than electrical signals,

enabling higher speeds, lower power consumption, and longer distances for data transmission. The main idea lies in integrating photonic devices such as modulators, detectors, optical multiplexers, waveguides, laser diodes, and many more in a single chip. SiP technologies are having massive growth, they exceeded copper cabling capabilities and from 2025 will be more and more used in interconnection for processor chips. At the moment one of the best opportunities for Silicon Photonics lies in the Data Centers. Common semiconductor chips are designed by companies but fabrication is committed to external fabs, especially in China and Taiwan; Silicon photonics is different since it is a fabless technology, the foundry creates and shares under NDA a Process Design Kit (PDK), files used to design the integrated circuit. One of the main advantages of silicon photonics is its potential for low-cost, high-volume manufacturing. It is also compatible with existing CMOS fabrication techniques, which can lead to further cost reductions and improved performance. As said previously, silicon photonics has various applications, including data center interconnects, high-speed computing, telecommunications, and sensing. It has the potential to enable faster and more efficient data transfer, which could lead to advances in fields such as artificial intelligence, big data, and the internet of things. The market forecast has increased in the last years as shown in figure 1.6.

1.6.1 Neuromorphic computing in silicon photonics

In the last decade, the emerging interest in silicon photonics has created new opportunities for new exploration in many areas. In particular, since today, the field of artificial intelligence is sweeping more and more towards human life and research, new ways to improve consumption, speed, and compatibility have been explored. One way is the adaptation of silicon photonics integrated circuits in neuromorphic computing. Photonic devices play an important role since they can modulate an input power, e.g. with optical modulators such as electro-optic modulators, and work as weights for neuromorphic matrices. These circuits are compatible with common CMOS semiconductor technology giving



	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Others	$0	$0	$0	$0	$0	$0	$0	$1	$2	$2	$4	$5
Telecom / Datacom	$23	$26	$32	$32	$31	$34	$37	$36	$45	$52	$63	$70
Data Centres	$0	$0	$0	$0	$0	$54	$106	$208	$252	$292	$329	$410
HPC	$0	$3	$8	$14	$21	$34	$45	$85	$146	$189	$216	$235
TOTAL	**$23**	**$29**	**$39**	**$46**	**$52**	**$122**	**$188**	**$330**	**$444**	**$535**	**$612**	**$720**

Figure 1.6: overview of Silicon photonics market forecast from [6]

the possibility to mix the architectures taking all the advantages in terms of consumption, speed, and easy integration. Therefore, how silicon photonics can be used in neuromorphic computing is through the implementation of optical synapses, which can provide a more efficient and flexible way to implement neural connections. Optical synapses can be implemented using silicon photonics by using light to modulate the strength of the connection between neurons, which can be adjusted in real time based on the needs of the neural network. Overall, the use of silicon photonics in neuromorphic computing has the potential to enable the creation of highly efficient and scalable artificial neural networks that can perform a wide range of tasks, from image and speech recognition to natural language processing and robotics.

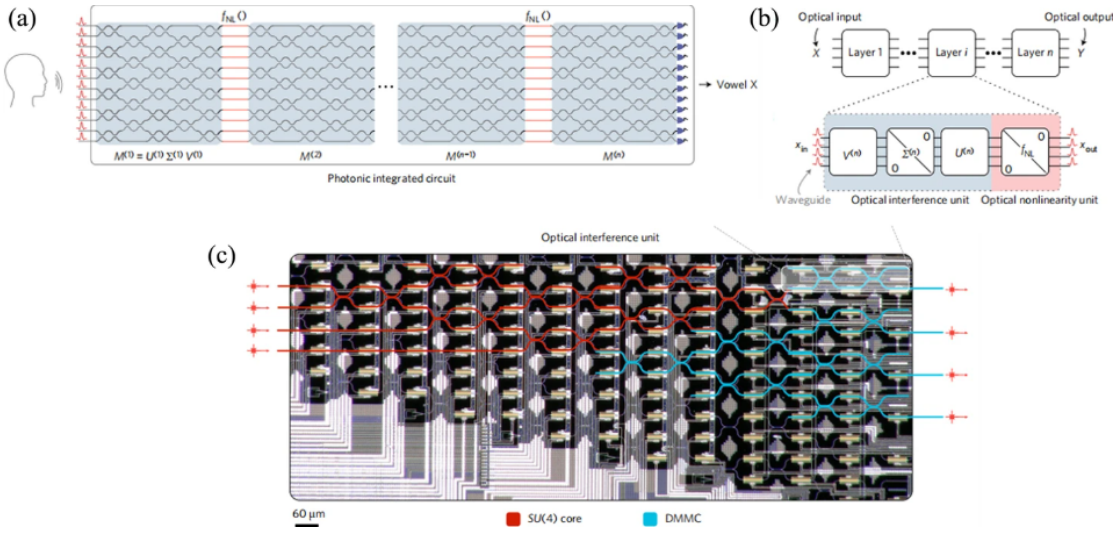


Figure 1.7: Principle of interference for linear operation in neural network [9]. (a) Coherent nanophotonic circuit with multiple matrices. (b) Every layer of the ANN is composed of Optical Interference Unit (OUI) and optical nonlinear unit. (c) The internal structure of OIU

.

Chapter 2

The Circuit

2.1 An overview

The circuit under analysis is a design made for a multicore fiber switch, in detail a 7×7 switch optimized with low-loss grating coupling, designed by a research group at of the Electrical and Photonics Engineering High-Speed Optical Communications Department of the Technical University of Denmark (DTU). In particular, switches are composed by Mach-Zehnder Interferometers (MZIs), and the coupling before and after the MZI is reached with MultiMode Interference (MMI).

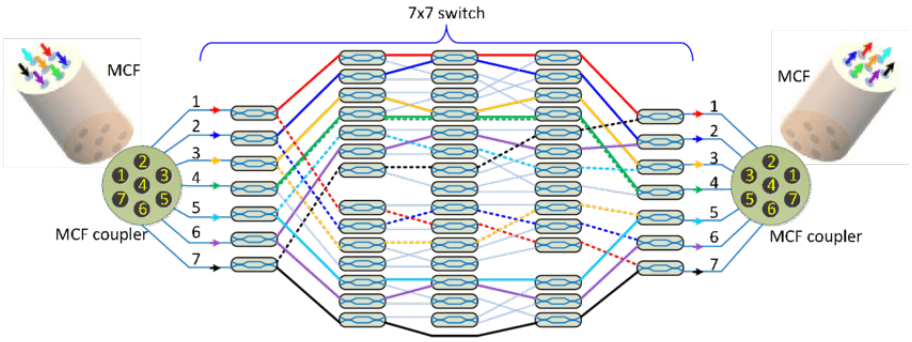


Figure 2.1: Representation of the 7×7 multicore fiber switches circuit from [10].

Even if this structure was designed for a different objective, we can use a small part of it for neuromorphic computing purposes.

The research group at DTU used a small part of this circuit, a 3×3

block, to evaluate how an optical circuit can be optimal for neuromorphic computing even if accurate programming photonic integrated circuits for matrix multiplication remains a complicated task related to the thermal crosstalk into the circuit.[11]

The work they have done begins with the analysis of the subcircuit of the 7×7 MZI switches; they used a 3×3 Mach-Zehnder interferometer matrix that is well suited for matrix multiplication. The linear weights of the matrix are tuned with thermal phase shifts in MZI. The main goal of [11] is to find a model that describes the linear weights as functions of applied voltages. The starting point is the MZI analytic expression in a simple physics-based approach, but results in high modeling errors since thermal crosstalk is not involved in this model and is the limiting factor for scaling up the dimensions of Optical Neural Networks.

The experimental setup is based on an input signal that is an amplified spontaneous emission from 1527 nm to 1568 nm with a maximum 5 dB variation. It is connected to a selector with polarization control, then the optic beam is coupled with a grating coupling into the circuit, then the optical power is reached at the end of each guide with a grating coupling.

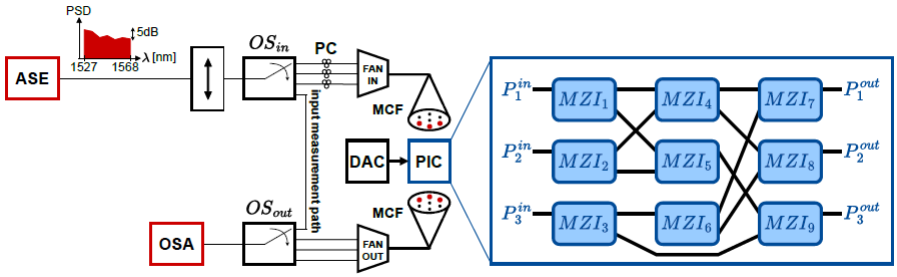


Figure 2.2: Description of the experiment performed on the 3×3 subcircuit [11].

In detail, we received the GDSII file of the entire circuit and the zoom of the circuit used in which we can note the interconnections between MZIs.

In particular, from Fig. 2.4 it is clear how we have the three inputs on the left and how the three outputs are placed on the right.

The experiment was carried out as follows:

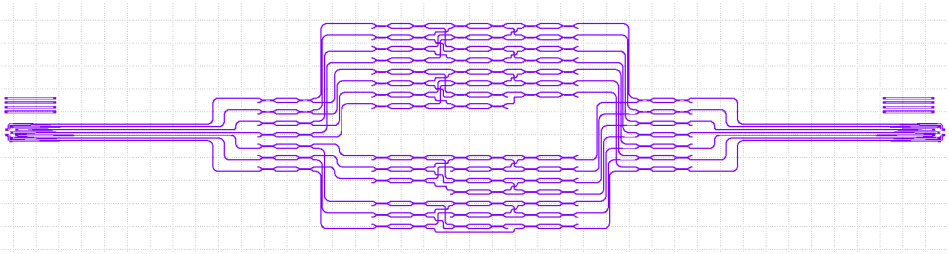


Figure 2.3: Mask of the 7×7 circuit with the grating coupling.

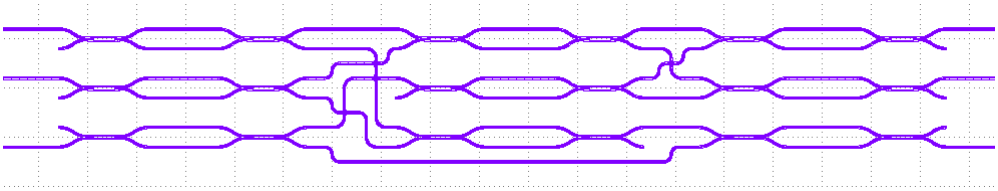


Figure 2.4: Detail of the 3×3 circuit used in the experiment.

1. The optical power is applied at input 1.
2. MZI 1 is activated, it is applied a voltage from 0 V to 2 V with a step of 0.2 V. Here outputs are evaluated.
3. The MZI 1 is turned off and it has waited some seconds for the cooling of the system.
4. Then the procedure is repeated for all the other MZIs one at a time evaluating the output powers.
5. After that, the optical input is changed to input 2, and the experiment is repeated.
6. So on for input 3.
7. Finally, the weights of the circuit can be resumed as the ratio between each output and the input power, as shown in Figs. 2.5-2.13.

In order to understand better the behavior of the circuit that I will analyze in detail, it is important to check and describe the photonic elements that are used such as Mach-Zehnders interferometers, multimode interferometers, grating couplers, and crossing waveguides.

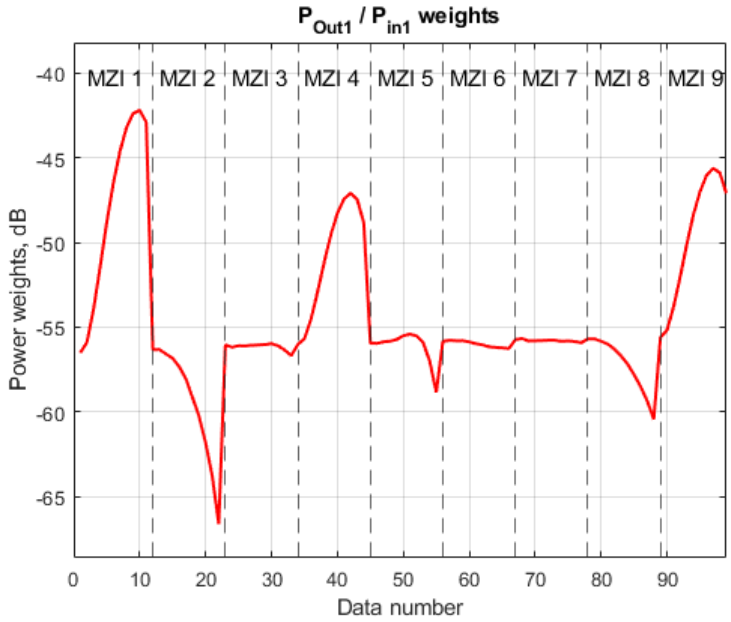


Figure 2.5: Input in port 1, output in port 1, as a function of the MZIs voltages.

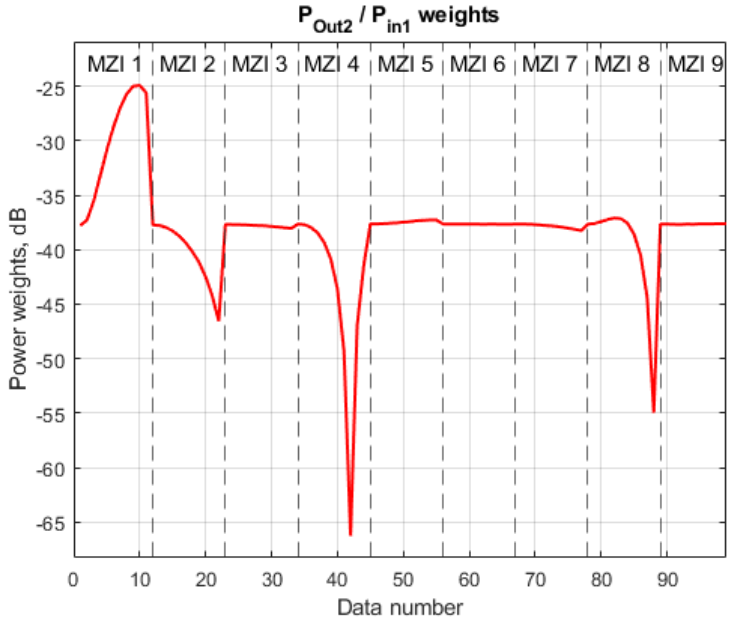


Figure 2.6: Input in port 1, output in port 2, as a function of the MZIs voltages.

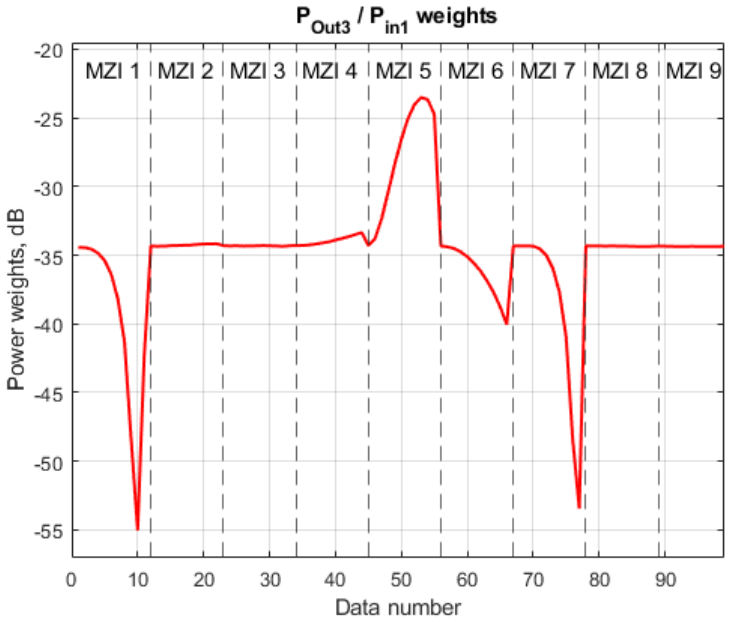


Figure 2.7: Input in port 1, output in port 3, as a function of the MZIs voltages

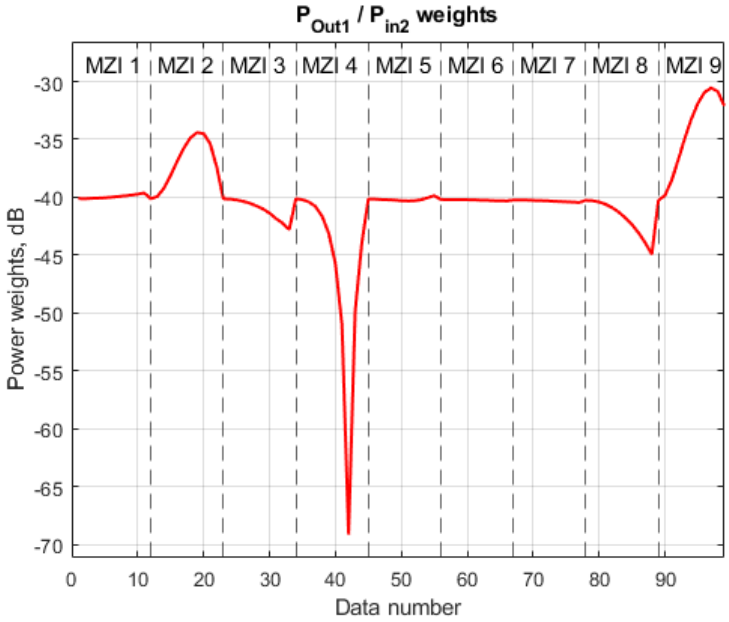


Figure 2.8: Input in port 2, output in port 1, as a function of the MZIs voltages.

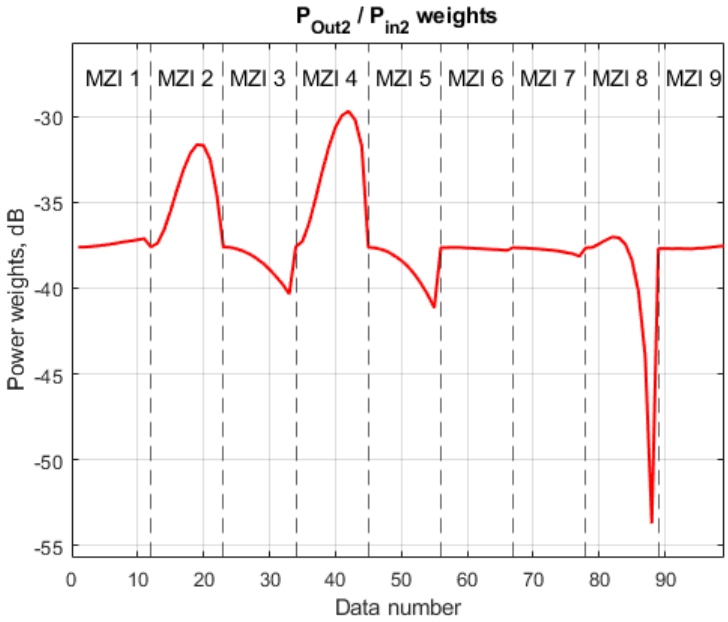


Figure 2.9: Input in port 2, output in port 2, as a function of the MZIs voltages.

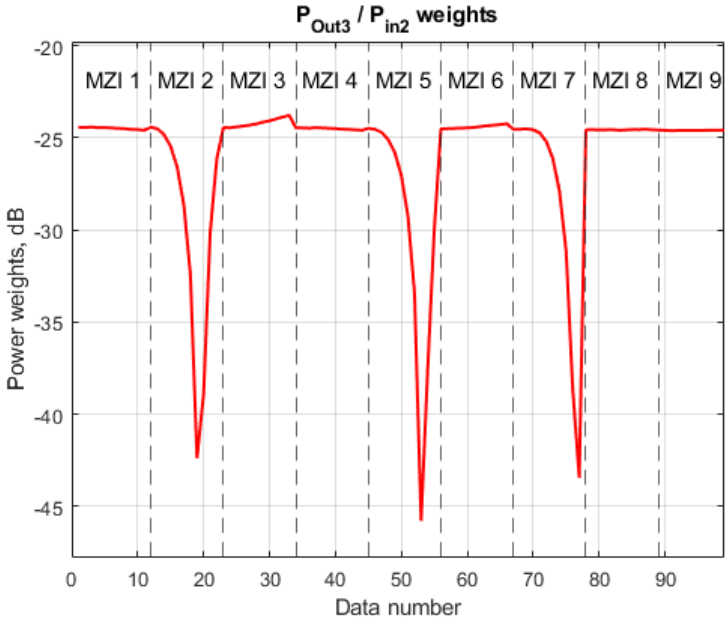


Figure 2.10: Input in port 2, output in port 3, as a function of the MZIs voltages.

