FACULTY OF ENGINEERING
AND ARCHITECTURE

The Wavelength Dimension in Photonic Reservoir Computing

Emmanuel Gooskens

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Photonics Engineering

Supervisors
Prof. Peter Bienstman, PhD* - Prof. Joni Dambre, PhD**

* Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

** Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

May 2024

GHENT
UNIVERSITY

ISBN 978-94-6355-830-3
NUR 959, 965
Wettelijk depot: D/2024/10.500/35

Members of the Examination Board

Chair

Prof. Luc Dupré, PhD, Ghent University

Other members entitled to vote

Prof. Jeroen Beeckman, PhD, Ghent University
Prof. Wim Bogaerts, PhD, Ghent University
Alfredo De Rossi, PhD, Thales, France
Prof. Serge Massar, PhD, Université libre de Bruxelles

Supervisors

Prof. Peter Bienstman, PhD, Ghent University
Prof. Joni Dambre, PhD, Ghent University

Acknowledgements

The idea of being a researcher has always appealed to me, so in 2013 when I started pursuing my bachelor degree in engineering, I was already thinking of eventually doing a PhD. At the time this seemed like a far-off and difficult-to-achieve goal and in the course of 5 years of studies it had faded to the back of my mind. The idea resurfaced when it became time to pick my master thesis topic. Knowing that the Photonics Research Group was actively looking for PhD students at the time, I was conscious of the fact that a master thesis might be the starting point of doing a PhD. By that time doing a PhD had lost much of the mystique it had held 5 years earlier and I had heard of and read about many wildly different experiences of PhD students. The point all these stories agreed upon was that one of the deciding factors in having a successful and enjoyable PhD track was having a good promotor. My choice of master thesis and my decision to eventually pursue a PhD was in no small part with this in mind and I am happy to report this turned out to be a course of action I never regretted.

My first acknowledgement thus goes out to the promotors of both my master thesis and PhD. Prof. Peter Bienstman, who from the first lecture of 'Wiskunde voor de fotonica' has been an excellent teacher and above all a kind person and my co-promotor, Prof. Joni Dambre, whose sharp insights were formative for my academic career.

A second deciding factor in having a successful and enjoyable PhD turned out to be my fellow PhD students. Alessio Lugnan started out as my supervisor for my master thesis, a task to which he showed commitment I greatly admire him for to this day, and continued to be a great colleague. Floris Laporte, Stijn Sackesyn and Chonghuai Ma who were tremendously welcoming and helpful during the early years of my PhD and whose work formed much of the basis of this thesis. Sarah Masaad, who joined our group after me but quickly became an indispensable partner in many tasks.

Laurens Bogaert, Nishant Singh and Joris Van Kerrebrouck, who often lent me their vast expertise and time in the field of electronics, for no other reason than that they are kind and helpful people.

Koen Alexander who helped me get started on being a teaching assistant for microphotonics despite being very busy with wrapping up his PhD and getting through visa issues.

Not those closest to me in research but often those closest to my in physical distance were my office mates. Artur Hermans, Tom Vanackere, Tom Sistermans, Valeria Bonito Oliva, Robbe De Prins, Emiel Dieussaert, Dennis Maes, Awanish Pandey and Abdul Rahim. The pandemic and its effects caused me to be less often in the office than I wanted, but I enjoyed every moment in your company.

Those that I only saw for a few weeks but who in that time helped greatly with performing some of the most challenging work done for this thesis, thank you Elger Vlieg and Alessandro Lupo.

And of course last mentioned but certainly not the least among my colleagues, Ewoud Vissers. Always capable and willing to help no matter the problem, be it research related or personal, and a great friend. This PhD would have been much diminished had he not sat down for the first time across from me more than 4 years ago.

And then there are all those others with whom I had lunch, conversations, movie nights, game nights, ... My thanks to all PhD students with whom I interacted during these years in the PRG, I have only positive memories of you all.

Other than my promotors and fellow PhD students I can of course not forget to mention the group's professors who served both as my lecturers throughout my master studies and as my fellow researchers during my PhD. I can attest all of them are thoughtful people readily willing to lend their insight both to their students and colleagues. In the context of my PhD I want to explicitly mention the following professors. Prof. Roel Baets, for whom I was a teaching assistant for several years. Both when studying under him and working with him he was the right mix of demanding and supportive. Prof. Wim Bogaerts, who at various points took the time to give me indispensable advice concerning the chip design I did and even went so far as spending significant time helping me get the design error free. Truly he went above and beyond for a PhD student for whom he was not even the promotor! And last but not least Prof. Yanlu Li who oversaw the fabrication runs I was a part of and without whose help neither me or my project partners would have gotten our chips.

Then there are of course all those without whom the group could not function. My thanks to the people taking care of the administration, the measurement lab, the cleanroom, the servers and all those other crucial components. Kristien, Ilse&Ilse, Bert, Clemens, Steven, Liesbeth,... As with my fellow PhD students I need to summarize. Thank you to all those without whose work this thesis would never have been finished. Your work is not always seen but always felt.

I would also like to thank all of my jury members, for taking the time to review

my manuscript. Especially I would like to thank Peter again. Your feedback was invaluable and I am humbled by the care and time you took in helping me write this thesis.

This thesis could not have happened without support from the home front. I would like to thank my mother without whom I would let so many small things slip through the cracks in life that eventually its structure would come down. My brother whose importance to me can not be expressed in words. My grandparents for their constant support. My best friends Thomas and Kirsten who greatly enrich my life in ways that all the money in the world could not.

Many others have shaped my life and work during and before this thesis, sometimes in ways I may not always be aware of. Thank you to all those people for all the small acts of kindness and behind the scenes support that keep me going and brought this work to completion!

Gent, February 2024
Emmanuel Gooskens

Contents

List of Acronyms

A

AI	artificial intelligence
AWG	arrayed waveguide grating
ASIC	application-specific integrated circuit

B

BER	bit error rate
BTO	barium titanate

C

CMOS	complementary metal–oxide–semiconductor
CWDM	coarse wavelength division multiplexing
CW	continuous wave

D

DC	directional coupler
DSP	digital signal processor

DWDM	dense wavelength division multiplexing

E

EDFA	erbium doped fiber amplifier

F

FEC	forward error correction
FSR	free spectral range

G

GC	grating coupler
GD	gradient descent
GPU	graphical processing unit

L

LASSO	least absolute shrinkage and selection operator

M

ML	Machine Learning
MMI	multimode interferometer
MPW	Multi-Project Wafer
MZI	Mach-Zehnder interferometer
MSE	mean squared error

MAC	multiply-accumulate
MVM	matrix-vector multiplication
MZM	Mach-Zehnder modulator

N

NN	neural network

O

OOK	on-off keying
OSA	optical spectrum analyzer

P

PCB	printed circuit board
PCM	phase change material
PRC	photonic reservoir computing
PRG	Photonic Research Group
PTP	peak-to-peak
PD	photodetector
PIC	photonic integrated circuit
PDK	process development kit

R

RC	reservoir computing
RNN	recurrent neural network
RTO	real time oscilloscope

S

SOA	semiconductor optical amplifier
SOI	silicon on insulator
SiN	silicon nitride
SMBO	Sequential Model-Based Global Optimization
SNR	signal-to-noise ratio
SPM	self-phase modulation

T

TDL	tapped delay line
TIA	transimpedance amplifier
TPE	Tree of Parzen Estimators

V

VOA	variable optical attenuators

W

WDM	wavelength division multiplexing

X

XPM	cross-phase modulation
XGM	cross-gain modulation

Samenvatting

1 Inleiding

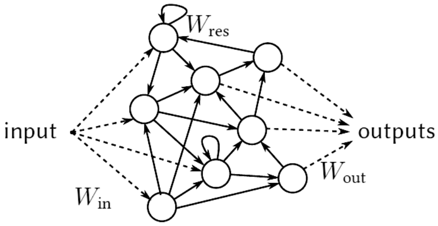
De vraag naar gegevensbandbreedte in huidige toepassingen zoals videostreaming is aanzienlijk en neemt alleen maar toe, omdat we oplossingen zoeken voor een groot aantal hedendaagse uitdagingen, zoals gepersonaliseerde gezondheidszorg, door steeds meer gegevens te verzamelen en te analyseren. Veel van die gegevens worden getransporteerd in het optische domein, dat wil zeggen via optische vezels. Een van de redenen hiervoor is dat verschillende golflengtes van licht onder conventionele omstandigheden niet met elkaar interageren, waardoor ze parallel kunnen worden verzonden. Dit heet multiplexing met golflengteverdeling. Hierdoor hebben optische vezels een enorm potentieel aan gegevensbandbreedte [1].

Bij de interfaces van de optische vezel ontstaan echter knelpunten. Hier worden gegevens omgezet van het optische domein naar het digitaal elektronisch domein. Naast de vertraging die deze omzetting met zich meebrengt, heeft het digitale elektronische domein te lijden van inherente vertraging en slechte energie-inefficiëntie, die voortkomen uit de voornamelijk sequentiële aard van de bewerkingen, en uit de elektrische verliezen. Traditionele op elektronica gebaseerde systemen bereiken hun fysieke grenzen door de beperkingen op de grootte van transistors en het opladen van elektrische verbindingen, wat significante verbeteringen voor deze problemen in de weg staat [2].

In dit proefschrift onderzoeken we fotonisch rekenen [3–6], als alternatief voor digitaal elektronisch computergebruik voor specifieke toepassingen. De voornaamste focus ligt op het mogelijk maken van multiplexen met golflengteverdeling voor fotonisch reservoir rekenen voor standaardbewerkingen op optische telecommunicatiekanalen, zoals signaalegalisatie. We kijken ook naar het mogelijk maken van multiplexen met golflengteverdeling in een fotonisch tensorkernsysteem voor matrix-vector vermenigvuldigingen (MVM). In beide gevallen kunnen we door multiplexen met golflengteverdeling in te schakelen, de systeemprestaties enorm verhogen zonder toename in chipoppervlakte, elektrische stuurcomplexiteit of energieverbruik.

Fotonisch reservoir rekenen in deze context is een systeem dat bestaat uit knooppunten die geconnecteerd zijn in fotonische hardware en daarbij een zogenaamd reservoir vormen. Dit reservoir is een dynamisch systeem dat een ingangssignaal door temporele en ruimtelijke menging transformeert in een interne reservoirtoestand. Een eenvoudige lineaire laag, genaamd de readout, die deze toestand interpreteert, wordt toegevoegd. De readout doet dit door het nemen van een lineaire combinatie van een subset van de knooppuntsignalen, zoals getoond in Fig. 1. De gewenste output kan worden bekomen door enkel het optimaliseren van de gewichten van deze lineaire combinatie.

Een fotonisch tensorkernsysteem aan de andere kant is een geïntegreerd optisch systeem dat specifiek is ontworpen om matrix-vector vermenigvuldigingen, elementaire bewerkingen die voorkomen in een verscheidenheid van machine-learning toepassingen, te versnellen.



Figuur 1: Schematische weergave van een reservoirrekensysteem. Het ingangssignaal wordt het reservoir ingestuurd. In het reservoir mengt het signaal zich met vorige versies van zichzelf vanwege de dynamische architectuur van het reservoir. Elke keer wordt een deel van de reservoirtoestand uitgelezen door de lineaire laag, met gewichten W_{out}, voor het produceren van een tijdsafhankelijke output. Figuur afkomstig uit [7].

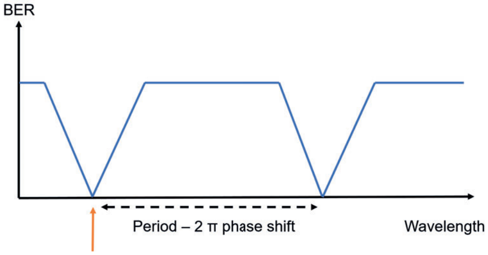
2 Resultaten

2.1 Meerdere golflengten in fotonisch reservoir rekenen

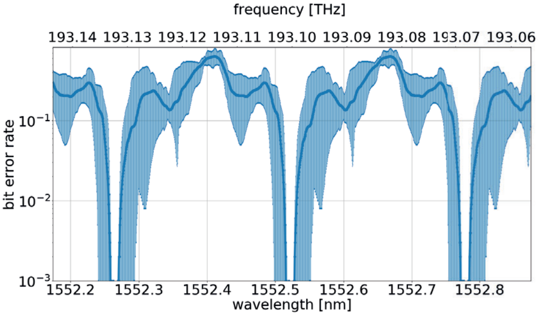
Een obstakel bij het gebruik van meerdere golflengten in fotonisch reservoir rekenen is dat, zonder bijkomende maatregelen, de performantie normaal gezien afneemt bij het variëren van de golflengte, vanwege de resulterende variatie in de faseinformatie. Dit leidt tot een gewijzigde signaalmenging waarvoor de readoutgewichten niet zijn getraind en dit leidt tot een incorrecte weging en recombinatie van knooppunt outputs. Een belangrijk aandachtspunt is dat we slechts één set gewichten gebruiken voor alle golflengten. Het hebben van een afzonderlijke set gewichten voor elke golflengte, zou de voordelen van multiplexen met golflengteverdeling teniet doen wat betreft compactheid. We simuleren geen echte golflengteverdeling in de zin dat we inter-golflengte interactie mechanismen niet in rekening brengen. Voor deze eerste stappen naar golflengteverdeling toe ligt de focus op het verkennen van hoe dit enkel-readout systeem kan gerealiseerd worden in dit vereenvoudigd scenario.

Vertrekkend van simulaties, hebben we twee aanpakken onderzocht om dit te bereiken in ons fotonisch reservoirrekensysteem. Hier presenteren we de resultaten specifiek voor de niet-lineaire signaalegalisatie taak, de commercieel meest relevante taak die we hebben onderzocht.

De eerste aanpak bestaat uit het ontwerpen van de verbindingslengten in het reservoir om periodiciteit in de faseinformatie te benutten. Door ervoor te zorgen dat verschillende golflengten dezelfde faseinformatie delen, wordt een enkel-readout systeem voor meerdere golflengten bereikt voor een groot aantal regelmatig gespreide golflengten (Figuren 2 en 3).

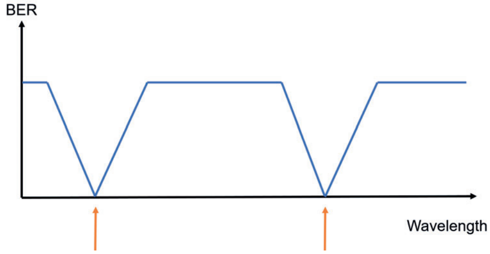


Figuur 2: Het benutten van ontworpen verbindingslengten. Training gebeurt enkel voor één golflengte, zoals aangeduid door de oranje pijl.

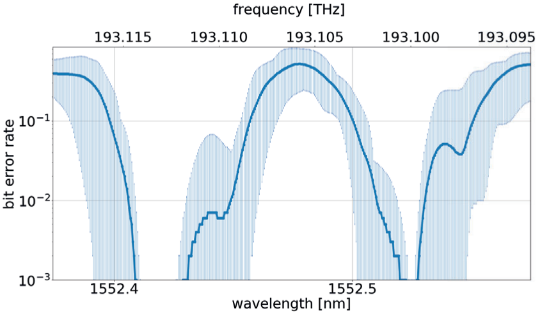


Figuur 3: Performantie bij het gebruik van ontworpen verbindingslengten voor de signaalegalisatie taak. Foutmarges duiden minimum en maximum behaalde bit error rate (BER) aan over 10 verschillende reservoirs elk met hun eigen fabricageafwijkingen.

De tweede aanpak bestaat uit het rekening houden met de beoogde multiplexing golflengtekanalen tijdens de training (Figuren 4 en 5). Dit maakt het trainen moeilijker en het aantal golflengtekanalen die de readout kan verwerken op deze manier is veel beperkter dan in de vorige aanpak. Echter, de mogelijke golflengtekanalen zijn niet langer afhankelijk van de specifieke uitvoering van het reservoir.

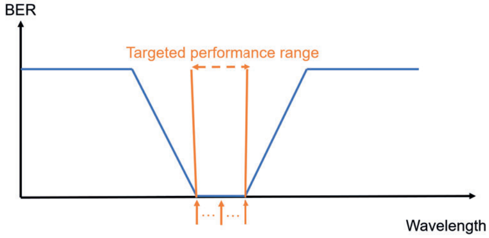


Figuur 4: Illustratie van meervoudige, in dit geval 2, golflengte-training. Training gebeurt enkel voor twee golflengten zoals aangeduid door de oranje pijlen.

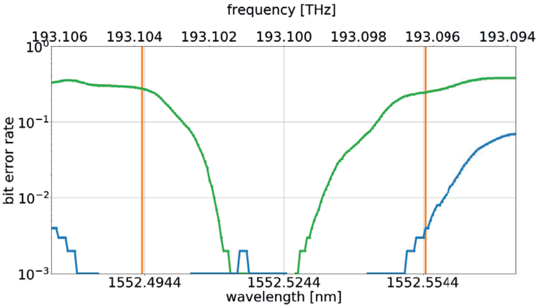


Figuur 5: Meervoudige golflengte-training voor de signaalegalisatie taak. Golflengtekanalen bereiken $< 10^{-3}$ *gemiddelde bit error rate (BER). Foutmarges duiden minimum en maximum bereikte (BER) aan over 10 verschillende reservoirs elk met hun eigen fabricageafwijkingen.*

Uit deze tweede aanpak ontstond het idee om gebruik te maken van meervoudige golflengte-training om het overigens zeer smalle golflengtebereik rond de beoogde golflengtekanalen waarover goede prestaties worden bereikt, te vergroten. Dit werd voornamelijk gedaan om de laserdrift van de ingangsgolflengte te verminderen, maar verbetert ook de robuustheid van de uitlezing van variaties in faseinformatie, door b.v. temperatuurvariatie. Voor de signaalegalisatie taak werd het goede prestatiebereik vergroot van 24,5 pm/3 GHz naar 95,4 pm/11,9 GHz. Dit is een aanzienlijke verbetering vergeleken met de ± 25 pm golflengtestabiliteit van de Menara networks 5ZR0A00-TNBL 50GHz C-band tunable transceiver, een relevante zendontvanger voor onze beoogde toepassingen (Figuren 6 en 7).

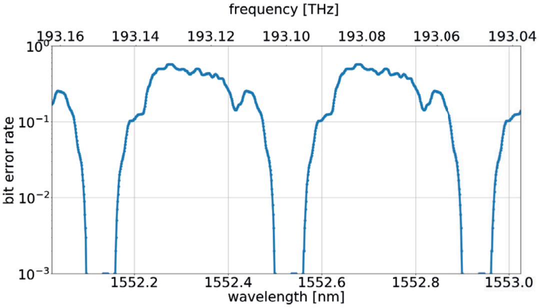


Figuur 6: Verbeteren van het golflengtebereik. De oranje pijlen duiden de golflengten aan die werden getraind. De oranje grenzen duiden het beoogde golflengtebereik, over dewelke we goede performantie wensen te bekomen, aan.

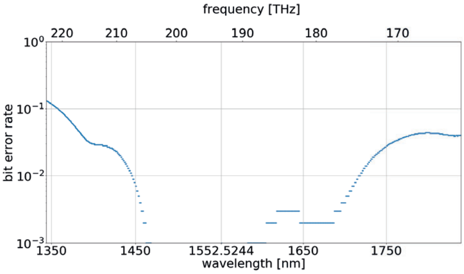


Figuur 7: Performantie voor de signaalegalisatie taak voor 5 getrainde golflengten. De 5 golflengten bestaan uit het centrum en de uitersten van het beoogde golflengtebereik en twee tussenliggende golflengten. Groen duidt 1-golflengte-training aan en blauw 5-golflengte-training. De oranje grenzen duiden het beoogde golflengtebereik, over dewelke we goede performantie wensen te bekomen, aan.

Door deze aanpakken te combineren, hebben we laten zien hoe door het gebruik van ontworpen verbindingslengten in combinatie met meervoudige golflengte-training voor het verbeteren van het golflengtebereik, we robuuste enkel-readout werking over vele kanalen kunnen bekomen (Figuren 8 en 9).



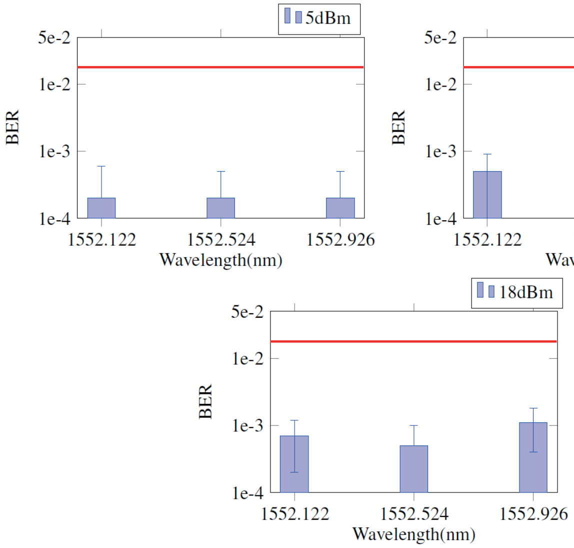
Figuur 8: Robuuste enkel-readout werking voor de niet-lineaire signaalegalisatie taak voor één enkel reservoir. Andere reservoirs tonen gelijkaardige kenmerken met enige variatie in de breedte van het golflengtebereik met goede prestaties. Allen bereiken $< 10^{-3}$ *BER voor de doelgolflengten.*



Figuur 9: BER voor doelgolflengten (BER buiten de doel golflengten wordt niet weergegeven) toont robuuste enkel-readout werking voor de niet-lineaire signaalegalisatie taak en het reservoir van de vorige figuur. Voor zowel dit en alle andere reservoirs heeft dispersie enkel aanzienlijke invloed op de performantie na >1 THz verwijderd van de doelgolflengten. Dit komt overeen met vele tientallen tot honderden golflengtekanalen.

Dispersie beperkt uiteindelijk het aantal WDM-kanalen waarvoor goede prestaties worden behaald. In de praktijk blijkt dat we makkelijk kunnen voorzien in tientallen tot honderden standaard ITU-T kanalen [8].

Na deze successen hebben we experimenteel een eerste belangrijke stap gezet richting multiplexen met golflengteverdeling in een geïntegreerd, op golfgeleiders gebaseerd fotonische reservoirrekenplatform dat slechts beperkte nabewerking gebruikt, dat wil zeggen een digitaal geïmplementeerde readout en het genereren van vertraagde kopieën van gemeten knooppuntsignalen. Wij hebben gebruik gemaakt van de meervoudige golflengte-training op reeds gefabriceerde chips. De $1.5\,10^{-2}$ forward error correction (FEC)-limiet van het P1-PTX-2-100G-WDM optische transportsysteem [9], een relevant systeem voor onze doel applicaties, was onze benchmark. We genereerden een vervormd signaal door een 28 Gbps signaal doorheen een glasvezel van 25 km te sturen. Het ingangsvermogen van de vezel bepaalt de hoeveelheid niet-lineaire vervorming die het signaal ondergaat en daarmee de moeilijkheid van de egalisatietaak. Het systeem met enkele readout bleek goed te werken tot ten minste 3 standaard ITU-T golflengten voor (niet-lineaire) signaalegalisatie taken met verschillende moeilijkheidsgraden, zoals weergegeven in Fig. 10.



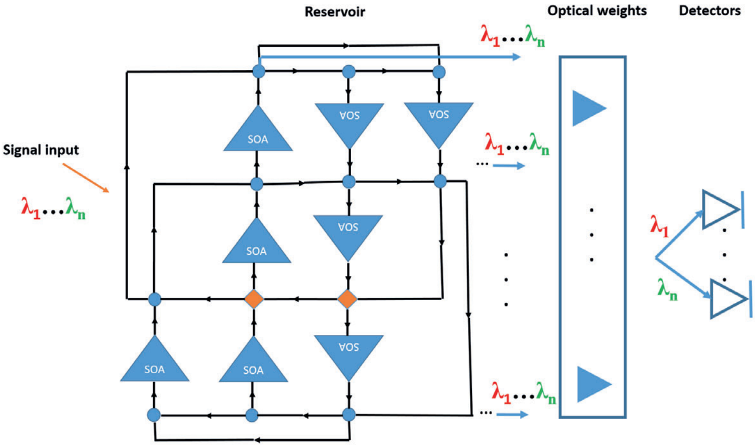
Figuur 10: Gemiddelde BER, met foutmarges die de standaardafwijking aangeven over de kruisvalidatie-iteraties voor training met meerdere golflengten. Er werd rekening gehouden met alle weergegeven golflengten voor de optimalisatie van de tijdsverschuiving en de readout training. Het ingangsvermogen van de glasvezel was 5 dBm, 10 dBm en 18 dBm. De horizontale rode lijn illustreert de relevante FEC-limiet.

We hebben ook aangetoond dat het reservoir beter presteert dan een eenvoudig getapt vertragingslijnfilter voor hogere niveaus van signaalvervorming en dit bevestigt het vermogen van het systeem om niet-lineaire taken aan te pakken.

Daarnaast hebben we voor het eerst experimenten uitgevoerd voor online signaalegalisatie met een optische geïntegreerde readout, zij het alleen voor het geval met één golflengte. We lieten een verbetering zien met een factor ≈ 3 in BER vergeleken met het geval van signaalegalisatie zonder reservoir.

2.2 Integratie van niet-lineaire componenten in het fotonische reservoir voor verbeterde niet-lineariteit

Met als doel meer niet-lineariteit toe te voegen aan een fotonisch reservoirrekensysteem door toevoeging van niet-lineaire componenten in het reservoir, werd de integratie van halfgeleider optische versterkers in het reservoir bestudeerd (Fig. 11).



Figuur 11: Schema van het reservoir met halfgeleider optische versterkers. Signaalinvoer wordt geïnjecteerd in de oranje ruitvormige knooppunten. Alle knooppunten zijn verbonden met de readout. Pijlen geven de voortplantingsrichting van het signaal doorheen het reservoir aan.

Er werd een geïntegreerd reservoir met halfgeleider optische versterkers vervaardigd, de eerste karakteriseringsstappen werden uitgevoerd en enkele problemen die een grondige analyse van deze chip zouden bemoeilijken werden vastgesteld. De voornaamste zorg was de stroomcapaciteit van de wirebonds, rekening houdend met de hoge stroom eisen van de halfgeleider optische versterkers om voldoende optische versterking te bereiken.

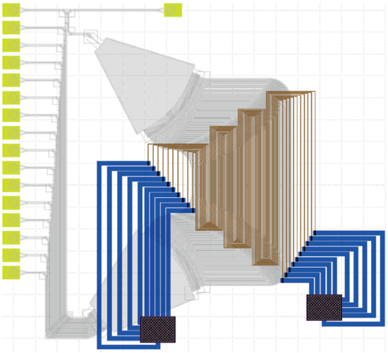
2.3 Arrayed waveguide grating demultiplexer

Een arrayed waveguide grating (AWG) demultiplexer is ontworpen om golflengte multiplexering mogelijk te maken in een geïntegreerde fotonische tensorkern. De (ambitieuze) eisen waren voldoende voldaan bij het eerste generatie ontwerp (Tab. 1).

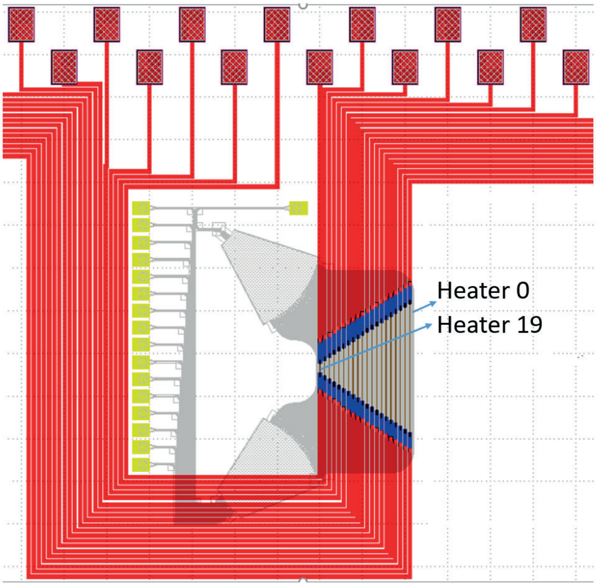
Vereisten	Simulaties	Metingen	Conclusies
#kanalen=16	16	16	OK
FSR>16x500GHz	>16x500GHz	>16x500GHz	OK
Kanaal spacing=500GHz	500GHz	500GHz	OK
Piek afwijking<<1nm	<1nm	1-10nm	actieve afstemming nodig
Kanaal breedte=480GHz met flat top respons en steile roll-off	Imperfect, next-neighbor crosstalk ≈ 4.5dB	Imperfect, next-neighbor crosstalk ≈ 4.5dB	moeilijk om ideale specificaties te bereiken
Insertion loss	<4dB	<15dB	acceptabel
Compact design	790x873 micron	790x873 micron	voldoende compact

Tabel 1: AWG design vereisten versus gesimuleerde en gemeten resultaten

Zoals blijkt uit de gemeten piekafwijkingen, die voortkomen uit fabricagefouten, is extra afstemming van het golflengtekanaal vereist, maar dit werd verwacht. De ontwerpen van de eerste generatie hadden twee verschillende types thermische afstemverwarmers: een type met één verwarmingselement en een type met meerdere verwarmingselementen (Figuren 12 en 13).

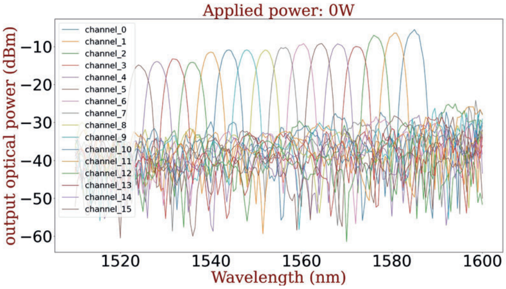


Figuur 12: Blauwe kleur geeft metalen lijnen aan. De bruine kleur duidt op een wolfraamverwarmer. Ontwerp met één verwarmingselement. De kronkelende wolfraamverwarmingslijnen delen hetzelfde aardings- en signaalverbindingspad en fungeren dus als één groot verwarmingselement.

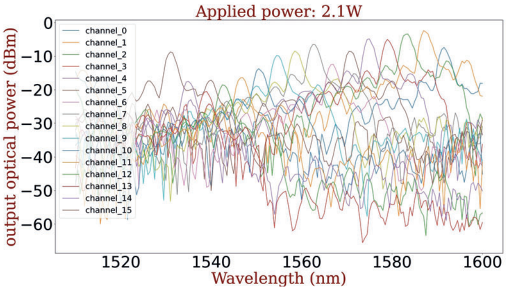


Figuur 13: Blauwe en rode kleuren geven metalen lijnen aan op verschillende metaalrouteringslagen. De bruine kleur duidt op een wolfraamverwarmer. Ontwerp met meerdere verwarmingselementen. Elke rechte verwarmingslijn heeft een eigen verbindingspad (deze zijn niet allemaal afgebeeld) en kan dus onafhankelijk van de andere verwarmingslijnen worden aangestuurd.

Voor het ontwerp met één verwarmer hebben we bereikt dat golflengtekanalen naar grotere golflengten verschuiven met $\approx 2 - 3$ nm per Watt aan energieverbruik. Dit is bemoedigend, maar verdere verbeteringen op de verwarmingsuniformiteit zijn noodzakelijk (Figuren 14 en 15).



Figuur 14: De AWG kanaal spectra wanneer er geen verwarming wordt toegepast.



Figuur 15: De AWG kanaal spectra voor thermische afstemming met 2.1W energieverbruik voor het ontwerp met één verwarmingselement. De performantie met de verwarming is zichtbaar gedegradeerd. Dit is door onvoldoende verwarmingsuniformiteit.

Voor het ontwerp met meerdere verwarmingselementen was de verwarmingsuniformiteit ook slecht, hoewel verder onderzoek naar de stroom distributieconfiguraties nog verbeteringen kan opleveren.

Om de verwarmingsuniformiteit te verbeteren, hebben we zojuist een nieuwe generatie ontwerpen afgerond, namelijk een verbeterd ontwerp met één verwarmingselement en een ontwerp met hybride verwarmingselementen.

3 Conclusie

Het voornaamste doel van dit werk was om ons bestaand fotonisch reservoirrekensysteem te verbeteren door gebruik te maken van de golflengtedimensie. Er werden belangrijke eerste stappen gezet richting fotonisch reservoirrekensystemen met volledig geïntegreerde optische readouts, geschikt voor signaalegalisatie voor signalen met golflengteverdeling multiplexing. We hebben ook systemen onderzocht met verbeterde niet-lineariteit.

Ten slotte werd een AWG-demultiplexer ontworpen om multiplexen met golflengteverdeling in een fotonisch tensorkernsysteem mogelijk te maken. Zo een demultiplexer zou ook een rol kunnen spelen in ons toekomstig fotonische reservoirrekensysteem.

Summary

1 Introduction

The demand for data bandwidth in current applications such as video streaming is significant and is only growing, as we seek out solutions to a myriad of contemporary challenges, e.g. personalised healthcare, through ever greater collecting and analysing of data. A lot of that data is transported in the optical domain, that is to say by optical fiber. One reason for this is that different wavelengths of lights do, under conventional circumstances, not interact with each other, which allows their transmission in parallel, using so called wavelength division multiplexing (WDM) signals. This enables optical fiber to have an enormous data bandwidth potential [1].

However, at the interfaces of the optical fiber bottlenecks arise. Here, data is converted from the optical domain to the digital electronics domain. In addition to the latency imposed by this conversion, the digital electronic domain suffers from inherent latency and power inefficiency, stemming from the mainly sequential nature of its operations and electrical losses respectively. Traditional electronics-based systems are reaching physical limits due to transistor size limitations and charging of electrical lines, which impedes any significant improvements on these issues [2].

In this thesis, we investigate photonic computing [3–6], as an alternative to digital electronic computing for specific use cases. The main focus is on enabling WDM in photonic reservoir computing (PRC) for standard operations on optical telecommunication channels, such as signal equalization. We also look at enabling WDM in a photonic tensor core system for matrix-vector multiplication (MVM). In both cases, enabling WDM allows us to greatly increase system performance without increases in chip footprint, electrical steering complexity or power consumption.

Photonic reservoir computing in this context is a system that consists of nodes that are connected in photonics hardware, forming a so-called reservoir. This reservoir is a dynamical system that transforms an input signal through temporal and spatial mixing to an internal reservoir state. A simple linear layer, referred to as the readout, is added, which interprets this state. The readout does so by taking a linear combination of a subset of the node signals as shown in Fig. 16. The desired output can be obtained by only optimizing the weights of this linear combination.

A photonic tensor core on the other hand is an integrated optical system that is specifically designed to speed up matrix-vector multiplications, which is a primitive that occurs in a variety of machine-learning applications.

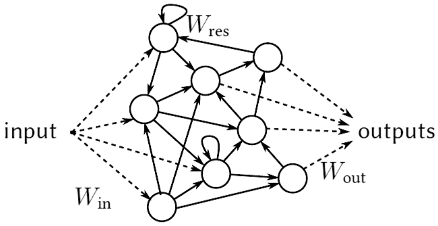


Figure 16: Schematic representation of a reservoir computing system. The input signal is fed into the reservoir. Inside the reservoir, the signal mixes with previous versions of itself due to the dynamic architecture of the reservoir. At each time, (a part of) the reservoir state is read out by the linear layer, with weights W_{out}, to produce a time dependent output. Figure taken from [7].

2 Results

2.1 Multi-wavelength photonic reservoir computing

An obstacle to using multi-wavelength photonic reservoir computing is that, without additional measures, the performance normally goes down when varying the wavelength, because of the resulting variation in reservoir phase information. This leads to altered signal mixing for which the readout weights were not trained, leading to an incorrect weighting and recombination of node outputs. A main point of note is that we will only use a single set of weights for all wavelengths. Indeed, having a separate set of weights for every wavelength would negate the benefits of WDM with respect to compactness. We do not simulate true WDM in that we do not account for inter-wavelength crosstalk mechanisms. For these first steps towards WDM the focus is on exploring how this single-readout system may be achieved in this simplified scenario.

Starting from simulations, we explored two approaches to achieving this in our PRC system. Here we present the results specifically for the nonlinear signal equalization task, which is the most commercially relevant task we investigated.

The first approach consists of engineering the interconnection lengths in the reservoir to exploit phase periodicity in the reservoir phase information. By ensuring several wavelengths share the same phase information, a single-readout system handling multiple wavelengths is achieved at a large number of regularly spaced wavelength channels (Figs. 17 and 18).

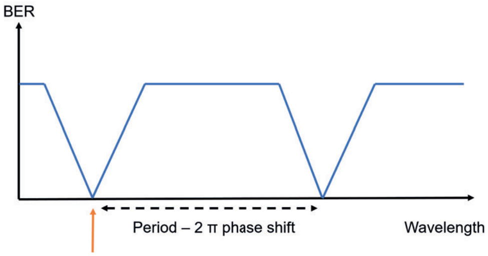


Figure 17: Exploiting engineered interconnection length. Training only occurs for one wavelength, as indicated by the orange arrow.

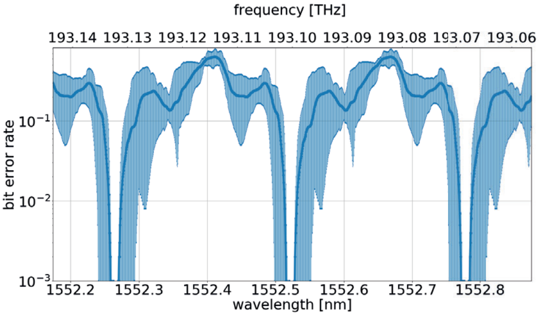


Figure 18: Performance when using engineered interconnection lengths for the signal equalization task. Error bars indicate minimum and maximum achieved bit error rate (BER) over 10 different reservoirs each with their own fabrication errors.

The second approach consists of taking into account the targeted WDM wavelength channels during training (Figs. 19 and 20). This makes training more difficult and the number of wavelength channels the readout can handle in this way is much more limited than that of the previous approach. However, the possible WDM channels no longer depend on the specific implementation of the reservoir.

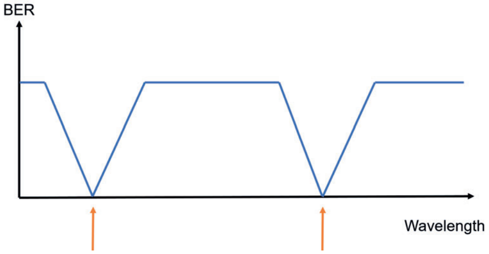


Figure 19: Illustrating multiple, two in this case, wavelength channel training. Training occurs at two wavelengths as indicated by the orange arrows.

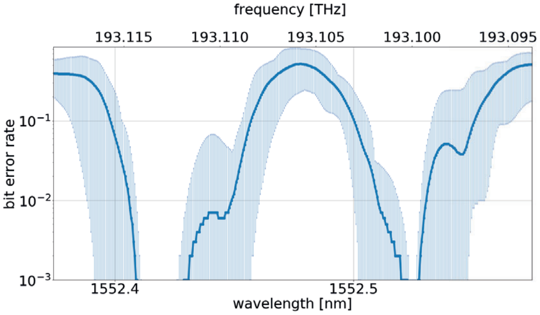


Figure 20: Multiple-wavelength channel training for signal equalization task. Wavelength channels achieve $< 10^{-3}$ mean BER. Error bars indicate minimum and maximum achieved BER for 10 different reservoirs with their own simulated fabrication errors.

