Universiteit Gent
Faculteit Ingenieurswetenschappen
Vakgroep Informatietechnologie

Photonic Reservoir Computing with a Network of Coupled Semiconductor Optical Amplifiers

Fotonisch Reservoir Computing met een netwerk van gekoppelde optische halfgeleiderversterkers

Kristof Vandoorne

Proefschrift tot het bekomen van de graad van
Doctor in de Ingenieurswetenschappen:
Fotonica
Academiejaar 2010-2011

Promotoren:

Prof. Dr. Ir. Peter Bienstman
Prof. Dr. Ir. Joni Dambre

Examencommissie:

Prof. Dr. Ir. Daniël De Zutter (voorzitter)	Universiteit Gent, INTEC
Prof. Dr. Ir. Peter Bienstman (promotor)	Universiteit Gent, INTEC
Prof. Dr. Ir. Joni Dambre (promotor)	Universiteit Gent, ELIS
Prof. Dr. Ir. Geert Morthier (secretaris)	Universiteit Gent, INTEC
Prof. Dr. Ir. Benjamin Schrauwen	Universiteit Gent, ELIS
Dr. Bart Wyns	Universiteit Gent, SYSTeMS
Prof. Dr. Gordon Pipa	Universität Osnabrück, IKW
Prof. Dr. Serge Massar	Université Libre de Bruxelles, LIQ

Universiteit Gent
Faculteit Ingenieurswetenschappen

Vakgroep Informatietechnologie
Sint-Pietersnieuwstraat 41, B-9000 Gent, België

Tel.: +32-9-264.33.39
Fax.: +32-9-331.35.93

Dit werk kwam tot stand in het kader van een
specialisatiebeurs van het Bijzonder Onderzoeksfonds (BOF)
van de Universiteit Gent

Dankwoord

'10 years, man, 10. Where have you been for 10 years?', zegt John Cusack's beste vriend in Grosse Point Blank wanneer Cusack na tien jaar terug in zijn geboortestad verzeild raakt. Om het in zijn eigen woorden te zeggen 'I freaked out, joined the army, went into business for myself, I am a professional killer.' Mijn bezigheden van de laatste vijf jaar zijn van een iets menslievender aard en het resultaat staat in de rest van dit boekje.

Er was een jongen die in 2005 plots het idee in zijn hoofd geplant kreeg (door een mail over studiebeurzen in het buitenland) om naar Japan te gaan. Zonder Roel Baets zou het waarschijnlijk bij een droom gebleven zijn, maar hij stelde me een heel uitdagend en nieuw doctoraatsonderzoek voor waar ik gretig op in ging. Ik hoop dat hij er geen spijt van heeft, ik alvast niet. Zonder zijn jaren bouwen aan wat waarschijnlijk nu de grootste fotonica onderzoeksgroep is van België, zou het niet mogelijk geweest zijn.

Gedurende dit doctoraat stonden vooral twee mensen me bij, ook wel algemeen bekend onder de term 'promotoren': Peter Bienstman en Joni Dambre. Peter moet ik voor zo veel bedanken, maar vooral voor zijn onvoorwaardelijke steun, subtiele bijsturingen en ongebreidelde optimisme. Zeker toen het in Japan wat moeilijker ging. Een doctoraat is vaak een beetje jezelf versus de wereld en dan is het fijn dat je toch iemand hebt die weet wat je doet en je steunt. Joni is pas later betrokken geraakt bij mijn onderzoek en het is vooral achteraf dat ik bemerk hoe zeer al je opmerkingen en nauwkeurigheid altijd tot een beter resultaat geleid hebben. Bovendien wil ik je bedanken om interesse te tonen op het moment dat het Euraziatische continent ons scheidde.

I also want to thank the other members of my jury, Geert Morthier, Benjamin Schrauwen, Bart Wyns, Serge Massar and Gordon Pipa for their questions, comments and suggestions and chairman Daniël Dezutter for his chairing. Arai-sensei and Nishiyama-sensei for hosting me for an entire year at Tokyo Institute of Technology and people like Kasukawa-san, Okumura-san, Takei-san, Mizuki-san, Inoue-san, Atsumi-san, Ito-san, Shindo-san, Maeda-san, Mizuho and Murat for making that year so much more agreeable. Een iemand die hier zeker een

speciale vermelding verdient is David Verstraeten van het reservoir lab, wiens cruciale ondersteuning niet voldoende benadrukt kan worden en waar ik naast antwoorden op mijn vele vragen, meestal ook een inspirerend gesprek kon verwachten.

Hoewel ik mezelf nog altijd niet zo voel, ben ik eigenlijk een ancien in de groep en in die vijf jaar heb ik veel mensen zien komen en gaan, te veel om op te noemen, zeker omdat de groep in die jaren veranderd is van een eerder kleine, hechtere groep naar een geheel van bijna 50 doctoraatsstudenten met de altijd vriendelijke en ondersteunende rol van de Ilses en Kristien. Maar de groep op zich is een misschien onderschat deel van het doctoreren. Er wordt soms gezegd 'no man is an island' (als eerste door John Donne volgens wikipedia), hoewel dat soms wel verwacht wordt van een doctoraatsstudent. Het praktische nut dat je van elkaars ervaringen hebt, is eigenlijk nog het minste.

In het begin werd ik al vlug onder de vleugels genomen door de oude garde, Peter VDS en Wouter VP, en samen hebben ze er voor gezorgd dat Gent niet enkel een woonst in de week was (zoals dat normaal is voor studenten) maar mijn echte thuis werd, ook in het weekend. In dat eerste jaar groeide ik steeds meer toe naar mijn jaargenoten Wout en Koen en dat resulteerde in een heerlijk jaar samen als bureaugenootjes in mijn tweede jaar. In die twee jaar en ook in het jaar Japan was Nathalie naast het werk mijn vaste rots in de branding waar ik zonder spijt op terugkijk.

Verder ben ik in die jaren een bont allegaartje van interessante persoonlijkheden tegengekomen op het technicum. Mensen zoals Shankar, mijn favoriete Tamil (ook wel de enige die ik ken), Stevan, de geschiedenisleerkracht die ik nooit gehad heb, Gunther, misschien ben je nu wel een professor maar in mijn hart zul je altijd een vuile postdoc blijven, Martin en Thomas, mijn opvolgers, Pauline en Nannicha, who love to hate me, Sam, de meest rechtse linkse rakker die ik ken en Joris, blond and boring don't go together bitch, en ongetwijfeld anderen die ik nu vergeet zoals Katrien, Dave en Lieven, Joost, Jonathan en Kasia. De laatste twee jaar was ik omringd door mensen waar ik een ongelooflijk grote genegenheid voor ben gaan voelen en die ik langzamerhand zou plegen te bestempelen als vrienden in plaats van collega's: Wout, zelden is een voorgevoel over iemand meer bewaarheid geworden dan bij jou, Marie, de onpeilbare diepte, Elewout, die ik liever heb dan ie zich waarschijnlijk kan inbeelden[1], Cristina, die soms het gewicht van de wereld op haar schouders draagt om het er gelukkig ook af en toe eens af te dansen en Karel, die ik even graag heb als de anderen. Ik ben benieuwd wat de tijd hiermee zal aanvangen.

Nabijheid maakt samen optrekken natuurlijk makkelijker en daarom wil ik ook al mijn vrienden bedanken die geografisch en professioneel verder van me

[1] http://www.youtube.com/watch?v=lL4L4Uv5rf0

afstaan maar die ik gelukkig veel meer zie dan enkel op trouwfeesten alleen en die ik, hoop ik, voor de rest van mijn leven in denkbeeldige dankwoorden kan blijven schrijven. Enkelen ga ik toch met naam vermelden: Jeroen, Ine, Coryn, Cruysse, Lieven en Felix, Stijn en Joke, Driesl en Liesl, Kevin en Tom, Dieter, Tijs en Thijs, Otte, Theo, Bengan en Moritz, Irem, Kim, Nathalie, Elke en Tom, Heleen en Davy, en ... Bert.

Daarnaast zijn er nog mijn ouders, mijn familie en mijn allerliefste zus natuurlijk.

Gent, september 2011
Kristof Vandoorne

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

AR	Anti-reflection
AGC	Adaptive gain controllers
ANN	Artificial neural networks
ASE	Amplified spontaneous emission
ASR	Automatic speech recognition
AWG	Arrayed waveguide grating

C

CH	Carrier heating
CW	Continuous wave

E

ER	Error rate
ESN	Echo state network

F

FB	Feedback
FCA	Free-carrier absorption

FWM	Four-wave mixing

G

GVD	Group-velocity dispersion

H

HMM	Hidden Markov model
HWR	Half-wave rectifier

I

InP	Indium phosphide

L

LED	Light-emitting diode
LSM	Liquid state machine

M

ML	Machine learning
MMI	Multimode interferference
MZI	Mach-Zehnder interferometer

P

PCB	Printed circuit board

Q

QW	Quantum well

R

RC	Reservoir computing
RNN	Recurrent neural network

S

SHB	Spectral hole burning
SLED	Superluminescent light-emitting diode
SNR	Signal-to-noise ratio
SOA	Semiconductor optical amplifier
SOI	Silicon-on-Insulator
SPM	Self-phase modulation

T

TE	Transverse electric
TM	Transverse magnetic
TPA	Two-photon absorption
TW	Traveling wave

V

VOA	Variable optical attenuator

W

WDM	Wavelength-division multiplexing
WER	Word error rate
WG	Waveguide
WTA	Winner-take-all

X

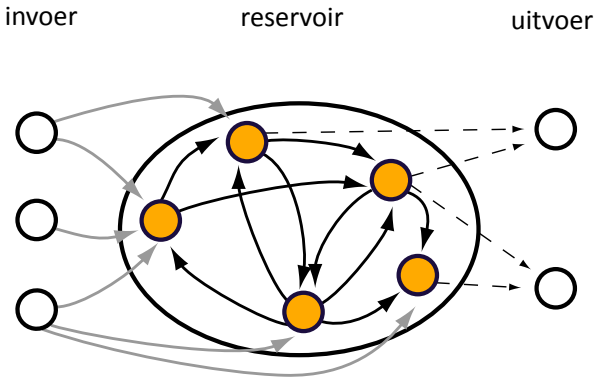
XGM	Cross-gain modulation
XPM	Cross-phase modulation

Nederlandse samenvatting –Summary in Dutch–

Sinds de industriële revolutie in de achttiende eeuw wordt het tijdperk waarin we leven vaak omschreven als het industriële tijdperk omdat onze economie en samenleving er zo sterk door veranderd en bepaald zijn. Recentelijk is men echter beginnen spreken over het informatietijdperk omdat informatie een steeds belangrijkere grondstof wordt in de economie. In de 20ste eeuw werd de verwerking van gegevens voornamelijk gedaan door computers die gebaseerd waren op ideeën van Turing en Von Neumann uit het midden van die eeuw. De drie basiskenmerken zijn dat oplossingen expliciet geprogrammeerd zijn (via zogenaamde algoritmes), dat de verwerking gebeurt op basis van vaste tijdsintervallen om zeker te zijn dat de overgangsverschijnselen uitgestorven zijn en een stabiele toestand bereikt is, en ten slotte dat de informatie wordt voorgesteld door discrete symbolen (nullen en enen voor digitale systemen). Deze toestellen zijn zo alomtegenwoordig dat het eenvoudig is om te vergeten dat er nog andere manieren zijn om gegevens te verwerken. Het beste voorbeeld van zo'n andere aanpak is het menselijk brein want mensen kunnen nog altijd veel beter omgaan met vele problemen die te maken hebben met classificatie en herkenning. Wij leren via voorbeelden en oefening, in tegenstelling tot een computer die algoritmes gebruikt die op voorhand opgesteld zijn. Het trainen van machines om bepaalde taken op te lossen is een domein op zich en *reservoir computing* is een verzamelnaam van het laatste decennium voor systemen die bestaan uit dynamische recurrente netwerken van simpele rekenknopen gecombineerd met een eenvoudige (meestal lineaire) uitleesfunctie (Figuur 1).

1 Reservoir computing

Bij reservoir computing wordt een netwerk dat op een willekeurige manier geconstrueerd is, het *reservoir*, geëxciteerd door een invoer. Dit reservoir wordt niet getraind, maar de respons van elke knoop wordt gecombineerd door middel van een simpele, vaak lineaire uitleesfunctie. Deze kan getraind worden met



Figuur 1: Schematische voorstelling van een reservoir computing systeem. De invoer wordt aan (alle of een deel van de) reservoir knopen aangeboden. De knopen zijn op een willekeurige manier met elkaar verbonden en met elke verbinding is een bepaald gewicht geassocieerd. De toestand van (alle of een deel van de) reservoir knopen wordt geleverd aan de uitleesfunctie (pijlen met streepjes) en enkel het gewicht van deze verbindingen wordt aangepast tijdens de training van het systeem.

eenvoudige standaardmethodes zodat de eigenlijke uitvoer zo dicht mogelijk aansluit bij de gewenste uitvoer. De moeilijkheden bij het trainen van recurrente netwerken worden zo vermeden, omdat enkel de gewichten van de uitleesfunctie aangepast worden. Reservoir computing combineert zo het geheugen en de spatio-temporele verwerkingskracht van dynamische systemen, met het gemak van lineaire regressie. Een manier om het succes van reservoir computing te verklaren is het reservoir te bekijken als een hoog-dimensionale spatio-temporele filtering van de invoer. Hierbij mengt het reservoir de verschillende invoersignalen op een niet-lineaire manier en projecteert deze naar een hoog-dimensionale toestandsruimte waar het eenvoudiger is om de essentiële kenmerken te onderscheiden. Reservoir computing is al met succes gebruikt voor verschillende complexe taken en presteert vaak even goed of zelfs beter dan de state-of-the-art. Een voorbeeld van een standaardtaak is een chaotische tijdreeks die moet voorspeld worden (de Mackey-Glass tijdreeks), waarbij reservoir computing een voorspellingsnauwkeurigheid bereikt die verschillende grootteordes beter is dan andere methodes.

Hoewel het reservoir zelf niet getraind wordt, is het dynamisch regime waarin het vertoeft zeer cruciaal voor de prestatie van het systeem. De beste prestaties worden meestal bekomen als het reservoir op de rand van stabiliteit zit, dus tussen stabiel en onstabiel gedrag in, omdat daar het geheugen van het systeem

maximaal is. het is daarom nodig om het dynamisch regime van het reservoir te kunnen afstellen. Het exacte optimum is vaak afhankelijk van de specifieke taak.

Reservoirs zijn inherent parallelle structuren, maar tot hiertoe zijn de meeste implementaties gebeurd in software. Dit zorgt voor lange simulatietijden bij grote reservoirs en lange invoer. Een implementatie die intrinsiek parallel is en waarbij alle knopen dus simultaan berekend worden, zou kunnen leiden tot een drastische vermindering van rekentijd en vermogenverbruik.

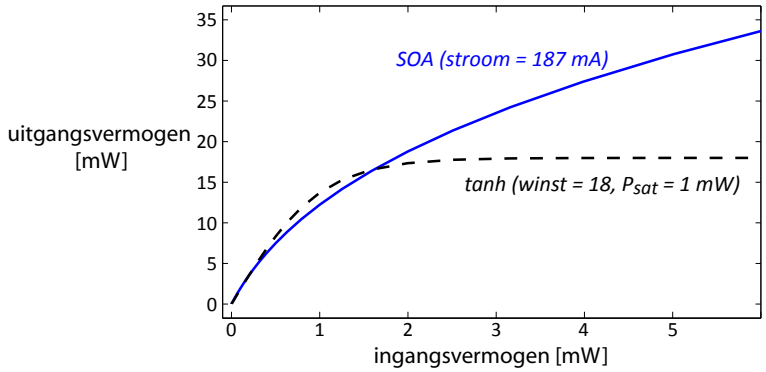
Traditionele methodes gebruiken een tangens hyperbolicus (tanh) (streepjeslijn in Figuur 2) als knoop waarbij het argument de gewogen som is van alle invoer van die knoop. Recent onderzoek heeft echter aangetoond dat een breed gamma aan niet-lineaire dynamische systemen gebruikt kan worden als reservoir.

2 Fotonica

Fotonica is de discipline die licht bestudeert. Net zoals in micro-elektronica, worden in geïntegreerde fotonica componenten geminiaturiseerd en gecombineerd op een chip. Het is een interessante technologie om reservoirs te maken, omdat er verschillende niet-lineaire mechanismen voorkomen die op verschillende tijdschalen werken. Het belooft ook om energiezuiniger te zijn dan elektronische implementaties, vooral voor signalen met een hoge bandbreedte.

Om een aantal redenen werden halfgeleider optische versterkers (SOAs) gekozen als knopen voor een eerste implementatie van fotonisch reservoir computing. Eerst en vooral compenseert de knoop zelf voor allerlei verliezen in het systeem. Ten tweede zijn ze breedbandig, wat de fabricatie makkelijker maakt in vergelijking met resonators waarbij de resonanties (gedeeltelijk) moeten overlappen. Ten derde lijkt hun statische invoer–uitvoer curve op die van de positieve tak van een tanh (Figuur 2), waardoor het eenvoudiger is om de bestaande kennis van reservoir computing toe te passen op fotonische reservoirs.

Hoewel een reservoir met SOAs nog niet de belofte inlost om heel snel en vermogenefficiënt te zijn, is het een interessant platform om de verschillen te bestuderen tussen fotonische en klassieke tanh reservoirs. Ten eerste kan licht bij een bepaalde golflengte voorgesteld worden door een complexe amplitude, in tegenstelling tot klassieke reservoirs die bijna altijd met reële signalen werken. Ten tweede hebben SOAs een interne dynamiek, terwijl tanh knopen puur statisch zijn. Ten derde zal op een vlakke chip de topologie van een geïntegreerd network van SOAs onderhevig zijn aan meer beperkingen dan een netwerk in software. Baantjes die elkaar kruisen dienen immers zoveel mogelijk vermeden te worden en daarom hebben we vaak gebruik gemaakt van een zogenaamde *draaikolk* topologie (Figuur 3). In deze topologie zijn knopen enkel verbonden met hun dichtste buren en het netwerk kan daardoor makkelijk vergroot worden met



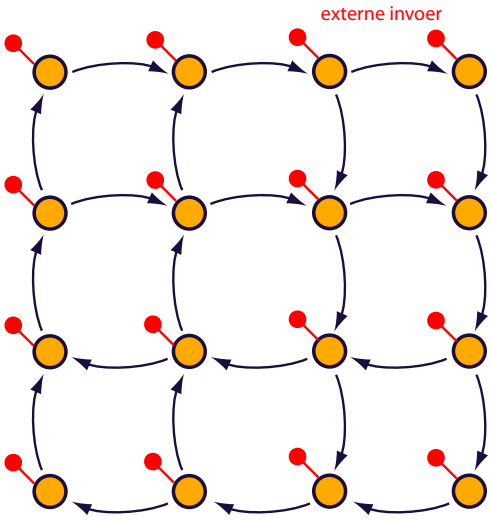
Figuur 2: Gemodelleerde curve van uitgangsvermogen versus ingangsvermogen voor een SOA die aangestuurd wordt met 187 mA en een tanh met een winst van een factor 18 en een saturatie vermogen van 1 mW.

behoud van een gelijke lengte voor alle verbindingen. Dit is vooral handig om de invloed van de verbindingen (bepaald door drie parameters: tijdsvertraging, verlies en fasedraaiing) op de prestatie van het systeem te bestuderen. In het volgende stuk bespreken we de prestatie van gesimuleerde SOA reservoirs.

3 Resultaten

De taak die we gebruiken om reservoirs te evalueren is een geïsoleerde cijfer herkenningstaak (de cijfers gaan van nul tot negen en ze worden door 5 vrouwen uitgesproken). Aan de audio voegen we gebabbel toe als achtergrondruis met een signaal–ruis verhouding van 3 dB om de taak moeilijker te maken. Na voorverwerking van de audio (standaard bij spraakherkenning) hebben we een invoer van 77 kanalen. Omdat de invoer zo hoog-dimensionaal is, hebben we voldoende knopen nodig in het reservoir en daarom gebruiken we een 9×9 draaikolk reservoir met 81 knopen. Per knoop worden de 77 invoerkanalen op een willekeurige manier gemengd en positief gemaakt door ze naar boven te verschuiven met het minimum over alle invoer. De respons van alle knopen wordt doorgegeven aan tien lineaire uitleesfuncties (een voor elk cijfer) en elk wordt getraind met een deel van de audio-voorbeelden en getest op een ander deel. De uitleesfunctie met de sterkste respons voor een bepaalde input wint en bepaalt dus welk cijfer er herkend wordt.

Het geheugen in het reservoir wordt mede bepaald door de typische tijdschalen van het reservoir zelf en daarom moet het reservoir aangepast worden aan de typische tijdschalen van de invoer of omgekeerd. Audio is traag in vergelijking

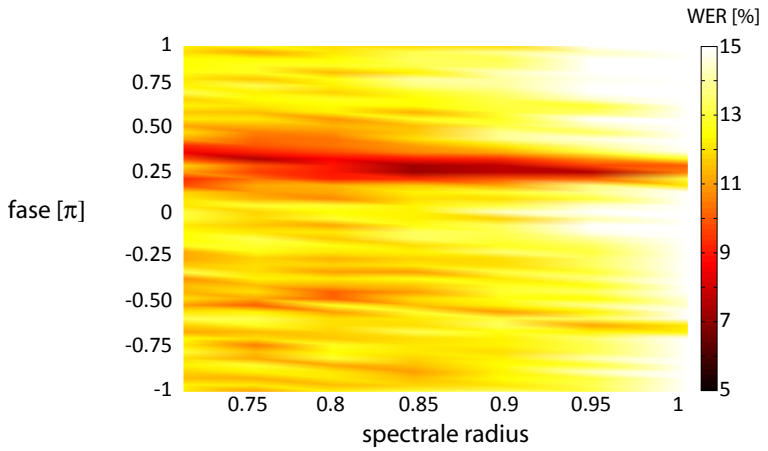


Figuur 3: 4×4 draaikolk topologie.

met de typische vertragingen op een chip en daarom hebben we de audio signalen versneld tot ze maar een paar honderd ps duren. De prestatie wordt gemeten als een fout, in dit geval de verhouding van voorbeelden die verkeerd herkend worden tot het totale aantal voorbeelden (WER).

Een resultaat voor een fotonisch reservoir met een vertraging van 6.25 ps in de verbindingen is te zien in Figuur 4. Hoe donkerder de kleur, hoe kleiner de fout en hoe beter de prestatie. De prestatie wordt onderzocht in functie van twee van de drie interconnectieparameters: verlies en fasedraaiing. De Y-as toont de gelijke fasedraaiing in al de verbindingen, terwijl de X-as de spectrale radius van het reservoir toont. De spectrale radius is een maatstaf voor de stabiliteit van een netwerk en wordt gedefinieerd als de grootste eigenwaarde van de Jacobiaan van het systeem bij het punt waar de winst maximaal is. Voor een waarde rond 1 komt dit overeen met de rand van stabiliteit en kan het beïnvloed worden door het verlies in de verbindingen te verhogen of te verlagen. Figuur 4 toont dat de optimale prestatie voor dit fotonisch reservoir zich situeert bij spectrale radii die enigszins kleiner zijn dan 1, maar ook dat de prestatie heel gevoelig is aan de fasedraaiing omdat enkel voor een nauwe regio voor de fase de fout klein is.

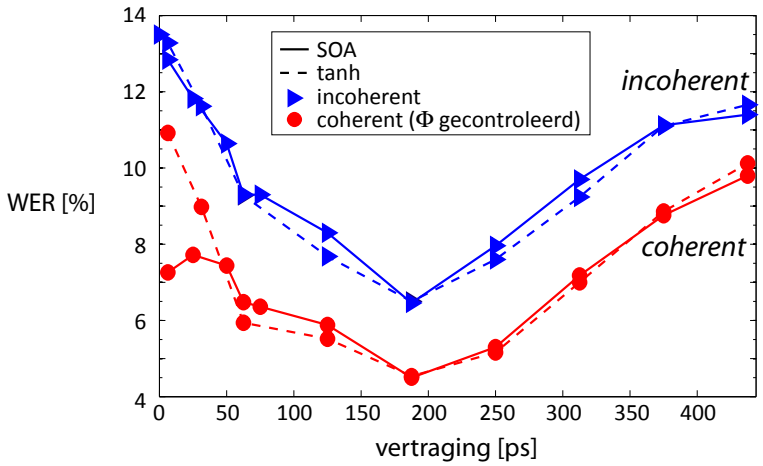
Als we de beste prestatie nemen van Figuur 4 over elke spectrale radius en fasedraaiing en er dus vanuit gaan dat we de fase perfect kunnen controleren, kunnen we die vergelijken met de optimale prestatie van klassieke tanh reservoirs. In Figuur 5 vergelijken we coherente en incoherente SOA reservoirs, alsook klassieke tanh reservoirs. Bij beide wordt de invloed onderzocht van de derde interconnectieparameter: de tijdsvertraging. Het eerste wat opvalt is dat de



Figuur 4: Fout (WER) voor coherente SOA reservoirs met een draaikolk topologie en een vertraging van 6.25 ps in de verbindingen. De spectrale radius wordt veranderd door het verlies in de verbindingen aan te passen. De fase is duidelijk de kritieke parameter, aangezien een goede prestatie enkel voorkomt in een smal gebied voor de fase.

optimale prestatie van al die reservoirs bij dezelfde tijdsvertraging ligt. Die komt overeen met ongeveer de helft van de duur van de audio signalen. De tijdsvertraging van verbindingen wordt meestal niet bestudeerd bij klassieke reservoirs. Zowel fasedraaiing als tijdsvertraging zijn belangrijke parameters en als beide gecontroleerd kunnen worden, is een substantiële prestatieverbetering mogelijk ten opzichte van klassieke reservoirs. Figuur 5 toont ook dat als we de fase niet kunnen controleren (incoherente SOA reservoirs), de prestatie nog altijd gelijkaardig is aan die van klassieke reservoirs.

Andere conclusies van ons werk zijn dat procesvariaties niet zo veel invloed hebben, maar dat het belangrijk is om de invloed van de ruis van SOAs (ASE) terug te dringen, aangezien deze zorgt voor een verminderde prestatie.



Figuur 5: Incoherente versus coherente tanh en SOA reservoirs voor verschillende waarden van de vertraging in de verbindingen. Voor de coherente reservoirs wordt het optimale resultaat gebruikt over fasedraaiingen en spectrale radius (fase gecontroleerd). Er is een duidelijk optimum bij alle reservoirs als ze een tijdsvertraging hebben van 187.5 ps. Coherente reservoirs presteren beter dan incoherente reservoirs.

English summary

The present age is often described as the information age, the natural successor to the industrial age that began in the 18th century, as information is becoming the key commodity that defines our lives and economy. The manipulation of data in the 20th century was predominantly done with computers, which are based on ideas formulated by Turing and Von Neumann in the middle of that century. The key characteristics are that solutions are explicitly programmed (algorithms), computation happens on a time-step basis to make sure that transients have died out in between two stable states and the information is represented using discrete symbols (zeros and ones for digital systems). These devices are so ubiquitous that it is easy to be unaware of the fact that there are other ways to do computation, mostly for tasks in the domain of classification and recognition, that are not easily solvable with traditional computing paradigms. The best example of alternative approaches can be found in nature, for example in the human brain, which excels in those domains. We compute by learning from examples and through practice, unlike a computer which uses predefined algorithms. Machine learning is the field of training systems to perform tasks and within this field, reservoir computing is an umbrella term from the last decade for a set of learning systems that consist of a dynamical recurrent network of simple computational nodes combined with a simple (usually linear) readout function (Figure 1).

1 Reservoir computing

In the reservoir computing framework, a randomly initialized network of nonlinear components, the *reservoir*, is excited by an input signal. The network itself is left untrained but the responses of all network nodes (*neurons*) to the input stimulus are combined through a simple, often linear, readout function. The readout can be optimized (*trained*) with simple and well-established methods such that its output matches the required output signal as closely as possible according to a certain error metric. In this way, as only the readout is changed, the difficulties of training recurrent networks are avoided. Hence, the reservoir computing approach combines the memory and spatio-temporal processing

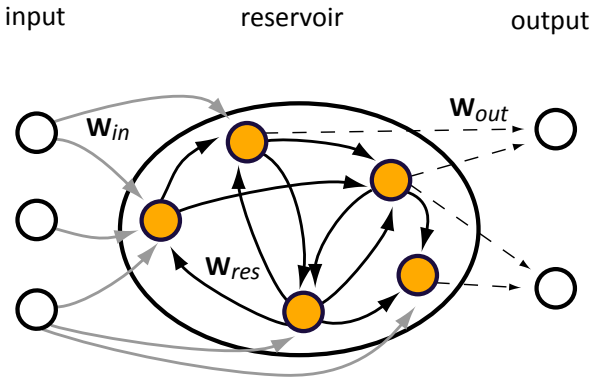


Figure 1: Schematic representation of a reservoir computing system. The inputs are fed into (all or a fraction of) the reservoir nodes, which are randomly interconnected with recurrent, weighed connections. The state of (all or a fraction of) the reservoir nodes is then fed into the outputs (dashed arrows). These dashed arrows are the only connections where the weight is changed during training.

of dynamical systems with the ease of training of linear regression. One way of explaining the success of reservoir computing is to view the reservoir as performing a high-dimensional spatio-temporal pre-processing or filtering of the input. In this view, the reservoir essentially performs a nonlinear mixing of the input signals into a high-dimensional feature space, so that the interesting features are more easily extracted by the readout. Reservoir computing has been demonstrated to give similar results as state-of-the-art techniques, or outperform them, for several complex machine learning tasks. An example is a benchmark task where a chaotic time series has to be predicted (the Mackey-Glass time series), and where reservoir computing reaches a prediction accuracy several orders of magnitude better than other methods.

Although the reservoir itself remains untrained, its performance depends critically on its dynamical regime. Optimal performance is usually obtained near the edge of stability, i.e., the region in between stable and unstable or chaotic behavior, because this regime optimizes the system's memory . This region is determined by the total amount of gain and loss in the network. Hence, to obtain good performance, we need to be able to tune a reservoir's dynamic regime to this edge-of-stability, using a small number of global parameters. However, this optimum is generally task-dependent.

Reservoirs are intrinsically parallel structures. However, most implementations thus far have been software based, leading to extensive simulation times

for large reservoirs and long input time series. A fully parallel implementation, where all node states are computed simultaneously, would drastically speed up computation times and might also lead to power savings. Traditionally, reservoirs have typically been made with a hyperbolic tangent (tanh) as node (dashed line in Figure 2), working in the weighed sum of its inputs, but recent work has indicated that a wide range of sufficiently high-dimensional nonlinear dynamical systems can be used as a reservoir.

2 Photonics

Photonics studies light in all its aspects. Integrated photonics, just like microelectronics, miniaturizes all the components and combines them on a single chip. It is an interesting candidate technology for building reservoirs, because it offers a range of different nonlinear interactions working on different time scales. It also offers the promise of being more power efficient than electronic implementations, especially for processing high-bandwidth signals.

Semiconductor Optical Amplifiers (SOAs) were chosen as a first testbed for an integrated version of photonic reservoir computing for a number of reasons. First of all, SOAs provide gain and therefore no separate component is needed to compensate for losses. Second, they are broadband, which relaxes fabrication tolerances in comparison to resonator-type devices, such as coupled cavities or ring resonators, where the resonances of different nodes usually have to (at least partially) overlap. Third, their steady state characteristic somewhat resembles the upper part of a tanh (Figure 2), making them a perfect bridge between the existing knowledge of reservoir computing with tanh recurrent networks and the new field of photonic reservoir computing.

Although an SOA reservoir does not yet fulfill the promised power efficiency or high speed, it presents a suitable platform for evaluating the differences between a photonic implementation and traditional software implementations. In particular, the topology and the behavior of both nodes and interconnections differ on several points from what is usually modeled in simulation. First of all, coherent light can be modeled with complex-valued amplitudes, unlike real-valued amplitudes used in classical reservoirs. Furthermore SOAs have internal dynamics unlike static tanh functions and since integrated photonic chips are planar, we want to avoid too many crossings and therefore we introduced the so-called *swirl* topology that was used in a lot of our simulations (Figure 3). It is a nearest neighbor topology and can be easily scaled up to larger reservoirs, while keeping the length of all connections equal, which makes it easier to investigate interconnection properties. The connections are modeled by three parameters: propagation delay, loss and phase change (since we are working with coherent

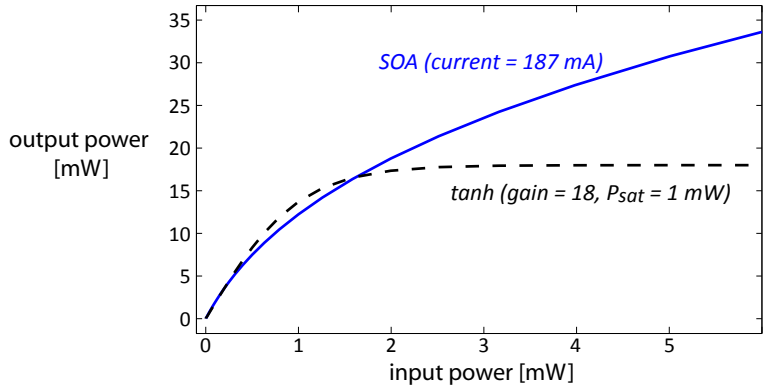


Figure 2: The modeled (power in – power out) curve for an SOA and a tanh with gain (18) and power saturation (1 mW).

light). In the next section we will discuss the performance of simulated SOA reservoirs and the impact of those parameters.

3 Results

The task we use to evaluate the performance of the reservoirs is an isolated digit recognition task (the digits 'zero' to 'nine', uttered by 5 female speakers) with added babble noise with a signal-to-noise radio of 3 dB to increase the complexity of the task. In our experiments the input consists of 77 channels, coming from the pre-processing of the speech data. With such high-dimensional input, the number of nodes needs to be sufficiently large. All the experiments were therefore done with a rectangular 9×9 swirl network of 81 nodes. Before feeding input to the network, the different channels were mixed together in a random manner for every node and made non-negative by shifting them upwards with the minimum over all the inputs. The output of all the nodes is given to ten linear classifiers (one for every digit). These classifiers are trained using a part of the dataset and tested on a different part. The classifier with the strongest response for a certain input 'wins' and determines which digit is recognized.

Reservoir memory is related to the typical time scales of the reservoir itself. Therefore, to achieve optimal memory in a reservoir, the relevant time scales of the input signals must be adapted to those of the physical reservoir implementation. Audio signals are very slow, so we accelerated the speech signal to accord with time scales typical for the delays in our SOA network. The new durations of the digits are in the order of a few hundred ps. The performance is measured

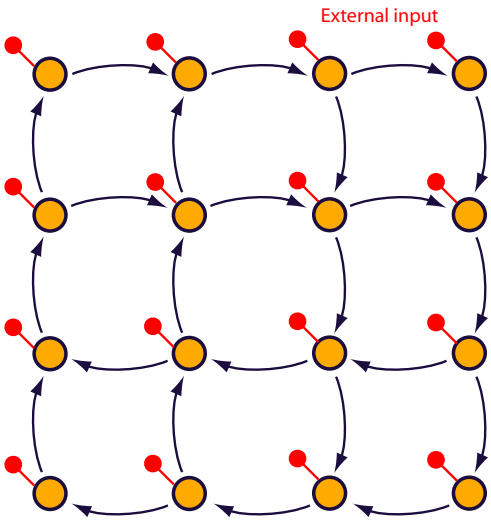


Figure 3: 4×4 swirl topology.

with the Word Error Rate (WER), which gives the ratio of incorrectly classified samples to the total number of samples.

A result for a photonic reservoir can be seen in Figure 4 with a connection delay of 6.25 ps. The darker the color, the better the performance. The performance is scanned over two of the three connection parameters: loss and phase change. The Y-axis scans the phase change in all the connections (since all connections are equally long, their phase change can be equal as well), while the X-axis represents the spectral radius of the network. The spectral radius is a measure of the stability of the network and is defined as the largest eigenvalue of the system's Jacobian at its maximal gain state. Values around unity correspond to the edge of stability previously mentioned. It can be influenced by the amount of loss in the connections. Figure 4 shows that for photonic reservoirs, the optimal performance can be found for a spectral radius slightly below unity. It also shows that the performance is very sensitive to the phase as good performance is only found for a narrow band in phase space.

If we take the optimal performance of Figure 4 over every phase change and spectral radius, thus assuming perfect phase control, we can compare it with the optimal performance of classical reservoirs. Figure 5 compares coherent and incoherent SOA reservoirs, and classical reservoirs scanned over the third interconnection parameter, its time delay. A first remarkable fact is that all the reservoirs have an optimal performance at the same optimal delay, roughly half of the duration of the audio signal. Interconnection delay is a parameter usually not considered in classical reservoirs. A second important parameter is the phase

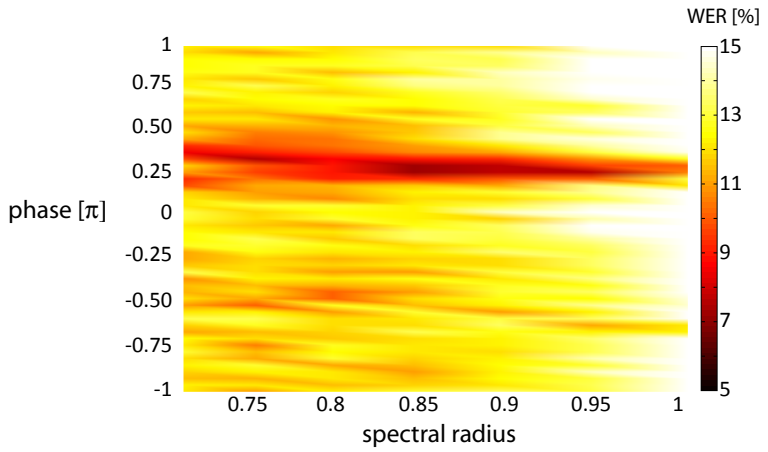


Figure 4: WER for swirl SOA reservoirs with coherent light and a delay of 6.25 ps where the spectral radius is changed through the attenuation in all the connections. Here, phase is a more critical parameter than spectral radius, as good performance is only found for a narrow band in phase space.

shift in the interconnections. If this can be controlled, a significant performance improvement is achieved and combined with the optimal delay it yields SOA reservoirs that outperform classical reservoirs. It also shows that if we cannot control the phase (incoherent SOAs), the performance is still similar to that of classical tanh reservoirs.

Other conclusions from our work are that process variations do not turn out to be very important, but careful considerations need to be taken into account to limit amplifier noise (ASE), since it degrades the performance significantly when left unmitigated.

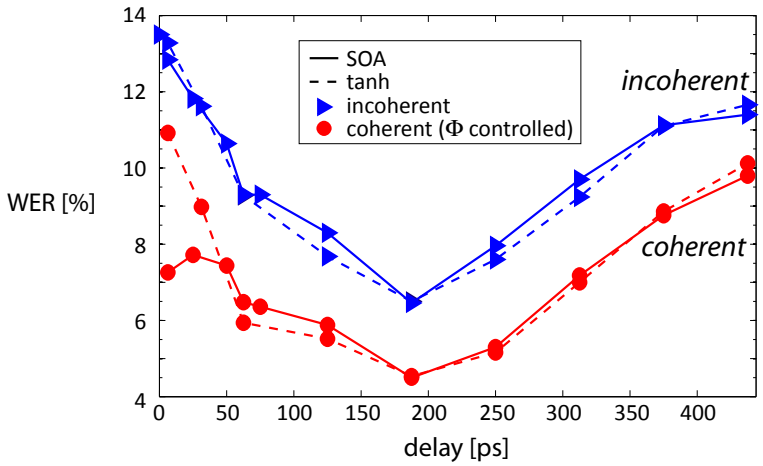


Figure 5: Incoherent versus coherent tanh and SOA networks for different values of the interconnection time delay. For the coherent networks the optimal value over spectral radius and phase was used (phase-controlled). There is a clear optimum for all networks at an interconnection delay of 187.5 ps, and it is also apparent that coherent reservoirs outperform incoherent reservoirs.

1 Introduction

The present age is often described as the information age, the natural successor to the industrial age that began in the 18th century, as information is becoming the key commodity that defines our lives and economy. A clear exponent of this is a company like Google whose profits come from the gathering of information from its users. Two key ingredients were required for this transition: the advent of computers and the internet that links these computers together in one world wide network.

This PhD thesis is at the crossing of two fields: those of artificial intelligence and photonics.

The science at the basis of computers is that of computation (or computer science) and the basic framework behind present day computing devices was formulated by Turing and Von Neumann in the middle of the 20th century. The key characteristics are that solutions are explicitly programmed (algorithms), computation happens on a time-step basis to make sure that transients have died out in between two stable states and the information is represented using discrete symbols (zeros and ones for digital systems). These devices are so ubiquitous that it is easy to forget that there are other ways to perform computation, mostly for tasks in the domain of classification and recognition, that are not easily solvable with traditional computing paradigms. The best example of alternative approaches can be found in nature, for example in the human brain, which excels in those domains. Artificial intelligence is a broad field where attempts are

made to mimic or emulate human-like computational power. It is from this field that the concept of Reservoir Computing (RC) has emerged in the last decade.

Photonics is as broad a field as artificial intelligence, with applications ranging from the harvesting of solar power with photovoltaics and imaging through cameras, LCDs and plasma screens, to the transportation of huge amounts of data through optical fibers. These optical fibers are at the core of the other ingredient of this information age, the internet, as carrying the vast amounts of digital data would be impossible without them. The processing of this optical data nowadays still happens with conversions to the electrical domain, where computers with electronic chips do the computation. It should come as no surprise that not only for reasons of elegance but more importantly for reasons involving speed and power consumption, people pursue doing this computation in the optical domain as well. Since all the information is in the digital domain, this research is labeled as digital photonics [1].

This PhD thesis investigates whether photonics could also be used for reservoir computing, a non-digital way of computation. We will start this introduction by briefly describing what reservoir computing is.

1.1 Reservoir computing

Nowadays no human would ever consider competing with a computer doing calculations, but when it comes to running on our two legs and understanding speech, for example, we still have the upper hand. We learn these things by training from examples and practice, unlike a computer which uses pre-defined methods, called algorithms. Machine learning is the field of training systems to perform a task and reservoir computing is an umbrella term for a set of learning systems that consist of a dynamical recurrent network of simple computational nodes combined with a simple (usually linear) readout function [2].

The recurrence in the network gives the system memory which is needed in many real-world applications such as speech recognition and time series prediction. The reservoir itself can be seen as a complex preprocessing filter for the linear readout, boosting its computational power. In this view, the reservoir essentially performs a nonlinear mixing of the input signals into a high-dimensional feature space, so that the interesting features are more easily extracted by the readout [3]. In practice, this works best when the reservoir itself is in a certain dynamical regime, often described as the *edge of stability*, i.e., the region in between stable and unstable or chaotic behavior, because this regime optimizes the system's computational capabilities [4]. This regime is dynamically rich, but still stable. Once instability and chaos come into play, the system becomes too sensitive to initial conditions and the information about the inputs will be destroyed by the wild dynamics.

Although only a decade old, the concept has attracted much attention [5–7] because of its simplicity, since the reservoir only has to be tuned to the right dynamical regime and the training of the linear readout happens with simple and well-established methods. A second reason is that reservoir computing gives similar results as state-of-the-art techniques, or outperforms them, for several complex machine learning tasks. An example is a benchmark task where a chaotic time series has to be predicted (the Mackey-Glass chaotic time series), and where RC reaches a prediction accuracy several orders of magnitude better than classical methods [5].

Reservoirs are intrinsically parallel structures. However, most implementations thus far have been software based, leading to extensive simulation times for large reservoirs and long input time series. A fully parallel implementation, where all neuron states are computed simultaneously, would drastically speed up computation times and might also lead to power savings. Recent work has indicated that a wide range of sufficiently high-dimensional nonlinear dynamical systems can be used as a reservoir [8–12]. In the next section we will see why photonics is an interesting hardware platform for RC.

1.2 Photonics

Photonics studies light in all its aspects. Light is a part of the electromagnetic spectrum ranging from the ultraviolet to the infrared. Humans can only see a fraction of this spectrum and this part is therefore not surprisingly called visible light. The colors we see are nothing more than wavelength bands. An example of people's fascination with light throughout history is Newton's famous prism experiment where he showed that light is separated in its color spectrum when passing through a prism. However, it was only with the invention in 1960 of the laser source, which can generate an intense narrow beam of light of a single wavelength, that the field of photonics really emerged beyond lighting applications [13]. At the time described as a "solution looking for a problem" it has led to applications such as data transmission over optical fibers, eye surgery and sensing of all kinds of molecules.

The dominating technology for computation in the 20th century has been electronics and especially micro-electronics. Already in 1959, Richard Feynman gave a visionary talk in which he coined the famous phrase "There is plenty of room at the bottom", indicating that by scaling devices to smaller sizes, the functional density could be increased[1]. This has manifested itself by the observation (called Moore's law [14]) that the amount of transistors on an integrated chip

[1] A transcript of the talk is available at http://www.its.caltech.edu/~feynman/plenty.html

doubles roughly every year[2]. Computers have billions of transistors on a single chip nowadays, without an accompanying price increase. This process has made ever more powerful computers available for ever more people and led to the dominance of digital computation.