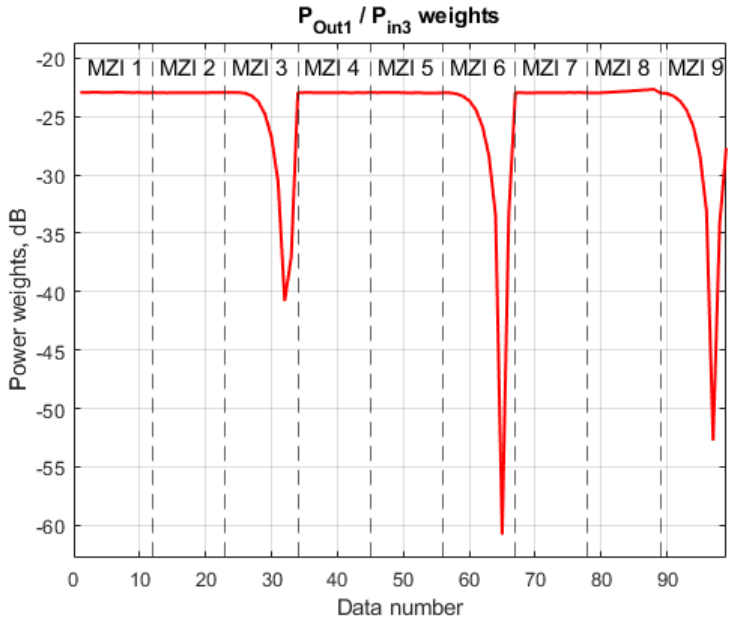


Figure 2.11: Input in port 3, output in port 1, as a function of the MZIs voltages.

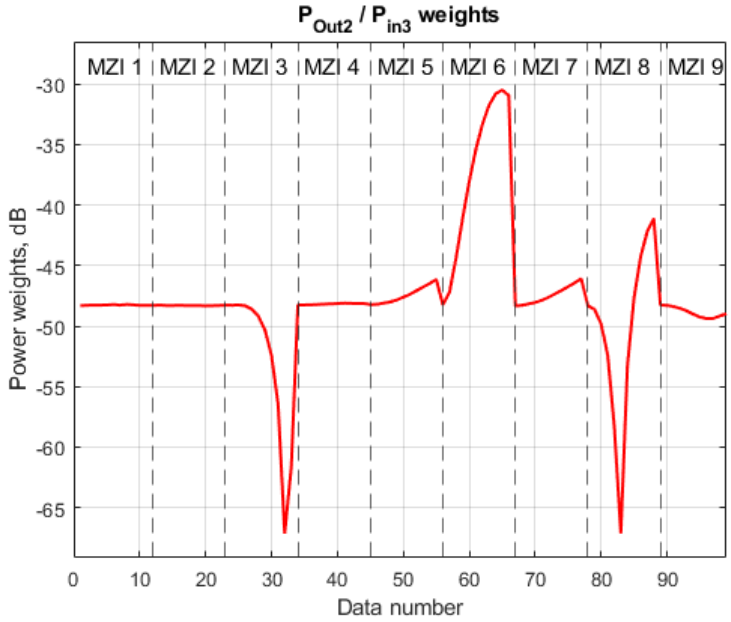


Figure 2.12: Input in port 3, output in port 2, as a function of the MZIs voltages.

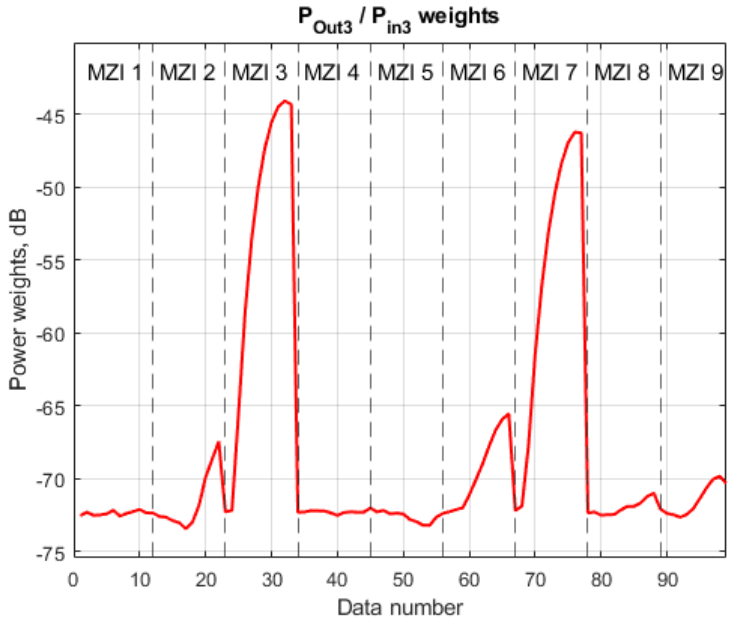


Figure 2.13: Input in port 3, output in port 3, as a function of the MZIs voltages.

2.2 MultiMode Interferometers

A MultiMode Interferometer (MMI) is a largely used component in silicon photonics. It is a passive device that splits and combines the input light compactly and efficiently, with a reduced footprint, and is therefore useful for an integrated circuit.

In an MMI, two or more narrow single transversal mode waveguides are connected to a wider multimode waveguide; the wider region is called the multimode section. In this section, the light propagates through multiple modes, which causes the light to interfere with itself. The interference pattern depends on the input mode and the width of the multimode section.

The MMI device can be designed to split the input light into multiple outputs, or to combine multiple inputs into a single output. The splitting or combining effect is achieved by controlling the width of the multimode section, which determines the phase relationship between the different modes of light.

By selecting the correct length of the multimode section we can

choose how the optical beam is split since the interference that occurs inside is predictable. We talk about self-imaging length (SIL) when the output optical image is the same as the input and the analytical formula is well known:

$$L_{si} = \frac{8 n_{eff} W_{eff}^2}{\lambda_0} \tag{2.1}$$

where n_{eff} is the effective index, W_{eff} is the effective width of the MultiMode section evaluated with the lateral resonance condition and λ_0 is the free space wavelength [13].

Knowing the self-imaging length we can predict where the mirrored image is positioned, at $L_{si}/2$, and where the self- and mirrored imaging occurs, at $L_{si}/4$ and $3L_{si}/4$. (Fig. 2.14)

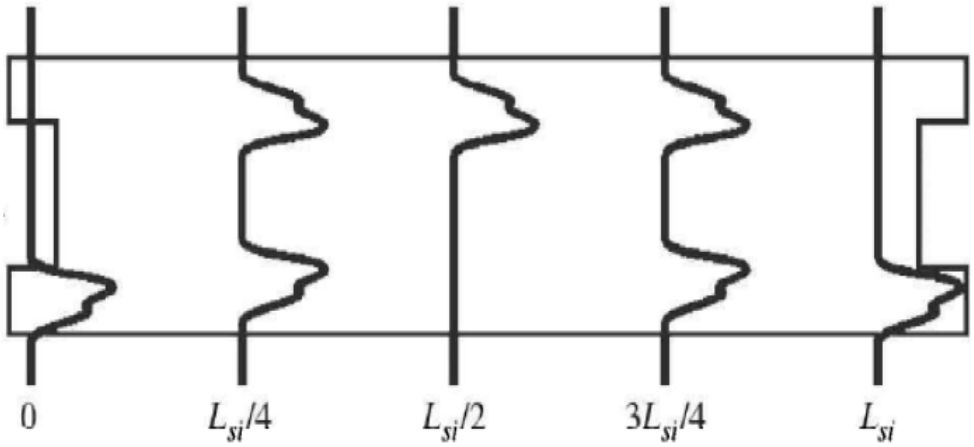


Figure 2.14: Interference imaging prediction into an MMI taken from [13]

More generally the interference mechanisms of different geometries can be resumed in Table 2.1 below.

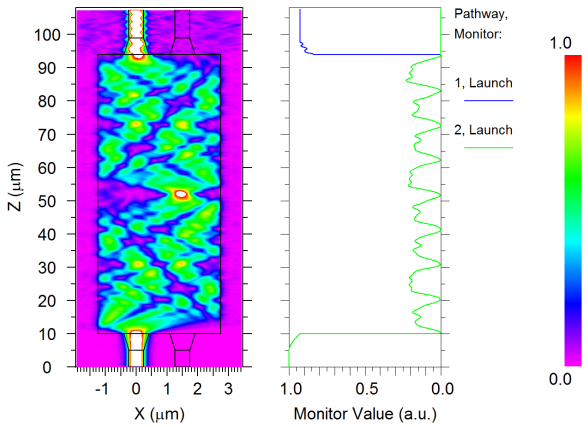
The pattern is easily predictable, MMIs can be integrated with a small footprint, have low inherent losses, large optical bandwidths, and low polarization dependence making this element a key photonic component for example in coupling applications such as in the case of this circuit.

Interference Mechanism	General	Symmetric
Inputs x Outputs	N x N	1 x N
First single image distance	$L_{si}/2$	$L_{si}/8$
First N-fold image distance	$L_{si}/2N$	$L_{si}/8N$
Excitation restrictions	none	no excitation of modes 1,3,5,...
Input position(s)	any	$x = 0$
Output positions	same as inputs	at each d_{eff}/N from the MMI edge

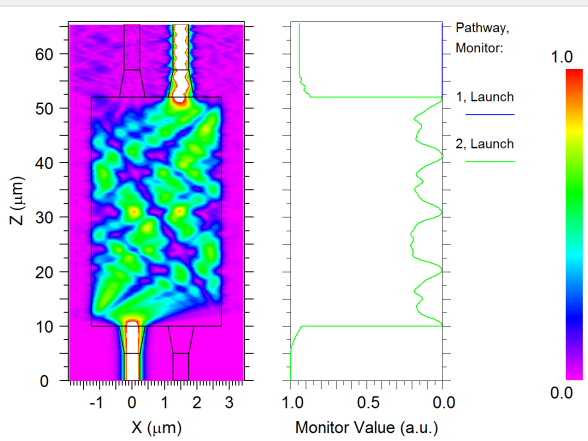
Table 2.1: Properties of the general and restricted (Paired and symmetric) Interference components, Table taken from [13].

2.2.1 MMI RSoft analysis

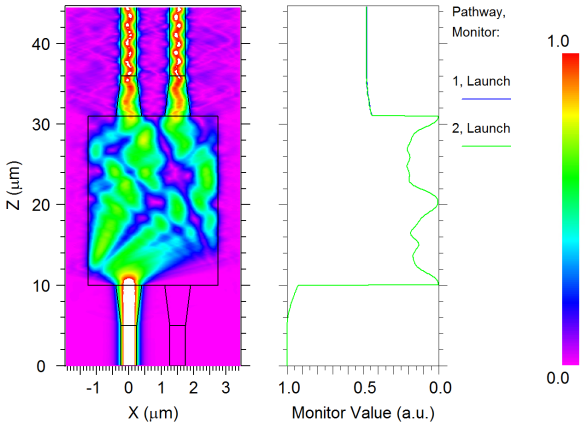
In RSoft, I explored the MMI working principle, evaluating three MMIs with the same sizes but different lengths. In Fig. 2.15a-c, the theoretical behavior is verified, we inject a field into one of the two input waveguides for the considered device, the self-imaging length L_{si} is 84 µm: The field profile observed at this length is a copy of the input. In Fig. 2.15a-c, the map on the left plots the modulus of the field $E(x, z)$, while the overlap integral $\int |E|^2 \, \mathrm{d}x / \int_{-\infty}^{+\infty} |E|^2 \, \mathrm{d}x$ over the waveguide or MMI width W is shown on the right. At $L_{si}/2$ the field is in the cross condition. Finally, for a device with length, $L_{si}/4$ the input power is equally split on the two output ports. For this analysis, I used an Input field wavelength equal to 1.55 µm, a unitary input power, on the left waveguide as in Fig. 2.16.



(a) RSoft 2D map (left) and the integral (right) of the electric field in the MMI with length, $L_{si} = 84\,\mu m$.



(b) RSoft 2D map (left) and the integral (right) of the electric field in the MMI with length, $L_{si}/2 = 42\,\mu m$.



(c) RSoft 2D map (left) and the integral (right) of the electric field in the MMI with length, $L_{si}/4 = 21\,\mu m$.

Figure 2.15: RSoft analysis of MMIs at different lengths.

For what concerns the circuit used in the experiment, the MMI has a length of 43.5 µm, a width of 6 µm, waveguides are 0.5 µm width with a cone injection into the larger part that has a length of 10 µm and the final part 1.5 µm width; a similar approach is used at the output.

In RSoft I started with a Beam Propagation Method (BPM) based analysis since the device is much longer than wider. The input field is shown in Fig. 2.16, where the area is centered into the left waveguide before entering into the MMI. The wavelength used is a standard 1.55 µm even if there is a slight variation of the results in the wavelength range used in the experiment (1527 nm-1568 nm). However, in the end, the power partition in the entire circuit will not suffer from this simplification, as shown in Fig. 2.17. As we can see in Fig. 2.18, a single MMI splits equally the input power. In the experiment one of the two outputs of this first MMI is phase shifted by a heater, finally, the two beams are reassembled with an identical MMI. If we consider the entire structure with two MMIs, we are in cross-state natively if no phase shift is applied.

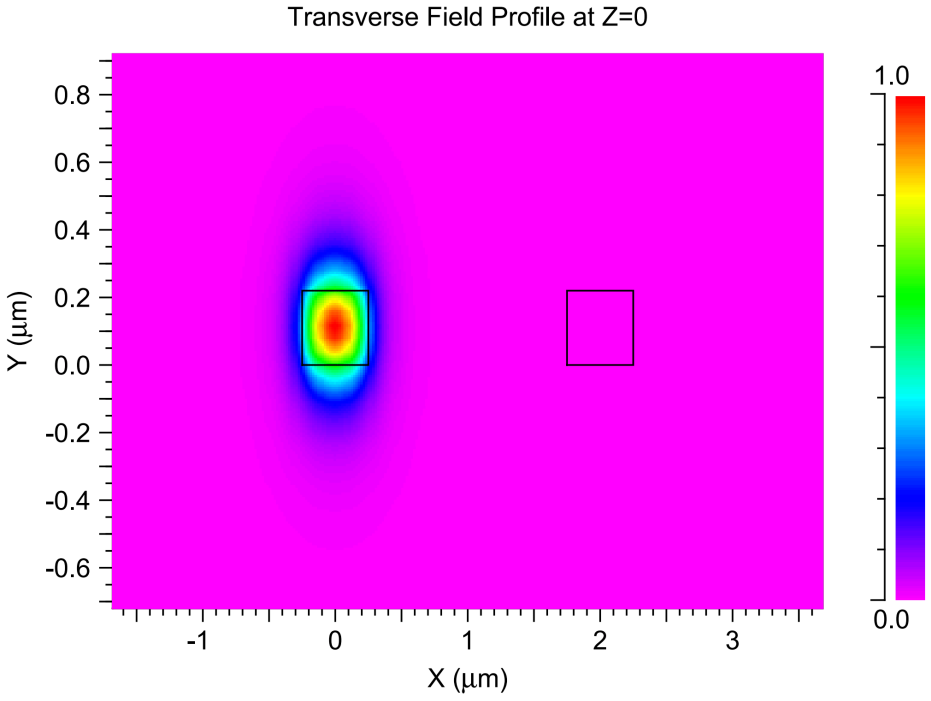


Figure 2.16: Map of RSoft input field.

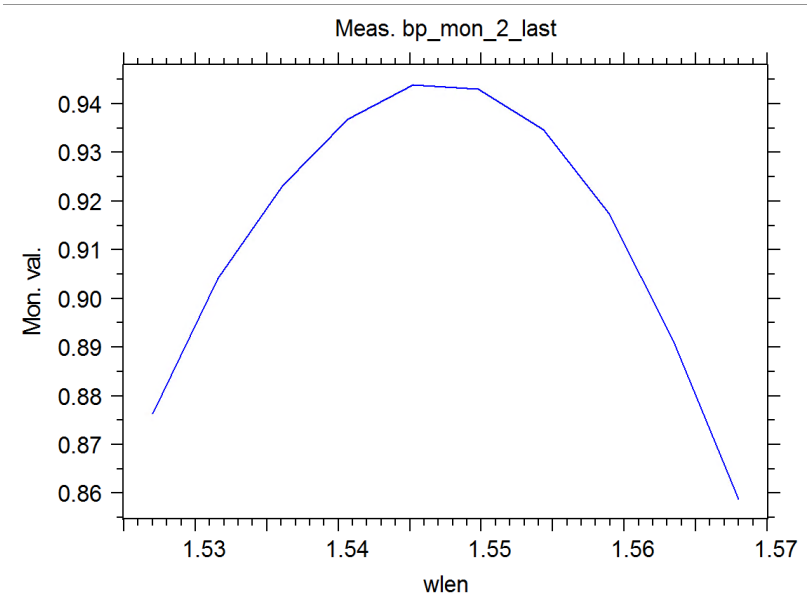


Figure 2.17: Output power of one of the two waveguides by varying the input field wavelength.