From this second approach evolved the idea of using multiple wavelength training to enlarge the otherwise very narrow wavelength range around the targeted wavelength channels over which good performance is achieved. This was done mainly for mitigating input wavelength laser drift, but also improves the robustness of the readout to variations in phase information, by e.g. temperature variation. For the signal equalization task the good performance range was increased from 24.5 pm/3 GHz to 95.4 pm/11.9 GHz. This is a significant improvement when compared to the ± 25 pm wavelength stability of the Menara networks 5ZR0A00-TNBL 50GHz C-band tunable transceiver, which is a relevant transceiver for our targeted application (Figs. 21 and 22).

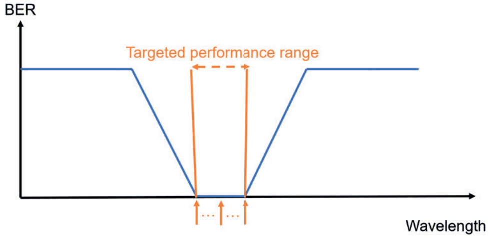


Figure 21: Improving good performance wavelength range. The orange arrows indicate the wavelengths that are trained. The orange boundaries indicate the targeted wavelength range over which to achieve good performance.

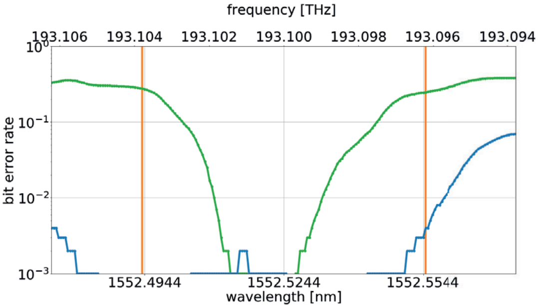


Figure 22: Performance for the signal equalization task for 5 training wavelengths. The 5 wavelengths consisted of the center and extremes of the targeted range and two intermediary wavelengths. Green indicates 1-wavelength-training and blue 5-wavelength-training. The orange boundaries indicate the targeted wavelength range over which to achieve good performance.

Combining these approaches, we showed how by making use of engineered interconnection lengths in tandem with the multiple wavelength technique to improve the good performance wavelength range, we can achieve robust single-readout operation over many channels (Figs. 23 and 24).

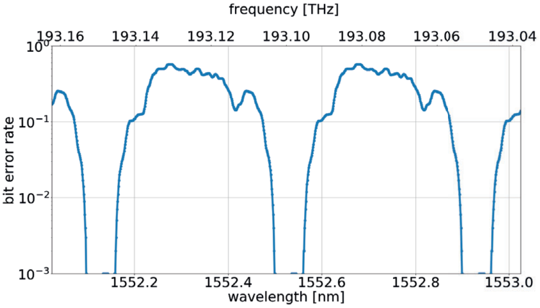


Figure 23: Robust single-readout operation performance for the nonlinear signal equalization task for one particular reservoir. Other reservoirs showed similar characteristics with some variation in the width of the good performance wavelength range. All achieve $< 10^{-3}$ *BER at target wavelengths.*

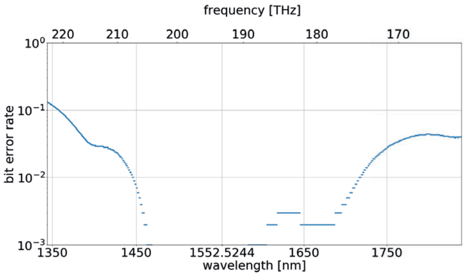


Figure 24: BER at target channels indicates robust single-readout operation (BER outside of the target channels not plotted) for the nonlinear signal equalization task and reservoir of 23. For both this case and all other reservoirs, dispersion only significantly impacts performance after >1 THz wavelength spacing away from the designed wavelength channel. This corresponds to many tens to hundreds of dense wavelength division multiplexing (DWDM) channels.

Dispersion ultimately limits the number of WDM channels for which good performance can be achieved. In practice, it turns out that we can easily accommodate tens to hundreds of standard-spaced ITU-T channels [8].

Following these successes, we have experimentally taken a first important step towards WDM in an integrated waveguide-based photonic reservoir computing platform using only limited post processing i.e. a digitally implemented readout and the generating of delayed copies of measured node signals. We made use of the multiple wavelength training approach on already fabricated chips. The $1.5\,10^{-2}$ forward error correction (FEC) limit of the P1-PTX-2-100G-WDM optical transport system [9], a relevant system for our targeted applications, was our benchmark. We generated a distorted signal by sending a 28 Gbps signal through a 25 km fiber. The fiber input power determines the amount of nonlinear distortion the signal underwent and thus the difficulty of the equalization task. The single-readout system was shown to work well up to at least 3 standard ITU-T wavelengths for (nonlinear) signal equalization tasks of various difficulty, as shown in Fig. 25.

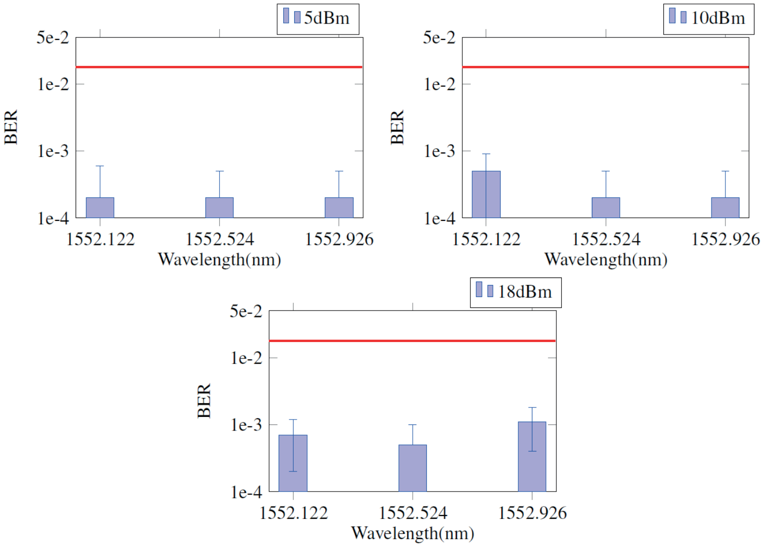


Figure 25: Average BER, with error bars indicating standard deviation over the cross validation iterations for multiple wavelength training. All wavelengths displayed were taken into account for time offset optimization and readout weight training. Fiber input powers were 5 dBm, 10 dBm and 18 dBm. The horizontal red line illustrates the relevant FEC limit.

We also showed that the reservoir outperforms a simple tapped delay line filter for higher levels of signal distortion and this confirms the capacity of the system to tackle nonlinear tasks.

In addition we performed, for the first time, experiments for online signal equalization with an optical integrated readout, albeit only for the single-wavelength case. We showed an improvement by a factor ≈ 3 in BER compared to the case of signal equalization without the reservoir.

2.2 Integrating nonlinear components in the photonic reservoir for enhanced nonlinearity

With the goal of adding more nonlinearity to a PRC system by inclusion of nonlinear components in the reservoir, the integration of semiconductor optical amplifier (SOA)s in the reservoir was studied (Fig. 26).

An integrated reservoir with SOAs was fabricated, initial characterization steps were

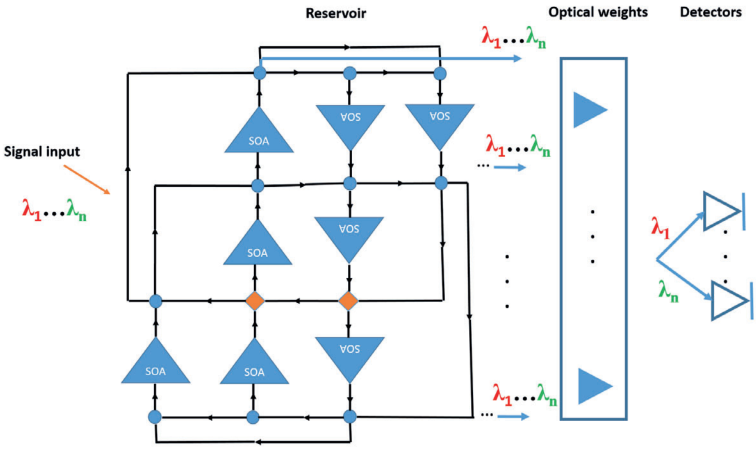


Figure 26: Schematic of the SOA reservoir. Signal input is injected in the orange diamond-shaped nodes. All nodes are connected to the readout. Arrows indicate propagation direction of the signal throughout the reservoir.

performed and some of the concerns that would complicate a thorough analysis of this chip were identified. The main concern was wirebond current capacity, considering the high current requirements of the SOAs in order to achieve sufficient optical amplification.

2.3 Arrayed waveguide grating demultiplexer

An arrayed waveguide grating (AWG) demultiplexer was designed to enable WDM in an integrated photonic tensor core. The (ambitious) requirements were adequately met by first generation designs (Tab. 2).

Requirements	Simulations	Measurements	Conclusion
#channels=16	16	16	OK
FSR>16x500GHz	>16x500GHz	>16x500GHz	OK
Channel spacing=500GHz	500GHz	500GHz	OK
Peak deviation<<1nm	<1nm	1-10nm	active tuning needed
Channel width=480GHz with flat top response and steep roll-off	Imperfect, next-neighbor crosstalk $\approx$ 4.5dB	Imperfect, next-neighbor crosstalk $\approx$ 4.5dB	challenging to obtain ideal specification
Insertion loss	<4dB	<15dB	acceptable
Compact design	790x873 micron	790x873 micron	sufficiently compact

Table 2: AWG design requirements versus simulated and measured results

As shown by the measured peak deviations, which originate from fabrication errors, additional wavelength channel tuning is required but this was anticipated. First-generation designs had two distinct thermal wavelength channel tuning heater designs, namely single heater and multiple heaters (Figs. 27 and 28).

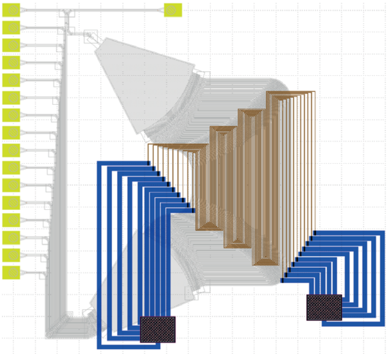


Figure 27: Blue color indicates metal lines. Brown color indicates tungsten heater. Single-heater design. The meandering tungsten heater lines share the same ground and signal bondpad and thus act as one big heater element.

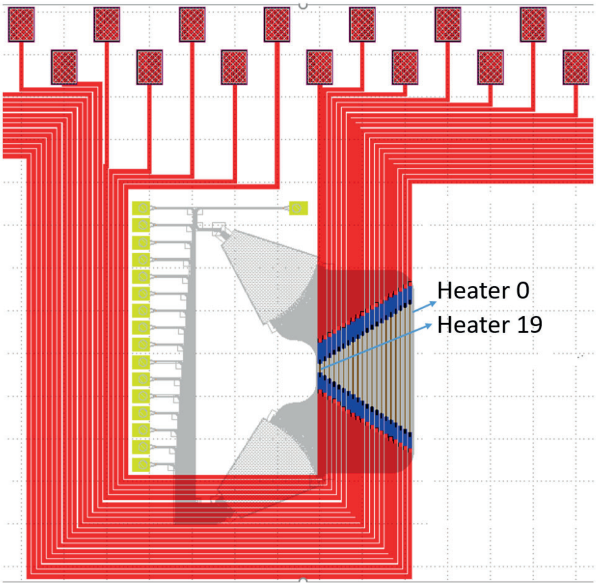


Figure 28: Blue and red colors indicate metal lines at different metal routing layers. Brown color indicates tungsten heater. Multiple-heater design. Every straight heater line has its own bondpad (not all of which are shown) and can thus be steered independently from the other heater lines.

For the single-heater design, we achieved wavelength channels shift to larger wavelengths with $\approx 2-3$ nm per Watt of power dissipation. This is encouraging, but further improvements on heater uniformity are necessary (Figs. 29 and 30).

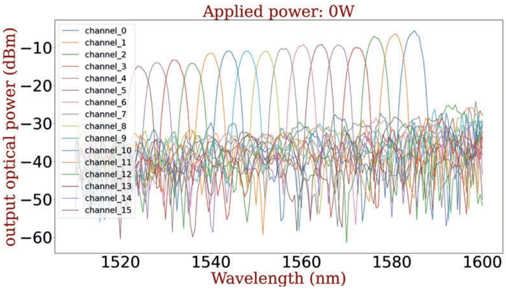


Figure 29: The AWG channels spectra when no heating is applied.

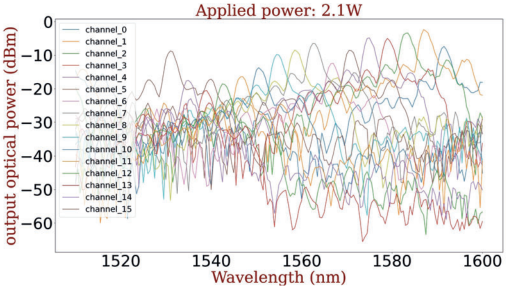


Figure 30: The AWG channels spectra for thermal tuning with 2.1W of power dissipation when using the single heater design. The performance when heating is visibly degraded this is due to insufficient heating uniformity.

For the multiple-heater design, heating uniformity was also poor, although further research on current distribution configurations may yet bring improvements.

To improve heating uniformity, we just taped out two new second-generation designs, namely an improved single-heater design and a hybrid-heater design.

3 Conclusion

The main goal of this work was to improve upon our existing PRC system by exploiting the wavelength dimension. Important first steps were taken towards PRC systems with fully integrated optical readouts, capable of doing WDM signal equalization. We also explored systems with enhanced nonlinearity.

Finally, an AWG demultiplexer was designed to enable WDM in a photonic tensor core. Such a demultiplexer could also have a role in our future fully realized WDM capable PRC systems.

1

Introduction

This chapter seeks to put this PhD topic in its proper context by explaining why improving wavelength division multiplexed optical telecommunications is a highly relevant scientific challenge and how it can be achieved through optical processing, more specifically photonic reservoir computing (PRC). This introductory chapter further lists the structure of the thesis, its various contributions and the corresponding scientific publications.

1.1 Wavelength division multiplexed optical telecommunications

A multitude of contemporary commercial applications and industrial processes have developed a great demand for data bandwidth for both communication and processing. Examples include not only the highly visible increase in video streaming for both entertainment and communication purposes, but also the increasing digitisation of many commercial and industrial sectors such as agriculture, manufacturing and healthcare. This demand for data bandwidth for both communication and processing is growing, as we seek out solutions to a myriad of contemporary challenges, e.g. personalised healthcare or predicting consumer behavior through ever greater collecting and analysing of data.

”Between 2018 and 2022, global data bandwidth nearly tripled with an increase of 28% in 2022 alone, reaching an estimated total by the end of 2022 just shy of 1 Pbps” [10]. A lot of that data is transported in the optical domain, that is to say by optical fiber. Optical fiber has some important advantages over other data transmission options, advantages that are rooted in the physical properties of light. Among the most important of these advantages is that different wavelengths of lights do, under conventional circumstances, not interact with each other, which allows their transmission in parallel, using so called wavelength division multiplexing (WDM) signals. This enables optical fiber to have an enormous data bandwidth potential [1].

1.2 Optical processing

While it is true that optical fiber has an enormous data bandwidth potential, at the interfaces of the optical fiber, bottlenecks arise. Here, data is converted from the optical domain to the digital electronics domain, that is to say bits on a computer, and the speed of this operation is limited by e.g. the detector. Additionally, the digital electronic domain suffers from inherent latency and power inefficiency, stemming from the mainly sequential nature of its operations and electrical losses respectively. Traditional electronics-based systems are reaching physical limits due to transistor size limitations and charging of electrical lines, which impedes any significant improvements on these issues [2].

It is currently still unthinkable to completely replace modern digital electronic computing, which is well-established and offers great flexibility in serving all sorts of applications. Still, there is increasing interest in alternative computing paradigms such as quantum computing or photonics-based computation/optical processing. Photonics-based hardware approaches offer several advantages, in particular high data bandwidth, especially when employing WDM, and this makes these approaches attractive alternatives. There are many potential photonics-based hardware implementations for data processing, such as those focusing on matrix multiplication, which is of interest for convolutional processing and deep learning applications [6, 11, 12]. We are however mainly concerned with photonic reservoir computing [3–5], as an alternative to digital electronic computing for the use case of standard operations on WDM optical telecommunication channels, such as signal equalization (Fig. 1.1) and header recognition.

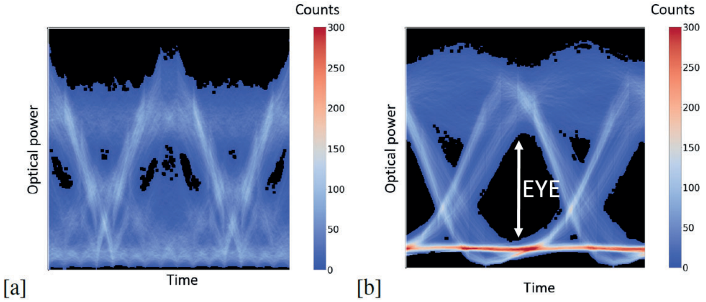


Figure 1.1: The eye diagrams here are constructed by dividing a simulated OOK signal in time segments of 3 bit periods long and folding these segments over each other. For each pixel in the eye diagram, the color bar represents the number of times a time trace passed through it. (a) Eye diagram distorted signal after 2000km long dispersion- and loss-compensated optical fiber with Kerr nonlinearity. (b) Equalized signal. The opening of the eye in an eye diagram gives a measure of how easy it is to discern between 0 and 1 bits in this OOK signal.

Traditionally, these operations are carried out by converting light into an electrical signal, which is then electronically processed by a digital signal processor (DSP) and subsequently converted back into the optical domain if needed. Apart from the obvious power considerations related to this transformation to the electrical domain and back, there are also the aforementioned optical-electrical interface latencies and fundamental scaling issues for electronic digital processing that limit the DSP technology to signals well below the limits of optical communication, even when not exploiting the potential of WDM. Indeed, digital signals in the electrical domain become notably difficult to process beyond 20 Gbps, while single-wavelength optical data transmission rates easily reach 100 Gbps or more [7, 13]. With PRC, the processing itself takes place in the optical domain, thus solving these issues.

The eventual end goal of our research on PRC, of which this thesis is only a part, would be to have PRC systems capable of optically processing the data of a wide variety of links in the telecom/datacom industries at speeds well above 64 Gbps for tens of WDM channels in parallel, with bit error rates below the forward error correction (FEC) limit of commercial transceivers such as the P1-PTX-2-100G-WDM optical transport system with a FEC limit of $1.5\ 10^{-2}$ [9].

In the long term we could scale up our WDM PRC systems to Tbaud/s which is orders of magnitudes better than traditional digital signal processing approaches. Our current chip operations however were deliberately slowed down as we need to be able to read them out using our available measurement equipment.

The method through which the processing is carried out, reservoir computing, originates in Machine Learning (ML), more specifically in neuromorphic-inspired neural networks, as we will now discuss.

1.3 Photonic reservoir computing

ML is a field of computer science in which computer systems are enabled to learn without being explicitly programmed [14, 15]. The computer systems are instead fed a large amount of inputs and corresponding desired outputs. Using a suitable ML algorithm, they are configured to solve a certain problem. This approach enables solving problems for which no good conventional rule-based approaches are available. A neural network (NN) is a ML system which consists of interconnected nodes, analogous to neurons in the brain (Fig. 1.2). Each node takes a weighted sum of its inputs and sends it typically through a nonlinear function, e.g. a sigmoid, before passing it on. By adapting the weights of the linear combinations according to the ML algorithm, the output of the NN can be matched to the desired output associated with the input.

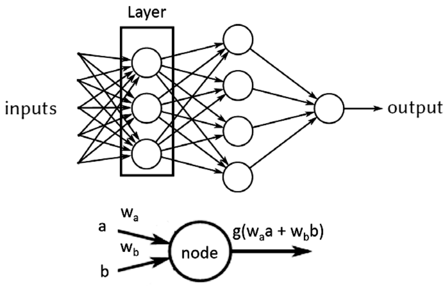


Figure 1.2: Top figure shows a feedforward neural network, i.e. a NN where information flows unidirectionally through successive layers of nodes. Figure taken from [7] with adjustments. Bottom figure illustrates how a node transforms a weighted sum of its inputs through a nonlinear function g().

A NN where the connections between nodes form cycles, thus preserving inside of their internal states a nonlinear transformation of the input history, is called a recurrent neural network (RNN) (Fig. 1.3). Such a RNN is a dynamical system with memory, allowing it to process temporal information, which makes it more suited for tasks such as pattern prediction. However, such networks are often very difficult to train.

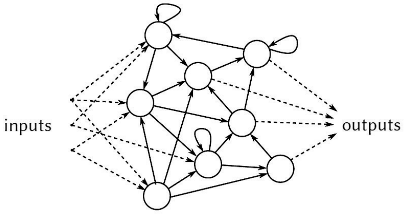


Figure 1.3: Recurrent neural network. Figure taken from [7].

To address this issue, a related approach is reservoir computing (RC), which employs a randomly initialised fixed RNN, called the reservoir, which is left untrained and to which a simple linear layer is added (Fig. 1.4). Only this linear layer, with weights W_{out}, is trained, greatly facilitating the practical application of RNNs for hardware implementations, as only the weights in the final layer need to be configurable and there is a greater tolerance to fabrication errors in the reservoir itself [3–5, 16, 17].

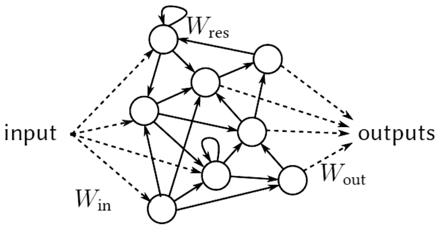


Figure 1.4: Schematic representation of a reservoir computing system. The input signal is fed into the reservoir. Inside the reservoir, the signal mixes with previous versions of itself due to the dynamic architecture of the reservoir. At each time, (a part of) the reservoir state is read out by the linear layer, with weights W_{out}, to produce a time dependent output. Figure taken from [7].

This brings us to PRC, which refers to the implementation of a reservoir computing system on a photonics hardware platform, combining the increased data bandwidth potential of optical processing (see section 1.2) with the flexible ML-based computing and hardware suitability offered by RC. There are many potential PRC implementations. Among those investigated are systems consisting of a single non-linear node with feedback and free-space reservoir systems [18–37]. In this thesis however, the explored hardware platform is an implementation of multiple spatially distributed nodes, with the interconnections consisting of waveguides and the nodes themselves of photonic components, in particular multimode interferometer (MMI)s, all integrated on-chip [16, 38–42]. Having the hardware implementation integrated on-chip has the advantages of being compact, stable, fast and cost-efficient.

1.4 Objectives

The proposed PRC system implementation, consisting of spatially distributed nodes integrated on-chip, has been proven to perform well for various tasks such as bit-level tasks, nonlinear dispersion compensation and isolated spoken digit recognition [38–40, 43]. However, the footprint of the used reservoirs is typically on the order of one to a few tens of mm^2. This large footprint translates into added cost, which negatively impacts economic viability.

The main scientific objective of this work is to improve upon this existing PRC implementation by, for the first time in an integrated system, exploiting the wavelength dimension, in particular making use of WDM. This will enable us to achieve high system performance, particularly in terms of data bandwidth, without increasing the silicon chip footprint. This in turn will allow us to tackle relevant industrial problems, most notably in optical telecommunication, in a cost-efficient way.

1.5 Thesis outline

The PRC paradigm, on which this thesis is based, sits at the intersection between machine learning and (silicon) photonics. Both fields are extremely broad and their combination is still relatively novel. Therefore, introductions covering all essential information and terminology in both domains will be given in chapters 2 and 3.

In chapter 4, we will explain in detail the main photonic reservoir design and the various ways in which to implement the reservoir weights. In chapters 5 and 6, the main work undertaken for this thesis will be discussed. These chapters cover simulation and experimental work respectively, both based on the design outlined in chapter 4, focusing on the exploitation of WDM in PRC, which is the most promising path to fulfilling the research objective stated in 1.4.

In chapter 7, the benefits of increased reservoir nonlinearity is briefly explored.

This is done by studying the integration of semiconductor optical amplifier (SOA)s in the reservoir.

Finally, chapter 8 deals with work undertaken for a European research project in which the author got involved. It concerns WDM in another form of photonic neuromorphic computing (a matrix-vector multiplication accelerator) and helps place this thesis in the broader context of novel photonic neuromorphic computing paradigms, highlighting again the important role of the wavelength dimension.

1.6 Attribution of work

The work undertaken for this PhD thesis builds heavily upon the work of past and contemporary PhD students which worked on PRC [7,40,44,45].

We wish to highlight the work performed by the following people. Floris Laporte developed the Photontorch software used for the main simulation work discussed in chapter 5. Stijn Sackesyn designed the photonic integrated circuit (PIC)s used for the main experimental work discussed in chapter 6. Chonghuai Ma designed the PIC as well as the contact masks for subsequent processing for the work discussed in chapter 7. The simulations and measurements themselves, as well as the accompanying data analysis and processing described in chapters 5 and 6 were performed by the author. For section 6.2 the author wishes to explicitly thank Alessandro Lupo and Elger Vlieg who during their secondment in our group helped perform the experiments described in the section. The final lithography steps and measurements, as well as the accompanying data analysis performed in chapter 7 were done by the author. The PIC design, including accompanying simulations, measurements and data analysis in chapter 8 were all done by the author.

1.7 List of publications

This dissertation has led to the following list of publications in peer-reviewed journals and international conferences.

1.7.1 Journal publications

1. **E. Gooskens**, S. Sackesyn, J. Dambre, P. Bienstman, "Experimental results on nonlinear distortion compensation using photonic reservoir computing with a single set of weights for different wavelengths", *Scientific Reports*, vol. 13, no. 21399, 2023

2. **E. Gooskens**, F. Laporte, C. Ma, S. Sackesyn, J. Dambre, P. Bienstman, “Wavelength dimension in waveguide-based photonic reservoir computing”, *Optics Express*, vol. 30, no. 9, pp. 15 634–15 647, 2022
3. S. Masaad, **E. Gooskens**, S. Sackesyn, J. Dambre, P. Bienstman, ”Photonic reservoir computing for nonlinear equalization of 64-QAM signals with a Kramers-Kronig receiver”, *Nanophotonics*, vol. 12, no. 5, pp. 925–935, 2022
4. A. Lugnan, **E. Gooskens**, J. Vatin, J. Dambre, P. Bienstman, ”Machine learning issues and opportunities in ultrafast particle classification for label-free microflow cytometry”, *Scientific Reports*, vol. 10, no. 1, pp. 1-13, 2020
5. A. Lugnan, A. Katumba, F. Laporte, M. Freiberger, S. Sackesyn, C. Ma, **E. Gooskens**, J. Dambre, P. Bienstman, ”Photonic neuromorphic information processing and reservoir computing”, *APL Photonics*, vol. 5, no. 2, pp. 020901, 2020
6. C. Ma, J. Van Kerrebrouck, H. Deng, S. Sackesyn, **E. Gooskens**, B. Bai, J. Dambre, P. Bienstman, ”Integrated photonic reservoir computing with an all-optical readout”, *Optics Express*, vol. 31, no. 21, pp 34843-34854, 2023

1.7.2 Publications in international conferences

1. **E. Gooskens**, F. Laporte, S. Sackesyn, C. Ma, P. Bienstman, “Wavelength Multiplexing in Photonic Reservoir Computing”, *Annual Symposium of the IEEE Photonics Society Benelux Chapter 2021*, Mons Belgium, 25–26 November 2021
2. **E. Gooskens**, S. Sackesyn, S. Masaad, J. Dambre, P. Bienstman, ”Photonic Reservoir Computing for Wavelength Multiplexed Nonlinear Fiber Distortion Mitigation”, *IEEE SiPhotonics 2023*, Arlington Virginia United States of America, 4-7 April 2023
3. P. Bienstman, A. Lugnan, S. Aggarwal, F. Bruckerhoff-Pluckelmann, W. Pernice, H. Bhaskaran, C. Ma, S. Sackesyn, **E. Gooskens**, S. Masaad, M. Gouda, R. De Prins, ”Optical Computing in Silicon Photonics: Self-Adapting Ring Networks and Quantum Recurrent Neural Networks”, *Natural and Physical Computing (NNPC)*, Hannover Germany, October 2023
4. P. Bienstman, A. Lugnan, C. Ma, S. Sackesyn, **E.J.C. Gooskens**, S. Masaad, M. Gouda, R. De Prins, ”Coherent optical computing in silicon photonics”, *Coherent Network Computing (CNC)*, United States, 2022

5. S. Masaad, **E.J.C. Gooskens**, S. Sackesyn, J. Dambre, P. Bienstman, "Photonic Reservoir Computing for Nonlinear Equalization of 64-QAM Signals with a Kramers-Kronig Receiver", *European Conference on Optical Communication*, Switzerland, 2022
6. G. Von Hünefeld, G. Ronninger, P. Safari, I. Sackey, R. Thomas, P. Cegielski, S. Suckow, E. Seker, D. Stahl, S. Masaad, **E.J.C. Gooskens**, P. Bienstman, C. Schubert, J. Fischer, R. Freund, " Enabling optical modulation format identification using an integrated photonic reservoir and a digital multiclass classifier", *ECOC*, Switzerland, 2021
7. P. Bienstman, J. Dambre, A. Lugnan, S. Sackesyn, C. Ma, **E.J.C. Gooskens**, M. Gouda, S. Masaad, " Photonic Neuromorphic Computing Using Silicon Chips", *Huawei STW*, 2021
8. A. Lugnan, S. Sackesyn, C. Ma, **E.J.C. Gooskens**, M. Gouda, S. Masaad, J. Dambre, P. Bienstman, " Reservoir computing for high-speed photonic information processing", *Photonics in Switching and Computing*, Washington DC United States, 27–29 September 2021
9. A. Lugnan, S. Sackesyn, C. Ma, **E.J.C. Gooskens**, M. Gouda, S. Masaad, Joni Dambre, P. Bienstman, "Photonic reservoir computing for high-speed neuromorphic computing applications", *2021 IEEE Summer Topicals Meeting Series*, Mexico, 2021
10. P. Bienstman, J. Dambre, A. Lugnan, S. Sackesyn, C. Ma, **E.J.C. Gooskens**, M. Gouda, S. Masaad, " Silicon photonics for brain-inspired neuromorphic information processing", *1st Workshop on Neuromorphic Photonics*, 2020
11. F. Laporte, A. Katumba, M. Freiberger, A. Lugnan, S. Sackesyn, C. Ma, **E.J.C. Gooskens**, J. Dambre, P. Bienstman, "Photonic Reservoir Computing", *Photonic Integration Week*, Spain, 2020
12. P. Bienstman, J. Dambre, A. Katumba, M. Freiberger, F. Laporte, A. Lugnan, S. Sackesyn, C. Ma, **E. Gooskens**, "Non-linear signal equalisation using silicon photonic reservoir computing", *ECOC machine learning workshop*, Ireland, 2019
13. P. Bienstman, J. Dambre, A. Katumba, M. Freiberger, F. Laporte, A. Lugnan, S. Sackesyn, C. Ma, **E. Gooskens**, "Silicon photonics reservoir computing at 32 Gbit/s", *5th Workshop on Dynamical Systems and Brain-Inspired Information Processing*, Germany, 2019

2

Machine learning

This chapter explains RC via a bottom-up approach. After an introduction to ML, we first describe simple linear ML models, with special attention to the problem of overfitting. We then describe how starting from this theory on linear models, nonlinear models can be implemented through NNs and how this relates to RC.

2.1 Introduction

ML is a field of computer science in which computer systems are enabled to learn without being explicitly programmed [14, 15]. The computer systems are instead fed a large amount of (labelled) data, and a suitable ML algorithm configures them to arrive at a solution to a certain problem. This process is often referred to as training the system. As will be explained in this chapter, although very powerful, ML is not (yet) at the stage of making human intervention completely redundant. A ML algorithm does not necessarily always succeed in finding a solution to the problem at hand, and the programmer then needs to tweak the algorithm or use an entirely different one altogether. The collecting and pre-processing of data also still often requires considerable human effort. Nevertheless, ML is one of the most successful subsets of artificial intelligence (AI) being pursued in recent years. ML enables solving problems for which no good conventional rule-based approaches are available, such as pattern recognition and prediction. With the growing interest in intelligent computer systems capable of performing human-like tasks, such as driving cars or communicating in natural language [46], and the growing need for large-scale data processing, there has been a surge of interest in machine learning. In this chapter, some key concepts from the field of ML will be introduced with a focus on the most relevant ones used in later chapters.

2.2 Linear models

Of all ML models, linear models are the simplest, both conceptually and in terms of implementation. As the name suggests, linear models express their output y as a linear function of the inputs x_i (equation 2.1).

$$y = \sum_{i=1}^{n} w_i x_i + w_0 \tag{2.1}$$

w_i are called the weight parameters of the linear model with w_0 being called the bias parameter.

Linear models can be divided into regression models (which output continuous numerical values) and classification models (which output a discrete label out of a number of distinct possibilities).

2.2.1 Linear regression

Linear regression can best be explained by visualising the input-output relation. Indeed, a linear model can be visualised as an n-dimensional linear object (n being the number of inputs) in an $n + 1$ dimensional space. For the simplest model consisting of only 1 input, this is a simple line in a 2D plot (Fig. 2.1(a)). For a model consisting of 2 inputs, this is a plane in a 3D representation (Fig. 2.1(b)). Linear regression seeks to minimise the distance between the linear object and the known training samples in this space by configuring the weight parameters w_i of the linear model.

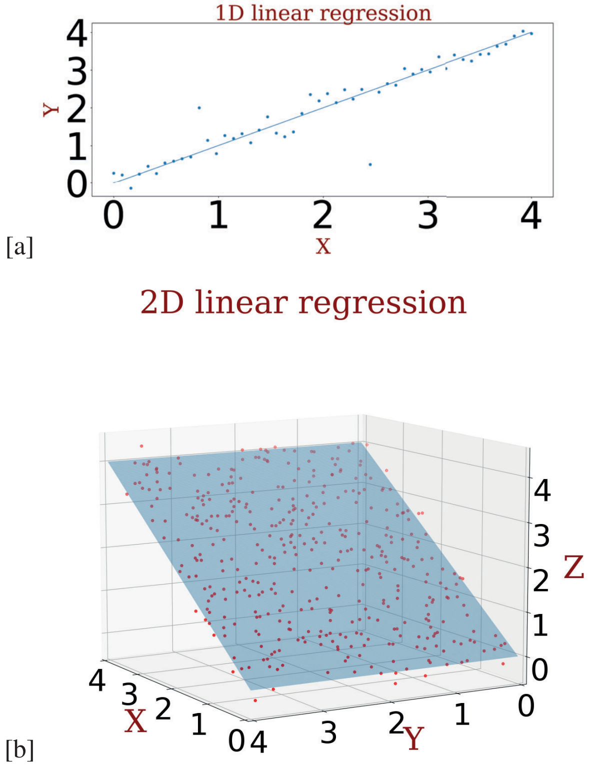


Figure 2.1: Illustration linear regression. (a) 1D-dimensional linear object, a line (b) 2D-dimensional linear object, a plane

When we collect m different inputs and their target outputs in a training set, equation 2.1 can be generalized in matrix form which will allow us to find the optimal configuration for the weight parameters w_i. We define X as the input matrix with corresponding known target vector Y. Each column in X represents the inputs corresponding to an output y_i $i \in [1, m]$. The first row of the X matrix consisting of 1's relates to the bias term. W is defined as an $n+1$-dimensional weight vector with w_0 corresponding to the bias weight.

$$X = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_{11} & x_{12} & \dots & x_{1m} \\ \vdots & \ddots & \vdots & \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} Y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} W = \begin{bmatrix} w_0 \\ \vdots \\ w_n \end{bmatrix}$$

When we apply the model described by the weights W to the dataset described by the matrix X, the result is a set of outputs for each training sample, $R(W, X)$, which is given by

$$R(W, X) = W^T X \tag{2.2}$$

The goal is now to find the 'optimal' weights W. We define $E(W)$ as the error corresponding to the weights W. This error (also called 'loss' in machine learning) can be defined in different ways. Common error metrics are the sum-of-squares error (equation 2.3) or mean squared error (MSE) (equation 2.4).

$$E(W) = |Y - R(W, X)|^2 \tag{2.3}$$

$$E(W) = \frac{1}{m}|Y - R(W, X)|^2 \tag{2.4}$$

The optimal solution for either of these two error metrics is found by applying the so-called Moore-Penrose pseudoinverse [47]:

$$W = (XX^T)^{-1}XY^T \tag{2.5}$$

This solution is optimal in the sense that the model is now configured to have the best possible fit to the known target data used in training. This is not a guarantee that the solution is optimal for unknown target data. This problem, overfitting, is discussed in more detail in the following section.

2.2.2 Overfitting

Overfitting happens when the model is too specialized on the data it was presented with during training as illustrated in Fig. 2.2. As a result does not generalize well. 'Generalizing well' means that the model can get good performance on data that was not seen during training. The disparity between the performance on the training and test data is called the Generalization Gap and is thus large in the unwanted case of overfitting [48].

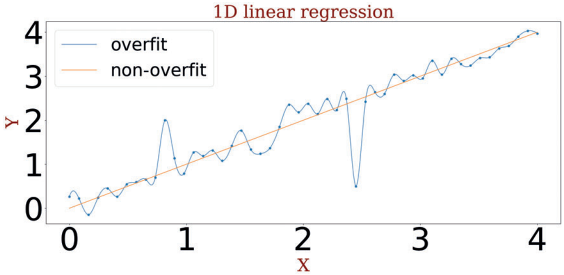


Figure 2.2: Illustration of overfitting. The blue overfitted curve fits the shown train data better. This is most clear when looking at the outliers around x=1 and x=2.5. However it is the non-overfitted model of the orange curve that, despite noise and outliers, correctly deduced the clear linear relation between x and y and will be able to generalize better on unseen data.

How to avoid this problem of overfitting? In general, one should have sufficient training data as compared to the number of weight parameters. A higher ratio #(weight parameters)/#(training samples) increases the capability and tendency of a system to specialize in the training data. This leads to the choice of either increasing the amount of available training data or decreasing the number of weight parameters. The former is usually the preferred approach but is not always possible. Another technique of avoiding overfitting is regularization, which we will discuss next.

2.2.3 Regularization and ridge regression

Regularization in the context of this work refers to a technique to avoid too large weights, which helps to prevent overfitting by limiting the impact of excess weight parameters. It is achieved by adding an extra term to the loss function which depends on weight magnitude, so that equation 2.4 becomes

$$E(W) = \frac{1}{m}|Y - R(W, X)|^2 + \lambda|W|^p \tag{2.6}$$

λ is referred to as the regularization hyperparameter and can be used to tune the amount of regularization. The two most common regularization methods are L1-regularization or least absolute shrinkage and selection operator (LASSO) regularization, for which p = 1, and L2 regularization or Tikhonov regularization, for which p = 2. The application of L2 regularization to linear regression is called ridge regression. For ridge regression, the optimal solution for the weights is found as

$$W = (XX^T - \lambda I)^{-1}XY^T \tag{2.7}$$

Here I stands for the identity matrix. In this work, ridge regression is always used. Indeed, regularization, although not always necessarily in the form outlined here, is standard good practice for ML systems.