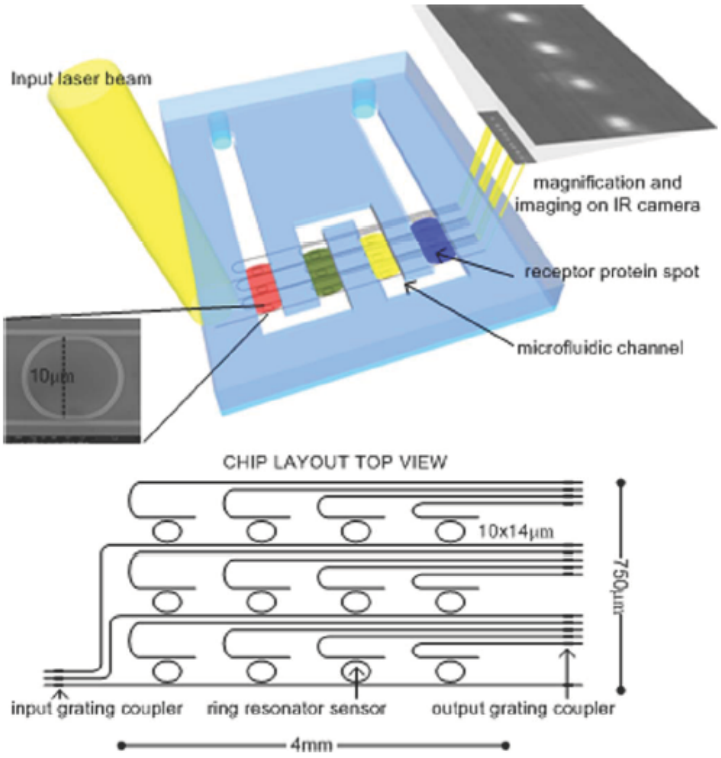
Generating and manipulating data is one thing, moving it from one place to another is something different. The loss in coaxial cables depends on the frequency (often expressed in bitrates[3]), which limits how fast one can send information through those wires. Since they are too lossy at high frequencies to send data over long distances, photonics took over as it offers a higher bandwidth at lower losses [16]. Photons are massless particles and in vacuum they travel according to the laws of physics at the vacuum speed of light, almost 300 million m/s. In free-space the beam of a laser spreads out due to diffraction [17], but with the advent of optical fibers it became possible to guide the beam inside such a fiber [18], when the refractive index of the core is higher than that of the surrounding cladding. It took a while to bring the losses down from 20 dB/km to 0.2 dB/km of today's fibers. Nowadays, the backbone of the internet, submarine and over land, allows bitrates up to 69 Tbit/s over a single fiber [19].

The distances at which photonic communication links have an advantage over electric connections have steadily decreased over the years as the demand for bandwidth has increased. Nowadays, there is a strong focus on doing even the communication between multi-core chips in a computer with optical interconnects [20]. Since all the information is already in the optical domain, there has been a lot of research to do all this digital computation optically instead of hopping back and forth between the electronic and optical domain [1]. However, photons are not electrons and up till now people have found it hard to put photonics in the existing electronics–transistor framework, which is till now the dominant implementation of the Turing model, simultaneously satisfying the ever more stringent power consumption limitations [21].

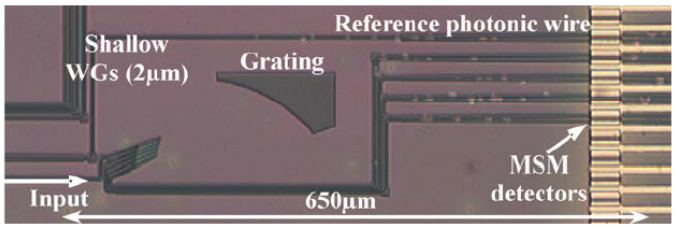
It is not possible to realize optical interconnects on chip without integration and miniaturization as has happened with electronics. While transistors do get smaller all the time, photonic components are not easily scaled down as their size is determined by the index contrast between the core and the cladding in waveguides. Optical fibers for example have a very small index contrast of only a few percent, which means that the bending radius should be quite large, or the light is not guided anymore. Silicon-on-insulator on the other hand is a material system with a very high index contrast. Here, a silicon layer with an index of 3.45 in the near infrared, rests on a silicon oxide layer with an index of 1.45. This contrast allows for bend radii down to a few μm without significant additional loss [22].

[2] In 1975 Moore altered his prediction to doubling every two years [15].

[3] Frequency here is about how fast switching between different states can happen. In a digital world this means between what represents '1' and what '0', hence bitrates.



(a) integrated sensor



(b) integrated receiver

Figure 1.1: Two examples of integrated photonics. (a) shows an integrated sensor where different ring filters are coated with materials sensitive to different bio-materials. (b) shows a top view of an integrated receiver combining normal waveguides, detectors and a wavelength selective device to make an optical spectrum analyzer. The typical dimensions are shown.

Silicon is also the dominant material in the micro-electronics semiconductor industry, which allows using well-known CMOS equipment and development. Silicon photonics is the core focus of the Photonics Research Group in Gent and research is ongoing to integrate lasers, modulators and detectors onto a single chip for applications such as sensing [23], telecom [24, 25] and optical signal processing (Figure 1.1).

Reservoir computing harnesses the computational power of dynamical recurrent networks, and since (integrated) photonics is a hardware platform capable of handling huge amounts of data with low losses at high-speed, it is interesting to investigate whether it would be useful for building reservoirs. It also offers a range of different nonlinear interactions working on different time scales and the promise of being more power efficient than electronic implementations, especially for processing high-bandwidth signals.

1.3 Thesis outline

The goal of this dissertation is to investigate whether integrated photonics is a useful hardware platform for reservoir computing. The second chapter gives an introduction to reservoir computing, its history and applications. The third chapter discusses the relevant photonics, focusing on Semiconductor Optical Amplifiers (SOA), which we used as nodes in our photonic reservoir.

On the simulation side we mostly used two different tasks: an easy signal classification task and a much harder speech recognition task. Since the methodology used for both differs, we have opted to separate the results for both tasks in two chapters. Chapter 4 describes the results on the signal classification task and it gives an indication about the parameters that matter the most for photonic reservoirs: coherence and interconnection delay. Chapter 5 uses an isolated digit speech recognition task to systematically compare classical hyperbolic tangent (tanh) and photonic SOA reservoirs to quantitatively investigate which differences matter and to what degree. As such it confirms most of the results from Chapter 4 but investigates them in much greater detail and therefore the most important conclusions from this thesis can be found in this chapter.

Chapter 6 discusses the different approaches we followed to verify our simulation results on a small-scale hardware implementation. We have not been able to perform measurements on a hardware chip yet, but while writing this thesis, fabrication of a chip has been finished and hence measuring this chip is work for future research.

1.4 Publications

The results obtained within this work have been published in two peer reviewed journals and were presented at various national and international conferences. The following list gives an overview.

Publications in international journals

1. K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman. *Parallel reservoir computing using optical amplifiers.* IEEE Transactions on Neural Networks, 22(9):1469–1481, September 2011.

2. K. Vandoorne, W. Dierckx, B. Schrauwen, D. Verstraeten, R. Baets, P. Bienstman, and J. Van Campenhout. *Toward optical signal processing using Photonic Reservoir Computing.* Optics Express, 16(15):11182–11192, July 2008.

Publications in international conferences

3. K. Vandoorne, M. Fiers, T. Van Vaerenbergh, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. *Advances in photonic reservoir computing on an integrated platform.* In International Conference on Transparent Optical Networks (ICTON), **invited**, page Mo.B5.5, Stockholm, Sweden, June 2011.

4. K. Vandoorne, M. Fiers, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. *Optical signal processing with a network of semiconductor optical amplifiers in the context of photonic reservoir computing.* In SPIE Photonics West - OPTO, page 79420P, San Francisco, CA, USA, January 2011.

5. K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman. *Delays in photonic reservoir computing with Semiconductor Optical Amplifiers.* In IEEE Photonics Benelux Chapter Annual Symposium, pages 25–28, Delft, Netherlands, November 2010.

6. K. Vandoorne, M. Fiers, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. *Photonic Reservoir Computing: a New Approach to Optical Information Processing.* In Workshop on Cognitive and neural models for automated processing of speech and text (CONAS), Gent, Belgium, July 2010.

7. K. Vandoorne, M. Fiers, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. *Photonic Reservoir Computing: a New Approach to Optical*

Information Processing. In International Conference on Transparent Optical Networks (ICTON), **invited**, page Th.A4.3, Munich, Germany, June 2010.

8. P. Bienstman, K. Vandoorne, M. Fiers, D. Verstraeten, J. Dambre, and B. Schrauwen. *Photonic reservoir computing as a new paradigm for optical information processing*. In Photonics North, **invited**, page 196 (paper PDS-2-1-4), Niagara Falls, Canada, June 2010.

9. G. Roelkens, O. Raz, W. green, S. Assefa, M. Tassaert, S. Keyvaninia, K. Vandoorne, D. Van Thourhout, R. Baets, and Y. Vlasov. *Towards a low-power optically-pumped nanophotonic semiconductor amplifier heterogeneously integrated with SOI waveguides*. In 7th IEEE International Conference on Group IV Photonics (GFP), pages 16–18 (paper WB4), Beijing, China, September 2010.

10. K. Vandoorne, W. Dierckx, B. Schrauwen, D. Verstraeten, P. Bienstman, R. Baets, and J. Van Campenhout. *Photonic reservoir computing with coupled semiconductro optical amplifiers*. In First International Workshop on Optical Supercomputing (OSC), LNCS 5172, pages 46–55 Vienna, Austria, August 2008.

11. K. Vandoorne, P. Bienstman, and R. Baets. *Photonic Reservoir Computing: interconnected Semiconductor Optical Amplifiers*. In ePIXnet Spring School - Technology for Photonics Integration, page 14, Portoferraio, Elba Island, Italy, May 2008.

12. K. Vandoorne and P. Bienstman. *A photonic implementation of reservoir computing*. In 12th Annual Symposium of the IEEE/LEOS Benelux Chapter, pages 195–198, Brussels, Belgium, December 2007.

13. K. Vandoorne and P. Bienstman. *A photonic implementation of reservoir computing*. In Workshop of the IEEE/LEOS Benelux Chapter, pages 25–26, Eindhoven, Netherlands, May 2007.

14. K. Vandoorne, P. Bienstman, and R. Baets. *A photonic implementation of reservoir computing*. In ePIXnet Winter School - Applications of Photonic Integration, page 65, Pontresina, Switzerland, March 2007.

Publications in national conferences

15. K. Vandoorne and P. Bienstman. *Photonic reservoir computing with SOAs and delays*. FirW PhD Symposium, page 139, Gent, Belgium, December 2010.

References

[1] K. Huybrechts. *Digital Photonics Using Single Laser Diodes for All-Optical Network Nodes.* PhD thesis, Ghent University, 2010.

[2] D. Verstraeten. *Reservoir Computing : computation with dynamical systems.* PhD thesis, Ghent University, 2009.

[3] T. M. Cover. *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition.* IEEE Transactions On Electronic Computers, EC-14(3):326–334, 1965.

[4] R. Legenstein and W. Maass. *What makes a dynamical system computationally powerful?* In S. Haykin, J. C. Principe, T.J. Sejnowski, and J.G. McWhirter, editors, New Directions in Statistical Signal Processing: From Systems to Brains, pages 127–154. MIT Press, 2007.

[5] H. Jaeger and H. Haas. *Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication.* Science, 304(5667):78–80, April 2004.

[6] W. Maass, T. Natschläger, and H. Markram. *Real-time computing without stable states: A new framework for neural computation based on perturbations.* Neural Computation, 14(11):2531–2560, 2002.

[7] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. *An experimental unification of reservoir computing methods.* Neural Networks, 20(3):391–403, April 2007.

[8] D. Verstraeten, S. Xavier-de-Souza, B. Schrauwen, J. A. K. Suykens, D. Stroobandt, and J. Vandewalle. *Pattern classification with CNNs as reservoirs.* In Proceedings of the International Symposium on Nonlinear Theory and its Applications (NOLTA), pages 101–104, Budapest, Hungary, September 2008.

[9] K. P. Dockendorf, I. Park, P. He, J. C. Principe, and T. B. DeMarse. *Liquid state machines and cultured cortical networks: The separation property.* Biosystems, 95(2):90–97, 2009.

[10] C. Fernando and S. Sojakka. *Pattern recognition in a bucket.* In Proceedings of Advances in Artificial Life, volume 2801, pages 588–597, Dortmund, Germany, September 2003.

[11] W. J. Goh and N. Crook. *Pattern recognition using chaotic transients.* In M. Verleysen, editor, Proceedings of the 15th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, April 2007.

[12] B. Jones, D. Stekel, J. Rowe, and C. Fernando. *Is there a Liquid State Machine in the Bacterium Escherichia Coli?* In Proceedings of the IEEE Symposium on Artificial Life (ALIFE), pages 187–191, 2007.

[13] T. H. Maiman. *Stimulated optical radiation in Ruby.* Nature, 187(4736):493–494, 1960.

[14] G. E. Moore. *Cramming More Components Onto Integrated Circuits.* Electronics, 38(8), April 1965.

[15] G. E. Moore. *Progress In Digital Integrated Electronics.* In Proceedings of the International Electron Devices Meeting, pages 11–13, Washington DC, USA, 1975.

[16] G. P. Agrawal. *Fiber-Optic Communication Systems.* John Wiley & Sons, Inc., New York, NY, USA, 2002.

[17] B. E. A. Saleh and M. V. Teich. *Fundamentals of Photonics.* John Wiley & Sons, Inc., New York, NY, USA, 1991.

[18] K. C. Kao and G. A. Hockham. *Dielectric-fibre surface waveguides for optical frequencies.* Proceedings of the Insitution of Electrical Engineers, 113(7):1151–1158, 1966.

[19] A. Sano, H. Masuda, T. Kobayashi, M. Fujiwara, K. Horikoshi, E. Yoshida, Y. Miyamoto, M. Matsui, M. Mizoguchi, H. Yamazaki, Y. Sakamaki, and H. Ishii. *69.1-Tb/s (432 x 171-Gb/s) C- and Extended L-Band Transmission over 240 km Using PDM-16-QAM Modulation and Digital Coherent Detection.* In Conference on Optical Fiber Communication (OFC), page PDPB7, 2010.

[20] D. A. B. Miller. *Device Requirements for Optical Interconnects to Silicon Chips.* Proceedings of the IEEE, 97(7):1166–1185, July 2009.

[21] D. A. B. Miller. *Are optical transistors the logical next step?* Nature Photonics, 4(1):3–5, January 2010.

[22] W. Bogaerts, S. K. Selvaraja, P. Dumon, J. Brouckaert, K. De Vos, D. Van Thourhout, and R. Baets. *Silicon-on-Insulator Spectral Filters Fabricated With CMOS Technology.* IEEE Journal of Selected Topics in Quantum Electronics, 16(1):33–44, 2010.

[23] K. De Vos, J. Girones, T. Claes, Y. De Koninck, S. Popelka, E. Schacht, R. Baets, and P. Bienstman. *Multiplexed Antibody Detection With an Array of Silicon-on-Insulator Microring Resonators.* IEEE Photonics Journal, 1(4):225–235, October 2009.

[24] J. Brouckaert, G. Roelkens, S. K. Selvaraja, W. Bogaerts, P. Dumon, S. Verstuyft, D. Van Thourhout, and R. Baets. *Silicon-on-Insulator CWDM Power Monitor/Receiver With Integrated Thin-Film InGaAs Photodetectors.* IEEE Photonics Technology Letters, 21(19):1423–1425, October 2009.

[25] L. Chen, K. Preston, S. Manipatruni, and M. Lipson. *Integrated GHz silicon photonic interconnect with micrometer-scale modulators and detectors.* Optics Express, 17(17):15248–15256, August 2009.

2
Reservoir Computing

Reservoir computing is a decade-old framework for computation, in which the computational capabilities of nonlinear dynamical systems are harnessed. It is different from the nowadays dominant computing paradigm with the von Neumann architecture as its most important exponent. The objective of this dissertation was to evaluate if photonic reservoirs would be a useful hardware platform for reservoir computing and vice versa. The first part of this chapter will give a very short introduction to computing. The second part portrays a short historic overview of how this new paradigm of RC saw the light of day. The third part describes a type of classical reservoir implementation (echo state networks), which we will use as a baseline to evaluate photonic reservoirs. A last part discusses how reservoirs are used and trained. A very good introduction to reservoir computing from a more computer science point-of-view can be found in the PhD thesis of David Verstraeten [1].

2.1 Introduction

Computers and similar electronic devices are ubiquitous in our society and their computation is based on algorithms. The father of this approach, Alan Turing, described an abstract device that forms the basis of present day computing devices. There are three defining features about this approach: first, the behavior of the computer is explicitly programmed, second, the mode of operation is timestep-

based and third, the symbols read and written are discrete and not continuous. The popularity of this approach by no means implies that it is the only way to do computation. The central idea of the RC paradigm is that computation is also possible with complex dynamical systems that are not explicitly programmed [1].

Intuitively, it is quite easy to understand how such a different kind of computation would work, since humans themselves are a prime example. Humans are often described as being intelligent, partly because we can learn from experiences. Although it is hard to give a rigorous definition of this intelligence, it is clear that computers do not have it. Basically a computer is a very powerful calculator that does exactly what it is programmed to do, but nothing more. For example, if a user performs a certain task repeatedly, a computer will usually not pick up this pattern and suggest the user to take over this task. Another aspect is that humans cope seemingly effortlessly with problems such as speech and handwriting recognition, image classification, language processing, moving in a new environment (ignoring for a moment that it takes a baby years of exposure and training to reach a sufficient level of language processing). The point remains that explicitly programmed solutions, even though they have been researched for years, are unable to reach the performance level of humans on these types of tasks. This paragraph also touches upon two fundamental properties of this kind of computation: *learning* or *training* and *examples*. Humans attain language understanding by hearing lots of example sentences, distilling patterns from it, generalizing those patterns, until they can make previously unseen patterns themselves, that just 'feel' right.

The tool that is responsible for this ability is our brain, so it must not come as a surprise that whole fields such as machine learning and artificial intelligence have emerged trying to capture similar computational powers from systems inspired by our brain. Machine learning methods share the common property that they are not programmed in a conventional sense, but that they learn (by generalizing) from examples.

One way to classify Machine Learning (ML) methods is to discern between different methods of learning: *unsupervised*, *reinforced* and *supervised*. Unsupervised learning methods try to find regularities in the data without any user input about the data or expected properties and they are especially useful for data clustering applications [2, 3]. On the other hand, when the system is evaluated during reinforcement learning, an indication is given of how well it is performing but not the exact desired behavior. This system of rewards and/or penalties is popular in the robotics community where it is easier to say how well the robot is doing, as opposed to what it should be doing exactly. Finally, in supervised learning, the desired output for every input is known and used during training. While this information is not always available in real-world problems, it greatly enhances the learning process and accuracy. In this work we use reservoir com-

puting as a supervised learning method and the next section describes its origins that lie within models usually called Artificial Neural Networks (ANN) that are loosely inspired by the brain and hence somewhat biologically plausible.

2.2 History

Although it is not the intention here to give a full overview of the whole field of ML and ANNs, it is interesting to see how RC came as an answer to the problem of how to easily use recurrent neural networks. But to answer that question we will first discuss what recurrent neural networks are.

2.2.1 Artificial neural networks

Artificial neural networks were developed as models of our brain structure and they range from very simple to hugely complex in an attempt to make them more realistic. Some study them to obtain a better understanding of the human brain, others for engineering purposes to solve complex tasks. This work belongs to the second category. In essence, ANNs consist of input/output processing nodes, called neurons, that are connected into a network with weighted connections (Figure 2.1). The input–output behavior of a neuron is shown in the inset of Figure 2.1 and the output of a neuron with index j can be expressed as:

$$y_j = f\left(\sum_{i \in S_j} w_{ij} y_i\right), \tag{2.1}$$

where w_{ij} is the weight of the connection between neuron i and neuron j, y_i is the output (often called *activation*) of neuron i and S_j is the set of neurons connected to neuron j's input. f is a transfer or activation function and usually nonlinear.

There are some fundamental properties that allow us to group ANNs together. A first property concerns the nature of communication between the neurons which can either be analog or spiking. When it is analog, the inputs are mixed together with Equation 2.1. Some of the commonly used functions are a linear function, a piecewise linear function, a threshold function or some kind of sigmoid function. All these functions (apart from the linear one) have a so-called squashing effect, which means that the output will remain bounded, no matter what the input is.

Spiking neurons are a class of biologically more realistic neuron models [4, 5] with weighted connections just as in analog neural networks. However, they communicate through isolated spikes instead of continuous values, just like biological neurons. Since the spikes can be considered identical, the information they convey is entirely encoded in the precise timing. They have some internal

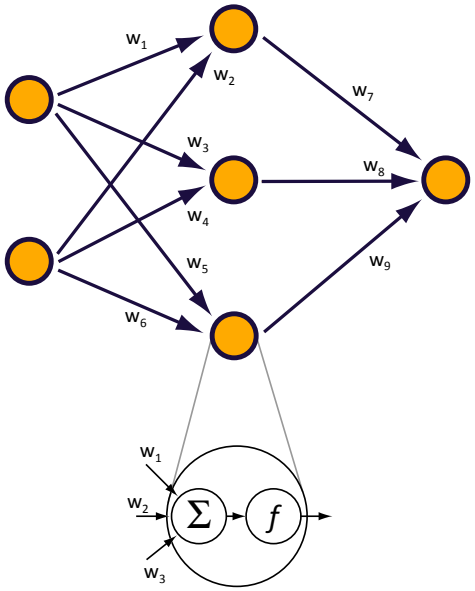


Figure 2.1: This shows an example of an artificial neural network where nodes are connected and each connection has a certain weight w_i. The close-up of one node reveals its interior working: the different inputs, weighted with their corresponding weight w_i, are summed together and then fed through a certain function f. This kind of ANN is called a feedforward neural network, since there are no feedbacks. The inputs are fed into the network from the left, information flows through the network to the right, where the output is collected. Usually those networks are organized into layers as shown here.

memory and it was shown theoretically that they are capable of performing more complex operations than sigmoidal analog neurons [6].

2.2.2 Recurrent neural networks

The topology of a neural network is determined by its weight matrix **W** where an element $\mathbf{W}[i, j] = w_{ij}$ is the weight of a connection between nodes i and j. When there is no connection, w_{ij} is zero. The absence or presence of feedback connections in the topology of ANNs has a very big impact on their training and usefulness. Two main categories can be distinguished and they are called respectively *feedforward neural networks* and *Recurrent Neural Networks* (RNN).

Feedforward neural networks have their topology usually arranged in layers with information flowing from the input layer, through one or more *hidden* layers (so-called because their state is usually not observed), until an output layer is reached (Figure 2.1). This type of network is also known by the name of multi-layer perceptron. The values of the hidden and output neurons are completely determined by the input values and hence these networks are a mapping from input to output [7]. Many learning rules exist and the most famous one is a combination of error-backpropagation and stochastic gradient descent where the gradient of the error at the output is calculated in function of the adjustable weights and then fed backwards, adapting the weights of the network to minimize the error [8].

A disadvantage is the lack of memory due to the layered architecture, which renders feedforward neural networks rather unsuitable for processing temporal information such as speech or robot control where the order in which the information is presented is important. One approach to circumvent this is using a tapped delay line, and feeding the network not just with the present sample, but also with previous samples (depending on the length of the delay line). This approach is called a time-delay neural network and was first introduced in [9] and translates a temporal problem into a spatial one. Disadvantages are the finite and artificial length of the delay and the need for many parameters for long delays.

A different approach for handling temporal problems, is allowing feedback connections in the network, which results in so-called recurrent neural networks. In this case connections are also characterized by a delay, which introduces a form of memory into the network, because information keeps on circulating. It also turns the network into a dynamic system since the output of the system no longer depends on just the current inputs, but also on the previous state of the network (and hence recursively on all past inputs). This makes them ideal for inherently temporal problems such as speech recognition and machine control. They also have the advantages of feedforward networks which include

noise robustness, learning by example and the ability to model highly nonlinear systems. They have been applied successfully in the learning of context-free and context-sensitive languages [10, 11], control and modeling of complex dynamical systems [12], speech recognition [13, 14] and handwriting recognition [15].

Although RNNs are very useful, their large-scale use in practical applications has been hampered by the fact that many learning rules suffer from problems such as slow convergence rates, many local optima, bifurcations, and high computational costs [16, 17]. The origin of these problems lies with the dynamical nature of the system, which makes that the training process has to do two highly complex tasks simultaneously: optimizing the network parameters so that the RNN operates in the correct dynamical regime and also obtain the desired input–output behavior. The essence of reservoir computing is precisely decoupling these two tasks. Tuning the system to its optimal dynamical regime and obtaining the best output will be done separately, by splitting up the network into a reservoir and a readout.

2.2.3 Founding fathers of reservoir computing

Success has many fathers or maybe in science it is more appropriate to say that when the time and conditions are ripe for a certain idea, it is often discovered by more than one person at the same time. Calculus is a famous example and it is now generally accepted that important breakthroughs were achieved by both Isaac Newton and Gottfried Wilhelm Leibniz independently. However, in those days accusations of plagiarism flew over the English Channel and it divided continental and English-speaking mathematicians for quite a while afterwards. Put in that perspective, the 'discovery' of reservoir computing was surprisingly uneventful, but not less interesting. Usually two founding fathers are credited: Herbert Jaeger and Wolfgang Maass. Maass wrote a paper in 2002 titled "Real-time computing without stable states: A new framework for neural computation based on perturbations" [18] in which he introduced the concept of Liquid-State Machines (LSMs), while Jaeger introduced Echo State Networks (ESNs) in a report published in 2001 [19]. Their appeal and reason why they attracted so much research interest lies in their ease of use and excellent performance on many tasks.

Jaeger, coming from an engineering background, circumvented the training problems for RNNs, described in the previous section, by simply not training the network at all. First, a RNN (usually with tanh nodes) is created with random topology and random weights and it is globally scaled to a desirable regime. Desirable means that the network should have the echo state property, which means — informally — that the network should asymptotically forget its initial state when driven by an external signal. The recurrent connections allow for

information from past inputs to keep circulating and the echo state property means that the network should eventually forget about them, which is often described as *fading memory*. The mixing of current inputs and fading past states of the network are called *echoes* in this framework. The second step is using the response of the network to train a simple linear classifier, which can often be done with one-shot methods. Section 2.4 will discuss this in detail. The dynamics of the network should be rich enough that the mixing of the inputs boosts the computational power of the linear readout, but not so rich that it does not forget its past inputs eventually. Therefore it is often called computation at the *edge of stability*. There exist some measures to mark the optimal dynamical regime and often they work well as a rule of thumb to make good performing networks [20], but there can be considerable variation between different random reservoirs in the same regime and it is a topic of ongoing research to find the sufficient conditions and constraints for a good ESN [21].

The second founding father, Maass, coming from a background in neuroscience, proposed liquid state machines as an abstract computational framework consisting of two parts. The first part is a high-dimensional filter or mapping that maps the current and past inputs onto a state vector, the second part maps this state using a (usually memoryless) linear readout function onto an output vector. The filter should have the point-wise separation property, which says that different input vectors should be mapped to different state vectors, while the readout should have the universal approximation property, meaning it should be able to approximate any function on a closed and bounded domain with arbitrary precision. In the first seminal paper from W. Maass [18], it was shown that a LSM satisfying these two properties can be trained to approximate any stationary mapping on time-varying inputs with fading memory. In practice the filter is usually a RNN with spiking neurons, combined with a linear readout. Although it was conceived to obtain a deeper insight in the workings of the brain, it has been popular as well for engineering purposes [22, 23].

Two years later J. J. Steil proposed a new learning rule for RNNs called Backpropagation-Decorrelation, where the learning rule itself is only applied to the output layer, while the internal weights are chosen with a suitable initial global scaling [24]. Similar ideas were even lingering in the decade before, but then for some reason it did not spark interest in the research community [25, 26]. For a more thorough and complete overview of the field, we refer to [27–30].

2.2.4 Generic reservoir computing

Notwithstanding the differences between the previously discussed approaches (e.g., tanh vs spiking neurons,...), the similarities led people to coin the new term reservoir computing [31]. It is a simple and powerful framework for harnessing

and using the computational capabilities of nonlinear dynamical systems, called reservoirs, combined with a simple readout. The fundamental idea is that the reservoir does a suitable nonlinear mapping with a fading memory of the input signals into a higher dimensional space[1], which enables the use of relatively simple but computationally undemanding linear classification or regression algorithms [1]. Because of this, its applicability exceeds the world of neural networks and neuroscience it was was born in. Complex nonlinear filters are used in many research fields, and RC provides an elegant and powerful framework for using those for computation. Examples are numerous from using actual in vitro grown neural networks [33], over chaotic attractors [34], the gene regulatory network of a bacterium [35], Cellular Nonlinear/Neural Networks [36] to optical fibers in a nonlinear feedback loop [37]. Some people have even literally used a bucket of water as a reservoir to do speech recognition [38].

Since the framework is this generic, we have researched in this work if integrated photonics would be a suitable hardware platform for reservoir computing.

2.2.5 Applications

As mentioned before, the elegance of reservoir computing is only a part of its appeal, its successful employment for a variety of tasks plays a major role as well. Examples are dynamic pattern classification [39], autonomous sine generation [19], grammar modeling [40] and the computation of highly nonlinear functions on the instantaneous rates of spike trains [41]. Another field where RC has made several contributions has been robotics, where it was used to control a simulated robot arm [42], to model an existing robot controller [43], to perform object tracking and motion prediction [44, 45], and event detection [46, 47]. The Reservoir Lab of ELIS has studied robot localization and event detection [48], behavior switching [49] and autonomous place cell discovery [50].

Applications in the field of digital signal processing include speech recognition [22, 51–54], noise modeling [55], Brain-Machine interfacing [56] and monitoring a power system [57]. The generation and prediction of monthly time series [58] and chaotic time series [39, 59–63] has also been a popular research topic and we want to highlight one particular example, where the Mackey-Glass chaotic time series could be predicted with several orders of magnitude better accuracy than classical techniques [55].

2.2.6 Optical neural networks

People from both the photonics and neural network community have long been interested in using optics as a hardware platform for neural networks. A com-

[1] The concept of transferring data to a higher-dimensional space to make the linear separation easier, is also the basis for so-called support vector machine methods [32].

puter computes the nodes of an ANN sequentially, while a hardware platform allows a parallel execution. Much of the research has been focused on building feedforward optical networks, imitating the threshold function of Figure 2.1 or spiking behavior and the training and adapting of the weights (often with free-space optics) [64–70]. This approach is fundamentally different from reservoir computing where the intrinsic dynamical properties of nonlinear systems are used, instead of trying to emulate software functionality for which they are not always suited[2]. However, testing which of the previous conceived optical implementations of optical neural networks are useful for reservoir computing is an interesting topic of future research.

This part described a short history of reservoir computing, from the origins in machine learning and artificial neural networks, to the general concept of computing with nonlinear dynamical systems. In the next section we will discuss the standard networks we used to compare our photonic reservoirs with.

2.3 Classical reservoir computing

After the ad hoc description given earlier, this section will first provide a more precise definition of what reservoir computing actually is, followed by a detailed description of the kind of classical reservoir we will be using in this dissertation: echo state networks. Finally we discuss the parameter used to qualify the dynamics in the network — the spectral radius.

2.3.1 Definition

Since RC is an umbrella term for the ensemble of techniques previously mentioned, it is hard to conclude upon an exact definition. We therefore introduce the description by David Verstraeten from his PhD thesis which says that a RC system consists of two parts: the reservoir and the readout function (Figure 2.2). The reservoir is a (usually) nonlinear dynamical system, consisting of a recurrently coupled network of relatively simple computational nodes. Classically, the connections between the nodes are randomly created and globally rescaled so that a suitable dynamical regime is reached. The readout function (or simply readout) is a linear classification or regression algorithm which is trained by example, using simple training mechanisms such as linear regression [1]. Because only the readout layer is trained, the same reservoir can be used to solve different task simultaneously on the same input as different readouts can be used for one reservoir (e.g., for both speech recognition and speaker identification).

[2]There is a similar ongoing discussion about optical computing and if it would not be better to adopt a new computational framework designed to exploit the intrinsic properties of light [71, 72], instead of mimicking transistor functionality.

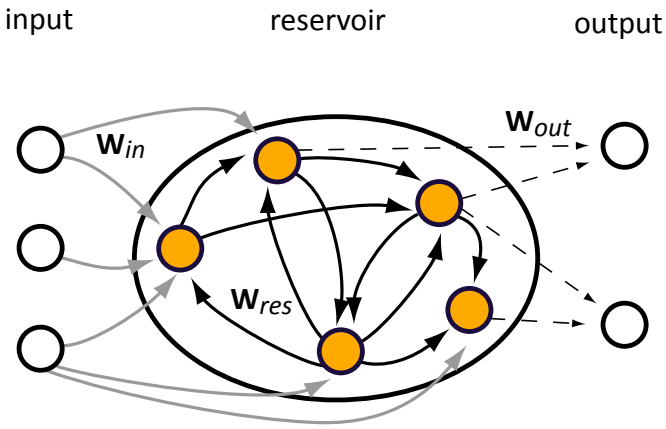


Figure 2.2: Schematic representation of a RC system. The inputs are fed into (all or a fraction of) the reservoir nodes, which are randomly interconnected with recurrent connections. The state of (all or a fraction of) the reservoir nodes is then fed into the outputs (dashed arrows). With every connection a weight is associated and these are grouped into weight matrices $\mathbf{W}$. The dashed arrows are the only connections where the weight is changed during training.

In this thesis the usefulness of integrated photonic reservoirs is researched and we will see that photonic reservoirs differ in at least one aspect from this definition: their topology is not randomly created, but instead is structured (Section 3.2.2).

2.3.2 Echo state networks

Reservoirs exist in many flavors, but the most common are still the original concepts discussed in Section 2.2.3 where either analogue sigmoid neurons (ESNs) or spiking neurons (LSMs) are used. Our proposed photonic reservoir architecture (described in the next chapter) most closely resembles an ESN, so that's what we will use as a baseline to evaluate our implementation's performance. The ESN paradigm was originally formulated in discrete time through the reservoir state update function:

$$\mathbf{x}[t+\Delta t] = (1-\eta)\mathbf{x}[t] + \eta f(\mathbf{W}_{in}\mathbf{u}[t] + \mathbf{W}_{res}\mathbf{x}[t]). \tag{2.2}$$

In this equation $\mathbf{x}[t]$ is a vector containing the states of all neurons at time t and $\mathbf{u}[t]$ collates the values of the input stimuli[3] at time t. The neuron function f

[3] In many complex tasks, such as speech recognition or robot control, the reservoir is driven by a large number of signals.

is usually a sigmoid function (e.g., the commonly used hyperbolic tangent or *tanh* function). The matrices $\mathbf{W}$ are the weight matrices for all the connections from the input to the reservoir ($\mathbf{W}_{in}$) and inside the reservoir itself ($\mathbf{W}_{res}$). The parameter η is called the *leak rate* and reflects the strength with which the neuron states are fed back linearly to their own outputs[4]. This neuron model, usually referred to as *leaky integrator neurons*, was introduced in [19], but we use the formalism from [1]. It effectively adds a first-order recursive low-pass filter to every neuron. If η equals one, the next state of the reservoir depends only on the external input and the current state of the reservoir. The closer η is to zero, the more the history of the previous states becomes important. For many temporal tasks (such as speech recognition), it is crucial that a reservoir has sufficient memory of past inputs [73]. Using leaky integrator neurons is a way to increase this memory.

The weights of the weight matrices $\mathbf{W}$ can be drawn randomly from a continuous distribution or a discrete set (e.g., $\{-1, 1\}$), and the sparsity is controlled by a connection fraction. $\mathbf{W}_{in}$ determines how many nodes are fed by input signals. For $\mathbf{W}_{res}$ the sparsity does not seem to matter much for ESNs but is important for LSMs [74]. Both weight matrices are globally scaled by a scale factor. For $\mathbf{W}_{in}$ this scaling determines how strongly the reservoir is driven by its inputs. For $\mathbf{W}_{res}$, the scaling has an important influence, which is described in the next section.

2.3.3 Spectral radius

For a reservoir to be useful for general computation on time series, it must have *fading memory*. This means that, although the reservoir state should depend on past inputs, this impact should fade away over time. A common way to obtain this is by rescaling the reservoir weight matrix $\mathbf{W}_{res}$ such that the reservoir is stable. The closer the reservoir dynamics are tuned to the edge of stability, the longer the system retains some information about past inputs. Tuning the reservoir dynamics close to the edge of stability has been found to yield optimal reservoir behavior for many tasks [31, 74].

A common parameter for tuning the dynamic regime of a reservoir is its *spectral radius* ρ, defined as the largest of the absolute values of the eigenvalues of the system's Jacobian at its maximal gain state. When all neurons have a maximum gain of one, as is the case with tanh neurons, this corresponds to the largest eigenvalue of the network's interconnection weight matrix $\mathbf{W}_{res}$. When the maximum gain differs from one, an upper bound can be obtained by incorporating the maximal gain of the neuron functions f. Let $\mathbf{W}_{res}$ be an $n \times n$ matrix with complex or real elements ($\mathbf{W}_{res} \in \mathbb{C}^{n \times n}$) and $\mathbf{G}_{max}$ an $n \times n$ matrix with the

[4] Often λ is used as the parameter for leak rate [1, 52], but to avoid confusion with photonics where this symbol usually denotes the wavelength of light, we use η.

maximum gain of the neurons, then an upper bound for the spectral radius ρ is the largest of the absolute value of its eigenvalues [75]:

$$\rho(\mathbf{W}_{res}) \leq \max_{1\leq i\leq n} \left|\text{eig}_i(\mathbf{G}_{max}\mathbf{W}_{res})\right|. \tag{2.3}$$

The spectral radius is an indication of the stability of the network, because systems theory tells us that a network will be unstable when $\rho > 1$ for zero input (when the reservoir behaves as a linear system)[5]. For non-zero input the network can become unstable for $\rho > 1$, but not necessarily so, since at each point in time, the actual gain of a nonlinear reservoir depends on its state and its input signal(s). Instability can occur if this gain on average exceeds one. As, by definition, the spectral radius gives an upper bound to the system's gain at each point in time, the edge of stability is usually found for a spectral radius slightly above one. However, tuning the spectral radius close to unity often yields reservoirs with close to optimal performance[6]. For randomly generated connection matrices, the spectral radius can be tuned to a value $\rho_{desired}$ by scaling the entire connection matrix $\mathbf{W}_{res}$ by a factor $\frac{\rho_{desired}}{\rho_{\mathbf{W}_{res}}}$.

2.4 Performance evaluation

Reservoir computing systems learn by generalizing from examples. In its simplest form this is done in a two-phase process in which the whole dataset is split into a training set and a test set. In the first phase the output weights (combined in a matrix $\mathbf{W}_{out}$) are trained and determined on the training set. In the second phase, the performance is evaluated on the test set with the weights from the first phase.

For the first phase, we start by taking all the samples from the training set and feeding them sequentially into the reservoir (after each sample, the reservoir is reset to each original state, usually zero). If we collect all the reservoir states into one big matrix $\mathbf{A}$[7], we need to find weights in order that

$$\mathbf{A}\mathbf{W}_{out} = \mathbf{B}, \tag{2.4}$$

is satisfied, with $\mathbf{B}$ being the desired outputs for the classifiers (Figure 2.3). Since the readout is usually time invariant, the weights in $\mathbf{W}_{out}$ do not change over

[5] Intuitively it can be understood as follows: over time, the weight matrix is a map, applied several times k. Matrix theory says this is related to taking the power of the eigenvalues and therefore the spectral radius ρ^k. When $\rho > 1$ this will grow over time, when $\rho < 1$ this will eventually return to zero for small perturbations.

[6] Actually the spectral radius, although often used in practice, is not a sufficient algebraic condition for the echo state property as mentioned in Section 2.2.3. The only known sufficient algebraic conditions for the echo state property are related to singular values (not eigenvalues) [20].

[7] In case not only the reservoir states are fed to the readout, but also the original inputs and/or a bias term, these are also put in $\mathbf{A}$.

time and there is only one weight per node, which means that Equation 2.4 will be overdetermined. The solution is found by doing least squares regression, i.e., finding $\mathbf{W}_{out}$ such that

$$\mathbf{W}_{out} = \arg\min_{\mathbf{W}} \|\mathbf{AW} - \mathbf{B}\|^2, \tag{2.5}$$

the quadratic error, is minimized. This can be done in one step with the Moore-Penrose generalized matrix inverse or pseudo-inverse $\mathbf{A}^\dagger$ of the matrix $\mathbf{A}$ [76, 77], which is defined as[8]: $\mathbf{A}^\dagger = (\mathbf{A^H A})^{-1}\mathbf{A^H}$ and

$$\mathbf{W}_{out} = \mathbf{A}^\dagger \mathbf{B} = (\mathbf{A^H A})^{-1}\mathbf{A^H B}. \tag{2.6}$$

The system can then be tested on the test set, by calculating its output

$$\mathbf{y} = \mathbf{W}_{out}\mathbf{A}', \tag{2.7}$$

with $\mathbf{A}'$ the reservoir response to the inputs from the test set. The final performance is determined by using some kind of metric to measure the difference between $\mathbf{B}$ and $\mathbf{y}$, which yields a certain performance depending on the error measure used for a specific task — usually some kind of Error Rate (ER). Ideally, the result should be the same for the training and test set, but one problem with Equation 2.6 is that it tends to overfit the training data and will therefore perform worse on unseen data. A solution to this, is controlling the model complexity which can be accomplished by keeping the norm of the weights small, and this will be discussed in the next section about regularization. Two other techniques that are common practice in machine learning, and that increase the robustness of the training are discussed in the two sections after that.

2.4.1 Regularization

A reservoir computing system should generalize from samples and discover the underlying model, but there is often a danger of overfitting, when the system learns the training samples almost by heart. When this happens the performance on the test data is considerably worse than on the training data. If we would train a system to distinguish between the sound of two words, ideally we want the system to generalize in such a way that the speaker's tone and voice are not important. Overfitting in this case would mean that only the pronunciation of the words from one specific speaker would be recognized[9]. This is clearly a

[8] $\mathbf{A^H}$ is the conjugate transpose of $\mathbf{A}$. When all the values in $\mathbf{A}$ and $\mathbf{B}$ are real as is often the case for classical reservoir computing, the conjugate transpose is replaced by the transpose $\mathbf{A^T}$ in this formula.

[9] Speaker recognition in itself is an interesting problem, where ideally the system should recognize the speaker, no matter what words he or she says.

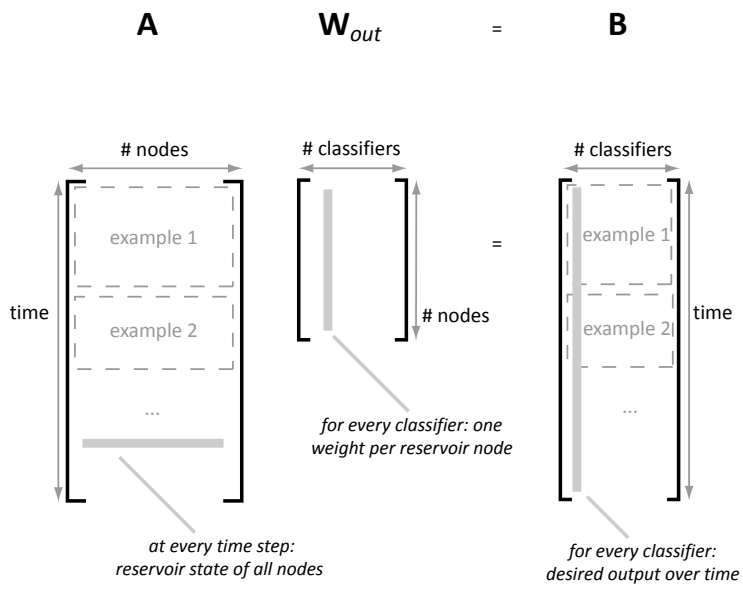


Figure 2.3: The dimensions for the matrix multiplication of Equation 2.4 are shown. **A** contains the collection of reservoir states over time concatenated for different input samples. The readout is time invariant, so $\mathbf{W}_{out}$ has one weight per reservoir node. Different classifiers can be trained on the same reservoir output as indicated by the columns of $\mathbf{W}_{out}$. The desired outputs in **B** are defined for every time step, input sample and classifier.

situation that has to be avoided, and a common approach is to add an additional penalty term to Equation 2.5 related to the norm of the weights, which yields:

$$\mathbf{W}_{opt} = \arg\min_{\mathbf{W}} \|\mathbf{AW} - \mathbf{B}\|_k + \gamma \|\mathbf{W}\|_l , \tag{2.8}$$

where $\|\cdot\|_p$ is the p norm, and γ is the regularization parameter controlling the tradeoff between the error on the training set and the weight norm. Depending on the values of k and l different regularization models exist, but in our case we always use $k = l = 2$, which corresponds to the Euclidian norms and it results in so-called *ridge regression* or Tikhonov regression [78]. As with standard least squares regression, $\mathbf{W}_{out}$ can also be found in one step with the Moore-Penrose inverse:

$$\mathbf{W}_{out} = (\mathbf{A}^{\mathbf{H}}\mathbf{A} + \gamma \mathbf{I})^{-1}\mathbf{A}^{\mathbf{H}}\mathbf{B}. \tag{2.9}$$

An equivalent approach for reducing the absolute magnitude of the weights (the weight norm), is adding noise to the data after the dynamical system [79]. However, adding noise is inherently non-deterministic and it was not used in this work.