2.3 Mach-Zehnder Interferometer

The optical modulator is the key point of neuromorphic processing in silicon photonics: the optical input of these modulators can be modified in phase, amplitude and polarization through electro-optics, thermo-optics, electro-absorption or magneto-optic effects. Research on these solutions in integrated optical systems is growing rapidly, since they are suitable for low power consumption, high speed, and large modulation depth requests[14]. In particular electro-optic modulators are one of the mainly used in optical neural networks where the amplitude modulation is reached through the interference of two phase-modulated beams such as in Mach-Zehnder, arms of the directional coupler, or resonance shift of ring resonators. The physical mechanism behind this is modulation of the refractive index of the silicon layer with the application of an electrical field or carrier injection. Focusing on MZI, the considered parameters are the ratio between *output*/*input* light intensity based on input voltage applied, The V_π that is the voltage needed to have a complete disruptive interference in which the two beams reach a phase

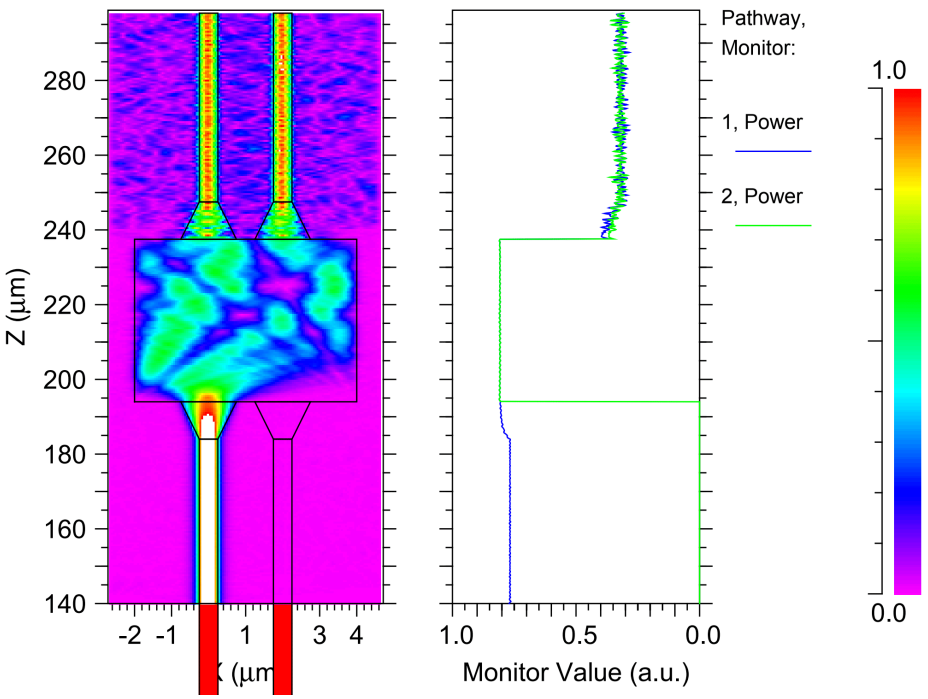


Figure 2.18: RSoft 2D map (left) and the integral (right) of the electric field of the MMI used in the experiment.

difference of 180 °C, then the extinction ratio, a coefficient related to the ratio between the ON state output power and the OFF state output power. Electro-optic modulators are also good performers because they have a high extinction ratio, good frequency response, larger optical bandwidth, and large ON-OFF voltage. They can be integrated into circuits, using silicon-based Mach-Zehnder interferometers. The physics behind is based on the thermal modulation of optical properties such as refractive index. The temperature modulation is driven by a resistive heater, in which is applied an electrical signal, in fact, there is a temperature dependence on the refractive index coefficient that can be resumed in the following formula:

$$n(T) \approx n(T_0) + \frac{\mathrm{d}dn}{\mathrm{d}dT}\bigg|_{T_0} (T - T_0) \tag{2.2}$$

Therefore, an approximated linear temperature dependence can be used to modulate the light intensity since the difference in refractive

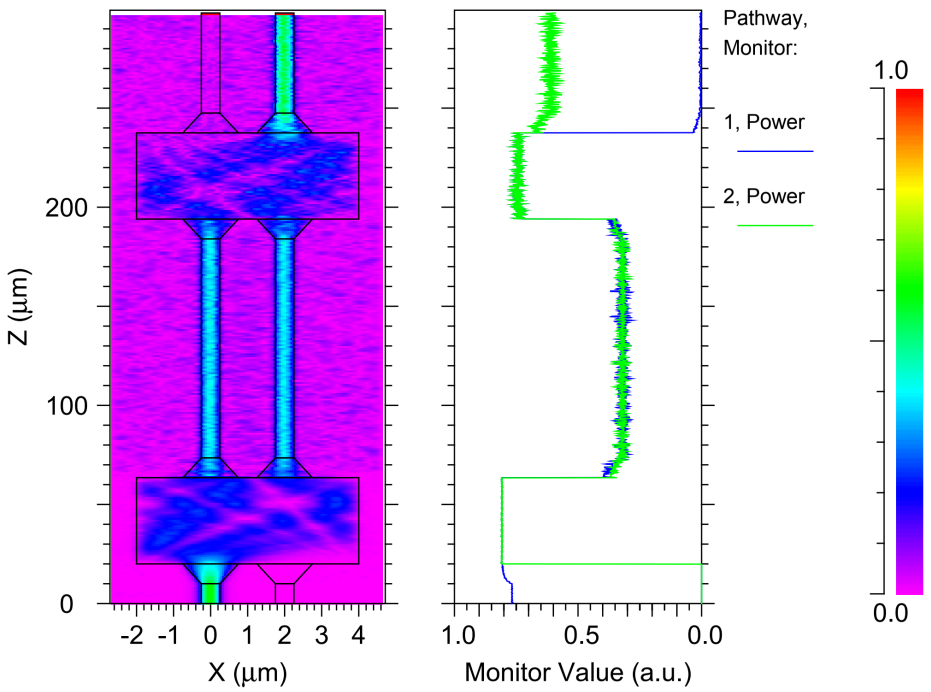


Figure 2.19: RSoft 2D map (left) and the integral (right) of the electric field of two MMIs used in the experiment.

index causes a shift in the phase of the light inside the material. In Mach-Zehnder interferometers the light beam is split into two waveguides by a Y-splitter or a MMI, then the structure can be symmetric or asymmetric depending on the fact that waveguides are heated by two different heaters or just one waveguide is phase-shifted by a heater. Then the light beams are reassembled by a Y-combiner or an MMI, and the difference in phase causes a modulation in output due to the interference.

2.3.1 MZI RSoft analysis

In the circuit analyzed, the Mach-Zehnder interferometers are built asymmetrically with a single heater onto one waveguide, introducing a phase shift just for the light beam propagating in that silicon guide, ignoring for the moment the thermal crosstalk. To complete the RSoft study on the behavior of the MZI, I considered half the structure of the

complete block, Fig. 2.21, giving two identical input powers, Fig. 2.20, and giving to one of them a phase variation.

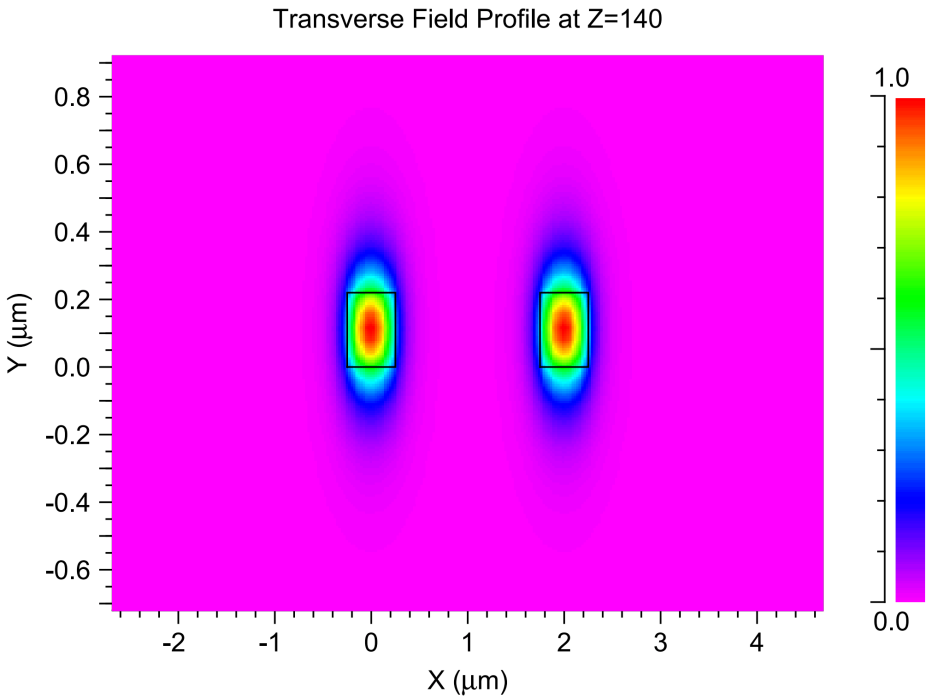


Figure 2.20: Map of the RSoft double input field.

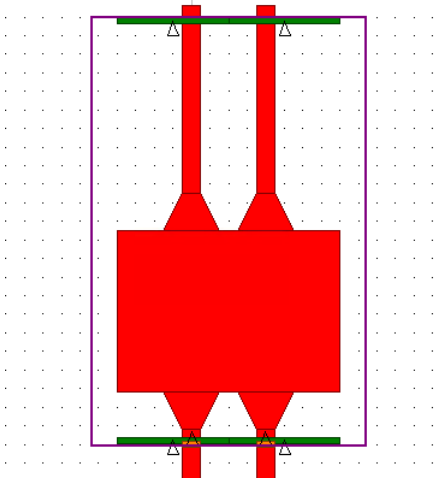


Figure 2.21: RSoft structure to simulate the behavior of MZI.

By scanning over a variation of the phase of the second input power

from 0 °C to 400 °C, I saved the output monitors after the MMI, and the result is a sinusoidal variation of the output field in the two outputs, as shown in the graph 2.22 where it is compared to classic $sin(x)$ and $-sin(x)$. Also at $\pi/2$ in our case, power goes out from the top-left output.

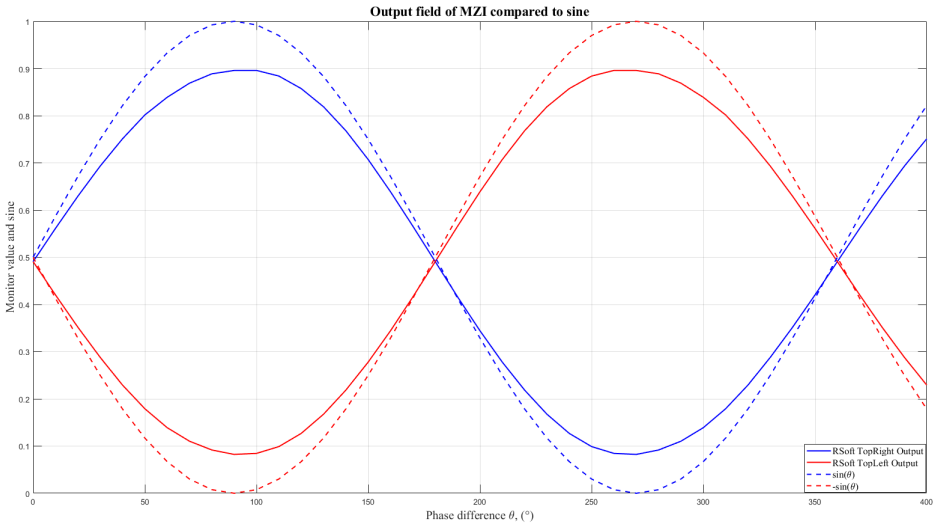


Figure 2.22: Comparison between the results from RSoft scan of output fields and sine behavior.

2.4 Cross-connection

In a small chip, it is always important to minimize the area used and reduce the length of circuit lines, and here the cross-connections play an important role. In optical circuits, the silicon-based cross-connections are on the same plain level, so the most important thing is to reduce the optical crosstalk between them in order to limit the losses in the travel. An ingenious solution is found in [15], and is reused in the circuit in the analysis where they use an MMI-based cross connection, and this type of filter reduces crosstalk, acting better than a conventional plain crossing.

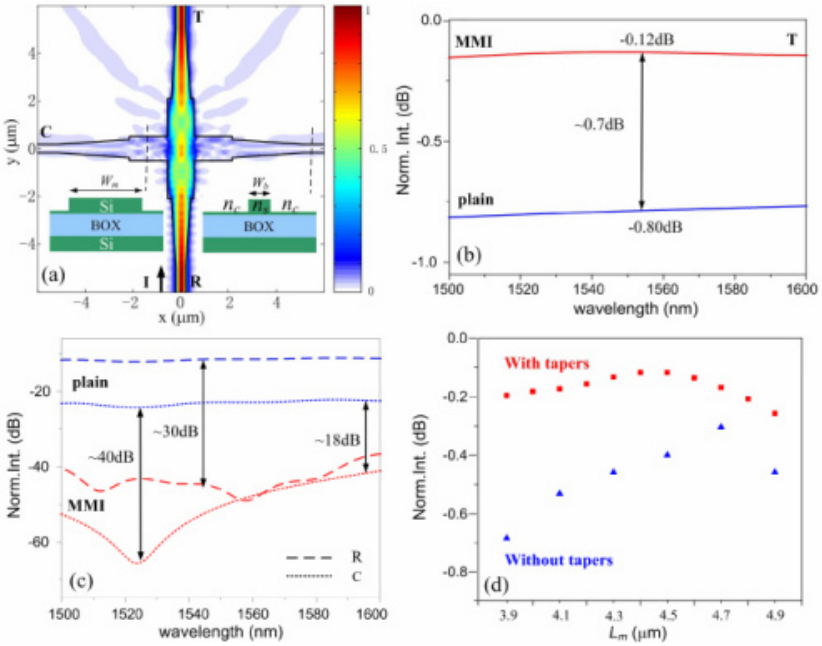


Figure 2.23: Simulation of the MMI cross-connection taken from [15]. (a) BPM simulation of the MMI crossing at wavelength 1.55 µm. (b) Transmission spectra of the MMI crossing (red line) and of the conventional waveguide crossing (blue line). (c) Cross-port transmission spectra and back-reflection-port spectra of the MMI crossing (red dotted line and red dashed line) and of the plain crossing (blue dotted line and blue dashed line). (d) Throughput-port transmission of the MMI crossing with tapers (square dots) and without tapers (triangular dots) as a function of MMI length (L_m).

2.4.1 Cross-connection RSoft analysis

I have done a similar simulation for the MMI cross-connection in RSoft as shown in Fig. 2.24. An interference pattern occurs in the MMI and this limits the crosstalk with the horizontal line. Sizes are equal for both lines since the cross has to perform in both cases, the optical beam on the vertical line and the optical beam on the horizontal line.

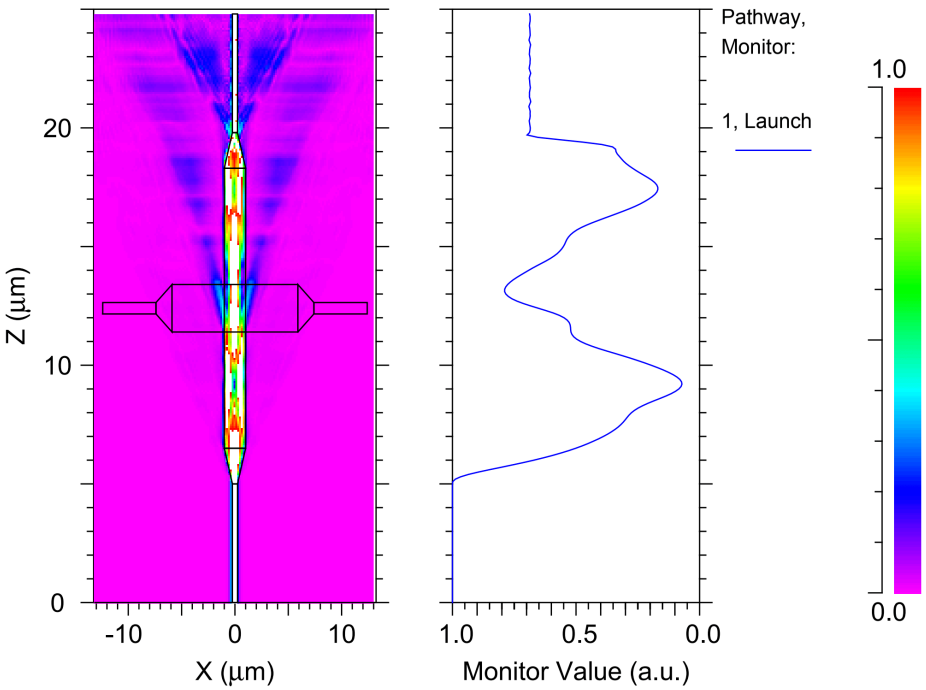


Figure 2.24: RSoft 2D map (left) and the integral (right) of the electric field of the cross-connection used in the experiment.

Chapter 3

Thermal Analysis

3.1 The effect of temperature

Mach-Zehnder interferometers, work with tuning, using a Thermo-Optic Phase Shifter (TOPS) that plays the main role in the light modulation process. As said, silicon photonics is the solution for a photonic integrated circuit for implementation in CMOS electronics and the PIC is based on optical effects through phase shifters with thermo-optics effect. The physics behind is related to an effective index variation due to thermal shift, generally reached thanks to a resistive heater. In general, silicon thermal properties are well known and the thermo-optic coefficient is $1.8 \times 10^{-4}\,\mathrm{K}^{-1}$.Thermo-optic phase shifters are widely chosen for integrated PIC due to low cost, simple design, small footprint, and easy fabrication in SOI[17]. TOPS has a simple structure, a silicon waveguide, and a resistive heater where the tuning voltage is applied. It can have different structures; probably the easiest one for fabrication is the structure with the waveguide below a single heater, separated by an oxide layer. This structure is also used in the circuit analyzed. Different cross-section TOPS used in photonic circuits are reported in Fig. 3.1: Fig.3.1(a) represents TOPS with a heater on top of the waveguide that is our experimental case, 3.1(b) is a TOPS with strip waveguide with doped silicon on both sides as heaters, 3.1(c) is a TOPS with rib waveguide with doped silicon on both sides as heaters, 3.1(d) is a TOPS with rib waveguide with doped-silicon heaters directly integrated, 3.1(e) is a TOPS with strip waveguide with hybrid heaters, 3.1(f) is a TOPS

with an air-gap trench and an undercut on the silicon substrate, 3.1(g) is a TOPS with folded waveguide and heater on top, 3.1(h) is a TOPS with multi-pass waveguides with the heater on top[17].

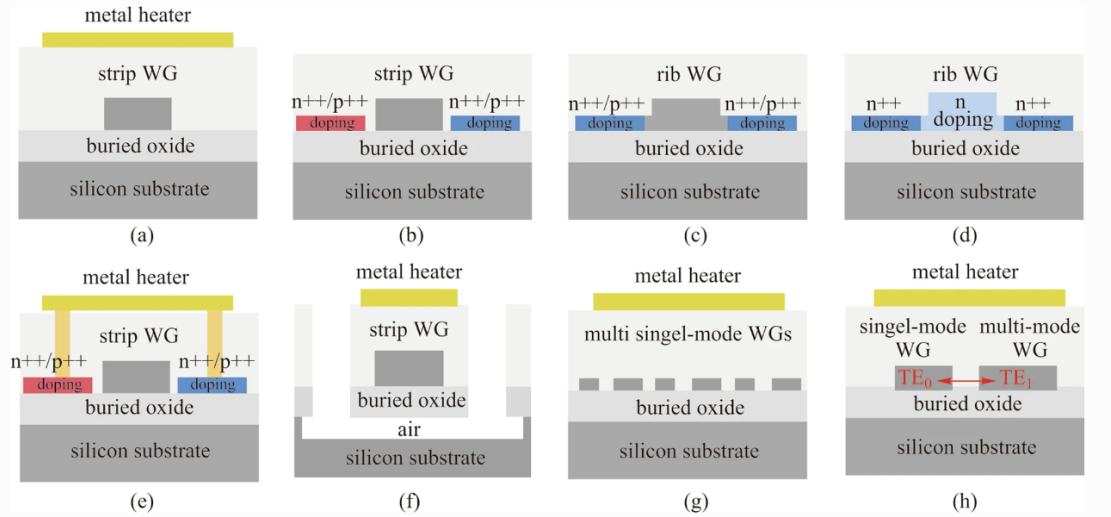


Figure 3.1: Different TOPS structures used in photonic integrated circuits taken from [17].

Understanding the working principle of the circuit is a must for thermal analysis. In particular, the central issue of a photonic integrated circuit that limits the area reduction is the thermal crosstalk that occurs between different MZI. Heat diffusion is the problem and often creates undesirable thermo-optic-based phase shifts on other waveguides. Therefore, it is important to study heat propagation to design the correct distances between waveguides. In the experiment circuit, thermal crosstalk is present and leads to errors in model prediction. Improving the model needs a readjustment that takes into account also thermal crosstalk.

3.2 Thermal simulation

Thermal analysis is performed using multi-physical simulation with the COMSOL software, the structure is a BOX with dimensions 1000 µm long, 1200 µm wide, and 1000 µm depth, Fig.3.2. As I said, the structure is based on TOPS with a resistive heater on top of the waveguide with

an oxide layer in between. In the 3×3 circuit, the distances between the three MZIs pack in a column and the following packs of three MZIs are enough to ignore the thermal crosstalk through them, so the analysis is done taking just three MZIs. Another simplification is done, and I considered only the waveguides inside the MZ, where the light beam is split and where the thermal shift is made. In the end, we have six waveguides and three of them have heaters on top. Distances are important and, in this case, the distance between the waveguide with heater and the waveguide without it is 40 µm and they represent the split waveguides of the same MZI, instead, the distance between two Mach-Zehnders is 60 µm.