2.2.4 Linear classifier

Section 2.2.1 on linear regression described a technique to make estimations of continuous target values y given a set of features x. This section however describes classifiers that should output discrete target values, also referred to as classes. An example of this is determining the bit values contained in a distorted optical signal. Although this problem is distinct, similar techniques as previously discussed can be applied to find the boundaries between those classes.

Let us consider as an example a binary classification task (i.e. one having two classes) in a multi-dimensional feature space. Such a classification task can be performed by adding a simple threshold t as hyperparameter, after taking a linear combination of the features:

$$y = H\left(\sum_{i=1}^{n} w_i x_i + w_0 - t\right) \tag{2.8}$$

Here, H stands for the Heaviside step function:

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases} \tag{2.9}$$

The most simple way of training a linear classifier is to just use the MSE error metric to optimize the weights of the linear combination, using the discrete label as a target. Indeed, even though the goal is to predict discrete class labels, one can just pretend they are continuous to calculate the MSE. Using this approach, the weight parameters can easily be calculated using the Moore-Penrose pseudoinverse.

One issue with this simple approach is that of outliers, i.e. data points that deviate significantly from the norm. These outliers will influence the result considerably, as they will be counted more heavily due to the quadratic MSE loss. Another issue with this simple approach is that of unbalanced data, i.e. data where not all classes are evenly represented. This can cause the regression to focus on the most strongly represented classes at the expense of others.

A more sophisticated approach, which can counteract some of these issues, is to apply to the output of the linear combination not a Heaviside function, but a nonlinear sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.10}$$

The output of the sigmoid is bounded between 0 and 1, thus helping to constrain although not fully eliminating the effect of outliers. It also results in a small contribution to the error in the case of a correctly classified sample point and a large contribution to the error in the case of a misclassified sample point, which better represents the purpose of the classifier and helps with unbalanced data. This additional use of the sigmoid function is referred to as logistic regression. It moreover has the advantage that the outputs can be interpreted as a probability.

In this thesis, we will in general deal with tasks that have balanced data without significant outliers. E.g, a distorted OOK telecom signal for signal equalization has a very close to equal distribution of ’0’ and ’1’ bits, with their corresponding power values contained within certain limits. The simple implementation of ridge regression with thresholding is thus sufficient for our purposes and will be used to implement classifiers throughout the thesis.

2.3 Artificial neural networks

Obviously, for a linear classifier to perform well, the data has to be linearly separable in the feature space. In addition, it is important to note that, the higher the dimension N of the feature space, the easier it will be for the classifier to separate the classes. This is true even when the number of data points is still much larger than the number of features [4].

To be able to classify nonlinear data, people have introduced artificial NNs. These are very powerful and popular models, vaguely inspired by biological neural networks that constitute the cerebrum. They nonlinearly transform a low-dimensional, input into some high-dimensional intermediate feature space. In that way, a NN can efficiently solve complicated nonlinear classification problems [4, 19]. This is illustrated in Fig. 2.3. Likewise, nonlinear regression problems can be solved by NNs as well.

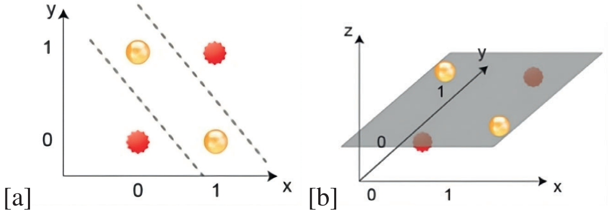


Figure 2.3: In (a) the XOR problem in a two-dimensional input space: a 0 result is illustrated by a red star and a 1 by a yellow sphere. The yellow spheres and the red stars cannot be separated by a single straight line and a linear model will thus be of no use. (b) A NN finds a solution after a nonlinear mapping into higher (three)-dimensional space. Figure taken from [49].

A NN consists of artificial neurons, often called the nodes of the network, that are interconnected by trainable artificial synapses that are often called the weights (parameters) of the network. Each node takes a weighted sum of its inputs and sends it through a nonlinear function, which is referred to as the node's activation function and is typically a sigmoid, before passing it on to (an)other node(s) via a (trainable) weight. By adapting these weights, the final output(s) of the NN can be matched to the desired output(s) associated with the input.

The nodes in an NN can in principle be organized and interconnected in any conceivable way. We will now discuss different possible topologies and explain the methods and challenges concerning the configuration of their weight parameters.

2.3.1 Feedforward neural network

In a feedforward NN, the flow of information is unidirectional, i.e. flowing through successive layers of nodes (see Fig. 2.4).

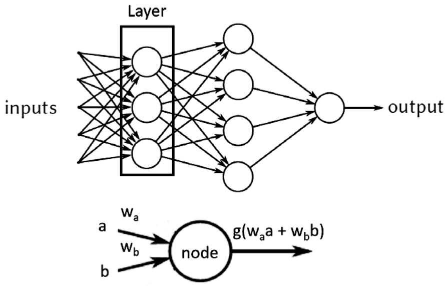


Figure 2.4: Top figure shows a feedforward neural network, i.e. a NN where information flows unidirectionally through successive layers of nodes. Figure taken from [7] with adjustments. Bottom figure illustrates how a node transforms a weighted sum of its inputs through a nonlinear function g().

The most impressive advances made in AI such as language models [46] are usually based on so-called deep NNs, which are essentially feedforward NNs with a large number of nodes and weights [50]. The use of deep NNs is also referred to as deep learning.

As an example, the popular natural language processing system ChatGPT had 175 billion parameters in version 3 [51] and at the time of writing the most recent version 4 is rumored to have over 1 trillion parameters. This comes at computational and energy costs, particularly when training the system [52,53]. The main principles behind this training are discussed in the next section.

2.3.2 Gradient descent

The most simple solution to training a NN would be to just calculate the error (or loss) for every weight parameter combination possible and see which combination yields the optimal result. This weight parameter sweep approach however becomes computationally infeasible for cases with either a lot of weight parameters or broad and high-resolution possible weight parameter ranges. A better solution is gradient descent (GD), a ML algorithm that can be employed to find a solution in an iterative but smart way, in those cases where a closed-form solution can not be found. GD operates by updating the weights in such a way that every subsequent training iteration should lead to a decrease in the loss. To achieve this one needs to calculate the gradient of the loss function with respect its weight parameters $\nabla_W E$. Then, $-\nabla_W E$ is the direction in the weight parameter space along which the error decreases, as illustrated in Fig. 2.5. This explains the name gradient descent.

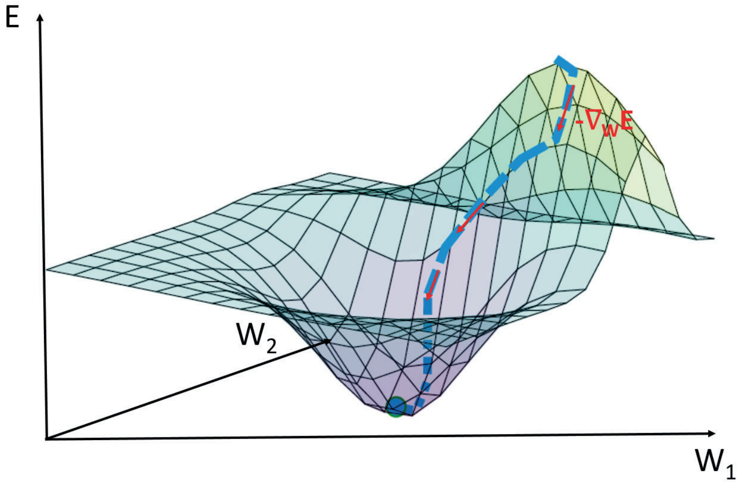


Figure 2.5: During gradient descent, the minimum of the loss function is found by iteratively updating the weights in the direction that the error E decreases. This direction is determined by the negative gradient $-\nabla_W E$ of the loss function. Here this is illustrated with 2 weight parameters W_1 and W_2 [7].

In case of a feedforward neural network, the calculation of the gradient needs to proceed backward through the layers of the network. The backward calculation of the gradient through NNs is referred to as backpropagation [54].

Assume a small variation dz_n^i corresponding to layer i and node n in the network, z being the output of the node before applying the activation function (Fig. 2.6).

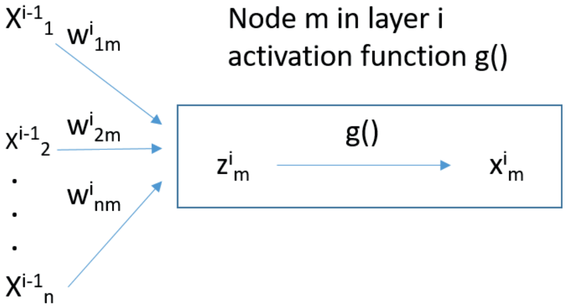


Figure 2.6: Illustration of the various symbols used in the explanation on backpropagation.

This variation will propagate to the end of the NN such that the loss changes by a value dL:

$$dL = \frac{\partial L}{\partial z_n^i} dz_n^i \tag{2.11}$$

To calculate this value, let's first focus on the last layer of the network. Using the chain rule, we can write:

$$\frac{\partial L}{\partial z_n^L} = \frac{\partial L}{\partial x_n^L} \frac{\partial x_n^L}{\partial z_n^L} = \frac{\partial L}{\partial x_n^L} \frac{\partial g(z_n^L)}{\partial z_n^L} \tag{2.12}$$

Here, x_n^L denotes the output of node n of the last layer L of the network, after applying the node's activation function g (Fig. 2.6).

The next step is to work our way backwards through the network, and relate what is happening at layer $i + 1$ to the earlier layer i. The chain rule allows us to express this as:

$$\frac{\partial L}{\partial z_n^i} = \sum_m \frac{\partial L}{\partial z_m^{i+1}} \frac{\partial z_m^{i+1}}{\partial z_n^i} \tag{2.13}$$

We can say that the change of the loss L with respect to a variation z_n^i in layer i depends on the changes of the loss L due to the resulting variations z_m^{i+1} in the next layer $i + 1$.

This can be rewritten as:

$$\frac{\partial L}{\partial z_n^i} = \sum_m \frac{\partial L}{\partial z_m^{i+1}} \frac{\partial \sum_k w_{km}^{i+1} x_k^i}{\partial z_n^i} = \sum_m w_{nm}^{i+1} \frac{\partial L}{\partial z_m^{i+1}} \frac{\partial g(z_n^i)}{\partial z_n^i} \tag{2.14}$$

Here, w_{nm}^{i+1} denotes the weight of the interconnection between node n of layer i and node m of layer $i + 1$ (Fig. 2.6).

Now that we have derived all these intermediate results, we can finally write down what we are ultimately interested in, namely the derivative of the loss function with the respect to the weights, as this is what we need for gradient descent. Using the chain rule we can write:

$$\begin{aligned}
\frac{\partial L}{\partial w_{nm}^{i+1}} &= \frac{\partial L}{\partial z_m^{i+1}} \frac{\partial z_m^{i+1}}{\partial w_{nm}^{i+1}} \\
&= \frac{\partial L}{\partial z_m^{i+1}} \frac{\partial \sum_k w_{km}^{i+1} x_k^i}{\partial w_{nm}^{i+1}} \\
&= \frac{\partial L}{\partial z_m^{i+1}} x_n^i \\
&= \frac{\partial L}{\partial z_m^{i+1}} g(z_n^i)
\end{aligned} \tag{2.15}$$

To summarise, the calculation of the gradients via backpropagation proceeds in roughly 3 steps:

1. calculate $\frac{\partial L}{\partial z_n^L}$ at each neuron of the last layer of the network using equation 2.12
2. calculate $\frac{\partial L}{\partial z_n^i}$ at each neuron of each layer using all the errors of all the neurons at the next layer using equation 2.14
3. calculate the gradient of the loss with respect to the weight between layer i and $i+1$ $\frac{\partial L}{\partial w_{nm}^{i+1}}$ by using $\frac{\partial L}{\partial z_m^{i+1}}$ and the activation at layer i as in equation 2.15.

The MSE, used to define the closed-form solution in 2.5, is still the error metric/-function employed when using GD in this work.

Mathematically every iterative update to apply to the weight parameters can be calculated as follows:

$$W_{update} = -\lambda \nabla_W E \tag{2.16}$$

Here λ stands for the learning rate hyperparameter, which in the above equation is a strictly positive number. This parameter typically needs to be tuned to get a good result. Next we will describe a way in which to do this.

Adam optimization algorithm

If we choose the learning rate parameter λ too small, then reaching the global minimum becomes computationally expensive, as the number of training iterations will increases, and the algorithm can get stuck in local minima. On the other hand, choosing λ too large can lead to the loss/error evolving semi-randomly, overshooting the global minimum or not reaching it. A possible solution lies in a dynamic approach of setting the learning rate, on an individual weight basis. One implementation is the so-called Adam optimization algorithm [55].

The algorithm makes use of exponential moving averages of the gradient, $m(t)$, and the squared gradient, $v(t)$:

$$\begin{aligned} m(t) &= \beta_1 m(t-1) + (1-\beta_1) g(t) \\ v(t) &= \beta_2 v(t-1) + (1-\beta_2) g^2(t) \end{aligned} \tag{2.17}$$

These are often referred to as the estimates of the first and second moments of the gradient, i.e. mean and uncentered variance. β_1 and β_2 are hyperparameters of the model with recommended values, which were employed in this work, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The moving averages are initialized as (vectors of) 0's. The explanation on how the Adam algorithm avoids the moment estimates being biased towards zero during the initial timesteps is outside the scope of this section and the interested reader is referred to [55] for this and other details on the algorithm implementation. The name Adam is derived from 'adaptive moment estimation'.

Working with the first moment of the gradient instead of just the gradient can help the algorithm to converge faster to the minimum, as it makes ever larger update steps because of the momentum it acquires for every successful iteration. This gives the algorithm a better chance to power through local minima or plateaus in the loss function.

Adam also uses the second moment of the gradient, as a way to keep track of how much a weight has already been updated. This is done to balance out the update speed among the weights, making sure all weights are trained at roughly the same rate. Doing so is beneficial as it allows for a more thorough exploration of the weight parameter space, giving a better chance to the algorithm of finding the global minimum.

2.3.3 Recurrent neural networks

A disadvantage of a feedforward NN is the lack of memory due to the information flowing unidirectionally in the layered architecture. This renders regular feedforward NNs less naturally suited to processing temporal information, a requirement for tasks such as pattern recognition and prediction or robot/vehicle control. There are ways to process temporal tasks with feed-forward NNs by translating the temporal problem into a spatial one. This can be done for instance by using a tapped delay line, and feeding the network not just with the current sample, but also with past samples. This approach is called a time-delay neural network and was first introduced in [56]. Disadvantages are the rapid increase of the delay line length and the number of NN weights with the amount of required memory. A better solution is the use of a RNN, which we will explain now.

A RNN is a NN where the connections between nodes form cycles, thus preserving in their internal states a nonlinear transformation of the input history (Fig. 2.7).

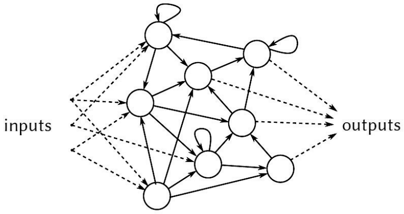


Figure 2.7: Recurrent neural network. Figure taken from [7].

Such a RNN is a dynamical system, since the output of the system no longer exclusively depends on the current input, but it has memory, allowing it to process temporal information. The main disadvantage of RNNs comes from the increased difficulty of their training. This is due to to the fact that small changes in the weights can lead to significant changes all throughout the network, because of the increased inter-dependencies between weights.

In addition to this, RNNs also present challenges to the gradient descent algorithm described in section 2.3.2. More specifically, it creates problems in the calculation of the gradient via backpropagation. A RNN can be interpreted as a NN with an infinite amount of layers. To make backpropagation possible we unroll the RNN in time. Each unrolled timestep has a copy of the network that serves as the input to the subsequent timestep. Each timestep can be seen as a layer of a feedforward NN. Errors are then calculated and accumulated backwards through the timesteps. We call this technique Backpropagation Through Time. However, calculating the gradients of the error with respect to the weights backwards through a large number of timesteps can lead to ever increasing gradients (the exploding gradient problem), or gradients that are converging to zero (the vanishing gradient problem) [57,58].

To address this problem, the concept of Reservoir Computing was introduced, which we will describe next.

2.3.4 Reservoir computing

The high dimensional feature space of NNs makes for more complicated ML models and more computationally intensive training, with the risk of not reaching an optimal solution, as well as requiring more data to avoid the problem of overfitting discussed in section 2.2.2. The dynamical nature of a RNN presents additional challenges to the training as discussed in 2.3.3.

The strength of RC systems, which are a variation of the earlier discussed RNNs, lies in that they avoid these issues and are thus much easier to train. RC also greatly facilitates the practical application of RNNs for hardware implementations, including a greater tolerance to fabrication errors, through reducing the number of configurable parameters and separating these from the dynamical part of the system [3–5, 16, 17].

RC employs a randomly initialised fixed RNN, called the reservoir, which is left untrained and to which a simple linear layer is added (Fig. 2.8). Only this linear layer is trained. The training of the linear layer can be done using a simple linear model for which explicit, non-iterative formulas are available (see Section 2.2.1). The dynamics of the reservoir should be rich enough that the mixing of the inputs boosts the computational power of the linear layer, but again not so rich that it does not forget its past inputs eventually and becomes chaotic, meaning that the output of the system depends so strongly on initial starting conditions as to become seemingly random. Therefore, the system should display what is often called fading memory and outputs that are reproducible and robust against noise, while still exhibiting sufficiently different/similar dynamical outputs to different/similar inputs.

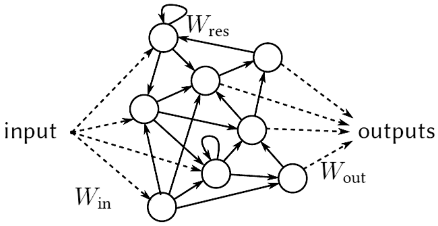


Figure 2.8: Schematic representation of a reservoir computing system. The input signal is fed into the reservoir. Inside the reservoir, the signal mixes with previous versions of itself due to the dynamic architecture of the reservoir. At each time, (a part of) the reservoir state is read out by the linear layer, with weights W_{out}, to produce a time dependent output. Figure taken from [7].

It should be noted that the use of a RNN as the reservoir is strictly speaking not required and the reservoir can in practice be any dynamical system obeying a broad set of constraints with regards to its dynamical regime as noted earlier. One extreme example is the use of a bucket of water as a reservoir for speech recognition [59]. Such exotic reservoir designs are however not presented in this thesis, focusing instead on RNNs implemented in hardware as photonic reservoirs.

3

Silicon photonics

This chapter explains what silicon photonics entails and then lists and explains all photonic components relevant for this thesis.

3.1 Photonics

Photonics refers to the understanding and exploitation of the behavior of light propagation and its interaction with materials. Integrated photonics consists of the exploitation of components integrated on a chip, in which light propagates through structures with the smallest features of the order of the wavelength of light, or even smaller [60]. This evolution is reminiscent of the evolution in the field of electronics in the second half of the last century, whereby components and systems were scaled down continuously and became increasingly complex. Similarly, this was referred to as integrated electronics or more often microelectronics. In this chapter key concepts from the field of integrated photonics will be introduced, with a focus on the most relevant ones used in later chapters.

3.2 Silicon photonics

The family of hardware platforms in which we implement our photonic computing paradigm is referred to as silicon photonics. In the broadest possible sense, this means integrated photonics with silicon or silicon nitride as the optical material system. This is possible due to the transparency of silicon for wavelengths in both the C-band (wavelengths around 1550 nm) and O-band (wavelengths around 1300 nm). These wavelengths are the main wavelengths used in optical telecommunications. Possibly, we can integrate other materials on top of silicon, such as III-V compound devices [61].

The main advantage of silicon photonics is its compatibility with the production processes of complementary metal–oxide–semiconductor (CMOS) electronics, which is an already well established industry. In this way, the main benefit of digital electronics, namely cost-effective mass production at sub-micrometer precision, is preserved in this new approach. In addition, this raises the possibility of co-integrating electronics, e.g. transimpedance amplifiers, and photonics on the same silicon chip.

We will now explain the different material systems used in this thesis in more detail.

3.3 Silicon on insulator, silicon nitride and III-V compounds

Two types of silicon photonics hardware platforms were used in the making of this thesis. IMEC's iSiPP silicon on insulator (SOI) platform [62] and Ligentec's silicon nitride (SiN) platform [63]. For the SiN platform the silicon core of the SOI-system is replaced by a SiN core [64]. The SiN platform has the disadvantage of a lower index contrast which leads to bigger photonic structures and thus increased chip footprint. On the other hand, the main advantage of SiN platform is its lower optical loss.

III-V compounds, meaning the combination of an element from group III in the periodic table with an element from group V, are the dominant material system for making active components such as light sources and optical amplifiers. The reason why we need to resort to III-V components to make many active components is that they have a direct bandgap, as opposed to the indirect bandgap of silicon. The integration of III-V components on the silicon platform for this work was done in our research group, using its micro-transfer printing techniques [65–69].

In integrated electronics, ubiquitous components are transistors and capacitors. Similarly, in integrated photonics there exist frequently used components, although there are more of them. We will now present a number of these with particular relevance for the rest of the thesis.

3.4 Integrated waveguide

One of the cornerstones of silicon photonics is its ability to manufacture light-guiding structures, the analog of transmission lines in electronics. Indeed, the fundamental idea behind integrated optics is the manipulation of light by waveguides and not by free-space optical components like lenses and mirrors [60]. Whereas electrical current is confined to materials with higher conductivity, light is confined to materials with higher refractive index.

In silicon photonics, a silicon or SiN waveguide layer is used as a core and often SiO_2 or air is used as the cladding (Fig. 3.1) [70]. The higher index contrast between core and cladding, in case of a silicon core, increases the confinement of the light into the core material, which makes it easier to realize smaller bend radii and guide light more efficiently [71]. A direct consequence is that the device footprint is reduced significantly, an advantage for chip footprint intensive photonic reservoir implementations. On the other hand, the higher index contrast makes the waveguide more sensitive to surface scattering loss because of waveguide (sidewall) roughness (Fig. 3.2), so a trade-off is involved.

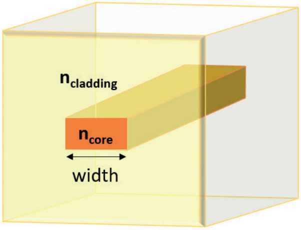


Figure 3.1: An example of a possible waveguide structure which consists of a central high refractive index strip surrounded by lower index material ($n_{core} > n_{cladding}$) and thus providing optical confinement in 2 dimensions.

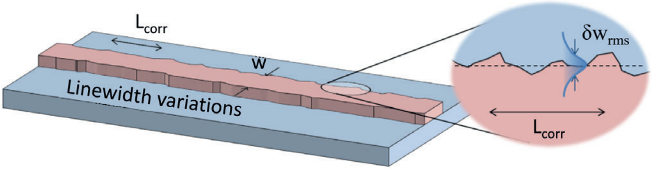


Figure 3.2: Illustration of sidewall roughness. Figure taken from [72].

An optical mode is a propagating electromagnetic wave of which the transversal shape, amplitude and polarization, does not change during propagation. It thus describes an intensity pattern of light propagating through the waveguide which determines the effective refractive index the light feels, i.e. a weighted average of the core and cladding refractive indices. This in turn determines the phase change and the time delay a light pulse incurs when travelling over a certain distance in the waveguide. Both phase change and time delay are important characteristics of a waveguide interconnection which guide their design, in particular their length. The phase change $\Delta\phi$ and time delay Δt over a length l are described by the effective refractive index n_{eff} and group index n_{g} respectively:

$$\begin{aligned} \Delta\phi &= \frac{2\pi}{\lambda} l \, n_{\text{eff}}(\lambda) \\ \Delta t &= (l/c) n_{\text{g}}(\lambda) \end{aligned} \tag{3.1}$$

Here λ signifies wavelength and c stands for the speed of light. Note that n_{eff} and n_{g} are wavelength dependent, this is referred to as dispersion. Due to fabrication errors, such as the aforementioned sidewall roughness, there is an uncertainty on the phase change, which results in phase errors which need to be accounted for.

Whether a waveguide supports a single optical mode (single-mode waveguide) or multiple optical modes (multimode waveguide) is dependent on the waveguide dimensions. Single-mode waveguides are often preferred, also in the case of this work. Reasons are the avoidance of mode conversion losses and the advantage of having only a single phase associated with an optical signal propagating through the system.

3.5 Grating coupler

An important operation in silicon photonics is low-loss coupling of light between optical fibers or free-space optics and integrated waveguide devices. To this end a grating coupler (GC) structure can be employed (Fig. 3.3) [73, 74]. Diffractive grating structures consist of a 1- or 2-dimensional periodic variation in the refractive index of the structure, achieved by for instance etching or deposition. The light is directed into such diffractive grating structures at the surface of the chip, resulting in a change of propagation direction and coupling into the plane of the chip. Apart from changing the direction of the light, an additional problem is caused by the large modal mismatch between a single-mode fiber, which has a mode field diameter of 8-10 μm at 1310-1550 nm, and an integrated waveguide with a width of several hundred nanometers to around a micrometer. To this end, a spot-size converter based on a tapered waveguide of hundreds of microns in length can be employed between the grating coupler and the final on-chip waveguide.

Grating couplers are alignment tolerant and easy to work with in experiments. However, they have a higher loss compared to more straight-forward edge couplers, which could be preferred in case power budget is an important issue.

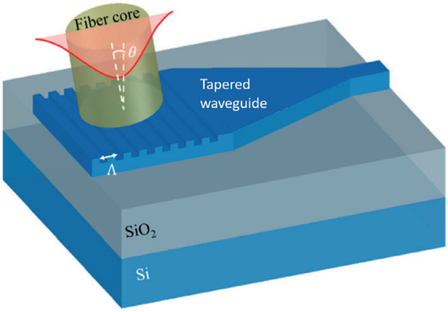


Figure 3.3: Grating coupler illustration. The light is coupled at an angle θ in/out of the plane of the chip. The grating coupler has 1-dimensional periodic variation with period Λ. Figure taken from [74] with adjustments.

3.6 Directional coupler

A directional coupler consists of two waveguides running parallel to each other (Fig. 3.4).

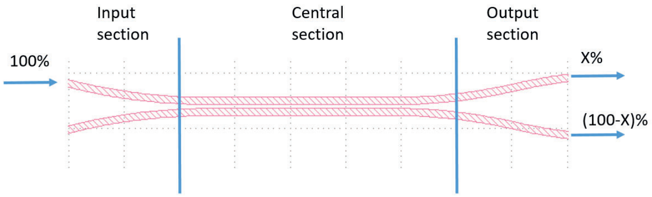


Figure 3.4: Directional coupler design with waveguide cladding omitted for clarity.

A directional coupler can be divided into 3 sections: a central section and an input and output section. The coupling naturally takes place in the central section between the input and output sections. Optical power is transferred from one waveguide to the other. The transfer is produced by synchronous optical coupling based on the overlapping of mode profiles in the two waveguides. To obtain sufficient energy transfer, the lightwaves in the two waveguides must have nearly the same phase velocity. The amount of energy transfer or splitting ratio can be set from 0% to 100% by adjusting waveguide separation and interaction length [75].

3.7 Multimode interferometer

A $M \times N$ MMI structure has the function to distribute the light between its M ports on one side and its N ports on the other side in a predetermined way which depends on the MMI design. It consists primarily of a broad multimode section that interfaces with regular single-mode integrated waveguides (Fig. 3.5). When an MMI is excited by an incident wave, the field profile is decomposed into the eigenmodes of the central structure. Even though there is no exchange of energy between these eigenmodes, they propagate at different velocities, and the result is an interference pattern that changes along the length of the MMI. The intermodal interference is accompanied by so-called self-imaging effects of the input field which is exciting the multimode waveguide [76, 77]. The design of an MMI is usually done by way of simulations, because of the complex mathematics involved in accurately modelling the structure.

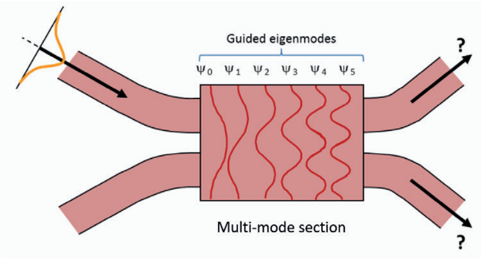


Figure 3.5: 2x2 multimode interferometer. Figure taken from [76] with adjustments.

3.8 Mach-Zehnder interferometer

A Mach-Zehnder interferometer (MZI) consists of an input waveguide, a 1x2 3dB splitter, two waveguide arms, a 2x1 or 2x2 combiner and 1 or 2 output waveguide(s). The splitter and combiner (which can be identical components but used in the opposite sense) can be a previously described MMI [60]. Other options are a symmetrical Y-junction, not reviewed here, or a directional coupler, section 3.6. A MZI transforms a difference $\Delta\Phi$ in phase between the arms of the interferometer to an intensity variation (amplitude modulation) in the output waveguide(s) through the interference of the light (Fig. 3.6).

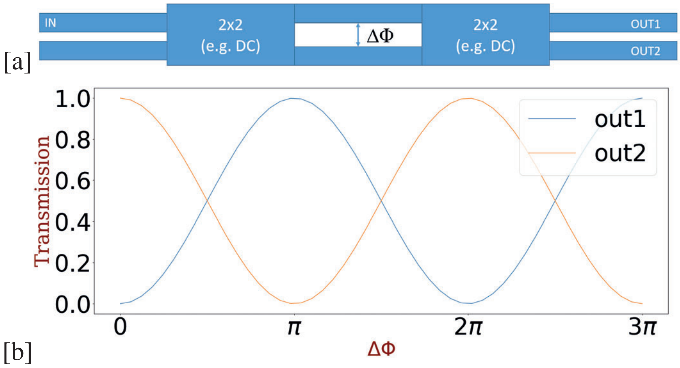


Figure 3.6: (a) Illustration Mach-Zehnder interferometer. (b) Transmission as function of $\Delta\Phi$.

4

Photonic reservoir design and readout implementation

This chapter discusses in detail the reservoir design. It emphasizes the importance of the time dynamics in the reservoir and how the design can take these into account. It discusses how an optimized method of injecting signals into the reservoir can impact performance. It proceeds by taking a look at one particular kind of fabrication imperfection that is of special importance to the use of WDM. Next the chapter moves on to a discussion of the various readout implementations, namely electrical versus optical and integrated versus non integrated. Special attention is devoted to the most complicated case of an integrated optical readout.

4.1 Introduction

Throughout this work, several different reservoirs, with accompanying trainable linear layer implementations, were experimentally measured or simulated. All the reservoirs can be said to have the same underlying design structure. That is to say, they share the same node interconnection layout but not necessarily the same number of nodes. The trainable linear layer of a reservoir will henceforth be referred to as the ’reservoir readout’ or simply the ’readout’. All the different readout implementations share the same purpose, namely to form a linear combination of the reservoir outputs. However, differences arise in whether these readouts are implemented optically (with the photodetector after the linear combination), or electrically (with the photodetector before the linear combination), and whether they are integrated on a chip or implemented digitally on a separate computer.

This common reservoir structure with the possible variations therein and the different possible readout implementations will be explained and motivated in this chapter. First, we start with a look at the design of the reservoir itself.

4.2 Reservoir design

As mentioned in section 1.3, there is a broad range of possible optical reservoirs. In this thesis, the explored hardware platform is an implementation of multiple spatially distributed nodes with the interconnections consisting of waveguides. The node themselves are either 3x3 MMIs or two directional coupler (DC)s combined to create a three-by-three structure (Fig. 4.1) [16, 38–42].

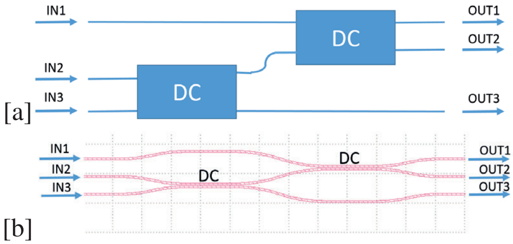


Figure 4.1: DC-based node (a) schematic (b) design with waveguide cladding omitted for clarity. IN2 and OUT2 are reserved for interfacing to and from the reservoir with the remaining inputs and outputs available for inter-node connections. Note that IN1 does not couple to OUT3. This will lead to altered, but not necessarily inferior, reservoir dynamics compared to 3x3 MMIs [42].

Using DCs over MMIs can significantly reduce optical loss, as the latter is prone to have increased losses due to suboptimal design and fabrication [40]. One drawback of DCs however, is that their splitting ratio is sensitive to fabrication tolerances, often leading to performance discrepancies between design and fabrication. The splitting ratio can even deviate locally within the same die. However, a main strength of reservoir computing, as noted in 2.3.4, is to be tolerant of such deviations. Indeed, simulations indicate that a maximum splitting ratio deviation of 30:70 should not significantly impact performance [42], rendering the DC-based low-loss node suitable for a PRC system and confirming the robustness of our approach to variations inside the reservoir.

The chip architecture is designed to cause interference between the different light paths reaching each reservoir node. The combination of interference and a photodetector creates a quadratic nonlinearity, such that even a passive reservoir computing system has nonlinearity.

Each 3x3 node structure has 2 inputs and 2 outputs which are used for inter-node connections, with the remaining input and output serving as interfaces to and from the reservoir. The fraction of light being tapped out of the reservoir at a node is a design parameter which influences the optical losses and the memory of the reservoir. The optimal fraction, which will likely be dependent on the task and the other reservoir design parameters, is subject of ongoing research that is not part of this thesis.

The investigated reservoirs all shared the same layout/architecture, shown in Fig. 4.2 (a) and referred to as the four-port architecture, but could differ in the number of nodes.

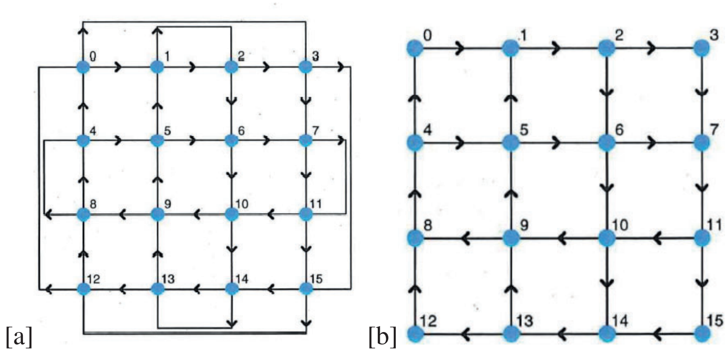


Figure 4.2: (a) the improved four-port architecture used throughout this work. (b) the previous swirl architecture. Note that in the swirl architecture, in contrast to the four-port architectures, not every node is symmetrical, i.e. having an equal number of inputs and outputs. Figures taken from [78].

The outer connections of the four-port architecture were an improvement on the older swirl architecture shown in Fig. 4.2 (b) and employed in [38]. The four-port architecture suffers fewer losses, by avoiding at the edges of the reservoir MMIs with a different number of input and output ports. Such asymmetric MMIs lead to inherent modal radiation losses. Not only does the four-port architecture avoid these undesired losses, but this strategy also contributes to a more uniform power distribution and additional mixing between states that were topologically far apart in the swirl architecture. This can be done by exploiting extra output ports of the node structures smartly. A better power uniformity not only increases the richness of the dynamics, but it also facilitates measuring the states in an experimental setup as node signal-to-noise ratio (SNR) is improved. This is an important consideration because, as noted chip, losses could be high due to suboptimal design and fabrication of the MMIs [40,79].

Most of our reservoirs used 16 nodes. This number was proven in past work [38,80] to be sufficient to achieve good performance on bit level tasks, such as header recognition or boolean operations, as well as on signal equalization tasks.

We will now take a closer look at the time dynamics of this type of reservoir.

4.3 Time dynamics of the reservoir

For a passive reservoir, the only timescales that matter are related to the speed, i.e. the modulation frequency, of the signal itself and the node interconnection delays. For active reservoirs, the time dynamics of the active components, e.g. SOAs, need to be taken into account and the situation becomes more complex.

The node interconnection delays are usually normalised and expressed in terms of bit period of the binary input signal. This is because interconnection delays should be matched to the modulation rate of the input signal for optimal performance. Past research on this subject was undertaken on the swirl architecture (Fig. 4.2b) as documented in [38], see Fig. 4.3.

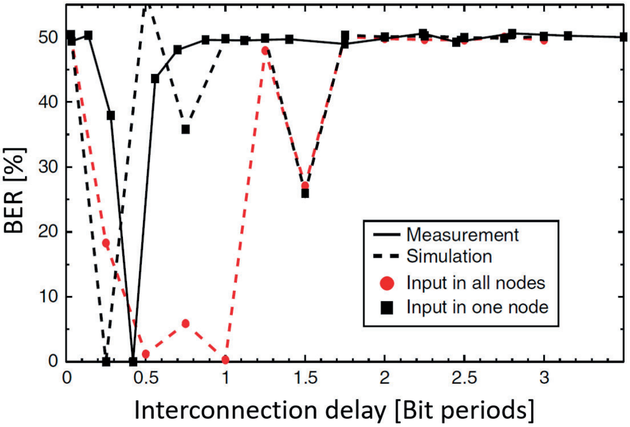


Figure 4.3: Performance as a function of interconnection delays. The task used is the XOR operation between the 1-bit delayed signal and the 2-bit delayed signal. Figure taken from [38].

For the fourport architecture (Fig. 4.2b) the length of the added outer connections is chosen to be an integer times that of the inner connections.

The interconnection delays should be sufficiently long such that the time dynamics of the reservoir are sufficiently slow to be fully captured by our measurement equipment, i.e. the detector. On the other hand, the interconnection delays should not be too long. Indeed, if the delay between nodes exceeds the modulation bit period, bits become 'hidden' in between nodes, causing a blind spot in the memory of the reservoir.

As one can see from Fig. 4.3 the optimal interconnection delays can depend on the input strategy used, i.e. in which nodes the input is injected. We will now investigate the possible input strategies in more detail.

4.4 Input strategy

As we have seen, the choice of the number and configuration of the input nodes can impact the performance of the reservoir.

The most obvious way to get the signal into an integrated photonic reservoir is to inject it in a single node, for example with a grating coupler (Section 3.5), and allow it to propagate throughout the network. This approach can suffer from poor reservoir dynamics due to optical losses inside the reservoir, which limit the effective number of delayed feedback paths that interfere with each other. To counteract this, a multiple-input node strategy with the same total input power injected into the reservoir but distributed over different nodes can be employed. As shown in [80], the enhanced reservoir dynamics, i.e. an increase in the number of possible mixing combinations of the signals from which the machine learning algorithm can extract useful features, can contribute considerably to the computing power of the system. [80] concludes ”that injection of power into the central nodes of the reservoir provides the best combination of performance and energy efficiency”. This aforementioned energy efficiency is another important consideration, as it increases the SNR of the reservoir nodes.

With the outer connections of the four-port architecture, which were absent in [80], it has become less clear which nodes are central nodes in which to inject the power. The important aspect to keep in mind when selecting input nodes is that the injected signals spread to and mix in all nodes of the reservoir to maximize the mixing combinations. This can be achieved by choosing the input nodes to be spread out over the reservoir. The number of input nodes chosen depends on the size of the reservoir, with larger reservoirs being allotted more input nodes. Inputting power into multiple nodes of an integrated photonic reservoir can be achieved by having power splitters after the input grating coupler to divide the power towards several different nodes.