2.4.2 Cross-validation

Cross-validation is a technique to avoid misleading results in small datasets due to a bad choice of the training and test set. If, for example, a class is present in the test set, but not in the training set, the readout will probably misclassify it, since it was never trained for in the first place. In essence, cross-validation means dividing the dataset in a number of subsets K. Ideally all subsets should have an equal size, but if not possible, they should be as similar in size as possible. $K-1$ subsets are then used for training, and the remaining subset is used to test the trained model with the techniques described in the previous section. This process is repeated K times (called K "folds"), so every subset is used once for testing (and all the other times for training). The final result of the reservoir is the average over the performances of every fold. The number of folds K is a tradeoff between computational cost and performance accuracy. In the extreme case of having as many folds as samples, the accuracy will be high since every sample will be used once for testing, but the calculation time will increase accordingly. In this thesis we always used five folds ($K = 5$).

Cross-validation is also used to determine the optimal value of the regularization parameter γ for a certain training set and this is depicted in Figure 2.4. First, a grid of possible values for γ is created (often logarithmically spaced to cover a broad range), and for every value the training set from Figure 2.5 is divided into K' folds. For every fold, the training set is split in a test set (test_γ) and a training set (training_γ). $\mathbf{W}_{out}$ is calculated for the training_γ set with Equation 2.9 and the corresponding output for the test_γ set with Equation 2.7. The average

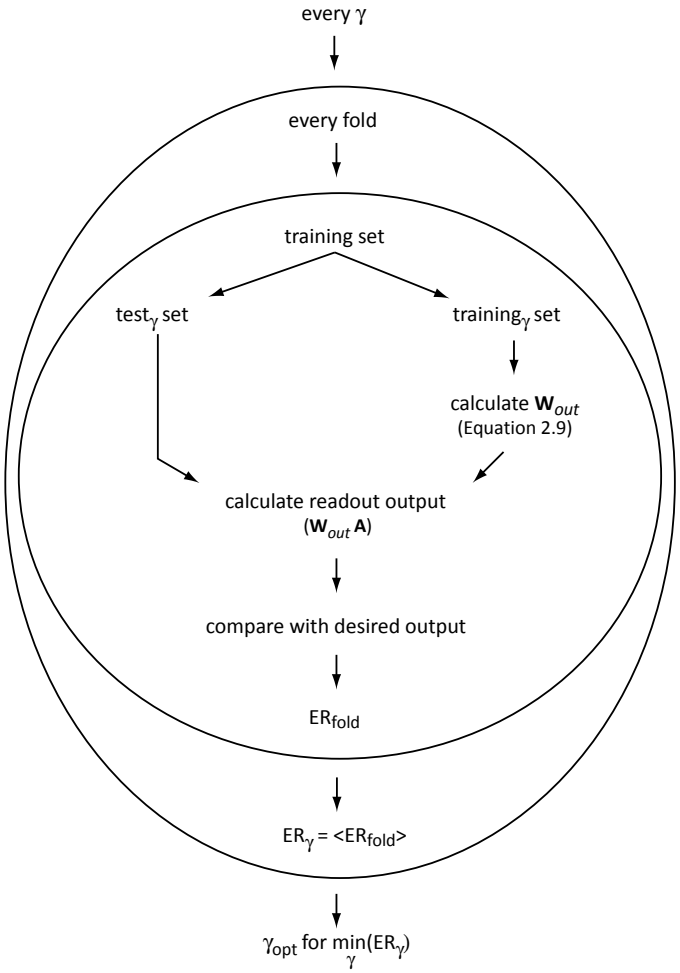


Figure 2.4: The optimal value of the regularization parameter γ is sought with grid-searching and cross-validation. First a grid of possible values is created (usually logarithmically to cover a broad range). For one value, the training set is again divided into two sets: a training set (train$_\gamma$) and a test set (test$_\gamma$), as many times as there are folds. For every fold, $\mathbf{W}_{out}$ is sought on the training$_\gamma$ set with Equation 2.9 and used to calculate the final readout of the system with the test$_\gamma$ set. Comparing this with the desired output, yields an ER for every fold. Averaging the ER over the folds, gives an ER for every regularization parameter. The optimal γ is the one corresponding with the lowest error rate and this is the γ_{opt} used in Figure 2.5.

result over the K' folds determines the error rate for that value of γ and the γ with the lowest value is the optimal regularization parameter γ_{opt}. Since this is an iterative process, a new array of possible γ values can be created around the found γ_{opt} and the whole process of Figure 2.4 repeated. In our simulations the process of creating a grid of values for γ was done three times. While this whole scheme can be computationally very costly (especially for a large number of folds and a large number of iterations for finding the γ_{opt}), it guarantees a near-optimal regularization [1]. In this dissertation the number of K' folds was also five.

The entire training process with regularization and cross-validation is shown in Figure 2.5. For every of the K folds, the samples are divided into a training and test set, and for the training set an optimal value for the regularization parameter γ is searched as discussed previously (Figure 2.4). The optimal value for γ is then used to determine the best readout weights through Equation 2.9. The output for the test set is calculated (Equation 2.7), and compared with the desired output, which yields a certain performance.

In this work we use the samples from the dataset both for training and testing through the process of cross-validation. However, in machine learning it is common to save yet another part of the data only for testing purposes (a validation set) and therefore never use it for training. This is especially relevant when trying to get the best performance on a benchmark task while ensuring generalization by the system. Since in this work we focused on the relative differences in performance between photonic and classical reservoirs, we chose to work without a separate validation set.

2.4.3 Unbalanced datasets and Fisher relabeling

The third and last mechanism we use to improve the robustness of the training is Fisher relabeling when confronted with unbalanced datasets. This can for example occur in a two-class classification, when there are substantially more examples of one class. This situation occurs also with multi-class problems with M classes (such as the speech recognition task of Chapter 5) when one-versus-all classifiers are used, in which case one classifier for each class needs to be trained to discern it from all the other classes [80]. In this case, even if the original dataset is balanced (i.e., it has as many examples for each class), it is unbalanced from the classifier point of view, since there will be $M-1$ as many negative examples as positive ones.

This imbalance has an effect on the generalization abilities of the classifier, since with the least squares method the separating hyperplane will shift towards the classes with most examples (dashed line in Figure 2.6). Ideally we want the hyperplane to lie in the middle between the one class and all the other classes

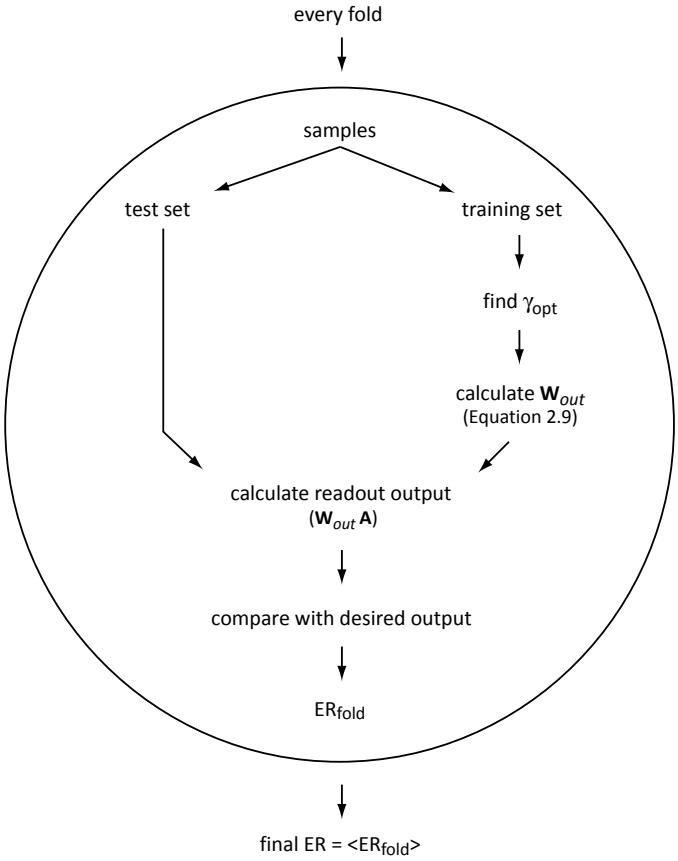


Figure 2.5: This shows the basic outline for training and testing with ridge regression and cross-validation. It is repeated as often as there are folds (K times), which should be less than or equal to the number of samples. For every fold the samples are divided into two sets: one for training and one for testing. The training set is first used to determine an optimal value for the regularization parameter γ (this process is shown in Figure 2.4). Once γ_{opt} is found, it is used to calculate the output weights $\mathbf{W}_{out}$ with Equation 2.9. These weights are then used on the unseen test data and give the final readout output of the system and when compared with the desired output, it yields the error rate for that fold. The error rate of the entire system is the average over all the folds

(solid line in Figure 2.6) and in order to achieve this, we can relabel the classes during training from the usual $\{-1,1\}$ for positive and negative examples, to $\left\{\frac{n_1+n_2}{n_1},-\frac{n_1+n_2}{n_2}\right\}$ for the case of two classes, but it can easily be extended to multiple classes. The new labels reflect the unbalance and counteract the shifting of the hyperplane[10]. Although recent work has shown that it is more correct rescaling both inputs and labels with $\left\{\sqrt{\frac{n_1+n_2}{n_1}},-\sqrt{\frac{n_1+n_2}{n_2}}\right\}$[11], our implementation used the old version.

In RC there can also be a time imbalance between examples with different durations, as the reservoir state at every timestep can be considered an example of a given class. This occurs for example with isolated digit recognition, since different digits can be shorter or longer in time. To determine the new labels, the instances of positive and negative examples are counted over time in the matrix **B** with desired outputs (Figure 2.3). It was experimentally shown that Fisher labeling leads to an improvement in performance in such a case [1].

The relabeling itself is only done in the training phase, not in the test phase. This means that the desired outputs in **B** are relabeled in Equation 2.9 for training, but when calculating the output for the test set through Equation 2.7, the output **y** is compared with the original **B**, not with the relabeled one, to determine the performance of the system.

2.4.4 Toolbox

In the Reservoir Lab (ELIS department, Ghent University) with which we collaborated, a simulation framework was developed to perform a wide range of RC simulations. This framework was originally written in MATLAB and is publicly available[12] together with its documentation. Different standard reservoirs (ESNs, LSMs), datasets, tasks and training mechanisms are built into the framework and the selection happens through a configuration file. At the same time, it is easy to write your own code and plug it into the toolbox, which is what we have done to simulate our photonic reservoirs with the first version of the toolbox. The three training mechanisms and their parameters previously discussed (regularization, cross-validation and Fisher relabeling) can be selected with the configuration file. Recently the development of this toolbox has been ceased in favor of a new framework, called *Oger*, written in Python[13]. The switch of the photonics code to this new toolbox has been completed recently by M. Fiers and T. Van Vaerenbergh from the Photonics Research Group, but all the simulations in this thesis were

[10] [80] shows that relabeling is equivalent to the so-called Fisher discriminant, which is a function that builds a linear hyperplane aiming to maximize the separation between two classes, while simultaneously minimizing the variance of samples within a class. This explains the name.

[11] This was communicated during the Machine Learning course taught at Ghent University.

[12] http://reslab.elis.ugent.be/rctoolbox

[13] http://organic.elis.ugent.be/organic/engine

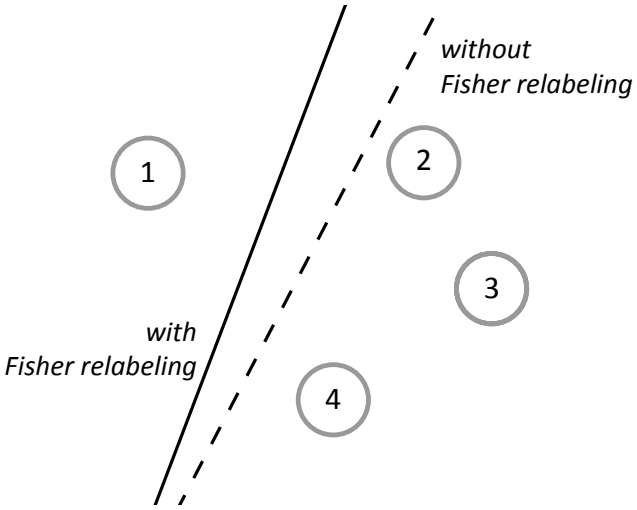


Figure 2.6: Depiction of the effect of Fisher relabeling on the location of the hyperplane dividing different classes (gray circles), in this case between class 1 and the other classes. Without Fisher relabeling the hyperplane is drawn towards the classes most frequent in the dataset (dashed line), but if the classes are relabeled this effect can be negated (solid line).

done with the first version of the Matlab tooblox.

2.5 Conclusions

In this chapter we first gave a short history of the concept of reservoir computing, which uses a nonlinear dynamical system combined with a simple readout for computation. We described a type of classical reservoir architecture (ESNs) which we will use in this dissertation to compare our photonic reservoirs with and evaluate their performance. In the last part we discussed the training methods used in this thesis for all the reservoirs and their implementation. In the next chapter we will introduce our photonic reservoirs and their most important features.

References

[1] D. Verstraeten. *Reservoir Computing : computation with dynamical systems.* PhD thesis, Ghent University, 2009.

[2] T. Kohonen. *Self-organizing maps.* Springer, 3rd edition, 2001.

[3] J. Hartigan. *Clustering algorithms.* John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 1975.

[4] W. Maass and C. M. Bishop. *Pulsed Neural Networks.* Bradford Books/MIT Press, Cambridge, MA, USA, 2001.

[5] B. Schrauwen. *Towards applicable spiking neural networks.* PhD thesis, Ghent University, 2008.

[6] W. Maass. *Noisy Spiking Neurons with Temporal Coding have more Computational Power than Sigmoidal Neurons.* In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, Proceedings of Advances in Neural Information Processing Systems (NIPS), pages 211–217, Denver, CO, USA, 1997. MIT Press.

[7] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[8] D. Rumelhart, G. Hinton, and R. Williams. *Parallel Distributed Processing*, chapter Learning internal representations by error propagation. MIT Press, Cambridge, MA, USA, 1986.

[9] 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, 3(37):328–339, March 1989.

[10] P. Rodriquez. *Simple recurrent networks learn context-free and context-sensitive languages by counting.* Neural Computation, 9(13):2093–2118, September 2001.

[11] F. Gers and J. Schmidhuber. *LSTM recurrent networks learn simple context free and context sensitive languages.* IEEE Transactions on Neural Networks, 6(12):1333–1340, November 2001.

[12] J. Suykens, J. Vandewalle, and B. De Moor. *Artificial Neural Networks for Modeling and Control of Non-Linear Systems.* Springer, 1996.

[13] A. J. Robinson. *An Application of Recurrent Nets to Phone Probability Estimation.* IEEE Transactions on Neural Networks, 5(2):298–305, 1994.

[14] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber. *Biologically Plausible Speech Recognition with LSTM Neural Nets*. In J. IJspeert, editor, Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (BioADIT), pages 175–184, Lausanne, Switzerland, January 2004.

[15] A. Graves and J. Schmidhuber. *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks*. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Proceedings of Advances in Neural Information Processing Systems (NIPS), pages 545–552, Vancouver, Canada, December 2009.

[16] B. Hammer and J. J. Steil. *Perspectives on learning with recurrent neural networks*. In Proceedings of the 10th European Symposium on Artifical Neural Networks (ESANN), Bruges, Belgium, April 2002.

[17] B. Pearlmutter. *Gradient calculations for dynamic recurrent neural networks: A survey*. IEEE Transactions on Neural Networks, 5(6):1212–1228, September 1995.

[18] W. Maass, T. Natschläger, and H. Markram. *Real-time computing without stable states: A new framework for neural computation based on perturbations*. Neural Computation, 14(11):2531–2560, 2002.

[19] H. Jaeger. *The "echo state" approach to analysing and training recurrent neural networks*. Technical Report GMD 148, German National Research Center for Information Technology, 2001.

[20] M. Bühner and P. Young. *A tighter bound for the echo state property*. IEEE Transactions on Neural Networks, 17(3):820–824, 2006.

[21] S. Jarvis, S. Rotter, and U. Egert. *Increased robustness and intermittent dynamics in structured Reservoir Networks with feedback*. In M. Verleysen, editor, Proceedings of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, April 2011.

[22] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout. *Isolated word recognition with the Liquid State Machine: a case study*. Information Processing Letters, 95(6):521–528, 2005.

[23] P. Joshi and W. Maass. *Movement generation with circuits of spiking neurons*. Neural computation, 17(8):1715–1738, August 2005.

[24] J. J. Steil. *Backpropagation-Decorrelation: Online recurrent learning with O(N) complexity*. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), volume 2, pages 843–848, 2004.

[25] P. Dominey. *Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning*. Biological Cybernetics, 73(3):265–274, August 1995.

[26] D. V. Buonomano and M. M. Merzenich. *Temporal information transformed into a spatial code by a neural network with realistic properties*. Science, 267(5200):1028–1030, February 1995.

[27] B. Schrauwen, D. Verstraeten, and J. Van Campenhout. *An overview of reservoir computing: theory, applications and implementations*. In Proceedings of the 15th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, April 2007.

[28] M. Lukoševičius and H. Jaeger. *Overview of Reservoir Recipes*. Technical report, Jacobs University, 2007.

[29] B. Hammer, B. Schrauwen, and J. J. Steil. *Recent advances in efficient learning of recurrent networks*. In Proceedings of the 17th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, April 2009.

[30] M. Lukoševičius and H. Jaeger. *Reservoir computing approaches to recurrent neural network training*. Computer Science Review, 3(3):127–149, August 2009.

[31] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. *An experimental unification of reservoir computing methods*. Neural Networks, 20(3):391–403, April 2007.

[32] V. N. Vapnik. *An overview of statistical learning theory*. IEEE Transactions on Neural Networks, 10(5):988–999, 1999.

[33] K. P. Dockendorf, I. Park, P. He, J. C. Principe, and T. B. DeMarse. *Liquid state machines and cultured cortical networks: The separation property*. Biosystems, 95(2):90–97, 2009.

[34] W. J. Goh and N. Crook. *Pattern recognition using chaotic transients*. In M. Verleysen, editor, Proceedings of the 15th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, April 2007.

[35] B. Jones, D. Stekel, J. Rowe, and C. Fernando. *Is there a Liquid State Machine in the Bacterium Escherichia Coli?* In Proceedings of the IEEE Symposium on Artificial Life (ALIFE), pages 187–191, 2007.

[36] D. Verstraeten, S. Xavier-de-Souza, B. Schrauwen, J. A. K. Suykens, D. Stroobandt, and J. Vandewalle. *Pattern classification with CNNs as reservoirs.* In Proceedings of the International Symposium on Nonlinear Theory and its Applications (NOLTA), pages 101–104, Budapest, Hungary, September 2008.

[37] Y. Paquot, F. Duport, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. *Artificial intelligence at light speed: toward optoelectronic reservoir computing.* Belgian Physical Society Magazine, 3:15–22, 2010.

[38] C. Fernando and S. Sojakka. *Pattern recognition in a bucket.* In Proceedings of Advances in Artificial Life, volume 2801, pages 588–597, Dortmund, Germany, September 2003.

[39] H. Jaeger. *Short term memory in echo state networks.* Technical Report GMD 152, German National Research Center for Information Technology, 2001.

[40] M. H. Tong, A. D. Bickett, E. M. Christiansen, and G. W. Cottrell. *Learning grammatical structure with echo state networks.* Neural Networks, 20(3):424–432, 2007.

[41] W. Maass, T. Natschläger, and H. Markram. *Fading memory and kernel properties of generic cortical microcircuit models.* Journal of Physiology, 98(4-6):315–330, 2004.

[42] P. Joshi and W. Maass. *Movement generation and control with generic neural microcircuits.* In J. IJspeert, editor, Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (BioADIT), pages 258–273, Lausanne, Switzerland, January 2004.

[43] H. Burgsteiner. *On learning with recurrent spiking neural networks and their applications to robot control with real-world devices.* PhD thesis, Graz University of Technology., 2005.

[44] H. Burgsteiner. *Training networks of biological realistic spiking neurons for real-time robot control.* In Proceedings of the 9th International Conference on Engineering Applications of Neural Networks, pages 129–136, Lille, France, August 2005.

[45] W. Maass, R. A. Legenstein, and H. Markram. *A new approach towards vision suggested by biologically realistic neural microcircuit models.* In Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision (BMCV), volume 2525, pages 282–293, Tubingen, Germany, November 2002.

[46] J. Hertzberg, H. Jaeger, and F. Schonherr. *Learning to Ground Fact Symbols in Behavior-Based Robots.* In F. van Harmelen, editor, Proceedings of the 15th European Conference on Artificial Intelligence, pages 708–712, Lyon, France, July 2002.

[47] H. Jaeger. *Reservoir riddles: suggestions for echo state network research (extended abstract).* In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), pages 1460–1462, Montreal, Canada, July 2005. IEEE.

[48] E. A. Antonelo, B. Schrauwen, and D. Stroobandt. *Event detection and localization for small mobile robots using reservoir computing.* Neural networks, 21(6):862–871, 2008.

[49] E. A. Antonelo, B. Schrauwen, and D. Stroobandt. *Modeling multiple autonomous robot behaviors and behavior switching with a single reservoir computing network.* In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), pages 1843–1848, Singapore, October 2008.

[50] E. A. Antonelo, B. Schrauwen, and D. Stroobandt. *Unsupervised Learning in Reservoir Computing: Modeling Hippocampal Place Cells for Small Mobile Robots.* In Proceedings of the International Conference on Artificial Neural Networks (ICANN), 2009.

[51] M. D. Skowronski and J. G. Harris. *Automatic speech recognition using a predictive echo state network classifier.* Neural Networks, 20(3):414–423, 2007.

[52] H. Jaeger. *Discovering multiscale dynamical features with hierarchical echo state networks.* Technical Report 10, German National Research Center for Information Technology, 2007.

[53] A. Ghani, T. McGinnity, L. Maguire, and J. Harkin. *Neuro-inspired Speech Recognition with Recurrent Spiking Neurons.* In Proceedings of the International Conference on Artificial Neural Networks, Part I (ICANN), pages 513–522, 2008.

[54] W. Maass, T. Natschläger, and H. Markram. *A Model for Real-Time Computation in Generic Neural Microcircuits.* In S. Becker, S. Thrun, and K. Obermayer, editors, Proceedings of Advances in Neural Information Processing Systems (NIPS), pages 229–236. MIT Press, 2003.

[55] H. Jaeger and H. Haas. *Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication.* Science, 304(5667):78–80, April 2004.

[56] Y. Rao, S.-P. Kim, J. Sanchez, D. Erdogmus, J. Principe, J. Carmena, M. Lebedev, and M. Nicolelis. *Learning mappings in brain machine interfaces with echo state networks.* In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 233–236, Philadelphia, PA, USA, March 2005.

[57] G. K. Venayagamoorthy. *Online design of an echo state network based wide area monitor for a multimachine power system.* Neural networks, 20(3):404–413, April 2007.

[58] F. Wyffels and B. Schrauwen. *A comparative study of Reservoir Computing strategies for monthly time series prediction.* Neurocomputing, 73(10-12):1958–1964, June 2010.

[59] S. F. Crone, K. Nikolopoulos, and M. Hibon. *Automatic modelling and forecasting with artificial neural networks - a forecasting competition evaluation.* Technical report, Lancaster University Management School, 2008.

[60] F. Wyffels, B. Schrauwen, and D. Stroobandt. *Using reservoir computing in a decomposition approach for time series prediction.* In A. Lendasse, editor, Proceedings of the European Symposium on Time Series Prediction, pages 149–158, 2008.

[61] J. J. Steil. *Memory in Backpropagation-Decorrelation O(N) Efficient Online Recurrent Learning.* In Proceedings of the International Conference on Artificial Neural Networks (ICANN), pages 649–654, Warsaw, Poland, September 2005.

[62] J. J. Steil. *Online stability of backpropagation-decorrelation recurrent learning.* Neurocomputing, 69(7-9):642–650, March 2006.

[63] H. Jaeger. *Adaptive nonlinear system identification with echo state networks.* In Proceedings of Advances in Neural Information Processing Systems (NIPS), pages 593–600. MIT Press, 2003.

[64] E. C. Mos. *Optical Neural Network based on Laser Diode Longitudinal Modes.* PhD thesis, Eindhoven University of Technology, 1999.

[65] M. Hill, E. Edward, E. Frietman, H. de Waardt, H. J. S. Dorren, and G. Khoe. *All Fiber-Optic Neural Network Using Coupled SOA Based Ring Lasers.* IEEE Transactions on Neural Networks, 13(6):1504–1513, 2002.

[66] K. Kravtsov, M. P. Fok, D. Rosenbluth, and P. R. Prucnal. *Ultrafast all-optical implementation of a leaky integrate-and-fire neuron.* Optics Express, 19(3):211–221, January 2011.

[67] G. Moagar-Poladian and M. Bulinski. *Optical reconfigurable neuron by using the transverse Pockels effect.* Journal of Optoelectronics and Advanced Materials, 4(4):929 – 936, December 2002.

[68] A. S. Younger and E. Redd. *Fixed-Weight learning Neural Networks on Optical Hardware.* In Proceedings of the International Joint Conference on Neural Networks (IJCNN), pages 3457–3463, Atlanta, GA, USA, June 2009.

[69] S. Jutamulia and F. T. S. Yu. *Overview of hybrid optical neural networks.* Optics & Laser Technology, 28(2):59–72, March 1996.

[70] H. Y. S. Li, Y. Qiao, and D. Psaltis. *Optical Network for Real-Time Face Recognition.* Applied Optics, 32(26):5026–5035, September 1993.

[71] D. A. B. Miller. *Are optical transistors the logical next step?* Nature Photonics, 4(1):3–5, January 2010.

[72] J. Hardy and J. Shamir. *Optics inspired logic architecture.* Optics Express, 15(1):150–165, January 2007.

[73] E. A. Antonelo, B. Schrauwen, X. Dutoit, D. Stroobandt, and M. Nuttin. *Event detection and localization in mobile robot navigation using reservoir computing.* In Proceedings of the International Conference on Artificial Neural Networks (ICANN), pages 660–669, Porto, Portugal, September 2007.

[74] L. Büsing, B. Schrauwen, and R. Legenstein. *Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons.* Neural computation, 22(5):1272–1311, May 2010.

[75] I. S. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and Products, 6th ed.* Academic Press, San Diego, CA, USA, 2000.

[76] E. H. Moore. *On the reciprocal of the general algebraic matrix.* In Bulletin of the American Mathematical Society, volume 26, pages 394–395, 1920.

[77] R. Penrose. *A generalized inverse for matrices.* In Proceedings of the Cambridge Philosophical Society, volume 51, pages 406–413, 1955.

[78] A. N. Tikhonov and V. I. Arsenin. *Solutions of ill-posed problems*. Scripta series in mathematics. Winston ; distributed solely by Halsted Press, Washington; New York, 1977.

[79] F. Wyffels, B. Schrauwen, and D. Stroobandt. *Stable Output Feedback in Reservoir Computing Using Ridge Regression*. In Proceedings of the 18th International Conference on Artificial Neural Networks (ICANN), pages 808–817, Prague, Czech Republic, September 2008.

[80] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2001.

3

Photonic reservoir computing

This chapter describes our photonic reservoir architecture and highlights its differences with traditional ESNs. The first part discusses the photonic component we have used as a network node, a Semiconductor Optical Amplifier (SOA), the second part deals with the connectivity and network structure between these nodes. Semiconductor optical amplifiers were chosen as a first testbed for an integrated version of photonic reservoir computing for a number of reasons. First of all, SOAs provide gain and therefore no separate component is needed to compensate for losses. Second, they are broadband, which relaxes fabrication tolerances in comparison to resonator-type devices, such as coupled cavities or ring resonators, where the resonances of different nodes usually have to (at least partially) overlap. Third, their steady state characteristic somewhat resembles the upper part of a hyperbolic tangent, making them a perfect bridge between the existing knowledge of reservoir computing with *tanh* neural networks and the new field of photonic reservoir computing.

3.1 Semiconductor optical amplifiers

Losses occur in every realistic system and these have to be compensated. E.g., even though the loss for light signals in optical fibers (0.1–0.2 dB/km) is incredibly low, repeaters are still needed every 100 km. In practice the signal is usually distorted by a range of effects such as dispersion, nonlinear interaction and

noise, so that the signal needs to be reamplified, reshaped and retimed. 3R-regeneration is the combination of those three operations and in most networks nowadays it is done with electronic repeaters, so that the optical signal needs to be transferred to the electrical domain and back. A topic of intensive research is to implement this, and also routing, all-optically. The PhD thesis of Koen Huybrechts is an excellent example of recent progress being made [1].

In optical fiber communication there are basically two approaches to amplification: the first is to have a fiber-based mechanism that amplifies the signal in a distributed way along a section of the fiber. The second approach is to first send the signal into a chip where it is amplified by an integrated amplifier, after which it is send back into a fiber. Since it is easier to stay in a fiber once you are in a fiber, it should come as no surprise that the fiber amplifiers were hugely successful once discovered. The two basic mechanisms are Raman amplification and gain through rear-earth elements such as Erbium [2]. The advent of Erbium-Doped Fiber Amplifiers (EDFAs) proved to be so successful that it slowed down research on coherent optical networks for almost 10 years, because the EDFAs' broadband amplification combined with Wavelength Division Multiplexing (WDM: sending information on multiple wavelengths independently and simultaneously through the same fiber) provided more bandwidth than required. Coherent networks exploit better the information capacity of light by also detecting the phase of an optical signal, and they have been making a comeback in recent years because the EDFA–WDM combination now struggles to meet future bandwidth requirements [3–5]. This will be discussed in Section 3.2.5.

The most popular way for amplification integrated on a chip is using semiconductor optical amplifiers. They can provide a very high gain over short distances (hundreds of micrometers) and can be electrically pumped, as opposed to for example EDFAs which reach the same gain levels over meters and need to be pumped optically with high-power lasers. The next section discusses some basic theory and properties of SOAs.

3.1.1 SOA principles

Light is an electromagnetic wave and its electric field can exert a force on the electric charges and dipoles in matter, causing them to vibrate or accelerate. The rules of quantum mechanics determine that these charges can only have certain energy levels and the nature of these levels allows to group materials into isolated atoms, metals, semiconductors and insulators [6]. In solids the allowed energy levels are no longer discrete levels such as in isolated atoms, but bands. The bands are filled with electrons, starting from the lowest levels. The highest band that is filled with electrons is called the valence band. The band above it is called the conduction band. Electrons in the conduction band can move

freely through the solid and can carry current. Depending on the separation between the valence and conduction band, solids can be divided into metals, semiconductors and insulators with different electrical properties (Figure 3.1). In a metal the conduction band and valence band overlap, so there are always free electrons available for carrying current. Insulators on the other hand have a large band gap (typically > 3 eV) and it is difficult for electrons to bridge the gap. Semiconductors have a small band gap and electrons can fairly easily make the jump, providing that the necessary energy — in the form of heat, light or an electric field — is provided. It is this small band gap that makes semiconductors so useful for light generation in Light-Emitting Diodes (LEDs) and lasers.

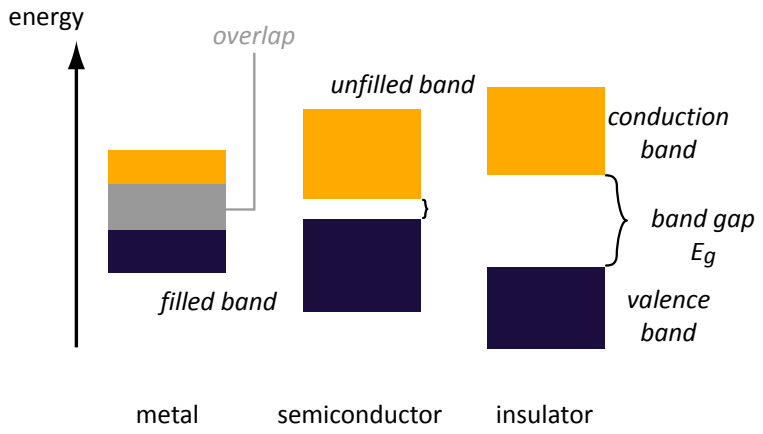


Figure 3.1: Simplified diagram of the electron energy levels in solids. The levels of energies that electrons are allowed to occupy are bands and they are filled from the lowest energy up. The highest energy band filled with electrons is called the valence band, and the band above this one is called the conduction band and electrons in this band can move freely through the solid. The energy that separates these two bands is called the band gap E_g. In insulators the band gap is large and it is difficult for electrons to move from the valence to the conduction band. Semiconductors have a small band gap, so only a little bit of energy is required for electrons to bridge the gap. The valence band and conduction band overlap for metals, so there are always electrons that can move freely and carry current.

The operation of SOAs is similar to that of semiconductor lasers. Mostly through electrical pumping (although it can also be done optically), carriers are brought into an excited state within the gain medium, i.e., bridging the band gap by going from the valence band to the conduction band. A photon can interact with these excited carriers, forcing the carrier to release its energy and returning

to its ground state, while simultaneously emitting a photon (Figure 3.2(a)). This photon will be identical to the original photon having the same frequency, phase and direction. This process is called *stimulated emission*. The other two possible interactions between carriers and photons are *absorption*, where a photon excites a carrier from its ground state (Figure 3.2(b)), and *spontaneous emission*, where an excited carrier without external stimulation falls back to its ground state while emitting a photon (figure 3.2(c)). All three processes occur simultaneously but only when a state of population inversion is reached (there are more carriers in their excited state than in their ground state) is stimulated emission the dominant process[1]. Pumping is needed to obtain population inversion [6].

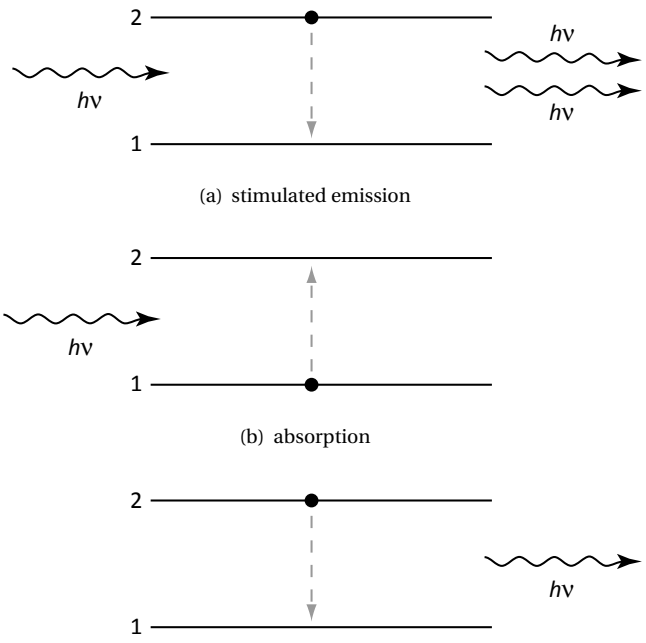
In a laser, feedback with some kind of mirror, is added to reflect the light back and forth and let it pass multiple times through the gain medium. This will result in some form of wavelength selectivity, making the laser a coherent source. For an amplifier, broadband operation is desired, so the challenge is to remove any form of wavelength selectivity, other than that of the gain medium itself. A Traveling-Wave SOA (TW SOA) is an amplifier through which the light only passes once, and therefore any reflection needs to be eliminated as much as possible. This is usually done by adding a high-quality anti-reflection coating onto the facets, sometimes combined with an angled facet (Figure 3.3(a)). If G is the gain from the SOA, R_1 and R_2 the reflectivities for both facets (Figure 3.3(a)), then an SOA is considered to be a TW amplifier when

$$G\sqrt{R_1 R_2} < 0.17 \tag{3.1}$$

is satisfied [2]. For an SOA with a gain of 30 dB ($G = 1000$), each facet has to have a reflectivity of less than 1.7×10^{-4}, which is very small and requires careful design.

SOAs are usually made from III-V compound semiconductors such as GaAs/AlGaAs, InP/InGaAsP or InP/InAlGaAs depending on the central wavelength where one wants amplification. A combination of two materials is given because one is used as the active layer and the other for the cladding layers (Figure 3.3(a)). The active layer has a smaller band gap than the surrounding cladding layers which confines the injected carriers coming from and through the cladding layers to the active region, where they can amplify light through stimulated emission (Figure 3.2(a)). Materials with a smaller band gap also have a higher refractive index, which makes that the active layer has a higher refractive index than the surrounding cladding layers. This is the basis for waveguiding in optics (also in optical fibers) and light is therefore generated in and guided by the active layer [2].

[1] In semiconductors the carriers actually consist of electrons and holes, although it is easy to think just in terms of electrons as carriers. Framed in terms of holes and electrons, an electron bridges the band gap when it absorbs a photon, leaving a (positively charged) hole behind. Spontaneous and stimulated emission are electron-hole recombinations, where a photon is released in the process.



(a) stimulated emission

(b) absorption

(c) spontaneous emission

Figure 3.2: Interactions between photons and electrons on two energy levels with an energy difference $h\nu$ (h is Planck's constant and ν is the frequency of the electromagnetic wave): (a) stimulated emission whereby an incoming photon with energy $h\nu$ stimulates an excited electron to emit a clone photon (same direction, frequency and phase) while relaxing to its ground state; (b) an incoming photon with energy $h\nu$ gets absorbed by an electron causing it to jump from the lower energy state to the higher; (c) an excited electron falls back from the high energy state to the lower one, releasing it excess energy by emitting a photon with energy $h\nu$.

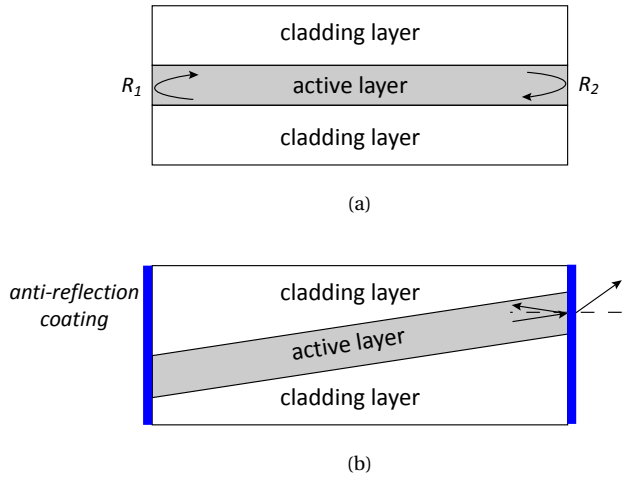


Figure 3.3: Top view of the structure of an SOA: (a) the active layer that provides gain is sandwiched between two cladding layers that confine both light and carriers into the active region. Normal facets reflect a portion of the light; (b) to make a traveling-wave SOA the reflectivity of the facets has to be decreased. Two approaches are shown: one is to tilt the active region so that the reflected light will no longer be guided (indicated by the arrows). The second is to use anti-reflection coatings on the facets.

The two most relevant nonlinear characteristics of SOAs for reservoir computing are *gain saturation* and *carrier lifetime*. Gain saturation means that the gain will decrease for higher input powers as can be seen in Figure 3.4, where the relation between input power and corresponding output power is portrayed for SOAs. The upper part of a hyperbolic tangent with the same gain for zero input is also shown. As mentioned before, the gain in an SOA originates from excited carriers being consumed by photons, creating copies of the photons trough stimulated emission. If more photons are inserted into the SOA, more carriers are consumed. This, in turn, decreases the density of excited carriers, leaving less opportunity for other photons to be amplified and thus the gain diminishes. This effect is the basis for gain saturation in SOAs. The steady state characteristic of an SOA resembles a hyperbolic tangent but it does not completely saturate (Figure 3.4). In our simulations, the input of the SOAs is often in the more linear regime with input powers not exceeding 1 mW. In this region, both curves are very similar.

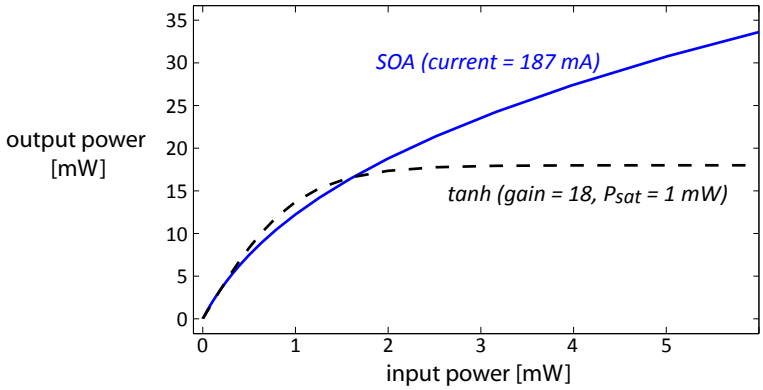


Figure 3.4: The modeled (power in – power out) curve for an SOA and a tanh with gain (18) and power saturation (1 mW).

When carriers are stimulated into an excited state, they only 'live' for a certain amount of time before they relax to their ground state. The mean time associated with this process is called the *carrier lifetime*, with typical values of 100–300 ps. One could say that the SOA encodes information about its past inputs into its internal state variable, the carrier density. The dynamic response of this variable to the input power resembles that of a first order system with the carrier lifetime as a dominant time constant. This is especially important for large, abrupt changes in the input power. When such a change occurs, the gain cannot change abruptly to the corresponding value on the steady state curve and will initially stay the same. After that initial response, the gain will relax to its steady state

value with the carrier lifetime as time constant, as can be seen in Figure 3.5. This means that, for a sudden increase in the input, which should saturate the SOA and decrease its gain, the output will initially 'overreact' to the input change, followed by a relaxation as the gain approaches its lower steady state value. A similar effect occurs for a sudden decrease of the input power (Figure 3.5)[2]. This dynamic carrier effect is one thing that sets the behavior of SOAs apart from a hyperbolic tangent which is a purely static component.

Light as an electromagnetic wave has a polarization which describes the orientation of the oscillations. As a convention it is specified by the orientation of the electric field. In free space the general polarization is elliptic with linear and circular as special cases [6]. In waveguiding structures it is slightly more complex, but when the dimensions are small enough so that only one mode is guided, it depends on the geometry. Structures such as single-mode fibers have circular symmetry and support two orthogonal polarizations that are degenerate, while integrated waveguides and also SOAs are usually inherently polarization specific due to their asymmetry (e.g., a rectangular profile) and therefore those circuits and components only work well for a single polarization [7]. The fact that the gain of SOAs is often polarization dependent, is an issue when light is guided by optical fibers supporting two orthogonal polarizations and where due to random imperfections power gets transferred between the two. However, if we work with polarization specific integrated waveguides and SOAs and if we control the polarization of the input light with polarization controllers, only the desired polarization will be excited and maintained throughout the circuit [8]. We have therefore only considered light of a single polarization.

3.1.2 Model

The previous section described the basic properties of SOAs and now we will explore a model that incorporates them. The unidirectional model we use in our simulations was proposed by Agrawal [9]. It allows for fast computation and captures all the basic features, such as gain saturation, carrier dynamics and a phase shift depending on the gain.

If we start from an ideal traveling-wave amplifier and assume that the active-region dimensions only support a single waveguide mode and that the light is

[2]The story can also be told from the perspective of the carriers. They are fed into the active region with a constant rate, dependent on the current. Without input power and thanks to the losses due to spontaneous emission, the carrier density will stabilize at a certain value. If light enters the SOA, carriers are consumed and the carrier density decreases (since the input current remains the same), ultimately leading to gain saturation at high input powers. If there is a sudden increase in input power, the amount of carriers are initially sufficient to provide enough gain, but as they get consumed the input current is too low to sustain that amount of carriers and their number decreases and with them the gain.

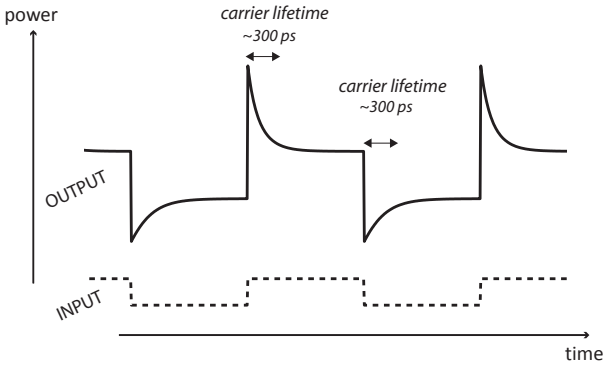


Figure 3.5: The modeled response to sudden changes in the input power for an SOA. At first when there is a change in input, the SOA reacts with the same gain as before, after which it relaxes with the carrier lifetime as time constant to its steady state gain for the new input power.

linearly polarized and stays that way, we can write the electrical field inside the SOA as [9]

$$E(x,y,z,t) = \hat{\mathbf{x}}\mathrm{Re}\left\{F(x,y)A(z,t)\exp(j(\beta_0 z - \omega_0 t))\right\}, \tag{3.2}$$

where $\hat{\mathbf{x}}$ is the polarization unit vector, β_0 the propagation constant at the carrier frequency ω_0, $F(x,y)$ is the waveguide-mode distribution and $A(z,t)$ the slowly-varying envelope. x,y are the transverse dimensions, z the direction of propagation and t is time. $F(x,y)$ can be approximated by a Gaussian distribution and its frequency dependence can be ignored for pulses whose spectral width $\Delta\omega$ is much smaller than ω_0 [2]. The waveguide-mode distribution itself is not relevant for us, but we will incorporate its transverse effects by introducing the confinement factor Γ which describes the fraction of the transverse mode in the active region:

$$\Gamma = \frac{\int_0^W \int_0^H \left|F(x,y)\right|^2 \mathrm{d}x\,\mathrm{d}y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left|F(x,y)\right|^2 \mathrm{d}x\,\mathrm{d}y}, \tag{3.3}$$

where H and W are the height and width of the active region.