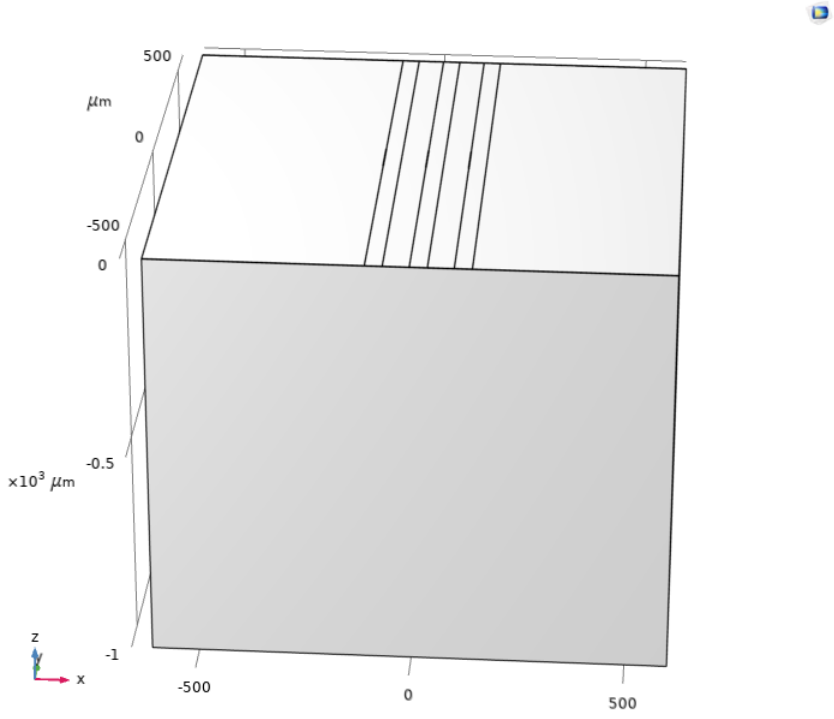


Figure 3.2: Box of COMSOL simulation for thermal analysis, it consists of 6 waveguides, and 3 resistive heaters (not visible in this scale).

As I said, the structure is based on TOPS with a resistive heater on top of the waveguide with an oxide layer in between. The heater is made of Titanium and has a height of 0.1 µm, a width of 1.8 µm and a length of 100 µm. The distance between the heater and the waveguide below

is 1 µm filled with silicon oxide. Then the waveguide has a section area of 0.5 µm× 0.25 µm, Fig.3.4. There's also an Aluminum mirror under the silicon oxide used for the grating coupling but for thermal purposes can be neglected, Fig.3.3.

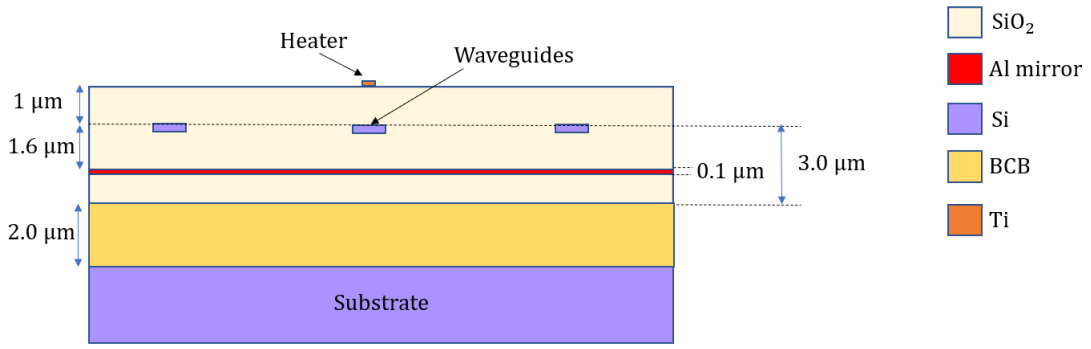


Figure 3.3: Layered structure

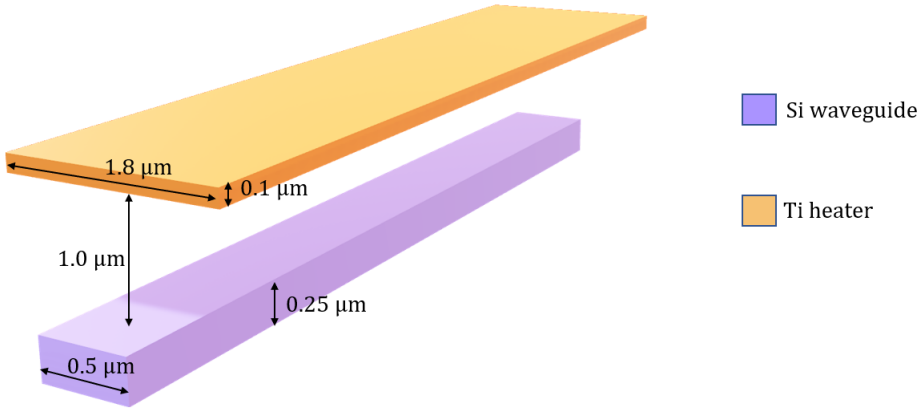


Figure 3.4: TOPS structure

In this COMSOL simulation the physics investigated is the Joule effect based on the heat generated from the current through the resistive metal heater, and the heat diffusion and propagation into the specific box structure. In a simulation, the key point is to set the correct boundary condition to have the right behavior of the model simulated. For what concerns the electric current, in COMSOL the current conservation related to the conduction model $J = \sigma E$ has to be set, where J is

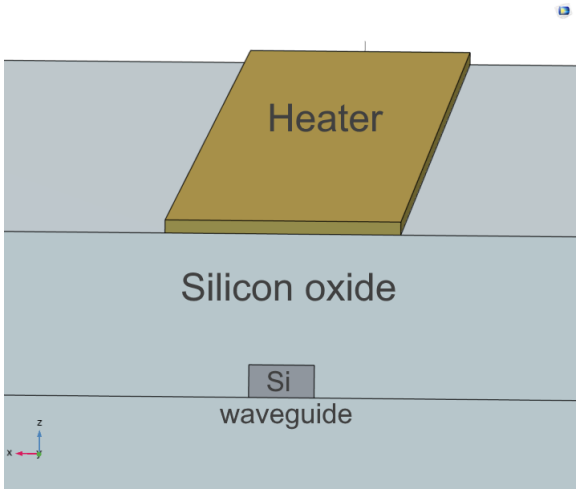


Figure 3.5: COMSOL 3D CAD of the Thermo-optic phase shifter

the current density, σ is the electrical conductivity and E is the electric field derived from the applied voltage. Also, electrical insulation is set to neglect leaks, and, then, ground and electric potential are added into the model to have the right current flow. The tricky part is evaluating the heat transfer in a solid. It starts by defining the ambient temperature, 293.15 K, as a reference; then all thermal coefficients such as thermal conductivity and heat capacity at constant pressure are taken from the definition of the materials; finally, the boundary conditions must be chosen carefully. At the lateral sides surfaces of the BOX, I set a Neumann boundary condition where the general inward heat flux is zero; the same thing applies to the lateral sides of the heater. For the bottom surface of the BOX, I set the ambient temperature as a reference to reach that surface with a Dirichlet boundary condition. Finally, the top surface of the BOX where the heat is generated needs specific conditions to be set to work correctly; I considered a convective heat flux onto the top surface that is a horizontal plane, the fluid is air, and the pressure, since this is a fanless system, is 1 atm with an external natural convection from upside and the external temperature at 293.15 K. After applying the boundary conditions, multiphysic simulations must be set with electromagnetic heating that links the electric currents with the heat transfer in the solids. I have performed different simulations under different conditions, applying 2 V, (see Fig.3.6) singularly at the

heaters and considering all possible combinations.

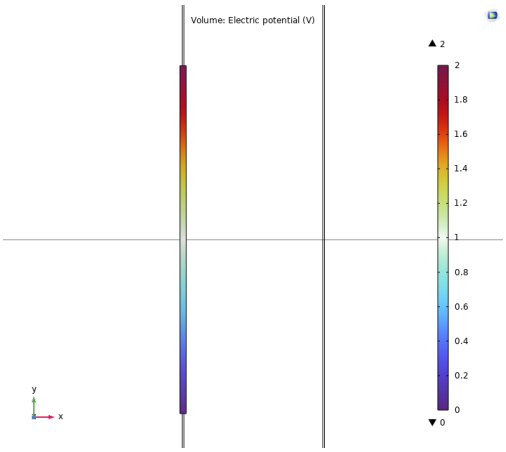


Figure 3.6: COMSOL simulation of the voltage applied on the heater

To perform the thermal analysis, I created three parameters, one for each heater input voltage, and, selectively, I applied 2 V in input in order to calculatethe entire thermal map of the box. In particular, the area of interest is the surface where the waveguides lie, so I took a slice of the box at the height of the waveguides and then plotted the thermal surface related to it. In Fig.3.7, the voltage is applied to the heater of the first Mach-Zehnder interferometer, the heat spreads out in a bubble shape and silicon oxide helps to extinguish it, the maximum temperature is 70.7 °C at the center of the waveguide of on top the active resistive heater. Thermal crosstalk is evident, in particular in the waveguide of the same MZ, leading to a minor phase difference between the two arms of the Mach-Zehnder; also the other MZIs are affected by weak thermal crosstalk. In Fig.3.9, the active heater is on the middle MZI, and the maximum temperature, as expected, is the same as in the previous case: what is different is the thermal crosstalk, since the thermal source is in the middle of the structure, it affects more the other MZIs, especially the one on the left. In Fig.3.11, the only active heater is the one on the right of the third Mach-Zehnder, again the maximum temperature is around 70.5 °C as in the previous analysis, and this time the thermal crosstalk is present also in the middle waveguides. These temperature analyses are specifically suited for the experiment that is done with a single voltage input ON at the time,

but for completeness and other applications, such as network training, a full spectrum of all possibilities with more heaters active at the time is needed. I began evaluating the case where the heater on the left and the one in the middle were active, Fig.3.13, the crosstalk arises significantly, the maximum temperature increases to 78.6 °C due to the fact that heat diffusion is less effective with two heaters in ON-state, thermal crosstalk highly affects the waveguide in the middle of the two heaters leading to a less effective phase shift in the first Mach-Zehnder. The case of MZI 1 and MZI 3 in ON-state is analyzed in Fig.3.15, here the maximum temperature is slightly lower since the two heaters ON are further away than in the previous case and the heat diffusion and extinction are more effective at 76.2 °C. Then I switched on the case with the middle and the right heaters actives, Fig.3.17, similar to the one in Fig.3.13, the maximum temperature is similar at 79.0 °C and in this situation the phase shift of the MZI 2 will be less effective. In the end, I consider the case where all the resistive heaters are in ON state, shown in figure 3.19, here the maximum temperature arises and goes to 85.5 °C, leading to massive thermal crosstalk on all waveguides, and this reduces the phase shift of MZIs.

MZI 1	MZI 2	MZI 3	T_{MAX}
ON	OFF	OFF	70.7 °C
OFF	ON	OFF	70.8 °C
OFF	OFF	ON	70.5 °C
ON	ON	OFF	78.6 °C
ON	OFF	ON	76.2 °C
OFF	ON	ON	79.0 °C
ON	ON	ON	85.5 °C

Table 3.1: Maximum temperatures reached with every combination.

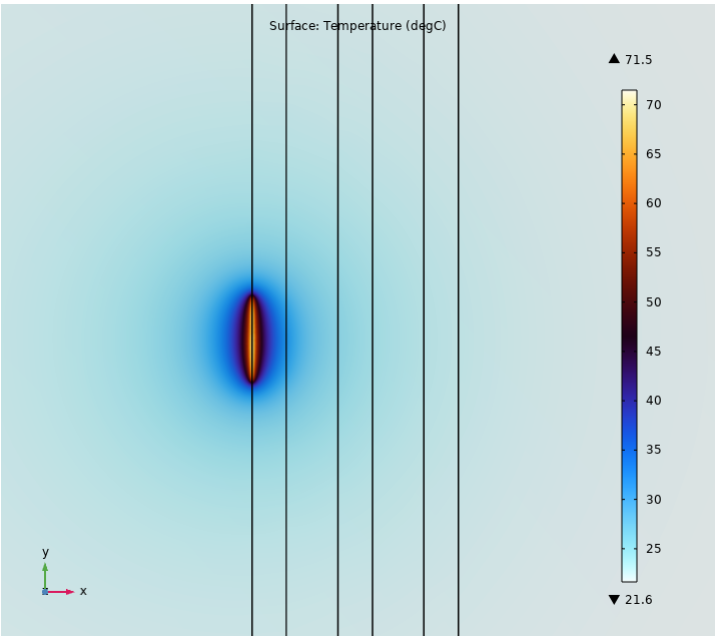


Figure 3.7: Thermal surface in COMSOL at waveguides height with the heater of MZI 1 active.

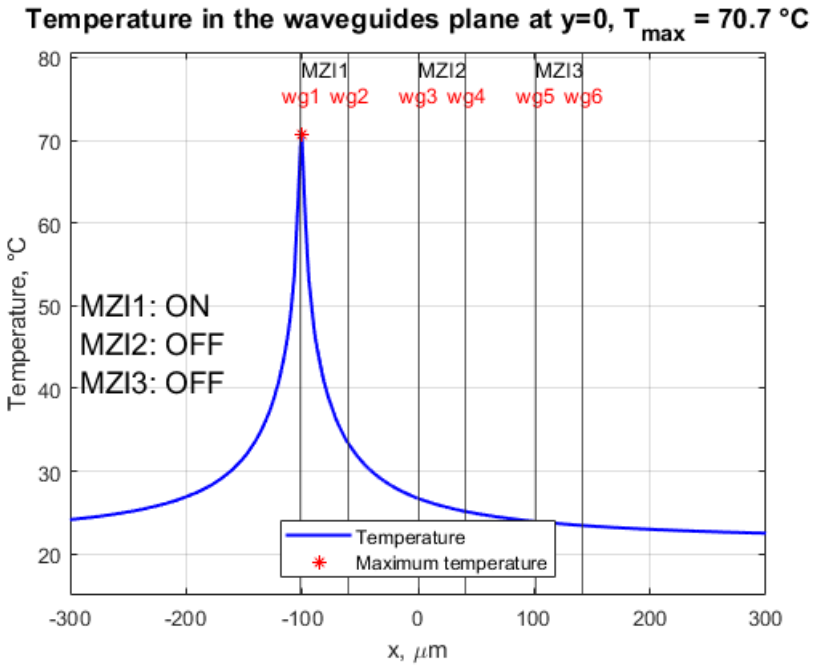


Figure 3.8: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 1 active.

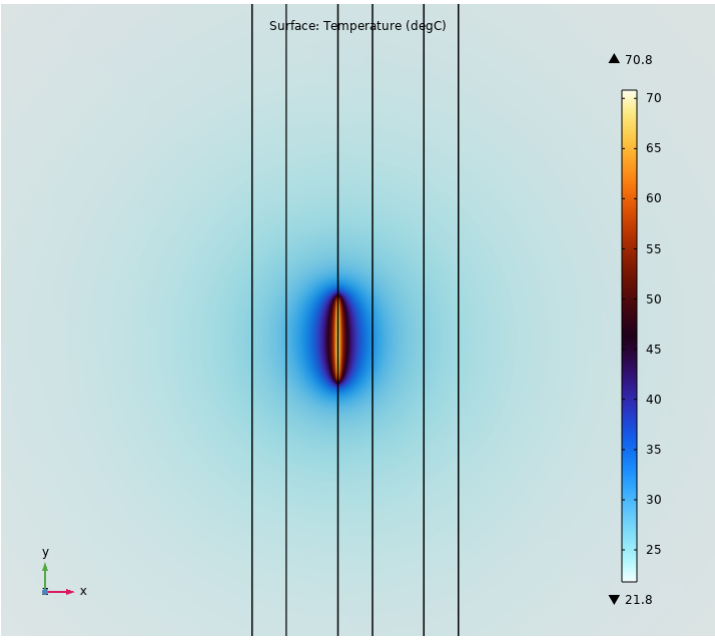


Figure 3.9: Thermal surface in COMSOL at waveguides height with the heater of MZI 2 active.

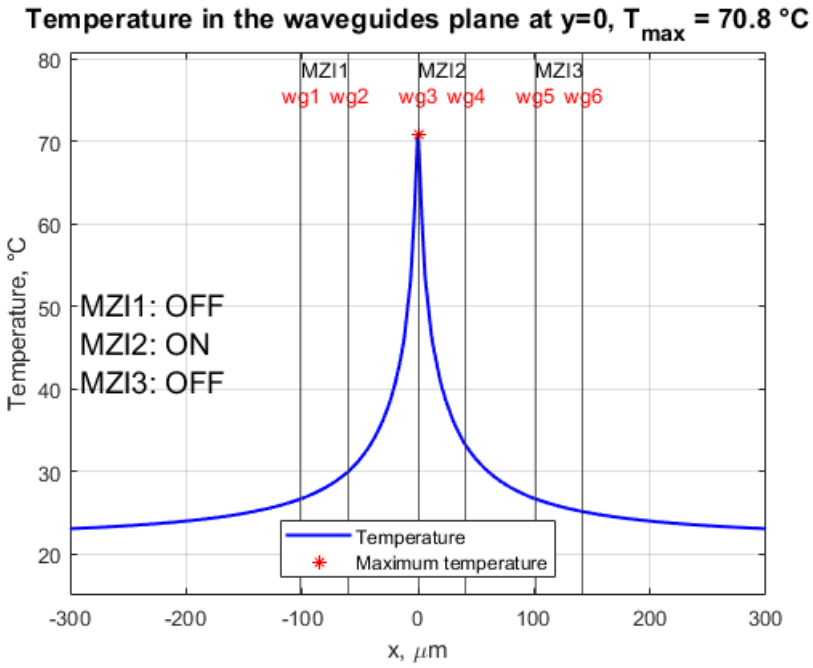


Figure 3.10: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 2 active.

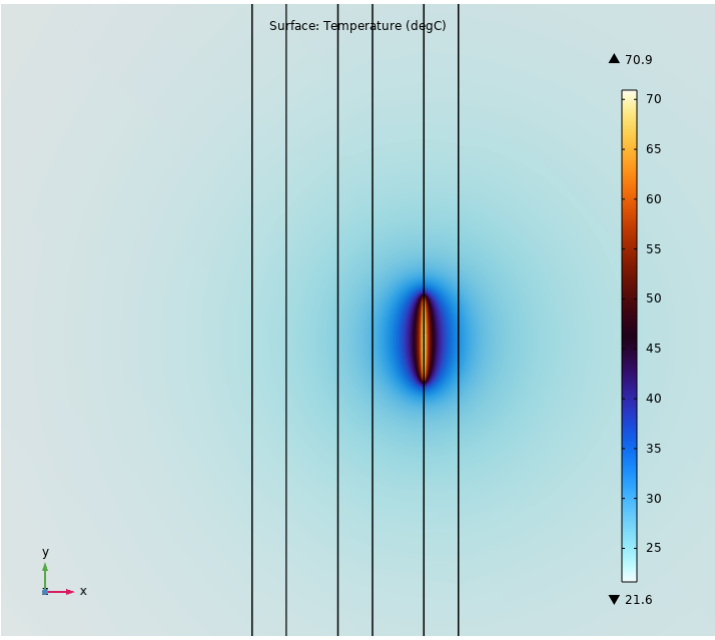


Figure 3.11: Thermal surface in COMSOL at waveguides height with the heater of MZI 3 active.

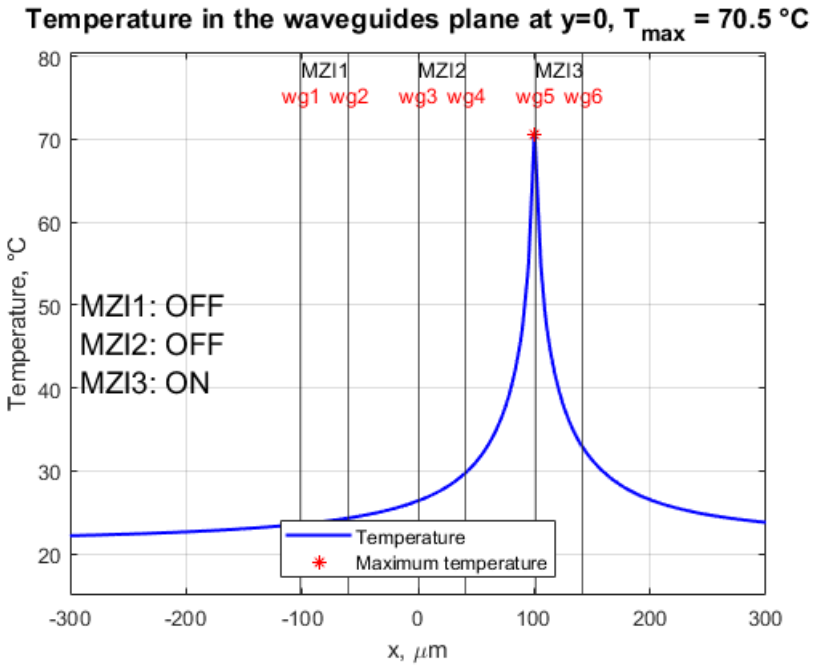


Figure 3.12: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 3 active.