Another aspect that influences the dynamics of the reservoir, apart from the just discussed input strategy, is phase randomisation due to fabrication errors, which we will discuss next.

4.5 Impact of phase errors in the waveguide interconnections

In section 3.4 we introduced the concept of phase errors. In this section we study how these impact our photonic reservoirs, specifically how it relates to their simulation and use. Due to the high refractive index contrast of the silicon photonics platform, the devices are very sensitive to geometric variations [81]. For this work, the intra-die geometric variation, specifically the variation related to the interconnection waveguides in a single reservoir, is of importance, as it impacts the phase information inside the reservoir. Phase information is a key aspect of PRC using coherent reservoirs, as interference effects play a key role in the reservoir dynamics.

Variations include thickness and width fluctuations, which are caused by process variations such as equipment and pattern density non-uniformity [82]. These width fluctuations can also be referred to as waveguide (sidewall) roughness (Fig. 3.2).

Roughness is most often studied in the context of propagation losses, but here its effect on the phase is of interest. This waveguide roughness depends on the fabrication process and will be very different for devices fabricated with deep UV lithography or e-beam lithography, and can vary between fabs. The thickness fluctuations on the other hand depend largely on the qualities of the source wafer. These width and thickness fluctuations impact the n_{eff} of the interconnection waveguides and thus the phase change light undergoes travelling through such an interconnection waveguide. From [82] $\Delta n_{\text{eff,intradie}} \approx 0.015$ and $\Delta n_{g,\text{intradie}} \approx 0.015$ for imec's Multi-Project Wafer (MPW) service, which uses 200 mm wafers and fabrication through 193 nm deep-UV lithography.

To accurately estimate the phase errors due to waveguide roughness we would need to assume a distribution for the effective index variation $\delta n_{\text{eff}}(x)$ and use the following equation:

$$\Delta\phi = \frac{2\pi}{\lambda}\int_0^l \delta n_{\text{eff}}(x)dx \tag{4.1}$$

It can be shown that the expected magnitude of the phase errors, $\Delta\phi$, scales with the square root of the waveguide length if the correlation length of $\delta n_{\text{eff}}(x)$ is much smaller than the length of the waveguide [83].

We simplify by assuming the worst case scenario, i.e. that δn_{eff} is constant over the length of the waveguide and $= \Delta n_{\text{eff,intradie}}$ so that the phase errors accumulate linearly over the length of the waveguide. Applying this to an interconnection waveguide with a length $l = 1mm$ (a typical length for the delay spirals inside the reservoir) at wavelength λ = 1550 nm, using:

$$\Delta\phi = \frac{2\pi}{\lambda} l \Delta n_{\text{eff,intradie}}, \tag{4.2}$$

the resulting phase change $\Delta\phi \approx 61 \gg 2\pi$. Although this worst case scenario assumed a systematic n_{eff} variation instead of variation caused by waveguide roughness and we thus overestimate $\Delta\phi$ this shows how sensitive our waveguide phase change is to n_{eff} variation.

In other words, the exact phase change accumulated over interconnection waveguides cannot be known. This does have to be taken into account in reservoir simulations and it means that each fabricated reservoir has its own phase values and needs its own individually trained readout for which identical performance is not guaranteed [16]. However, this does not preclude good system performance, as the readout can adapt to the specific reservoir phase dynamics during training. Indeed, not having to know or control the exact implementation of the reservoir is one of the key advantages of the reservoir computing paradigm.

Now that we have discussed the behavior of the reservoir itself, the next step is to combine signals from different nodes, which is performed by the readout.

4.6 Readout implementation

4.6.1 Electrical versus optical readout

A first distinction in the type of readout can be made between electrical readouts and optical readouts. The primary difference between them lies in whether the node signals are converted into electrical signals prior to the weighted-sum operation (i.e. the linear combination) or after this sum. An optical readout applies complex-valued weights directly to the coherent optical signals, with the final readout output being the opto-electrical conversion of the superposition of all the weighted inputs by the photodetector (PD) (Fig. 4.4 (a)). Conversely, an electrical readout first detects and converts coherent optical signals from the reservoir, subsequently weighing them using real-valued weights and summing them in the digital electrical domain (Fig. 4.4 (b)).

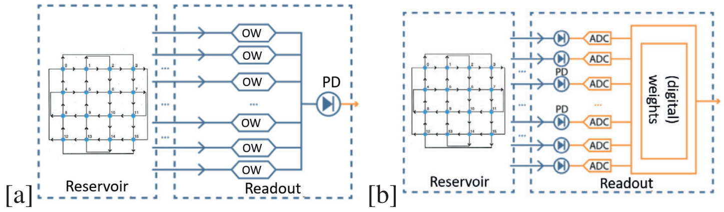


Figure 4.4: (a) optical readout OW stands for optical weights (b) electrical readout ADC stands for analog to digital converter. Figures taken from [42] with adjustments.

An optical readout is a faster, more powerful and more efficient system, as it has a processing time that is quasi-instantaneous, compared to the latency of a digital electronic implementation. It also only needs a single photodetector and analog-to-digital converter to read out the output of the entire system. The main drawback of an optical readout is the more difficult training, considering that one has to weigh both amplitude and phase of the optical signals. This will now be examined in detail.

Difficulty of training an optical readout

For a linear readout, we would expect to be able to use simple linear regression to do the training. However, the optical signals in the optical readout contain both amplitude and phase information and thus need to be described using complex valued data. However, for many tasks our desired outputs are usually real valued, e.g. the ideal bit values contained within a distorted amplitude-modulated optical signal. As mentioned before, this real-valued data is obtained from the linear combination of the optical signals by a nonlinear operation, i.e. the opto-electronic conversion of the photodetector, which is a squaring operation. There is now the important question of how to define a complex-valued target data for the linear combination before this nonlinear conversion step takes place. Different complex target values Y can give rise to the same real value after conversion, as we are projecting from the higher dimensional complex-valued space to the lower dimensional real-valued space. Putting the imaginary part of Y to a fixed value (e.g. zero) is the most simple solution, but this severely and unnecessarily limits the target space of the linear combination, leading to sub-optimal solutions. To illustrate this mathematically, we adapt equation 2.5 to the complex domain by exchanging the transpose for the hermitian transpose:

$$W = (XX^H)^{-1}XY^H \tag{4.3}$$

The linear combination is then given by equation 4.4.

$$Y = W^H X \tag{4.4}$$

Conversion to real-valued output data by an idealized PD is then given by equation 4.5.

$$|Y|^2 = |W^H X|^2 \tag{4.5}$$

If we only care about the norm of Y, and not its phase, we don't have enough information to construct a target Y that can be used for equation 4.3. What we should do instead is minimising the following error function:

$$E(W) = \frac{1}{2m} \left|Y - |W^H X|\right|^2 \tag{4.6}$$

The enclosing of the expression $W^H X$ into the expression for the complex norm $||$ makes it so that the complex-valued weights W are free to take on any phase they need. However, in this case, a closed-form solution is no longer available. To deal with this problem, a more computationally intensive alternative has to be used, one that can find a solution through working directly with the error. We already introduced such an alternative, namely gradient descent, in the context of multi-layered NNs in section 2.3.2. In this thesis, we will use gradient descent in the form of backpropagation, but other possibilities such as evolutionary algorithms or simple random search exist.

We now move on to the second distinction we make in the type of readout, namely integrated versus non-integrated.

4.6.2 Integrated versus non-integrated readout

An integrated readout refers to having the weighting elements (and preferably also the detector) on-chip. This is preferred mainly because it leads to a much more compact and economically viable system. However, non-integrated readouts are potentially worthwhile systems in their own right and can serve as an intermediary step.

For an electrical readout in particular, it is easy to implement a non-integrated readout by implementing the weights digitally on a computer. Having a digitally implemented readout also opens up additional signal post-processing possibilities, be it at a computational cost. A notable example is optimizing the delays between node signals and adding delayed copies of node signals. A drawback of such a non-integrated electrical readout is that each channel needs its own high-speed photodiode and analog-to-digital converter.

The most ambitious target is a system with an integrated readout and optical weights. This is the system with the highest potential, but also with the highest level of technological difficulty, both in chip design and experimental measurements. The implementation of the optical weights in such an integrated readout will now be discussed in more detail.

4.6.3 Implementation of optical weights in an integrated readout

Having off-chip optical weighting elements involves coupling out the light from a high number of nodes, e.g. using a fiber array, connecting these to an equally large number of bulky optical weights, e.g. Mach-Zehnder modulator (MZM)s, and combining the resulting optical signals in a PD. This is leads to a convoluted setup which poses a number of practical difficulties, both in a real application setting and a lab setting. Hence an optical readout is best implemented on-chip, with the weighted optical signals getting combined on chip at a common PD or GC. In the latter case, the combined optical signal is then coupled out to an off-chip PD.

Different possibilities exist for the implementation of optical weights. Depending on the chosen implementation, integrated optical weights can have restrictions on weight resolution and modulation depth in contrast to digitally implemented weights. The effects of this and how this can be counteracted using specific algorithms are studied in detail in [84], for the case of non-volatile weights implemented in barium titanate (BTO).

Common building blocks in these implementations are configurable phase shifters, i.e. components which alter the lightwave's optical phase depending on the applied electrical signal. There are a number of phase shifter technologies of which we will now give some examples. Tungsten or doped silicon heaters, such as those offered in IMEC's iSiPP platform [85], affect the refractive index of the optical medium and thus the lightwave's optical phase by inducing a temperature change. While these heaters can be made compact, they are power hungry and susceptible to thermal crosstalk [86]. Micro-electromechanical systems (MEM)s utilize mechanical actuation of a perturbation close to the waveguide such as moving a silicon rail closer/further from the waveguide core, changing the effective index of the propagating optical mode [87]. Yet another alternative are phase shifters which employ BTO material. BTO is a ferroelectric material that induces a refractive index contrast via the Pockels effect. This effect can be described by the Pockels coefficient r_0:

$$\Delta n = -\frac{1}{2} n_0^3 r_0 E_{el}, \tag{4.7}$$

where Δn is the change in refractive index due to applied electric field E_{el} in a material with refractive index n_0 in absence of any electric field [88, 89]. For BTO the Pockels coefficient is very high $r_0 = 923pmV^{-1}$. An advantage of BTO weights is that these weights can be tuned in a non-volatile way, leading to reduced power consumption, using domain polarization switching. This is achieved by applying large field pulses to the BTO film. Depending on the pulse amplitude, pulse length and number of pulses, the domain configuration switches, resulting in a non-volatile change of the effective Pockels coefficient [88].

A possible weight implementation is an MZI with phase shifters on both arms (Figs. 4.5 (a) and 4.6). Single-arm steering, where only one MZI arm is equipped with a phase shifter is also a possibility (Fig. 4.5 (b)) [42]. In the case of single-arm steering, the MZI weight by itself can only control either the phase or the amplitude but not both simultaneously. This can however be easily remedied by making use of an additional phase shifter on the waveguide directly after the MZI (Fig. 4.5 (b)).

In that case, the MZI can be used for amplitude modulation and the phase can then be corrected by the additional phase shifter. The ability to adjust phase without adjusting amplitude can make training easier.

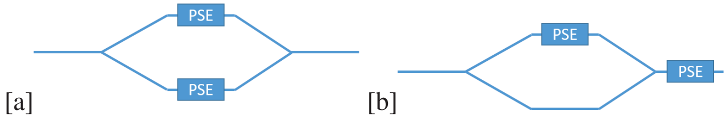


Figure 4.5: Schematic of types of weight implementations. PSE stands for phase shifter element, e.g. a tungsten heater or BTO material. (a) two-arm steering allows for both amplitude and phase modulation via the MZI with a more guaranteed good amplitude modulation depth at the expense of more challenging training. (b) single-arm steering of the MZI allows it to set the amplitude while the subsequent phase shifter after the MZI sets the phase.

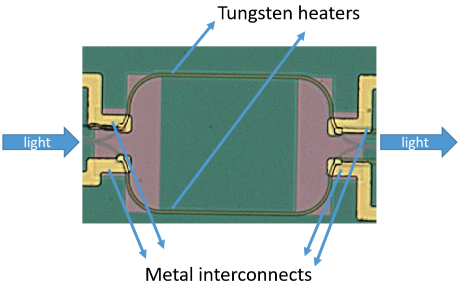


Figure 4.6: Implemented 2-arm steered MZI weight using tungsten heaters. The metal interconnects apply voltage/current over/through the tungsten heaters.

5

Simulations on multi-wavelength photonic reservoir computing

The goal of this chapter is to prove the possibility of WDM in PRC by way of simulations. First, we describe the two key aspects of the simulation framework, namely the photonic circuit simulator and the simulated architecture. The simulation results themselves are then introduced in two steps, starting with bit-level tasks like delayed XOR, before moving on to the signal equalization task. In both cases we describe how engineering the reservoir interconnection lengths or multiple-wavelength training can result in a single set of weights correctly processing multiple standard ITU-T WDM wavelengths and how readout performance can be made robust against practical variations a real-life application might encounter, in particular wavelength laser drift.

5.1 Photonic circuit simulators

There are many simulators for designing photonic integrated circuits, such as Aspic [90], Luceda Caphe [91,92], Ansys Lumerical Interconnect [93], and VPI Photonics [94]. All have their strengths and weaknesses. For the purposes of this work, it was important to be able to simulate the large amounts of data needed for machine learning quickly and preferably for multiple wavelengths in parallel. To this end, we opted to make use of a photonic circuit simulator named Photontorch [7], which was developed in our group.

Photontorch leverages the popular deep-learning framework PyTorch to reimagine photonic circuits as sparsely connected complex-valued neural networks. This allows for highly parallel simulation of large photonic circuits on graphical processing units in time and frequency domain. At the same time, all parameters of each individual component can easily be optimized with well-established machine learning algorithms such as backpropagation [95,96]. Photontorch's main advantage for our purposes was the fact that the state vector x is defined as a big monolithic multidimensional PyTorch tensor. The PyTorch tensor is a highly optimized array, which, as opposed to the more commonly used Numpy ndarray [97], can be placed on the graphical processing unit (GPU) of a computer. Its dimensions describe, apart from the number of memory-containing nodes, the different wavelengths (and/or modes) used in the simulation. The scatter matrices, describing the optical workings of components, and the connection matrix, describing the connections between components, can have an extra dimension for each wavelength as well, which allows for very fast simultaneous simulation for multiple wavelengths at once on a GPU.

Photontorch was developed by F. Laporte, but the author did contribute to some updates to Photontorch and helped write a collection of standardized code making use of Photontorch as a library. We would like to acknowledge Nvidia for supplying our research group with 4 Geforce GTX 1080 GPUs on which the simulations were performed.

Now that we settled on a simulation framework, let us move on towards the implementation of the PRC system in said framework.

5.2 Simulated architecture

A lot of work on PRC in general and on our type of reservoir in particular has already been done [38, 40, 98]. As was outlined in section 1.4, this work differentiates itself from and improves on what already exists by taking the first steps towards WDM to enhance the footprint-to-data-bandwidth ratio of our systems, which is one of the bottlenecks standing in the way of the system feasibility and commercial viability. A main point of note is that we will only use a single set of weights for all wavelengths. Indeed, having a separate set of weights for every wavelength would significantly increase chip footprint, electrical steering complexity and power consumption, thereby negating the aim of introducing WDM. We do not simulate true WDM in that we do not account for inter-wavelength crosstalk mechanisms. For these first steps towards WDM the focus is on exploring how this single-readout system may be achieved in this simplified scenario.

The simulated system is shown schematically in Fig. 5.1. It is an implementation of the four-port architecture as described in section 4.2 and Fig. 4.2. The nodes are implemented as 3x3 MMIs, which are represented in the simulations by their scatter matrices. A multiple input node strategy as discussed in 4.4 was followed. The interconnects between the nodes are implemented as integrated waveguides which are characterized by their optical loss, their effective refractive index and their group index to account for loss, phase and time delay respectively. The integrated waveguide on which our implementation was based, was a strip waveguide with a width $w = 0.45\mu m$, thickness $h = 0.22\mu m$, loss = 150dB/m, $n_{\text{eff}}(1550nm) = 2.28$ and $n_g(1550nm) = 4.56$. First-order dispersion of n_{eff} is considered.

As was explained in section 4.5, the phase change introduced by the interconnections will be considered as a uniformly distributed random variable. These random phases are set during the initialization of our simulated reservoirs.

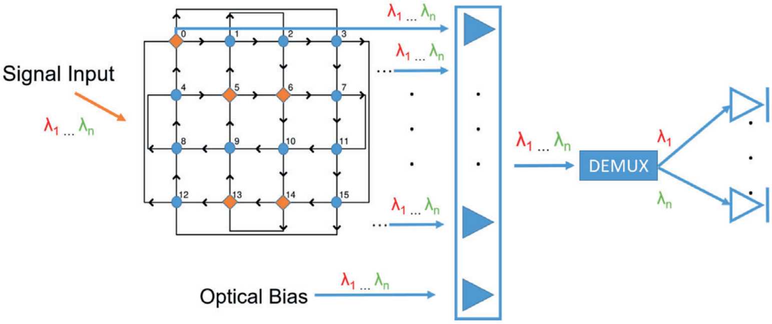


Figure 5.1: Schematic of the simulated system. Signal input in injected in the orange diamond shaped nodes. All nodes are connected to the readout. Arrows indicate propagation direction of the signal throughout the reservoir. DEMUX stands for wavelength division demultiplexer. A DEMUX could for instance be implemented using an AWG.

For the readout, we simulate an integrated optical readout, meaning the weights will manipulate both the amplitude and the phase of the signals. These signals, represented by complex-valued signals, will interfere on the detector. Our simulated weights have no restrictions on weight resolution and modulation depth, in contrast to physically implemented weights. For ways to deal with these restrictions in a physical implemented system using specific algorithms, we refer to [84], for the case of non-volatile weights implemented in BTO.

Our demultiplexer component is considered ideal in the simulations, i.e. no crosstalk between wavelengths or added optical loss.

The photodetector is implemented as a low-pass butter filter with filter order 4 followed by a transimpedance amplifier (TIA). It converts input optical power to a voltage value. The responsivity of the PD is 0.5 A/W, the transimpedance of the TIA Z_T is a 1000 Ω. We account for PD shot noise with variance VAR_{shot} and thermal noise with variance $VAR_{thermal}$ as well as noise added by the TIA with variance VAR_{TIA}:

$$\begin{aligned} VAR_{shot} &= 4kT/R_{load} f_{cutoff} \\ VAR_{thermal} &= 2q(I_{signal} + I_{dark}) f_{cutoff} \\ VAR_{TIA} &= 4kT/Z_T f_{cutoff} 2\pi Z_T^2 \end{aligned} \tag{5.1}$$

Here, k stands for the Boltzmann constant 1.3806488e-23 J/K, T stands for the temperature and is set to 300K, R_{load} stands for the load resistance and is set to 50Ω, f_{cutoff} stands for the cutoff frequency and is set to 40GHz, q stands for the elementary charge 1.6e-19 C, I_{dark} stands for the dark current and is set to 3nA.

During simulations we employed a simulation timestep of 1/640GHz. This means the simulation rate was 20 times more granular than the used bitrate of 32Gbps. Simulations showed that further increasing the simulation rate past this point had no impact on system performance.

Now that we have established the implementation of our simulated system, we move on to discuss the tasks we wish to solve with the simulated system, starting with the delayed XOR task.

5.3 Delayed 2-bit XOR task

The first task we will study is an example of a bit-level task. These refer to tasks where the desired output should be high for certain bit sequences and low for others. They are simple to define and thus a good entry point for our simulated system. Bit-level tasks can be balanced or unbalanced, depending on whether the number of combinations with high desired output equals those with low desired output. In the case of unbalanced tasks, we need to take into account that the baseline performance of random guessing to beat should no longer be a 50% bit error rate (BER). Indeed, a system that learns to always output the most likely output class, can potentially get low BER in a severely unbalanced scenario, but this can of course not be viewed as a success. Bit-level tasks also vary in whether they are linear or nonlinear. The two bit sequences [1,1],[1,0] are linearly separable from [0,0],[0,1], but this is not the case for [1,0],[0,1] and [0,0],[1,1], as can be easily seen by drawing them in a 2D diagram as shown in Fig. 2.3 (a). A nonlinear task will obviously require sufficient system nonlinearity, either provided in the reservoir itself or at the readout. In addition, increasing the number of bits in the targeted bit sequences increases the number of bit sequences to be recognised by the readout, and thus the difficulty of the training, as well as the memory requirements of the reservoir.

Of particular interest is the so-called delayed 2-bit XOR task, which we choose as our first task to test our simulated system. This delayed 2-bit XOR task consists of performing the Boolean XOR operation using the current and previous bit of an idealized OOK 32 Gbps optical signal. This task allows to confirm a PRC system's memory and nonlinear capability.

Let us now take a look at how we will train and test our simulated system.

The readout weights are trained on a training bit stream of 1000 bits using the Adam optimization algorithm, as implemented by the PyTorch Python library based on [55, 99], with as error metric the MSE as defined by equation 2.4.

The threshold, according on which outputs are classified as 1 or 0, is optimized for the lowest possible BER on the train data processed by this trained readout. The trained readout and optimized threshold are then applied to the states of the reservoir for a separate test bit stream of 100,000 bits in order to compute the test BER. Since we use 10^5 bits in our simulations, good practice is to limit the resolution of and crop the BER at 10^{-3}, i.e. 2 orders of magnitude higher than the lowest BER one can find in the simulation by simple counting [100].

To investigate if reliable performance can be achieved despite fabrication errors, 10 different reservoirs, each with different random phases, are always trained and tested.

Having settled on a task and a procedure with which to train and test it, we will now discuss how wavelength impacts performance of PRC systems and a method on how to address this by tweaking the length of the interconnections.

5.3.1 Engineered interconnection lengths

As a first step, let us study how and why performance of our PRC system varies with wavelength. System performance normally goes down when varying the wavelength because of the resulting variation in phase shifts in the waveguide interconnections. This leads to altered signal mixing for which the readout was not trained, leading to an incorrect weighting and recombination of node outputs.

However, when all interconnections are of identical length, there will be frequency changes for which the corresponding phase shift variation equals an integer multiple of 2π. Thus, the reservoir phase information stays unchanged for these frequency shifts and the readout performance is unchanged. The frequency change inducing a 2π phase shift is approximately constant, with variation being caused by dispersion. This gives rise to approximate frequency periodicity in readout performance (Fig. 5.2). By engineering the waveguide interconnection length, one can ensure that the frequency spacing between desired WDM channels corresponds to this period [16].

The term engineered interconnection lengths for the purpose of this work always refers to interconnections of identical length, with this length being chosen such as to achieve a desired frequency periodicity in readout performance.

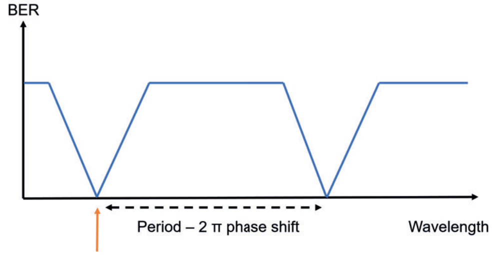


Figure 5.2: Exploiting engineered interconnection length. Training only occurs for one wavelength, as indicated by the orange arrow.

An important boundary condition when engineering the interconnection length, is that the interconnection length should not surpass the distance that light can travel during a single bit period (d_{max}). This causes previously injected bits to be in transit in between nodes, hidden from the readout. These bits cannot contribute to the output until they reach the next nodes. Therefore, the system has gaps in its memory, which is detrimental for tasks requiring memory, such as the 2-bit delayed XOR task studied here. There is thus an upper limit imposed on the interconnection length, which depends on the input bitrate. This in turn imposes a lower limit on the frequency periodicity. For the bitrate B of 32 Gbps, employed for the bit-level tasks for all simulation work, using the following equations:

$$d_{\mathrm{max}} = \frac{c}{B n_{\mathrm{g}}} \tag{5.2}$$

$$\frac{2\pi n_{\mathrm{eff}}(\lambda)}{\lambda} d_{\mathrm{max}} = \frac{2\pi n_{\mathrm{eff}}(\lambda + \Delta\lambda)}{\lambda + \Delta\lambda} d_{\mathrm{max}} + 2\pi \tag{5.3}$$

The maximum interconnection length works out to $d_{\mathrm{max}} \approx 2$ mm and the minimum wavelength shift $\Delta\lambda \approx 0.257$ nm. Here, c stands for the speed of light and n_{eff} is the effective index of the waveguide.

Additionally, the spectral width of the signal due to modulation also sets a lower bound on the frequency spacing of the wavelength channels.

We now test this method using our delayed 2-bit XOR task. Results, using the minimum periodicity of 32 Gbps, are shown in Fig. 5.3.

At 1552.5244 nm, and the wavelengths separated from it by the period, a BER $< 10^{-3}$ is achieved for all reservoirs.

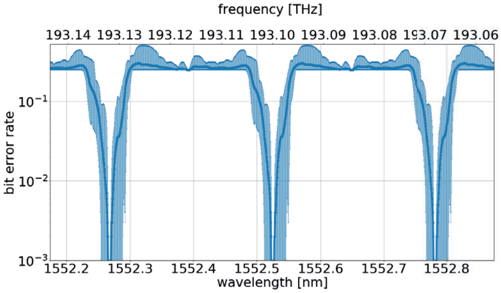


Figure 5.3: Performance when using engineered interconnection lengths for the delayed 2-bit XOR task. Error bars indicate minimum and maximum achieved BER over 10 different reservoirs each with their own fabrication errors.

This method of engineering the reservoir offers the possibility of achieving WDM for a large number of wavelength channels with the caveat of course that it offers limited flexibility, as one is bound to the set wavelength channel spacing. Let us now take a look at a more flexible approach of achieving a single-readout system handling multiple wavelengths, by taking into account multiple wavelengths during training.

5.3.2 Multiple-wavelength training

In this section, we present another method for achieving a single-readout system handling multiple wavelengths, based on minimizing the MSE for multiple wavelength channels simultaneously (Fig. 5.4).

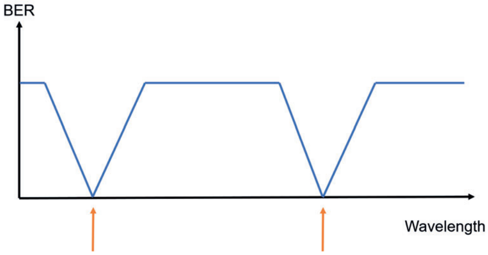


Figure 5.4: Illustrating multiple, two in this case, wavelength channel training. Training occurs at two wavelengths as indicated by the orange arrows.

Taking into account more than one wavelength during training poses a greater challenge as a machine learning task. However, for the tasks we studied here, it turns out that this is not prohibitive and performance can be as good as that achieved for single wavelength training.

Results are illustrated in Fig. 5.5, for the delayed 2-bit XOR task trained for the standard ITU-T WDM wavelength channels 1552.4239 nm and 1552.5244 nm.

The results show that the readouts achieve good performance at both trained wavelength channels, for all 10 different reservoirs.

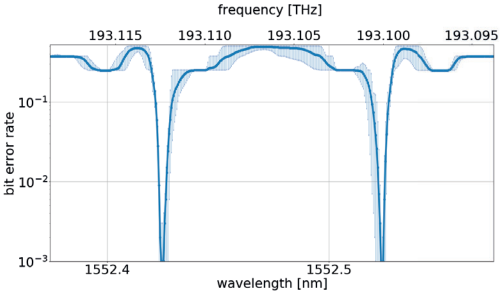


Figure 5.5: Multiple-wavelength training for 2-bit delayed XOR task. Wavelength channels achieve 0.2% *and* 0.1% *mean BER respectively. Error bars indicate minimum and maximum achieved BER for 10 different reservoirs with their own simulated fabrication errors.*

These results are at the heart of what we are trying to achieve in this thesis. Let us thus also look at another task to confirm the method's robustness. We again choose a bit-level task, this time a bit sequence recognition task. For such recognition tasks, the output needs to be 1 when observing certain target sequences of an idealized OOK 32Gbps optical signal, and 0 for the non-target sequences. We choose the bit recognition task to be 3 bits long and balanced with [111], [110], [101] and [011] as the target sequences. Results displayed in Fig. 5.6 again show that we achieve good performance at both trained wavelength channels for all 10 different reservoirs.

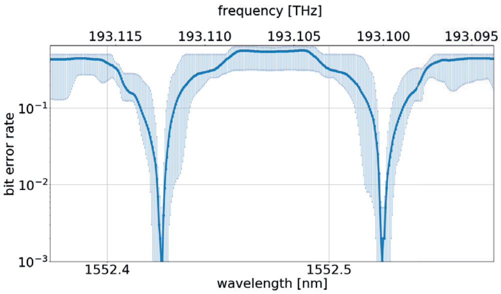


Figure 5.6: Multiple-wavelength training for balanced bit sequence recognition task. Both wavelength channels achieve 0.2% *mean BER. Error bars indicate minimum and maximum achieved BER for 10 different reservoirs with their own fabrication errors.*

This method is promising in achieving WDM in PRC. However we are limited in the number of wavelength channels the readout can handle at once. Starting from 3 wavelengths the training seems to become too difficult, as shown in Fig. 5.7.

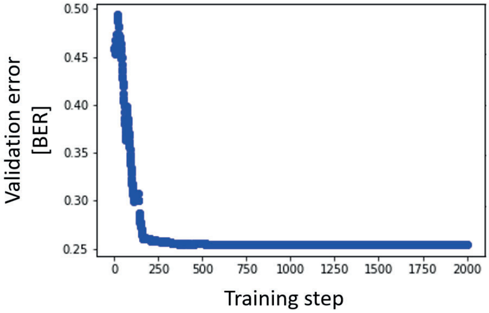


Figure 5.7: Evolution validation data BER during training with three wavelengths for one particular architecture initialization. The training fails to achieve a BER below 25% on the validation data, so this task is too challenging.

Results in this and the previous section are promising, but we still have the issue that there is only a very narrow wavelength range around the targeted wavelength channels over which good performance is achieved. We will now address this.

5.3.3 Mitigating wavelength laser drift

The input laser wavelength can drift over time, moving away from the optimal performance point, which can be a problem if the operating range is rather narrow. Therefore, in this section, we aim to enlarge the wavelength range over which good performance is achieved. Note that we suppose that this drift takes place on timescales much larger than the bit period. If not, the reservoir dynamics would be significantly altered compared to those used for training the readout, which would be detrimental to the performance.

As there is no closed-form analytical relation between this stable operating wavelength range and the readout weights, it is not possible to directly train the readout for this metric. Instead, we do something similar as in the previous section 5.3.2, and train the readout to minimize the MSE for multiple wavelengths symmetrically situated around the targeted wavelength channel. The main difference is that now these wavelengths are closely spaced around a single target WDM channel, rather than each training wavelength corresponding to a different WDM channel.

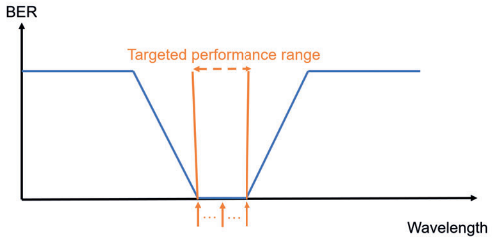


Figure 5.8: Improving good performance wavelength range. The orange arrows indicate the wavelengths that are trained. For a bit-level task such as our 2-bit delayed XOR task, it is common that only training 3 wavelengths (the center and extremes of the targeted good performance range) is necessary. The orange boundaries indicate the targeted wavelength range over which to achieve good performance.

The number of wavelengths trained and the wavelength spacing between them need to be considered. For some reservoirs it suffices to train the target wavelength channel and the extremes of the targeted wavelength range. For other reservoirs, multiple closely spaced intermediary wavelengths are trained, so as to maintain consistent good performance over the targeted wavelength range. The difficulty thus lies in maximising the width of the wavelength range over which consistent good performance is achieved, without decreased performance in between trained wavelengths.

Results are displayed in Fig. 5.9 for two specific reservoir initializations for a nonlinear bit level task, namely the 2-bit delayed XOR task.

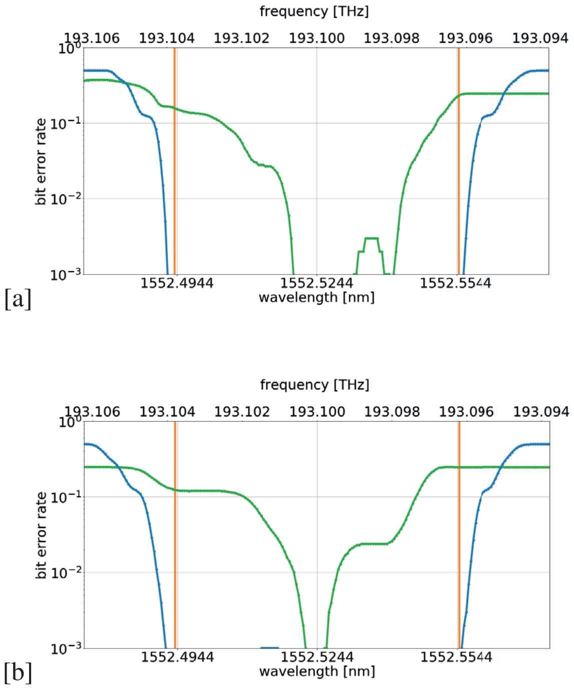


Figure 5.9: Performance for the delayed 2-bit XOR task. Each figure corresponds to a particular reservoir initialization with its own random phases in the interconnects. Green indicates training of only the targeted wavelength channel and blue indicates 3-wavelength training. The orange boundaries indicate the targeted wavelength range over which to achieve good performance.

For the delayed 2-bit XOR task, averaged over 10 different reservoirs, good performance, defined as $< 1\%$ bit error rate (BER), is achieved over a wavelength range of 63.1 pm or a corresponding frequency range of 7.8 GHz in case of multiple wavelength training. This is a significant improvement compared to the 24.5 pm and 3 GHz ranges in case of 1-wavelength training. For comparison, the commercially available Menara networks 5ZR0A00-TNBL 50GHz C-band tunable transceiver lists its wavelength stability after startup as ± 25 pm, thus a total wavelength variation of 50 pm [101].

In addition, a high sensitivity to wavelength implies high sensitivity to environmental effects (e.g. temperature), which also act on the reservoir phase information by impacting the interconnection waveguides refractive index. This enlarging of the good performance wavelength range thus is useful for practical applications beyond just wavelength drift considerations.

We will now propose to combine the methods of sections 5.3.1 and 5.3.3 to achieve a single-readout system performing well at a high number of wavelength channels, combined with a high degree of robustness to input wavelength variation.

5.3.4 Robust WDM operation

Fig. 5.10 (b) shows that a combination of training multiple wavelengths and engineering the reservoir interconnections, as shown in Fig. 5.10 (a), is indeed successful.

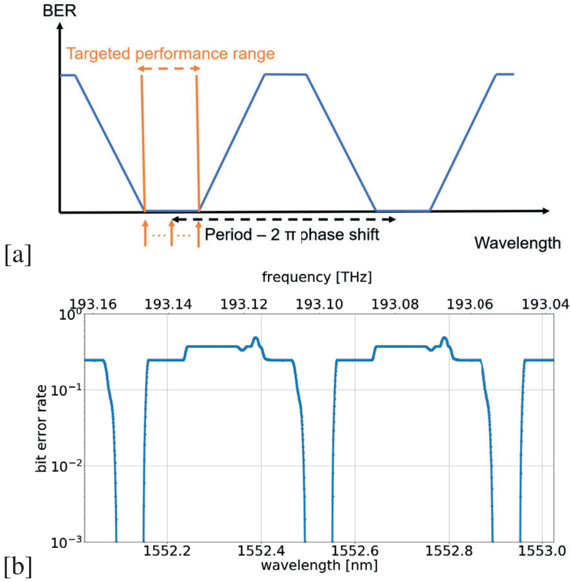


Figure 5.10: Robust WDM operation (a) The orange arrows indicate the wavelengths that are trained. For these particular reservoirs and tasks it was again sufficient to only train the target wavelength and the extremes of the targeted wavelength range over which to achieve good performance. Some other studied reservoirs needed inclusion of additional wavelengths in the targeted wavelength range during training to maintain good performance over the entire range. (b) Performance for the delayed 2-bit XOR task for one particular reservoir. Other reservoirs showed similar characteristics with some variation in the width of the good performance wavelength range. All achieve $< 10^{-3}$ *BER at target wavelengths.*

This combined approach results in a system where input wavelength variation is taken into account during training and where processing power is greatly increased without the need for an increased system footprint. By broadening the operating range of the readout, any imperfection of the performance periodicity (e.g. due to dispersion) is also mitigated. This allows for multiplexing many wavelength channels even with large wavelength spacing.

Dispersion ultimately limits the number of WDM channels for which good performance can be achieved. In practice, it turns out that we can easily accommodate tens to hundreds of 50-GHz-spaced channels, with 1552.5244 nm as the central channel, when using this combined technique for larger good performance wavelength range. This is demonstrated in Fig. 5.11, where the BER for 50-GHz-spaced channels is displayed. It is clear that the BER remains near optimal several THz away from the central channel for which the interconnection length was designed. Note that for clarity we only show the BER at these channels. In between them, the BER has a similar evolution as in 5.10.

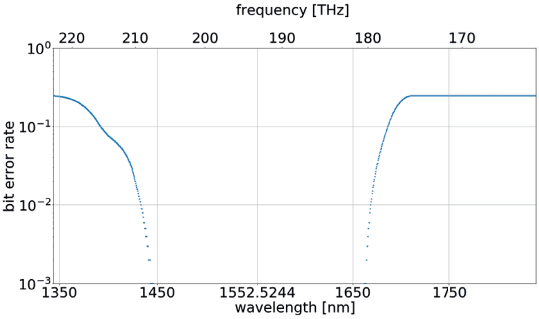


Figure 5.11: BER at target channels indicates robust single-readout operation (BER outside of the target channels not plotted) for the delayed 2-bit XOR task and the reservoir of Fig. 5.10[b]. Dispersion only significantly impacts performance after >1 THz from the designed wavelength channel. This corresponds to many tens to hundreds of DWDM channels.

We have now successfully used our simulations to showcase the possibility of WDM in PRC and have already started thinking about practical difficulties we might encounter in a real application, by studying the input wavelength variation problem, which is linked to the wider issue of readout phase information sensitivity. However, we still limited ourselves to bit-level tasks up until now and these have only so many application possibilities. We will now expand to a much more practical and commercially relevant application in industry, namely that of signal equalization.

5.4 Signal equalization task

Signal equalization aims to eliminate the (nonlinear) distortion of an optical signal after propagation through an optical link. This distortion consist of a linear contribution due to dispersion and a nonlinear contribution due to the dependence of the refractive index on the electric field strength of the travelling light pulse, i.e. the Kerr effect, which is one of the fundamental problems at the high bandwidths that are being targeted in this work. Especially at high powers, the Kerr effect can be responsible for multiple nonlinear optical effects, like self-phase modulation (SPM), cross-phase modulation (XPM) and four-wave mixing [102]. Signal equalization is currently typically handled by electronic Digital Signal Processing (DSP) chips, but these are costly and struggle at ever increasing data bandwidths.

For the nonlinear signal equalization task, we use as data a simulated OOK 10 Gbps optical signal that has travelled through 2000km of dispersion- and loss-compensated optical fiber with Kerr nonlinearity and thus SPM. Simulations for acquisition of this data were performed using VPIphotonics Design Suite [94]. The target data is the perfect uncorrupted OOK bitstream, consisting of ’0’ and ’1’ values, to which we will compare our PD output after thresholding. Let us retrace our steps from the earlier sections for this new task, note that our training and testing procedure remain unchanged.