The amplitude $A(z,t)$ of the pulse evolves as [2]:

$$\frac{\partial A}{\partial z} + \frac{1}{v_g}\frac{\partial A}{\partial t} = \frac{1}{2}(1 - j\beta_c)gA, \tag{3.4}$$

where carrier induced index changes are included through the linewidth enhancement factor β_c and the gain by the factor g. This equation says that while the pulse propagates (left side), it gets amplified, while simultaneously causing

index changes. v_g is the group velocity and it is the speed by which the pulse propagates ($v_g = c/n_g$ in which n_g is the group index and c the speed of light). In this equation we neglect dispersion (the wavelength dependence of the refractive index) since its effect is small for typical amplifier lengths ($L = 0.2$ to 0.5 mm) and pulse widths (> 1 ps) [9]. Also internal losses are neglected, which is common when the gain is high and also since it only results in a shift of the steady state curve. In reservoir computing the total gain and loss is important, so the 'lack' of loss in the SOA model can be compensated by having higher losses in the waveguides.

SOAs can provide gain through the process of stimulated emission and excited carriers (Figure 3.2(a)), therefore the gain is determined by the carrier density and the following rate equation under the assumption that the transverse dimensions (width W and height H) are much smaller than the carrier diffusion length, while the length L is much larger[3]:

$$\frac{\partial N}{\partial t} = \frac{I}{qV} - \frac{N}{\tau_c} - \frac{g(N)}{\hbar\omega_0 HW}|A|^2, \tag{3.5}$$

where N is the carrier density per unit volume (for both holes and electrons), q is the elementary charge, V the volume of the active region, I the injection current, τ_c the carrier lifetime and $\hbar\omega_0$ the energy of the photons at the carrier frequency. The first term on the right hand side shows the rate by which carriers are added to the active region because of the input current, the second term the loss of carriers due to spontaneous relaxation of the carriers determined by their lifetime τ_c(Figure 3.2(c)) and the last term the loss of carriers due to amplification of the light. The term $|A|^2$ is the total power P of the light. Dividing it by the energy of one photon gives the total number of photons. The gain g per unit length is defined as [9]

$$g(N) = \Gamma\sigma_g(N - N_0), \tag{3.6}$$

where Γ is the previously introduced confinement factor, N_0 the carrier density needed to reach transparency and σ_g the differential gain. If the carrier density is below N_0, there is no amplification and the light gets absorbed. The equation says that only the light inside the active region gets amplified and only the carriers above transparency contribute to the gain.

To get the equations in a more convenient form, we first differentiate Equations 3.6 with respect to t and insert Equation 3.5:

$$\frac{\partial g}{\partial t} = \Gamma\sigma_g\frac{\partial N}{\partial t} = \Gamma\sigma_g\frac{I}{qV} - \Gamma\sigma_g\frac{N}{\tau_c} - \Gamma\sigma_g\frac{g}{\hbar\omega_0 HW}|A|^2. \tag{3.7}$$

[3] In the original paper [9] there is a small mistake for this formula as the factor HW is omitted, but in a later book [2] this was corrected. Both use a slightly different interpretation of N. The former uses it as the carrier density, while the latter as the total number of carriers in the active region.

By introducing the saturation energy E_{sat} of the amplifier

$$E_{sat} = \frac{\hbar\omega_0 HW}{\Gamma\sigma_g}, \tag{3.8}$$

and by artificially adding and subtracting the term $\Gamma\sigma_g N_0/\tau_c$ we rewrite Equation 3.7 as

$$\frac{\partial g}{\partial t} = \Gamma\sigma_g\frac{I}{qV} - \Gamma\sigma_g\frac{N}{\tau_c} + \Gamma\sigma_g\frac{N_0}{\tau_c} - \Gamma\sigma_g\frac{N_0}{\tau_c} - \frac{g}{E_{sat}}|A|^2, \tag{3.9}$$

which allows us to combine term 2 and 3 and inserting Equation 3.6

$$\begin{aligned}\frac{\partial g}{\partial t} &= \Gamma\sigma_g\frac{I}{qV} - \Gamma\sigma_g\frac{N_0}{\tau_c} - \frac{\Gamma\sigma_g(N-N_0)}{\tau_c} - \frac{g}{E_{sat}}|A|^2,\\ \frac{\partial g}{\partial t} &= \Gamma\sigma_g\frac{I}{qV} - \Gamma\sigma_g\frac{N_0}{\tau_c} - \frac{g}{\tau_c} - \frac{g}{E_{sat}}|A|^2.\end{aligned} \tag{3.10}$$

We can factorize the first two terms into

$$\frac{\Gamma\sigma_g N_0}{\tau_c}\left(\frac{I}{\frac{qVN_0}{\tau_c}} - 1\right), \tag{3.11}$$

and by introducing the current required for transparency I_0 as a new variable (VN_0 is the number of carriers needed for transparency in the active region and the current needed to supply it, depends on the carrier lifetime)

$$I_0 = \frac{qVN_0}{\tau_c}, \tag{3.12}$$

Equation 3.11 becomes

$$\frac{\Gamma\sigma_g N_0}{\tau_c}\left(\frac{I}{I_0} - 1\right). \tag{3.13}$$

Finally, we define the small-signal gain g_0 as

$$g_0 = \Gamma\sigma_g N_0\left(\frac{I}{I_0} - 1\right) \tag{3.14}$$

which gives us the following rate equation for the gain g when we insert it in Equation 3.10

$$\frac{\partial g}{\partial t} = \frac{g_0 - g}{\tau_c} - \frac{g}{E_{sat}}|A|^2. \tag{3.15}$$

This explains the name small-signal gain, since without input ($|A|^2 = 0$), Equation 3.15 shows that the steady state value of the gain will be equal to g_0.

Equation 3.4 describes the pulse propagation and Equation 3.15 the evolution of the gain inside the SOA, but we can simplify things further by making a transformation to another reference frame. Here, the reduced time $\tau = t - z/v_g$

is measured in a reference frame moving with the light and v_g is the speed of the pulse and the new frame. This gives us the following equations for the new coordinates (z', τ)

$$z' = z \tag{3.16}$$
$$\tau = t - \frac{z}{v_g}, \tag{3.17}$$

which means that the derivatives in Equation 3.4 become

$$A(z,t) = A'(z',\tau) \tag{3.18}$$

$$\begin{aligned} \frac{\partial A}{\partial t} &= \frac{\partial A'}{\partial z'}\frac{\partial z'}{\partial t} + \frac{\partial A'}{\partial \tau}\frac{\partial \tau}{\partial t} \\ \frac{\partial A}{\partial t} &= \frac{\partial A'}{\partial z'}0 + \frac{\partial A'}{\partial \tau}1 \\ \frac{\partial A}{\partial t} &= \frac{\partial A'}{\partial \tau} \end{aligned} \tag{3.19}$$

$$\begin{aligned} \frac{\partial A}{\partial z} &= \frac{\partial A'}{\partial z'}\frac{\partial z'}{\partial z} + \frac{\partial A'}{\partial \tau}\frac{\partial \tau}{\partial z} \\ \frac{\partial A}{\partial z} &= \frac{\partial A'}{\partial z'}1 + \frac{\partial A'}{\partial \tau}\frac{-1}{v_g} \\ \frac{\partial A}{\partial z} &= \frac{\partial A'}{\partial z'} - \frac{1}{v_g}\frac{\partial A'}{\partial \tau}, \end{aligned} \tag{3.20}$$

and Equation 3.4 itself becomes

$$\begin{aligned} \frac{\partial A'}{\partial z'} - \frac{1}{v_g}\frac{\partial A'}{\partial \tau} + \frac{1}{v_g}\frac{\partial A'}{\partial \tau} &= \frac{1}{2}(1 - j\beta_c)gA' \\ \frac{\partial A}{\partial z} &= \frac{1}{2}(1 - j\beta_c)gA, \end{aligned} \tag{3.21}$$

where we have dropped the accents again. The same can be done for the gain equation 3.15, but as there is only a time derivative as in Equation 3.19, the equation stays the same.

Our architecture allows the assumption of coherent light, meaning that internally all signals are not characterized by real-valued powers but by complex amplitudes A. When solving the wave equation 3.21, it is interesting to separate the power $P(z,\tau) = |A|^2$ and phase $\phi(z,\tau) = \arg(A)$ in this complex representation $A = \left[\sqrt{P}\exp(i\phi)\right]$ of the slowly-varying envelope of the light. Inserting this

new notation into Equation 3.21 gives on the left and right side

$$\begin{aligned}
\frac{\partial A}{\partial z} &= \exp(j\phi)\frac{\partial}{\partial z}(P^{\frac{1}{2}}) + \sqrt{P}\frac{\partial}{\partial z}(\exp(j\phi)) \\
&= \exp(j\phi)\frac{1}{2}P^{-\frac{1}{2}}\frac{\partial P}{\partial z} + \sqrt{P}j\frac{\partial \phi}{\partial z}\exp(j\phi) \\
\frac{1}{2}(1-j\beta_c)gA &= \frac{1}{2}(1-j\beta_c)g\sqrt{P}\exp(j\phi) \\
&= \frac{1}{2}g\sqrt{P}\exp(j\phi) - \frac{1}{2}j\beta_c g\sqrt{P}\exp(j\phi). \qquad (3.22)
\end{aligned}$$

If we take the two sides together, we can drop the factor $\exp(j\phi)$ and when we group the real and imaginary parts of the equation together, we obtain the following two equations for the power and phase

$$\begin{aligned}
\frac{1}{2\sqrt{P}}\frac{\partial P}{\partial z} &= \frac{1}{2}g\sqrt{P} \\
\frac{\partial P}{\partial z} &= gP \qquad (3.23)
\end{aligned}$$

$$\begin{aligned}
\sqrt{P}\frac{\partial \phi}{\partial z} &= -\frac{1}{2}\beta_c g\sqrt{P} \\
\frac{\partial \phi}{\partial z} &= -\frac{1}{2}\beta_c g. \qquad (3.24)
\end{aligned}$$

Equation 3.24 shows that the influence of the linewidth enhancement factor β_c is manifested through the phase of the light. Equations 3.23 and 3.24 can be integrated over the amplifier length L yielding the following equations for calculating the output power and phase of the SOA [9]:

$$P_{out}(\tau) = P_{in}(\tau)\exp[\mathfrak{h}(\tau)] \qquad (3.25)$$

$$\phi_{out}(\tau) = \phi_{in}(\tau) - \frac{1}{2}\beta_c\mathfrak{h}(\tau), \qquad (3.26)$$

where the functions $P_{in}(\tau)$ and $\phi_{in}(\tau)$ are the power and phase of the input pulse, $\mathfrak{h}(\tau)$ is the gain $g(z,\tau)$ integrated over the length L of the SOA (3.27):

$$\mathfrak{h}(\tau) = \int_0^L g(z,\tau)dz. \qquad (3.27)$$

We can also integrate Equation 3.15 over the length, replacing first gP with $\frac{\partial P}{\partial z}$ (Equation 3.23)

$$\frac{d\mathfrak{h}}{d\tau} = \frac{g_0 L - \mathfrak{h}}{\tau_c} - \frac{P_{out}(\tau) - P_{in}(\tau)}{E_{sat}}, \qquad (3.28)$$

by then inserting Equation 3.25, we obtain an ordinary differential equation for $\mathfrak{h}$

$$\begin{aligned}
\frac{d\mathfrak{h}}{d\tau} &= \frac{g_0 L - \mathfrak{h}}{\tau_c} - \frac{P_{in}(\tau)\exp(\mathfrak{h}) - P_{in}(\tau)}{E_{sat}} \\
\frac{d\mathfrak{h}}{d\tau} &= \frac{g_0 L - \mathfrak{h}}{\tau_c} - \frac{P_{in}(\tau)}{E_{sat}}[\exp(\mathfrak{h}) - 1]. \qquad (3.29)
\end{aligned}$$

parameter	value	unit	description
β_c	5	-	linewidth enhancement factor
v_g	0.8×10^8	ms^{-1}	group velocity
n_g	3.75	-	group index
c	3×10^8	ms^{-1}	speed of light in vacuum
h	6.626×10^{-34}	m^2kgs^{-1}	Planck's constant
L	500×10^{-6}	m	amplifier length
H	2×10^{-6}	m	active layer thickness
W	0.2×10^{-6}	m	active layer width
V	200×10^{-18}	m^3	active layer volume
τ_c	300×10^{-12}	s	spontaneous carrier lifetime
$\hbar\omega_0$	1.28×10^{-19}	J	photon energy
q	1.6×10^{-19}	C	elementary charge
Γ	0.3	-	confinement factor
σ_g	2.7×10^{-20}	m^2	differential gain
E_{sat}	6.32×10^{-12}	J	saturation energy
P_{sat}	0.0211	W	saturation power
N_0	1×10^{24}	m^{-3}	carrier density for transparency
I_0	0.1067	A	current for transparency

Table 3.1: The parameters used in the SOA model and their value, unit and description [10].

So the two dominant effects, gain saturation and carrier dynamics, are captured by respectively E_{sat} and τ_c. Sometimes the saturation power P_{sat} is used instead of saturation energy and it is defined as E_{sat}/τ_c. The values typically used are shown in Table 3.1 and typically an input current I of 0.187 A was used, corresponding to a small-signal gain g_0 of $6075\,m^{-1}$. The carrier lifetime could fulfill a similar role as the leak rate η for ESNs (Equation 2.2), but as opposed to η, τ_c cannot be tuned freely as it is a fundamental property of any SOA.

The last rate equation 3.29 is solved by a fixed time step based fourth order Runge-Kutta method and, unless stated otherwise, the time step is 6.25 ps corresponding to a sample frequency of 160 GHz [11–13]. The time step for solving

the SOA model and sampling the input are always the same.

3.1.3 Nonlinear characteristics

The main SOA characteristic is the input–output behavior with gain saturation as shown in Figure 3.4 and it is incorporated in our model. This gain saturation is the basis for many nonlinear effects in SOAs such as Self-Phase Modulation (SPM), cross-Gain Modulation (XGM) and cross-Phase Modulation (XPM). XGM and XPM occur when two wavelengths enter the SOA simultaneously and one of them is stronger than the other. If the strong signal is modulated, it will saturate the gain when high and hence decrease it for the other signal. In XGM, the weak signal will then be modulated because the gain will vary over time depending on the strong signals's power. The weaker input signal's modulation will be opposite to the original's. XPM is due to the dependence of the refractive index of the active region of an SOA on the carrier density. An incoming modulated strong signal will deplete these carriers (gain saturation) and will therefore modulate the refractive index and hence cause a different phase shift for the weaker signal.

XPM and XGM require two signals with different intensities and at different wavelengths. In this dissertation, signals are either incoherent or at one wavelength so these two effects do not play a role.

SPM is the same mechanism as XPM, as there is a gain dependent phase shift (Equation 3.26) when the signal saturates the SOA gain, modulating its own phase. It also causes a spectral shift [9]. Although it is incorporated, it only starts to play a significant role for high values of the input current and gain (20–30 dB). Since we mostly work with an input current of 187 mA (13 dB gain), the influence of SPM will be rather limited.

3.1.4 ASE simulation model

Here, we extend our model to include the effects of *Amplified Spontaneous Emission* (ASE), which is the main source of noise in SOAs and comes from the spontaneous relaxation of excited carriers and subsequent emission of photons whose frequency, phase and direction are for the most part not matched to those of the incoming photons (Figure 3.2(c)). The total spontaneous emission power P_{sp} is given by [14]:

$$P_{sp} = n_{sp}(G-1)h\nu_0 B_0 \tag{3.30}$$

where h is Planck's constant, ν_0 the optical signal frequency, B_0 a bandwith centered on ν_0 and G is the SOA gain. n_{sp} is the spontaneous-emission factor and is given by [2]

$$n_{sp} = \frac{N_2}{N_2 - N_1}, \tag{3.31}$$

where N_1 and N_2 are the atomic populations for the ground and excited states, respectively (Figure 3.2). Since Equation 3.30 is not dependent on the power of the input signal, the relative importance of ASE can be diminished by using higher input powers.

Figure 3.6 illustrates the output spectrum of an SOA receiving light at wavelength λ_0 at its input. The input is amplified, but there remains a background of amplified light at other wavelengths, i.e., ASE. The power level of this ASE increases with the pump current, but if we keep that fixed, Figure 3.6 shows two ways to decrease the influence of ASE noise and improving the Signal to Noise Ratio (SNR). The first is to increase the output signal power by increasing the input signal power as this will not affect the ASE power as discussed before. This approach is limited by the gain saturation. A second approach is to limit the bandwidth of the ASE, since the total amount of ASE power is determined by the optical bandwidth B_0 (Equation 3.30). The broader the bandwidth, the higher the power. If we cut away a part of this spectrum with a bandpass filter with bandwidth B_0 around the signal frequency ν_0, the ASE power is reduced, increasing the SNR.

Although complicated models exist where ASE is modeled through separate rate equations (e.g., in [15]), we used a simpler model where random complex amplitudes are added to the SOA outputs. It is based on dividing the ASE spectrum with bandwidth B_0 from Figure 3.6 into bins with width $\delta\nu$, around a central frequency ν_0 or central wavelength λ_0. The spectral component corresponding to each bin has a certain power and random phase and is added to the output field of the SOA [16]. The number of bins is $2M+1 = B_0/\delta\nu$.

The field for the spontaneous emission added to the SOA output field is then described by the following equation:

$$E_{sp}(t) = \sum_{m=-M}^{M} \sqrt{n_{sp}(G-1)h\nu_0\delta\nu}\,\exp(j(m2\pi\delta\nu t + \Phi_m)) \tag{3.32}$$

Φ_m is a random phase for each bin's component of the spontaneous emission and its value is taken from a uniform random distribution with interval $[-\pi,\pi]$, the rest of the expression inside the exp is determined by the distance between the spectral component and the central frequency. The values typically used are $\nu_0 = 193.1\,\text{THz}$ and $n_{sp} = 2$. In our simulations we vary B_0, which can be done in practice by adding a filter after the SOAs, and kept the number of bins constant at 75.

3.1.5 Advanced models

The model we use captures the most important aspects of SOAs and is perfect to draw conclusions about the basic properties of SOA reservoirs. It does not,

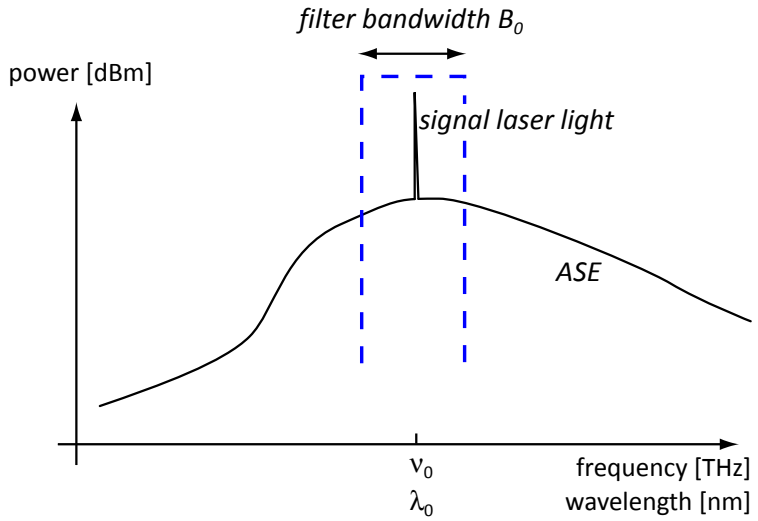


Figure 3.6: Drawing of the measured output spectrum of an SOA with a single laser input signal at wavelength λ_0. The sharp peak corresponds with the amplified input, superimposed on a background of amplified spontaneous emission. The blue dashed curve shows an ideal bandpass filter used to filter out the amplified input signal. The Y-axis has a logarithmic (dB) scale; the X-axis is in frequency or wavelength units.

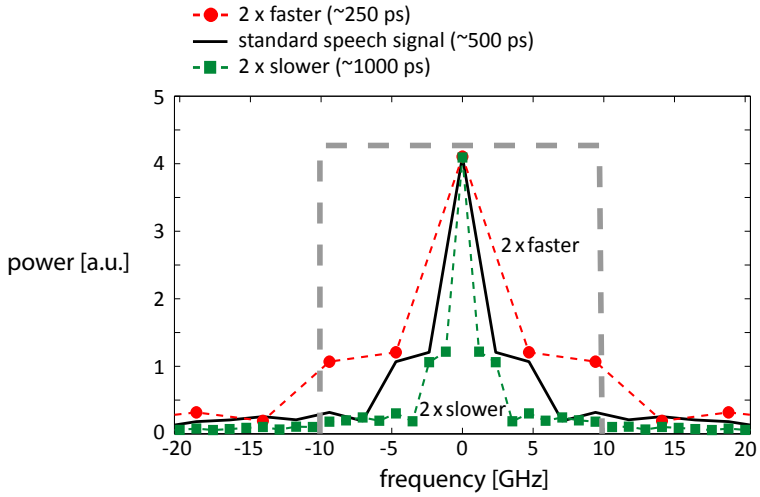


Figure 3.7: Spectrum of a typical audio signal used in Chapter 5 (solid black line). It falls within a 10 GHz basket (thick dashed gray line), where the model is sufficiently accurate and even for a faster signal this remains the case.

however, include all realistic effects. Although they were not investigated in the context of this work, it might be interesting to investigate them in the future, especially for signals at speeds higher than studied in this PhD thesis. A paper from R. Gutiérrez-Castrejón and M. Duelk gives a nice overview of different SOA models. They classify them depending on the effects they model in comparison to their own [15], which is unidirectional but takes all kinds of ultrafast nonlinear effects into account such as Carrier Heating (CH), Spectral Hole Burning (SBH) and Two-Photon Absorption (TPA). The carrier lifetime is not constant as in Agrawal's model. There are separate rate equations for the ASE (instead of just adding a random amplitude vector). They claim that ASE becomes important for SOAs longer than 500 μm, which is not the case in our simulations.

A first thing to note is that a lot of nonlinear effects only start playing a role at very high speeds, often expressed in terms of bit rates since most of the high-speed applications are in telecom where huge amounts of digital data are sent over optical fibers. A bit rate of 10 Gbit/s for example corresponds to pulse widths of 100 ps and a frequency content mainly within the 10 GHz range. The signal classification task of Chapter 4 uses signals with a repetition rate of only 0.5 GHz, while the frequency content of a typical audio signal of Chapter 5 is mostly within a 10 GHz range (thick gray dashed rectangle in Figure 3.7). In this range the Agrawal model remains valid. The same message that a lot of effects such as gain dispersion and Group-Velocity Dispersion (GVD) become relevant only for

pulse widths < 10 ps is also stated in the PhD thesis of Z. Li [17] and in Razaghi et al. [18]. Li's model incorporates ultrafast nonlinear effects such as two-photon absorption, free-carrier absorption, self-phase modulation, carrier heating, spectral and spatial hole burning. It also models the two orthogonal polarizations (TE and TM) instead of just the one polarization in Agrawal's model. Razaghi's model includes the same nonlinear effects as Li's only for one polarization but is bidirectional. This allows the investigation of reflections in the network or input from both sides [18, 19]. These models are an interesting starting point for future research on SOA reservoirs with high-speed signals. At the same time, it would be interesting to validate the conclusions from this work drawn from a simpler model. However, they are a lot more complex, so a trade-off has to be made between modeling accuracy and computational load.

3.2 Connectivity

In the previous part we have described SOAs, the optical nodes of our photonic reservoir. In this part we will discuss the architecture that connects them.

3.2.1 Connection weights

In our architecture, light can be represented by either power values or complex amplitudes. For the latter case, connections are modeled as complex weights instead of pure attenuators. This can be summarized in the complex representation of optical signals at the output of a connection:

$$A_{out}(t+\Delta t) = A_{in}(t)\exp\left[-\frac{1}{2}(\alpha_c+\alpha_d d)\right]\exp\left[j\frac{2\pi}{\lambda_0}n_{eff}d\right] \tag{3.33}$$

$$= A_{in}(t)\exp[-\frac{\alpha}{2}]\exp\left[j\Delta\Phi\right], \tag{3.34}$$

where λ_0 is the vacuum wavelength of the carrier signal, n_{eff} is the effective index — a number quantifying the phase delay per unit length in a waveguide, relative to the phase delay in vacuum. d is the physical length of the connection. α_c represents the fixed loss mechanisms (e.g., fan-out splitters, fan-in combiners and coupling losses) and $d\alpha_d$ the length dependent propagation loss in the connection[4]. Δt is the propagation delay of the connection, which is again proportional to its length. We see that the input–output relationship for a connection should now be modeled by a complex-valued connection weight

$$w = \exp[-\frac{\alpha}{2}]\exp\left[j\Delta\Phi\right], \tag{3.35}$$

[4]The factor 1/2 in the factor for the losses in Equations 3.33, 3.34 and 3.35 is due to the fact that the losses are defined for the power of the light.

and a delay Δt, whereas traditional ESN-connections are instantaneous and have real-valued connection weights. The physical length of the interconnections, affecting the loss α, phase change $\Delta\Phi$ and time delay Δt will turn out to be an important new optimization parameter for our architecture.

The fact that we are working with complex-valued signals and weights requires an extension of the definition of spectral radius to incorporate the effects of interference. We use the magnitude of the largest eigenvalue of the complex weight matrix, which consists of the complex interconnection weights multiplied by the square root of the maximal gain in the SOA's state curve (the tangent to this curve at zero input).

3.2.2 Connection topology

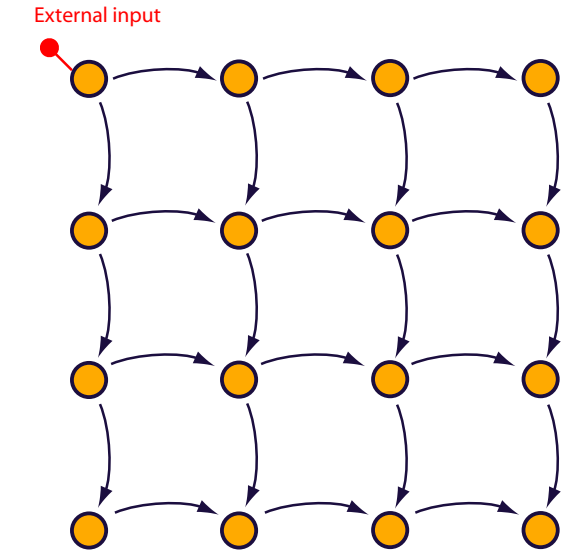
Besides the fact that photonic reservoirs can have complex interconnection weights, their topology itself is subject to some restrictions in contrast to simulated classical reservoirs. In classical reservoirs the connection topology is randomly initialized. The network can be sparsely or fully interconnected. For analog — but not spiking — networks the connectivity has little influence on the performance [20]. Photonic chips, and most other hardware implementations, are implemented on a planar surface. Furthermore, in contrast to, e.g., multilayer electronic VLSI implementations, interconnections on single-layer optical chips cannot cross without some losses (e.g., typically −0.16 dB in Silicon-On-Insulator (SOI) [21]). Therefore crossings should be avoided where possible.

Throughout this book, several topologies will be used, especially in Chapter 4 where the influence of the topology on the performance will be investigated. All the studied topologies start from a so-called *waterfall* topology (Figure 3.8(a), where the splitter and coupler losses at the in- and outputs of the SOAs are shown as well (Figure 3.8(b)). The input was always fed into a single node in the upper left corner as this would be easier for a hardware implementation. Several feedback mechanisms can be distinguished such as:

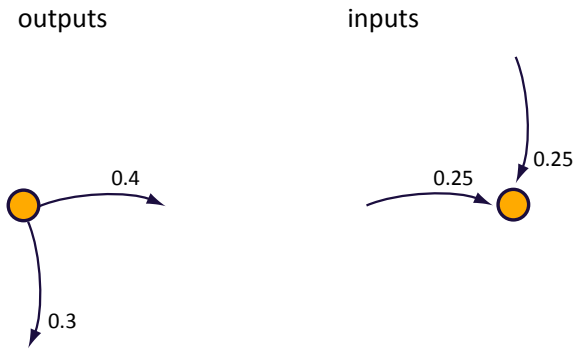
- *edge feedback*: the feedback connections are at the edges of the SOA grid, routed in such a way that they do not cross (Figure 3.9).

- *self feedback*: every SOA has one feedback connection to itself (Figure 3.10).

- *bidirectional feedback*: here the connections are between the same nodes as in the waterfall topology but also in the opposite direction (Figure 3.11). Although photonic waveguides are in essence always bidirectional, we use them in this work in an unidirectional way, since our SOA model is also unidirectional.



(a)



(b)

Figure 3.8: (a) 4×4 waterfall topology. (b) fixed loss mechanisms (α_c in Equation 3.33) in the topology because of couplers and splitters before and after every SOA. Their values (shown for intensities) were once chosen ad hoc and then not changed afterwards.

- *northwest feedback*: the feedback is between two close nodes in the rectangular grid, only if they are oriented in a northwest direction from each other (Figure 3.12).

Figure 3.13 shows the planar type of topology used in Chapter 5, which slightly differs from the previous ones. We refer to it as the *swirl* topology, because the connections are oriented as if they were in a whirling motion. As with the SOAs, we use all connections in a unidirectional way with an input and an output and we neglect the influence of backscattering and backreflection. For any number of nodes that can be arranged in a rectangular grid, the topology can be constructed as follows:

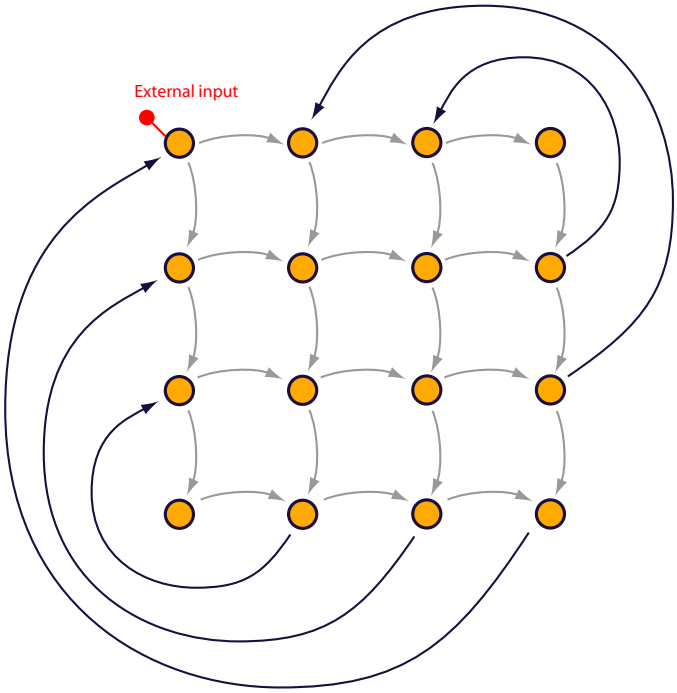
1. order the nodes in a rectangular grid and generate all nearest neighbor connections

2. all vertical connections on or to the left of the center get an upwards orientation, those to the right of the center get a downwards orientation

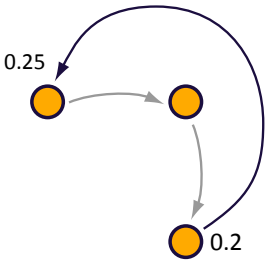
3. all horizontal connections on or above the center are oriented to the right, those below the center are oriented to the left

This topology has some advantages over the previous topologies as it only uses nearest neighbor interconnections, avoiding crossings altogether, while still providing a large number of feedback cycles. All interconnections can be implemented with comparable lengths and it can also be easily scaled to larger network sizes, without affecting interconnection lengths. This allows us to treat the interconnection length as a global optimization parameter. The area and design of concrete components needed for an integrated implementation of this topology heavily depend on the material system used[5]. Figure 3.14 shows the spectrum of the eigenvalues for a swirl and random topology. The higher degree of structure for the swirl topology in comparison to random topologies is also reflected in the eigenvalue spectrum.

[5]On chips, light is guided in waveguides which need a core that has a higher refractive index than the surrounding cladding layers. The index contrast of a material system determines the size of the waveguides and other components. The higher the index contrast, the smaller the components and the sharper the possible bends. We have ordered a chip with 12 SOAs (3×4 swirl configuration) on an InP platform (http://www.jeppix.eu/) with low index contrast. The size of this chip is 16 mm^2 with bending radii of 100 μm. However, in a material system with high index contrast, the dimensions shrink and bending radii down to 5 μm are possible with negligible loss (0.009 dB/90°) [22]. Amplifiers in this material system are still in an experimental phase [23], but when it has matured, this technology will offer tenfold chip size reductions or more. If further scaling is required, a transition to smaller and more low-power components such as coupled cavities is an option.



(a)



(b)

Figure 3.9: (a) 4×4 waterfall topology with feedback around the edges of the network. (b) fixed loss mechanisms (α_c in Equation 3.33) in the topology because of couplers and splitters before and after every SOA. Their values (shown for intensities) were once chosen ad hoc and then not changed afterwards. Only the weights for the edge feedback connections are shows as the other weights are the same as in Figure 3.8(b)).

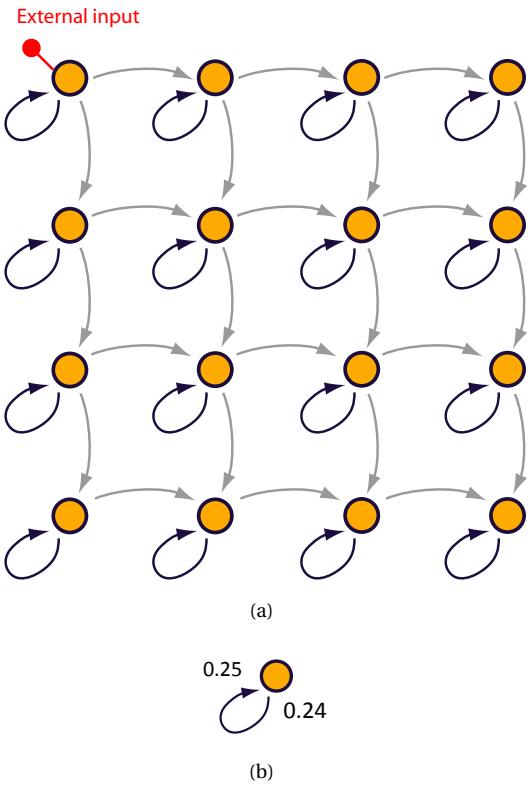


Figure 3.10: (a) 4×4 waterfall topology with feedback from every node to itself (*self feedback*). (b) fixed loss mechanisms (α_c in Equation 3.33) in the topology because of couplers and splitters before and after every SOA. Their values (shown for intensities) were once chosen ad hoc and then not changed afterwards. Only the weights for the self feedback connections are shown as the other weights are the same as in Figure 3.8(b)).

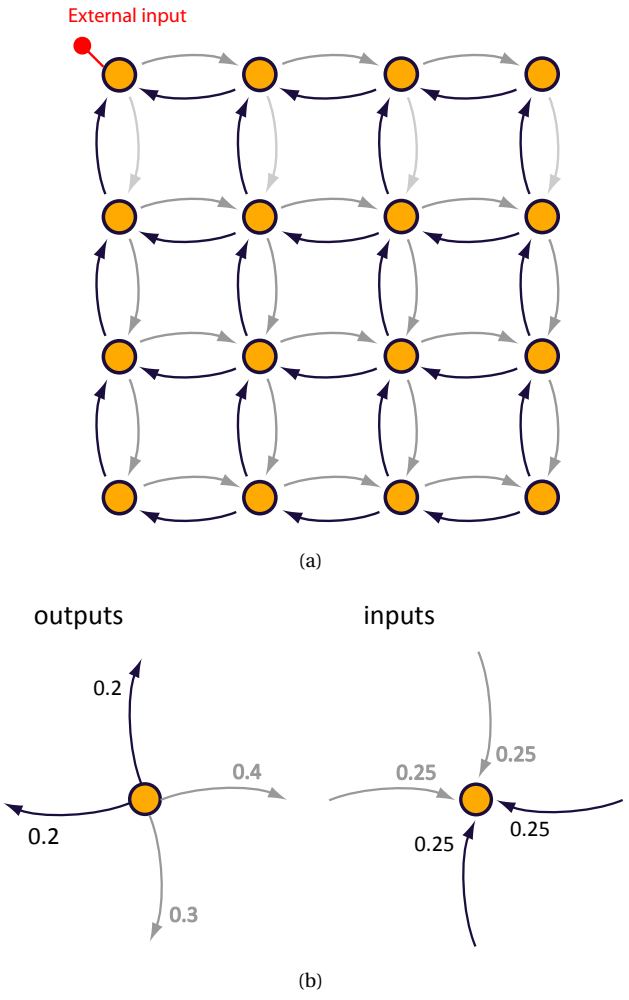


Figure 3.11: (a) 4×4 waterfall topology with connections also going in the opposite direction (*bidirectional feedback*). (b) fixed loss mechanisms (α_c in Equation 3.33) in the topology because of couplers and splitters before and after every SOA. Their values (shown for intensities) were once chosen ad hoc and then not changed afterwards. The weights for the waterfall connections are the same as in Figure 3.8(b)).

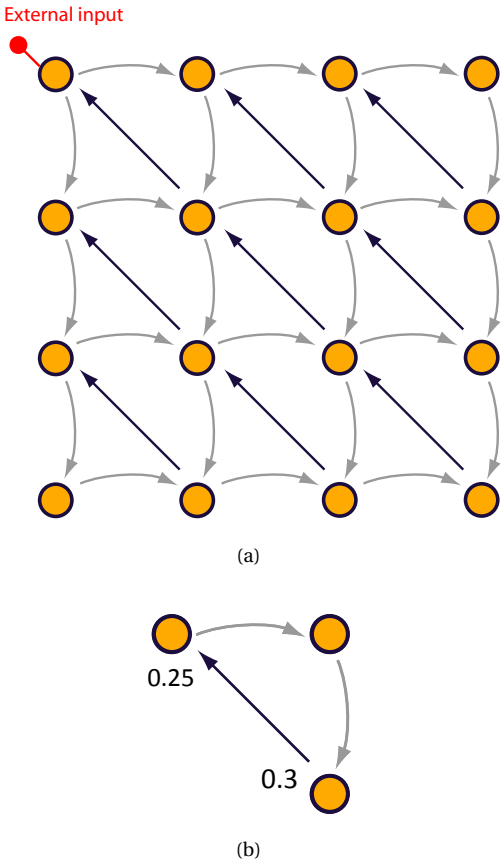
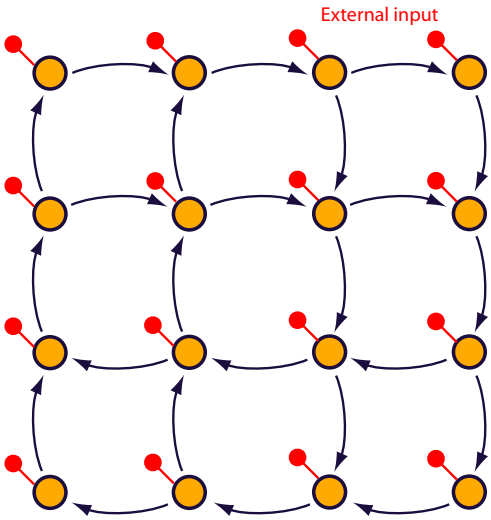
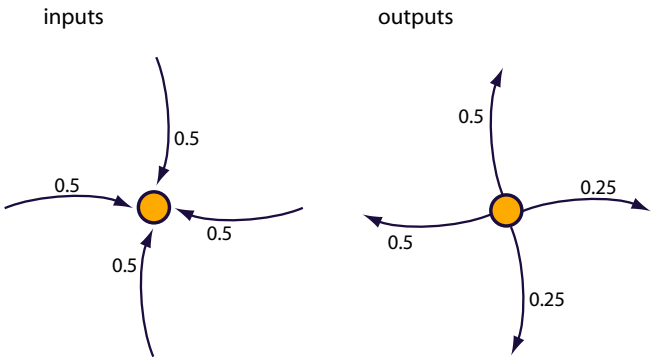


Figure 3.12: (a) 4×4 waterfall topology with feedback connections in a northwest direction(*northwest feedback*). (b) fixed loss mechanisms (α_c in Equation 3.33) in the topology because of couplers and splitters before and after every SOA. Their values (shown for intensities) were once chosen ad hoc and then not changed afterwards. Only the weights for the northwest feedback connections are shown as the other weights are the same as in Figure 3.8(b)).



(a)



(b)

Figure 3.13: (a) 4×4 swirl topology. (b) fixed loss mechanisms (α_c in Equation 3.33) in the topology because of couplers and splitters before and after every SOA. Their values (shown for intensities) were once chosen ad hoc and then not changed afterwards.

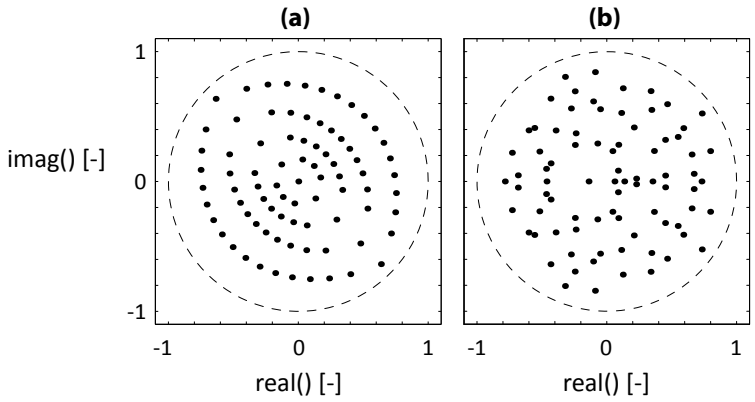


Figure 3.14: Eigenvalue spectrum of the interconnection matrix (including the node gain) of two topologies with a spectral radius of 0.9 and with 81 nodes. (a) 9×9 swirl topology. (b) random topology. The axes represent the real and imaginary parts of the eigenvalues, while the circles show the location of eigenvalues which amplitude equals unity.

3.2.3 Coherent network simulation

The simulation of coherent networks happens essentially in the same way as in Equation 2.2 for classical networks but with all weight matrices adapted to complex amplitudes and not power values. However, because the SOA model works with power and phase values and not directly with complex amplitudes, the inputs and outputs of the nodes are converted between the two representations when needed. This also happens for coherent tanh networks as the tanh function is applied to the signal's intensity.

At every time step two separate actions are performed. First, the inputs for all the nodes ($\mathbf{IN}_{nodes}$) are updated by means of

$$\mathbf{IN}_{nodes}[t+\Delta t] = \mathbf{W}_{in}\mathbf{u}[t] + \mathbf{W}_{res}\mathbf{x}[t - \text{interconnection delay}], \tag{3.36}$$

where the reservoir states $\mathbf{x}$ are taken as far back in time as the delay in the interconnections requires. All signals in this equation are in complex amplitudes and the weights are adapted accordingly. The interconnection delay is always an integer number of times the simulation time step Δt.

The second action is the conversion of these inputs to the corresponding power and phase values and these are used for updating the states of the nodes. This happens with Equations 3.25, 3.26 and 3.29 for SOAs. After the update, the new power and phase values are transferred back to complex amplitudes, which are needed for and used in Equation 3.36.

3.2.4 Limitations of the model

The previous sections described the model used in our photonic reservoir. This section discusses its limitations and the circumstances when it is valid and when it is not. The first problem with the model arises when the spectral radius is greater than unity, which can lead to an unstable and chaotic response, not only in classical reservoirs, but also in photonic reservoirs with our architecture, especially when you drive them far enough beyond the unity spectral radius. However, in our photonic reservoir losses occur in the couplers, splitters and connections, while gain is provided by the SOAs. When there is feedback in the network, we actually create the conditions for lasing when the gain is sufficiently high to compensate for all the losses, and this is only the case when the spectral radius is greater than unity. This lasing behavior is not incorporated in our model, i.e., no lasing action will come forth since only the amplification of the light is considered and not, e.g., gain clamping. In this regime, our numerical model runs into instabilities or chaos, and the results for such reservoirs are usually very bad, as is the case for classical reservoirs. This, however, does not need to be the case in reality, since in this regime different scenarios are possible. From strong input light inhibiting laser action [24], or one laser dominating others combined with chaotic behavior associated with delayed feedback, to different lasers interacting with each other. Such mechanisms can lead to all kinds of interesting nonlinear interactions such as chaos synchronization [25–29], and they could result in dynamics that are useful for reservoir computing. However, they are an interesting topic for future research and not considered here.