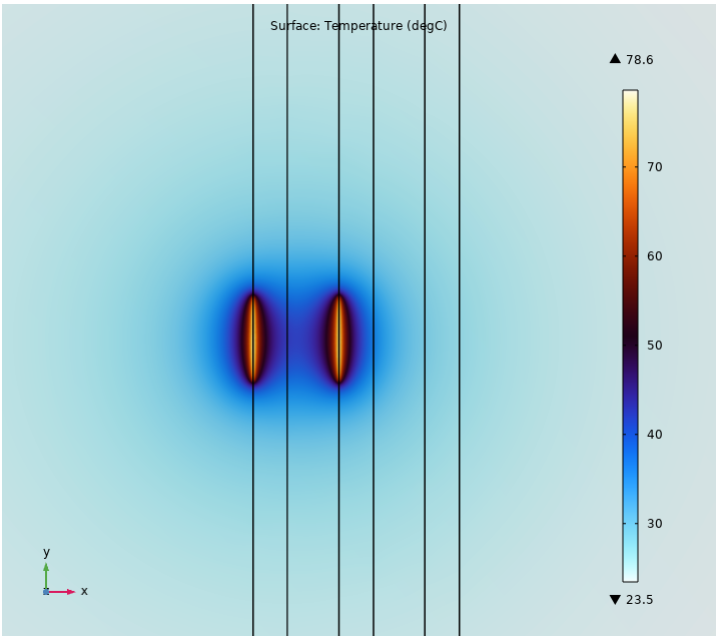


Figure 3.13: Thermal surface in COMSOL at waveguides height with the heaters of MZI 1 and MZI 2 active.

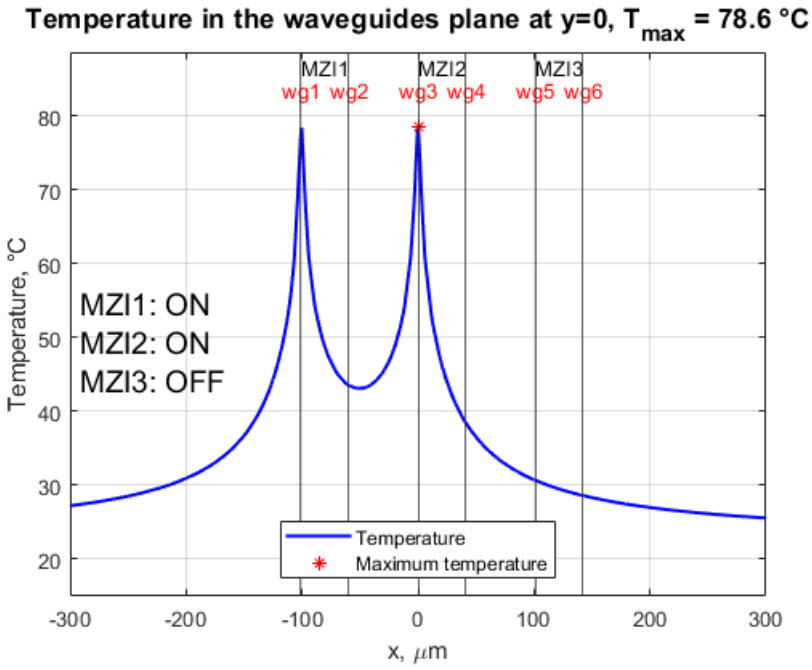


Figure 3.14: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 1 and MZI 2 active.

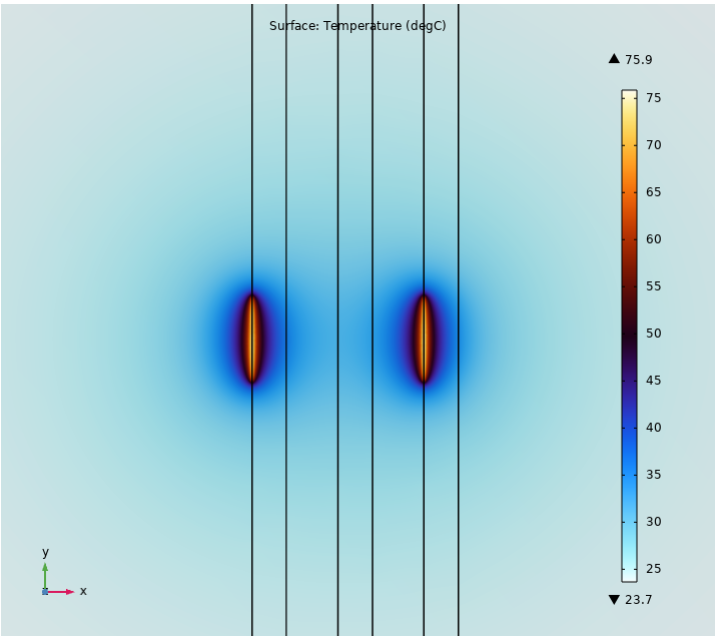


Figure 3.15: Thermal surface in COMSOL at waveguides height with the heaters of MZI 1 and MZI 3 active.

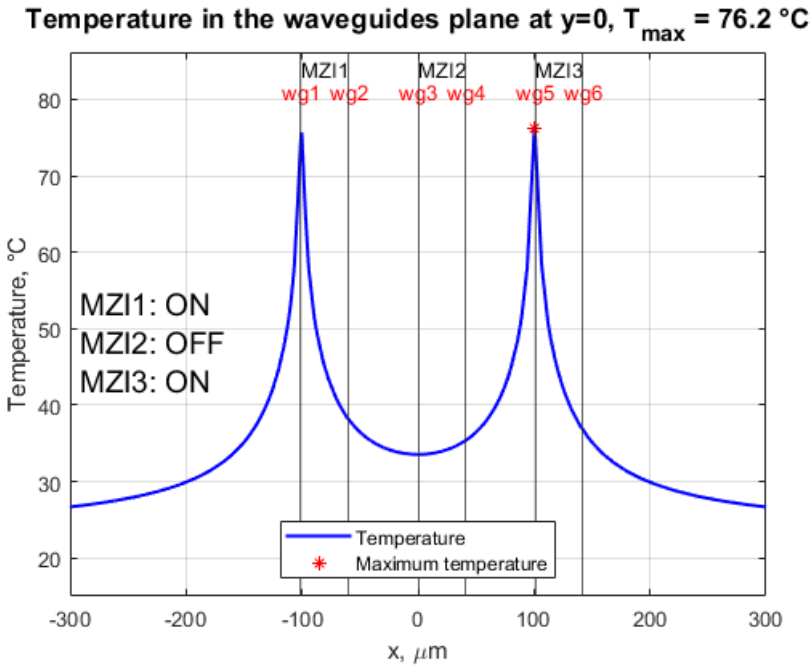


Figure 3.16: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 1 and MZI 3 active.

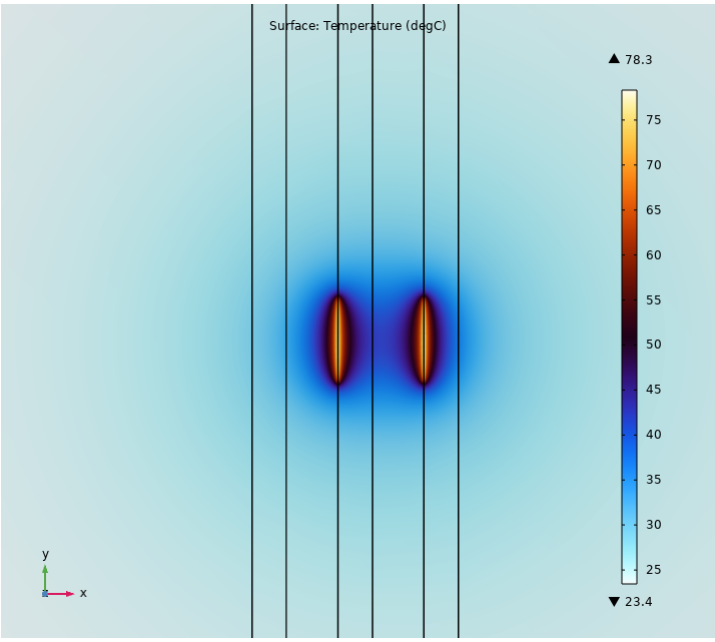


Figure 3.17: Thermal surface in COMSOL at waveguides height with the heaters of MZI 2 and MZI 3 active.

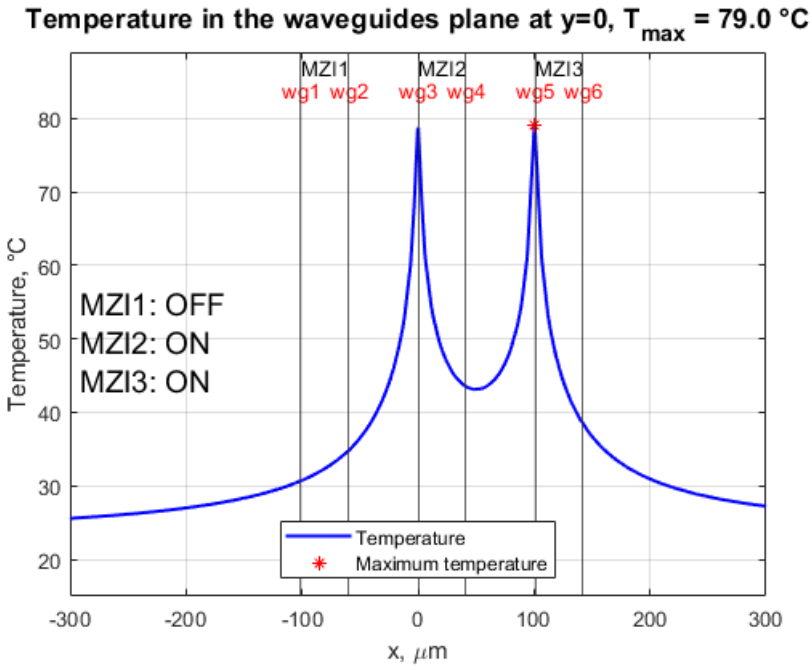


Figure 3.18: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 2 and MZI 3 active.

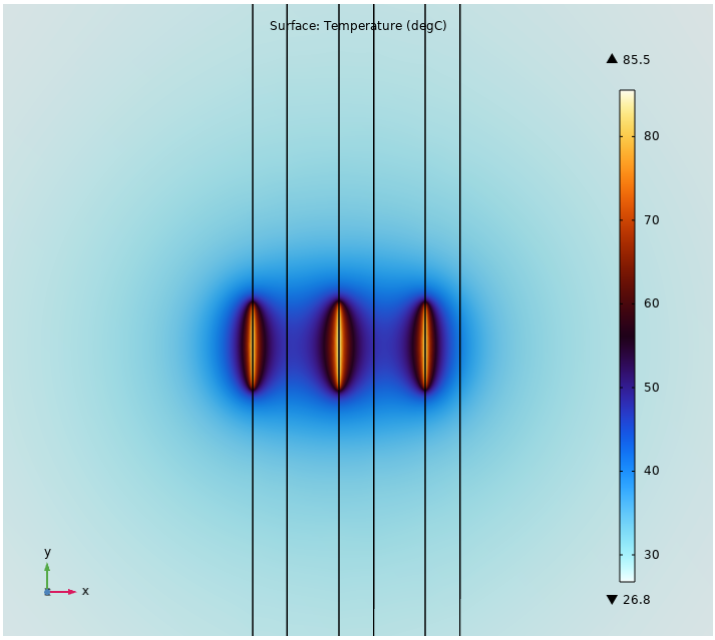


Figure 3.19: Thermal surface in COMSOL at waveguides height with all the heaters active.

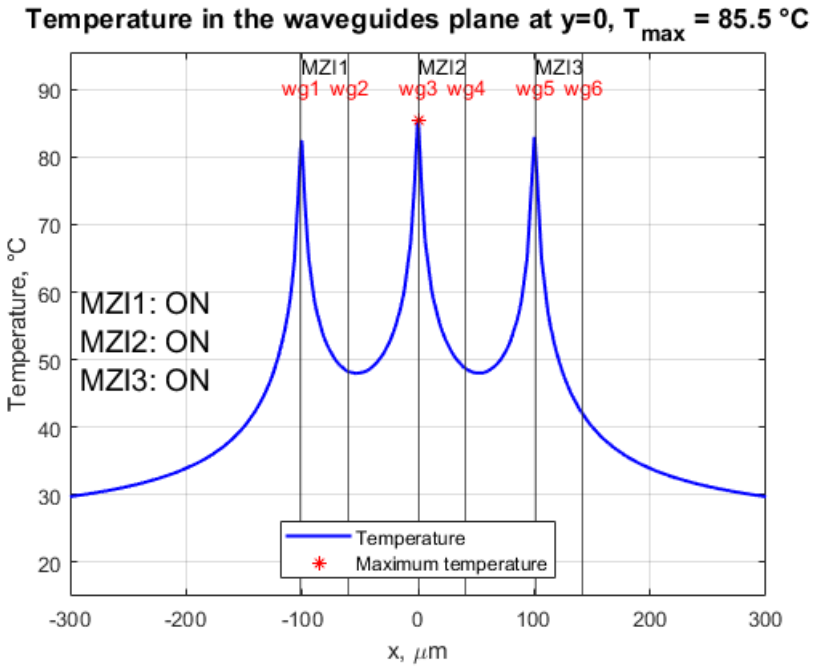


Figure 3.20: Temperature profile at $y = 0\,\mu\text{m}$ with MZI 1, MZI 2, and MZI 3 active.

3.3 Post-Process data analysis

After the thermal analysis process on COMSOL, I used all temperature data sets for the post-process. The aim of this post process is to find the phase shift of lights in the waveguides and subsequently on MZIs. As first, I have taken all data sets from temperature surfaces and set them in *.csv* files, to use these data on MATLAB. Then starts the formulation of the script for the post-process.

The script starts with the initialization of some physical constants, such as thermo-optic coefficient $\mathrm{d}n/\mathrm{d}T = 1.87 \cdot 10^{-4}\,(K^{-1})$, length of the heater $L = 100\,(\mu m)$ and wavelength range from $1524\,(nm)$ to $1568\,(nm)$.

```
%% Initialization
%Silicon dn/dT coefficient (1/K)
dn_dT = 1.87e-4;
%length of the heater (m)
L = 100e-6;
%wavelength range (m)
lambda = linspace(1.527e-6, 1.568e-6,10);
```

As second, a check of the active MZIs is performed to choose the correct dataset

```
%% check the MZI in ON-state: upper MZ, middle
   MZ, lower MZ
MZI1 = 1;
MZI2 = 0;
MZI3 = 0;
```

Then the required data sets are imported depending on the MZI flags set in the previous code lines

```
%% if state to choose the correct data file

if (MZI1 == 1 & MZI2 == 0 & MZI3 == 0)
  A=load('temp_data_MZI1_ON_2V.csv');
  N = 'MZI1 ON';
elseif(MZI1 == 0 & MZI2 == 1 & MZI3 == 0)
  A=load('temp_data_MZI2_ON_2V.csv');
```

```
  N = 'MZI2 ON';
elseif(MZI1 == 0 & MZI2 == 0 & MZI3 == 1)
  A=load('temp_data_MZI3_ON_2V.csv');
  N = 'MZI3 ON';
elseif(MZI1 == 1 & MZI2 == 1 & MZI3 == 0)
  A=load('temp_data_MZI1_ON_MZI2_ON_2V.csv');
  N = 'MZI1 ON and MZI2 ON';
elseif (MZI1 == 1 & MZI2 == 0 & MZI3 == 1)
  A=load('temp_data_MZI1_MZI3_ON_2V.csv');
  N = 'MZI1 ON and MZI3 ON';
elseif (MZI1 == 0 & MZI2 == 1 & MZI3 == 1)
  A=load('temp_data_MZI2_ON_MZI3_ON_2V.csv');
  N = 'MZI2 ON and MZI3 ON';
elseif (MZI1 == 1 & MZI2 == 1 & MZI3 == 1)
  A=load('temp_data_MZI1_ON_MZI2_ON_MZI3_ON_2V.
     csv');
  N = 'MZI1 ON and MZI2 ON and MZI3 ON';
end
```

After this preliminary procedure, the post-processing starts, first with the evaluation of the average temperatures in particular zones of the surface, in particular where the six waveguides are located, leading to a mean temperature of the waveguides. With the formula [19]

$$T_\pi = \frac{\lambda}{2L\mathrm{d}n/\mathrm{d}T} \tag{3.1}$$

I calculate the theoretic temperature needed for a π phase shift if no crosstalk is involved, it results at 42 °C. But know from the research that this circuit needs 2 V to reach a π phase shift, and with this voltage applied, the simulations reach significantly higher temperature values, meaning that the thermal crosstalk limits the phase shift of the entire MZI.

```
%% Evaluation of average temperature
X=A(:,1);
Y=A(:,2);
T=A(:,4)-273.15;
```

```
%waveguide positions (um)
Xwg=[-102 -100; -61.5 -59.5; -1 1; 39.5 41.5;
   100 102; 140.5 142.5];
Ywg=[-90 90; -90 90; -90 90; -90 90; -90 90; -90
    90;];
%Average temperature vector (degree C)
TavgWg=zeros(1,size(Xwg,1));
%loop over the waveguides
for k=1:size(Xwg,1)
     InsideWg=X>=Xwg(k,1) & X<=Xwg(k,2) & Y>=Ywg(
        k,1) & Y<=Ywg(k,2);
     if sum(InsideWg)<10
          warning('Not enough points');
     end
     TavgWg(k)=mean(T(InsideWg));
end
temp_pi = lambda./(2*L*dn_dT);  % (degree C)
```

Now I needed to fit the data to find the relationship between temperature and the effective index of silicon: the table used was found in [18], also I selected the data related to the wavelength used, which is 1550 nm, then interpolation is done to find the characteristic.

Silicon effective index								
λ (nm)	100 K	150 K	200 K	250 K	293 K	350 K	400 K	450 K
1500	3.4506	3.4568	3.4641	3.4722	3.4799	3.4908	3.5010	3.5117
1550	3.4457	3.4529	3.4601	3.4684	3.4757	3.4855	3.4956	3.5072
1600	3.4432	3.4493	3.4564	3.4644	3.4719	3.4826	3.4926	3.5032

Table 3.2: Fitting table for temperature vs effective index of silicon taken from [18].

```
%% Fitting data from https://srd.nist.gov/
   jpcrdreprint/1.555624.pdf
Trif=[100, 150, 200, 250, 293, 350, 400, 450,
   500, 550, 600, 650, 700, 750]-293;  %degC
```

```
Index=[3.4506, 3.4568, 3.4641, 3.4722, 3.4799,
  3.4908, 3.5010, 3.5117, 3.5228, 3.5343,
  3.5460, 3.5578, 3.5697, 3.5816]; %neff
TInterpolata=spline(Trif,Index); %interpolation
```

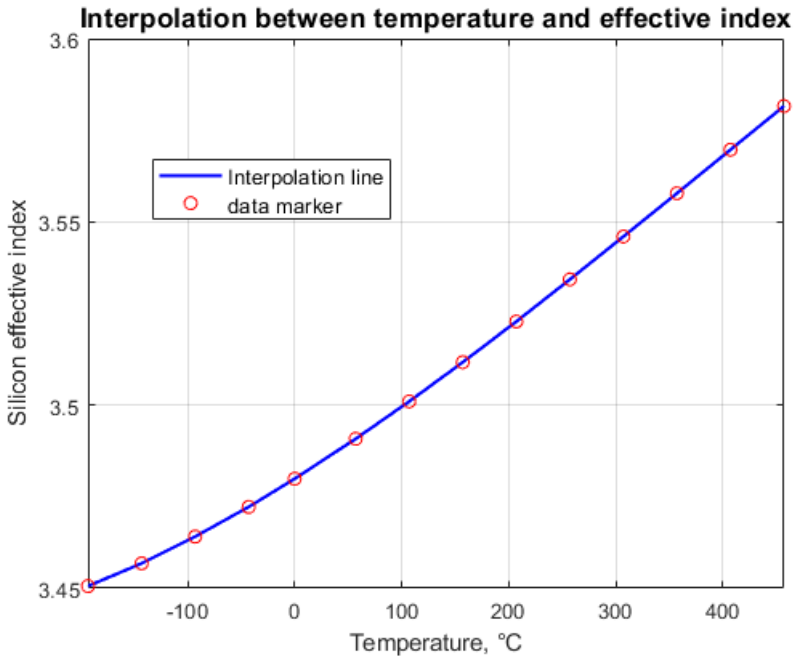


Figure 3.21: Interpolation of the data from Table 3.2 between the effective index and the temperature.