5.4.1 Engineered interconnection lengths

We start with engineering the interconnection lengths which leads to similar results as before (Fig. 5.12).

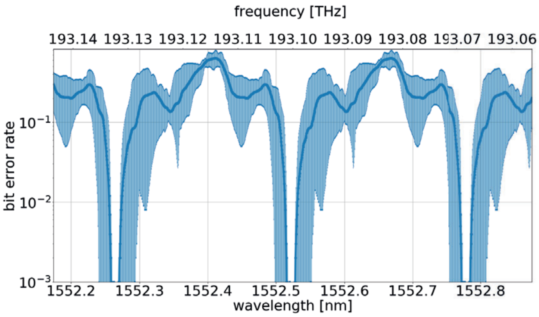


Figure 5.12: Performance when using engineered interconnection lengths for the signal equalization task. Error bars indicate minimum and maximum achieved BER over 10 different reservoirs each with their own fabrication errors.

When considering signal equalization, it can also be helpful to take a look at the eye diagrams of the distorted and equalized signals as shown in Fig. 5.13.

The eye diagrams here are constructed by dividing the OOK signal in time segments of 3 bit periods long and folding these segments over each other. For each pixel in the eye diagram, the color bar represents the number of times a time trace passed through it.

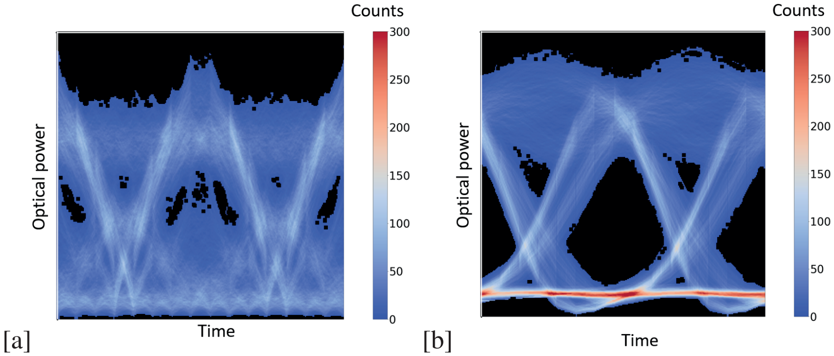


Figure 5.13: Nonlinear signal equalization. (a) Eye diagram for original data stream after 2000km long dispersion- and loss-compensated optical fiber with Kerr nonlinearity. (b) Eye diagram after reservoir for wavelength 1552.5244 nm.

5.4.2 Multiple-wavelength channel training

Next up is again taking a look at multiple wavelength channel training, now again with as goal not to increase the good performance wavelength range but to achieve good performance for 2 channels (Fig. 5.14). Again our results are similar to those achieved for the XOR task.

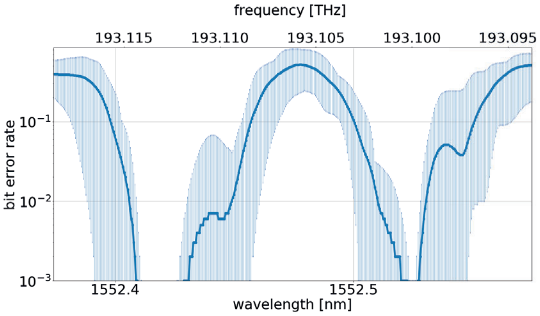


Figure 5.14: Multiple-wavelength channel training for signal equalization task. Wavelength channels achieve $< 10^{-3}$ mean BER. Error bars indicate minimum and maximum achieved BER for 10 different reservoirs with their own simulated fabrication errors.

In Fig. 5.15 we visualize the eye diagrams of nonlinear signal equalization performed for 2 wavelengths. It is clear that compared to the original eye diagram, the eye diagrams at both wavelengths are much improved.

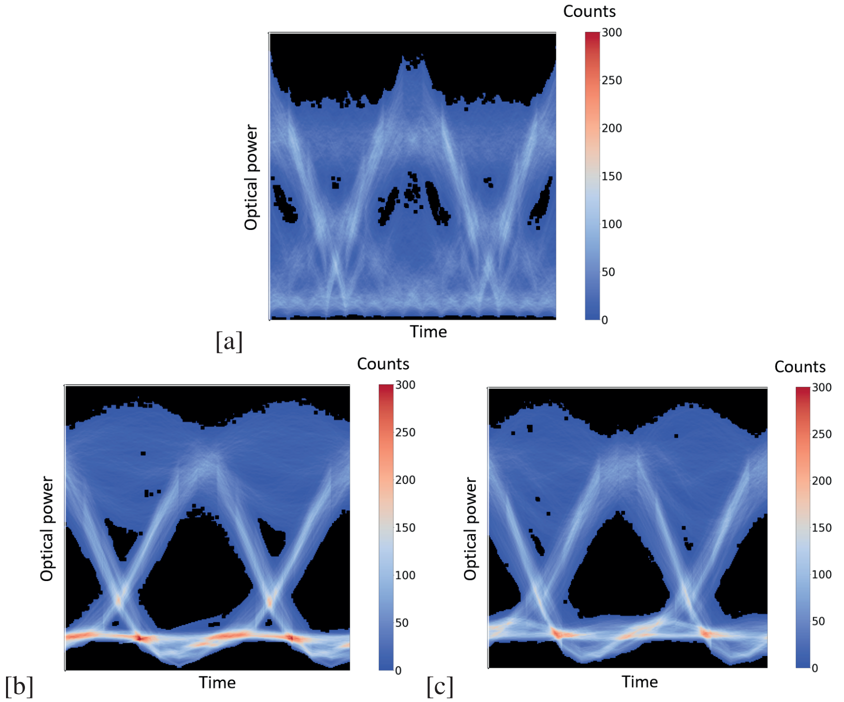


Figure 5.15: Multiple-wavelength training for nonlinear signal equalization. (a) Eye diagram for original data stream after 2000 km long dispersion- and loss-compensated optical fiber with Kerr nonlinearity. (b) Eye diagram after reservoir for wavelength 1552.4239 nm. (c) Eye diagram after reservoir for wavelength 1552.5244 nm.

5.4.3 Mitigating wavelength laser drift

For mitigating the wavelength laser drift it turns out that the results are largely similar but that this task appears to benefit more often from some more training wavelengths (5 instead of 3) in the good performance wavelength range (Fig. 5.16).

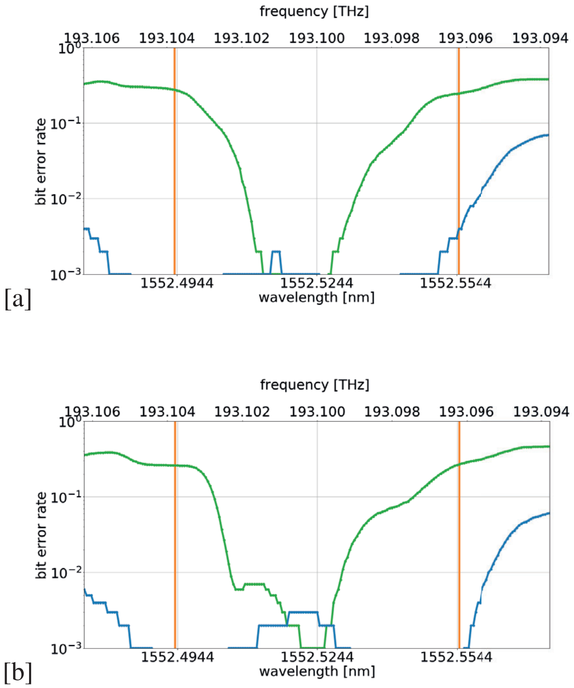


Figure 5.16: Performance for the signal equalization task for 5 training wavelengths. The 5 wavelengths consisted of the center and extremes of the targeted range and two intermediary wavelengths. Each figure corresponds to a particular reservoir initialization. Green indicates 1-wavelength-training and blue 5-wavelength-training. The orange boundaries indicate the targeted wavelength range over which to achieve good performance.

For the nonlinear signal equalization task, averaged over 10 different reservoirs, good performance, again defined as $< 1\%$ bit error rate (BER), is achieved over a wavelength range of 95.4 pm or a corresponding frequency range of 11.9 GHz in case of multiple wavelength training. This is a significant improvement compared to the 41.0 pm and 5.1 GHz ranges in case of 1-wavelength training.

5.4.4 Robust WDM operation

Once again we can make use of engineered interconnection lengths in tandem with our technique to improve the good performance wavelength range to achieve good performance at many channels, using a single-readout system, with attention to readout phase information sensitivity (Figs. 5.17 and 5.18).

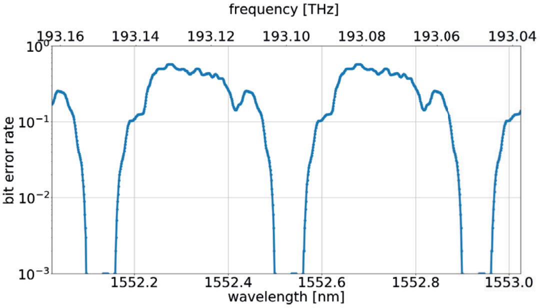


Figure 5.17: Robust WDM operation performance for the nonlinear signal equalization task for one particular reservoir. Other reservoirs showed similar characteristics with some variation in the width of the good performance wavelength range. All achieve $< 10^{-3}$ *BER at target wavelengths.*

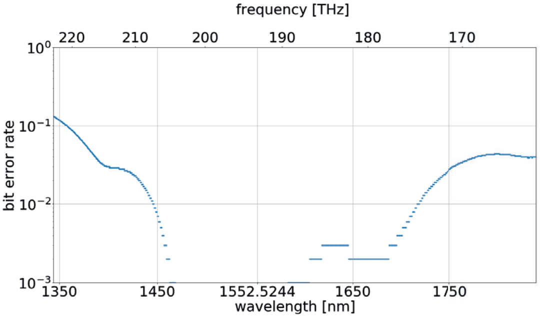


Figure 5.18: BER at target channels indicates robust WDM operation (BER outside of the target channels not plotted) for the nonlinear signal equalization task and reservoir of 5.17. For both this case and all other reservoirs, dispersion only significantly impacts performance after >1 THz away from the designed wavelength channel. This corresponds to many tens to hundreds of DWDM channels.

With these results on the signal equalization task, we have successfully used simulations to lay the groundwork of this thesis. However, to truly prove the possibility of WDM in PRC for real life applications, we need to move beyond simulations to an experimental setup using a real on-chip integrated circuit. This is the focus of the next chapter.

6

Experimental work on multi-wavelength photonic reservoir computing

This chapter focuses on experimental measurements for both a non-integrated electrical readout and an integrated optical readout. For each case, we describe the used measurement setup, the performed experiments and their results and the conclusions. For the integrated optical readout, special attention is given to the specific type of optical weights used, namely BTO-based weights.

6.1 Non-integrated electrical readout

6.1.1 Measurement setup

Before one can obtain qualitative measurement results, one needs a good measurement setup, which we will introduce in this section. The setup, used for non-integrated readouts, is shown in Fig. 6.1. In this case, the readout is implemented digitally on the computer.

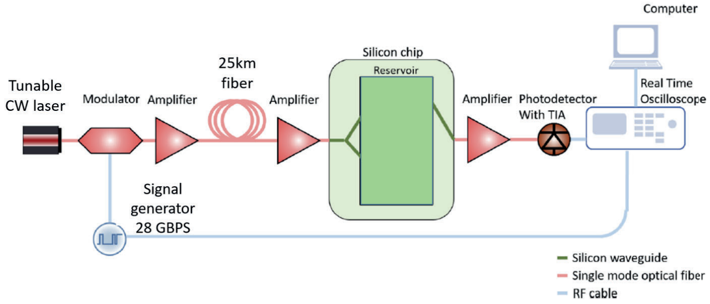


Figure 6.1: Schematic illustration of the setup used in the experiments with non-integrated readout. For a non-integrated readout, the readout is implemented digitally on the computer. Figure taken from [40] with adjustments. The signal generator was a Keysight M8195a arbitrary waveform generator and the real time oscilloscope was a Teledyne LeCroy SDA 830Zi-B for which we used the 160 Gsps ports.

For the signal equalization task, a single-mode fiber of 25 km with an attenuation of 0.2 dB/km at 1550 nm was added between the modulator and chip, in order to provide signal distortion. The signal is amplified before being sent through the fiber in order to increase nonlinear distortion effects. By setting the fiber input power, we can specify the difficulty of the equalization task to be solved by our reservoir computer, as can be seen from the eye diagrams in Fig. 6.2.

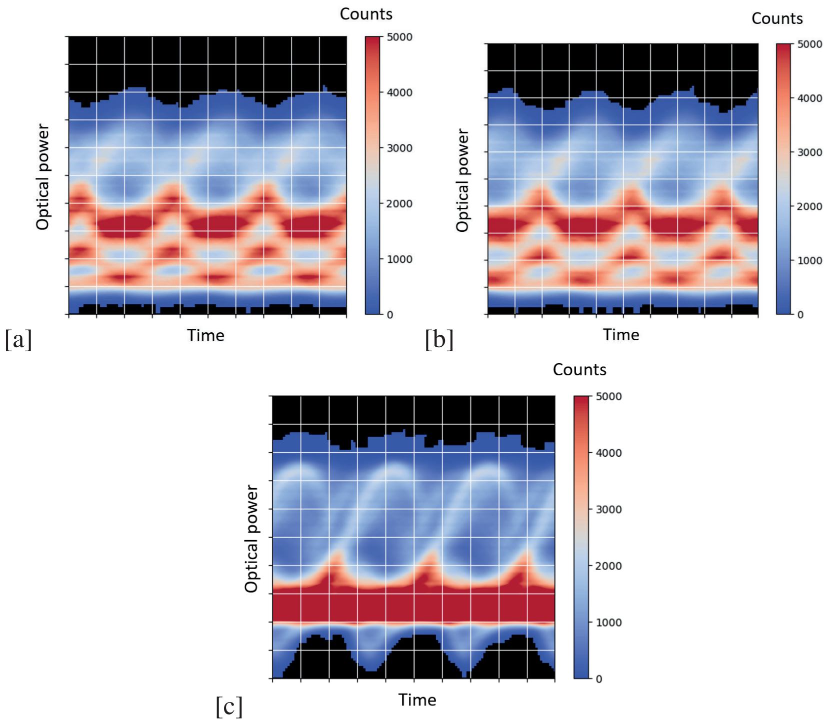


Figure 6.2: Eye diagrams at the output of the fiber for different input powers. (a) 5 dBm (b) 10 dBm (c) 18 dBm

A second amplifier before the chip is set in such a way that the optical power at the input of the reservoir is the same for all experiments. This is to ensure that experiments for different fiber input powers are more consistent, e.g. having identical expected node output powers helps with aligning fibers in an identical way, which ensures a fairer comparison of their performance. High chip losses, due to suboptimal design and fabrication of the MMIs, necessitate the use of a final amplifier after the chip, in order to boost the signal to within the detection range of the photodetector. All three amplifiers are power-clamped erbium doped fiber amplifier (EDFA)s. These amplifiers can also introduce some nonlinearity. In case of the amplifier preceding the chip, this leads to a more nonlinearly distorted signal. In case of the amplifier after the chip, this has the potential to enhance our PRC system's nonlinearity.

The input signal consists of approximately 100.000 bits, modulated using OOK at 28 Gbps. This was the highest modulation rate possible given the available equipment, as higher modulation rates lead to visible degradation in the measured input signal.

We will study 3 standard ITU-T 50 GHz separated wavelengths, namely 1552.122 nm, 1552.524 nm, and 1552.926 nm. It is important to mention that the experiments do not use a true WDM signal, where all wavelengths are injected simultaneously into the fiber, instead the different wavelengths are injected separately and measured sequentially. Although this neglects nonlinear crosstalk between the wavelengths, it is an important first step towards true WDM.

Now that we have covered our measurement setup, we can move on to the actual measurements.

6.1.2 Experiments with non-integrated electrical readout

The experiments presented in this section are all done using a so-called electrical readout strategy, with the weights being implemented digitally on a computer. The reservoir chip has 16 nodes in a 4x4 four-port architecture as shown in Fig. 4.2 (a). The nodes are implemented as 3x3 MMIs, which distribute the optical power in their input ports equally over their output ports (Fig. 6.3).

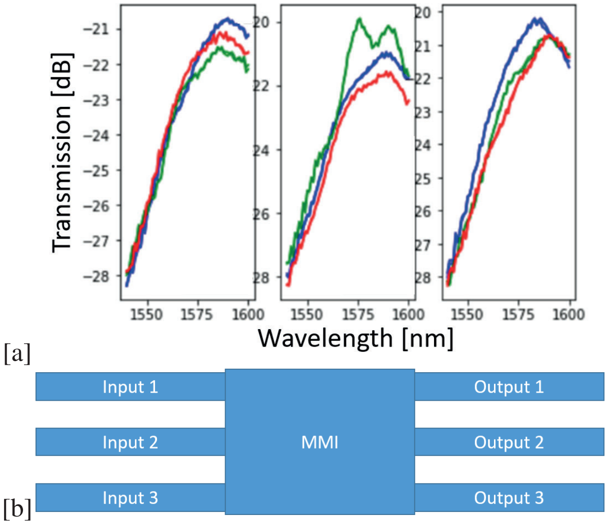


Figure 6.3: (a)Transmission for a 3x3 MMI node from input 1 (left), input 2 (middle), input 3 (right) to output 1 (blue), output 2 (green) and output 3 (red). Measurements performed by Stijn Sackesyn. Input power to chip was 0 dBm for each instance. Interfacing to and from chip was done via vertical grating couplers. (b) Illustration of how MMI ports for figure a were numbered.

Two inputs and two outputs are used for inter-node connections and the remaining input and output serve as interfaces to and from the reservoir. Input is injected in four nodes spread throughout the reservoir simultaneously by use of 1×2 splitters after the input grating coupler (Fig. 6.4). The chip is made on a silicon nitride platform (through Ligentec in the context of the European H2020 Nebula project), and has a footprint of $\approx 31mm^2$. The interconnection length between nodes was $\approx 2.1mm$ for the inner reservoir connections, $\approx 4.2mm$ for the long vertical outer connections and $\approx 6.3mm$ for the long horizontal outer node connections. These lengths were chosen, by the designer Stijn Sackesyn, so as to correspond to an integer times half a bit period of delay for our preferred bitrate of 28 Gbps. Simulations and past work such as that shown in Fig. 4.3 had shown this to be a sensible choice. 28 Gbps was the preferred bitrate as this was determined to be well within specification of our lab equipment such as the Keysight M8195a arbitrary waveform generator. The interconnection loss was $\approx 0.47dB/cm$.

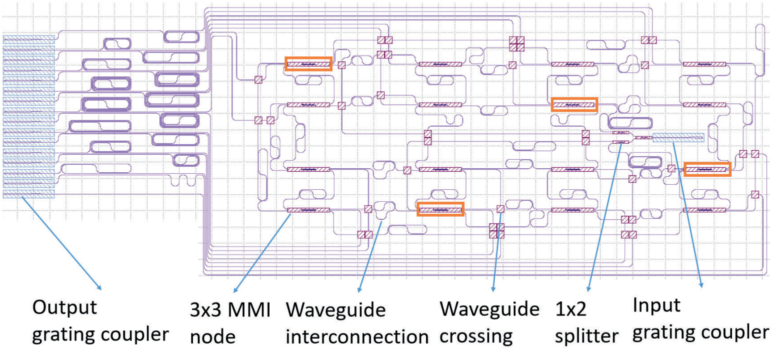


Figure 6.4: Four-port architecture photonic reservoir integrated on chip. Light is injected into the input grating coupler at the right. Using 1×2 splitters the light is then distributed over four nodes spread throughout the reservoir indicated by orange boxes.

The nodes are optically probed and detected by an external photodetector, one at a time. The entire input signal of 100 000 bits, modulated using on-off keying at 28 Gbps, is periodically repeated in order to accomplish this. The electrical time traces collected at each measurement are saved on a computer and the linear combination of these traces is done in post-processing on a computer. We repeat that we will only be using one single set of weights for all wavelengths, as explained in section 5.2. For a digitally implemented readout, doing so obviously does not result in chip footprint savings, simplified electrical steering or reduced power consumption. However, we see a non-integrated system as an intermediary step, and therefore, we try to keep as close as possible to an eventual integrated readout system.

Since we do not measure different nodes at the same time, but rather sequentially, we must align these signals to the bitstream given as input to the signal generator. We do this by realigning them with respect to a known header signal (a long string of 0 input bits) contained in the input signal, which can be recognized in every node signal. An optimizaton step is then done by optimizing a time offset for every node to obtain the optimal alignment of the nodes among themselves with respect to system performance. This can be done for instance by using a Tree of Parzen Estimators (TPE) algorithm [103], which minimizes the average BER across the used node signals for all wavelengths for a validation dataset. Other methods for temporally aligning the nodes were tried, such as a Bayesian optimization algorithm [104] and a brute-force method which sequentially went through all the nodes sweeping over a number of discrete time shifts. This brute-force method had the drawback that the order in which the nodes' temporal alignments were optimized influenced the outcome. To counteract this, the process had to be repeated several times, each time with a new random order in which to optimize the nodes, after which the outcome with the best BER was picked. The advantage of Sequential Model-Based Global Optimization (SMBO) algorithms such as the TPE and Bayesian algorithm is that they, in contrast to the brute force method, make an informed choice on which time offsets combination to test next using an expected improvement criterion. This leads them to find a good solution in far fewer iterations, an important consideration as evaluating a combination of time offsets involves a computationally expensive readout training and testing step. It also makes them less susceptible to the starting point used in contrast to the brute force method. For details on the expected improvement criterion used by these SMBOs we refer to [103, 104].

We ended up using a TPE algorithm as implemented in [105], as it seemed to give better results than the Bayesian optimization algorithm as implemented in [104] and was much more practical to use than the time-consuming brute force method.

The training of the weights consists of a ridge regression algorithm, as implemented by the sklearn Python library [106] in which normalization is allowed, and makes use of a cross-validation implementation. Indeed, the data for each wavelength is split into 10 folds, and in each iteration 9 folds serve as training data for the ridge regression and 1 fold serves as test data. In this way, after 10 iterations, all 10 folds were used for testing. The average and standard deviation of the BERs over the 10 iterations are then used to display the results. (The use of ridge regression instead of the more complicated Adam gradient descent algorithm as in [16] was possible because of the electrical readout only having real-valued weights, versus the complex-valued weights of an optical readout.) The labels used for training are the symbols in the digital on-off keying bit stream. The cross-validation data set consisted of 20 000 bits. For the optimization of the time offsets, a separate validation set consisting of 10 000 bits was used.

Note that the the temporal alignment is being optimized on the basis of the BER. The optimization thus seeks to find the best possible temporal alignment for each node, and not necessarily the actual temporal alignment that would exist between nodes in case we would be able to measure all nodes in parallel. In a system with an integrated electrical readout that is being read out in parallel, introducing such an additional optimal temporal alignment after chip fabrication is trivial. In a system with an integrated optical readout, an additional per-node delay would have to be taken into account during the chip design.

Finally, during post-processing, three additional delayed copies of the node signals (delayed with 1, 2 and 3 bits, respectively) are used in conjunction with the original measured node signal. This increases the number of effective outputs of the reservoir, and is similar to adding a feedforward equalizer in the electrical domain on top of the optical reservoir. In Fig. 6.5 it is shown how adding these delayed copies favorably impacts performance. Although each delayed copy has its own readout weight, we still opted to make these weights independent on wavelength, to align closer to a future optical readout implementation.

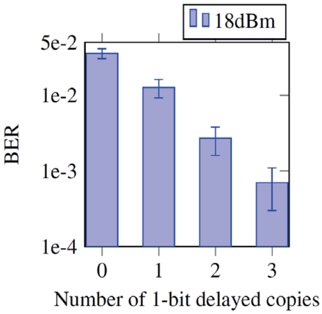


Figure 6.5: Average BER over the cross-validation iterations as a function of number of delayed copies. Error bars indicate the standard deviation of the BER over the cross-validation iterations. The fiber input power was 18 dBm and all 3 ITU-T wavelengths were included in training.

As in section 5.3.2, we take into account more than one wavelength during training (Fig. 5.4), which allows the readout to adapt to the phase dynamics associated with each wavelength. However, this poses a greater challenge as a machine learning task [16]. Fig. 6.6 compares the performance for different wavelengths in the case of only one wavelength being trained and in the case of multiple wavelengths being trained.

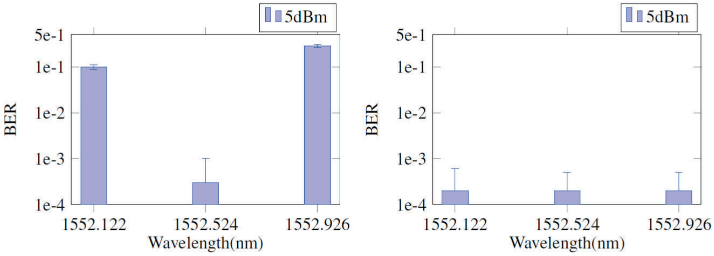


Figure 6.6: Average BER, with error bars indicating standard deviation over the cross validation iterations for single-wavelength training (left) versus multiple-wavelength training (right) at 5 dBm fiber input power. In the left case the wavelength for which time offsets were optimized and readout weights were trained was 1552.524 nm, whereas in the right case, all wavelengths were taken into account.

In the left figure, it is clear that the performance is only good for the wavelength that was trained (1552.524 nm). For the right figure on the other hand, all wavelengths shown were incorporated in the training set, and this results in good performance for all wavelengths. Again, we want to stress that we use the same weights for all the wavelengths.

The attentive reader will notice that the BER for the single trained wavelength in figure 6.6 (left) was slightly higher than that of any of the trained wavelengths in figure 6.6 (right) which is counter-intuitive. This can be attributed to the time offsets optimization step where the TPE algorithm is not always guaranteed to find the global optimal solution for a given number of optimization iterations.

The results in Fig. 6.6 were for a low fiber input power of 5 dBm, where we do not expect significant nonlinearity. We will now investigate the performance for higher input powers like 10 and 18 dBm. In Fig. 6.7, it is shown that the system can still perform well for at least up to 3 standard 50 GHz spaced ITU-T wavelengths at these higher levels of difficulty.

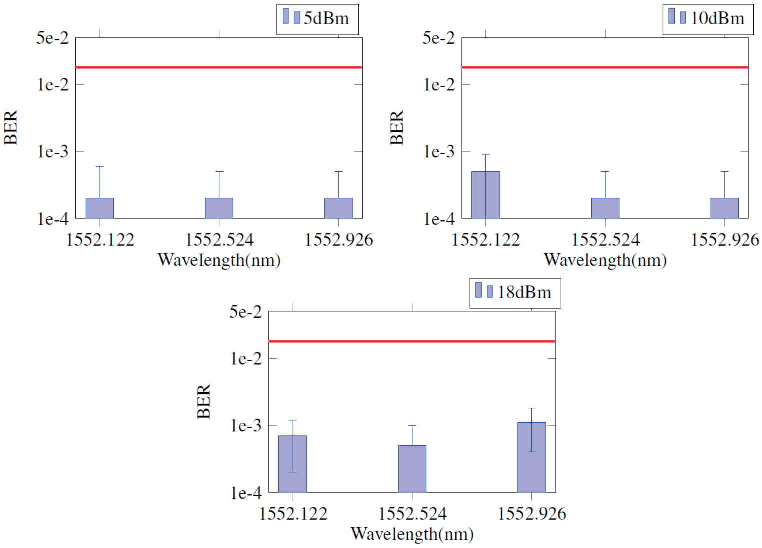


Figure 6.7: Average BER, with error bars indicating standard deviation over the cross-validation iterations for multiple wavelength training. All wavelengths displayed were taken into account for time offset optimization and readout weight training. Fiber input powers were 5 dBm, 10 dBm and 18 dBm. The horizontal red line illustrates the relevant FEC limit.

For all three wavelength channels, the BER stays well below $1.5\,10^{-2}$, which is the FEC limit of the P1-PTX-2-100G-WDM optical transport system [9]. This system is suitable as a reference, because it has strong dense WDM capabilities in the C-band and is therefore the type of interface that could be paired with this PRC system. Still, for the most difficult case at 18 dBm, the BER starts to degrade, and it becomes harder for the single set of readout weights to equalise all wavelengths equally.

Finally, we will compare the performance of our system with an electrical tapped delay line (TDL) filter. The PRC system used 16 nodes with three additional delayed copies of the node signals, which translates to 64 readout weights. In order to have a fair comparison, we will also use 64 taps for the TDL. Just as in [40], the reservoir outperforms such a TDL in the case of significant nonlinear distortion, i.e. when the input power is 18 dBm as shown in Fig. 6.8 (left). Again, all wavelengths processed by a single set of readout weights. Even the worst performing wavelength has a BER well below that of a TDL filter optimized for only one wavelength.

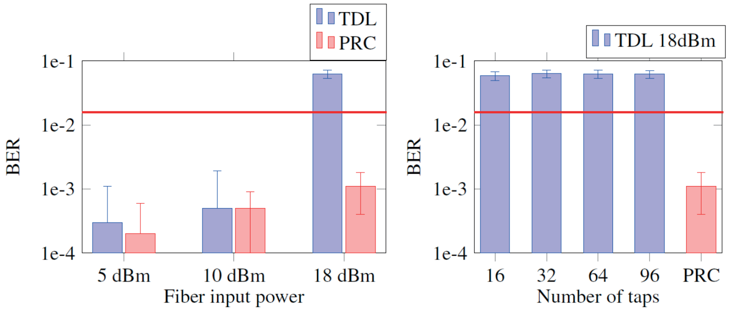


Figure 6.8: Average BER, with error bars indicating standard deviation, over the cross validation iterations for a tapped delay line, trained and tested for one wavelength (1552.524 nm) in comparison with worst performing single-readout PRC wavelength. (Left) For various levels of input power. (Right) For different number of tapped delay line taps at 18 dBm input power. The horizontal red line illustrates the relevant FEC limit.

Note that the TDL's inferior performance stems from its inability to equalize the nonlinear distortion due to the TDL lacking nonlinearity. In the reservoir, the combination of interference and a photodiode creates a quadratic nonlinearity and hence the capability to tackle nonlinear problems, as we have also seen from the work done on simulations. The TDL's inability to equalize nonlinear distortion is independent of the number of taps used, as can be seen in Fig. 6.8 (right).

6.1.3 Conclusions non-integrated electrical readout

We have experimentally taken a first important step towards WDM in an integrated waveguide-based photonic reservoir computing platform using only limited post-processing, i.e. a digitally implemented readout and the generating of delayed copies of measured node signals. When using the FEC limit of $1.5\,10^{-2}$ as a benchmark, the single-readout system was shown to work well up to at least 3 standard ITU-T wavelengths for 28 Gbps (nonlinear) signal equalization tasks. The reservoir outperforms a simple tapped delay line filter for higher levels of signal distortion. This confirms the capacity of the system to tackle nonlinear tasks. By reaffirming experimentally that the nonlinear problem solving capability of the system remains intact and that performance does not degrade to unacceptable levels when utilizing a single set of readout weights, we have shown the possibility of WDM in PRC systems.

We will now move on and investigate a system with an integrated optical readout focusing on a single wavelength channel, as a first step toward a future WDM system, as this type of readout is still at the early stages of research [42].

6.2 Integrated optical readout

In the integrated optical readout, all node signals were optically weighted on chip, both in phase and amplitude, before being combined to a single grating coupler and being coupled out of chip to an external photodiode (Fig. 6.9). The chip is once again made on a silicon nitride platform (through Ligentec in the context of the European H2020 Nebula project). The reservoir is a downscaled version of the one in figure 6.4 in that it only has 8 instead of 16 nodes. The interconnection lengths are still $\approx 2.1mm$ for the inner reservoir connections and $\approx 4.2mm$ for the long vertical outer connections.

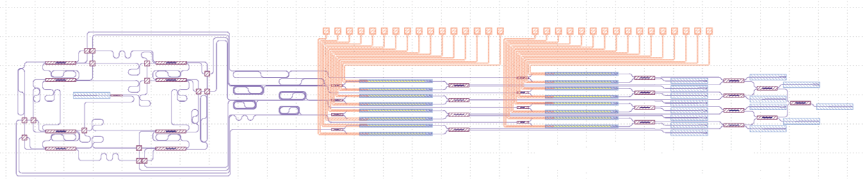


Figure 6.9: Reservoir (left of picture) with attached integrated readout (right of picture)

An integrated optical readout is more challenging in terms of chip design and experimental measurements. In addition, limiting factors compared to a digitally implemented electrical readout are weight resolution and modulation depth, as well as the loss of additional post-processing on node signals. However, the ability to weigh both amplitude and phase, in addition to exploiting the optical interference between node signals are important potential advantages compared to an electrical readout. This section will once again focus on the nonlinear signal equalization task used for the non-integrated electrical readout, albeit without WDM.

Let us now take a closer look at the integrated readout, more specifically at how the weights are implemented.

6.2.1 BTO optical weights

The optical readout in question is implemented by way of MZIs equipped with non-volatile BTO phase shifters on both arms (Fig. 6.10), as explained in section 4.6.3. The deposition of the BTO was done by Lumiphase [107] in the context of the European H2020 Nebula project.

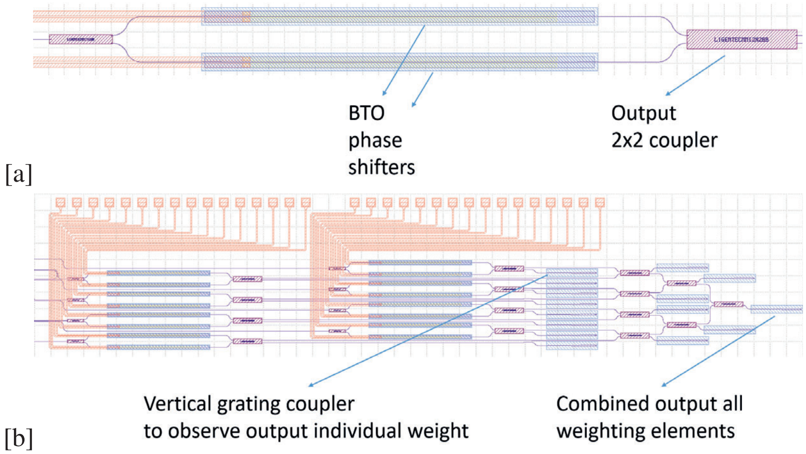


Figure 6.10: (a) BTO weighting element. From left to right: an MZI is equipped with BTO phase shifters on both arms. One output of the 2x2 coupler leads to a vertical grating or edge coupler, so that each channel can be individually observed. The other output connects to the tree that combines all weighted channels. (b) Configuration of all the optical weights in the integrated readout.

Of particular interest is the peak-to-peak (PTP) modulation depth of these optical weights (Fig. 6.11).

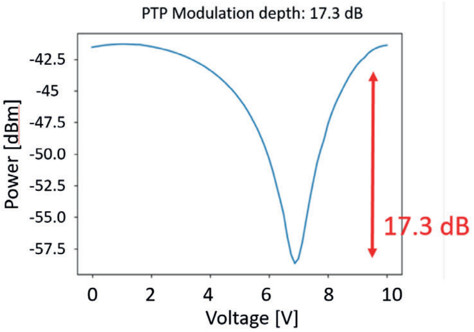


Figure 6.11: Modulation depth BTO weighting element measurement result.

The modulation depth showed great variation from weighting element to weighting element, even within the same chip, as shown in Fig. 6.12.

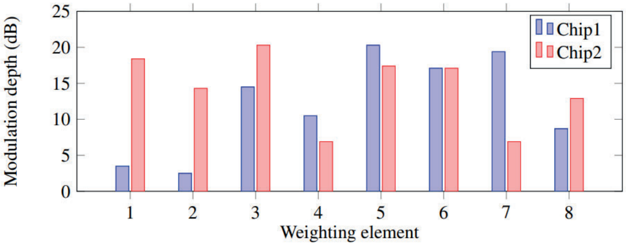


Figure 6.12: Modulation depth for 2 different chips each with 8 weighting elements. The modulation depth varies greatly form weighting element to weighting element.

The current hypothesis for this variation and often poor modulation depth is that there can be unbalanced signal recombination from both of the MZI arms. This could be caused by fabrication errors in the 2x2 output coupler or by asymmetric optical losses in the MZI arms due to non-uniformity in the BTO deposition. However, further study into this problem is required to come to a definitive answer. As seen in Fig. 6.13, these weights also exhibit some hysteresis, which can be attributed to ferroelectric domain poling. This is a further aspect that can complicate the training of these weights as the optimal voltage to apply to a weight can vary over the course of the training process, depending on its history.

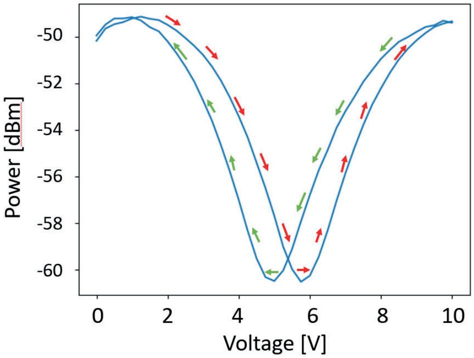


Figure 6.13: The BTO weighting elements show hysteresis in their output power when sweeping the voltage up and down.

Having reviewed the readout implementation, we move on to the measurements starting with a discussion on the measurement setup.

6.2.2 Measurement setup

Fig. 6.14 shows a detailed overview of the measurement setup used for online training on an integrated readout. Working with an integrated readout presents additional technical difficulties compared to the non-integrated readout of section 6.1.1 and as such we will be paying extra attention to the details concerning the configuration of this setup.

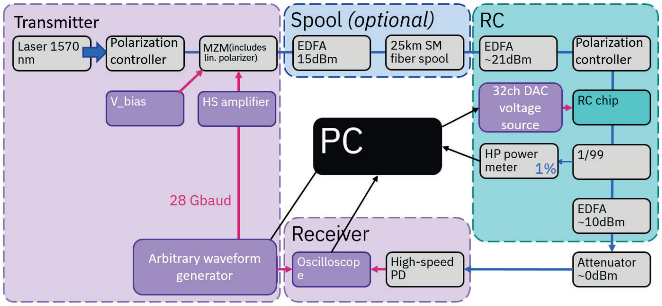


Figure 6.14: Schematic illustration of the setup used in the online training of integrated readout experiments.

An arbitrary waveform generator, a Keysight M8195a, is used to generate a 28 Gbaud electric signal that drives an MZM intensity modulator and thus modulates the continuous wave laser. Correctly biasing the MZM to operate around the center of its linear regime is crucial in getting a good signal on the oscilloscope, a Teledyne LeCroy SDA 830Zi-B. For the more difficult case of an integrated readout, it is advisable to amplify the signal generated by the arbitrary waveform generator so that roughly the entire linear range of the MZM is covered. That way, one gets maximum contrast between 0 and 1 input bits for the OOK signal. Additionally, replacing long RF electrical cables here by shorter ones can have a measurable impact on the signal quality.

The signal is then transmitted through a 25 km optical fiber spool, which distorts the signal both linearly (due to dispersion) and nonlinearly (due to self-phase modulation). The higher the input power injected by the preceding EDFA, the higher the nonlinear distortion in the fiber. Note that the fiber is removed when back-to-back system characterization is needed, such as when evaluating its intrinsic noise and optical losses.