A second issue deals with the complex weights introduced in Section 3.2.1. In our simulations we model the connections as complex weights with a phase change and a loss (Equation 3.34). This is correct when working with light at one wavelength. However, when we modulate the light to transfer some information to it, the bandwidth of the laser light increases with its information content. Even if we have a laser spectrum with a peak of only a few kHz wide (possible with external cavity lasers), the width will increase if we add information to it with a certain bandwidth as in Figure 3.7. This means that different parts of the signal have different frequencies and correspond to different phase changes. This becomes relevant in our topologies with feedback that makes light interfere multiple times with itself.

This is a good time to validate our assumption that we can accurately model the signal by using monochromatic light. As an intuitive small-scale example of this multiple interference, we take the inner ring of the swirl topology with four nodes (Figure 3.15) and ignore the rest of the topology. Depending on the gain of the SOAs and losses in the couplers and connections, the light with original amplitude $A_0 = \sqrt{P_0}$ will interfere with amplitudes hA_0, h^2A_0, h^3A_0,... with $h = r\exp(j\Delta\Phi)$. r should be smaller than unity and from here on we will

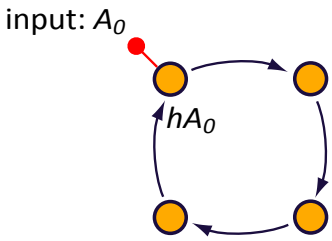


Figure 3.15: The basic loop from the swirl topology with four SOAs. If A is the amplitude of the incoming light, then the amplitude interfering with A is hA, with h a complex number.

replace r by the spectral radius ρ, because that is the highest gain in one or more feedback loops of the network (but a similar analysis holds for other loops with different values for r). The resulting intensity of all the interferences is [6]:

$$P = \frac{P_0}{(1-\rho)^2 + 4\rho\sin^2(\Delta\Phi/2)}, \tag{3.37}$$

where $\Delta\Phi$ is the same as in Equation 3.34. This is often written in another form:

$$P = \frac{P_{max}}{(1+(2\mathfrak{F}/\pi)^2\sin^2(\Delta\Phi/2)}, \tag{3.38}$$

where

$$P_{max} = \frac{P_0}{(1-\rho)^2} \tag{3.39}$$

and the parameter $\mathfrak{F}$

$$\mathfrak{F} = \frac{\pi\sqrt{\rho}}{(1-\rho)} \tag{3.40}$$

is called the *finesse*. Figure 3.16 shows the resulting intensity of the light in such a structure, depending on the phase difference $\Delta\Phi$ between the waves. When it is a multiple of 2π the waves interfere constructively and the transmission is high. The closer ρ is to unity, the sharper the peaks are and the higher P_{max} is. It is this $\Delta\Phi$ that will be varied in most simulations in Chapter 4 and 5. The situation where $\rho > 1$ is one where lasing action will occur as mentioned before.

Table 3.2 compares the phase difference $\Delta\varphi$ between different frequency components[6] for a bandwidth of 1 GHz and 10 GHz. The length of the ring is $\mathcal{N} \times 4 \times 460.8\,\mu$m with $460.8\,\mu$m the length of one connection; $n_{eff} = 2.38$, a

[6] We use a different symbol ($\Delta\varphi$) here to distinguish it from the phase difference $\Delta\Phi$ between multiple interferences at the same frequency introduced before.

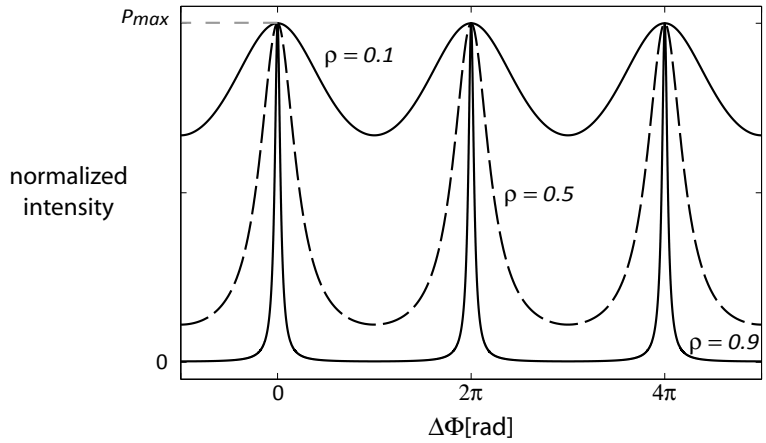


Figure 3.16: Interference of an infinite number of waves of progressively smaller amplitudes and equal phase differences. ρ determines the rate of decreasing amplitudes and $\Delta\Phi$ is the phase difference between the waves. Smaller spectral radii have less pronounced dips. The curves for different ρ are normalized with respect to P_{max}.

typical value for a waveguide on the Silicon-on-Insulator platform[7] and $\mathcal{N}$ is the number of times we take this length. In later chapters, the length of the interconnections will be varied and some typical values for the time delay Δt of one connection are shown in the table. For connections as long as a typical SOA ($\mathcal{N} = 1$) and the moderate bandwidth (1 GHz) we typically consider, the monochromatic approximation is still valid. However, a bandwidth of 10 GHz already starts to fall outside the peaks in Figure 3.16 when the spectral radius is close to unity. When we increase the total loop length, the approximation that we can just let beams interfere at one wavelength, is clearly not valid anymore. It is important that this issue is addressed in future research. To do this rigorously, the signal could be transformed to the spectral domain, where every frequency component would be multiplied by its own phase change (a function of λ in Equation 3.33), and then back to the temporal domain. The same could be done by applying a filter in the temporal domain [30]. Neither is however included in the architecture used in this work.

The influence will surely be a distortion of the signal, although it is hard to tell how severe and detrimental this will be, if there will be an effect at all. It will be less for smaller spectral radii (Figure 3.16) and signals with a smaller bandwidth

[7] The platform itself is not important and the exact values for the length and n_{eff} are not either. We have just chosen them to have an indication of the typical values involved and also so it would amount to the same time delay as a typical SOA of 500 μm

$\mathcal{N}$	Δt	bandwidth	bandwidth	$\Delta\varphi$
	ps	GHz	pm	rad
1	6.25	1	8	0.054π
		10	80	0.54π
8	50	1	8	0.43π
		10	80	4.3π
30	187.5	1	8	1.63π
		10	80	16.3π

Table 3.2: Phase difference $\Delta\varphi$ between different frequency components of the same optical signal, due to the bandwidth of the signal. Δt is the time delay of one interconnection, $\mathcal{N}$ is the number of times the length of the connections is increased; the bandwidth of the signal is shown both in frequency and wavelength units. For larger bandwidths and longer lengths, $\Delta\varphi$ starts to be large enough to alter between constructive and destructive interference within the bandwidth of the signal (Figure 3.16).

and shorter interconnections (Table 3.2). However, since interconnection delay will turn out to be an important parameter, staying in a safe zone might not be an interesting option from a reservoir performance point-of-view. The signals of the speech recognition task of Chapter 5 have a high bandwidth (Figure 3.7), but are also rather short in the configuration we used, so the interference at longer delays actually only happens a few times, instead of the infinite number of times assumed in this analysis.

The same analysis applies for the case of incoherent light as well, since multiple-beam interference will result in the frequency behavior of Figure 3.16, with some wavelengths becoming more prominent due to constructive interference. Using intensities is then maybe not allowed anymore to accurately describe the behavior of the reservoir.

A third aspect is that in realistic waveguides other effects such as dispersion, group-velocity dispersion and nonlinear effects can come into play, but usually only for high input powers and long connections not explored in our simulations. Dispersion is the phenomenon that different frequency components travel at a slightly different speed, which means that the refractive index is wavelength dependent. This is important for pulse propagation in telecom systems since it causes pulse broadening. The origin lies with the propagation constant β

depending on the refractive index n [2]:

$$\beta = n\frac{2\pi\nu}{c} = n\frac{\omega}{c}, \tag{3.41}$$

where c is the speed of light, ν is the frequency and ω the angular frequency. Since n is frequency dependent, and hence β also, it is often expanded in a Taylor series around the carrier frequency or wavelength (ω_0, ν_0 or λ_0) for pulses with a bandwidth $\Delta\omega \ll \omega_0$ [2]:

$$\beta(\omega) = \beta_0 + \beta_1(\Delta\omega) + \frac{\beta_2}{2}(\Delta\omega)^2 + \frac{\beta_3}{6}(\Delta\omega)^6. \tag{3.42}$$

$\beta_0 = \frac{2\pi}{\lambda_0} n_{eff}$ is the phase delay for the carrier frequency (already introduced in Equation 3.2 and equal to the phase factor of the complex weights in Equation 3.33), while $\beta_m = (d^m\beta/d^m\omega)_{\omega=\omega_0}$ are the higher order terms related to orders of dispersion. β_1 is the inverse of the group velocity v_g, while β_2 is related to the group-velocity dispersion. The equation for the propagation of a pulse in waveguides and optical fibers where $A(z,t)$ is the slowly varying amplitude is then [31]

$$\frac{\partial A}{\partial z} + \beta_1\frac{\partial A}{\partial t} + \frac{j\beta_2}{2}\frac{\partial^2 A}{\partial^2 t} - \frac{\beta_3}{6}\frac{\partial^3 A}{\partial^3 t} = 0 \tag{3.43}$$

and it resembles the equation for pulse propagation inside an SOA (Equation 3.4). Here, however, there is no amplification or loss included (the right hand side of the equation is therefore zero), and on the left hand side, terms to include dispersion effects have been added. In the absence of dispersion ($\beta_2 = \beta_3 = 0$), the optical pulse propagates with a speed $v_g = c/n_g$[8] without change in shape and we have investigated this ideal situation in our simulations.

When high powers are involved, nonlinear effects such as SPM, XPM and four-wave mixing can come into play. The effects of SPM and XPM can be modeled by modifying Equation 3.43 and adding some terms on the right side, which results in the so-called nonlinear Schrödinger equation, which has the form [31]

$$\frac{\partial A}{\partial z} + \beta_1\frac{\partial A}{\partial t} + \frac{j\beta_2}{2}\frac{\partial^2 A}{\partial^2 t} = -\frac{\alpha}{2}A - j\gamma|A|^2 A, \tag{3.44}$$

if we neglect the third order dispersion ($\beta_3 = 0$). The α term is related to the losses and the term on the right to the nonlinearities. This higher accuracy of modeling comes of course at a computational cost, while their effects can be minimal, especially when the powers are low and the distances short (this equation is often used to model km long fibers). In SOI waveguides for example the effects of SPM appear to be limited when powers are smaller than 30 mW [32], and the GVD only starts being a problem for 100 Gbit/s signals traveling in wires over 1 m in length [33]. Both situations were not encountered in our simulations.

[8] In the absence of dispersion the group velocity actually equals the phase velocity, or equivalently the group index equals the effective index $n_g = n_{eff}$

3.2.5 Coherent readout

Light detection is usually done with a simple photodectector that detects the intensity of the incoming light. In semiconductor photodiodes, an incoming photon creates, through the process of absorption (Figure 3.2(b)), a free carrier (electron-hole pair) and hence when a proper voltage is applied a current will flow, depending on the intensity of the incoming light. Therefore we supply the readout function of the reservoir with the light intensities and discard any kind of phase information. This has been standard practice in optical fiber communication, because the combination of WDM and EDFAs provided enough bandwidth. However, in recent years this has started to change and research is more and more geared towards coherent detection and more complex modulation schemes, as has been common in the world of radio communications for decades. Coherent detection captures not only the power but also the phase of the electromagnetic wave and different schemes exist. Usually this is done by letting the incoming light interfere with a local oscillator (a laser) that has the same or almost the same frequency as the incoming light (homodyne and heterodyne detection). An additional advantage is that the sensitivity of the receiver is enhanced, although this would only become relevant when working with very small powers. It would allow us to use a smaller fraction of the SOA output for the readout function. These advantages come at a higher complexity and demand on the components (e.g., the frequency of the lasers involved needs to be very stable). A lot more details can be found in [2], but they are not important for this work and the only thing that matters is that it is possible to retrieve phase information.

In Section 5.4.2 we will investigate what happens if we not only supply the readout with the power values, but also with the phase values. In this case the weights of the readout will still be real-valued, just as the phase and power values we supply. In the future it might be interesting to generalize the training to complex weights, working directly on the complex amplitudes of the light, an approach already successfully used with feedforward neural networks [34].

3.3 Conclusions

This chapter discussed the details of our photonic reservoir. First it described the SOA, which acts as a node in the reservoir and its most important features (gain saturation, carrier lifetime and gain dependent phase shift) and the corresponding model. In the second part we described the architecture that allows light to carry information. Both completely incoherent light and light at a single wavelength are supported.

We saw that the steady state characteristic of an SOA resembles a tanh, commonly used in traditional reservoir computing, but only for the upper branch,

since light is measured with intensities. Optical inputs will therefore be positive. However, when working at a single wavelength, the representation is done with complex signals and weights. This is different from classical real-valued reservoir computing and it also means that the spectral radius has to calculated for the complex weights to include interference effects. Another important difference between ESNs and our photonic reservoirs are the topology constraints and the fact that SOA nodes have their own dynamics. Due to these differences, it is not a priori obvious if SOA networks make for good reservoirs. We will study this in the next chapter, where an SOA reservoir will be used to solve a signal classification task.

References

[1] K. Huybrechts. *Digital Photonics Using Single Laser Diodes for All-Optical Network Nodes.* PhD thesis, Ghent University, 2010.

[2] G. P. Agrawal. *Fiber-Optic Communication Systems.* John Wiley & Sons, Inc., New York, NY, USA, 2002.

[3] G. Li. *Recent advances in coherent optical communication.* Advances in Optics and Photonics, 1(2):279–307, February 2009.

[4] E. Ip, A. P. T. Lau, D. J. F. Barros, and J. M. Kahn. *Coherent detection in optical fiber systems.* Optics Express, 16(2):753–791, January 2008.

[5] H. Sun, K.-T. Wu, and K. Roberts. *Real-time measurements of a 40 Gb/s coherent system.* Optics Express, 16(2):873–879, January 2008.

[6] B. E. A. Saleh and M. V. Teich. *Fundamentals of Photonics.* John Wiley & Sons, Inc., New York, NY, USA, 1991.

[7] P. De Heyn, B. Kuyken, D. Vermeulen, W. Bogaerts, and D. Van Thourhout. *High-performance low-loss silicon-on-insulator microring resonators using TM-polarized light.* In Proceedings of the Optical Fiber Communication Conference and Exposition (OFC) and The National Fiber Optic Engineers Conference (NFOEC), page OThV2, Los Angelos, USA, March 2011.

[8] W. Bogaerts, D. Taillaert, P. Dumon, D. Van Thourhout, and R. Baets. *A polarization-diversity wavelength duplexer circuit in silicon-on-insulator photonic wires.* Optics Express, 15(4):1567–1578, 2007.

[9] G. P. Agrawal and N. A. Olsson. *Self-Phase Modulation and Spectral Broadening of Optical Pulses in Semiconductor-Laser Amplifiers.* IEEE Journal of Quantum Electronics, 25(11):2297–2306, 1989.

[10] W. D'Oosterlinck. *Non-Linear Behaviour in a Semiconductor Optical Amplifier and Laser Diode Feedback Scheme.* PhD thesis, Ghent University, 2007.

[11] C. Runge. *Ueber die numerische Auflösung von Differentialgleichungen.* Mathematische Annalen, 46:167–178, 1895.

[12] M. W. Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen.* Zeitschrift für Mathematik und Physik, 46:435–453, 1901.

[13] J. C. Butcher. *Numerical methods for ordinary differential equations.* John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2008.

[14] A. Yariv. *Signal-to-noise considerations in fiber links with periodic or distributed optical amplification.* Optics Letters, 15(19):1064–1066, October 1990.

[15] R. Gutiérrez-Castrejón and M. Duelk. *Uni-Directional Time-Domain Bulk SOA Simulator Considering Carrier Depletion by Amplified Spontaneous Emission.* IEEE Journal of Quantum Electronics, 42(6):581–588, June 2006.

[16] N. A. Olsson. *Lightwave Systems with Optical Amplifiers.* Journal of Lightwave Technology, 7(7):1071–1082, 1989.

[17] Z. Li. *Ultrafast all-optical signal processing using semiconductor optical ampliers.* PhD thesis, Technische Universiteit Eindhoven, 2007.

[18] M. Razaghi, V. Ahmadi, and M. J. Connelly. *Comprehensive Finite-Difference Time-Dependent Beam Propagation Model of Counterpropagating Picosecond Pulses in a Semiconductor Optical Amplifier.* Journal of Lightwave Technology, 27(15):3162–3174, August 2009.

[19] S. Bischoff, A. Buxens, H. N. Poulsen, A. T. Clausen, and J. Mork. *Bidirectional four-wave mixing in semiconductor optical amplifiers: theory and experiment.* Journal of Lightwave Technology, 17(9):1617–1625, 1999.

[20] L. Büsing, B. Schrauwen, and R. Legenstein. *Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons.* Neural computation, 22(5):1272–1311, May 2010.

[21] W. Bogaerts, P. Dumon, D. Van Thourhout, and R. Baets. *Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides.* Optics Letters, 32(19):2801–2803, 2007.

[22] W. Bogaerts, S. K. Selvaraja, P. Dumon, J. Brouckaert, K. De Vos, D. Van Thourhout, and R. Baets. *Silicon-on-Insulator Spectral Filters Fabricated With CMOS Technology.* IEEE Journal of Selected Topics in Quantum Electronics, 16(1):33–44, 2010.

[23] M. Tassaert, S. Keyvaninia, D. Van Thourhout, W. M. J. Green, Y. Vlasov, and G. Roelkens. *A nanophotonic InP/InGaAlAs optical amplifier integrated on a Silicon-On-Insulator waveguide circuit.* In Proceedings of Information Photonics, Ottawa, Canada, May 2011.

[24] K. Huybrechts, G. Morthier, and R. Baets. *Fast all-optical flip-flop based on a single distributed feedback laser diode.* Optics Express, 16(15):11405–10, July 2008.

[25] V. Flunkert, O. D'Huys, J. Danckaert, I. Fischer, and E. Schöll. *Bubbling in delay-coupled lasers.* Physical Review E, 79(6):1–4, June 2009.

[26] R. Vicente, I. Fischer, and C. Mirasso. *Synchronization properties of three delay-coupled semiconductor lasers.* Physical Review E, 78(6):1–11, December 2008.

[27] M. C. Van der Sande, G.and Soriano, I. Fischer, and C. R. Mirasso. *Dynamics, correlation scaling, and synchronization behavior in rings of delay-coupled oscillators.* Physical Review E, 77(5):1–4, May 2008.

[28] O. D'Huys, R. Vicente, T. Erneux, J. Danckaert, and I. Fischer. *Synchronization properties of network motifs: influence of coupling delay and symmetry.* Chaos (Woodbury, N.Y.), 18(3):037116, September 2008.

[29] S. Sunada, T. Harayama, K. Arai, K. Yoshimura, P. Davis, K. Tsuzuki, and A. Uchida. *Chaos laser chips with delayed optical feedback using a passive ring waveguide.* Optics Express, 19(7):5713–5724, 2011.

[30] W. Li, W. P. Huang, and X. Li. *Digital Filter Approach for Simulation of a Complex Integrated Laser Diode Based on the Traveling-Wave Model.* IEEE Journal of Quantum Electronics, 40(5):473–480, May 2004.

[31] G. P. Agrawal. *Nonlinear Fiber Optics.* Academic Press, San Diego, CA, USA, 3rd edition, 2001.

[32] E. Dulkeith, Y. A. Vlasov, X. Chen, N. C. Panoiu, and R. M. Osgood. *Self-phase-modulation in submicron silicon-on-insulator photonic wires.* Optics Express, 14(12):5524–5534, June 2006.

[33] E. Dulkeith, F. Xia, L. Schares, W. M. Green, L. Sekaric, and Y.A . Vlasov. *Group index and group velocity dispersion in silicon-on-insulator photonic wires.* Optics Express, 14(9):3853–3863, May 2006.

[34] H. G. Zimmermann, A. Minin, and V. Kusherbaeva. *Comparison of the Complex Valued and Real Valued Neural Networks Trained with Gradient Descent and Random Search Algorithms.* In M. Verleysen, editor, Proceedings of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, April 2011.

4

Signal classification

The previous chapter revealed that photonic reservoirs differ from classical implementations in a number of ways: a planar chip limits interconnection freedom, making random networks difficult to implement; light at one wavelength is represented by complex amplitudes in contrast with real valued signals for classical reservoirs and SOA nodes have dynamics not present in static tanh nodes. By using a simple classification task we will show that in spite of these differences a very good performance can be achieved [1]. In the second part of this chapter we show that the optimum can be quite sensitive to noise and we discuss several ways to improve the noise robustness such as larger sizes, different topologies and longer interconnection delays. In the last part of this chapter we replace the SOAs in our architecture with leaky integrator nodes. Their superior performance is rivaled by SOA networks when the optimal value of the interconnection delay is used.

The work in this chapter and on this task will give the insights about what differences really matter between SOA and tanh reservoirs, especially the crucial factor of interconnection delay. However, an exhaustive investigation is done in the next chapter with a much more difficult task: isolated digit recognition.

4.1 Signal classification task

The signal classification task used throughout this chapter is actually a signal template classification task, where two 'template' timeseries need to be distinguished from each other. The two templates are a square and a triangular waveform (Figure 4.1(a)). If the input signal changes, the output should follow as fast as possible and hence the Error Rate (ER) is defined as the fraction of the total signal time the reservoir misclassifies. Without noise this task is actually quite easy and explicitly programmed solutions can be designed. Although it becomes a lot harder with noise, it remains a proof-of-concept task. To make it easier to identify the most important parameters, noise was usually not added in this chapter. This means that the absolute ER is not important and only relative differences between different architectures matter. However, the noise robustness will be investigated in Section 4.2.

Figure 4.1 shows an example of the different steps in the training phase of a reservoir. In this case we use a 5×5 SOA reservoir with the topology from Figure 3.9 but the steps are generic. Figure 4.1(a) portrays the input changing at random moments from a square to a triangular wave and back. (b) has the corresponding desired output (dashed, −1 for a rectangular wave, +1 for a triangular wave). (c) shows the response to the input from (a) of some of the 25 SOAs. Since the input is fed into one node only, this node has a larger response than all the other nodes. $\mathbf{W}_{in}$ from Equation 2.2 is in this case nothing more than a scaling factor determining the (peak) intensity of the rectangular and triangular input into this one node. In all the experiments in this chapter this amplitude was 1.25 mW, in order to use the nonlinearity of the steady state curve of the SOA (Figure 3.4). (d) shows the result of the readout function, which is a linear combination of the different reservoir states, closely following the desired output (dashed line). Note that the weights of this readout can be negative and in such a way negative outputs can be realized. We envisage this training and readout in a first stage to be done off-line by an electronic chip. (e) has the final output of the system, obtained by applying a sign function to the result of the linear combination to map every output to +1 or −1. In this example the system manages to generate the desired output most of the time, barring some discrete spikes.

For the actual classification we use ridge regression to avoid over-fitting and fivefold cross-validation to make our results more robust [2]. The general approach was already discussed in Chapter 2 (Section 2.4), but here we will outline it in more detail for the signal classification task, based on Figure 4.2. We begin by making a dataset with a number of different samples (ten in our case) and their desired output. The samples each have around ten signal transitions over a signal length of 100 ns, but at different random instances. As the weights of

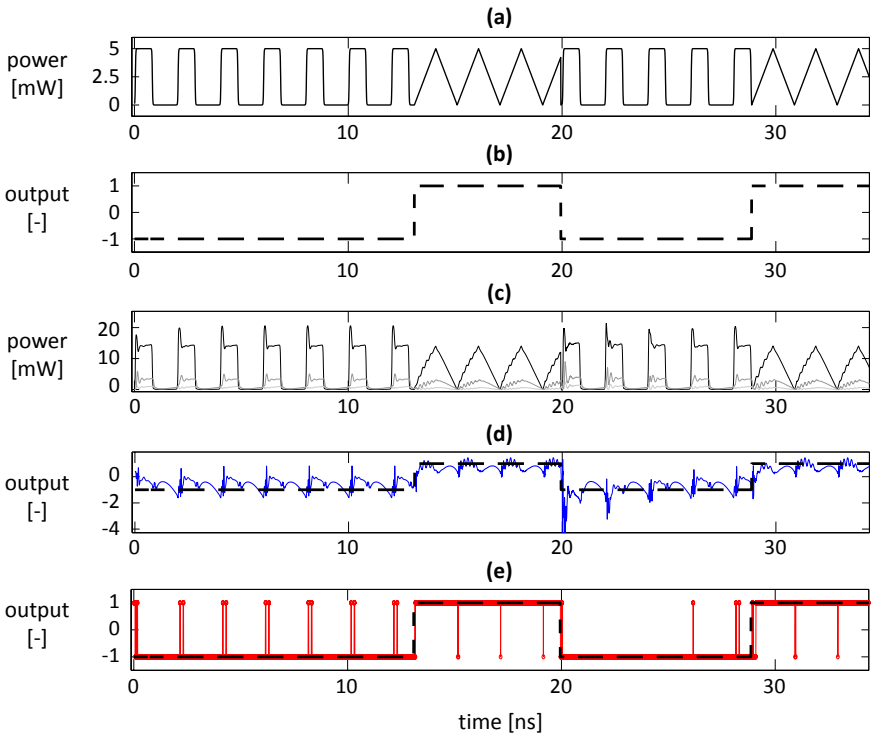


Figure 4.1: Signal classification task: (a) Input signal with different transitions between the rectangular and triangular waveform; (b) desired output (-1 for a rectangular wave, $+1$ for a triangular wave); (c) state (i.e., optical power level) of some of the reservoir nodes; (d) the approximation (blue) of the desired output (black) by the readout function; (e) final output of the system (red)

the readout function are time-invariant, a good reservoir should be able to handle transitions at any moment. In fivefold cross validation the set of samples is shuffled five times in a random way (in the remainder of the next two paragraphs the exact numbers used in our case are put between brackets). Each time the total set is divided into a training set (8 samples) and a test set (2 samples). After training on the corresponding subset and finding the optimal weights for the readout function, the weights are kept fixed and the reservoir is tested and validated on the test set. The final performance or error rate is the average test performance over the five folds (Figure 2.5).

Ridge regression adds a penalty term to the optimization problem: the norm of the weights multiplied with a regularization parameter γ (Equation 2.9). To find the optimal value of γ an iterative approach is used: first an initial logarithmically spaced array is defined with values for this parameter. For each of these values the training subset (8 samples) is again divided into two subsets: one for training (train_γ = 7 samples) and one for testing (test_γ = 1 sample). This as well is repeated five times, as often as there are folds, and the result for each regularization parameter is the average performance on the test_γ sets over all the folds (Figure 2.4). The regularization parameter that achieves the lowest average performance is selected[1] for the training and testing of the original sets.

4.1.1 Spectral radius

In classical tanh reservoirs with random topology, the most important parameter is the spectral radius ρ (Section 2.3.3), since it is a good measure for the dynamical regime of the reservoir. The spectral radius is the largest eigenvalue of the interconnection matrix containing all the gain and loss in the network. The weights in the interconnection matrix are real when the signals themselves are real-valued, but complex when working with a complex representation, such as for coherent light, to incorporate interference effects (Section 3.2.1). For linear networks, a spectral radius greater than unity results in the reservoir state growing exponentially. The nonlinear part of the tanh quenches the effective gain, and therefore the edge of stability occurs for spectral radii slightly larger than unity if the input is biased to a working point different from zero[2]. This can be seen in Figure 4.3(a), where the optimal regions (darkest) are around $\rho = 1$. The region with the better performance is somewhere between 0.8 and 1.05. For very small spectral radii and spectral radii greater than 1.1 the results degrade severely. This confirms the notion in reservoir computing that the optimal performance is

[1] Since this process is iterative, the selection of an array of regularization parameters can be repeated a few times. A new array will then have values closer around the optimal value from the previous array. In our simulations this defining of new arrays was done two times. Ideally as the value of the regularization parameter converges to its optimal value, the results should converge as well.

[2] A lower effective gain results in a lower effective spectral radius.

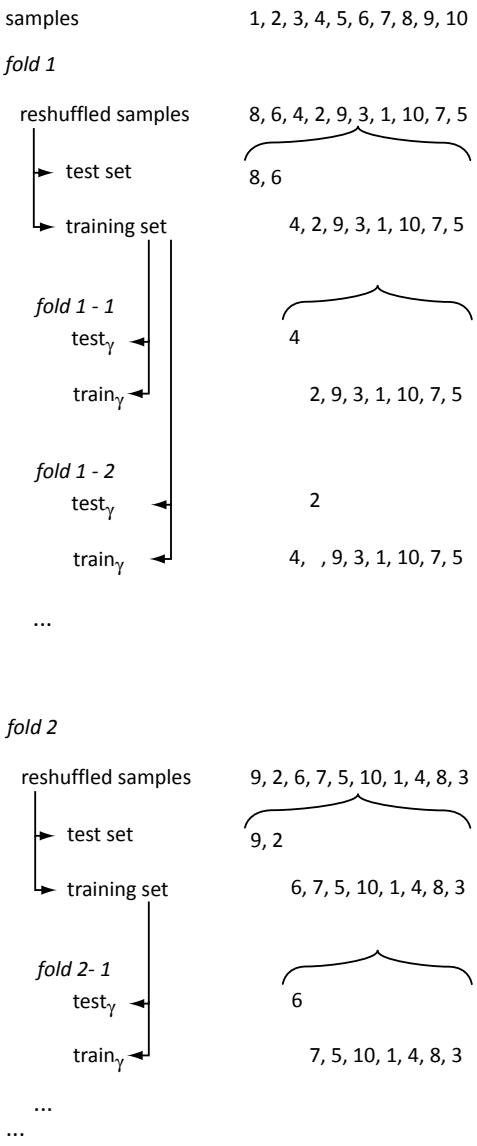


Figure 4.2: This shows how the 10 samples for the signal classification task are used during training and testing with cross-validation and ridge regression as shown in Figures 2.5 and 2.4. For the 5 folds, the samples are reshuffled in a random way and then divided in a training (8 samples) and test set (2 samples). To find the optimal regularization parameter, the training set is divided again in a train$_\gamma$ set (7 samples) and a test$_\gamma$ set (1 sample), also 5 times

usually achieved for spectral radii around unity [3].

Since the spectral radius characterizes the amount of gain or loss in the network, it can be altered by all mechanisms that influence them. Some of these mechanisms are usually fixed when making and designing the reservoir (interconnection loss, splitter and coupler ratios,...), while others can be changed during operation, such as the gain of the SOAs by varying the input current. It is conceptually interesting to change the loss in the connections because it is independent from any phase change. In contrast, the SOA input current and its impact on the gain are not orthogonal to the phase change since they are linked through Equation 3.26.

4.1.2 Phase

When working with coherent light and the corresponding complex representation of optical signals, the phase itself becomes important. We will illustrate the influence of phase by looking at two examples: one where the phase is changed in all interconnections, and one where it is changed only in certain connections. In Figure 4.3(a) we have used a waterfall topology with some feedback connections around the edges, from here onwards referred to as *edgeFB* (Figure 3.9). If, just for the purpose of gaining some conceptual insight, we assume that all the connections have equal length, we could change the phase $\Delta\Phi$ in every connection through the wavelength and hence the phase change would be the same for all connections (Equation 3.33). This means that the same phase term occurs for every nonzero element of the interconnection matrix $\mathbf{C}$ and hence that it can be written outside the matrix as a factor: $\mathbf{C} = \exp(j\Delta\Phi) \cdot \mathbf{C}'$. This means that the spectral radius as in Equation 2.3 is not influenced by the phase because its absolute value is unity ($\left|\exp(j\Delta\Phi)\right| = 1$). This is the reason why the edge of stability in Figure 4.3(a) is almost parallel to the phase axis. However, it is clear that the phase is important since the optimal regions are only present at some phases.

As a second example we have a reservoir where the phase was not changed in all the connections, but only in the dashed ones marked in Figure 4.5(a). In this case the phase change term cannot be isolated from the interconnection matrix and the spectral radius changes, not only because of different attenuations in the connections (X-axis), but also due to the phase change (Y-axis) as shown by the thick solid line marking places where the spectral radius equals one. Interestingly the line follows closely the edge between the regions with very good and very bad performance, typical for the edge of stability (Figure 4.4). This means that the spectral radius is still valid when working with complex amplitudes, but also that the spectral radius does not tell the whole story since the performance varies along the edge of stability (Figure 4.3). Also note that in this case we cannot

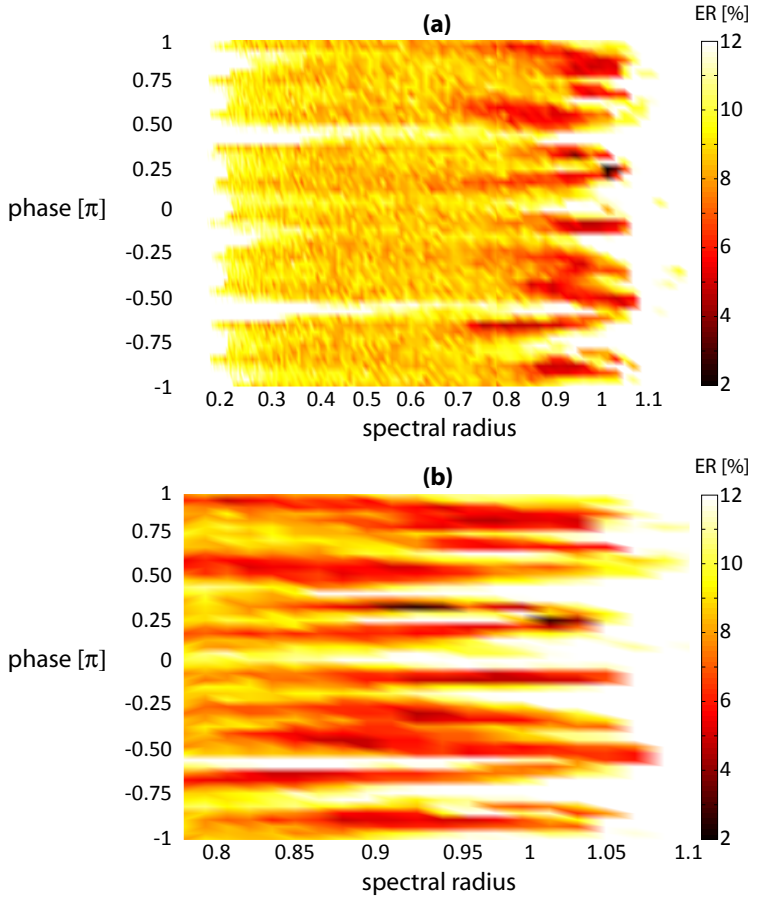


Figure 4.3: Performance of a coherent SOA network with a waterfall topology and feedback connections on the edges: (a) there are two tiny regions in $(\rho, \Delta\Phi)$ space with a very good performance. When the spectral radius is greater than 1.1 or very small, the performance deteriorates strongly; (b) the same plot but zoomed in at the region with the best performance.

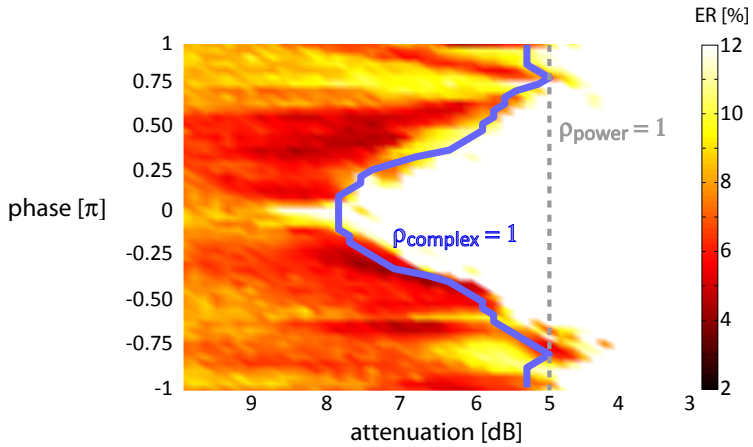


Figure 4.4: Performance of a coherent SOA network with a waterfall topology and feedback connections on the edges where the phase is only changed in certain connections of the waterfall network (the dashed ones marked in Figure 4.5(a)). The line is marked where the spectral radius of the complex weights $\rho_{complex}$ equals unity and it corresponds very well to the border between optimal and bad performance. The spectral radius calculated with the weights for the powers ρ_{power} and hence without phase information, is marked with a thick dashed line. For constructive interference (phase = 0) the difference between $\rho_{complex}$ and ρ_{power} is roughly 3 dB.

interchange the loss parameter that was actually swept (the attenuation in the connections) with the spectral radius like we could do in Figures 4.3(a) and (b).

When we look more closely at Figure 4.5(a), we can see that this actually contains a large number of Mach-Zehnder Interferometer (MZI) structures such as in Figure 4.5(b) where only one of the arms undergoes a phase change. This structure can be tuned from constructive interference to destructive interference and this determines the shape of the $\rho_{complex}$ line in Figure 4.4. When the interference is constructive (phase = 0), the losses in the system are minimal and hence a higher loss is needed (8 dB versus 5 dB) in the network for the same dynamical regime. The spectral radius can also be calculated from the power ratios of the weights (ρ_{power}), neglecting any influence of phase and interference. The dashed line in Figure 4.4 shows the corresponding ρ_{power} and it shows its inadequateness for identifying the dynamical regime. The MZI structure is the reason for the roughly 3 dB difference in attenuation between $\rho_{power} = 1$ and $\rho_{complex} = 1$ for constructive interference (phase = 0). When two waves with intensity I_0 interfere with each other, the resulting intensity depending on their phase difference will be between 0 (destructive) and $4I_0$ (constructive), while it will be $2I_0$ for incoherent light[3], a difference of a factor 2 or 3 dB. In our case the weights are not entirely symmetric (Figure 3.8(b)), there are SOAs in both arms that give a slightly different gain and phase change depending on the input power; and there is feedback which makes that for different topologies and weight distributions this intuitive reasoning no longer completely holds.

4.1.3 Resolution

An issue that was not addressed so far is the resolution used in the simulations. Even though we are simulating an analog network, we are sampling when calculating the network. In our case we use a fixed timestep solver, which means that we sample all our signals with a certain sample rate. We usually used 160 GHz (which means a sample period of 6.25 ps), corresponding to the time lights needs to travel through an SOA of 500 μm. When we double the sample frequency to 320 GHz but keep the delay between nodes the same (6.25 ps), we get the result of Figure 4.6(a). Comparing it with the result for the sample frequency normally

[3] When two waves with the same carrier frequency ν_0 or vacuum wavelength λ_o interfere with intensities I_1 and I_2, the resulting intensity is $I = I_1 + I_2 + 2\sqrt{I_1 I_2}\cos(\Delta\Phi)$ with $\Delta\Phi$ the phase difference between the two [4]. Now if we look at Figure 4.5(b), neglect the SOAs in both arms, assume that all the weights for the splitters and couplers are symmetrical and 1/2 for the intensities (the weights for the complex amplitudes are then $1/\sqrt{2}$), and take the intensity after the first SOA to be I_0, the two waves that will interfere at the input of the right SOA are $I_1 = I_2 = \frac{1}{2}\frac{1}{2}I_0 = 1/4 I_0$. Therefore the resulting intensity will be $I = 1/4 I_0 + 1/4 I_0 + 2\sqrt{(1/4 I_0)^2}\cos(\Delta\Phi)$ or $I = 1/2 I_0 + 1/2 I_0)\cos(\Delta\Phi)$ and depending on the phase difference $\Delta\Phi$ between the two waves the intensity will vary between 0 and I_0. When the light is incoherent, the phase relation is averaged out and the intensities are just combined into $I = 1/2 I_0$.

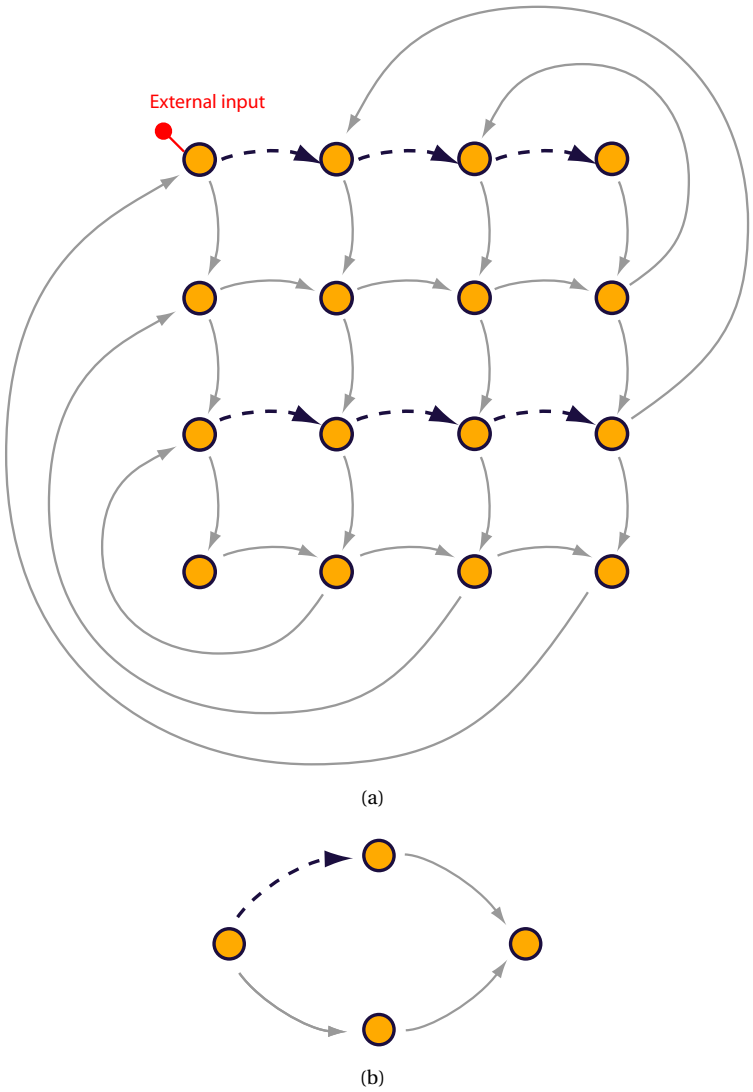


Figure 4.5: (a) The same topology as in Figure 3.9 but the connections which phase is changed are marked with a thicker dashed arrow (the horizontal lines on the odd rows). (b) Different view on a square block of 4 nodes of the waterfall topology, showing that changing the phase as in (a) actually corresponds to tuning only one arm of a Mach-Zehnder interferometer structure.

used (Figure 4.3(b)) we see that their performance topography are almost the same. Some minor variations exist due to the randomizations of sample reshuffling and transition instances, but the location of different regions and their boundaries correspond very well. Therefore we have used a sample frequency of 160 GHz for almost all simulations in this chapter and the next, unless stated otherwise.

When investigating the performance of a reservoir over two parameters (e.g., spectral radius and phase change as in Figure 4.3(a)), a certain resolution is chosen for the scanned parameters. Figure 4.6(b) shows that the boundaries in the vertical direction become sharper, when the resolution for the X-axis (attenuation in the connections and hence the spectral radius that corresponds with it) is increased fivefold. However, as with increasing the sample rate, the performance topography with its regions and boundaries are the same. The same is true for a fivefold increase of the Y-axis or phase axis resolution (Figure 4.6(c)). The regions are now a lot sharper in the horizontal direction. One example is the left one of the two optimal dark regions that only exists for a very narrow band of phase values (indicated with a dashed arrow). This means that it could be possible to miss such a narrow optimal region when we do not use a sufficiently high sampling of the X and Y-axis. The next section discusses the robustness of such optima.

4.2 Noise robustness

This section will show that it is important to take the robustness to noise into account, since it severely degrades the performance when the optimal region in $(\rho, \Delta\Phi)$ space is small. First, the rapid degradation in performance for a small optimum in $(\rho, \Delta\Phi)$ space due to noise will be discussed, followed by several approaches to improve the noise robustness.