For the phase shift evaluation, first of all, the ambient temperature is set at 20 °C, then the temperature difference between ambient and average is calculated. Effective index values for ambient temperature and average temperatures are found using the interpolated function. An effective index vector is created as the difference between the ambient effective index and average effective indices and these results can be used for a surface plot to put in evidence the effective index variation. Finally, the phase shift for single waveguides can be calculated with [19]

$$\Delta\phi = \frac{2\pi L \Delta T \, \mathrm{d}n/\,\mathrm{d}T}{\lambda} \frac{180}{\pi} \tag{3.2}$$

But we have not finished with thermal crosstalk, since the phase difference between the arms of the same Mach-Zehnder has to be investigated, just taking the difference between the previous results. Finally, the coefficients are calculated for circuit model simulation purposes.

```
%% Phase shift evaluation from COMSOL average
   temperature
%ambient temperature (degree C)
T0 = 20;
%temperature difference
delta_T = TavgWg - T0;
%effective index of silicon @T0 from fitting
neff_T0 = ppval(TInterpolata,T0);

%for loop: effective index at each avg. temp.
for j =1:length(TavgWg)
    T = TavgWg(j);
    neff(j) = ppval(TInterpolata,T);
end

%effective index variation
delta_neff = neff - neff_T0;

for i = 1 : length(delta_T)
    T = delta_T(i);
    for j = 1 : length(lambda)
        %from temperature to phase
        delta_phi(i,j) = ((2*pi*L*T*dn_dT)./
          lambda(j))*(180/pi);
    end
end

% phase diff. between the wgs. of the same MZI
phase = [delta_phi(1,1)-delta_phi(2,1),
  delta_phi(3,1)-delta_phi(4,1),delta_phi(5,1)-
  delta_phi(6,1)];
coeff = 180./phase;
```

Chapter 4

Complete circuit model

After the detailed thermal analysis and the evaluation of the corresponding phase shift into the Mach-Zehnders, it is now time to configure a model that describes the experimental circuit behavior. In MATLAB the analysis started with a script considering only the MZIs, ignoring the thermal crosstalk, just to identify the correct physics. As said, the split of the MZIs has a sine form, and, since we are talking of power, the partition can be described by a sine squared. The modus operandi is to check the output power at each Mach-Zehnder, then the output of the circuit is rebuilt using the previous functions. the phase input in this case is just large vectors containing quantities equally spaced between 0 and π.

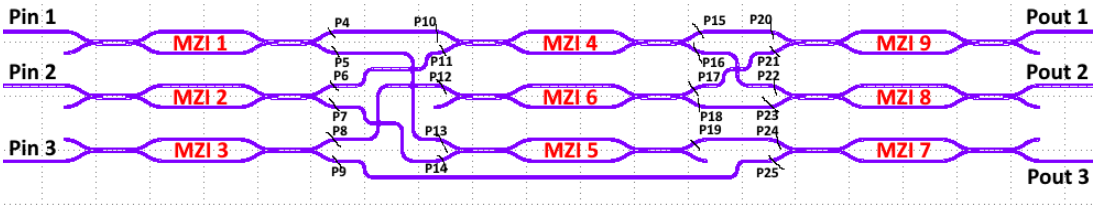


Figure 4.1: Experimental circuit layout with output powers at each MZI [11].

4.1 A first MATLAB implementation

From Fig. 4.1, equations that describe the circuit behavior are built, following power points are considered at each input and output of Mach-Zehnders to simplify the equations. MZI1, MZI2, MZI3, MZi4, MZI5, MZI7, MZI9 are in cross-state when no phase shift is applied, instead, MZI6 and MZI8 are in bar-state when there is no phase shift. The equations are built following the power points set in the figure. The reasoning behind the equations is to consider the fact that the optical outputs of every MZI oscillate as a sine squared due to the phase shift applied among the two values selected before as the maximum and minimum of the split. To express the fact that the MZI is in the opposite state of the one without applied voltage, a $\pi/2$ shift is used in addition to the thetas.

```
% MZI1, MZI2, MZI3, MZI4, MZI5, MZI7, MZI9 cross
    state; MZI6, MZI8 bar state
T=s1+(1-Loss(1)-2*s1)*sin(Theta(1,:) +
   PhaseOffset(1)).^2;
P(4,:) = P(1,:).*T;
P(5,:) = P(1,:).*(1-Loss(1)-T);
T=s1+(1-Loss(2)-2*s1)*sin(Theta(2,:) +
   PhaseOffset(2)).^2;
P(6,:) = P(2,:).*T;
P(7,:) = P(2,:).*(1-Loss(2)-T);
T=s1+(1-Loss(3)-2*s1)*sin(Theta(3,:) +
   PhaseOffset(3)).^2;
P(8,:) = P(3,:).*(1-Loss(3)-T);
P(9,:) = P(3,:).*T;
P(10,:) =  P(4,:);
P(11,:) =  P(6,:);
P(12,:) =  P(8,:);
P(13,:) =  P(5,:);
P(14,:) =  P(7,:);

T=s1+(1-Loss(4)-2*s1)*sin(Theta(4,:) +
   PhaseOffset(4)).^2;
```

```
P(15,:) = P(10,:).*T+P(11,:).*(1-Loss(4)-T);
P(16,:) = P(10,:).*(1-T)+P(11,:).*T;
T=s1+(1-Loss(6)-2*s1)*sin(Theta(6,:) + pi/2 +
   PhaseOffset(6)).^2;
P(17,:) = P(12,:).*T;
P(18,:) = P(12,:).*(1-Loss(6)-T);
T=s1 + s2*(sin(Theta(5,:)+PhaseOffset(5))).^2;
P(19,:) = P(13,:).*T+P(14,:).*(1-Loss(5)-T);
P(20,:) =  P(15,:);
P(21,:) =  P(17,:);
P(22,:) =  P(16,:);
P(23,:) =  P(18,:);
P(24,:) =  P(19,:);
P(25,:) =  P(9,:);
T=s1 + s2*((sin(Theta(9,:)+PhaseOffset(9))).^2);

Pout(1,:) = P(20,:).*T+P(21,:).*(1-Loss(9)-T);
T=s1+s2*((sin(Theta(8,:) + pi/2 + PhaseOffset(8)
   )).^2);
Pout(2,:) = P(22,:).*T+P(23,:).*(1-Loss(8)-T);
T=s1+s2*((sin(Theta(7,:)+PhaseOffset(7))).^2);
Pout(3,:) = P(24,:).*(1-Loss(7)-T)+P(25,:).*T;
```

The script starts with the load of the experimental data set of the weights they have found in the experiment. The Initialization of Theta is the vector said before, N is the number of iterations, and V is the voltage vector. Then I suppose that every MZI has a loss coefficient due to the structure, and all of them are imposed at 15%. In the end, we know these MZIs have a split in the two outputs that differs depending on the phase shift, but without a voltage applied the split is not totally in one output. The matrix of phases matrix is built with the MATLAB 'kron' command giving a variation of phase at one MZI at a time. In addition, a phase offset at each MZI is present, see Table 4.1, and they are added in the equations giving them an asymmetric behavior in the sine squared, since the phase starting point will not be zero, therefore if we apply a negative or a positive phase shift, these will give different results. This phase offset term is strictly related to the experimental

results, we can suppose that it is related to inequalities in MZI branches or could be related to residual heat from previous analysis. This last supposition is harder to prove since the offset values vary after every analysis.

MZI phase offsets (rad)								
MZI1	MZI2	MZI3	MZI4	MZI5	MZI6	MZI7	MZI8	MZI9
0.3	0.3	0.3	0.3	-0.15	0.0	0.1	0.2	0.2

Table 4.1: Phase offset values of the MZIs used in the circuit.

```
%number of considered MZI
NumMZI=9;
%number of voltage points
NumVPts=11;
%load experimental data
load('fitting_dataset.mat')
%MZI: residual power on weak branch
s1 = 0.01;
%MZI: power on strong branch
s2 = 1-2*s1;
%MZI: Phase offset
PhaseOffset  =  [0.3, 0.3, 0.3, 0.3, -0.15, 0.0,
    0.1,0.2, 0.2];
%Losses per MZI
Loss=0.15*ones(1,NumMZI);
%Input optical power (arbitrary units)
Pin = [1;0;0];

%Effective angles applied to the 9 MZIs
Theta=kron(eye(NumMZI),linspace(0,pi,NumVPts));
%%%%%%%%%%%%%%%%%%%%
```

The input optical power is applied at one channel a time in the experiment, so the three sets used in the simulations are set to $1-0-0$, $0-1-0$, and $0-0-1$. Then the vectors theta are built, in order to

have a correct phase shift on MZIs creating equally spaced vectors with elements between 0 and π. Finally, the weights are calculated, obviously for a comparison with the experimental data, so the ratios between each output, P_{out1}, P_{out2}, and P_{out3}, and the input are evaluated. Fig.s 4.2-4.10 show the results of the previous script where the blue lines are the simulated outputs and the red lines are the experimental results. It is clear how premature and simple is this solution: there is no thermal crosstalk involved at the moment. What we can see, instead, is the right positions of the peaks when Mach-Zehnders are directly involved in each analysis and give us the suggestion that the principles used to build the equations are correct. For example, in Fig. 4.3, the optical power starts from input 1 and is evaluated in output 2, the optical path passes through the MZI1, MZI4, and MZI8, as shown in the figure the equations correctly behave the peaks in these MZIs, the other variations present, as the variation of voltage in MZI2, are related to thermal crosstalk.

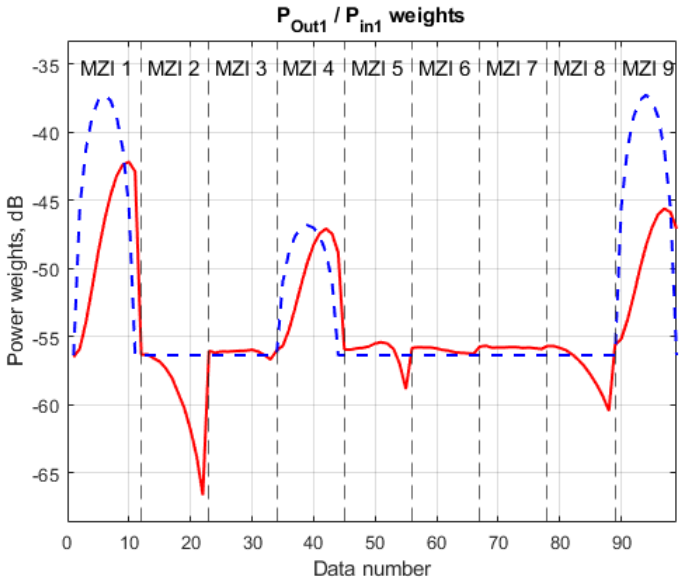


Figure 4.2: P_{out1}/P_{in1} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

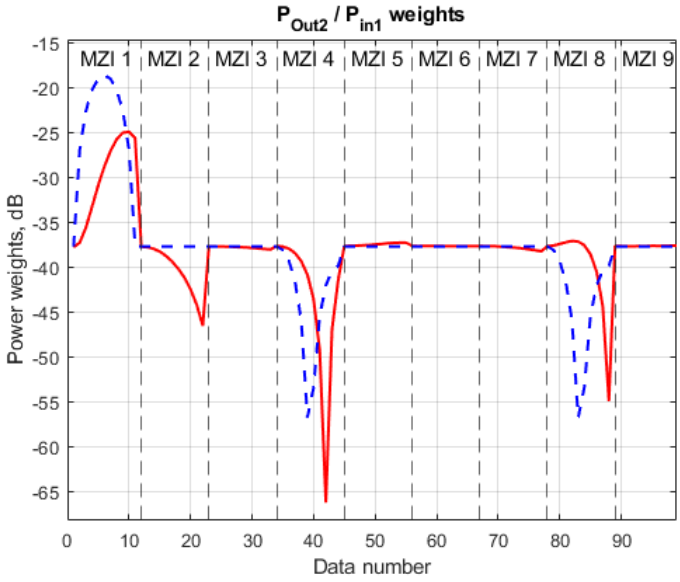


Figure 4.3: P_{out2}/P_{in1} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

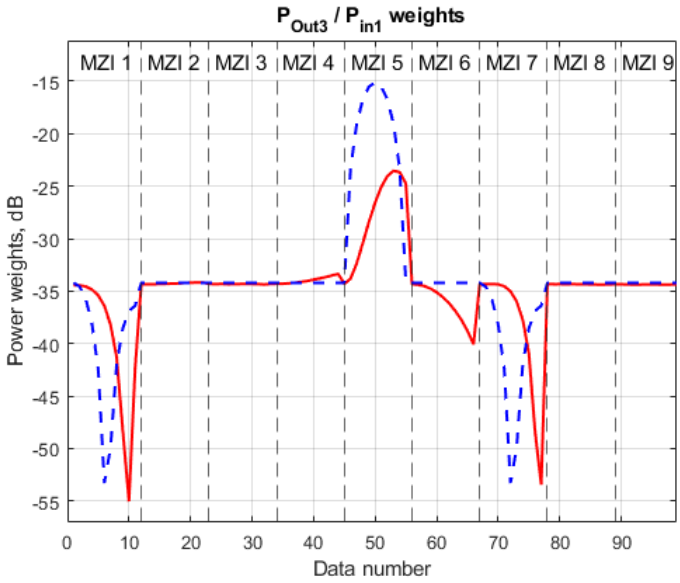


Figure 4.4: P_{out3}/P_{in1} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

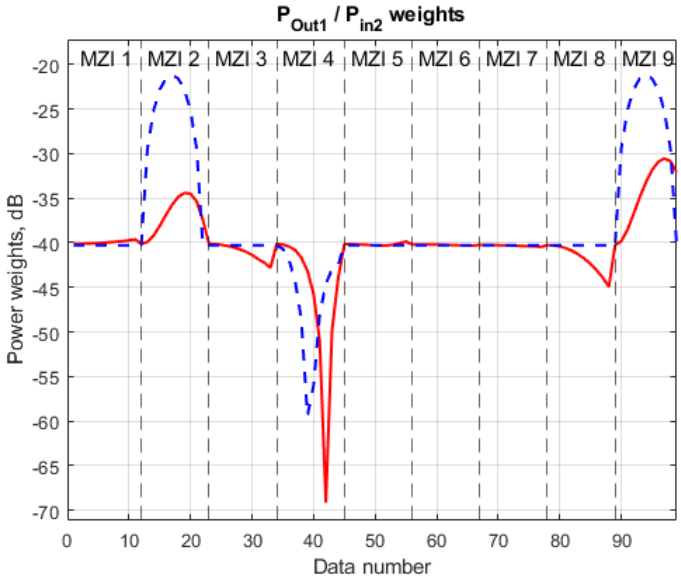


Figure 4.5: P_{out1}/P_{in2} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

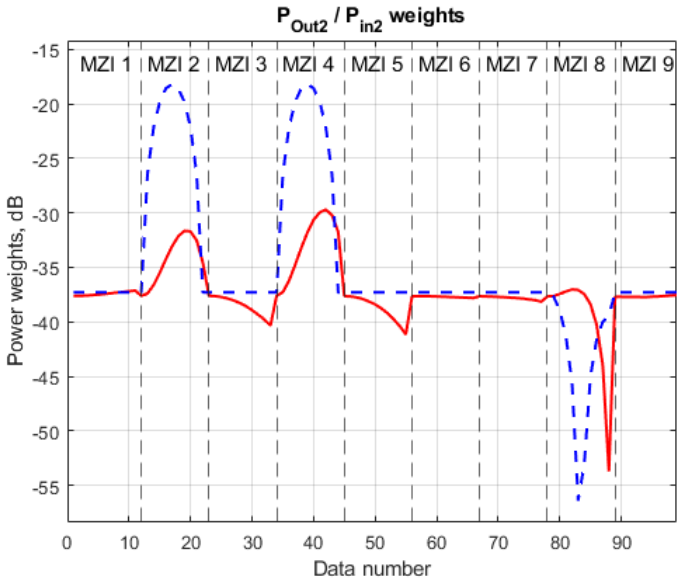


Figure 4.6: P_{out2}/P_{in2} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

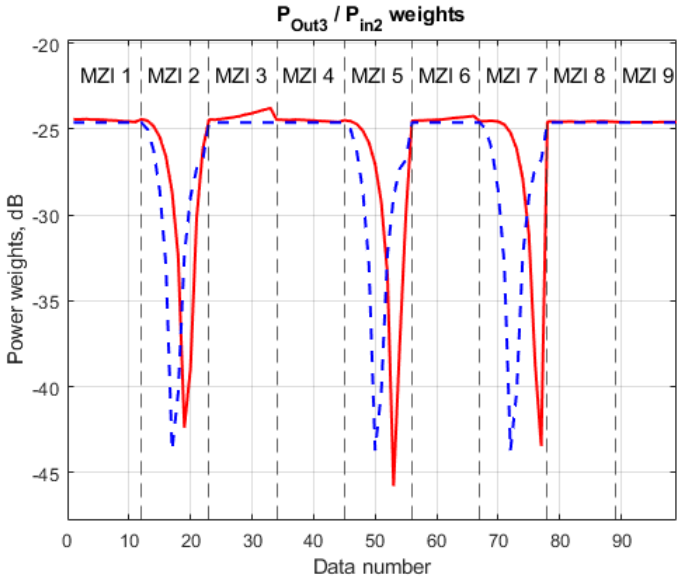


Figure 4.7: P_{out3}/P_{in2} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

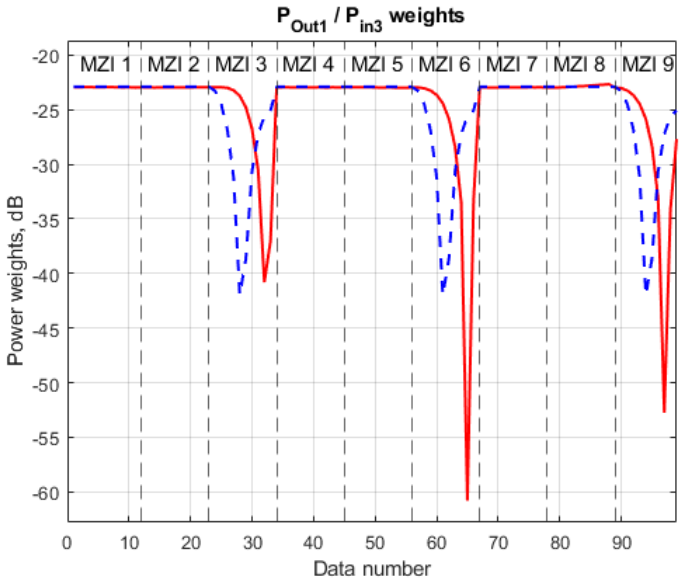


Figure 4.8: P_{out1}/P_{in3} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

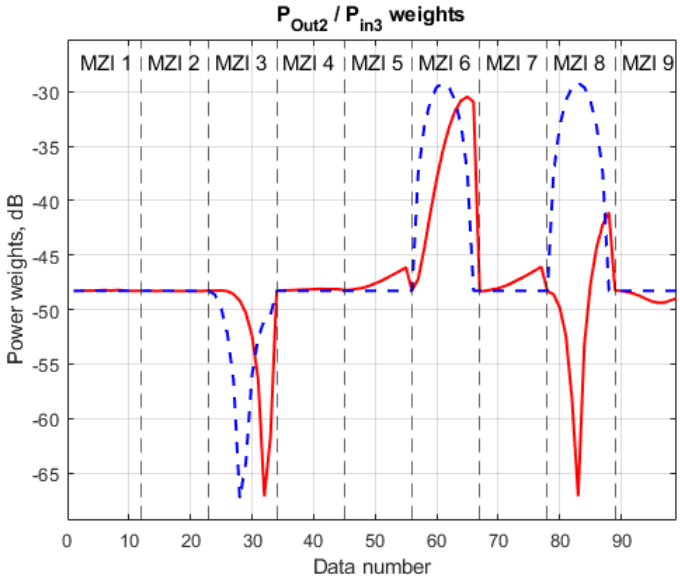


Figure 4.9: P_{out2}/P_{in3} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

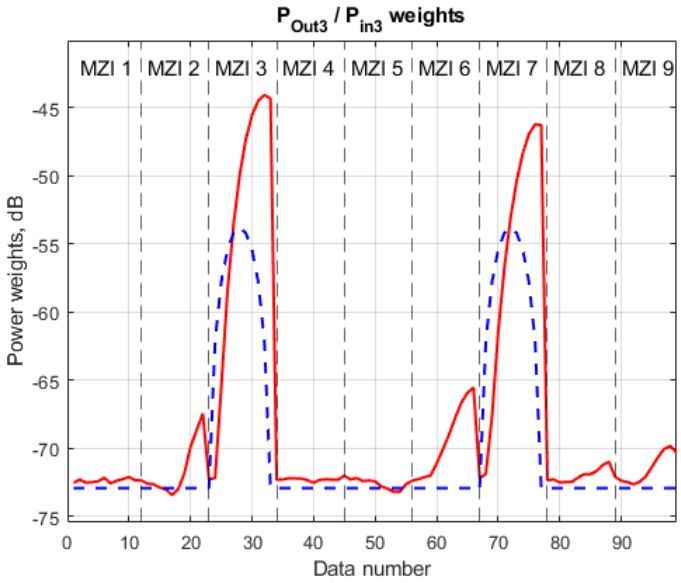


Figure 4.10: P_{out3}/P_{in3} weight. The blue dashed line represents the MATLAB simulation and the red line represents the experiment data.