The EDFA before the chip ensures a constant input power to the RC chip independent of the power injected in the spool, as was the case for the non-integrated readout setup. The 32-channel voltage source sets the voltages on the BTO weights of the RC chip. The EDFA and attenuator combination after the RC chip brings the output optical power to a suitable range for the high-speed PD. It is important to get the maximum acceptable input power to the PD without having to amplify the output signal too much, as every amplification step is a source of additional noise. Choosing the right EDFA and attenuator combination is thus crucial in getting a good SNR.

The signal from the PD is then captured by the real time oscilloscope (RTO). Note the Teledyne LeCroy SDA 830Zi-B has both 80Gsps and 160 Gsps ports, we used the 160 Gsps ports. The PD we used had a built-in TIA, which again is crucial in getting a good signal to the RTO.

One can see that there is quite a lot of amplification, both optical and electrical, going on in the setup. Due to input-output chip coupling losses, high reservoir losses and amplitude modulation introduced by the readout, the result is a setup where managing noise becomes of the utmost importance. The total loss amounts to ≈ 37 dB.

6.2.3 Experiments with integrated optical readout

The resulting SNR in the case of a randomly initialized readout is so low that we are unable to correctly discern a 0 from a 1 bit on the RTO, even in the absence of the distorting 25 km fiber. This makes performing the signal equalization task especially hard.

As such, we introduced a preliminary step in the training process to first find a set of weights that minimized the optical loss, as shown in Fig. 6.15(a). This would then place the weights in a favorable starting position for the training algorithm to perform signal equalization with as the target metric the resulting BER.

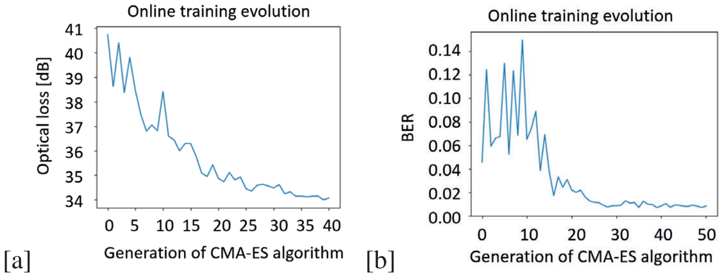


Figure 6.15: Training process. (a) Illustration of minimizing the optical loss. The initial configuration of the weights was such that the transmission of the readout at the final output was heavily suppressed. (b) After finding a maximum transmission configuration, weights are trained to find lowest BER for the signal equalization task.

As opposed to our previous results, where states from individual nodes are recorded and processed offline, we leverage here the integrated readout to realize online training and inference in a fiber telecom system. Since we no longer have easy observability of all the states, the optimization algorithm used here is the CMA-ES algorithm [108, 109]. This is a commonly used black-box evolutionary algorithm that iteratively searches for the optimal solution by generating new candidate solutions from previous best-performing candidate solutions. Results of our training are benchmarked against a no-reservoir case, where the same system is used, but without a reservoir and without the post-reservoir amplifier.

First, we do not include a fiber spool in the system and instead of equalization, we train the reservoir to reconstruct the transmitted signal. The results of this training, reported in Tab. 6.1, show that it is not possible to perfectly reconstruct the transmitted signal, resulting in a BER of about 1%.

Task	BER with RC chip (%)	BER without RC chip (%)
No spool	0.82	0.00
Spool with 15 dBm input	5.31	13.95

Table 6.1: Achieved BER when using a trained reservoir and when not using a reservoir

Ideal reconstruction is not possible due to the previously explained issues with very low SNR. This is also shown in Fig. 6.16, where the beginning of the transmitted signal is shown. The signal is again purposefully preceded by a header, a long trail of zeros, which helps to identify the start of the sequence (note that in our measurements, the input bit stream repeats periodically). In the right figure the variation around the 0-level is large as compared to the ratio between the 1-level and 0-level, indicating low SNR levels.

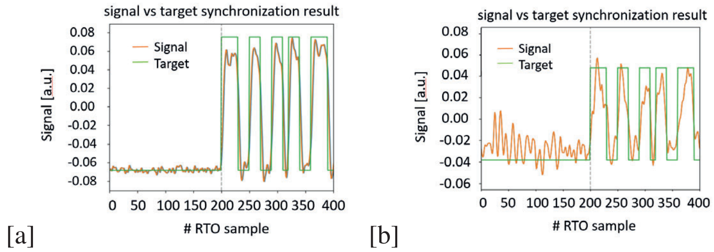


Figure 6.16: (a) Back-to-back signal, i.e. without fiber nor reservoir, showing better SNR levels. (b) Signal after trained reservoir and no fiber spool, showing low SNR levels. The signal axis has a linear scale, the signal and target were scaled and shifted to allow for easy visual comparison.

The signal reconstruction task we just discussed only requires memory, which is corrupted by the heavy losses in the reservoir. This is why we look at dispersion compensation as a second task, which requires nonlinearity as well. Therefore, we include a 25 km spool with a fiber input power of 15 dBm. After training the reservoir, a significant signal equalization is achieved, which as seen in Tab. 6.1, leads to around a 3 times improvement in BER compared to the case without the reservoir. The fact that we obtain any improvement at all, in spite of the severe SNR limitations, is very promising.

6.2.4 Conclusions integrated optical readout

These experimental results indicate that equalization of distorted signals using a photonic reservoir with integrated readout is viable. Promising results were achieved despite severe limitations from:

1. High optical losses and thus low SNR
2. Limited programmability of the weights due to the limited and variable modulation depth from BTO weights
3. The small size of the reservoir and thus the corresponding readout (8 nodes and 8 corresponding weights), which might be too constricted for this task

Chips for future work have already been fabricated with edge couplers instead of vertical grating couplers, which will help with optical loss (Fig. 6.17).

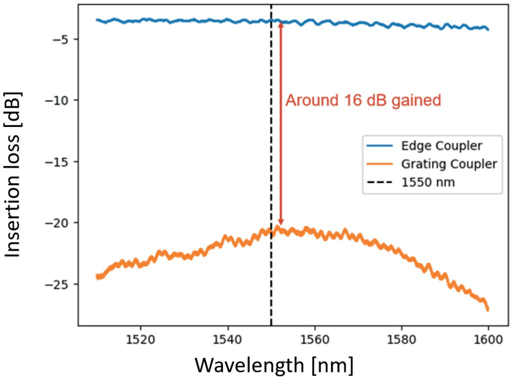


Figure 6.17: Transmission measurements of vertical grating couplers and edge couplers for the Ligentec SiN platform. Around 16 dB of power is gained, 8 dB per coupler, when comparing the edge coupler to the vertical grating coupler case.

The also have tungsten heater weights instead of BTO weights, which will reduce the limited programmability of the weights. Initial passive characterisation of these chips show an improvement of 10 to 20 dB in losses.

We have until now mostly focused on exploring the wavelength dimension by means of introducing WDM in PRC. We will now move on to chapters covering on the one hand how the wavelength dimension can serve other uses for PRC systems and on the other hand how WDM can be useful for integrated photonic computing systems beyond PRC.

7

Integrating nonlinear components in the photonic reservoir for enhanced nonlinearity

This chapter concerns adding nonlinearity to a PRC system by inclusion of nonlinear components in the reservoir. We start by introducing this concept and highlighting how it originally was conceived as a way of enhancing nonlinearity through inter-wavelength interaction. Then we move on to describing the reservoir and readout of the PRC system used in this chapter. The nonlinear component, a III-V SOA, is introduced. We go over the chip manufacturing steps involved in the integration of these nonlinear components, namely transfer printing, photolithography and electrical contacting. Next, we present the SOA characterization. Finally, some of the concerns that would complicate a thorough analysis of this chip are discussed.

7.1 Introduction

Section 2.3 explained how the reservoir should perform a nonlinear transformation of the input. However, the reservoirs shown in chapters 5 and 6 contained no optical nonlinear components. The required nonlinearity was the result of the combination of interference and a photodetector, creating a quadratic nonlinearity

For the work done in chapters 5 and 6, this nonlinearity proved sufficient. However, with an eye on future applications demanding more nonlinearity and in search of further improvements, this chapter concerns itself with adding nonlinear components to the reservoir itself.

Such an extra nonlinearity might be particularly useful when using true WDM signals where there is cross-phase modulation, or when using long-haul optical signals that experienced significant nonlinearity through being re-amplified several times over their journey.

The original idea, stemming from the work done on WDM, was to achieve such nonlinearity through gain saturation and inter-wavelength interaction. However, this chapter will mostly focus on the first steps towards this goal, mainly successfully integrating the nonlinear components in the reservoir. A next step would be to achieve enhanced nonlinearity in the single-wavelength case and only then should future work attempt to enhance the nonlinear dynamics through wavelength interaction.

Before we discuss these nonlinear components, we start with going over the specific reservoir and readout implementations used for this chapter.

7.2 Reservoir and readout implementation

The reservoir layout, schematically represented in Fig. 7.1, follows the four-port architecture model introduced in Fig. 4.2. A notable change compared to previous chapters is that this is a 4x3 reservoir as opposed to a 4x4 reservoir. As a result, the top right and bottom left nodes are missing the outer connections that were meant to avoid radiating away input power due to an unused output or a non-symmetrical node. For these nodes, a single 2x2 DC is used as opposed to 2 DCs forming a 3x3 structure (Fig. 4.1) for the regular nodes. This eliminates this issue.

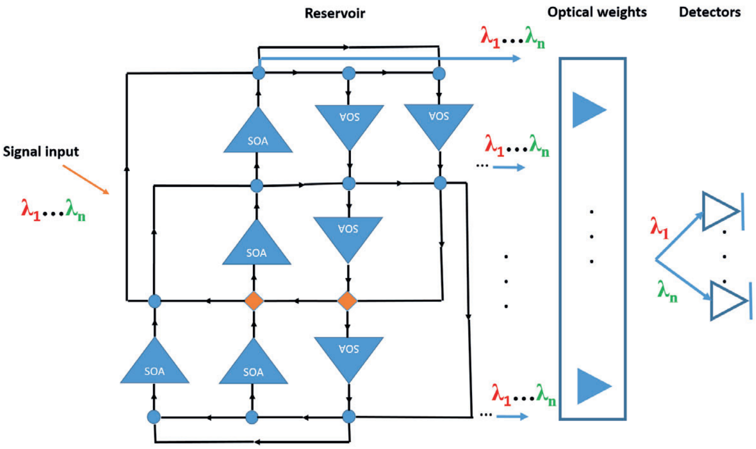


Figure 7.1: Schematic of the SOA reservoir. Signal input is injected in the orange diamond-shaped nodes. All nodes are connected to the readout. Arrows indicate propagation direction of the signal throughout the reservoir.

An integrated optical readout is used for this system, the weights being implemented by MZIs with tungsten heaters on both arms. Our system is integrated on a silicon chip in contrast to the SiN chips used in chapter 6. The chip was fabricated using the ebeam available in the Photonic Research Group (PRG).

The biggest departure from our previously discussed PRC implementations in chapters 5 and 6, is that there are now 8 SOAs included in the reservoir, which are placed in between the reservoir rows. We will now discuss these nonlinear components in more detail.

7.3 III-V semiconductor optical amplifiers

SOAs were developed during the 1980s for their potential applications in lightwave systems [110, 111]. However, with the advent of the doped-fiber and Raman amplifiers, SOAs are rarely used for fiber-loss compensation in lightwave systems. Still, they exhibit a nonlinearity that makes them suitable for several optical signal processing functions and potentially for PRC.

SOAs are pumped electrically by injecting electrons into their active region (Fig. 7.2).

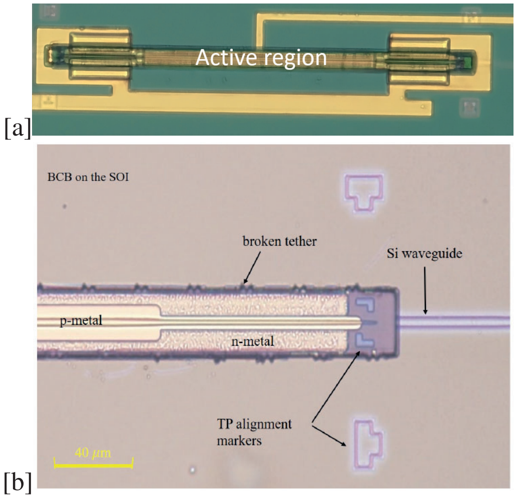


Figure 7.2: a)Picture of an SOA. b) Microscope image of a SOA after micro-transfer-printing on SOI. Figure taken from [112].

SOAs are travelling-wave amplifiers and their designs are similar to those of semiconductor lasers. If a continuous wave (CW) optical signal is injected into the device, this signal is amplified exponentially over a distance L with a single-pass gain $G = \exp(gL)$, as long as the gain remains unsaturated. However, it is relatively easy to saturate the gain coefficient g [113]. This gain saturation, which limits the usefulness of SOAs in lightwave systems as an optical amplifier, renders them potentially useful for enhancing the nonlinearity of our reservoir dynamics (while amplifying the signal at the same time).

As a result of gain saturation and the fact that any change in the carrier density affects not only the optical gain, but also the refractive index within the active region of an SOA, SOAs exhibit a strong carrier-induced third-order nonlinearity. They have effective nonlinear coefficients of $n_2 \sim 10^{-13} m^2/W$, which are seven orders of magnitude larger than those of silica fibers [114]. Although this nonlinearity does not respond on a femtosecond time scale, it is fast enough that it can be used to make devices operating at bit rates up to 40 Gb/s. This is important for our goal of using PRC to target high data bandwidth applications.

A way to understand the nonlinearity of the device, is by considering a pulse travelling through an SOA. The leading edge experiences the full gain as the amplifier is not yet saturated. The trailing edge on the other hand experiences the least gain, since the earlier part of the pulse has saturated the amplifier gain (Fig. 7.3).

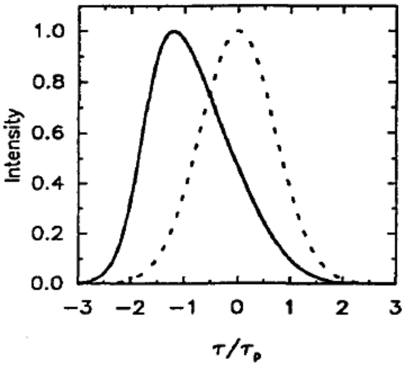


Figure 7.3: Illustration of the effect of SOA gain saturation. Normalized intensity is displayed versus normalized time with τ *signifying time and* τ_p *signifying pulse duration. The full line displays the shape of the SOA output and the dotted line the shape of the SOA input pulse. Figure taken from [114].*

Gain saturation also leads to a time-dependent phase shift across the pulse. Since the pulse modulates its own phase through gain saturation, this phenomenon is referred to as saturation-induced SPM [114, 115]. Also, when a control pulse is used to change the phase and amplification of the input signal, SOAs exhibit XPM and cross-gain modulation (XGM) [116].

Apart from this carrier-induced third-order nonlinearity, there are other nonlinearities such as spectral hole burning, the Kerr effect, two-photon absorption or four-wave mixing [117].

All of these nonlinearities could potentially play a role in enhancing reservoir dynamics, with some allowing for additional nonlinearity through inter-wavelength interaction.

The III-V epitaxial layer structure used to simulate, design and fabricate the SOAs is described in [69]. We now move on to transfer printing, the technique we use to integrate our SOAs on chip.

7.4 Transfer printing

There is a consensus that for silicon photonics monolithically integrated group IV lasers and amplifiers are still far from being practical, and that III-V semiconductor materials and devices, such as SOAs, are needed [69]. But how to integrate such devices on a PIC? Flip-chip integration and die-to-wafer bonding are two possible approaches, each with their own advantages. With die-to-wafer bonding, unstructured III-V dies are bonded epi-side down to a silicon photonic wafer. The unpatterned dies do not need accurate alignment, making this a low-cost and large-scale approach. Flip-chip integration consists of bonding the III-V chip on top of the silicon photonics chip, or into an etched recess. Testing the devices on the source wafer prior to integration is possible but the sequential assembly (using active or passive alignment) can be a bottleneck. Micro-transfer-printing, being developed e.g. in the PRG at Ghent University, allows for the integration of pre-processed III-V SOA on a silicon PIC, combining the advantages of both flip-chip integration and die-to-wafer bonding [69].

The process starts with the definition of the III-V opto-electronic components (in our case the SOAs), on a III-V source wafer, which has the active epitaxial layer stack grown on top of a release layer (e.g. InGaAs or InAlAs, for the InP material system). After patterning of the device and of the release layer, the structures are encapsulated and the release layer is selectively removed using FeCl3:H2O wet etching. This leaves the III-V components attached to the III-V substrate by thin tethers. With a polydimethylsiloxane stamp, one or more thin-film III-V components can be picked up from the source wafer and printed onto a silicon photonic target wafer. Then, the encapsulation is removed and the III-V devices are electrically contacted at wafer level, which can be done using photolithography. This approach enables pre-testing of the III-V devices on the source wafer, similar to flip-chip integration, but also massively parallel integration, similar to the die-to-wafer bonding approach [69]. The result in our case is shown in Fig. 7.4.

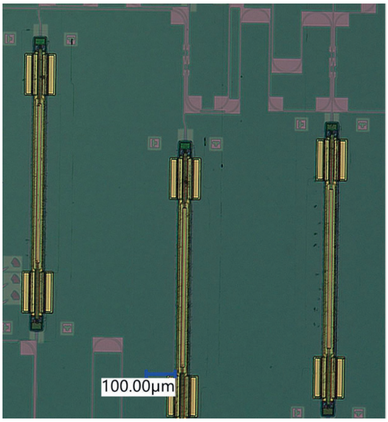


Figure 7.4: Microscope image of the micro-transfer-printed SOAs on the PIC before electrical contacting using photolithography.

For the specifics of the micro-transfer-printing approach used for the integration of the SOAs to the reservoir discussed in this chapter, we refer to [69] and we would like to thank Dr. Jing Zhang for his help in this matter. In order to avoid having to process III-V semiconductor devices on the SOI target wafer, all SOA processing steps were carried out on the III-V source, which, as discussed above, is a main advantage of transfer printing. The post-processing steps are limited to the removal of the encapsulation, via dry etching, planarization and electrical contacting, via photolithography. The author of this work was mainly involved in these last photolithography and contacting steps, and hence these will now be looked at in more detail.

7.5 Photolithography and electrical contacting

Photolithography is an optical means of transferring a pattern, in this case the desired surface on which to deposit gold for electrical contacting, on a substrate, in this case a PIC containing a photonic reservoir.

First, TI35E photoresist is placed on the entire surface of the PIC after applying the adhesion promotor TI prime. Application of resist and promotor are done using a spincoater at 3000 RPM for 40”. Both the application of the adhesion promotor and the photoresist are followed by a baking step, 3’ at 120° and 3’ at 100° respectively.

Second, the substrate is exposed to light for 55” with 325W of power, through a contact mask containing the desired pattern for the metal interconnections (Fig. 7.5).

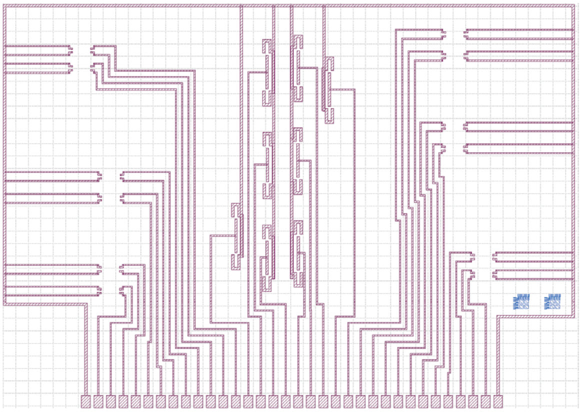


Figure 7.5: The contact mask used for the photonic reservoir with SOAs and integrated readout based on tungsten heater equipped MZIs

This modifies the molecular structure of the resist and changes its solubility. Depending on the type of resist, the exposed resist will either remain or be removed after developing. In our case, the non-exposed resist was removed. Such resist is called negative resist and we also use the term image reversal in relation to this. The exposure step is followed by a 15' wait, which allows nitrogen generated during the exposure to diffuse out of the resists [118]. Next is a baking step at 125° for 3', this is referred to as the reversal bake as it changes our resists behavior from positive to negative. Finally the exposure part of the image reversal process is ended by a flood exposure step of 185".

Third, after exposure, developing is carried out during 120" using a AZ400K:H2O 1:3 solution. After developing, the chip looks like Fig. 7.6).

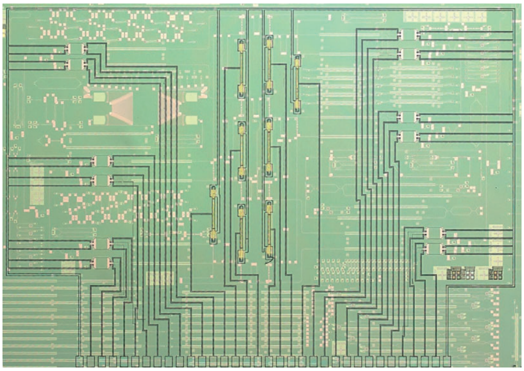


Figure 7.6: The PIC after developing.

Fourth, a titanium adhesion layer followed by the gold electrical conductor layer is deposited. These will only make contact with the PIC where the resist has been removed. This metal deposition step was carried out in our cleanroom by Steven Verstuyft. Then, the chip is immersed in a developer solution, which is typically aqueous and dissolves away the remaining photoresist, together with the metal layers deposited on top of it. In this case, an acetone bath with vibrations, in conjunction with acetone rinsing was used. In the end, the metal only remains on those places specified by the mask pattern (Fig. 7.8). Such a metallization scheme is also referred to as a lift-off scheme, since the undesired parts of the metal are washed away (Fig. 7.7).

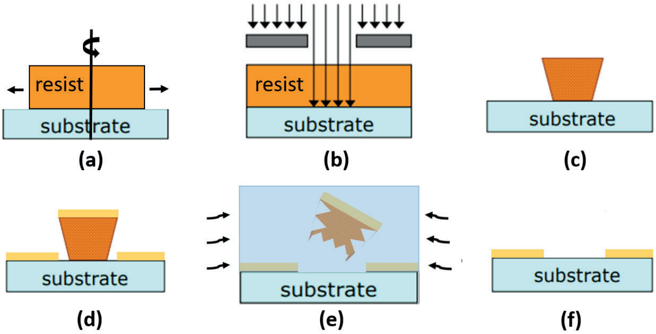


Figure 7.7: Illustration of the lift-off process [118]. (a) Using spincoating, resist is applied to the substrate. (b) The substrate is exposed to light through a contact mask. (c) The negative resist only remains where it was exposed to light after developing. (d) The metal is deposited. (e) An aceton bath with vibrations dissolves the photoresist and the metal on top of it is removed. (f) In the end the metal only remains there were the light was blocked by the mask during the exposure step.

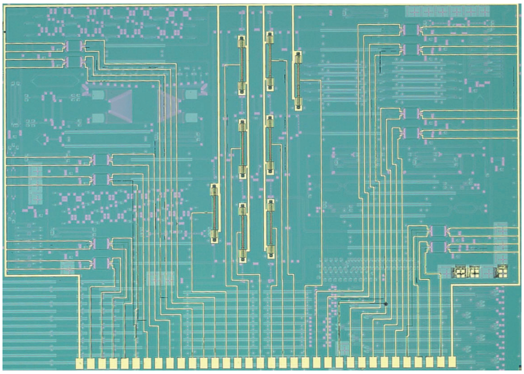


Figure 7.8: The PIC after metal deposition.

Having now arrived at integrated and electrically connected SOAs in the reservoir, we move on to the characterization of these devices.

7.6 SOA characterization

After all micro-transfer printing post processing steps, we tried to verify the performance of the SOAs. The most pressing concern was verifying whether no little pieces of gold remained stuck to the PIC after lift-off. This is a common concern for lift-off using an acetone bath with sonic vibrations in conjunction with rinsing, as it is a somewhat tricky manual process. This was especially true in this case, because the chip was fairly large and the surface over which photoresist remained was much larger than that where it was removed, leading to a lot of gold needing to be washed away. A small piece of gold remaining stuck on the chip is usually not a problem, unless it causes a short circuit in an SOA as illustrated in Fig. 7.9. It just so happens that the transfer-printed SOAs stand up from the surface of the chip and are thus the spots were loose gold pieces tend to get stuck on.

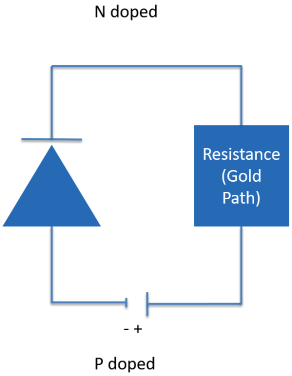


Figure 7.9: A piece of gold stuck between an SOA's n and p doped regions acts as a parallel resistance.

Such gold pieces cannot always be spotted visually and the most reliable way to identify this problem is to measure the IV-curve of each SOA. A short circuit, likely by an errant piece of gold, was indeed observed in 1 of the 8 SOAs (Fig. 7.10).

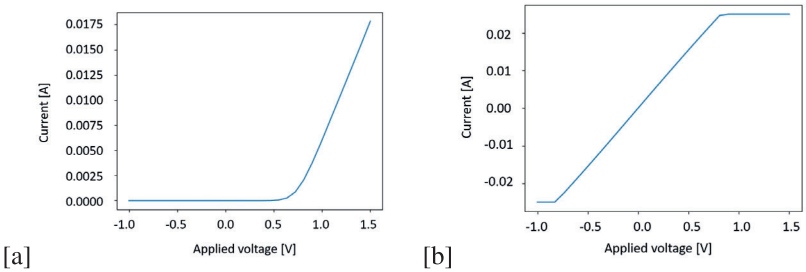


Figure 7.10: (a) IV curve of well functioning SOA. (b) IV curve of SOA with short circuit defect. The saturation of the current that can be observed was due to a current limit being applied to the power source.

The other SOAs showed an expected IV curve. However, this alone does not allow to draw us any conclusions about the optical amplification yet.

7.7 Optical loss, power dissipation and current limit concerns

As was seen in chapter 6, the optical losses inside the reservoir are significant, leading to the need of additional optical amplification after the chip in order to amplify the signal to within the detection range of the photodetector. However, an unpowered SOA is a source of significant loss (about 50 dB) on top of already existing losses inside the reservoir, as shown in Fig. 7.11.

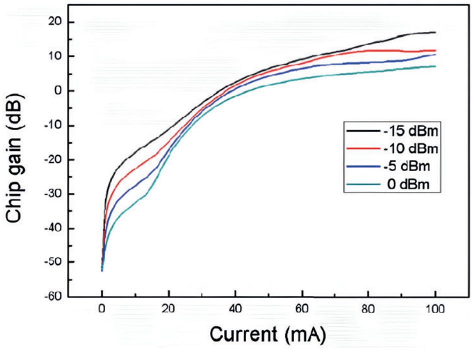


Figure 7.11: Chip gain vs. current at different input optical power (1550 nm wavelength) taken from [119].

It is thus necessary to supply at least enough power to the SOAs in order for them to reach optical transparency, as otherwise the signal in the reservoir would be too weak. However, the power and current requirements imposed in this way are significant and gave rise to a number of concerns.

Looking at Fig. 7.11, which should be representative of what we can expect from our SOAs, we can see that we can expect to need at least 40 mA per SOA just to have transparency. For amplification and nonlinearity, even more current will be required.

The first concern is that 8 SOAs on chip give rise to non-negligible power dissipation. This causes heating of the SOAs, which leads to increased resistance and therefore decreased voltage over the diode part of the SOAs. This finally results in decreased optical gain, so that it can be a problem to get all eight SOAs to operate at transparency. It is difficult to assess the seriousness of this issue due to the challenge of modelling this in simulations or accurately tracking chip temperature during experiments. To this end, we attempted to make use of a thermal camera. However, this method proved unreliable. This concern did not impede us to move ahead with measurements, but should be kept in mind for future work.

The second, more serious, concern is that the metal lines and wirebonds carrying the current from the two common ground bond pads only have a limited current capacity and still need to serve up to 24 tungsten heaters and 8 SOAs. We anticipated for this by making the ground lines 40 micron wide, compared to the 20 micron for the signal lines. A similar issue arises with the wirebonds. These are needed because, in order to experimentally test so many integrated weights and SOAs, it is necessary to make use of a printed circuit board (PCB) for electrical interfacing. These wire bonds between the chip and the PCB can also be a point of failure, breaking down in case of too high current [120]. To estimate this, we start from the observations in Fig. 7.12.

WIRE DIAMETER	GOLD			COPPER		
(Mils)	1mm long	2mm long	3mm long	1mm long	2mm long	3mm long
2	4.075	2.11	1.425	5.45	2.8	1.89
1.3	1.8	0.97	0.665	2.41	1.275	0.87
1.2	1.563	0.847	0.58	2.065	1.1	0.76
1	1.12	0.625	0.435	1.475	0.81	0.56
0.9	0.94	0.525	0.368	1.225	0.677	0.47
0.8	0.76	0.435	0.309	1	0.55	0.394
0.7	0.612	0.355	0.254	0.787	0.45	0.322
0.6	0.475	0.28	0.203	0.608	0.355	0.255

Figure 7.12: Summary of current values in Amperes for different wire combinations taken from [120]. The number relevant for this work is indicated by the red box.

Our wire bonds are made of gold and are 18 micron (0.7 mils) thick. We estimate them to be a bit longer than 3 mm, although the precise length is difficult to measure due to the curved path of the wire bonds. Since Fig. 7.12 tells us that such a wire can only carry 254 mA, it seemed prudent to stay well below $2 \times 254 = 508$ mA of total current through the two ground pads.

The integrated readout output was measured while only driving one or several of the SOAs with limited total current, staying under our calculated 508 mA by at least a factor 2 to be on the safe side, while sending an optical input signal into the reservoir. The readout was left unsteered, that is to say the weights were in their default random initialization, again out of total current considerations. The goal was simply to see if we could drive the SOAs sufficiently in order to get an optical output signal of sufficient strength. Unfortunately, the highest measured optical output signal was still $<$-70dB, several tens of dB too low for viable measurements with a modulated input signal and a setup as in chapter 6, which indicates that we are probably not able to send enough current to the amplifiers. In addition, the output signal seemed to be mostly due to SOA spontaneous emission. Indeed, varying the power of the input signal had non-measurable impact on the measured output signal, while varying current to the SOAs did clearly impact the measured output signal. Experiments to estimate nonlinearity were thus also out of reach.

The result was such that it was clear that several improvements would be needed before being able to proceed with measurements. Due to time constraints, we had to postpone further measurements. When continuing this work at later stage, we would first address our concern on the ground wirebond current capacity. We have two ideas for this:

1. Land additional wirebonds on the metal lines leading up to the ground bondpads or the ground bondpads themselves. This is challenging in terms of wirebonding, however.
2. Each MZI weight now has a tungsten heater on each arm. In theory, such a weight could function with only one-arm steering. We could thus electrically connect redundant signal bondpads of one or more weights with the ground bondpads in order to create more ground bondpads.

Once this concern were to be removed, we would run less risk of irreparably damaging this labor intensive chip.

For the next chapter, we will focus on using WDM to speed up matrix multiplications, an important primitive in machine learning.

8

Parallel convolutional processing using an integrated photonic tensor core

This chapter concerns a different photonic computing platform than PRC, but shares the focus on exploiting WDM. We start by introducing the context in which this work was undertaken, before moving on to a description of this hardware platform, a photonic tensor core, i.e. an accelerator for matrix-vector multiplications. We outline our contributions to the platform, which focus on the WDM aspect of the photonic tensor core. More specifically, we designed a thermally tuned wavelength AWG demultiplexer. We present the experimental results for both types of first-generation thermally tuned demultiplexers, single- and multiple-heater designs and end the chapter by introducing a second-generation design informed by these experimental results.

8.1 Introduction

This work was undertaken in the context of the European H2020 project Phoenics, which aims to develop an integrated photonic tensor core, i.e. an accelerator for matrix-vector multiplications. Although this is not a PRC system, the project shares many of the high-level goals with the main thesis topic. In particular, it exploits the frequency/wavelength space as an additional scaling dimension, parallelizing operations through WDM.

This approach to photonic computing differs from PRC in that it is only focused on a single specific functionality, namely multiply-accumulate (MAC) operations, for which it provides a dedicated hardware implementation. These operations are ubiquitous in ML. More specifically, they are a bottleneck of convolutional NNs, which are a type of NN highly effective for applications such as image classification, autonomous navigation and audio analysis in the frequency domain [50, 121, 122].

Let us now proceed to describe the photonic tensor core system.

8.2 Photonic tensor core

A photonic tensor core is a computationally specific integrated photonic hardware accelerator that is capable of performing MAC computations, the fundamental mathematical elements needed for matrix-vector multiplication (MVM) operations, at speeds of trillions of operations per second (tera-MACs per second). The tensor core can be considered as the optical analog of an application-specific integrated circuit (ASIC), with the application being MVM operations. The implementation considered in the Phoenics project, and our explanation thereof, is based on [6].

An MVM is achieved by configuration of phase change material (PCM) patches on a waveguide matrix. The PCM material used here is a GST ($Ge_2Sb_2Te_5$) film of ≈ 10 nm thick. In the crystalline PCM state, most of the incoming light is absorbed, while in the amorphous state, most of the light is transmitted. ”Intermediate transmission states can be chosen by controllably switching fractions of the PCM cell” [123, 124]. To achieve both positive and negative matrix elements, ‘0’ can be defined as an intermediate state between the crystalline and amorphous states [6]. The input vector is encoded in the amplitude of the optical signals sent to the different matrix inputs.

These optical signals are situated at different wavelengths so that they can be incoherently added, while avoiding unwanted interference. This is illustrated in Fig. 8.1.

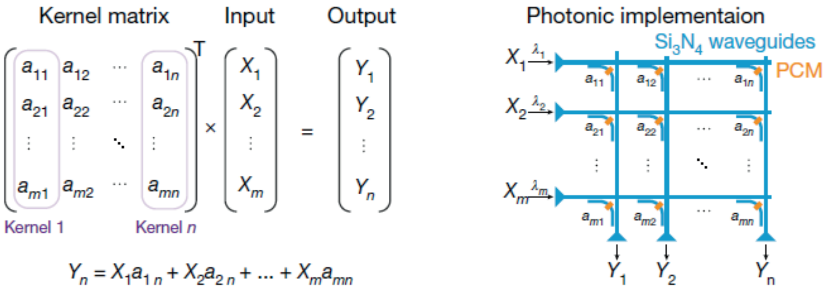


Figure 8.1: Basic MVM: an input vector $[X_1 ... X_m]$ is encoded in the amplitude of different wavelengths (λ_1 to λ_m) and sent to the corresponding matrix input waveguides. The matrix elements are inscribed in the state of PCM patches on the waveguides. The splitting ratios of the directional couplers are chosen such that the same fraction of the light for each input reaches the output [6].

In addition, this optical implementation of a matrix-vector multiplier allows the harnessing of WDM to execute parallel MVM operations. As Fig. 8.2 shows, the same matrix can be used to process several input vectors at the same time when all the individual vectors are encoded in different wavelength channels. A wavelength channel in this context refers to a collection of wavelengths, one for each of the elements in an input vector.

Figure 8.2: WDM in the matrix-vector multiplier. Each colored band represents a wavelength channel that contains multiple wavelengths on which to encode the inputs of a specific input vector. Figure adjusted from [6].

The wavelengths are derived from a photonic chip-based optical frequency comb (soliton microcombs [125]) driven by a continuous-wave laser, as shown in Fig. 8.3.

In our system, the lines will be separated by 20 GHz.

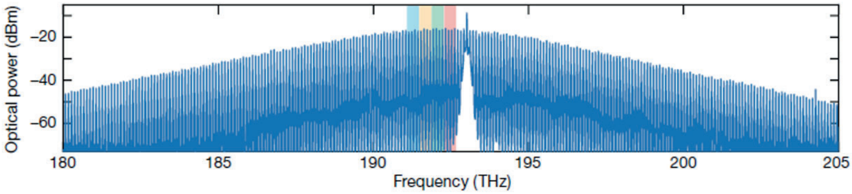


Figure 8.3: Measured spectrum of a single-soliton frequency comb. The bands of color indicate WDM wavelength channels. Each channel contains a number of wavelengths which amplitudes can be used to encode different input vectors [6].

Let us now describe how all of this fits into a larger system, as illustrated in Fig. 8.4. This figure shows 4 parallel MVMs with matrix size 4×4, whereas we will be targeting 16 parallel MVMs with matrix size 25×25.

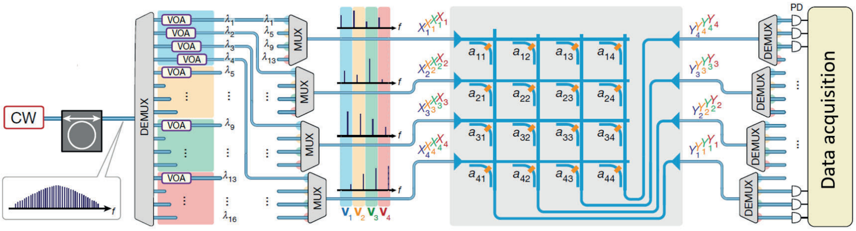


Figure 8.4: Schematic illustration of photonic tensor core system. The input vectors are generated from lines of a photonic chip-based optical frequency comb driven by a continuous-wave laser using wavelength division demultiplexers (DEMUXs) and variable optical attenuators (VOA's). The entries of different input vectors are grouped together employing wavelength multiplexing and sent to the on-chip multiplication unit that performs the calculations. After combining the correct wavelengths with optical wavelength division demultiplexers (DEMUXs), the multiplication results are accumulated on the PDs [6].

The encoding of the input vectors is done by means of demultiplexers (DEMUXs) followed by variable optical attenuators (VOA's). From each of the 16 wavelength channels (colored bands in the figure), the correct 25 entries of each parallel input vector are grouped together using 25 wavelength division multiplexers (MUXs), each one having 16 inputs, one for each of the parallel channels. These (de)multiplexers were implemented in polymer, and were designed by other partners in the project.

Next, the result is sent to the on-chip multiplication unit that performs multiplications with a matrix of size 25×25. Each of the 25 outputs is sent to a final DEMUX, which separates the 16 different parallel WDM channels. These are the components that we designed. Finally, these multiplication results are accumulated using 16×25 photodetectors.

Now that we have explained how the system works, we can translate this into requirements for our devices. We need 25 AWGs, each with 16 output wavelength channels. Each channel should fit 25 wavelengths, with each of these wavelengths separated by 20 GHz. This means that the channel spacing should be at least $25 \times 20 = 500$ GHz. In addition, the response for each wavelength within each channel should be as equal as possible. So, our components should have a flat-top response over the entire channel width and a steep roll-off. These are all very challenging requirements. More precisely, the project envisions an ideal situation with a channel guard band of only 20 GHz for a 500 GHz channel spacing, with a 10 THz FSR which translates to a roll-off of 10-20dB in a band that is just 1% of the FSR. In this work, we will attempt to approximate this to the best of our ability. Over time the project explored compromises such as downsizing the multiplication matrix to a 5x5 matrix, which then corresponds to the 5 inner frequencies per channel being of interest, which greatly relaxes the extremely ambitious roll-off requirement.