As argued in the PhD thesis of D.Verstraeten [5], the sigclass task is a simple task if no noise is added and therefore it is important to see how the performance is affected by noise on the input signal. We take the $(\rho, \Delta\Phi)$ point corresponding to the best performance, assuming we have perfect phase control, and let it run 1000 times with each run a different random shuffle of the samples with added noise (Figure 4.2) and different random transitions of the two waveforms in the input signals. The line with circles in Figure 4.7 shows how the performance of the topology with edge feedback (Figure 4.3(b)) suffers greatly from the addition of noise. Therefore, there is no need to increase the resolution of the scanned parameters further and find ever smaller optimal regions as investigated in the previous section. The resolution used in the rest of this chapter and the next will therefore be the same as the one in Figure 4.3 (sampling: 160 GHz; phase: 53

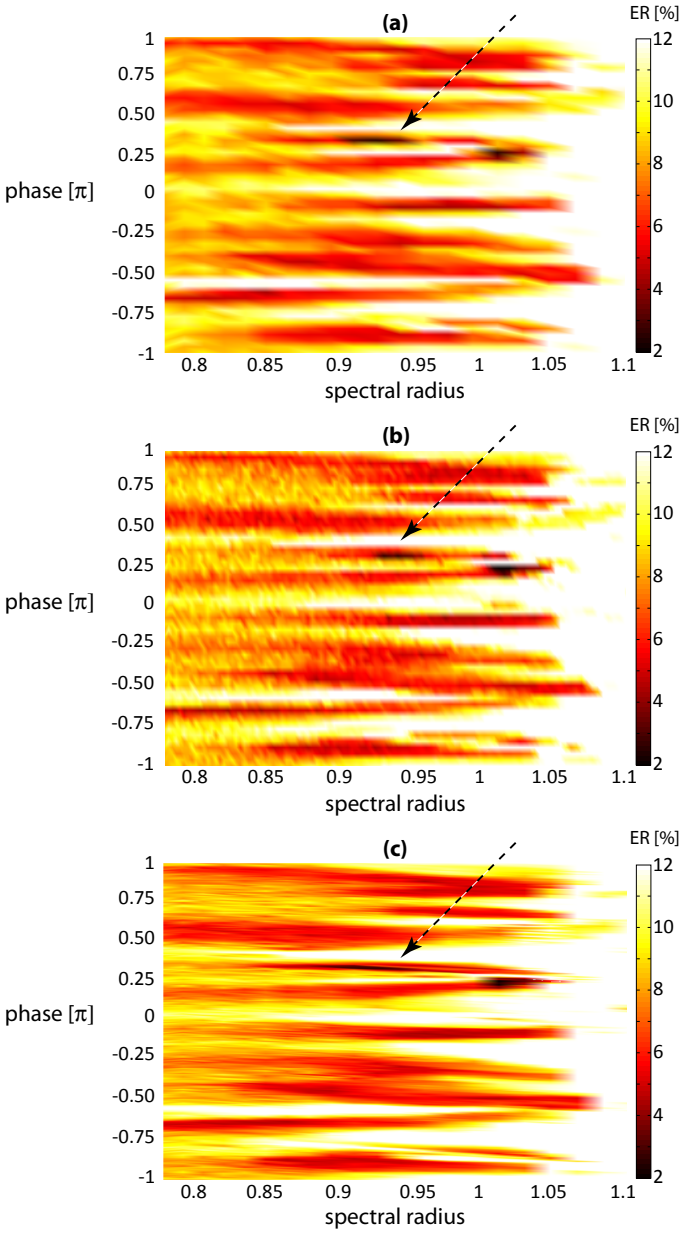


Figure 4.6: Three ways of increasing different aspects of the simulation resolution for the same edgeFB SOA network with an interconnection delay of 6.25 ps: (a) doubling the sample frequency from 160 GHz to 320 GHz; (b) increasing the resolution for the X-axis fivefold; (c) increasing the resolution for the Y-axis fivefold.

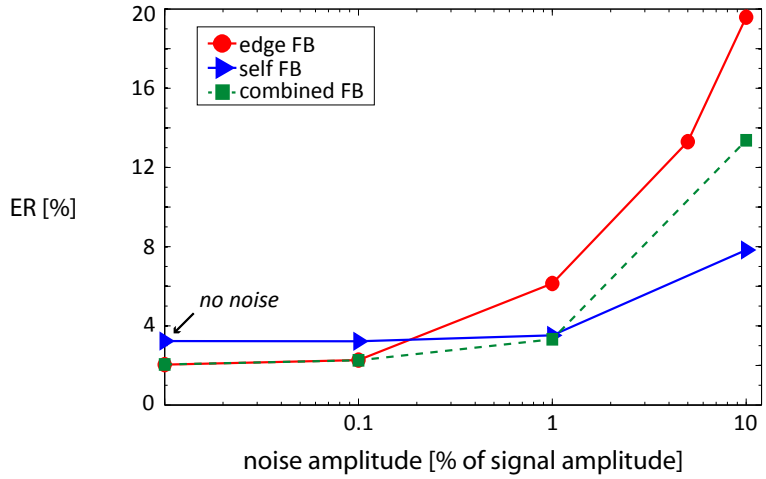


Figure 4.7: Different noise levels (the noise amplitude is shown as a percentage of the signal amplitude) for different topologies. For each topology the $(\rho, \Delta\Phi)$ point was chosen with the optimal performance. The topology with edge feedback is clearly the most sensitive to noise and using a topology with a larger optimal region in $(\rho, \Delta\Phi)$ space such as a network with self feedback (Figure 4.8) improves the robustness.

points between $[-\pi, \pi]$; attenuation spacing: 0.15 dB[4]).

The small optimal region was for a network with a very short interconnection delay (6.25 ps), a waterfall topology with feedback around the edges of the network and a size of 25 nodes in a square 5×5 configuration. Since small optimal regions are not so useful, it seems important to look for a reservoir that shows an optimal region for a wide range of $(\rho, \Delta\Phi)$ values and the next sections discuss several ways — different topologies, longer interconnection delays and a larger size — to make coherent SOA reservoirs more robust.

4.2.1 Topology

The waterfall topology with feedback around the edges of the network was proposed because it is easier from an integration point of view, since it avoids having many crossings. However, other topologies are possible and some of them are shown in Figures 3.10, 3.11 and 3.12. The delay is chosen at its standard value of 6.25 ps. Figure 4.8(a) shows the performance of a network with a waterfall topology but feedback from every node to itself (referred to as self feedback:

[4] In the next chapter the attenuation spacing will usually be chosen such that the corresponding spectral radius values are equidistant. The sampling frequency and phase resolution, however, remain the same as stated here.

Figure 3.10). Two things stand out: first is that the size of the optimal region is larger than for the network with edge feedback (Figure 4.3(a)); second is that the optimal region for this kind of topology is not around a spectral radius of unity[5]. That this kind of topology is more robust to noise was already shown in Figure 4.7. One loses some performance at very low noise levels, but the degradation is a lot slower for increasing noise levels.

Figure 4.8(b) shows the performance of a network where different feedback mechanisms are combined (Figure 4.9). The size of its optimal region is somewhere in between that of a network with self feedback (Figure 4.8(a)) and edge feedback (Figure 4.3(a)). This is also reflected in the summary of Figure 4.7 where the performance of a network with all feedback combined is between that of a network with only self or edge feedback. Although it is interesting that topology exerts influence on performance, we will now see that its benefits are marginal in comparison to the impact of interconnection delay.

4.2.2 Delay

All the experiments till now used a topology where all connections introduced an equal time delay of 6.25 ps, which is also the duration for light to propagate through the 500μm SOAs. This situation is very similar to classical tanh reservoirs without delay and where information is transferred every simulation step from one node to the next. We keep the restriction that all connections have the same length, but now globally increase their length. Figure 4.10 shows the influence on the performance. Initially it makes for larger optimal regions (dark), and at a delay of 56.25 ps, the optimal region comprises a very large portion of the $(\rho, \Delta\Phi)$ space. If we further increase the delay, the size of the optimal region remains, but its performance deteriorates (Figure 4.10(c)). We can conclude that there exists an optimal value for the delay for this task, extending the very small optimal region of the network with short delay (Figure 4.3), while keeping the same optimal performance. Increasing the delay further, the stability of performance is maintained but the optimal performance is lost. The enlargement of the optimal region, especially in phase space could be because for longer delays, the light interferes less often and hence the effects of interference of an infinite number of waves of progressively smaller amplitudes and equal phase differences as discussed in Section 3.2.4 and Figure 3.16, are less pronounced.

[5] This is actually the only topology with short delays that did not have an optimal region for spectral radii around unity (networks with longer delays, however, (Figure 4.10) also have optimal regions at smaller spectral radii). We did not investigate this and no proper explanation is readily available. We do know that the spectral radius is zero for a pure waterfall topology (it has no feedback) and therefore can not measure its dynamics; and that it works well for networks with some global feedback. So a self feedback topology with only local feedback could therefore be a hybrid network where the spectral radius as a measure for dynamics does not yet work well.

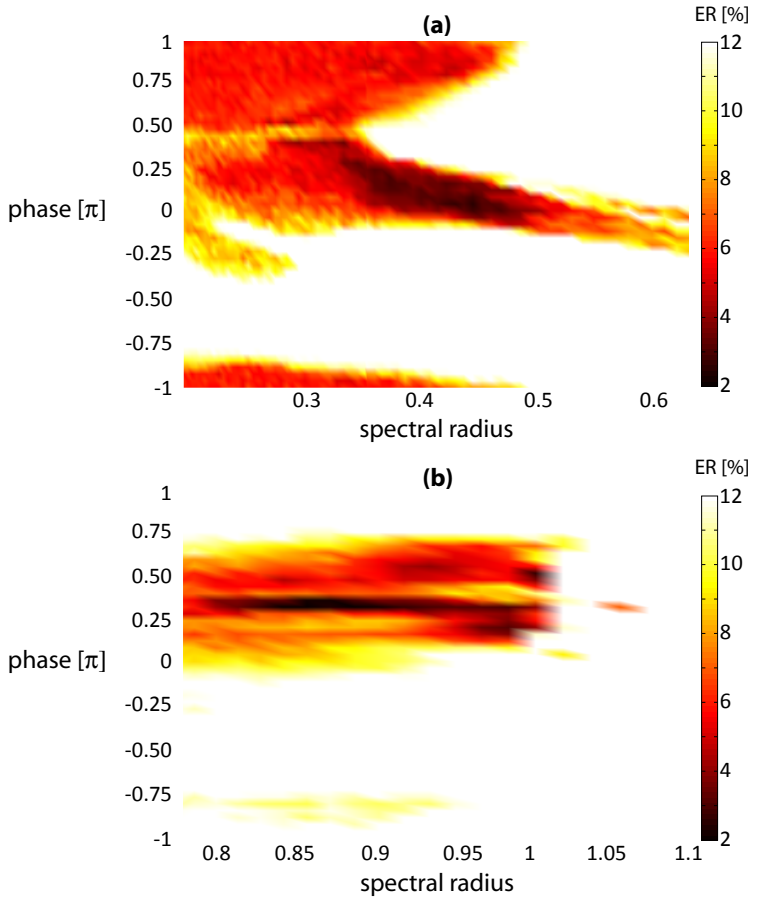


Figure 4.8: Different approaches to achieve larger optimal regions with different topologies: (a) a network with a topology with self feedback (Figure 3.10); (b) a network with different feedback mechanisms combined (Figure 4.9).

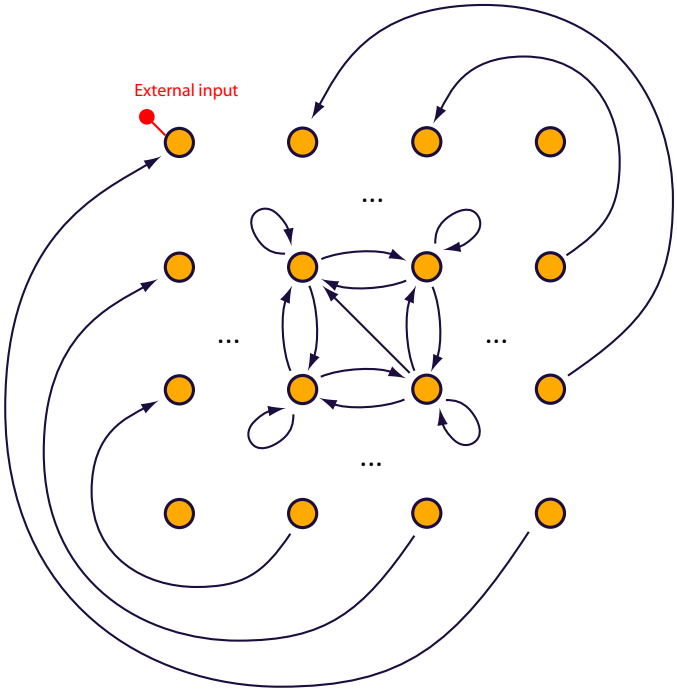


Figure 4.9: 4×4 topology with different feedback mechanisms combined (shown in Figures 3.9: edge feedback, 3.10: self feedback, 3.11: bidirectional feedback and 3.12: northwest feedback). A unit cell of the topology is drawn, together with the feedback around the edges of the network.

All this shows that delay is a very important parameter that should be taken into account when designing a reservoir for a task. Longer delays give the network a longer memory of the past, since information stays in it for longer periods, and that is an explanation as to why delay is such a critical parameter.

4.2.3 Size

Since we have been using a rather small-sized network of 25 nodes, it is expected that improvement is possible by increasing the network size, since it increases the amount of information offered to the linear readout function. This is confirmed by Figure 4.11 showing the performance of a 100 node reservoir (in a 10×10 configuration). The optimal region in comparison to the 5×5 network of Figure 4.3(b) is now extended to almost all the phases and a larger part of the spectral radius.

4.3 Comparison to classical coherent reservoirs

In the last section of this chapter we will see what happens if we replace the SOA nodes by tanh nodes in our architecture, while keeping the coherent behavior with complex amplitudes. We do this in order to investigate what differences between tanh and SOA nodes really matter.

4.3.1 Tanh

Tanh nodes are very commonly used in reservoir computing and form the basis of ESNs. When we use them instead of SOAs in a waterfall reservoir with edge feedback we get the performance of Figure 4.12(a), which is clearly worse than what we get with our standard SOA reservoir (Figure 4.3(b)). To explain the better performance of SOAs, let us see what happens when we turn off the linewidth enhancement factor of the SOAs ($\beta_c = 0$ in Equation 3.26)[6]: besides some very small optimal $(\rho, \Delta\Phi)$ regions around a spectral radius of 1, the performance is degraded everywhere to an error rate above 12 % (Figure 4.12(b)).

Switching off the linewidth enhancement factor β_c is the cleanest way to investigate the effect of the carrier lifetime[7], which can be most clearly seen when the input changes fast (Figure 3.5). τ_c provides some kind of memory mechanism because of its presence in the rate equation for the gain (Equation 3.29).

[6] In practice it is not possible to switch this off, although it its value can be changed by design and input current [6]. Switching it off, however, is instructive to gain insight.

[7] Comparing different values for τ_c is a bit tricky since it also influences the gain $\mathfrak{h}$. To compare SOAs with the same gain, but different carrier lifetimes the input current I has to be changed since both are inversely proportional $\tau_c I = C$ with $C = 5.610^{-11}$ for all our simulations. This was not explored in this dissertation.

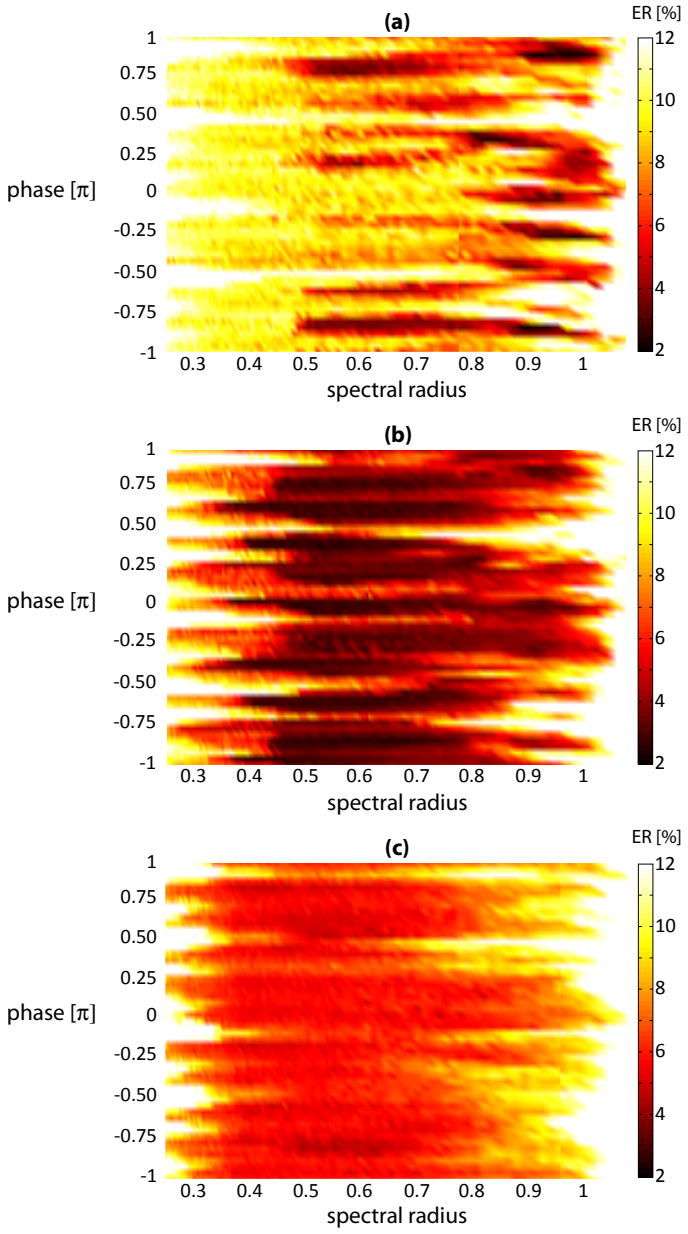


Figure 4.10: Performance of networks with different delays (a) 31.25 ps , (b) 56.25 ps , (c) 131.25 ps. It shows that there is an optimal delay for this network and task. The networks used in all other simulations have a delay of 6.25 ps and its performance is shown in Figure 4.3(a).

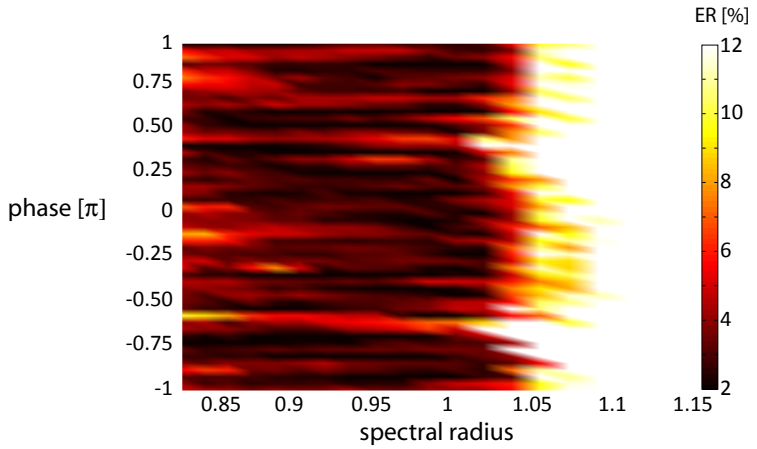


Figure 4.11: A network with edge feedback but with a size of 100 nodes (10×10), has a larger optimal $(\rho, \Delta\Phi)$ region.

This gain has an influence on the power (Equation 3.25), but also on the phase (Equation 3.26) and it is this influence that we can switch off by setting β_c to zero. The performance deterioration shows that internal node dynamics have a beneficial influence on SOA reservoirs and are probably the reason for the better performance of SOA reservoirs over classical tanh reservoirs.

4.3.2 Leak rate

A common approach for classical tanh reservoirs to add memory to the nodes is using leak rate. The leak rate parameter η is nothing more than a recursive first-order filter added to the nodes, which makes that past states of the node have an exponentially decreasing influence on the present node output (Equation 2.2). We were interested in how this so called leaky integrator neuron node type would behave when used in our novel architecture with complex amplitudes and weights. The performance with a leak rate of 0.0625[8] improves a lot and is for this particular task better than any of the SOA reservoirs used in this chapter (Figure 4.12(c)). The optimal region is almost phase independent and stretches out over a very large area in $(\rho, \Delta\Phi)$ space. The only SOA network that comes somewhat close to this, is the SOA network with optimal delay (Figure 4.10(b)). Leak rate slows the dynamics of the reservoir down and therefore it mostly redistributes the available memory of the reservoir over longer time scales instead of increasing it. Delay on the other hand is potentially more powerful as it not

[8] When simulating with a certain time step, the leak rate η equals the simulation time step divided by some time constant (e.g., the carrier lifetime). Here, the value of 0.0625 corresponds with a carrier lifetime of 100 ps and a time step of 6.25 ps.

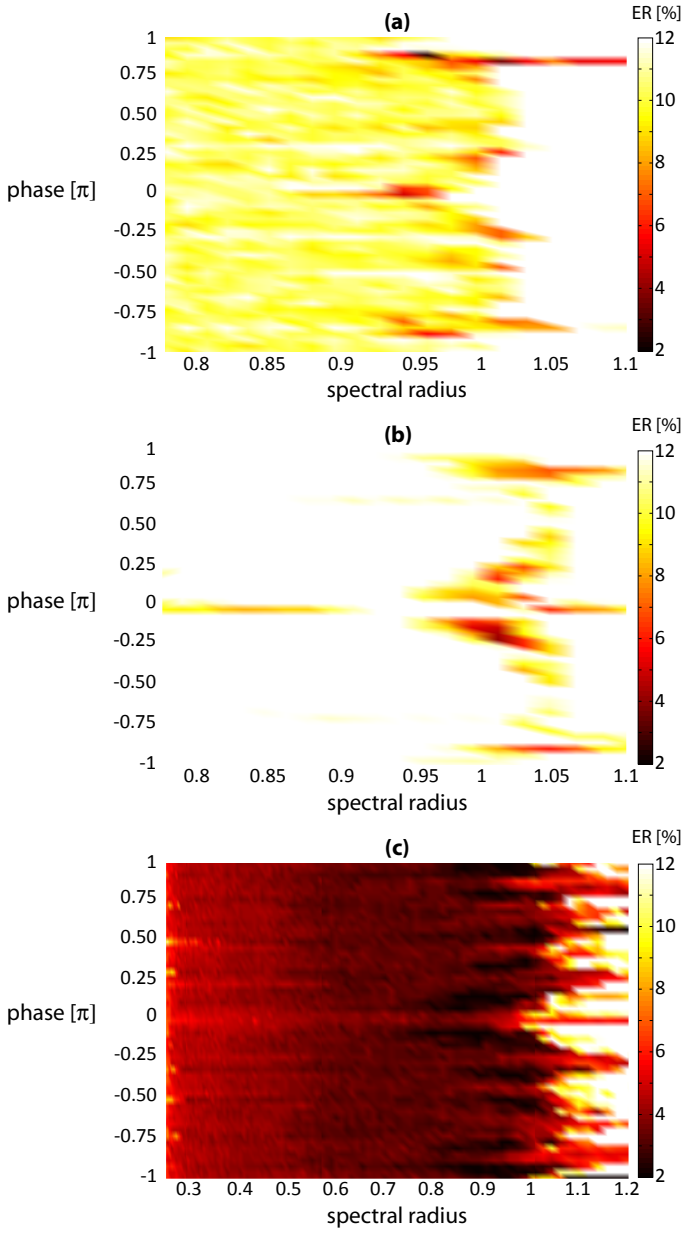


Figure 4.12: Performance of coherent networks with a 5×5 waterfall topology with edge feedback and a delay of 6.25 ps: (a) tanh reservoir, (b) SOA reservoir with the linewidth enhancement factor turned off ($\beta_c = 0$), (c) a tanh reservoir with leak rate ($\eta = 0.0625$).

only slows down the dynamics but also increases the memory of the reservoir as it adds extra states. This chapter also showed that bringing photonics inspired concepts, like phase, into classical RC is very interesting. Moreover photonic reservoir computing is a hardware implementation that naturally embodies phase influence.

4.4 Conclusions

In this chapter we have investigated whether an SOA network could be used for reservoir computing by applying it to a simple signal classification task, while coherence, topology constraints and internal dynamics set it apart from classical tanh reservoirs. The topology constraints together with the internal dynamics of SOAs show even a better performance than classical tanh reservoirs albeit for very small regions in $(\rho, \Delta\Phi)$ space. These regions prove to be very sensitive to noise. In our search for reservoirs with more noise robustness we have found that increasing the network size or employing different topologies can be used to achieve this. However, the parameter offering the most dramatic performance improvement turns out to be the interconnection delay between the nodes, achieving a performance that comes very close to that of coherent networks with leaky integrator neurons.

This chapter used a more qualitative approach to offer some insights as to why and when SOA reservoirs are useful. In the next chapter we will take this further by introducing a much more difficult speech recognition task and also using more quantitative analyses to study in detail the differences between SOA and tanh reservoirs.

References

[1] K. Vandoorne, W. Dierckx, B. Schrauwen, D. Verstraeten, R. Baets, P. Bienstman, and J. Van Campenhout. *Toward optical signal processing using Photonic Reservoir Computing.* Optics Express, 16(15):11182–11192, July 2008.

[2] A. N. Tikhonov and V. I. Arsenin. *Solutions of ill-posed problems.* Scripta series in mathematics. Winston ; distributed solely by Halsted Press, Washington; New York, 1977.

[3] R. Legenstein and W. Maass. *What makes a dynamical system computationally powerful?* In S. Haykin, J. C. Principe, T.J. Sejnowski, and J.G. McWhirter, editors, New Directions in Statistical Signal Processing: From Systems to Brains, pages 127–154. MIT Press, 2007.

[4] B. E. A. Saleh and M. C. Teich. *Fundamentals of photonics.* John Wiley & Sons, Inc., New York, NY, USA, 1991.

[5] D. Verstraeten. *Reservoir Computing : computation with dynamical systems.* PhD thesis, Ghent University, 2009.

[6] M. Osinski and J. Buus. *Linewidth Broadening Factor in Semiconductor-Lasers - an Overview.* IEEE Journal of Quantum Electronics, 23(1):9–29, 1987.

5 Speech recognition

The insights from the previous chapter are that delay in the interconnections and coherence are two critical parameters determining the performance of SOA reservoirs. In this chapter we will explore this exhaustively, investigating which of the differences between SOA reservoirs and classical tanh reservoirs really matter. Therefore we will use a much more complex task, i.e., isolated digit recognition. The first part of this chapter describes the task in detail, the second part identifies the most important parameters, the third part discusses the influence of noise and the last part deals with aspects of the readout. The first insight gained from this chapter is that delay is the most critical design parameter and it has to be taken into account when designing a photonic reservoir for a specific task. Second, the use of coherence combined with an optimal delay yields a better performance than the one achieved with classical leaky tanh reservoirs. Third, process variations hardly degrade the performance, but ASE does. Therefore careful consideration of how to mitigate the detrimental influence of ASE is required when designing photonic reservoirs with SOAs. The results of this chapter have been submitted for publication [1].

5.1 Speech recognition task

Speech recognition remains a nontrivial task to solve and neural networks have long been investigated as an alternative for standard speech recognition meth-

ods based on Hidden Markov Models (HMM), which seem to have reached their limits. The research interest in RNNs was reinvigorated in recent years especially due to the increasing availability of computational power [2–5]. Reservoir computing, too, has proven its worth for this task [6]. The task used in this chapter is the classification of isolated spoken digits (from 'zero' to 'nine'). In the data set, these 10 words are each spoken 10 times by 5 female speakers, giving 500 samples, taken from the TI46 speech corpus [7]. This data set is freely available online[1]. Since it is possible to achieve a WER very close to 0 %, babble noise from the NOISEX database[2] was added with a SNR of 3 dB [8]. While not pushing the boundaries of speech recognition itself — continuous speech recognition is a lot harder — this task is complex enough to evaluate the performance and potential of photonic reservoir computing.

5.1.1 Pre-processing

It is common practice in speech recognition to have some pre-processing of the raw speech signal. Methods often involve a transformation to the frequency domain and a selective filtering based on known psycho-acoustic properties of the human ear and/or spectral properties of speech. In his PhD thesis David Verstraeten compared different pre-processing methods[3] that vary in degree of realism and complexity [9]. He discovered that the Lyon passive ear model matches very well with the recognition capabilities of classical reservoir methods and therefore we use this model in our experiments [10].

The Lyon passive ear model is a model of the human inner ear or cochlea. Cochlea is Latin for snail or spiral referring to its coiled shape. It is filled with a liquid and the hair cells inside convert vibrations in the liquid to electric signals in the nerve cells. The response of the inner ear and its selectivity to certain frequencies is modeled with a notch filter bank, followed by Half-Wave Rectifiers (HWRs) and Adaptive Gain Controllers (AGCs) (Figure 5.1). The HWRs set any negative value from the signals after the filter bank to zero and provide a crude energy measure of the signal. The AGCs adapt their gain, depending on the input level of the corresponding channel and its nearest neighbors, to prevent its output from exceeding a fixed level. The AGCs' outputs are positive signals indicating the firing rate of neurons to the brain. Figure 5.2 shows an example of the response of the notch filters in the filter bank. A more detailed explanation of the model and its implementation can be found in a technical report from Malcolm Slaney [11].

[1] http://snn.elis.ugent.be/rctoolbox

[2] http://spib.rice.edu/spib/select_noise.html

[3] a Matlab implementation written by Malcolm Slaney is publically available at http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/

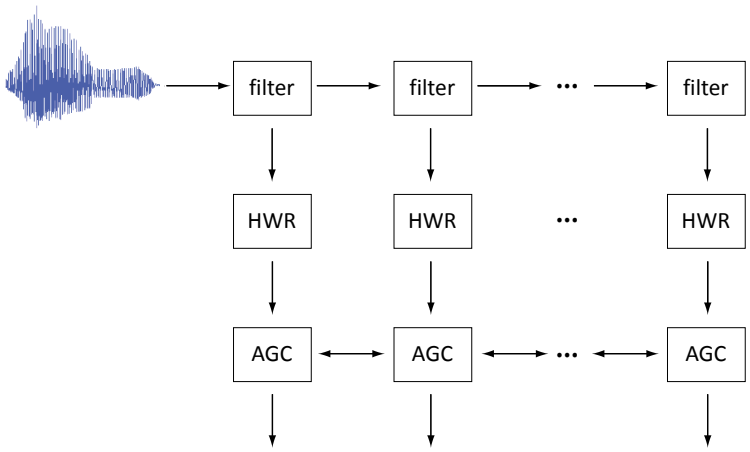


Figure 5.1: Schematic overview of the elements of the Lyon cochlear model: a cascading notch filter bank followed by half-wave rectifiers and adaptive gain controllers.

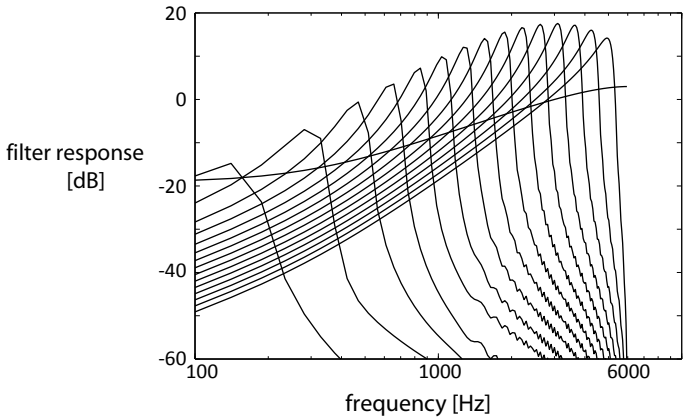


Figure 5.2: Filters' response for the Lyon passive ear model (every fifth filter is shown).

In our case the model returned 77 channels, which means that each audio sample gets transformed into 77 information streams. To shorten the simulation time, a decimation with a factor of 128 was applied[4]. Figure 5.3(a) shows an example output of the Lyon model for one speech sample.

5.1.2 Input and output

The different channels from the Lyon model can be fed directly into the reservoir nodes, but in our case we mix the channels differently for each node, with weights randomly alternating between -0.1 and 0.1. These weights are combined into the matrix $\mathbf{W}_{in}$ as in Equation 2.2 and multiplied by the inputs from the channels $\mathbf{u}_{channels}$ to obtain the inputs for every node $\mathbf{u}_{nodes}$[5]. An example of such an input for the nodes can be seen in Figure 5.3(b) and these can be negative. We will use power values for photonic reservoirs, so an extra step is needed, where all the inputs are made non-negative by shifting all the channels upwards with the minimum over the inputs for all the nodes (Vertical arrow in Figure 5.3(b)). We envisage this being done electronically, and then converting the resulting input signals to optical node inputs with a modulator. We make the arbitrary choice that the values on the Y-axis represent power values in watt, because this makes for realistic input power values remaining below 1 mW, which corresponds to the linear regime of an SOA driven by 187 mA (Figure 3.4). The range of input values can however be changed by applying input scaling. In the resulting topology, either all nodes or only a fraction of them can be driven by the external input signal. In the experiments presented in this chapter, all the nodes were driven by the external input.

The output is obtained by training ten distinct linear classifiers, one for each digit (Section 2.4). Each classifier should return $+1$ whenever its corresponding digit is spoken and -1 otherwise. An example can be seen in Figure 5.4 where (a) shows the desired output of the classifiers (in this case classifier 9 should win), while (b) shows the actual output after training with the methods from chapter 2. The actual output is not as clear cut as the desired output and to determine the actual classification a winner-take-all approach is used by taking the temporal mean over all the classifiers and selecting the one with the strongest positive

[4] Audio signals are sampled with a certain sample rate and for the audio signals in this chapter the rate was 12 kHz or 12000 times per second. This rate can be reduced by downsampling, in our case using only every 128th sample of the original signal. The sampling theorem guarantees that bandlimited signals (i.e., signals which have a maximum frequency) can be reconstructed perfectly from their sampled version, if the sampling rate is more than twice the maximum frequency (Nyquist frequency). In order for the downsampled signal to comply, a low-pass filter is applied first that filters out all the frequencies higher than half of the new sample rate. The whole process of first low-pass filtering and then downsampling is called decimation [12].

[5] The corresponding matrix dimensions for $\mathbf{u}_{nodes} = \mathbf{W}_{in}\mathbf{u}_{channels}$ are (#nodes×time) = (#nodes×#channels) * (#channels×time)

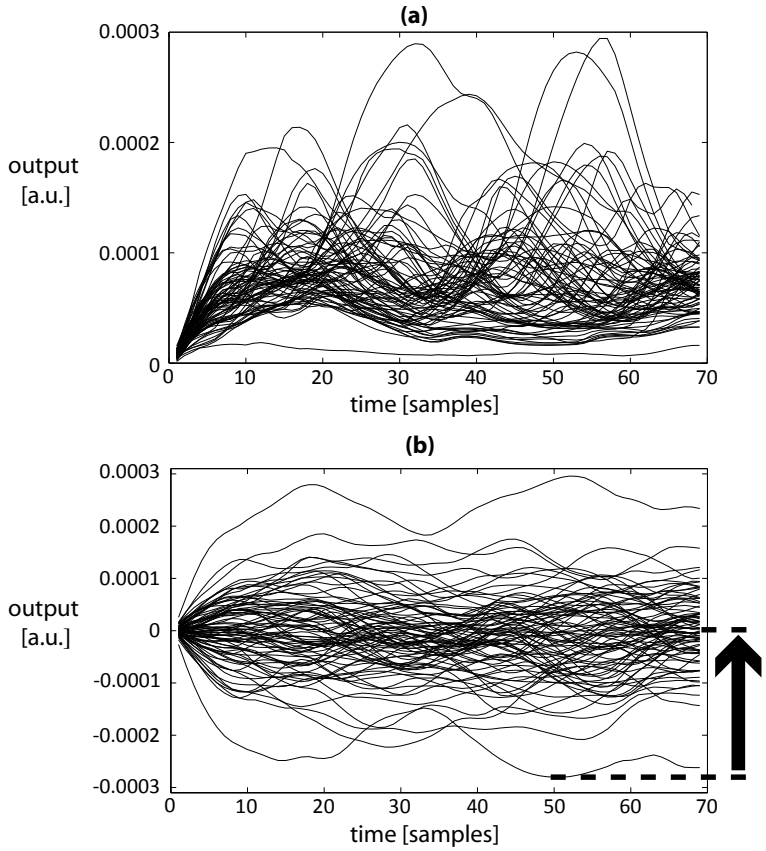


Figure 5.3: (a) shows the 77-channel output of the Lyon passive ear model for an audio signal; (b) shows the actual input for all 81 nodes of the reservoir with a different mixing of the 77 channels (from (a)) for every node. Since certain inputs are negative, all the inputs are shifted upwards in the vertical direction with the minimum over all the inputs for this sample (arrow) if power values are to be used. The X-axis shows the number of samples remaining after decimation with a factor of 128. The Y-axis has the dimensions returned by the Lyon model.

response.

Figure 5.5 shows the temporal means of training results as in Figure 5.4. The black bars are the means of the desired output, obviously still -1 and $+1$, while the white bars are the temporal mean of the trained classifiers. In the ideal case these two should be the same, however since we take a winner-take-all approach, the classification is correct if the classifier with the highest white bar (thicker edge) is also the one with a black bar equal to $+1$ as in Figure 5.5(a). Figure 5.5(b) presents a case of a sample that will be misclassified.

The total performance for this task is expressed by the Word Error Rate (WER), which is $\frac{N_{nc}}{N_{tot}}$, with N_{nc} the number of incorrectly classified samples, and N_{tot} the total number of samples. We use ridge regression to avoid over-fitting and five-fold cross-validation to make our results more robust as explained in Section 2.4. The mixing of the inputs and the cross-validation train and test sets vary from run to run and therefore the results are always averaged over 10 runs.

5.1.3 Classical network baseline

Throughout the rest of this chapter, we will compare our results to those of classical reservoirs with tanh neurons. In most experiments, the output of the Lyon ear model, consisting of 77 channels, was fed into a reservoir of 81 neurons. This relatively small number of neurons was a compromise between the photonic network simulation time that increases with larger networks, and using a number of nodes surpassing the number of channels. For a classical tanh reservoir with random interconnections, the optimal WER for this isolated digit speech recognition task with babble noise is around 11.7 % if we do not use leaky integrator neurons and around 7.7 % if we do (Figure 5.6). These values compare very favorably to the state-of-the-art, especially in the presence of babble noise[6]. The optimal leak rate is 0.1, the same value as was obtained by D. Verstraeten in his PhD thesis for this speech task [9].

One important issue deals with the time scales involved. Audio signals are much slower in comparison to the typical time scales of delays in an integrated optical network. Therefore we have accelerated the speech signal to correspond to those time scales. The delays on a chip vary with the physical length of the interconnections from a few ps to a few 100 ps, depending on the material systems and structures used. The audio samples used have a duration of about 0.5 ms. So, for the photonic reservoirs we feed them to the network 9 orders of magnitude faster, which makes for samples only lasting around 500 ps. Although we use this task to demonstrate the potential of photonic reservoir computing,

[6] In [13] the performance of a HMM with a front end specifically designed for noise robustness is tested with babble noise and SNRs of 30, 20 and 10 dB on a similar but different dataset. At 10 dB the WER is already 17.5 % and we use in this chapter a SNR of 3 dB

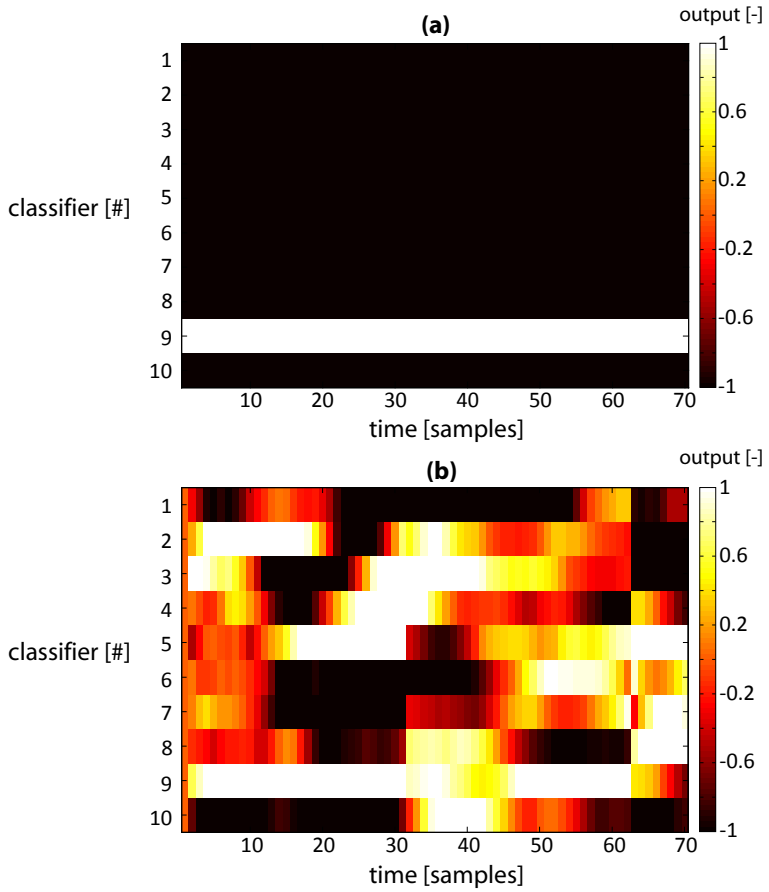


Figure 5.4: In the training phase with linear regression for the 10 classifiers (a) represents the desired output of the classifiers (+1 for one of the classifiers, −1 for the others), while (b) shows their output after training. In this case classifier 9 has an output of +1 or higher most of the time but ideally it should be for the entire timespan. For the other classifiers it shows that the training output approaches the desired output, but deviations can arise and even be large.

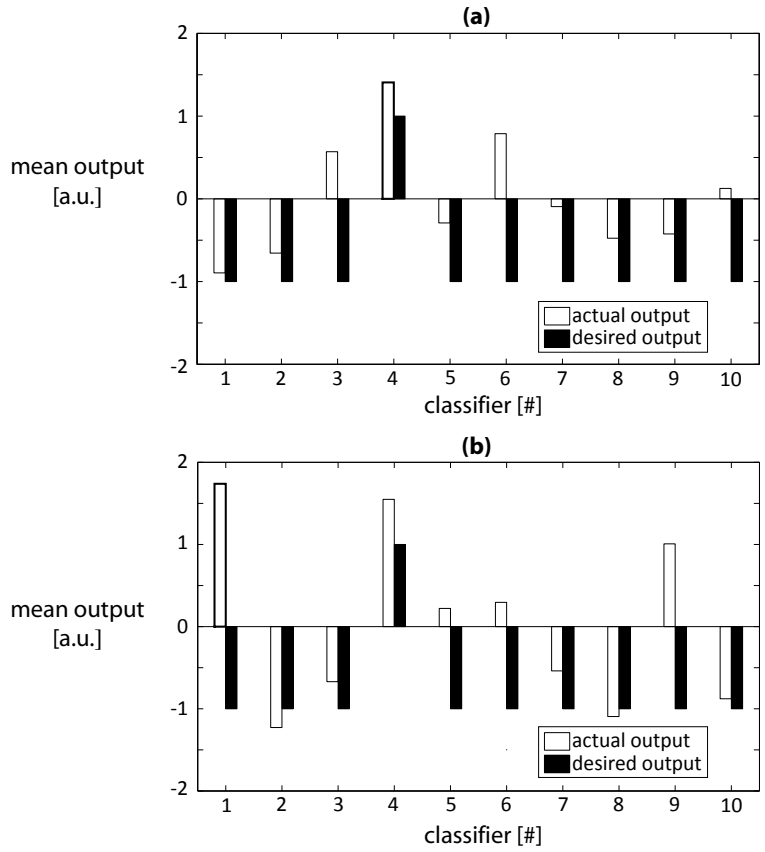


Figure 5.5: For classification we take the mean over the entire sample length of both the trained classifiers' output (white) and their desired output (black) and the classifier with the highest output wins (thicker edge). If this corresponds to the desired classifier, the classification is correct as in (a) where classifier 4 has the highest output; In (b) the classification is wrong since classifier 1 has the highest output while it should have been classifier 4.

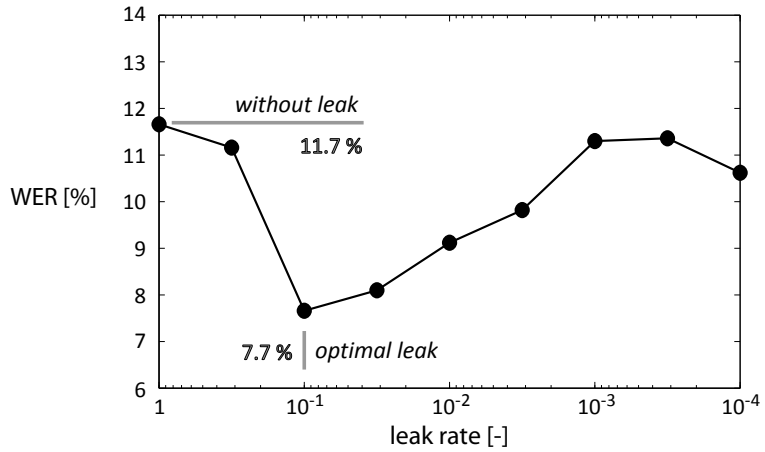


Figure 5.6: WER for classical ESNs with a sweep for the leak rate parameter η (Equation 2.2). This was done for networks with a spectral radius equal to unity. Using some leak rate is clearly advantageous for this speech task. To obtain a broad range of values, a logarithmic spacing was used yielding an optimal value of 0.1 for η.

we do not propose to use photonic reservoirs as a platform for standard real-time, slow audio signals.

5.2 Relevant Parameters

This section deals with the identification of the most important parameters in photonic reservoirs (delay and coherence), while simultaneously exploring which differences between photonic and tanh reservoirs really matter.

5.2.1 Tuning parameters

For traditional reservoir computing, a number of global parameters are usually tuned to optimize an architecture for a given task. In our design too, there are a number of parameters we can explore to optimize performance. Some of those need to be fixed at design time, while others can be changed at operation time to compensate for differences between the simulation model and the actual hardware. These can be caused by inaccuracies in the model, but also by process variations or temperature effects.