4.2 MATLAB analysis including thermal crosstalk

In this section I will deepen the analysis and try to explain more correctly the behavior of the experimental weights.

In previous figures, every variation of weight, outside of the direct influence of MZIs, is related to the effect of thermal crosstalk in the circuit and to the phase offset intrinsically present in the Mach-Zehnders.

4.2.1 Thermal crosstalk function

To consider these variables, a MATLAB function has to be introduced to manage the thermal crosstalk and the data obtained from the post-processing analysis with COMSOL data.

Since we consider the crosstalk influences each other only the three MZIs in columns, the evaluation of thermal crosstalk is similar for the triples MZI1 - MZI2 - MZI3, MZI4 - MZI6 - MZI5, and MZI9 - MZI8 - MZI7. In the function, the inputs are three related phases, and, the outputs are the three thermally-adjusted phases. A loop is used, every time a phase shift is not zero, will be adjusted, and will adjust the following MZIs phase with the coefficients obtained from the thermal post-process script already discussed.

```
function [Thetaa, Thetab, Thetac] = scalephases(
   Thetax,Thetay,Thetaz)
Thetaa = Thetax;
Thetab = Thetay;
Thetac = Thetaz;
for i = 1:length(Thetax)
    if not (Thetaa(i) == 0)
        Thetab(i) = Thetaa(i)/31.5976;
        Thetac(i) = Thetaa(i)/91.2628;
        Thetaa(i) = Thetaa(i)/1.9803;
    elseif not (Thetab(i) == 0)
        Thetaa(i) = -Thetab(i)/16.7288;
```

```
        Thetac(i) = Thetab(i)/31.9156;
        Thetab(i) = Thetab(i)/1.9665;
    elseif not (Thetac(i) == 0)
        Thetaa(i) = -Thetac(i)/63.6936;
        Thetab(i) = -Thetac(i)/15.8803;
        Thetac(i) = Thetac(i)/2.0067;
     end
end
end
```

This function is called as follows in the main script, just after the definition of theta:

```
[Theta1, Theta2, Theta3] = scalephases(Theta1,
  Theta2, Theta3);
[Theta4, Theta6, Theta5] = scalephases(Theta4,
  Theta6, Theta5);
[Theta9, Theta8, Theta7] = scalephases(Theta9,
  Theta8, Theta7);
```

4.2.2 Script for the complete analysis

This analysis is similar to the previous one: MZIs are adjusted with phase offsets, phase shifts are varied by the function introducing the thermal effects. Equations have the same principles rewritten in a more compact and efficient way to improve the MZIs behavior, a T variable is introduced to show how the transmission at the outputs oscillates. Phase offsets are important since they enable a different behavior of the MZIs, giving a dependence on the sign of the phase shift that occurs due to direct thermal control or thermal crosstalk. The results are shown in Fig.s 4.11-4.21. The figures show a huge improvement from the previous analysis, giving the possibility of having a better simulation of what happens in the circuit, thermal crosstalk is now implemented in the model, and the peaks related to it are visible.

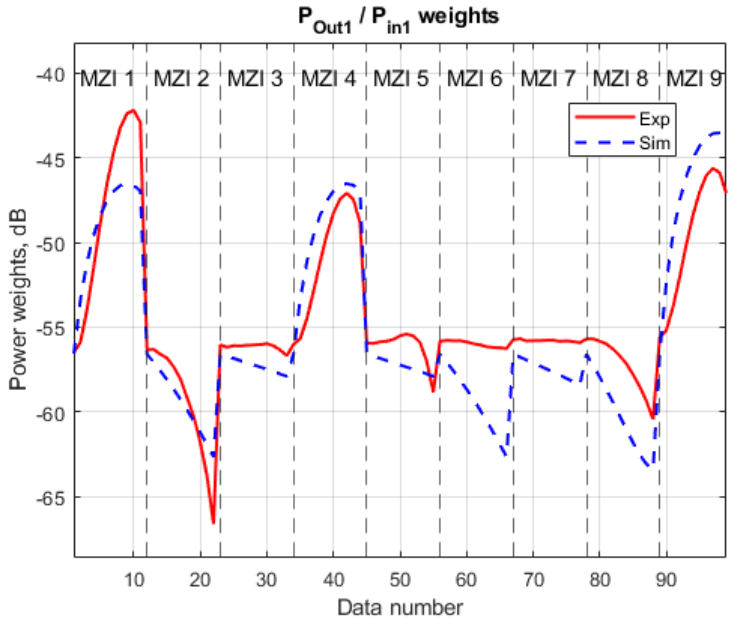


Figure 4.11: P_{out1}/P_{in1} weight. Thermal crosstalk correction and phase offset included.

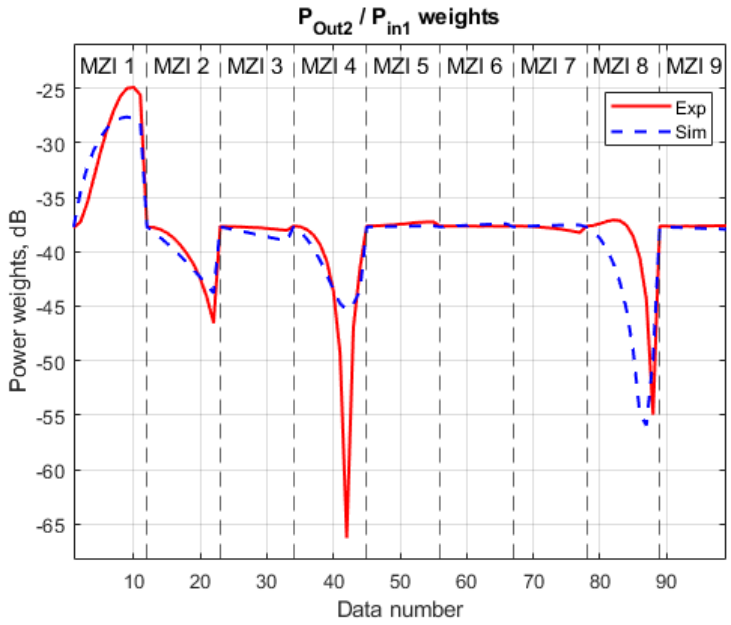


Figure 4.12: P_{out2}/P_{in1} weight. Thermal crosstalk correction and phase offset included.

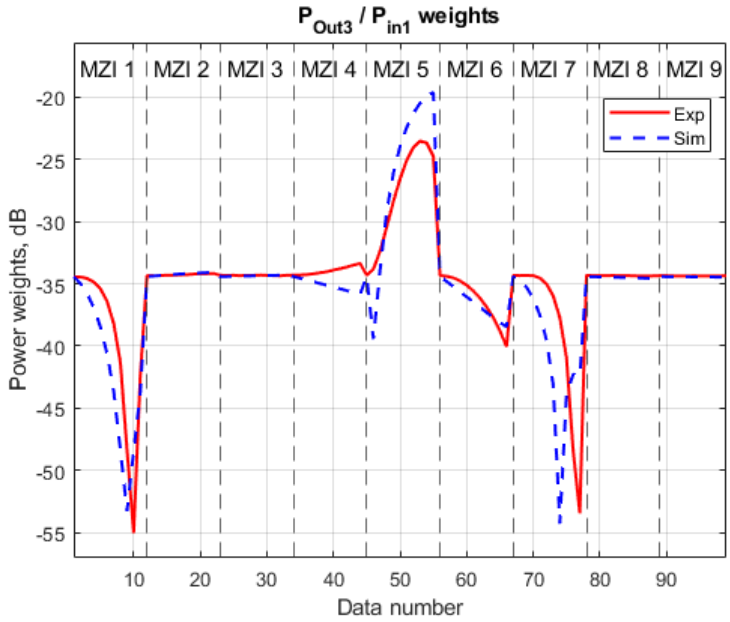


Figure 4.13: P_{out3}/P_{in1} weight. Thermal crosstalk correction and phase offset included.

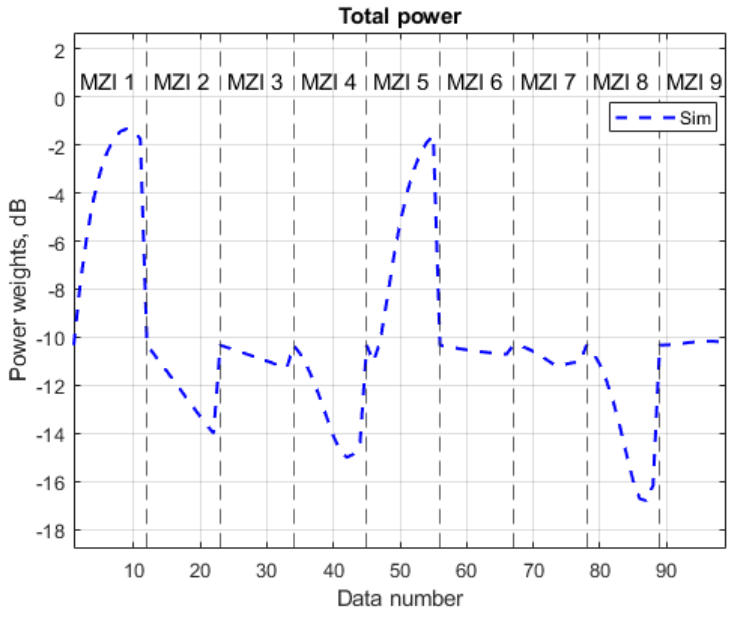


Figure 4.14: Total output power considering P_{in1}. Thermal crosstalk correction and phase offset included.

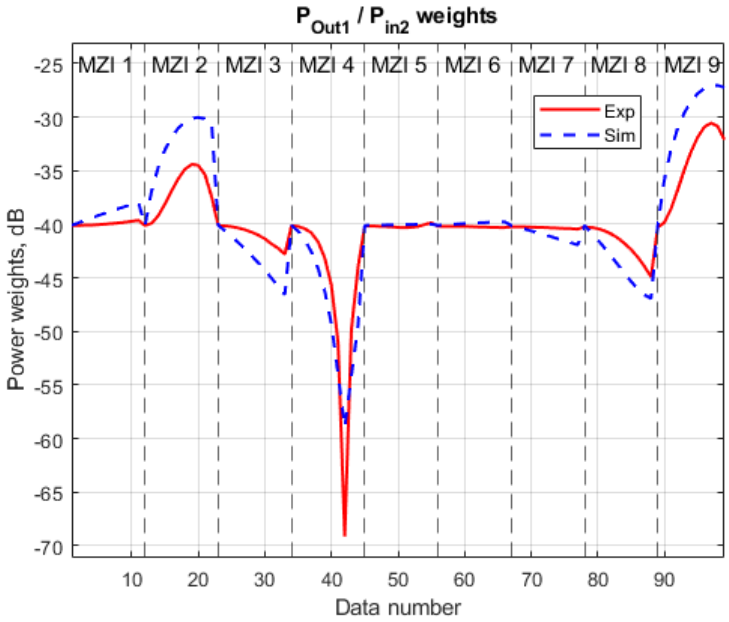


Figure 4.15: P_{out1}/P_{in2} weight. Thermal crosstalk correction and phase offset included.

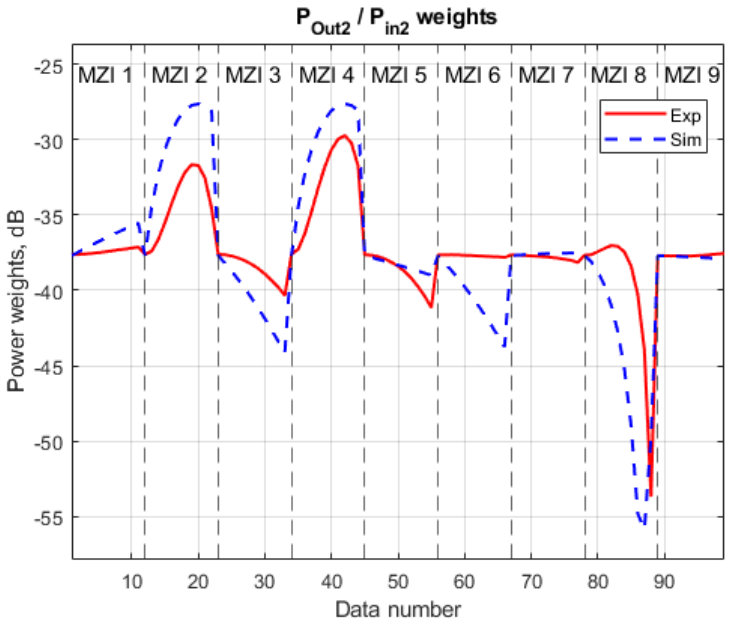


Figure 4.16: P_{out2}/P_{in2} weight. Thermal crosstalk correction and phase offset included.

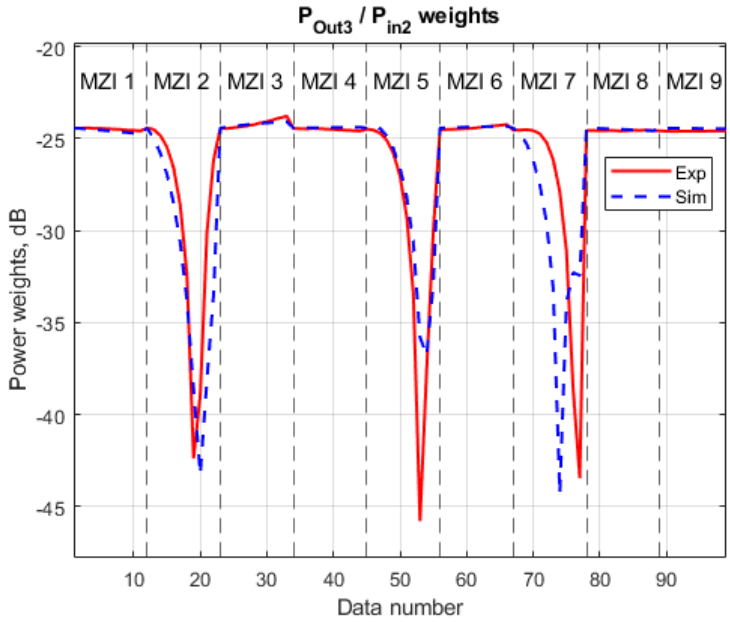


Figure 4.17: P_{out3}/P_{in2} weight. Thermal crosstalk correction and phase offset included.

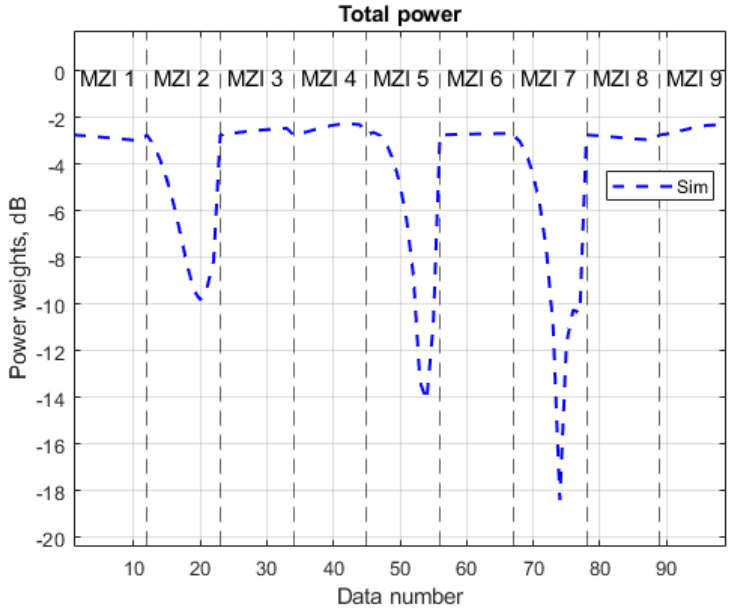


Figure 4.18: Total output power considering P_{in2}. Thermal crosstalk correction and phase offset included.

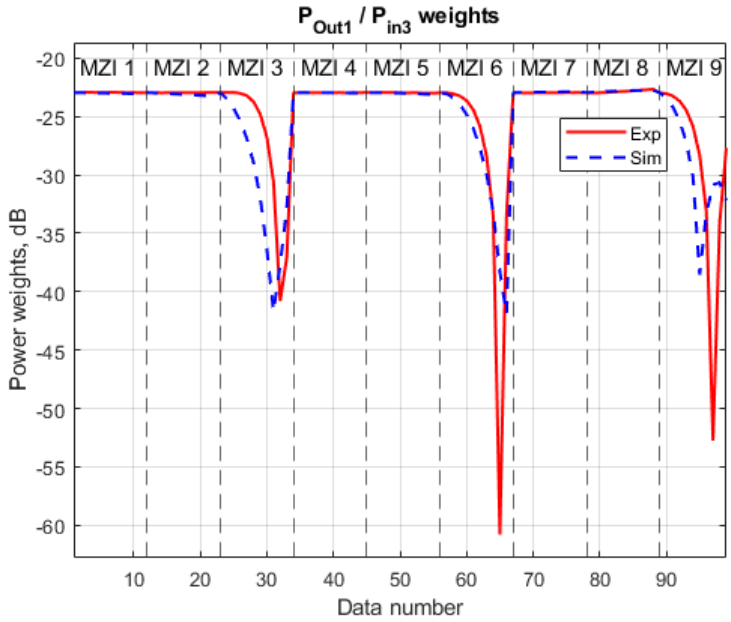


Figure 4.19: P_{out1}/P_{in3} weight. Thermal crosstalk correction and phase offset included.

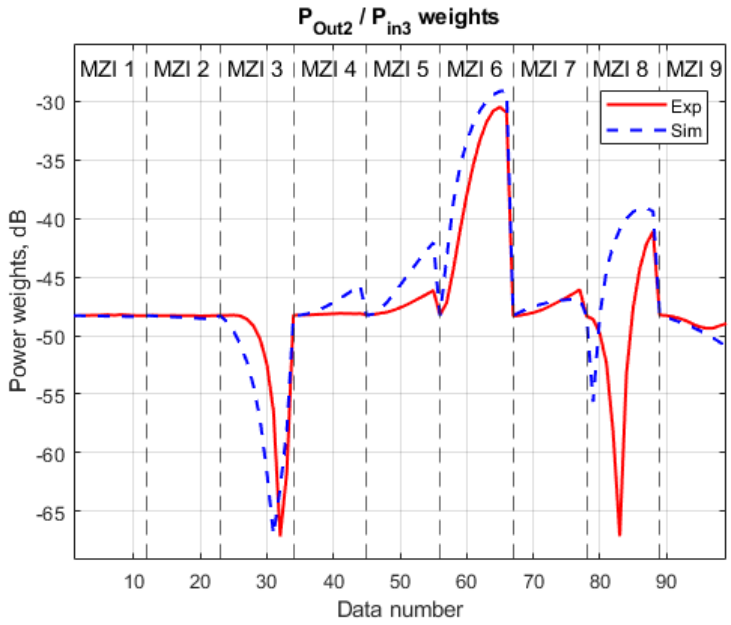


Figure 4.20: P_{out2}/P_{in3} weight. Thermal crosstalk correction and phase offset included.