Finally, since we need a high number of them, the footprint of the devices should be as small as possible. The tight bends and high refractive index contrast of silicon waveguides allow the realization of a compact AWG. However, these high-contrast waveguides are highly sensitive to phase errors for the same reason. These phase errors can arise from fabrication errors, a notable one being sidewall roughness as explained in sections 3.4 and 4.5 and illustrated in Fig. 3.2. Therefore, for silicon AWG crosstalk is a particular concern as well. This needs to be kept in mind, as our channel spacing is small with respect to our channel width. There is also the need for near perfect spectral overlap with the other AWG multiplexers in the system, not designed by the author. Considering these issues, wavelength channel tuning capability also became a requirement.

Having introduced the photonic tensor core, we now focus on the part to which we contributed, namely the WDM demultiplexer components.

8.3 AWG WDM demultiplexers

Our role in the project was to design the silicon AWG demultiplexers after the multiplication matrix. In the hardware implementation of PRC systems discussed in Chapter 6, such demultiplexers were not discussed, because there we read out wavelengths sequentially as opposed to in parallel. In a final system however, wavelength demultiplexers after the readout would be preferred, as sequential readout of the wavelengths would limit data bandwidth. The work done here for the Phoenics project is thus still relevant for our own future fully realized WDM capable PRC systems.

8.3.1 Design of AWG demultiplexer

We base our explanation of an AWG on the description in [126]. An AWG consists of two star-couplers and an array of waveguides with linearly increasing length, which introduce a linear progression of phase. A star-coupler acts as a splitter/combiner (Fig. 8.5).

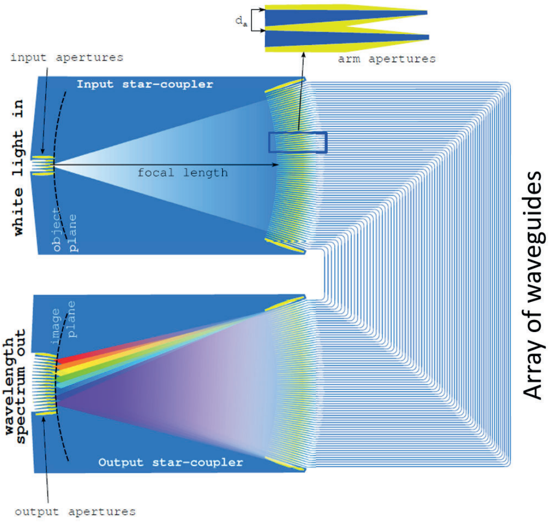


Figure 8.5: Basic operation of an AWG. Figure taken from [126] with adjustments.

A light beam exiting from a waveguide aperture enters into the input star-coupler and diffracts. This expending light beam is then distributed over the array of waveguides and propagates through these waveguides to the output star-coupler. The optical path length difference between two successive waveguides in the array is constant. Therefore, for the center wavelength, the field distribution at the entrance of the waveguide array will be reproduced at its exit aperture. In an ideal system, a perfect image of the object plane can then be formed and the field distribution in the image plane will have the same amplitude and phase as the field distribution of the object plane [127]. Obviously, the imaging is never perfect, because one uses a finite number of waveguides in the array and the array has a finite numerical aperture. The phase difference in the waveguide array varies with wavelength, because of the length difference of the waveguides and because of the wavelength dependence of the propagation constant. This phase delay will induce a tilt in the phase fronts in the output star-coupler and thus introduce a lateral shift of the image field. The spectral response of the AWG will therefore be determined by the overlap of this shifted field profile with the mode profile of the different output waveguide apertures located in the image plane. In conventional AWGs, the geometry of the aperture at the input (object plane) and output (image plane) is identical: this results in a maximal overlap when the shifted image is perfectly aligned with one of the output apertures (i.e. at a given wavelength). However, the coupled output power rapidly drops with a small image shift (i.e. a small wavelength shift) [126].

For the design, we target the IMEC iSiPP200 silicon-on-insulator platform. As for any fabrication platform, there are a number of design rules to obey, which are specified in the iSiPP200 process development kit (PDK).

The design and layout of the AWG itself was performed using Luceda's AWG designer [128]. During the design, we had to make a number of design choices, which we will discuss next.

Free spectral range

A lower bound for the free spectral range (FSR) in our application is the product of the required channel spacing and the number of wavelength channels, i.e. 16x500 GHz. Increasing the FSR to be larger than this minimum by adding guard bands is of course good practice. Therefore, the targeted FSR was varied across the first fabricated designs between 16x500,18x500 and 20x500 GHz, i.e. 0,2 and 4 dummy channels.

Waveguide width

A wide waveguide for the waveguide array is preferred, as this reduces the phase errors significantly. This is because the optical mode is confined very well in the waveguide and is not impacted as much by sidewall roughness. At the same time, it is important that the waveguide remains single-mode to avoid phase errors related to excitation of higher order modes in the waveguide. Increasing the width of the waveguides can also lead to decreased spacing between the waveguides, and therefore increased waveguide coupling and crosstalk, depending on the chosen waveguide array layout. This is discussed in more detail in the subsequent section.

Rectangular versus S-shaped AWGs

In rectangular AWGs, the straight waveguide sections are laid out along the x and y direction connected by bends of equal radius (Fig. 8.6(a)). In the high-contrast silicon platform, the best performance is achieved by these types of AWGs, because the rectangular shape makes them more fabrication tolerant compared to other shapes [126]. In our design, we needed relatively short waveguides to get a high FSR (16 channels with 500 GHz spacing imposes a FSR of at least 16x500 GHz). However, due to the rectangular shape, shorter waveguides will result in a smaller gap between two adjacent waveguides, which will increase the waveguide coupling and therefore the crosstalk (Fig. 8.6(b)).

A potential solution to this problem is employing the S-shaped AWG as an alternative layout (Fig. 8.6(c)). This design avoids the problem of decreasing waveguide spacing, as illustrated in Fig. 8.7, but has in general a worse performance than the rectangular AWG. The main reason for this is that the increased waveguide lengths which lead to a larger accumulation of phase errors due to fabrication errors, i.e. sidewall roughness. As mentioned, this a particular concern for the high-contrast silicon platform.

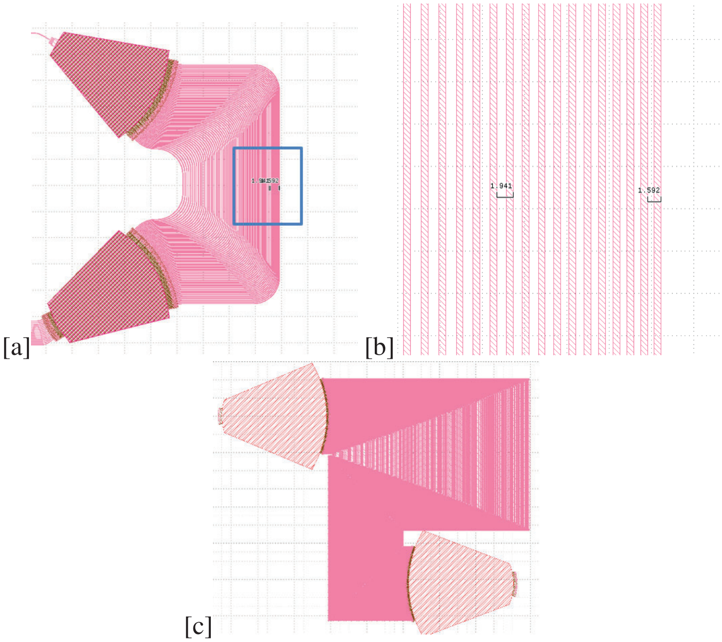


Figure 8.6: (a) Example of rectangular AWG. (b) Illustration of decreasing waveguide spacing for a zoomed section of the rectangular AWG. The spacing decreases from 1.941 μm to 1.592 μm after ten waveguides. (c) Example of S-shaped AWG

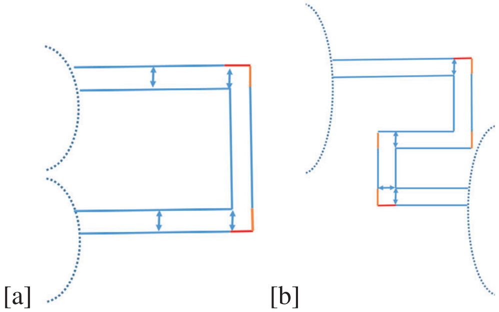


Figure 8.7: (a) It is not possible to maintain equal spacing between delay lines in the rectangular shape AWG without the lengthening the outer delay line by an amount that potentially exceeds the required fixed length difference. (In this illustration, the star couplers are non-rotated. Rotation of the star couplers can help match the extra length needed for equal spacing to the required fixed length difference for correct AWG operation.) (b) It is possible to maintain equal spacing between delay lines in a S-shape AWG without needing to lengthen the outer delay line. Therefore, lengthening the outer delay line by the required fixed length difference will not shorten the delay line spacing. Note in both cases the effect of the slight curvature of the star couplers was neglected for purpose of this illustration.

Eventually, it was decided to proceed with a rectangular AWG. The issue of decreasing waveguide spacing was alleviated somewhat by optimizing the angles of rotation of the star couplers, which slightly shifts the waveguide delay line layout.

With these adjustments, we are at the very limit of what is possible, with regards to the high FSR, with a rectangular AWG.

MMI input aperture

AWG wavelength channels are, because of the physics underlying their operation, not flat-top. The spectral response at a conventional AWG output is instead Gaussian-like. To alleviate this problem, we can make use of an MMI coupler as an input aperture, as was proposed by [129, 130] (Fig. 8.8).

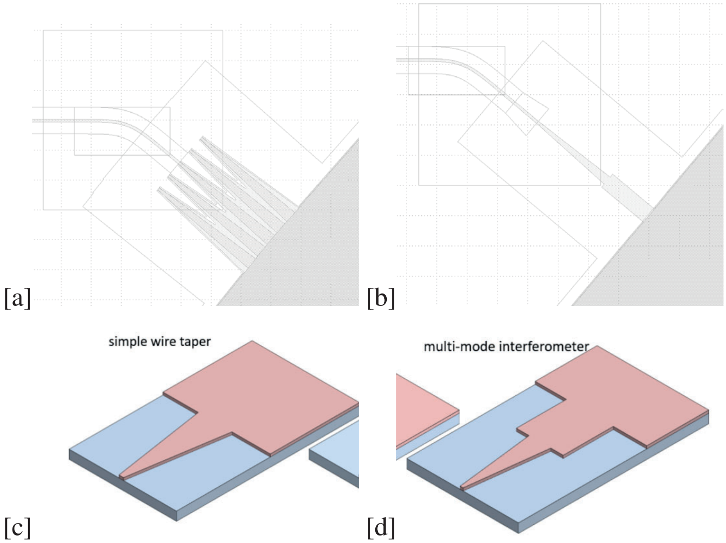


Figure 8.8: (a-c) Regular input aperture. (b-d) MMI coupler input aperture. Figures (c-d) taken from [72].

In this case, the input and output apertures are different, and are designed such that the output image of a channel will be much larger than the output aperture (Fig. 8.9).

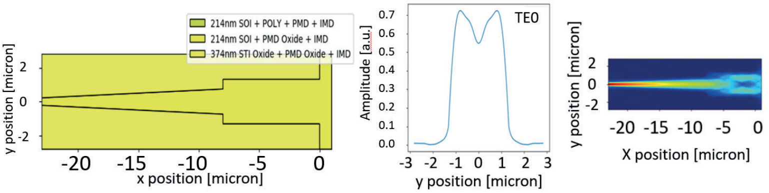


Figure 8.9: At the object plane of the AWG, the MMI aperture will generate a wider two-lobed field profile with its precise shape depending on the length, width and taper width of the MMI.

Therefore, when moving in a sufficiently small wavelength range around a channel's central wavelength, this extended image shifts but still maintains a complete overlap with the output aperture, such that there is no drop in transmitted power. This helps to improve the flat-top response and increase the channel width.

The drawback is that the output aperture no longer captures the entire image, leading to a loss in maximum transmission, i.e. an increase in insertion loss.

The final design of the rectangular AWG with MMI input aperture is shown in Fig. 8.10.

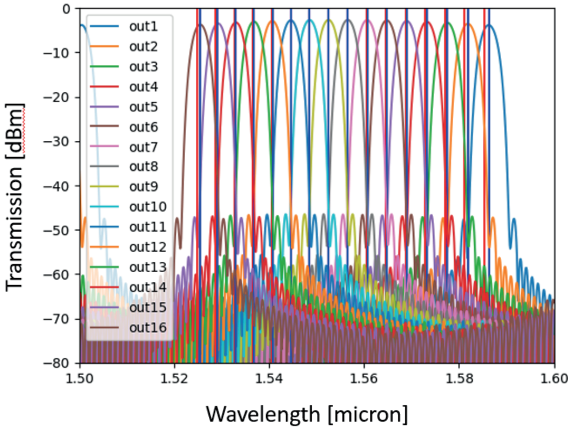


Figure 8.10: Simulated AWG response. Blue lines indicate the simulated peak location, red lines indicate the targeted peak location.

8.3.2 Thermal tuning

As stated in section 8.2, due to fabrication errors that can happen both for the multiplexers before the multiplication matrix, which are not designed by the author, and the demultiplexers after the multiplication matrix, which are designed by the author, it would be useful to have some post-fabrication control over the effective index of the demultiplexer waveguides, so as to be able to shift the channels. One obvious way of changing the effective index felt by a mode is by changing the temperature, which results in a shift of the refractive index of the material. This can be done by fabricating a heater using tungsten lines on top of the waveguide.

Ideally each waveguide would come equipped with its own tungsten heater line and suffer no thermal crosstalk from the other tungsten heater lines. This would allow for correction of individual waveguide fabrication errors. However, for the densely packed waveguides in our design this is not feasible. Instead, we focus on correcting for the average fabrication error, as fabrication errors across the delay lines consist in part of shared deterministic, mask discretization error [126], and slow-varying, with respect to AWG scale, systematic contributions [82].

There were two main metrics kept in mind during the design:

1. Maximum possible wavelength shift
2. Stability of demultiplexer performance under heating

Let us now discuss each of these in more detail.

First, the maximum possible wavelength shift is determined by maximum possible power dissipation. A power dissipation that is too small will fail to sufficiently impact the effective index and bring about a sufficient channel shift.

The power dissipation needed to shift the AWG spectra with one FSR can be estimated using the efficiency of our tungsten heater phase shifters, i.e. the power dissipation $P_{2\pi}$ needed for a 2π phase shift which is $\approx$ 50mW [131]. To shift over one FSR we need to increase the constant phase difference between our AWG delay lines by 2π. This means that our first (the shortest) delay line needs no additional phase delay, our second delay line needs an additional phase delay of 2π and our last (the longest) delay line needs an additional phase delay of $(n_{lines} - 1) \cdot 2\pi$ with n_{lines} being the number of delay lines in our AWG design. This translates to a total power dissipation P_{diss}^{FSR} for one FSR shift given by:

$$P_{diss}^{FSR} = \frac{n_{lines}}{2} P_{2\pi}(n_{lines} - 1) \qquad (8.1)$$

$n_{lines} = 59$ in our AWG design. In this way we obtain an estimated 71.1 W power dissipation for one FSR shift or $\approx$ 1.13 nm per Watt of power dissipation given a FSR of 10THz which translates to $\approx$ 80nm. This is an under estimation of the wavelength shift per Watt of power dissipation as it does not take into account the heat spillover between delay lines.

Power dissipation is mainly limited by the electrical current limits of the tungsten heaters themselves, by the metal lines carrying the current from the bondpads to the heaters and in particular by the interfaces between them. In addition, in order to experimentally test a high number of individual heater lines, it can be necessary to make use of a PCB for electrical interfacing. The wirebonds between the chip and the PCB can then also be a point of failure, breaking down in case of a current that is too high [120]. A good design can help alleviate some of these issues. E.g., metal lines can be designed to be as broad as possible, increasing their current capacity and the surface area of the metal line-tungsten heater interface can be increased, in order to decrease the risk of breakdown.

Second, the stability of demultiplexer performance depends on the heating uniformity. If the heating is not sufficiently uniform, it will introduce its own phase errors, as the refractive indices of different delay lines are not equally affected. This leads to increased crosstalk and eventually to a breakdown of the demultiplexer function and to the disappearance of the wavelength channels. Uniform heating mainly depends on uniform covering of the waveguides by the heating elements.

An ideal heater would thus be a single tungsten heater block covering the entire waveguide array, with enough extra 'spillover' to not have a temperature difference between the edges of the waveguide array and its center, equipped with monolithic tungsten interfaces at the metal lines. Such a design however is not possible, as this does not conform to allowed tungsten dimensions and density iSiPP200 PDK design rules. Therefore, this ideal has to be approximated by a collection of separate heater lines and grid-like tungsten interfaces as shown in Fig. 8.11.

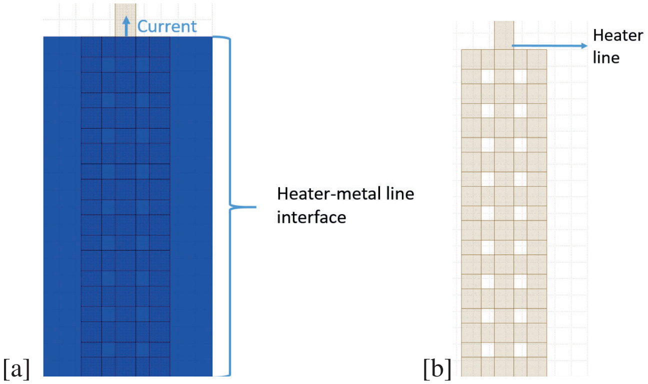


Figure 8.11: (a) Tungsten-metal interface (blue indicates metal line, brown indicates tungsten heater.) (b) Same as (a) but only showing tungsten. The tungsten heater line is split into a grid-like structure in order to maximise the tungsten-metal interface surface area and thus minimize the risk of breakdown due to high electrical current density.

Simultaneously having both sufficient power dissipation and uniform heating poses a significant challenge, because of the large number of delay lines needed to achieve our high number (16) of channels.

Two approaches were explored for the first fabrication run (Fig. 8.12):

1. a single-heater design, where a large tungsten heater is implemented as a collection of meandering heater lines through which the same current flows.
2. a multiple-heater design, where a number of straight heater lines covers several adjacent waveguides. Through each line, a different electrical current can flow.

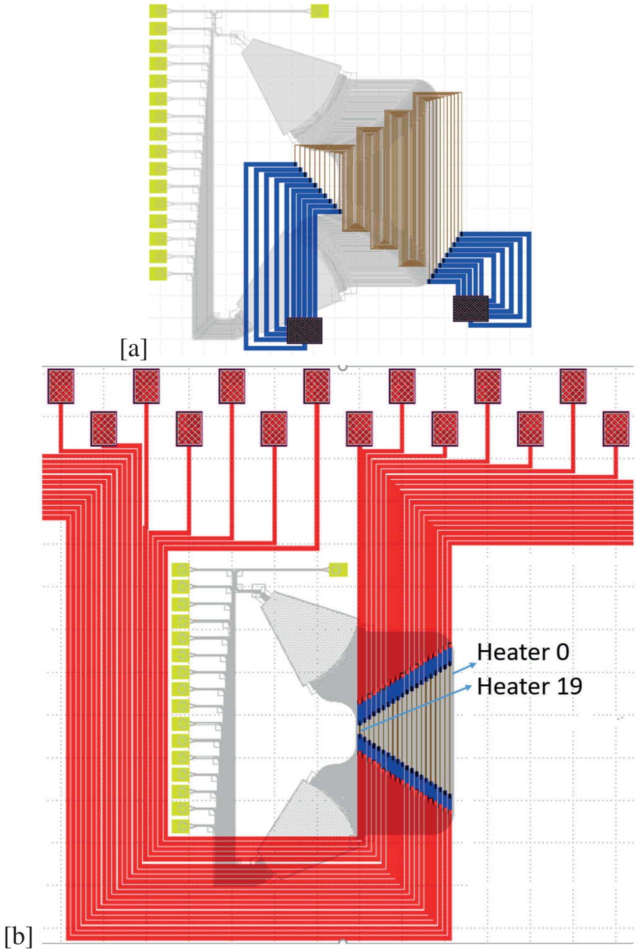


Figure 8.12: Blue and red colors indicate metal lines at different metal routing layers. Brown color indicates tungsten heater. (a) Single-heater design. The meandering tungsten heater lines share the same ground and signal bondpad and thus act as one big heater element. (b) Multiple-heater design. Every straight heater line has its own bondpad (not all of which are shown) and can thus be steered independently from the other heater lines. Heaters are numbered from longest to shortest heater line.

The single-heater design uses meandering heater lines, which leads to a much denser coverage of the delay lines by the heater lines. This coverage is only constrained by maximum tungsten density design rules. In this approach, we only need a single signal bondpad which leads to simple electrical routing and interfacing. For the single-heater design, it was of importance that the meandering heater lines were of equal length. Indeed, equal length translates into equal electrical resistance, which in turn translates to equal electrical current, power dissipation and more uniform heating.

The multiple-heater design allows for a more granular control over the heating and can thus help achieve uniform heating. It could even compensate for variation in fabrication error across the delay lines. However, the number of heater lines is limited by considerations such as electrical routing and electrical interfacing complexity. This limits in turn our control over the heating. One can immediately spot the drawbacks of the multiple-heater design: there is much less heater line coverage over the delay lines and thus lower maximum power dissipation. Of course, the multiple-heater design also has much more complicated electrical interfacing. In addition, the large number of bondpads and metal routing required, greatly increases the chip footprint. This is not ideal, considering that the final project design would require a large number of AWGs. One could remark that the number of required bondpads could be reduced from two per heater to only one per heater plus an additional common ground bondpad, simplifying electrical interfacing and reducing footprint. However, it was a conscious decision not to do this, in order to limit the current density in the wirebonds to the PCB.

An important remark for our thermal tuning is that our silicon AWGs have a positive temperature coefficient, meaning that when increasing the temperature, the wavelength channels will shift toward higher wavelengths. This means that one can only correct for wavelength deviations in one direction, unless one has a tuning range nearly equal to the system FSR. This however is, as we will shown from the results, an improbably large tuning range, more than an order of magnitude beyond what was achieved here. One the other hand, the polymer AWGs (not designed by us) that are used as multiplexers before the multiplication matrix, have a negative temperature coefficient, i.e. they shift in the other direction when increasing temperature. This means that the tuning of both polymer and silicon demultiplexers can be combined in order to fix rather large wavelength channel position discrepancies. However, it is possible that one loses a wavelength channel. This happens if the first wavelength channel of the polymer multiplexer is positioned at smaller wavelength than the first channel of the silicon demultiplexer. In this case, the tuning has to be carried out such that the second channel of the polymer multiplexer can overlap with the first channel of the silicon demultiplexer.

Having outlined the designs, let us move on to their measurements. We start by giving an overview of the measurement setups used and of how the measurements were performed.

8.3.3 Measurement setups and measurement process

There were in total 20 different AWG designs for the first fabrication run. Each has 16 output channels, so this consists of 20x16 grating couplers to which fiber probes would need to be aligned. However, apart from some initial manual calibration, we can automate the measurements using computer-controlled X-Y-Z movable stages for the fiber probes, a power meter connected to the computer and a file containing the GCs coordinates. Such a measurement setup was available within the photonics group (Fig. 8.13).

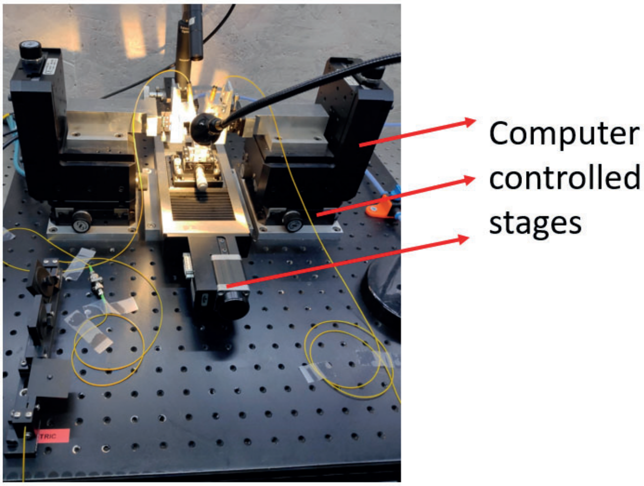


Figure 8.13: Semi-automatic setup: apart from initial manual calibration and the need for a coordinates file, a chip can be measured by a computer without need for human oversight.

However, alignment as judged by transmitted power was often poor compared to manual aligning. Thus, for results where accuracy was deemed important, the measurements were performed manually. The manual setup can consist of a tunable laser and an optical power meter connected to optical probes. In this case, the tunable laser is swept over the wavelength range to be measured. Such a sweep and accompanying data readout can take a while, as it takes a few seconds to measure one wavelength. Alternatively, one can make use of a broadband SLED in combination with a optical spectrum analyzer (OSA) to capture the data for the entire wavelength range at once. The latter technique is much faster and was thus preferred in case many measurements were needed.

However, using a tunable laser can sometimes be easier when aligning on the AWG, as one can tune the wavelength to the center wavelength of the channel being measured. This makes the expected output power, which is roughly the same for each channel, more predictable as compared to the SLED, where output power varies from channel to channel due to the interplay between the SLED spectrum, as shown in Fig. 8.14, and the channel spectrum.

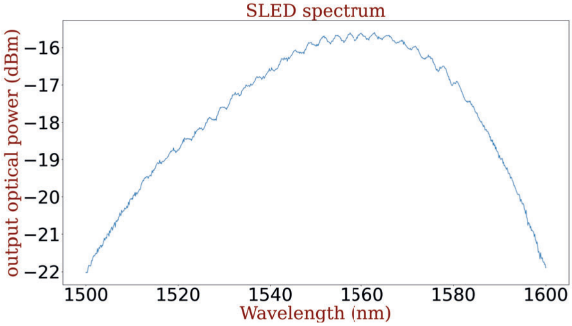


Figure 8.14: Optical spectrum of the SLED

If we do use an SLED, we need to manually normalise our results by compensating for its spectrum. Regardless of the light source, we also need to account for the spectrum of the grating couplers (Fig. 8.15) when we want to extract the performance of the AWG itself.

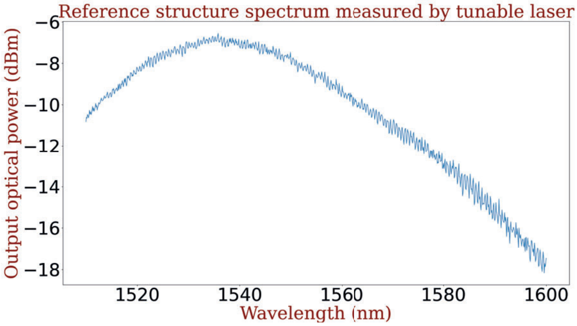


Figure 8.15: Optical spectrum of grating coupler measured with tunable laser.

In this thesis, the same grating coupler reference spectrum, measured after aligning for 1550 nm, was used to compensate all the AWG channels. However, as aligning the fibers to the grating couplers is wavelength dependent, it is recommended to compensate each AWG channel with a reference spectrum measured for its peak wavelength. In this way, one avoids overcompensating the AWG channels [126].

8.3.4 Single-heater design experimental results

First we present the results on the single-heater design (Fig. 8.12 (a)). For these measurements we opted to make use of the tunable laser light source.

The underlying AWG design was one with the aforementioned MMI input aperture for larger channel width and better flattop response. The FSR range was designed with 4 so-called dummy guard band channels in mind, so 20×500 GHz = 10 THz $\approx$ 80 nm. This was the maximum FSR among the design variations. This AWG design and heater combination was deemed to be the one most likely to best conform to the design requirements.

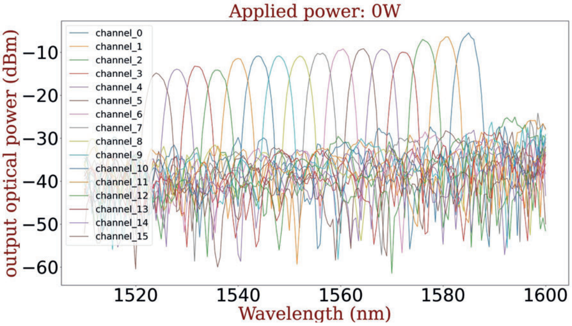


Figure 8.16: The AWG channels spectra with grating coupler spectrum compensation.

For the results of Fig. 8.16, the heater was not in use, so one can draw conclusions on the AWG design itself (Tab. 8.1).

Requirements	Simulations	Measurements	Conclusion
#channels=16	16	16	OK
FSR>16x500GHz	>16x500GHz	>16x500GHz	OK
Channel spacing=500GHz	500GHz	500GHz	OK
Peak deviation<<1nm	<1nm	1-10nm	active tuning needed
Channel width=480GHz with flat top response and steep roll-off	Imperfect, next-neighbor crosstalk $\approx$ 4.5dB	Imperfect, next-neighbor crosstalk $\approx$ 4.5dB	challenging to obtain ideal specification
Insertion loss	<4dB	<15dB	acceptable
Compact design	790x873 micron	790x873 micron	sufficiently compact

Table 8.1: AWG design requirements versus simulated and measured results

The measured characteristics are in line with our simulations. The AWG fulfils its demultiplexer role with each AWG output channel only having high transmission for a particular wavelength range. The FSR is large enough and the delay line waveguide spacing appears not to be problematic in terms of cross talk.

In terms of channel width and flattop it is clear that the 480 GHz width with perfect flattop response ideal is not reached, but this is no surprise, as this was a very challenging task. The next-neighbor crosstalk was calculated by looking for the maximum transmission, compensated for insertion loss, inside each channel due to its nearest neighbors. The channels for this operation are defined by $[\lambda_{peak} - \lambda_{spacing}/2, \lambda_{peak} + \lambda_{spacing}/2]$ with λ_{peak} being the channel peak wavelength and $\lambda_{spacing}$ being the channel spacing. There is a clear difference between simulated and measured insertion loss. In part, this is because of the choice of an MMI input aperture for better flattop response.

IV-curve measurement sweeps up to high powers were performed several times to ensure the heaters enduring integrity under high power dissipation scenarios. These showed that the single-heater design could reliably handle up to at least $\approx 5W$ and had a resistance of $\approx 236\Omega$ in its linear region where it is expected to operate (Fig. 8.17).

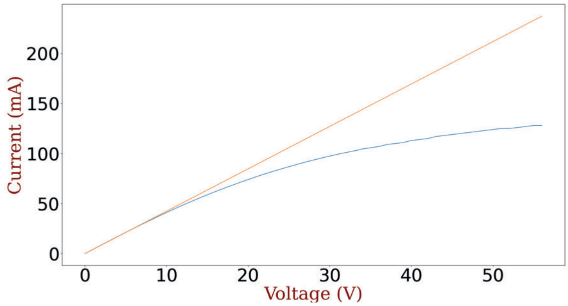


Figure 8.17: IV-curve of the single-heater design. Orange curve is IV-curve of ideal linear 236Ω resistance and blue curve is the measured heater IV-curve.

This tuning capability is demonstrated in Fig. 8.18. For various levels of dissipated power, one can see the wavelength channels shift to larger wavelengths with $\approx 2-3$ nm per Watt of power dissipation. This is in line with expectations based on our estimation in section 8.3.2.

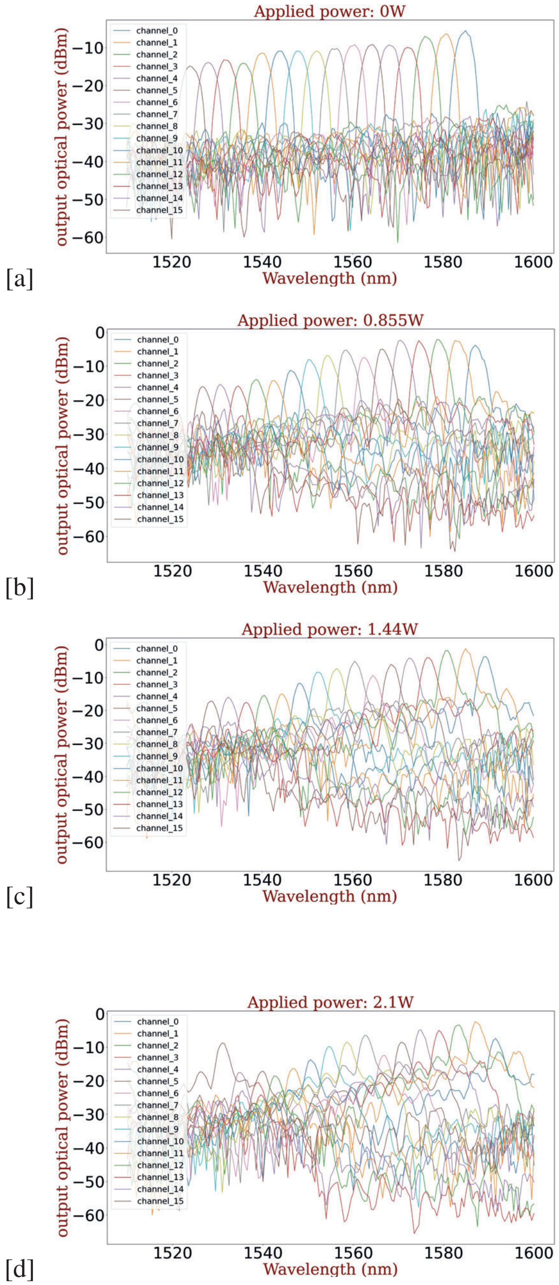


Figure 8.18: The AWG channels spectra for various levels of thermal tuning when using the single heater design (a) 0 W or no tuning (b) 0.855 W (c) 1.44 W (d) 2.1 W

The degradation of the demultiplexer performance at high powers is suspected to be mostly due to non-uniform heating of the delay lines, as explained in section 8.3.2. However, a more thorough investigation as to the causes, which could also include heat-spill into the star couplers, is kept in mind for future work, time constraints preventing its inclusion in this thesis. This performance degradation ultimately limits the maximum achievable wavelength shift to about 6 nm, i.e. 7-8% of FSR. We aimed to remedy this problem with the more detailed control we have over the heating using the multiple-heaters design, which we will discuss next.

8.3.5 Multiple-heater design experimental results

For the multiple-heater design, the underlying AWG was one without the input MMI aperture. In terms of FSR, this design again used our FSR of $\approx 80nm$.

For this measurement, we switched to using an SLED-OSA combination instead of making use of a tunable laser, mainly because with the multiple-heater design, there are a lot more heater configurations to test, making time-consuming wavelength sweeps impractical.

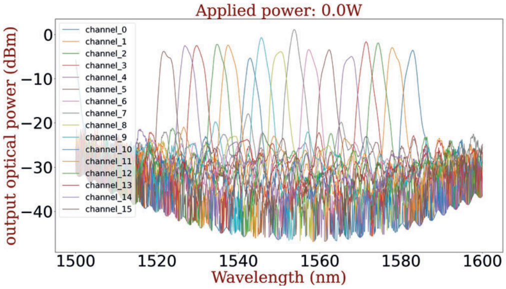


Figure 8.19: The AWG channels spectra with grating coupler and SLED spectra compensation

The results without power dissipation in the heaters are shown in Fig. 8.19. When we compare the results of the single-heater design with MMI (Fig. 8.16) to this design without MMI, we indeed see a decrease in channel width, a deterioration of the flattop response and a notable decrease of the insertion loss. For an exact estimate of the increased insertion loss due the MMI input aperture, it would be recommended to perform both measurements on MMI and non-MMI aperture designs using an identical light source. This is kept in mind for future work, time constraints preventing this being part of this thesis. If the increased insertion loss is acceptable, this indeed confirms that having a MMI input aperture is to the benefit of the AWG requirements. Whether insertion loss is acceptable or not depends in large part on the accumulated losses in the multiplication matrix preceding the AWG demultiplexers and this is part of ongoing research carried out by the project partners.

A second remark is that the position of the wavelength channels has shifted compared to Fig. 8.16. This is not caused by the difference in heater designs or input aperture, but by thickness variations over the wafer and differences in fabrication errors in the delay lines.

The IV-curves of each of the heaters in the multiple-heater design was measured so as to be able to correctly apply the desired current and thus heating (Fig. 8.20).

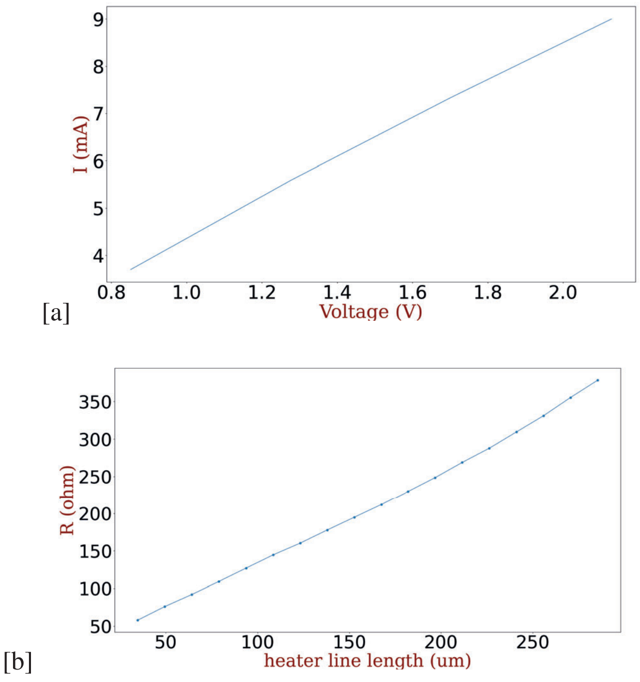


Figure 8.20: (a) IV-curve of a 182 micron heater in the multiple-heater design. (b) Resistance versus length of heater line.

We now proceed to measure the capability of the heater tuning to correct for such mistakes. In this case, there are a lot more options aside from just increasing the total dissipated power in the heater. For the first measurements, identical currents were applied to all the heaters. This is equivalent to treating the individual heaters as one big heater, i.e. a similar situation as the single-heater design (Fig. 8.21). This allows us to compare the heating uniformity of both designs.