At operation time, we can still tune the system's gain (spectral radius ρ) and the phase shift $\Delta\Phi$ in each connection. The gain can be altered by adapting the input current of the SOAs or the attenuation in the interconnections. As for

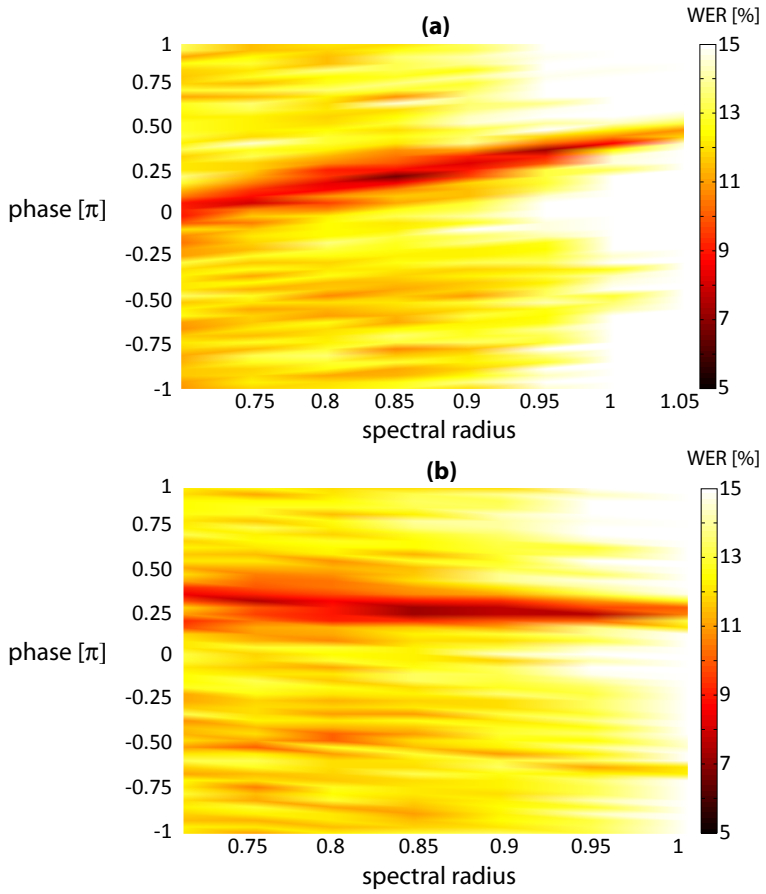


Figure 5.7: WER for swirl SOA reservoirs with coherent light and a delay of 6.25 ps. (a) the spectral radius is changed through the input current of all the SOAs; (b) the spectral radius is changed through the attenuation in all the connections.

phase shift, there are two ways to optimize this and both can be understood from Equation 3.33 and the definition of $\Delta\Phi$. Changing the wavelength is the easiest approach, although the requirements for the wavelength tunability become stricter for longer interconnections as the wavelength steps have to be smaller to scan the entire $[-\pi,\pi]$ range with the same accuracy. Another approach is to actively tune the effective index n_{eff} in a section of the waveguide to achieve the required phase shift. This depends only on the length of the tuning section and not on the total connection length as for the wavelength. Active tuning can be done for example with carrier injection [14–17], temperature [18, 19] or liquid crystals [20, 21]. The exact mechanism for sweeping the phase does not matter for our experiments.

At design time, we fix the physical properties of the SOAs, the length of the interconnections and the overall loss in the system. For most of our experiments, we have evaluated each set of physical design parameters as a function of the tunable parameters ρ and $\Delta\Phi$.

Note that the input current not only affects the gain, but also the phase change at the output of the SOAs (Equation 3.26). This phase interacts with the phase change of the interconnections, so input current and interconnect phase are not orthogonal parameters. This is illustrated in Figure 5.7(a), showing a $(\rho,\Delta\Phi)$ plot, averaged over 10 runs, of the performance (color-coded WER) of a SOA reservoir with a swirl topology and minimal-length interconnections. We can see that the optimal (dark) region is slanted. For the simulations in the rest of this chapter, we have therefore varied the spectral radius by adjusting the attenuation in the connections (as shown in Figure 5.7(b)). This simplification makes our results easier to interpret without affecting the conclusions that can be drawn from them.

Comparing Figure 5.7 with the baseline WER of 7.3 %, we can conclude that, for most of the parameter space of this particular example design, this network of SOAs performs considerably worse than traditional reservoirs with leaky tanh neurons. However, for a relatively narrow region of the parameter space, the performance improves and approaches that of the leaky tanh reservoirs. A result very similar to the results for the signal classification task with SOA reservoirs with short delays (Figure 4.3). Unfortunately, whether or not this optimum can be reached depends on the accuracy with which we can control the phase change in the interconnections. As this control is in practice not straightforward (sensitive to temperature and on-chip noise), a good architecture is one that shows a relatively broad $(\rho,\Delta\Phi)$-region with high performance. In the rest of this chapter, we will show various ways of achieving this.

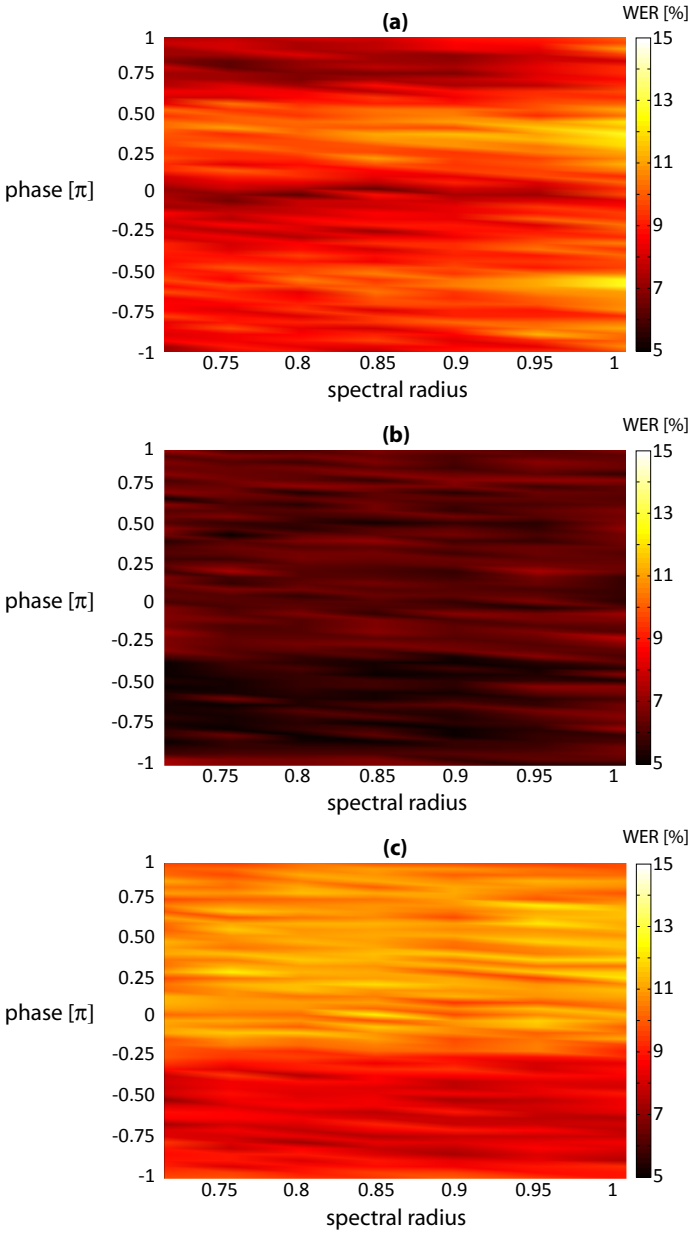


Figure 5.8: WER for SOA reservoirs with coherent light and different delays: (a) delay = 75 ps, (b) delay = 187.5 ps, (c) delay = 312.5 ps. The delay inside the SOAs is 6.25 ps. The performance of a network with delay 187.5 ps is better than the other two.

5.2.2 Delay

In traditional simulated reservoirs, time is usually discretized. At each simulated time step, all neuron states are updated using the previous neuron states and a new input sample. One could view this as the neurons having a single time step delay and the interconnections being instantaneous. In contrast, in our photonic implementation, time is continuous and no synchronization mechanism is used. The input signals are continuous signals, only discretized for simulation purposes. The SOAs have a certain propagation delay and so have the interconnections between them.

Figure 5.8 shows the results of $(\rho, \Delta\Phi)$ parameter sweeps for the SOA reservoir using three different interconnection delay values. It shows that performance can be optimized by sweeping the interconnection length across a wide range of values. Also, for longer delays, the performance becomes less sensitive to the phase change in the interconnections, making the design more robust. This confirms similar findings for the signal classification task (Figure 4.10). Here, the effects of multiple-beam interference as discussed in Section 3.2.4 and Figure 3.16, are probably even less pronounced because of the short duration of the audio signal.

Delays have been investigated for neural networks, especially regarding stability criteria [22–26] and their influence on synchronization [27–32]. However, before the start of this work, very little was known about the impact of interconnection delay in traditional reservoirs. In this section, we will investigate its impact in tanh reservoirs, operating in continuous time. For this analysis, we will assume that all interconnections have equal delay. We have analyzed tanh reservoirs with random and swirl topology, as well as with and without leak rate. For random networks the inputs and weights can be negative as is the case for typical reservoirs, while for swirl tanh networks we restrict ourselves to positive inputs and weights just like SOA networks. The results are summarized in Figure 5.9. For each delay value, we report the best score found in a $(\rho, \Delta\Phi)$ sweep.

First, we see that the differences related to positive input and topology are negligible, both for tanh networks with and without leak — except for very small delays. When leak rate is used, the swirl topology performs even slightly better than random networks. This means that the performance impact of using a planar topology is small in ESN reservoirs with 81 nodes. It is far less significant than the impact of memory-optimizations like the introduction of leak rate to the neurons. In the remainder of this chapter, we will therefore only compare the performance of our SOA reservoirs to that of tanh networks with positive weights and the swirl topology. This implies that later comparisons between swirl SOA and tanh networks will only differ in their node functionality. The influence of topology will only be investigated again in Section 5.2.7 about the impact of

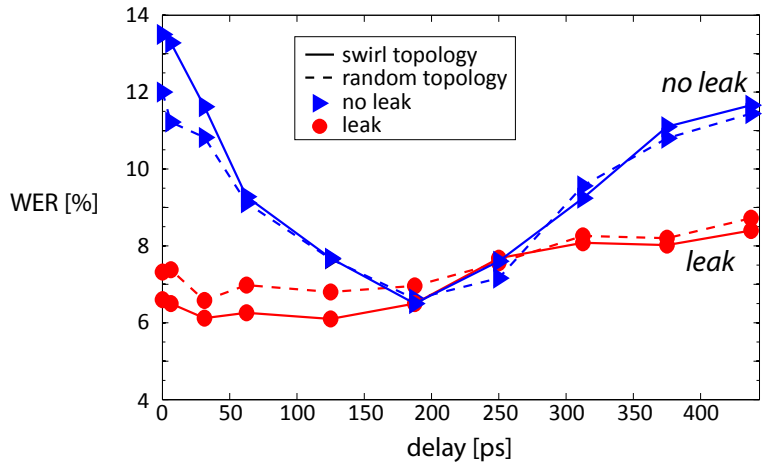


Figure 5.9: Performance of tanh networks (with swirl or random topologies and with or without leak rate in the nodes) in function of the interconnection delay between the nodes.The performance of tanh networks with random and swirl topology differs only slightly, while the presence of leak matters a lot. For reservoirs without leak rate there exists a delay for which the performance is optimal, and coincides with the performance with leak.

network size.

Second, Figure 5.9 shows that the optimal delay is not just a feature of SOA networks but also of tanh networks without leak. Reservoirs with leak rate are initially hardly affected by longer delays, but their performance decreases as well for very long delays, albeit slower than in the absence of leak rate. The result of a tanh network without leak rate at its optimal delay is actually comparable to that of a tanh network with leak rate and no (or minimal) delay. This reinforces the view from Chapter 4 that delay is an alternative approach to introducing memory next to leak rate.

For this application, the optimal performance can always be found around a delay of 190 ps, which roughly corresponds to half of the duration of the audio signals. Clearly, the interconnection length needs to be optimized for the task at hand as part of the design effort and cannot be changed after fabrication. In SOA implementations, long delays can be realized compactly by using spiraling waveguides[7]. Alternatively the input can be resampled to match the delays, but

[7]The longer the spiral, the higher the losses and the higher the SOA gain should be to compensate for these losses. In Silicon-on-Insulator spirals of 5 cm with a loss of 3 dB/cm have been used, corresponding to delays around 625 ps. This is longer than what we have used in this paper (~500 ps) [33]. However, recent improvements have reduced the waveguide loss to around 0.3 dB/cm, a tenfold

further research is required to investigate to what extent these two approaches are equivalent, although experiments with different signal input speeds suggests they are (cf. Section 5.2.5). Since there are limits to how fast one can resample the input and how long one can make the connections, a combination of both approaches will probably be needed.

5.2.3 Coherence

Our architecture can handle light at one wavelength coming from a laser[8], but it can also handle light from an incoherent source such as a Superluminescent Light-Emitting Diode (SLED)[9]. In the first case, the system must be modeled as coherent, i.e., using complex values for all state variables and connection weights. This implies that the number of internal state variables in coherent reservoirs is twice the number of nodes, but only half of them (the power values) are being used for the readout. In Section 5.4.2 we will compare using only power readout values versus using the other state variable (phase) or both of them, but all other experiments in this chapter are done with just the power values. In the second (SLED) case, the light is represented by power values everywhere (i.e., not only at the readout) instead of complex amplitudes. In this section, we quantify the impact of using coherent light on the performance of our SOA reservoir implementation. We will systematically refer to coherent and incoherent simulations, respectively, to make the distinction between both. Since our architecture allows the information to be carried by either light intensities only (incoherent light) or by light at a single wavelength (coherent light). Inter-wavelength interaction such as cross-phase modulation (XPM) and four-wave mixing (FWM) is not relevant in either scenario, so nonlinear effects between signals at a distinct set of wavelengths were not considered. We also work mostly in the linear low-power regime of the SOAs where these nonlinear interactions are very weak.

Plots like the ones in Figure 5.8 are very informative to qualitatively compare simulations for coherent reservoirs, but for a quantitative study we extract two values from these sweeps. The first value is the minimal result in a $(\rho, \Delta\Phi)$ plot, reflecting a situation of perfect phase control of the interconnections. For the

reduction, allowing for smaller SOA gains for the same spiral length [34].

[8] In practice, to have all the inputs on the same wavelength, light from one laser could be split into the number of nodes that require an input and each input would get its information transferred to it by a modulator.

[9] We did not use an actual laser or SLED source in our simulations. We explored the two extremes: perfect incoherent light which is broadband and where only intensities count, and perfect coherent light at one wavelength where light can be represented by complex amplitudes. The first can be approximated by using broadband sources such as SLEDs, the second by single mode lasers. Our experiments could be done for light around 1550 nm or 1300 nm where SOAs, lasers and SLEDS with bandwidths around 50 nm, are readily available.

second value, the $(\rho, \Delta\Phi)$ results are first averaged over all the phase values for each value of the spectral radius, corresponding to the average performance when no phase control is available. Then, the minimal value of these average performances over all the spectral radii is picked.

Since we will be comparing different $(\rho, \Delta\Phi)$ sweeps in the remainder of this chapter, with the information condensed into just one value, it is important to check how robust these results are. We expect some variations because of the randomness in each run when combining the 77 input channels and choosing the sets during grid search in the training and validation stage. Therefore we simulated four times the same coherent SOA reservoir with a delay of 6.25 ps as in Figure 5.7(b) (*sim 1*). Figure 5.10 reveals that the optimal region is located around the same phase for the four sweeps, but the optimal $(\rho, \Delta\Phi)$ combination itself is different (the darkest region differs for the four plots). Table 5.1 shows that the minimal result between the four sweeps only differs 0.1 %, while the standard deviations over 10 runs at the respective optimal $(\rho, \Delta\Phi)$ combinations (*local variability*) are at least 10 times greater. When looking at the same $(\rho, \Delta\Phi)$ point, here the optimum of *sim 2*, the difference in performance increases to 0.8 % (*spatial variability*). So when comparing individual simulations at one $(\rho, \Delta\Phi)$ point, one should not attribute too much importance to small differences, because of the large variances [10]. However, when comparing optimal results over whole $(\rho, \Delta\Phi)$ sweeps, this section has shown that they are quite robust and can therefore be used to compare different reservoirs or parameters.

The influence of coherence is addressed in Figures 5.11 and 5.12 where coherent and incoherent swirl reservoirs are compared using tanh and SOA nodes in the absence of leak rate. Figure 5.11 shows the performance after averaging over the phase, whereas Figure 5.12 shows results for reservoirs with perfect phase control. Both figures show that the use of SOA nodes or the addition of coherence do not alter the conclusions with respect to interconnection delay of the previous section. Indeed, in both figures, all four curves roughly follow the same trend and show an optimum for the same interconnection delay.

In Figure 5.11, the four curves are very close. Without phase control, as expected we observe only a minor performance benefit for the coherent simulations for all delay values except for the very smallest. As Figure 5.12 shows, a perfect control of the phase drastically improves the performance of coherent reservoirs, both with tanh and with SOA neurons: the coherent results are shifted downward by about 2 %, suggesting that coherence could improve state-of-the-art performance. In the remainder of this chapter when we refer to coherent results, perfect phase control will be assumed unless stated otherwise.

We can also conclude from Figures 5.11 and 5.12 that the replacement of

[10] An approach to increase the robustness at individual $(\rho, \Delta\Phi)$ points is to increase the number of runs, but this comes with a high computational cost.

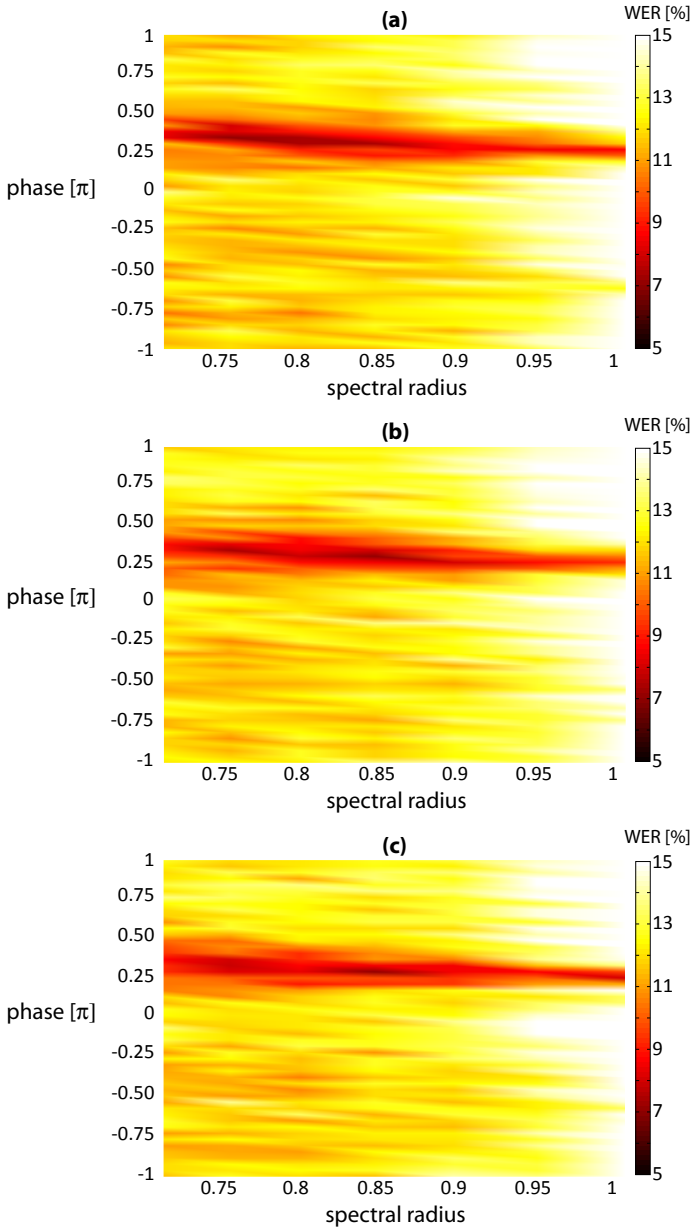


Figure 5.10: $(\rho, \Delta\Phi)$ sweeps of three times the same coherent SOA reservoir with swirl topology and a delay of 6.25 ps. (a) sim 2, (b) sim 3, (c) sim 4; sim 1 is shown in Figure 5.7(b).

	minimal result	σ (10 runs)	result @ optimum sim 2 ($\rho = 0.76, \Delta\Phi = 0.337\pi$)
sim 1	7.2 %	1.4 %	7.5 %
sim 2	7.1 %	1.2 %	7.1 %
sim 3	7.2 %	2.0 %	7.5 %
sim 4	7.1 %	1.9 %	7.9 %

Table 5.1: The optimal result for a $(\rho, \Delta\Phi)$ sweep with 10 runs is the minimum of the average performance and this minimum is at one optimal $(\rho, \Delta\Phi)$ combination. This table shows the optimal results of 4 $(\rho, \Delta\Phi)$ sweeps with 10 runs of the same coherent SOA network with a delay of 6.25 ps. In the second column, the standard deviation σ over 10 runs at the optimal $(\rho, \Delta\Phi)$ point is shown (local variability). The optimal point and corresponding minimal value differ slightly for every simulation as can be seen in Figure 5.10. The optimum for *sim 2* is at $\rho = 0.76$ and $\Delta\Phi = 0.337\pi$ and the last column shows the results for the other simulations at that same optimal point (spatial variability).

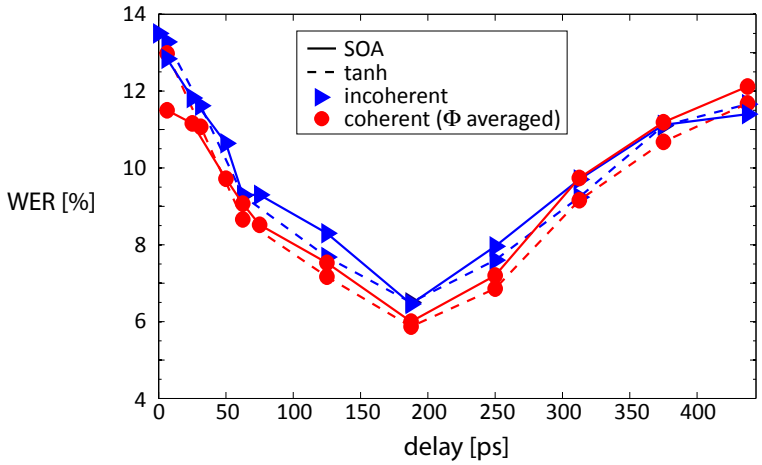


Figure 5.11: Incoherent versus coherent tanh and SOA networks where the coherent results are first averaged over the phase. All reservoirs had a swirl topology with positive weights and leak was not used. In this case there is hardly any improvement using coherent reservoirs.

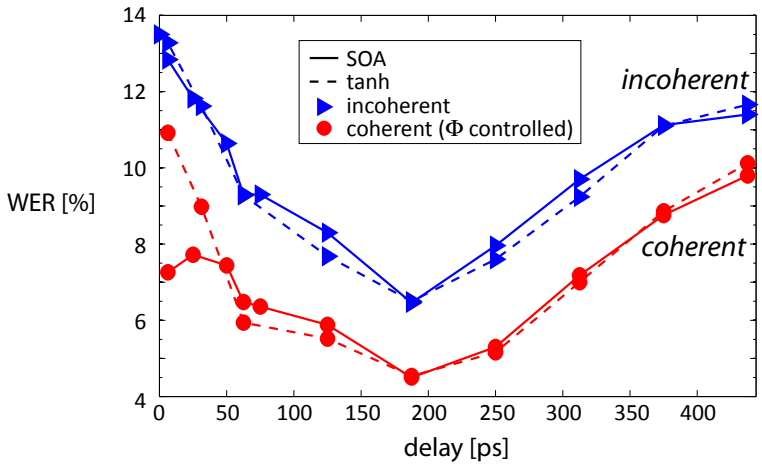


Figure 5.12: Incoherent versus coherent tanh and SOA networks. For the coherent networks the optimal value of the phase was used (phase-controlled). All reservoirs had a swirl topology with positive weights and leak was not used. In this case there is a clear improvement using coherent reservoirs.

	minimum
SOA swirl coherent ($\beta_c = 5$)	7.2 %
SOA swirl coherent ($\beta_c = 0$)	10.4 %
tanh swirl coherent	10.9 %

Table 5.2: The minimum results for a coherent tanh swirl network, a coherent SOA network and a coherent SOA network with β_c switched off. All have an interconnection delay of 6.25 ps and at this short delay the coherent SOA network clearly performs better than the other two.

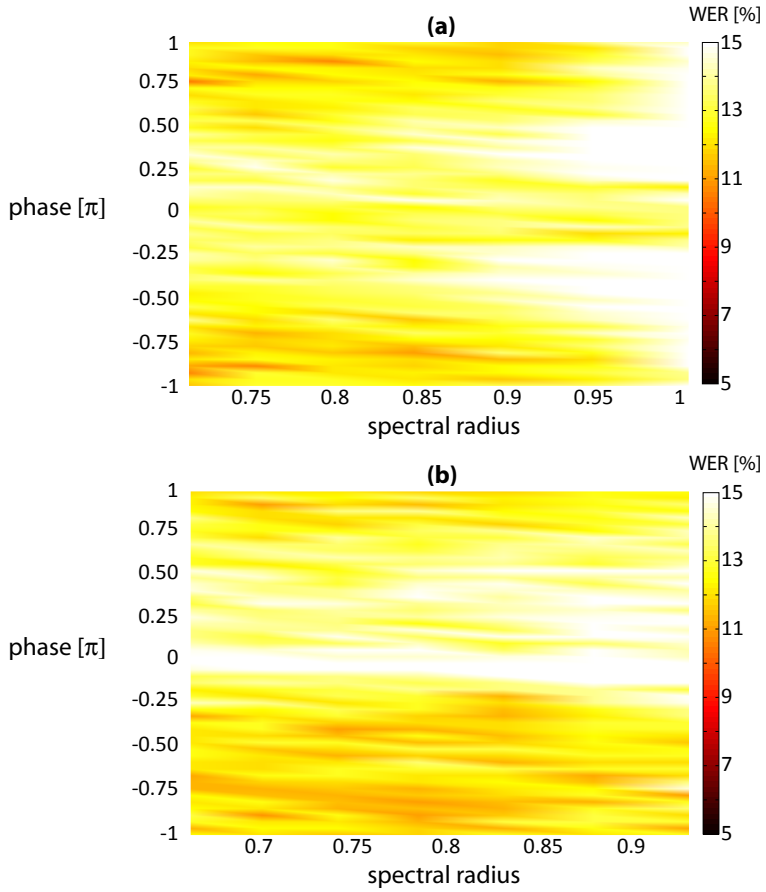


Figure 5.13: $(\rho, \Delta\Phi)$ plots for coherent networks with short delays (6.25 ps): (a) SOA reservoir with the linewidth enhancement factor switched off ($\beta_c = 0$); (b) tanh reservoir. The performance is similar as both lack the optimal region that coherent SOA reservoirs have (Figures 5.7 and 5.10 with $\beta_c = 5$).

input scaling	1	optimal
incoherent tanh without leak	10.9 %	9.9 %
incoherent tanh with leak	7.3 %	6.2 %
incoherent SOA with delay 0 ps	14.7 %	12.3 %
incoherent SOA with delay 187.5 ps	7.5 %	5.9 %

Table 5.3: The result for tanh networks with and without leak at the standard input scaling (1) and at their optimal input scaling. The delay was zero and the spectral radius 0.9. The result for incoherent SOA networks with delay 0 ps and the optimal delay of 187.5 ps and both with a spectral radius of 0.9.

tanh nodes by SOA nodes barely affects the performance, except in the coherent simulations for small interconnection delays, where SOA nodes are clearly beneficial. The rather good optimal performance of SOA networks with short delays, already shown in Figure 5.7(b), turned out to be for a very narrow region in $(\rho, \Delta\Phi)$ space. This optimal region is completely absent for coherent tanh swirl networks with a short delay (Figure 5.13(b)). The effect responsible for this improved performance for SOAs is the influence of the carrier lifetime τ_c on SOA gain and the phase shift inside the SOA through the linewidth enhancement factor β_c (Equation 3.26). The easiest way to artificially turn this influence off, is to set β_c to zero just as in Section 4.3.1[11] (setting $\tau_c = 0$ is not possible since it also influences the value of the SOA gain in the model). As in Chapter 4, the well-performing $(\rho, \Delta\Phi)$ regions disappear in this case (Figure 5.13(a)). The minimum results for these 3 situations tell the same story (Table 5.2). The presence of the carrier lifetime τ_c in the rate equation for the SOA gain (Equation 3.29), makes that past inputs exert an exponential decaying influence on the present output (Figure 3.5). This means that for short delays carrier lifetime acts as a third kind of memory mechanism beneficial for the performance, next to leak rate and interconnection delay. For networks without leak rate or sufficiently large delay, this third mechanism improves the performance of SOA reservoirs, but it requires phase control since the improvement almost vanishes completely with phase-averaged results (Figure 5.11). However, in contrast to leak rate, this effect is not strong enough to dominate other memory optimizations. For longer delays (over 50 ps) the advantage disappears as it goes beyond the reach of the carrier lifetime and its effect becomes negligible (Figure 5.12).

[11] Again, in reality it is not possible to switch this off or set it to zero as it is a fundamental parameter of semiconductor lasers and amplifiers [35].

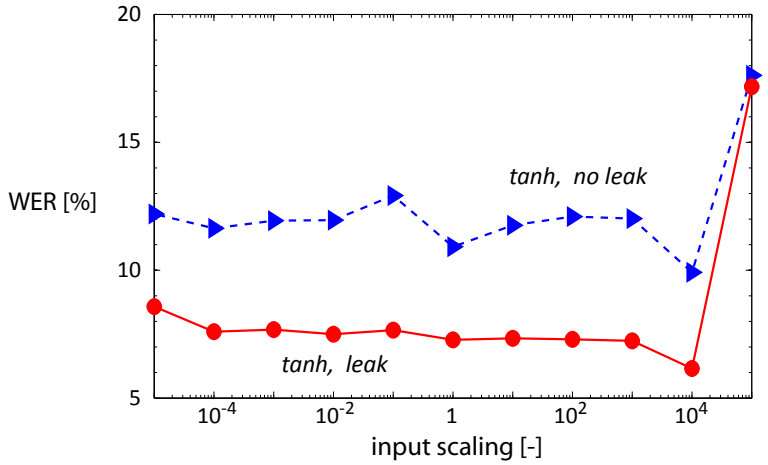


Figure 5.14: Sweep of the input scaling for two classical tanh networks without delay, with and without leak. For small input powers the performance stays almost the same and just before results deteriorate, a small optimum is present for a scaling of 10^4, which corresponds to powers around 10 W. In all the simulations of this chapter a scaling of 1 was used, corresponding to mW powers. For both simulations the optimal $(\rho, \Delta\Phi)$ combination was used after a $(\rho, \Delta\Phi)$ sweep for an input scaling equal to 1.

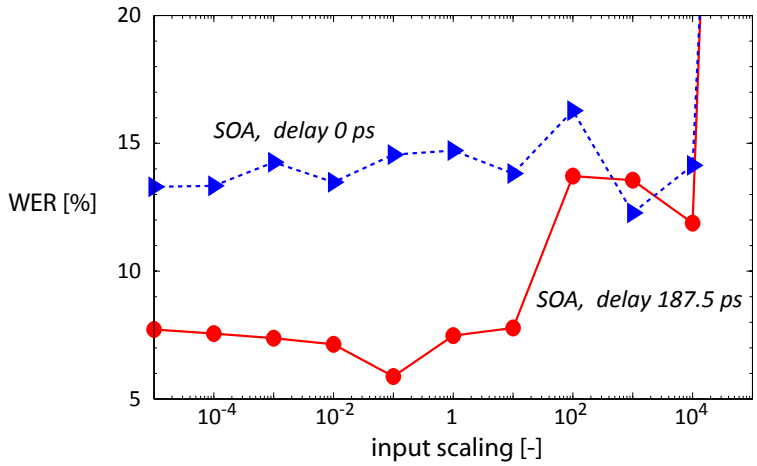


Figure 5.15: Sweep of the input scaling for two incoherent SOA reservoirs, one without delay and one with optimal delay (187.5 ps). The behavior for short delays is the same as for tanh networks, for the longer delay the range of optimal input scalings shifts to smaller values. In all the simulations of this chapter a scaling of 1 was used, corresponding to mW powers. For both simulations the optimal $(\rho, \Delta\Phi)$ combination was used after a $(\rho, \Delta\Phi)$ sweep for an input scaling equal to 1.

5.2.4 Input scaling

In general, we find that besides interconnection delay, coherence, and phase shift combined with carrier dynamics for small delays, all other behavioral differences between our implementation and traditional tanh reservoirs, e.g., gain saturation, have a negligible influence on performance for the input signal properties considered in these experiments. Indeed, in our experiments, all nodes were operating mostly in their low-power regime, for which the steady state curves for tanh and SOA are very well matched and almost linear (Figure 3.4). The bandwidth of the input signals was much lower than the frequency range of the node dynamics. The dynamic range of the node states can be changed by scaling up the power of the input signals. Selective experiments as in Figures 5.14 and 5.15, respectively for incoherent tanh and SOA reservoirs, show that results do not vary a lot for small input powers. The sweep of the input scaling was done at the optimal $(\rho, \Delta\Phi)$ combination, found after doing one time a $(\rho, \Delta\Phi)$ sweep for an input scaling equal to 1.

When increasing the input power for tanh reservoirs, a weak optimum was reached (at 10^4), before results deteriorated heavily (Figure 5.14). The point where the results become severely worse lies at a scaling of 10^5 which corresponds

	minimum
incoherent tanh with leak	6.1 %
incoherent SOA	6.5 %
coherent SOA	4.5 %

Table 5.4: The minimum results at the optimal delay for coherent and incoherent SOA swirl networks and tanh swirl networks with leak rate.

with powers around 100 W, unrealistically high when working with optical powers on a chip. For SOA reservoirs, the result is similar but for a longer delay, the situation is slightly different since the optimum is obtained for a smaller scaling (10^{-1} versus 10^{3}, Figure 5.15). Section 5.2.3 tells us that since the sweep for the input scaling was done for one specific $(\rho, \Delta\Phi)$ combination, we can expect the differences between results to be greater and partly due to the randomizations and therefore the relevance of small improvements to be minor. However, it could always be included as an extra optimization parameter when trying to find the optimal performance for a given network.

For several reasons we decided not to include this parameter as an extra sweep parameter: the improvement in the optimum is rather small (Table 5.3); the optimum is quite sensitive and dependent on the delay — Section 5.2.3 also tells us that differences between results tend to be greater when simulations are done at one specific $(\rho, \Delta\Phi)$ point; adding an extra sweep parameter consumes a lot of simulation time and resources and we are more concerned about the relative differences between different networks. The value 1 was chosen because then the input lies in the low-power regime where the results are not very sensitive to the input scaling (Figures 5.15 and 5.14).

As a summary of the most important curves so far, Figure 5.16 compares the optimal results achieved with classical networks (with leak) and SOA reservoirs (with and without coherence). Since the two most important parameters, delay and phase control, appear to be orthogonal (i.e., they can be optimized independently), the best design strategy consists of optimizing the delay first and providing a phase tuning mechanism. Under these conditions, SOAs with coherence at an optimal delay perform better than classical networks with leak rate, not only at the optimal delay but also for a relatively wide range of delay values around the optimal delay. This is also summarized in Table 5.4 showing that optimal coherent SOA reservoirs achieve error rates that are approximately 1.5 % lower. Although incoherent SOA networks are outperformed by coherent SOA networks, making their performance comparable to that of traditional leaky tanh networks, the measurement and fabrication tolerances are considerably

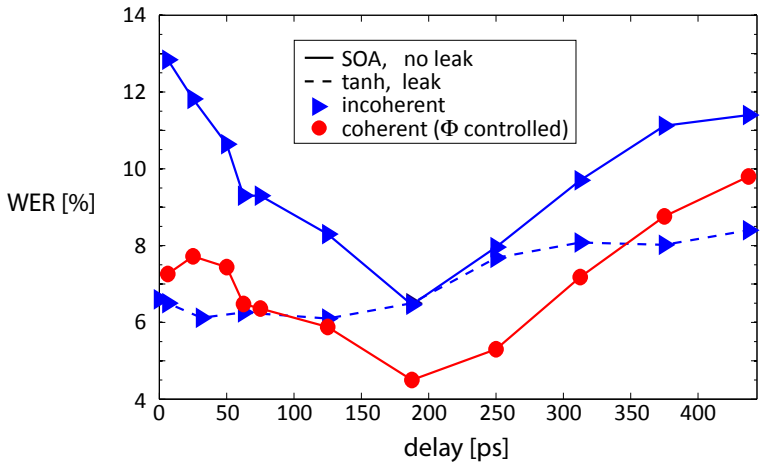


Figure 5.16: Incoherent and coherent phase-controlled SOA networks versus classical networks with leak rate. All reservoirs used a swirl topology with positive weights.

more relaxed when incoherent light is used, making this option an interesting candidate for an initial small-scale prototype.

5.2.5 Signal speed

In the beginning of this chapter, Section 5.1.3, we said that the rate at which the audio signals are fed into the photonic reservoir is matched to its typical time scales. It is however interesting to know whether our delay results depend completely on the speech signal itself. In that case we expect the optimal delay to shift according to the input rate. This is what happens as can be seen in Figure 5.17 where the minimum results of $(\rho, \Delta\Phi)$ sweeps are shown for rates two times slower and two times faster than our earlier experiments. The (a) part shows that, as the speed increases or decreases, the optimal delay shifts as well. In (b) we have rescaled the X-axis for the different curves the same way as their input rate, so their X-axes all match that of the original one (blue curve). Here the similarity between the three graphs implies that the optimal delay is indeed a feature of the audio signals themselves as it shifts according to the input rate. This also means that, confronted with a hardware implementation of a photonic reservoir with certain delays, we can change the input rate to match the delays. This answers the question from the end of Section 5.2.2: we show here that input speed and optimal delay are closely related. For slower signals there is no reason why this should break down, although there are practical limitations to delay lengths on a chip since losses increase with length and they consume area. For

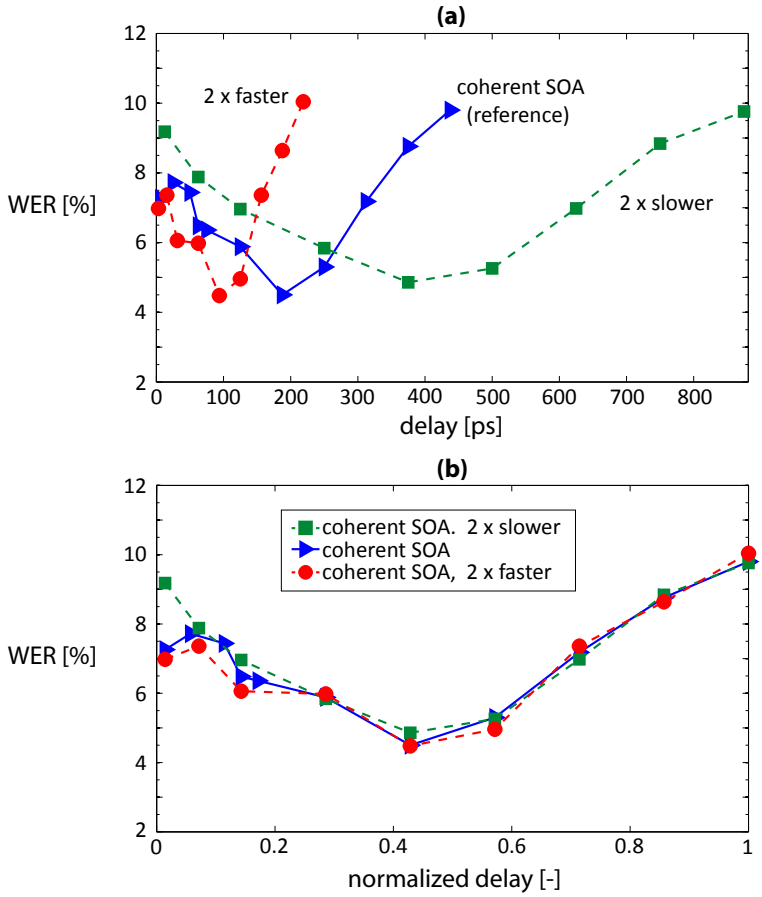


Figure 5.17: (a) Results for different speeds of the input signals fed into a coherent swirl SOA reservoir, (b) the same results but plotted on top of each other by changing the X-axis of the different curves the same way as their input rate.

ever faster signals there is first of all the issue of reaching the limitations of our present SOA model, although the spectral bandwidth of a two times faster audio signal still falls mostly within a 10 GHz bandwidth, hence within the validity of the model (Figure 3.7). Second, at some point the delay in the SOAs will dominate the interconnection delays.

These results confirm earlier findings about the importance of time scales [36], where it was shown that there is a link between input resampling and leak rate as there is here a relationship between input resampling and optimal delay.

The curves in Figure 5.17(b) reveal a divergence in performance for different

input speeds at small delays. The faster the input, the better the result at small delays. This again confirms that the carrier dynamics plays a role next to the delay. Since the time scale for the carrier lifetime τ_c is fixed, its effect will be stronger for faster signals and the same relative delay (fraction of the total speech signal), since it corresponds to shorter times in practice.

5.2.6 Input current

In the previous simulations we used an SOA input current of 186.7 mA corresponding to 13 dB steady state gain or a factor of around 20 and varied the attenuation in the connections. Figure 5.7 showed that similar changes in the currents of the SOAs yielded similar results, but if we increase the current and corresponding gain a lot to 288.6 mA and 30 dB gain respectively, the performance is a lot worse at the same input scaling (Figure 5.18). We changed the attenuation range in the connections in such a way that the $(\rho, \Delta\Phi)$ parameter space is almost the same as in the simulations with lower input current. When we decrease the input powers with a factor 10, the optimal results become the same as for the standard input current (Figure 5.18(b)), while the results averaged over the phase are only slightly worse. This implies that the optimal regions for higher input currents are smaller in the phase dimension.

An explanation as to why the results are better for 10 times smaller input powers, can be found in the different steady state characteristics for both currents (Figure 5.19). The gain remains almost the same at the standard current for input powers up to -5 dBm, which was always the case. For higher input currents the steady state is already starting to become nonlinear for input powers around -20 dBm. This region was explored with the 10 times smaller input powers. Since the results were better in this region, it means that the performance is better with near linear nodes and that memory, rather than nonlinearity, is critical for this task.

Note that with smaller input powers, the system also becomes more sensitive to noise (Section 3.1.4). It is therefore better to operate at smaller input currents, using sufficiently large signal powers.

5.2.7 Network size

In Section 5.1.3 we chose a network size of 81 nodes as a compromise between a size larger than the 77 channels of the Lyon model and a small reservoir for computational reasons. Figure 5.20 shows on the one hand that results quickly deteriorate for sizes smaller than the number of channels and on the other hand that it is possible to improve the performance by going to larger SOA reservoirs, although the observed improvement saturates. The top curve has the results averaged over the phase, the bottom one the phase-controlled results.

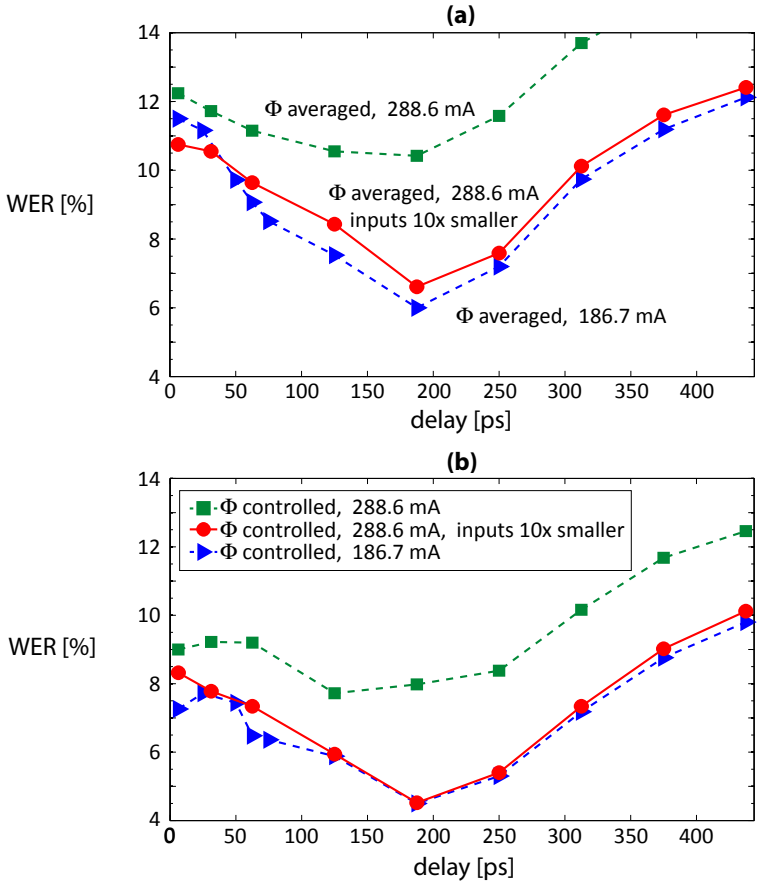


Figure 5.18: Results for different input currents for coherent swirl SOA reservoirs: (a) results are averaged over the phase; (b)results are taken at the optimal phase (phase-controlled). All previous simulations were done with a current of 186.7 mA (triangles). The results for a higher current (288.6 mA) are worse, unless the input is decreased with a factor 10. In that case the optimal results are almost the same as for a current of 186.7 mA, but the average results are worse. This means that the optimal region is smaller at higher currents.