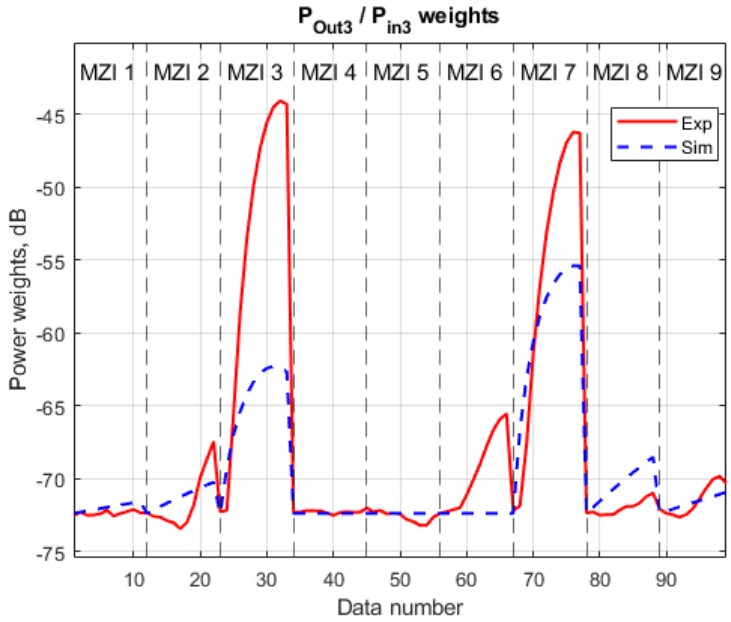


Figure 4.21: P_{out3}/P_{in3} weight. Thermal crosstalk correction and phase offset included.

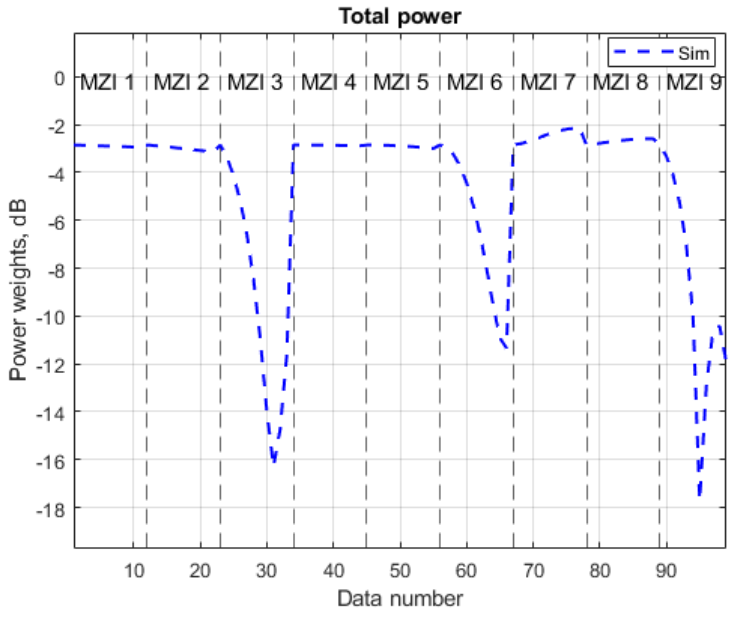


Figure 4.22: Total output power considering P_{in3}. Thermal crosstalk correction and phase offset included.

Also, in this case, we can take as an example, the Fig. 4.12 related to input 1 and output 2, The optical path passes through MZI1, MZI4, and MZI8, the peaks due to voltage variation in these MZIs are present and thermally-adjusted by the function that evaluates the thermal crosstalk also inside the same Mach-Zehnder; in addition, the thermal crosstalk related to near MZIs, such as MZI2, is now taken into account. Finally, the model describes the experimental behavior, giving us the attestation that thermal crosstalk is the main variable we have to consider in this type of optical circuit.

Chapter 5

Optimal control signals

The model that I developed is able to reproduce with reasonable accuracy the behavior of the real device. It can therefore be used to determine which control signals (voltages) must be applied to obtain the desired optical function at the output ports. For example, with a proper set of input voltages, output port 1 could be set to return the logical "AND" of the input optical signals, supposing that they can assume just high (logical 1) or low (logical 0) power levels. At the same time, output port 2 could return the logical "OR" of the same input signals. Since the network is programmable, the realized function can be dynamically changed simply by modifying the applied voltages.

The following analysis will focus on the determination of the voltages required to obtain at the same time a logical AND and port 1 and a logical OR at port 2. A Machine Learning based approach is often used for this type of reverse design problems: we can easily generate a large data set randomly changing the nine control voltages and collecting the output power for the 2^3 allowed combination of the logical signal at the three inputs. An agent can be then trained to answer to the question "which are the voltages required to obtain the desired output powers?". In this case, however, the solution to this problem is not unique, since different combinations of the control signals can answer to the question, and the training can become difficult and can lead to not accurate predictions [21]. I therefore decided to rely on optimization techniques, in particular Particle Swarm Optimization.

Table of truth 3 inputs			
INPUTS	AND	OR	XOR
'0'-'0'-'0'	'0'	'0'	'0'
'0'-'0'-'1'	'0'	'1'	'1'
'0'-'1'-'0'	'0'	'1'	'1'
'0'-'1'-'1'	'0'	'1'	'0'
'1'-'0'-'0'	'0'	'1'	'1'
'1'-'0'-'1'	'0'	'1'	'0'
'1'-'1'-'0'	'0'	'1'	'0'
'1'-'1'-'1'	'1'	'1'	'1'

Table 5.1: Table of truth of AND, OR, and XOR gates with three inputs

5.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a computational optimization technique inspired by the social behavior of birds flocking or schooling of fish. It is a population-based stochastic optimization algorithm that searches for the optimal solution by iteratively adjusting a group of particles in the search space. Each particle represents a potential solution to the problem and its position in the search space is updated on the basis of its own best solution and the best solution of its neighbors. The algorithm is commonly used in solving optimization problems in engineering, computer science, economics, and other fields.

PSO is implemented in MATLAB in the "particleswarm" function, available in the Optimization Toolbox.

5.1.1 Thresholds definition

Before dealing with the optimization, we need to define the thresholds to apply to each port in order to decide whether the received optical signals can be considered logical 1 or 0. To do so, I calculated the output at the 3 ports for 2^3 combinations of input signals, for 50000 random values of the control voltages. This data set was then used to evaluate the average and median otical power at the output. The

median was then used as a thresholds to differentiate between low- and high-level signals.

In the function "*getdata*" we include the parameters that can be useful for the model, such as phase offset of MZIs, split of the power in the Mach-Zehnder outputs, and the optical losses.

```
function Data=GetData()
Data.P_offset  =  [0.3, 0.3, 0.3, 0.3, -0.15, pi
   /2+0.0, 0.1,pi/2+0.2, 0.2];
NumMZI=length(Data.P_offset);
Data.s1 = 0.01*ones(1,NumMZI); % weak arm split
   coeff
Data.s2 = 1-2*Data.s1; % strong arm split coeff
Data.Loss=0.15*ones(1,NumMZI);
```

The function *MLCrosstalkFunction* uses the same equations used in the previous chapter for the weights calculations, including crosstalk effects.

```
function [Thetaa, Thetab, Thetac] =
   MLCrosstalkFunction(Thetax, Thetay, Thetaz)
Thetaa = Thetax;
Thetab = Thetay;
Thetac = Thetaz;
%%% cases %%%
%%% 1 - 0 - 0 %%%
if ((Thetaa ~= 0) && (Thetab == 0) && (Thetac ==
    0))
            Thetaa = Thetaa/1.9803;
            Thetab = Thetaa/31.5976;
            Thetac= Thetaa/91.2628;
%%% 0 - 1 - 0 %%%
elseif((Thetaa == 0) && (Thetab ~= 0) && (Thetac
    == 0))
            Thetaa = -Thetab/16.7288;
            Thetab = Thetab/1.9665;
            Thetac = Thetab/31.9156;
%%% 0 - 0 - 1 %%%
```

```
elseif((Thetaa == 0) && (Thetab == 0) && (Thetac
    ~= 0))
            Thetaa = -Thetac/63.6936;
            Thetab = -Thetac/15.8803;
            Thetac = Thetac/2.0067;
%%% 1 - 1 - 0 %%%
elseif((Thetaa ~= 0) && not(Thetab == 0) && (
   Thetac == 0))
            Thetaa = Thetaa/2.2098;
            Thetab = Thetab/1.8217;
            Thetac = Thetab/23.4211;
%%% 1 - 0 - 1 %%%
elseif((Thetaa ~= 0) && (Thetab == 0) && (Thetac
    ~= 0))
            Thetaa = Thetaa/2.0096;
            Thetab = -Thetaa/31.5958;
            Thetac = Thetac/1.9304;
%%% 0 - 1 - 1 %%%
elseif ((Thetaa == 0) && (Thetab ~= 0) && (
   Thetac ~= 0))
            Thetaa = -Thetab/13.0643;
            Thetab = Thetab/2.2023;
            Thetac = Thetac/1.8557;
%%% 0 - 0 - 0 %%%
elseif((Thetaa == 0) && (Thetab == 0) && (Thetac
    == 0))
            Thetaa = 0;
            Thetab = 0;
            Thetac = 0;
%%% 1 - 1 - 1 %%%
elseif ((Thetaa ~= 0) && (Thetab ~= 0) && (
   Thetac ~= 0))
            Thetaa = Thetaa/2.2874;
            Thetab = Thetab/2.0586;
            Thetac = Thetac/1.8180;
end
```

Finally, a third function called *CalculateNetworkOutput* is used to calculate the output powers that receive the parameters, the combination of input optical power and the voltages of the nine MZIs. This function is similar to the one discussed in the previous chapter.

Using the following script, nine random voltage values are applied in each simulation; thus, each line of the dataset will be composed by nine voltages between 0 V and 2 V, followed by 3×8=24 values of the output power calculated at the 3 output ports for the eight combinations of optical input power; this calculation is repeated 50000 times.

```
Data=GetData();
NumMZI=length(Data.s1);
NumRecords=50000;
hFile=fopen('Dataset.txt','w');
for kR=1:NumRecords
    V=2*rand(1,NumMZI);
    fprintf(hFile,'%g ',V);
    for kV=0:7
        Pins=dec2bin(kV,3)-'0';
        Pouts=CalculateNetworkOutput(Data,Pins,V
          );
        fprintf(hFile,'%g ',Pouts);
    end
    fprintf(hFile,'\n');
end
fclose(hFile);
```

The results of the histograms of the data set are shown in Fig.5.1. The calculation of the median returns threshold values equal to 0.076, 0.046, and 0.093 for the three ports, respectively.

5.1.2 Final optimisation

The calculated medians are used in the following routine to calculate the voltages; with respect to the calculated medians M, I used as thresholds $0.85\times M$ and $1.15\times M$ to identify the logical low and logical high signals, respectively, in order to allow for a central range were the logical output is undefined to account for optical noise.

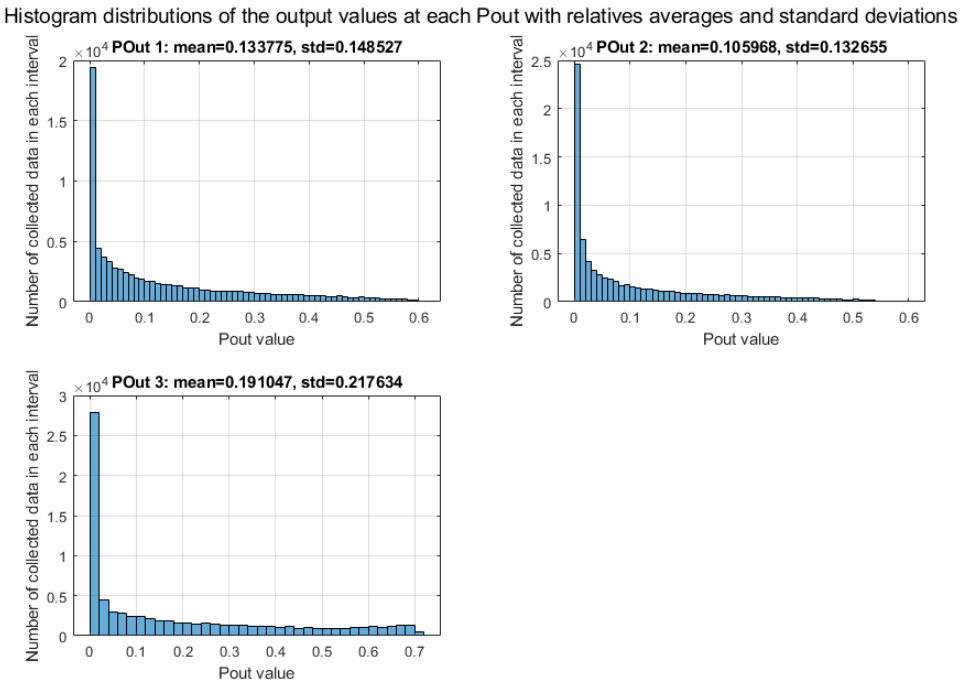


Figure 5.1: Histogram distributions of collected data for each output with interval range of 0.01 for the first two ports and 0.02 for the last port.

```
options = optimoptions('particleswarm',...
    'SwarmSize',1000,'HybridFcn',@fmincon,...
    'MaxStallIterations',100,...
    'MinNeighborsFraction',1);
V=particleswarm(@Opt,9,zeros(9,1), 2*ones(9,1),
   options );

function [Err,Ind]=Opt(V)
L=[0.076075600000000   0.046508300000000
   0.093086700000000]*0.85;
H=[0.076075600000000   0.046508300000000
   0.093086700000000]*1.15;
```

```
Data.P_offset  =  [0.3, 0.3, 0.3, 0.3, -0.15, pi
   /2+0.0, 0.1,pi/2+0.2, 0.2];
NumMZI=length(Data.P_offset);
Data.s1 = 0.01*ones(1,NumMZI); % Split on weak
   banch
Data.s2 = 1-2*Data.s1; % Split on string branch
Data.Loss=0.15*ones(1,NumMZI);

Data.PoutOffset=[-21,-24.5, -16.5];
Target=[0 0; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 1 1];

Target(Target==0)=-1;
Pouts=zeros(8,3);
for kV=0:7
    Pins=dec2bin(kV,3)-'0';
    Pouts(kV+1,:)=CalculateNetworkOutputMO(Data,
       Pins,V);
end
Res=(Pouts>H)-(Pouts<L);
%remove third column, unused
Res=Res(:,[1,2]);
%evaluate error as difference of Res with target
Err=sum(abs(Res-Target),'all');
%identify correct ports
Ind=Res==Target;
end
```

According to the PSO predictions, the voltages to apply to the nine MZIs to obtain the AND and OR functions at the first two ports are 1.847 V, 1.877 V, 0.009 V,0.6821 V, 0.833 V,1.775 V,1.180 V,0.406 V,0.169 V. The network programming can be changed simply modifying those voltages to implementing a different set of functions.

Chapter 6

Conclusions

In this thesis, a 3×3, voltage controlled, neuromorphic optical circuit was studied and simulated, understanding and reproducing the experimental measurements of a silicon photonics device designed at the Technical University of Denmark.

Beam Propagation Methods analysis, carried out in Synopsys RSoft, we used to fully characterize the optical components, namely MZIs, MMIs, and cross-connections, used in the circuit and estimate their optical losses.

Thermal simulations with COMSOL Multiphysics have shown that thermal crosstalk between adjacent MZIs is a limiting factor that must be taken into account to properly reproduce the experimental evidences.

The final model, written in MATLAB, is able to reproduce the measured behavior and allows the user to predict the output powers as a function of the applied voltage control signals and the input optical powers.

Optimization techniques were finally used to determine the control signals required to use this neuromorphic circuit as optical logical gate.

The methodology used in this thesis can applied to larger optical circuits, removing unnecessary distances between optical components, decreasing therefore the overall circuit footprint, and optimising the control strategy to implement the desired optical functions.

Bibliography

[1] R. Vishwa et al. 2020 IOP Conf. Ser.: Mater. Sci. Eng. 912 062029

[2] Xu Weilin, Wang Jingjuan, Yan Xiaobing, "Advances in Memristor-Based Neural Networks", Frontiers in Nanotechnology, 3 (2021)

[3] A. Abraham, "Artificial Neural Networks. In Handbook of Measuring System Design" (eds P.H. Sydenham and R. Thorn) (2005)

[4] C. D. Schuman, "A Survey of Neuromorphic Computing and Neural Networks in Hardware." ArXiv abs/1705.06963 (2017)

[5] S. Nafea, A. Dessouki, and E. El-Rabaie, "Memristor Overview up to 2015," 24. 79-106 (2015)

[6] D. Thomson, et al., "Roadmap on silicon photonics," Journal of Optics, 18(7):07300 (2016)

[7] Gray, D. et al., "Crystal Structure of Graphite , Graphene and Silicon," Physics for Solid State Applications (2009)

[8] https://www.el-cat.com/silicon-properties.htm

[9] Liu, J. et al., "Research progress in optical neural networks: theory, applications and developments," PhotoniX, 2, 5 (2021)

[10] Ding, Y. et al., "Reconfigurable SDM Switching Using Novel Silicon Photonic Integrated Circuit," Scientific Reports, 6, 39058 (2016)

[11] Cem, Ali, et al. "Data-driven Modeling of Mach-Zehnder Interferometer-based Optical Matrix Multipliers," arXiv preprint arXiv:2210.09171 (2022)

[12] E. Paolini et al., "Photonic-aware neural networks," Neural Computing and Applications,34, pp. 15589-15601 (2022)

[13] Soldano, Lb and E. C. M. Pennings. "Optical multi-mode interference devices based on self-imaging: principles and applications," Journal of Lightweight Technologies, 13, pp. 615-627 (1995)

[14] Mohammadi-Pouyan, S. et al. "High-performance Mach–Zehnder

modulator using tailored plasma dispersion effects in an ITO/graphene-based waveguide," Scientific Reports, 12, 12738 (2022)

[15] F. Xu and A. W. Poon, "Silicon cross-connect filters using microring resonator coupled multimode-interference-based waveguide crossings," Optics Express, 16, 8649-8657 (2008)

[16] S. Nevlacsil, P. Muellner, M. Sagmeister, J. Kraft, and R. Hainberger, "Broadband low loss and ultra-low crosstalk waveguide crossings based on a multimode interferometer for 840 nm operation," OSA Continuum, 3, 334-344 (2020)

[17] Liu, S., Feng, J., Tian, Y. et al. "Thermo-optic phase shifters based on silicon-on-insulator platform: state-of-the-art and a review," Frontiers in Optoelectronics, 15, 9 (2022)

[18] H. H. Li, "Refractive index of silicon and germanium and its wavelength and temperature derivatives," Journal of Physical and Chemical Reference Data, 9, 561-658 (1980)

[19] S. De et al., "Design and Simulation of Thermo-Optic Phase Shifters With Low Thermal Crosstalk for Dense Photonic Integration," in IEEE Access, 8, pp. 141632-141640, 2020, doi: 10.1109/ACCESS.2020.3013116.

[20] Y. Shen et al., “Deep learning with coherent nanophotonic circuits,” Nature Photonics, 11, pp. 441–446 (2017)

[21] Y. Xu et al., "Interfacing photonics with artificial intelligence: an innovative design strategy for photonic structures and devices based on artificial neural networks," Photonic Research, 4, pp. B135–B152 (2021)