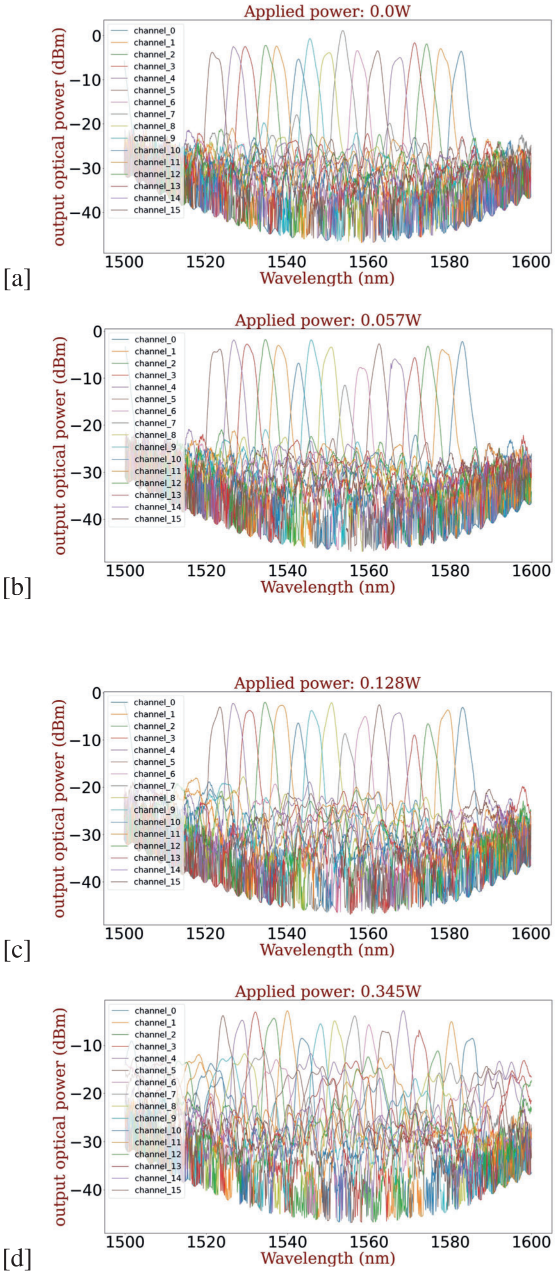


Figure 8.21: The AWG channels spectra for various levels of thermal tuning when driving all individual heaters at the same current. The mentioned powers refer to the total power dissipated in all heaters combined. (a) 0 W or no tuning (b) 0.057 W (c) 0.128 W (d) 0.345 W

Once again, the demultiplexer performance deteriorates with increasing power dissipation, and again non-uniform heating is to blame. Notice that when simply steering all the heaters with the same currents, the non-uniformity of the heating is considerably worse. The small amount of 0.345 W dissipated power already gives rise to significant performance breakdown, while the single-heater design was still working adequately at 0.855 W (compare Fig. 8.21 (d) to Fig. 8.18 (b)). The reason for this could be that the delay line coverage of the multiple-heater design is worse than that of the single-heater design. In particular, the distance between neighboring heater lines is rather large, which may make it difficult to maintain uniform temperature between them. Solving this problem requires a new generation of multiple-heater design which has better delay line coverage. Another concern is that the power dissipation is very low, leading to a very small maximum possible wavelength shift.

Next, measurements were performed where a current was sent through only one heater at a time. As can be seen from Fig. 8.22, neighboring channels seem to respond very differently to different heaters when looking at their wavelength shift. This illustrates that finding the optimal heater current configuration is not as straightforward as simply studying each heater in isolation.

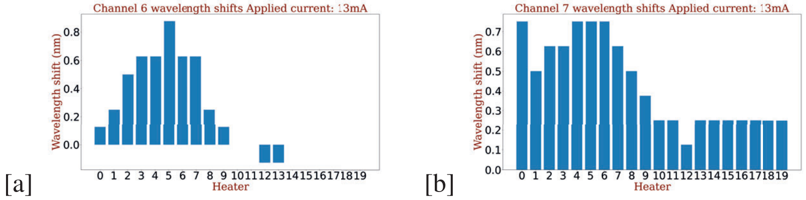


Figure 8.22: The wavelength shift for (a) channel 6 (b) channel 7, when sending current through different heaters. The results vary significantly. How the heaters were numbered is indicated in Fig. 8.12, i.e. the longest heater corresponds to heater 0 and the shortest to heater 19.

Further measurements on this multiple-heater design would have been steering the heater lines with total power ≈0.345W as in Fig. 8.21 (d), but with non equal currents through the heater lines to see to what extent heating uniformity could be improved despite the non-optimal design. However, because of time constraints, we opted to move on to designing new AWG demultiplexers and integrating them in the wider system, based on the obtained experimental results.

8.4 Second-generation design

Fig. 8.23 shows the second-generation layout for the single-heater design.

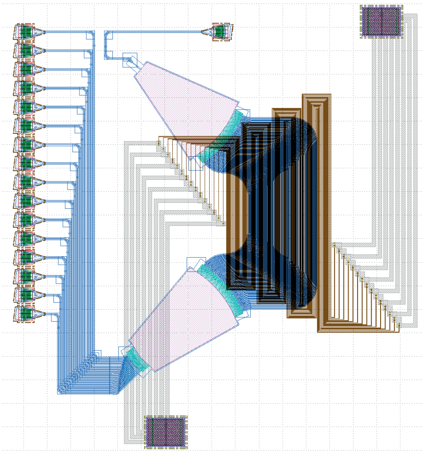


Figure 8.23: Second-generation single-heater design. Grey color indicates metal lines. Brown color indicates tungsten heater. Note in particular the long length and high number of heater lines.

This design has some key improvements that should make the heating more uniform:

- More heater lines in parallel, 15 instead of 10. This allows for more power dissipation, but puts this design at the very limit of the tungsten density allowed by the iSiPP200 PDK.
- Longer heater lines for better coverage. This comes at the cost of increased resistance per heater line and thus less power dissipation for a given voltage. However, this is partly offset by the increased number of heater lines.
- Small improvements to heater and metal line layouts to ensure all electrical paths between signal and ground bondpads have same resistance. This should ensure a more uniform heating.

This second-generation AWG design was combined with a 5×5 multiplication matrix not designed by the author (the original target of the project, a 25×25 matrix, was relaxed in the end). The multiplication matrix was made by the team of Prof. Dr. Wolfram Pernice, first at Muenster University and then at Heidelberg University. In particular we wish to acknowledge Frank Brückerhoff-Plückelmann, Lennart Meyer and Daniel Wendland for their contributions. Together we provided both a version, of the multiplication matrix with AWGs, with GCs and a version with PDs (Fig. 8.24).

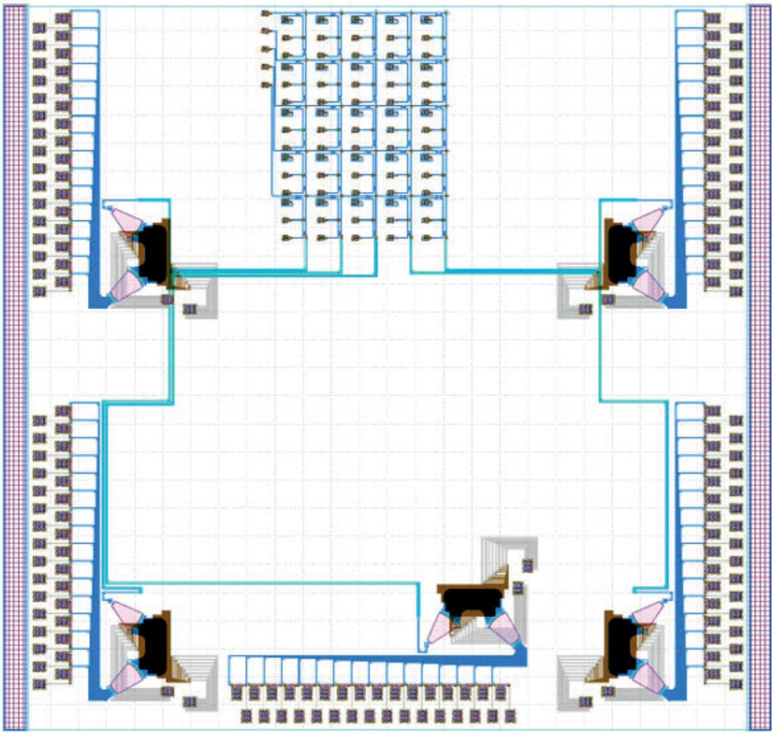


Figure 8.24: Single-heater second-generation design AWG demultiplexers integrated with multiplication matrix and PDs.

In response to both of the approaches in these first generation designs having shortcomings, another approach was devised. This design is a hybrid approach, trying to combine the single-heater design's less complicated electrical routing/interfacing and better delay line coverage with the multiple-heater design's more granular control over the heating across the delay lines (Fig. 8.25).

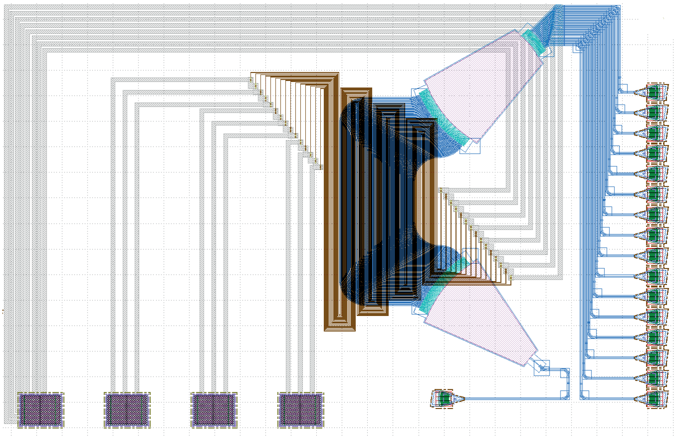


Figure 8.25: Hybrid-heater design. Grey color indicates metal lines. Brown color indicates tungsten heater. Note that there are 3 groups, consisting of 5 heater lines, in parallel each with their own signal bondpad.

This hybrid design is identical to the second-generation single-heater design, with the exception that it groups the heater lines in 3 groups of 5 heater lines and assigns each of these groups its own signal bondpad. There can thus be a different current sent through each group of heater lines, which allows for more control over the heating as compared to the regular single-heater design. This could help with heating uniformity. However, as each group of heater lines covers all delay lines, we lose the potential to finetune the heating for each individual delay line, which was part of the motivation behind the multiple-heater design.

These designs have just been taped out, and are yet to be fabricated.

9

Conclusion

9.1 Summary

The main goal of this work was to improve upon our existing waveguide-based PRC system by exploiting the wavelength dimension, in particular but not limited to making use of WDM. The motivation behind this was that WDM-enabled PRC systems might meet rising bandwidth requirements beyond the capability of traditional electronic-based computing systems.

This dissertation introduced the necessary concepts and gave an overview of the used PRC system. Then we started off with investigating WDM in such a system by way of simulations. Simulation results were promising, with a single-readout system shown to work well for many tens to hundreds of standard ITU-T channels for the industry-relevant task of signal equalization. We also accounted for readout phase information sensitivity, both by way of multiple-wavelength training and specially engineered reservoir interconnection lengths.

Building on these results, we turned to experiments where a first important step towards WDM in our PRC system was achieved. When using the FEC limit of $1.5\,10^{-2}$ as a benchmark, the single-readout system was shown to work well up to at least 3 standard ITU-T wavelengths for 28 Gbps (nonlinear) signal equalization tasks, using only limited post processing, i.e. a digitally implemented readout and the generating of delayed copies of measured node signals. We then proceeded to further explore PRC systems with an integrated optical readout. We showed for the first time the capability of such a system to perform online signal equalization, achieving around a factor of three improvement in BER compared to the case of signal equalization without the reservoir.

Having explored WDM, we investigated how by inclusion of nonlinear components, the nonlinearity of the system might be enhanced. Such an extra nonlinearity might be particularly useful when using true WDM signals, where there is cross-phase modulation or when using long-haul optical signals that experienced significant nonlinearity through being re-amplified several times over their journey. We described how such a chip can be designed and fabricated, as well some of the experimental concerns that would complicate a thorough analysis of this chip.

Finally, in the context of a European project undertaken in parallel to the thesis, an alternative photonic hardware computing platform, the photonic tensor core, was introduced. We showed how to design an AWG WDM demultiplexer for such a system, a component that might also play a role in our own WDM-enabled PRC systems as indicated in Fig. 5.1. The required AWG demultiplexer strained the limits of what is possible in terms of FSR, channel width, flat-top response and thermal tuning. Especially on the wavelength channel tuning part, significant research was performed centering on a single-heater and a multiple-heater design. Overall, first-generation designs largely fulfilled the requirements. Second-generation designs, focusing on further improvements of thermal tuning by way of an improved single-heater design and a hybrid-heater design, have just been taped out for fabrication.

9.2 Outlook

Important first steps were taken towards PRC systems with fully integrated optical readouts, capable of doing WDM signal equalization. We also explored systems with enhanced nonlinearity. Still, further research can be performed.

For the work done on the integrated optical readout in chapter 6, we would like to move from the single-wavelength case towards the multiple sequential wavelength case as was done for the non-integrated readout. Chips with lower loss are already available, and should prove very interesting to measure.

For the work done on the photonic reservoir with integrated SOAs in chapter 7, the measurements were postponed because of concerns involving chip integrity. Ideas were already presented in how to adapt the chip to resolve these concerns, after which we can proceed with the measurements. Again, the goal would be to first perform single-wavelength signal equalization followed by multiple-wavelength signal equalization. Once the chip has been shown to be able to perform in a similar fashion as to the integrated readout without SOAs chips, we would move on towards a more nonlinear task and try to measure the extra nonlinear processing capability offered by the inclusion of the SOAs. An example of such a more nonlinear task could simply be signal equalization but multiple longer fibers connected through amplifiers before the chip instead of our standard 25 km fiber.

Eventually we would of course like to move from using multiple sequential wavelengths towards true WDM signals, with all wavelengths being injected in parallel. This would allow us to do signal equalization taking into account all the nonlinearities of a real WDM system. In addition, it opens up the possibility of exploiting XPM and XGM in the SOA chip as a way of enhancing the reservoir nonlinearity.

However, true WDM measurements are quite challenging.

This work focused on signal equalization but in the future we would like to extend the work to other telecom applications like modulation format identification, anomaly detection and echo cancellation.

Concerning the work done on AWG demultiplexer design in chapter 8, the most important next step is measuring the as of yet to be fabricated second-generation designs. In the meantime, it could be worthwhile to revisit the multiple-heater design measurements. We could steer all the heater lines with non equal currents through the heater lines to see to what extent heating uniformity could be improved, despite the non-optimal design. If successful, this could inform the design of an improved second-generation multiple-heater design.

References

[1] M. Tan, B. Corcoran, X. Xu, J. Wu, A. Boes, T. G. Nguyen, S. T. Chu, B. E. Little, R. Morandotti, A. Mitchell, and D. J. Moss, "Ultra-high bandwidth optical data transmission with a microcomb," in *2020 International Topical Meeting on Microwave Photonics (MWP)*, pp. 78–82, 2020.

[2] D. A. B. Miller, "Attojoule optoelectronics for low-energy information processing and communications," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 346–396, 2017.

[3] M. Wolfgang, T. Natschläger, and H. Markram, "Real-time computing without stable states: a new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, pp. 2531–2560, November 2002.

[4] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Netw.*, vol. 20, pp. 391–403, April 2007.

[5] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, pp. 127–149, March 2009.

[6] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. L. Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. P. Pernice, and . H. Bhaskaran, "Parallel convolutional processing using an integrated photonic tensor core," *Nature*, vol. 589, p. 52–58, January 2021.

[7] F. Laporte, *Novel architectures for brain-inspired photonic computers*. Phd thesis, University of Ghent, Ghent, Belgium, March 2020. Available at `https://photonics.intec.ugent.be/publications/phd.asp?ID=259`.

[8] ITU-T, "G694.1: SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS Transmission media and optical systems characteristics – Characteristics of optical systems Spectral grids for

WDM applications: DWDM frequency grid." `https://www.itu.int/rec/T-REC-G.694.1-201202-I/en`.

[9] Juniper-Networks, "Forward error correction (fec) and bit error rate (ber)." `https://www.juniper.net/documentation/us/en/software/junos/interfaces-ethernet/topics/topic-map/fec-ber-otn-interfaces.html`.

[10] Paul Brodsky, "Internet traffic and capacity remain brisk." `https://blog.telegeography.com/internet-traffic-and-capacity-remain-brisk`. Accessed: 2023-06-2.

[11] G. Giamougiannis, A. Tsakyridis, M. Moralis-Pegios, G. Mourgias-Alexandris, A. R. Totovic, G. Dabos, M. Kirtas, N. Passalis, A. Tefas, D. Kalavrouziotis, D. Syrivelis, P. Bakopoulos, E. Mentovich, D. Lazovsky, and N. Pleros, "Neuromorphic silicon photonics with 50 ghz tiled matrix multiplication for deep-learning applications," *Advanced Photonics*, vol. 5, February 2023.

[12] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, p. 441–446, June 2017.

[13] D. M. Marom, R. Ryf, and D. T. Neilson, "Chapter 16 - networking and routing in space-division multiplexed systems," in *Optical Fiber Telecommunications VII* (A. E. Willner, ed.), pp. 719–750, Academic Press, 2020.

[14] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[15] E. Alpaydın, *Introduction to Machine Learning, Fourth Edition*. The MIT Press, 2020.

[16] E. Gooskens, F. Laporte, C. Ma, S. Sackesyn, J. Dambre, and P. Bienstman, "Wavelength dimension in waveguide-based photonic reservoir computing," *Optics Express*, vol. 30, no. 9, p. 15 634–15 647, 2022.

[17] G. V. der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, pp. 561–576, May 2017.

[18] D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger, "Tutorial: Photonic neural networks in delay systems," *Journal of Applied Physics*, vol. 124, p. 152004, October 2018.

[19] L. Appeltant, M. C. Soriano, G. V. der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, “Information processing using a single dynamical node as complex system,” *Nature Communications*, vol. 2, p. 468, September 2011.

[20] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, “Optoelectronic reservoir computing,” *Sci. Rep.*, vol. 2, p. 287, February 2012.

[21] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, “Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing,” *Opt. Express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[22] A. S. F. Duport, B. Schneider, M. Haelterman, and S. Massar, “All-optical reservoir computing,” *Opt. Express*, vol. 20, no. 20, pp. 22783–22795, 2012.

[23] Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar, “High-performance photonic reservoir computer based on a coherently driven passive cavity,” *Optica*, vol. 2, no. 5, pp. 438–446, 2015.

[24] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, “Parallel photonic information processing at gigabyte per second data rates using transient states,” *Nature Communications*, vol. 4, p. 1364, January 2013.

[25] A. Dejonckheere, A. Smerieri, L. Fang, J. I. Oudar, M. Haelterman, and S. Massar, “All-optical reservoir computer based on saturation of absorption,” *Opt. Express*, vol. 22, no. 9, pp. 10868–10881, 2014.

[26] M. C. Soriano, S. Ortin, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, “Optoelectronic reservoir computing: Tackling noise-induced performance degradation,” *Opt. Express*, vol. 21, no. 1, pp. 12–20, 2013.

[27] R. M. Nguimdo, G. Verschaffelt, J. Danckaert, and G. V. der Sande, “Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics,” *Opt. Express*, vol. 22, no. 7, pp. 8672–8686, 2014.

[28] K. Hicke, M. Escalona-Morán, D. Brunner, M. C. Soriano, I. Fischer, , and C. R. Mirasso, “Information processing using transient dynamics of semiconductor lasers subject to delayed feedback,” *IEEE J. Sel. Top. Quantum Electron*, vol. 19, no. 4, pp. 1501610–1501610, 2013.

[29] R. M. Nguimdo and T. Erneux, “Enhanced performances of a photonic reservoir computer based on a single delayed quantum cascade laser,” *Optics Letters*, vol. 44, no. 1, pp. 49–52, 2019.

[30] Y. Hou, G. Xia, W. Yang, D. Wang, E. Jayaprasath, Z. Jiang, C. Hu, and Z. Wu, “Prediction performance of reservoir computing system based on a semiconductor laser subject to double optical feedback and optical injection,” *Opt. Express*, vol. 26, no. 8, pp. 10211–10219, 2018.

[31] J. Vatin, D. Rontani, and M. Sciamanna, “Enhanced performance of a reservoir computer using polarization dynamics in vcsels,” *Optics Letters*, vol. 43, no. 18, pp. 4497–4500, 2018.

[32] J. Pauwels, G. Verschaffelt, S. Massar, and G. V. der Sande, “Distributed kerr non-linearity in a coherent all-optical fiber-ring reservoir computer,” *Frontiers in Physics*, vol. 7, october 2019.

[33] G. Mourgias-Alexandris, G. Dabos, N. Passalis, A. Totović, A. Tefas, and N. Pleros, “All-optical wdm recurrent neural networks with gating,” *IEEE J. Sel. Top. Quantum Electron*, vol. 26, pp. 1–7, May 2020.

[34] A. Argyris, J. Bueno, and I. Fischer, “Photonic machine learning implementation for signal recovery in optical communications,” *Sci. Rep.*, vol. 8, p. 8487, May 2018.

[35] A. Lugnan, A. Katumba, F. Laporte, M. Freiberger, S. Sackesyn, C. Ma, E. Gooskens, J. Dambre, and P. Bienstman, “Photonic neuromorphic information processing and reservoir computing,” *APL Photonics*, vol. 5, p. 020901, February 2020.

[36] L. Butschek, A. Akrout, E. Dimitriadou, A. Lupo, M. Haelterman, and S. Massar, “Photonic reservoir computer based on frequency multiplexing,” *Opt. Lett.*, vol. 47, pp. 782–785, Feb 2022.

[37] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, “All-optical machine learning using diffractive deep neural networks,” *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.

[38] K. Vandoorne, P. Mechet, T. V. Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, “Experimental demonstration of reservoir computing on a silicon photonics chip,” *Nature Communications*, vol. 5, pp. 1–6, March 2014.

[39] A. Katumba, M. Freiberger, F. Laporte, A. Lugnan, S. Sackesyn, C. Ma, J. Dambre, and P. Bienstman, “Neuromorphic computing based on silicon

photonics and reservoir computing," *IEEE J. Sel. Top. Quantum Electron*, vol. 24, pp. 1–10, April 2018.

[40] S. Sackesyn, C. Ma, J. Dambre, and P. Bienstman, "Experimental realization of integrated photonic reservoir computing for nonlinear fiber distortion compensation," *Opt. Express*, vol. 29, no. 20, pp. 30991 – 30997, 2021.

[41] S. Masaad, E. Gooskens, S. Sackesyn, J. Dambre, and P. Bienstman, "Photonic reservoir computing for nonlinear equalization of 64-qam signals with a kramers–kronig receiver," *Nanophotonics*, vol. 12, p. 925–935, October 2022.

[42] C. Ma, J. V. Kerrebrouck, H. Deng, S. Sackesyn, E. Gooskens, B. Bai, J. Dambre, and P. Bienstman, "Integrated photonic reservoir computing with an all-optical readout," *Opt. Express*, vol. 31, pp. 34843–34854, Oct 2023.

[43] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific Reports*, vol. 8, February 2018.

[44] Katumba, Andrew, *Energy-efficient photonic neuromorphic computing for telecommunication applications*. Phd thesis, University of Ghent, 2019. Available at `https://biblio.ugent.be/publication/8622749`.

[45] Lugnan, Alessio, *Photonics-based machine learning to speed up and simplify label-free flow cytometry*. Phd thesis, Ghent University, 2021. Available at `https://biblio.ugent.be/publication/8722653`.

[46] OpenAI, "Introducing chatgpt." `https://openai.com/blog/chatgpt`. Accessed: 2023-07-25.

[47] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, p. 406–413, 1955.

[48] Arun Addagatla, "Investigating underfitting and overfitting." `https://medium.com/geekculture/investigating-underfitting-and-overfitting-70382835e45c`. Accessed: 2023-10-18.

[49] L. Appeltant, *Reservoir computing based on delay-dynamical systems*. Phd thesis, Joint PhD Vrije Universiteit Brussel and Universitat de les Illes Balears, Brussel, Belgium, May 2012. Available at `https://ibdigital.uib.es/greenstone/collect/tesisUIB/index/assoc/TDX-1080/3-84144.dir/TDX-10803-84144.pdf`.

[50] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, March 2021.

[51] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. K. andTom Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada*, 2020.

[52] K. Martineau, "Shrinking deep learning's carbon footprint." `https://news.mit.edu/2020/shrinking-deep-learning-carbon-footprint-0807`. Accessed: 2023-08-23.

[53] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 1916–1920, 2017.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, p. 533–536, October 1986.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[56] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[57] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[58] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, p. 107–116, apr 1998.

[59] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *Advances in Artificial Life* (W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim, eds.), (Berlin, Heidelberg), pp. 588–597, Springer Berlin Heidelberg, 2003.

[60] D. V. Thourhout and R. Baets, "Microphotonics syllabus, ghent university," 2018-2019.

[61] B. Jalali and S. Fathpour, "Silicon photonics," *Journal of Lightwave Technology*, vol. 24, no. 12, pp. 4600–4615, 2006.

[62] imec, "Silicon photonic ics for prototyping and volume production." `https://www.imec-int.com/en/silicon-photonic-ICs-prototyping`. Accessed: 2023-07-27.

[63] Ligentec, "Light generating technologies using silicon nitride photonic integrated circuits (pics)." `https://www.ligentec.com/`. Accessed: 2023-07-27.

[64] R. Baets, A. Z. Subramanian, S. Clemmen, B. Kuyken, P. Bienstman, N. Le Thomas, G. Roelkens, D. Van Thourhout, P. Helin, and S. Severi, "Silicon photonics: Silicon nitride versus silicon-on-insulator," in *2016 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, 2016.

[65] T. Vanackere, T. Vandekerckhove, L. Bogaert, M. Billet, S. Poelman, S. Cuyvers, J. Van Kerrebrouck, A. Moerman, O. Caytan, N. Singh, S. Lemey, G. Torfs, P. Ossieur, G. Roelkens, S. Clemmen, and B. Kuyken, "Heterogeneous integration of a high-speed lithium niobate modulator on silicon nitride using micro-transfer printing," *APL Photonics*, vol. 8, p. 086102, 08 2023.

[66] T. Vandekerckhove, T. Vanackere, J. D. Witte, S. Cuyvers, L. Reis, M. Billet, G. Roelkens, S. Clemmen, and B. Kuyken, "Reliable micro-transfer printing method for heterogeneous integration of lithium niobate and semiconductor thin films," *Opt. Mater. Express*, vol. 13, pp. 1984–1993, Jul 2023.

[67] G. Roelkens, J. Zhang, L. Bogaert, M. Billet, D. Wang, B. Pan, C. J. Kruckel, E. Soltanian, D. Maes, T. Vanackere, T. Vandekerckhove, S. Cuyvers, J. De Witte, I. L. Lufungula, X. Guo, H. Li, S. Qin, G. Muliuk, S. Uvin, B. Haq, C. Op de Beeck, J. Goyvaerts, G. Lepage, P. Verheyen, J. Van Campenhout, G. Morthier, B. Kuyken, D. Van Thourhout, and R. Baets, "Micro-transfer printing for heterogeneous si photonic integrated circuits," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 29, no. 3: Photon. Elec. Co-Inte. and Adv. Trans. Print., pp. 1–14, 2023.

[68] E. Soltanian, G. Muliuk, S. Uvin, D. Wang, G. Lepage, P. Verheyen, J. V. Campenhout, S. Ertl, J. Rimböck, N. Vaissiere, D. Néel, J. Ramirez, J. Decobert, B. Kuyken, J. Zhang, and G. Roelkens, "Micro-transfer-printed narrow-linewidth iii-v-on-si double laser structure with a combined 110 nm tuning range," *Opt. Express*, vol. 30, pp. 39329–39339, Oct 2022.

[69] B. Haq, J. R. Vaskasi, J. Zhang, A. Gocalinska, E. Pelucchi, B. Corbett, and G. Roelkens, "Micro-transfer-printed iii-v-on-silicon c-band distributed feedback lasers," *Opt. Express*, vol. 28, pp. 32793–32801, Oct 2020.

[70] R. Soref, J. Schmidtchen, and K. Petermann, "Large single-mode rib waveguides in gesi-si and si-on-sio/sub 2/," *IEEE Journal of Quantum Electronics*, vol. 27, no. 8, pp. 1971–1974, 1991.

[71] W. Bogaerts and S. K. Selvaraja, "Compact single-mode silicon hybrid rib/strip waveguide with adiabatic bends," *IEEE Photonics Journal*, vol. 3, no. 3, pp. 422–432, 2011.

[72] W. Bogaerts, "photonic integrated circuits course jupyter notebooks, ghent university," 2021-2022.

[73] L. Cheng, S. Mao, Z. Li, Y. Han, and H. Y. Fu, "Grating couplers on silicon photonics: Design principles, emerging trends and practical issues," *Micromachines*, vol. 11, no. 7, 2020.

[74] L. Cheng, S. Mao, Z. Li, Y. Han, and H. Y. Fu, "Grating couplers on silicon photonics: Design principles, emerging trends and practical issues," *Micromachines*, vol. 11, no. 7, 2020.

[75] M. Kobayashi, *Guided-wave Optical Interconnect Techniques*. Elsevier Inc., 1993.

[76] L. Photonics, "Multi mode interferometers." `https://docs.lucedaphotonics.com/picazzo/filters/mmi/`. Accessed: 2023-07-25.

[77] M. Blahut and D. Kasprzak, "Multimode interference structures – properties and applications," *Optica Applicata*, vol. 34, no. 4, pp. 573–587, 2004.

[78] Stijn Sackesyn and Chonghuai Ma and Andrew Katumba and Joni Dambre and Peter Bienstman, "A Power-Efficient Architecture for On-Chip Reservoir Computing." Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions, pp 161–164.

[79] Stijn Sackesyn and Chonghuai Ma and Joni Dambre and Peter Bienstman, "An enhanced architecture for silicon photonic reservoir computing." Cognitive Computing 2018 - Merging Concepts with Hardware (2018), pp.1-2.

[80] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cogn Comput*, vol. 9, no. 3, pp. 307–314, 2017.

[81] Pathak, Shibnath and Van Thourhout, Dries and Bogaerts, Wim, “Design trade-offs for silicon-on-insulator-based AWGs for (de)multiplexer applications,” *OPTICS LETTERS*, vol. 38, no. 16, pp. 2961–2964, 2013.

[82] Y. Xing, J. Dong, S. Dwivedi, U. Khan, and W. Bogaerts, “Accurate extraction of fabricated geometry using optical measurment,” *Photonics Research*, vol. 6, pp. 1008–1020, November 2018.

[83] Q. Han, D. Robin, A. Gervais, M. Ménard, and W. Shi, “Phase errors and statistical analysis of silicon-nitride arrayed waveguide gratings,” *Opt. Express*, vol. 30, pp. 42784–42800, Nov 2022.

[84] C. Ma, F. Laporte, J. Dambre, and P. Bienstman, “Addressing limited weight resolution in a fully optical neuromorphic reservoir computing readout,” *Scientific Reports*, vol. 11, p. 3102, February 2021.

[85] M. Pantouvaki, S. A. Srinivasan, Y. Ban, P. D. Heyn, P. Verheyen, G. Lepage, H. Chen, J. D. Coster, N. Golshani, S. Balakrishnan, P. Absil, and J. V. Campenhout, “Active components for 50 gb/s nrz-ook optical interconnects in a silicon photonics platform,” *J. Lightwave Technol.*, vol. 35, pp. 631–638, Feb 2017.

[86] A. Masood, M. Pantouvaki, G. Lepage, P. Verheyen, J. V. Campenhout, P. Absil, D. V. Thourhout, and W. Bogaerts, “Comparison of heater architectures for thermal control of silicon photonic circuits,” in *Proceedings of the 10th International Conference on Group IV Photonics*, 2013.

[87] P. Edinger, A. Y. Takabayashi, C. Errando-Herranz, U. Khan, H. Sattari, P. Verheyen, W. Bogaerts, N. Quack, and K. B. Gylfason, “Silicon photonic microelectromechanical phase shifters for scalable programmable photonics,” *Opt. Lett.*, vol. 46, pp. 5671–5674, Nov 2021.

[88] P. S. STARK, *Integrated Photonic Computing*. PhD thesis, Besançon, France, October 2021. Available at https://cloud.femto-st.fr/nextcloud/index.php/s/qFQ48JrRAQdSbyd.

[89] J. Geler-Kremer, F. Eltes, P. Stark, D. Stark, D. Caimi, H. Siegwart, B. J. Offrein, J. Fompeyrine, and S. Abel, “A ferroelectric multilevel non-volatile photonic phase shifter,” *nature photonics*, vol. 16, no. 7, 2022.

[90] D. Melati, F. Morichetti, A. Canciamilla, D. Roncelli, F. M. Soares, A. Bakker, and A. Melloni, “Validation of the building-block-based approach for the design of photonic integrated circuits,” *J. Lightwave Technol.*, vol. 30, pp. 3610–3616, Dec 2012.

[91] M. Fiers, T. V. Vaerenbergh, K. Caluwaerts, D. V. Ginste, B. Schrauwen, J. Dambre, and P. Bienstman, “Time-domain and frequency-domain modeling of nonlinear optical components at the circuit-level using a node-based approach,” *J. Opt. Soc. Am. B*, vol. 29, pp. 896–900, May 2012.

[92] L. Photonics, “Caphe introduction.” `https://docs.lucedaphotonics.com/guides/caphe/`.

[93] A. Lumerical, “Ansys lumerical interconnect.” `https://www.ansys.com/products/photonics/interconnect`.

[94] VPI, “Photonics design automation.” `https://www.vpiphotonics.com/index.php`. Accessed: 2023-07-27.

[95] F. Laporte, J. Dambre, and P. Bienstman1, “Highly parallel simulation and optimization of photonic circuits in time and frequency domain based on the deep-learning framework pytorch,” *Sci. Rep.*, vol. 9, p. 5918, April 2019.

[96] F. Laporte, “Photontorch.” `https://docs.photontorch.com/`.

[97] T. E. Oliphant, “Guide to numpy.” `https://web.mit.edu/dvp/Public/numpybook.pdf`. Accessed: 2023-07-27.

[98] K. Sozos, A. Bogris, P. Bienstman, G. Sarantoglou, S. Deligiannidis, and C. Mesaritakis, “High-speed photonic neuromorphic computing using recurrent optical spectrum slicing neural networks,” *Communications Engineering*, vol. 1, pp. 1–10, October 2022.

[99] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization.” `https://arxiv.org/abs/1711.05101`.

[100] M. Jeruchim, “Techniques for estimating the bit error rate in the simulation of digital communication systems,” *IEEE Journal on Selected Areas in Communications*, vol. 2, pp. 153–170, January 1984.

[101] Menara Networks, “Datasheet 5ZR0A00-TNBL.” `http://menaranet.com/index.php?route=information/information&information_id=24`.

[102] E. Ip and J. M. Kahn, “Compensation of dispersion and nonlinear impairments using digital backpropagation,” *Journal of Lightwave Technology*, vol. 26, p. 3416–3425, October 2008.

[103] J. Bergstra, D. Yamins, and D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures,” in *Proceedings of the 30th International Conference on Machine*

Learning (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 115–123, PMLR, 17–19 Jun 2013.

[104] F. Nogueira, “Bayesian Optimization: Open source constrained global optimization tool for Python,” 2014.

[105] J. Bergstra, D. Yamins, and D. Cox, “Hyperopt: Distributed hyperparameter optimization,” 2013.

[106] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[107] Lumiphase, “Lumiphase:enabling every chip to interact through light.”

[108] N. Hansen, “The cma evolution strategy: A tutorial,” 2023.

[109] M. Shibata, “Lightweight covariance matrix adaptation evolution strategy (cma-es) implementation for python 3.,” 2023.

[110] N. Olsson, “Lightwave systems with optical amplifiers,” *Journal of Lightwave Technology*, vol. 7, no. 7, pp. 1071–1082, 1989.

[111] T. Saitoh, Y. Suzuki, and H. Tanaka, “Low noise characteristics of a gaas-algaas multiple-quantum-well semiconductor laser amplifier,” *IEEE Photonics Technology Letters*, vol. 2, no. 11, pp. 794–796, 1990.

[112] B. Haq, *III-V-on-Si SOAs and DFB/DBR Lasers Realised Using Micro-Transfer Printing*. PhD thesis, Ghent, Belgium, June 2021. Available at http://www.photonics.intec.ugent.be/publications/PhD.asp?ID=267.

[113] G. P. Agrawal, *Fiber-Optic Communication Systems, 4th Edition*. JOHN WILEY and SONS, INC., 2010.

[114] G. Agrawal and N. Olsson, “Self-phase modulation and spectral broadening of optical pulses in semiconductor laser amplifiers,” *IEEE Journal of Quantum Electronics*, vol. 25, no. 11, pp. 2297–2306, 1989.

[115] P. P. Baveja, D. N. Maywar, A. M. Kaplan, and G. P. Agrawal, “Self-phase modulation in semiconductor optical amplifiers: Impact of amplified spontaneous emission,” *IEEE Journal of Quantum Electronics*, vol. 46, no. 9, pp. 1396–1403, 2010.

[116] R. Inohara, K. Nishimura, M. Tsurusawa, and M. Usami, “Experimental analysis of cross-phase modulation and cross-gain modulation in soa-injecting cw assist light,” *IEEE Photonics Technology Letters*, vol. 15, no. 9, pp. 1192–1194, 2003.

[117] A. Marculescu, S. O. Dúill, C. Koos, W. Freude, and J. Leuthold, “Spectral signature of nonlinear effects in semiconductor optical amplifiers,” *Opt. Express*, vol. 25, pp. 29526–29559, Nov 2017.

[118] MicroChemicals, “Ti 35e image reversal resist technical data sheet,” 2016.

[119] Y. WU, Q. HUANG, S. KEYVANINIA, A. KATUMBA, J. ZHANG, W. XIE, G. MORTHIER, J.-J. HE, and G. ROELKENS, “All-optical wavelength conversion of 40 gb/s nrz data in a heterogeneously integrated iii-v-on-silicon soa,” *Optics Express*, vol. 24, September 2016.

[120] J. Shah, “Estimating bond wire current-carrying capacity.” `https://www.renesas.com/us/en/document/atc/power-systems-design-estimating-bond-wire-current-carrying-capacity`. Accessed: 2023-08-01.

[121] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 770–778, IEEE, 2016.

[122] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[123] X. Li, N. Youngblood, C. Ríos, Z. Cheng, C. D. Wright, W. H. Pernice, and H. Bhaskaran, “Fast and reliable storage using a 5 bit, nonvolatile photonic memory cell,” *Optica*, vol. 6, pp. 1–6, Jan 2019.

[124] J. Feldmann, M. Stegmaier, N. Gruhler, C. Ríos, H. Bhaskaran, C. D. Wright, and W. H. P. Pernice, “Calculating with light using a chip-scale all-optical abacus.,” *Nature Communications*, vol. 8, p. 1256, November 2017.

[125] T. Herr, V. Brasch, J. D. Jost, C. Y. Wang, N. M. Kondratiev, M. L. Gorodetsky, and T. J. Kippenberg, “Temporal solitons in optical microresonators,” *Nature Photonics*, vol. 8, pp. 145–152, February 2014.

[126] S. Pathak, *Silicon Nano-Photonics Based Arrayed Waveguide Gratings*. Phd thesis, University of Ghent, Ghent, Belgium, 2014. Available at `https://biblio.ugent.be/publication/5769227`.

[127] M. Smit and C. Van Dam, “Phasar-based wdm-devices: Principles, design and applications,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 2, no. 2, pp. 236–250, 1996.

[128] L. Photonics, “Luceda awg designer.” `https://www.lucedaphotonics.com/luceda-awg-designer`. Accessed: 2023-07-31.

[129] J. Soole, M. Amersfoort, H. Leblanc, N. Andreadakis, A. Rajhel, C. Caneau, R. Bhat, M. Koza, C. Youtsey, and I. Adesida, “Use of multimode interference couplers to broaden the passband of wavelength-dispersive integrated wdm filters,” *IEEE Photonics Technology Letters*, vol. 8, no. 10, pp. 1340–1342, 1996.

[130] D. Dai, W. Mei, and S. He, “Using a tapered mmi to flatten the passband of an awg,” *Optics Communications*, vol. 219, no. 1, pp. 233–239, 2003.

[131] A. Masood, M. Pantouvaki, D. Goossens, G. Lepage, P. Verheyen, J. V. Campenhout, P. Absil, D. V. Thourhout, and W. Bogaerts, “Fabrication and characterization of cmos-compatible integrated tungsten heaters for thermo-optic tuning in silicon photonics devices,” *Opt. Mater. Express*, vol. 4, pp. 1383–1388, Jul 2014.