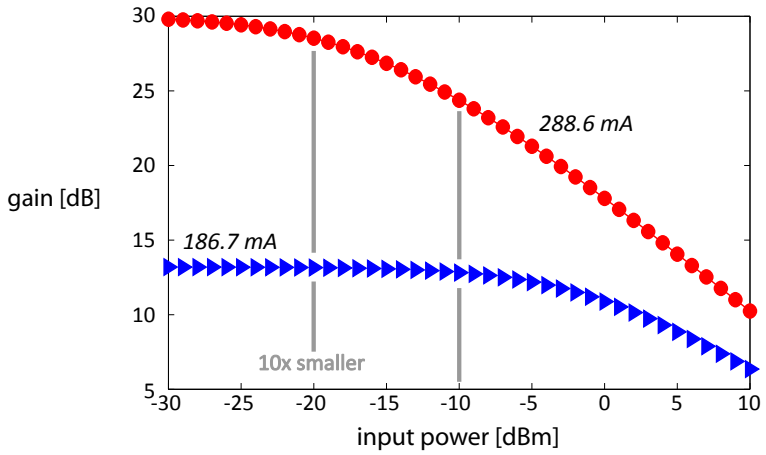


Figure 5.19: The steady state gain in dB plotted against the input power in dBm for different input currents for an SOA. This log-log plot shows that a higher current for SOAs results in a gain that is only linear for very small input powers. The vertical gray lines show the maximum input usually present: the right one is for most simulations in this chapter, the left one for inputs 10 times smaller. For the standard current of 187.6 mA this lower input hardly affects the gain. For the higher current, however, the gain is 5 dB decreased for the standard inputs and only 1.5 dB for tenfold smaller inputs.

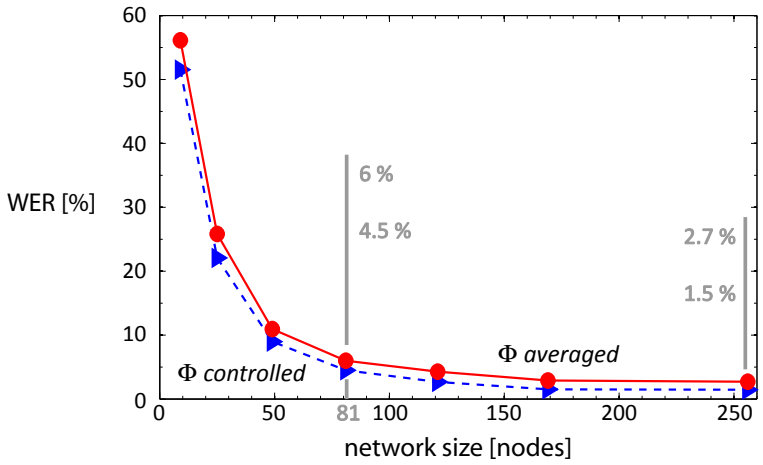


Figure 5.20: The performance of a coherent swirl SOA reservoir for different network sizes with optimal delay. The upper curve (solid–circles) shows the performance after averaging over the phase, the bottom curve (dashed–triangles) the optimal performance at the best phase. The difference remains almost constant around 1.5 % for sizes larger than 50 nodes. Increasing the network size helps to improve performance, although the gained advantage decreases for large networks. In this chapter we used a size of 81 nodes and the corresponding results are shown at the left gray line; the results of a 250 nodes network are shown at the right gray line.

This one is always better with the difference between the two curves remaining almost constant around 1.5 % for reservoir sizes larger than the number of Lyon channels.

Since the WER becomes so small for large reservoirs, it is interesting to use a log-scale for the Y-axis. Figure 5.21 compares incoherent tanh reservoirs with different topology (swirl versus random) and different memory (leak versus optimal delay). For any network size, the performance of networks with leak rate and networks with optimal delay is always similar. However, the results for different topologies start to diverge for sizes larger than 100 nodes when the WER for random topologies levels off around 5 %, while swirl topologies continue to have a decrease in WER down to 1 % (for reservoirs with 900 nodes and larger). Our previous conclusion that topology is not an important parameter, no longer holds for larger network sizes and it seems that some structuring of the network topology is beneficial for this task. Similar conclusions for a different task, where structuring of the network was done by clustering, have been reported [37]. Investigating whether the swirl topology really improves robustness in reservoir

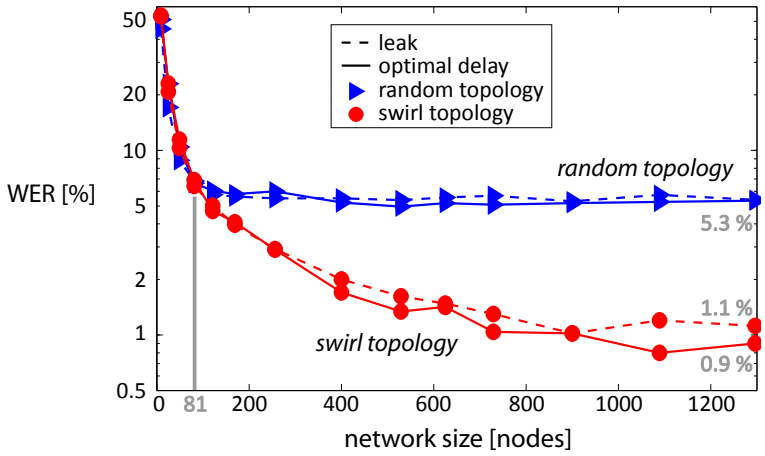


Figure 5.21: The performance of incoherent tanh networks with random and swirl topology for different network sizes. Random topologies have a better performance for small reservoir sizes, but their performances levels off at around 5% for sizes larger than 100 nodes. Reservoirs with a swirl topology on the other hand achieve lower WERs (down to 1% for reservoirs larger than 800 nodes). Leak rate and working with the optimal delay are equivalent for both topologies and any network size. The Y-axis has a log scale to better visualize the differences in performance for small WERs.

computing is an interesting topic for future research.

Figure 5.22 shows results for reservoirs which all have a swirl topology but with different node types and with and without coherence. All reservoirs show that, up to a certain size, a performance improvement can be obtained by using larger reservoirs. However, the error rates for coherent reservoirs decrease considerably faster than for incoherent reservoirs and therefore they reach their somewhat saturated optimal performance for smaller network sizes. If the network size is large enough, incoherent tanh reservoirs (as opposed to incoherent SOA reservoirs) achieve eventually almost the same low error rate as phase-controlled SOA reservoirs[12]. It could have been that the faster improvement in performance (in function of reservoir size) of phase-controlled reservoirs was due to a doubling of information in the system, since complex amplitudes could have twice the degree of freedom of power values. However, there is no clear-cut relationship.

[12] Due to computational constraints, coherent reservoirs were not investigated for sizes larger than 900 nodes.

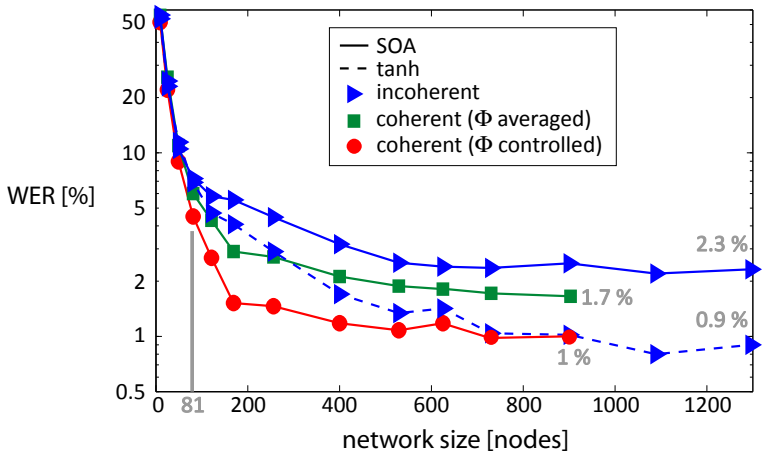


Figure 5.22: The performance in function of network size for swirl reservoirs with optimal delay and incoherent tanh, incoherent SOA or coherent SOA nodes. The most important observation is that all reservoirs have a decreasing WER in function of network size, which saturates at some point. However, the results of coherent networks drop considerably faster. Eventually, incoherent tanh reservoirs reach the best performance but only for large network sizes. The Y-axis has a log scale to better visualize the differences in performance for small WERs.

5.2.8 Babble noise

In the beginning of this chapter, Section 5.1, it was stated that babble noise with a SNR of 3 dB was added to the speech signal to make the task more complex. However, to compare with the state-of-the art, we also simulate the task once without babble noise. In this case, the trend of the results in function of delay is the same as before (Figure 5.23). By tuning the network to its optimal delay the performance improves from 1.1 % to 0.6 %. If we normalize the two curves relatively to their maximum value, the trend is actually stronger for speech without babble noise (Figure 5.23(b)). In his PhD thesis D. Verstraeten achieved an optimal performance of 0.5 % for a LSM reservoir of 1232 neurons, which means that we are not far from this result with a network of only 81 nodes. Section 5.2.7 showed that the performance could be further improved by going to reservoirs with larger sizes.

5.3 Robustness

This section describes the influence of two 'noise' mechanisms in photonic reservoir computing and speech recognition: process variations and ASE. Process variations are imperfections that arise during fabrication, while ASE is a noise source coming intrinsically from the SOAs themselves.

5.3.1 Process variations

The previous simulations were done under the assumption that all connections had the same length and hence they respond in the same way to, e.g., a wavelength shift. On a real chip this is never the case, since small process variations always occur. In Figure 5.24(a), we show results of a (ρ, λ_0) parameter sweep of a coherent SOA network with an interconnection delay of about 50 ps (i.e., an interconnection length of about 5 mm, depending on the material system). This corresponds roughly to the delays on a chip prototype that is currently being fabricated (Section 6.1.2). In such a design, process variations of about 100 nm are realistic. This means that, for this design, the variations are 4–5 orders of magnitude smaller than the total length of the connections and the resulting time differences because of this are negligible (fs range). The phase variations, however, are not. Looking back at Equation 3.33, the interconnection phase shift is determined by the vacuum wavelength λ_0, the interconnection length d and the effective index n_{eff}, which is in itself also wavelength dependent. If d is the same for all the connections as in Figure 5.24(a), the phase change for every connection will also be the same, but with length variations the impact of wavelength is felt differently for each interconnection.

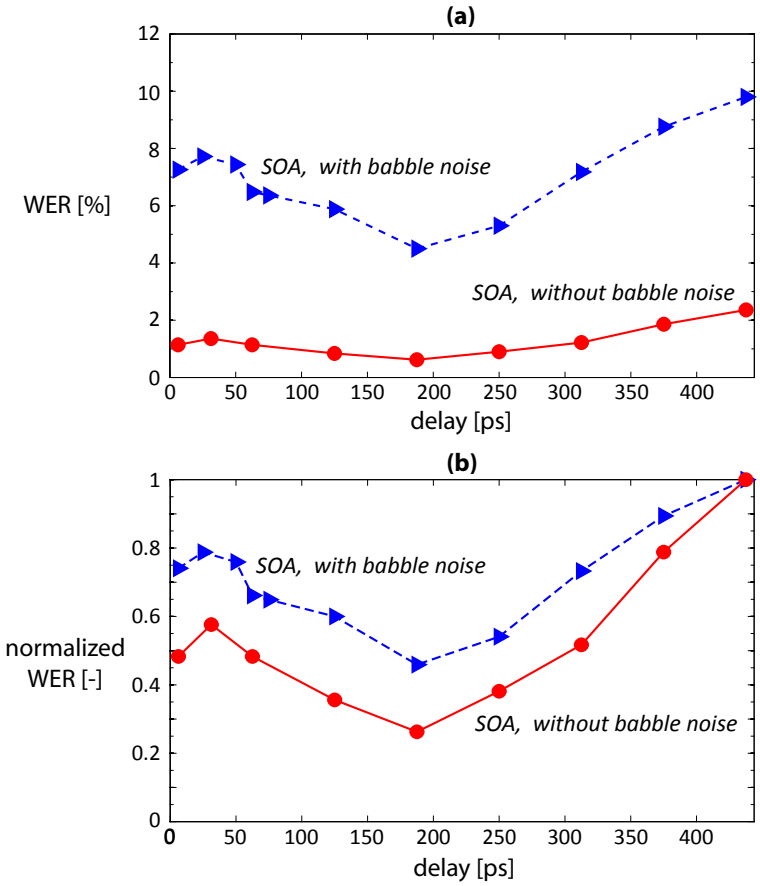


Figure 5.23: (a) Minimum results for coherent swirl SOA networks with and without babble noise added to the speech signal. The trend with optimal delay is the same for both with an optimal delay around 190 ps. (b) The same results but relative to their maximum value. The trend is relatively stronger when babble noise is not added.

σ	phase-controlled	averaged over the phase
0 nm	7.3 %	9.7 %
100 nm	7.9 %	9.6 %

Table 5.5: The minimum results for networks with 0nm and 100nm standard deviation σ on the connection lengths.

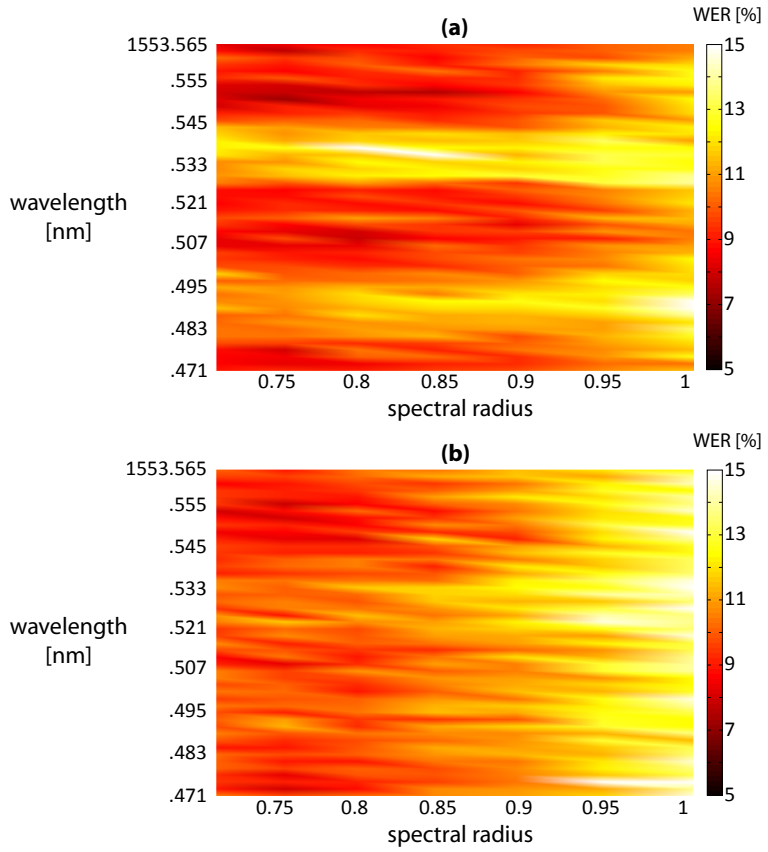


Figure 5.24: Two reservoirs with 50 ps delay. (a) all interconnections have the same length; (b) all interconnections have small length variations with a standard deviation of 100nm (the Y-axis shows the wavelength which affects the phase change in all the connections).

Figure 5.24(b) shows the impact of interconnection length variations on the performance results obtained by a (ρ, λ_0) parameter sweep. Except for variations on the nominal interconnection length, the setup for generating this figure was the same as for Figure 5.24(a). The length variations were randomly selected from a zero-mean gaussian distribution with a standard deviation of 100 nm. For all individual simulation runs, the same length values were used as would be the case for a practical implementation: once a chip is made, the fabrication imperfections remain the same. The Y-axis for both figures shows the same wavelength range. For (a) this corresponds with the same phase scan $[-\pi, \pi]$ as in previous simulations , but for (b) the phase scan will be different for every connection since the length variations will result in random phase additions.

From a visual comparison of both figures, we can conclude that the performance has become less phase dependent. The optimal regions have become smaller and the poorly performing phase regions have almost entirely disappeared. One would therefore expect the minimum errors with length variations to be worse (decrease in optimal performance), but the minimum errors after averaging over the phase (wavelength) results to be better (increase in robustness). Although Table 5.5 confirms this intuition, the differences are fairly small, so we can conclude that small length variations do not severely affect reservoir performance in SOA networks.

5.3.2 ASE

In the previous sections, all the SOA simulations were done without any amplified spontaneous emission, to focus purely on the effects of the major design parameters. In this section we will address the influence of ASE, since in practice, it can be very strong, severely degrading the signal quality. We used a simple model where random complex amplitudes are added to the SOA outputs (Section 3.1.4).

For the experiments, we took a coherent swirl SOA network with short and optimal delay, and selected the optimal $(\rho, \Delta\Phi)$ parameter combination from a $(\rho, \Delta\Phi)$ sweep without ASE. The input scaling was the same as in the previous experiments with input powers always below 1 mW. This network was used in Monte Carlo simulations with 1000 runs in which the mixing of the inputs and the cross-validation were varied from run to run. The results are shown in Figure 5.25 for different ASE powers (i.e., different filter bandwidths B_0 as in Figure 3.6). The left part of the graph corresponds to the unrealistic case of no ASE (put at 1 on the logarithmic X-axis for easier comparison), the right part of the graph corresponds to unfiltered ASE.

It is obvious that the effect of ASE can be detrimental to the performance of the reservoir and that careful considerations are needed to mitigate its influence. The two approaches discussed are both shown in Figure 5.25. In case the inputs

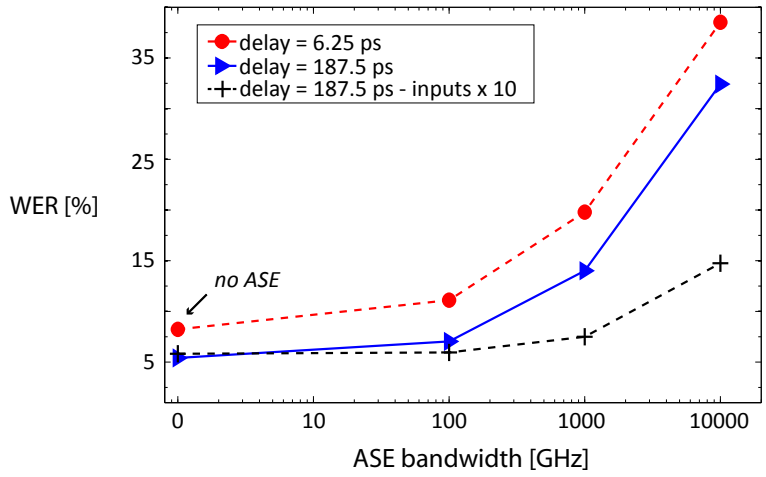


Figure 5.25: Different intensities for ASE noise are compared for two different coherent SOA swirl reservoirs — one with a short delay, the other with optimal delay. It shows the mean of all the results over 1000 runs. The left value corresponds to no ASE, the right value to unfiltered ASE. For the network with optimal delay, scaling the input 10 times higher makes the results less noise sensitive.

are scaled ten times higher, the degradation for unfiltered ASE is much lower. Using increasingly narrower filters up to 100 GHz[13] or narrower also limits the degradation. In practice some kind of filtering will be necessary to achieve the optimal performance. Integrated filters that are rather broadband with a 3 dB bandwidth of around 1000 GHz (or 8 nm) could be made for example with echelle gratings on the SOI platform [38, 39]. Narrow filters, however, complicate fabrication since all of them will often have to be tuned [18–21] or trimmed [40] afterwards to make them operate at the same filter wavelength. Filters based on high-order ring resonators with bandwidths around 1–2 GHz have been demonstrated on SOI with active tuning [41]. Alternative components for photonic reservoirs such as nonlinear coupled cavities, however, will not be susceptible to ASE, but they are a topic of future research.

[13] 100 GHz is a filter bandwidth that is for example used in the G.694.1 standard of the International Telecommunication Union (ITU) for Dense Wavelength Division Multiplexing (DWDM) channel spacings (http://www.itu.int/rec/T-REC-G.694.1-200206-I/en). This standard is used to determine which wavelengths and their spacing are to be used to send information over optical fibers. The same standard specifies channel spacings up to 12.5 GHz.

5.4 Training

In this section we will discuss some modifications to the training procedure that could be interesting when working with photonic reservoirs. So far, the training methods used were the same as for classical ESN reservoirs. However, with photonic reservoirs two things are fundamentally different: delay in the network and coherence of the signal.

The training window, here defined as the part of the signal used for training, is normally the same as the signal window, i.e., it starts at the beginning of the signal and ends when it stops (Figure 5.26). Delays in the network could be accounted for by using a different training window and that will be discussed in the first part of this section.

Although we work with coherence and complex signals, only the power values are ultimately fed into the readout part, thus throwing away some information. The second part of this section discusses different approaches.

5.4.1 Training windows

The training window used so far is as long as the audio signal (Figure 5.26). The longer the delays in the network, the longer it takes before the signal notices feedback. If the delay is as long as the audio signal itself, there will not be any effect from feedback during the training window. This corresponds to most of the results shown so far at the longest delay. The performance at those long delays is usually one of the two regions with the worst performance — the other one being at very short delays. So what if we would extend the training window to beyond the audio signal, to capture more of the feedback interactions? One approach is to shift the training window with some delay $\Delta t'$, the other one is to simply extend the training window. Both approaches are shown in Figure 5.26.

The first approach is to shift the training window (Figure 5.26(a)). We have tested the hypothesis that it would be best to shift it over the delay in the network by simulating different shifts of the training window at the optimal point in $(\rho, \Delta\Phi)$ space ($\rho = 0.85, \Delta\Phi = -0.43\pi$) obtained from the sweep in Figure 5.8(b) for a coherent swirl SOA reservoir with the optimal delay of 187.5 ps. The simulation is run as long as the shift of the training window requires by padding the inputs with zeros. For a zero window shift or extension, one would expect the results to be identical. This is not the case because of the greater variability when working at one $(\rho, \Delta\Phi)$ point as shown in Section 5.2.3. This also means that the weak optimum for a shift around 100 ps is probably not relevant (Figure 5.27 – dashed-circles-red). This optimum is even worse than the optimum of the $(\rho, \Delta\Phi)$ sweep (horizontal gray line) for a zero window shift (Table 5.6). Therefore we conclude that the difference in training results with shifted training windows are purely due to training variances and that it does not yield better performance.

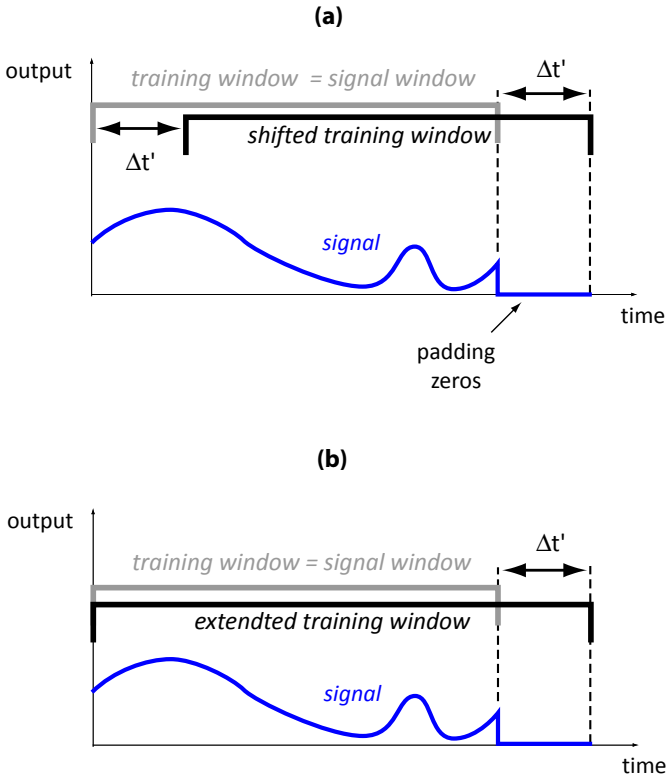


Figure 5.26: Two training windows that differ from the standard one which coincides with the actual signal: (a) shifted training window where the whole window is shifted over a certain time $\Delta t'$; (b) extended training window where the window is prolonged with a certain time $\Delta t'$. In both cases the signal is padded with zeros for as long as the training window shift or extension requires.

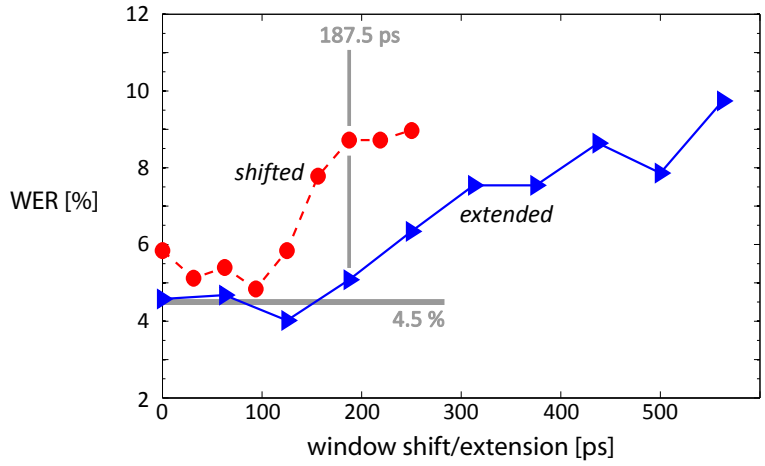


Figure 5.27: The horizontal gray line shows the optimal result for a $(\rho, \Delta\Phi)$ sweep of a coherent swirl SOA reservoir with optimal 187.5 ps delay (as in Figure 5.8(b)) which had $(\rho = 0.85, \Delta\Phi = -0.43\pi)$ as optimal combination. The curves are results for different training windows for the same reservoir with that delay (the corresponding window shift/extension is marked by the vertical gray line) and optimal $(\rho, \Delta\Phi)$ combination. The dashed-circles curve is the result for shifted windows, the solid-triangles one for extended windows. Both show that the performance stays the same up till around 125 ps, only to deteriorate strongly for longer extensions or shifts.

	minimum
after $(\rho, \Delta\Phi)$ sweep, window shift = 0 ps	4.5 %
optimal $(\rho, \Delta\Phi)$ point, window shift = 0 ps	5.8 %
optimal $(\rho, \Delta\Phi)$ point, optimal window shift = 93.75 ps	4.8 %

Table 5.6: Minimum results for a coherent SOA swirl network with an optimal interconnection delay of 187.5 ps. The first result is taken from a full $(\rho, \Delta\Phi)$ sweep without any shifting of the training window. The other two results are at the optimal $(\rho = 0.85, \Delta\Phi = -0.43\pi)$ point of that sweep without and with optimal window shift.

The results even deteriorate for shifts beyond 100 ps and this means that shifting the training window as much as the delay is not advisable. Perhaps it is more interesting to look at this result from the perspective of a practical measurement setup. There appears to be a tolerance about the exact location of the training window relative to the signal, as long as it does not deviate too much. This means that one can miss some of the initial response and still get a similar performance.

When shifting the training window some information is discarded and some added, but when extending the training window as in Figure 5.26(b) information is only added, so we expect this to yield better results than the previous approach. The same simulation at the optimal delay of 187.5 ps was done as for the shifted training windows (Figure 5.27 – solid-triangle-blue). For an extension of 125 ps a weak optimum is obtained and this time it is slightly better than the optimum for the full $(\rho, \Delta\Phi)$ sweep (Table 5.7 and horizontal gray line in Figure 5.27). Extending training windows was also tried for a network with a non-optimal delay, in this case 312.5 ps (the $(\rho, \Delta\Phi)$ sweep is shown in Figure 5.8(c)). For the same window extension (125 ps) a weak optimum is present, slightly better than the result for the $(\rho, \Delta\Phi)$ sweep (Figure 5.28 and Table 5.8). For both delays the results were obtained with simulations at one $(\rho, \Delta\Phi)$ combination. To really verify the existence of this optimum and of its dependence on the delay or audio signal, full $(\rho, \Delta\Phi)$ sweeps are required. Since the improvement is rather small, this was not pursued further in this work.

We can say that extending the training window too much deteriorates the performance (Figure 5.28). The reason is probably because the readout is time-invariant: the classifiers are trained for the whole length of the reservoir states, with one weight per node. After the end of the audio signal, the amplitude of the reservoir states decreases since they receive less input. The weights are fixed over time, and they can therefore be trained to work well for the initial or final average power of the states but not for both. This drop in amplitude can be seen

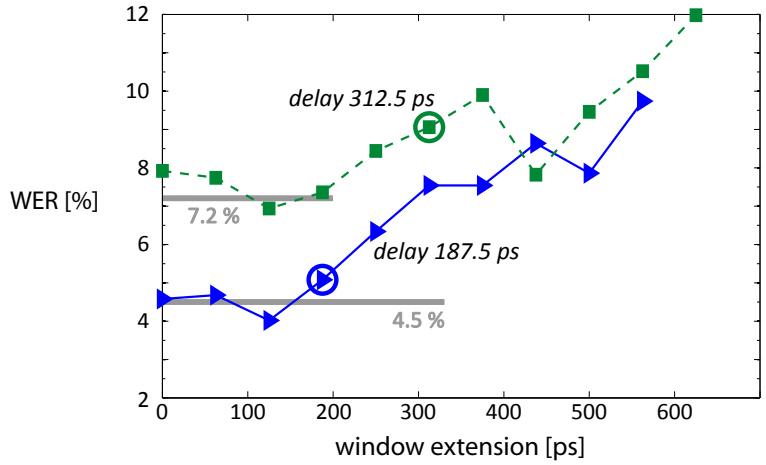


Figure 5.28: This shows the results for extending the training window for two different coherent swirl SOA reservoirs: one with delay 187.5 ps, the other with delay 312.5 ps (this delay is marked with a circle around the corresponding window extension data point). The optimal result after a $(\rho, \Delta\Phi)$ sweep (Figures 5.8(b) and (c)) is shown by the two gray horizontal lines. The corresponding optimal parameter combinations are $(\rho = 0.85, \Delta\Phi = -0.43\pi)$ and $(\rho = 0.71, \Delta\Phi = -0.50\pi)$ respectively. Both show a weak optimum for a window extension of 125 ps, slightly better than the sweep optima.

	minimum
after $(\rho, \Delta\Phi)$ sweep, window extension = 0 ps	4.5 %
optimal $(\rho, \Delta\Phi)$ point, window extension = 0 ps	4.6 %
optimal $(\rho, \Delta\Phi)$ point, optimal window shift = 125 ps	4.0 %

Table 5.7: Minimum results for a coherent SOA swirl network with an optimal interconnection delay of 187.5 ps. The first result is taken from a full $(\rho, \Delta\Phi)$ sweep without any extension of the training window. The other two results are at the optimal $(\rho = 0.85, \Delta\Phi = -0.43\pi)$ point of that sweep without and with optimal window extension.

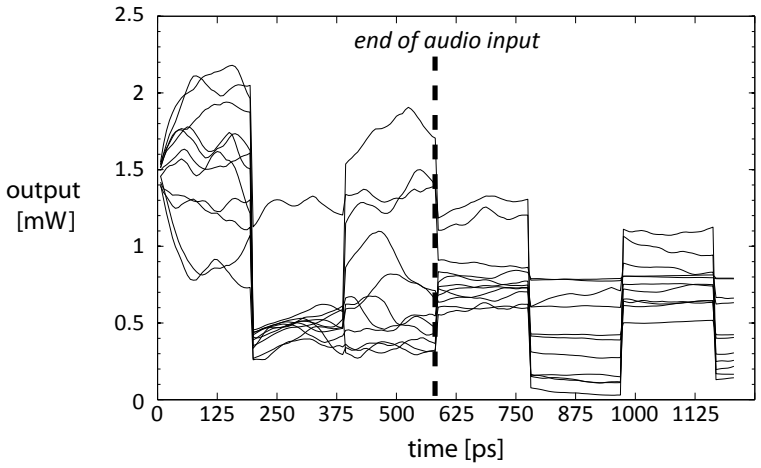


Figure 5.29: The reservoir states for a coherent swirl SOA reservoir with a delay of 187.5 ps and $(\rho = 0.85, \Delta\Phi = -0.43\pi)$. The training was extended with 625 ps (the end of the audio signal is marked with a vertical dashed line). The average power and variance of the states is clearly higher when the audio signal is still present as an input.

in Figure 5.29, together with a decrease in variance of the signals.

Extending the training window a bit hardly influences the performance, since the results stay around the optimal for a $(\rho, \Delta\Phi)$ sweep. This is again good news for the robustness of measurements on a hardware setup, since there is a tolerance, just as with shifted windows, to the length of the actual training window.

	minimum
after $(\rho, \Delta\Phi)$ sweep, window extension = 0 ps	7.2 %
optimal $(\rho, \Delta\Phi)$ point, window extension = 0 ps	7.9 %
optimal $(\rho, \Delta\Phi)$ point, optimal window shift = 125 ps	6.9 %

Table 5.8: Minimum results for a coherent SOA swirl network with an interconnection delay of 312.5 ps. The first result is the optimal result after a full $(\rho, \Delta\Phi)$ sweep without any extension of the training window. The other two results are at the optimal $(\rho = 0.71, \Delta\Phi = -0.50\pi)$ point of that sweep without and with optimal window extension.

5.4.2 Complex readout

When coherent light is used in simulations, the information is internally represented by complex amplitudes. Ideally, the training would involve finding the optimal complex weights for these complex signals. Since the present training approaches are real-valued and the intensity of the light is easily measured by a photodetector, the training was always done with real-valued powers and real weights. Coherent detection where the phase information is also extracted is possible but technologically a lot more challenging. We investigated what the influence would be of coherent detection by supplying the readout, still based on real-valued weights, both power and phase information. Table 5.9 reveals that using either the power or the phase yields similar results but combining both enhances the performance. Feeding both to the readout, doubles the number of variables available to the readout. For comparison, we also present results for a reservoir with (almost) twice the number of nodes but using only one variable type (either power or phase). We see that, for the network sizes involved, we can gain more performance by using larger reservoirs than by adding phase information. However, more research is necessary to verify whether this is a general trend. In reality, the extent to which either of the two approaches (complex readout and larger reservoirs) will be used, depends on their price and the desired performance.

Depending on the variable (power, phase or both) the readout receives, the optimal region in $(\rho, \Delta\Phi)$ space differs (Figure 5.30). With some imagination the $(\rho, \Delta\Phi)$ sweep of both power and phase can be seen as the superposition of the results when using either one. Table 5.9 shows that for both network sizes, using only phase information has a slightly better performance than using only power values. Future research could determine whether this is a coincidence or if there is a mechanism behind it. Another interesting topic for future research is implementing a true complex readout, where regression happens with complex

	global	after phase averaging
81 nodes, power only	4.5 %	6.1 %
81 nodes, phase only	4.1 %	6.1 %
81 nodes, power and phase	2.3 %	3.6 %
169 nodes, power only	1.9 %	3.5 %
169 nodes, phase only	1.5 %	2.9 %

Table 5.9: Minimum results with and without phase averaging for a coherent SOA swirl network with an optimal interconnection delay of 187.5 ps, for different complex training approaches. This is done for using only power information, only phase information or both and for two different reservoir sizes. Using both power and phase increases the performance but not as much as doubling the reservoir size.

variables and complex weights. Furthermore, this would allow to translate the readout directly to an optical hardware implementation.

5.5 Conclusions

This chapter has presented a detailed analysis of the performance of a photonic reservoir implementation based on integrated SOAs. We have systematically linked design parameters and behavioral properties of our implementation to the observed performance trends. One of the most important design parameters turned out to be the delay of the inter-node connections, a parameter that has thus far hardly been considered in traditional reservoir research. We have shown that the impact of this parameter is also found in traditional reservoirs without leak rate. A second important parameter is the phase shift in the interconnections. If this can be controlled, a significant performance improvement is achieved. Other design parameters were found to have less impact on performance. When both interconnection delay and phase shift can be optimized, our SOA implementation outperforms optimized traditional tanh reservoirs with leak rate, yielding an improvement over the state-of-the-art. Although it is difficult to compare results directly, especially for the experiments with babble noise, we achieve without babble noise a result of 0.6 % with a network of 81 nodes, while other reported results need 1232 nodes for 0.5 %.

If the phase shift can not be tuned, we still find an optimal performance similar to that of traditional simulated reservoirs. The influence of noise in

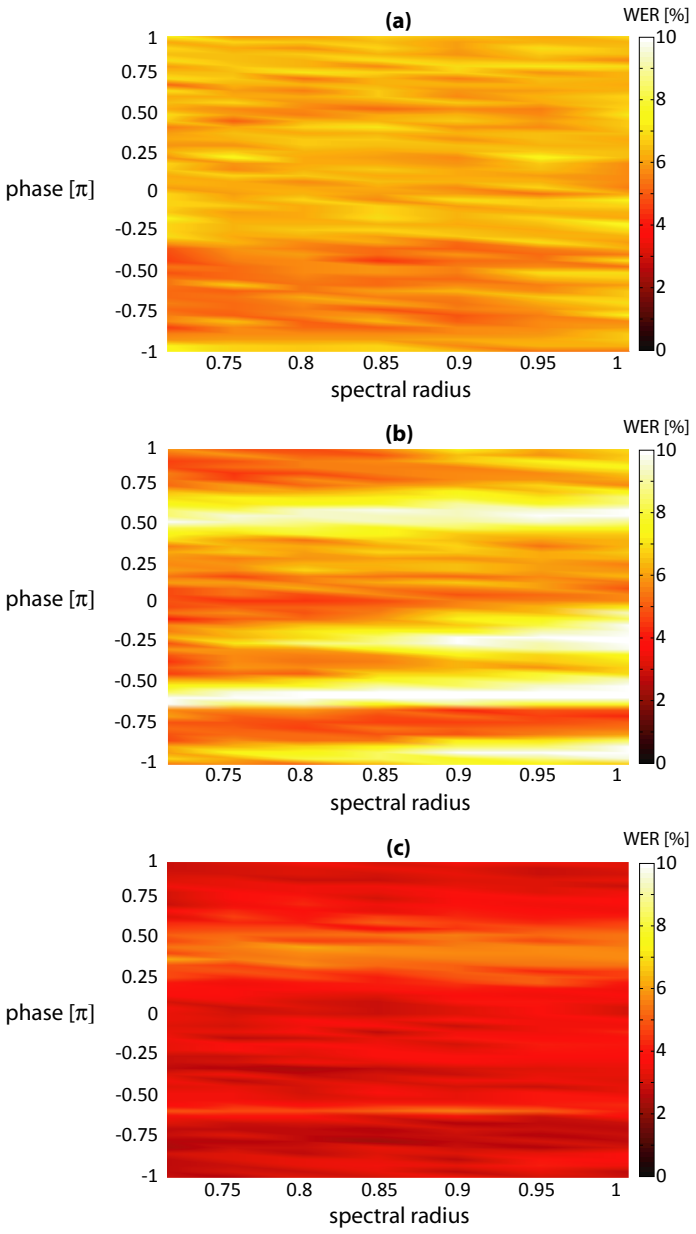


Figure 5.30: 3 different information streams offered to the readout for a coherent SOA network with optimal delay 187.5 ps and 81 nodes: (a) only the power values, (b) only the phase values, (c) both phase and power values.

the amplifiers and fabrication variations were also investigated. While process variations do not turn out to be very important, careful considerations to limit ASE will be necessary. We have found that the performance is quite robust against slightly shifting or extending the training window with respect to the signal window. With coherent detection, the performance for the same network size can be improved further. Although our analysis was performed for one specific task (recognizing isolated spoken digits with babble noise), we are confident that similar conclusions would follow for other tasks.

References

[1] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman. *Parallel Reservoir Computing Using Optical Amplifiers.* IEEE Transactions on Neural Networks, 22(9):1469–1481, September 2011.

[2] A. J. Robinson, G. D. Cook, D. P. W. Ellis, E. Fosler-Lussier, S. J. Renals, and D. A. G. Williams. *Connectionist speech recognition of Broadcast News.* Speech Communication, 37(1-2):27–45, 2002.

[3] A. J. Robinson. *An Application of Recurrent Nets to Phone Probability Estimation.* IEEE Transactions on Neural Networks, 5(2):298–305, 1994.

[4] A. I. García-Moral, R. Solera-Ureña, C. Peláez-Moreno, and F. Díaz de María. *Data Balancing for Efficient Training of Hybrid ANN/HMM Automatic Speech Recognition Systems.* IEEE Transactions on Audio, Speech & Language Processing, 19(3):468–481, 2011.

[5] A. I. García-Moral, R. Solera-Ureña, C. Peláez-Moreno, and F. Díaz de María. *Hybrid Models for Automatic Speech Recognition: A Comparison of Classical ANN and Kernel Based Methods.* In Proceedings of Advances in Nonlinear Speech Processing (NOLISP), Lecture Notes in Computer Science, pages 152–160, 2007.

[6] M. D. Skowronski and J. G. Harris. *Automatic speech recognition using a predictive echo state network classifier.* Neural Networks, 20(3):414–423, 2007.

[7] G. R. Doddington and T. B. Schalk. *Computers - Speech Recognition - Turning Theory to Practice.* IEEE Spectrum, 18(9):26–32, 1981.

[8] A. Varga and H. J. M. Steeneken. *Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems.* Speech Communication, 12(3):247–251, 1993.

[9] D. Verstraeten. *Reservoir Computing : computation with dynamical systems.* PhD thesis, Ghent University, 2009.

[10] R. Lyon. *A computational model of filtering, detection, and compression in the cochlea.* In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 7, pages 1282–1285, Paris, France, May 1982.

[11] M. Slaney. *Lyon's Cochlear Model.* Apple Computer Technical Report #13, 1988. http://www.slaney.org/malcolm/pubs.html.

[12] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

[13] Y. Deng, S. Chakrabartty, and G. Cauwenberghs. *Analog auditory perception model for robust speech recognition*. In IEEE International Joint Conference on Neural Networks (IJCNN), pages 1705–1709, 2004.

[14] A. Liu, L. Liao, D. Rubin, J. Basak, Y. Chetrit, H. Nguyen, R. Cohen, N. Izhaky, and M. Paniccia. *Recent development in a high-speed silicon optical modulator based on reverse-biased pn diode in a silicon waveguide*. Semiconductor Science and Technology, 23(6):064001, 2008.

[15] B. R. Bennett and R. A. Soref. *Electrorefraction and Electroabsorption in Inp, Gaas, Gasb, Inas, and Insb*. IEEE Journal of Quantum Electronics, 23(12):2159–2166, 1987.

[16] R. A. Soref and B. R. Bennett. *Electrooptical Effects in Silicon*. IEEE Journal of Quantum Electronics, 23(1):123–129, 1987.

[17] N.-N. Feng, S. Liao, D. Feng, P. Dong, D. Zheng, H. Liang, R. Shafiiha, G. Li, J. E. Cunningham, A. V. Krishnamoorthy, and M. Asghari. *High speed carrier-depletion modulators with 1.4V-cm $V_\pi L$ integrated on 0.25μm silicon-on-insulator waveguides*. Optics Express, 18(8):7994–7999, 2010.

[18] V. Passaro, F. Magno, and A. Tsarev. *Investigation of thermo-optic effect and multi-reflector tunable filter/multiplexer in SOI waveguides*. Optics Express, 13(9):3429–3437, 2005.

[19] R. L. Espinola, M. C. Tsai, J. T. Yardley, and R. M. Osgood. *Fast and low-power thermooptic switch on thin silicon-on-insulator*. IEEE Photonics Technology Letters, 15(10):1366–1368, 2003.

[20] W. De Cort, J. Beeckman, R. James, F. A. Fernández, R. Baets, and K. Neyts. *Tuning of silicon-on-insulator ring resonators with liquid crystal cladding using the longitudinal field component*. Optics Letters, 34(13):2054–2056, July 2009.

[21] W. De Cort, J. Beeckman, R. James, F. A. Fernández, R. Baets, and K. Neyts. *Tuning silicon-on-insulator ring resonators with in-plane switching liquid crystals*. Journal of the Optical Society of America B, 28(1):79–85, January 2011.

[22] C. M. Marcus and R. M. Westervelt. *Stability of analog neural networks with delay*. Physical Review A, 39(1):347–359, January 1989.

[23] N. Micheal and K. Wang. *Robust stability of nonlinear time-delay systems with applications to neural networks.* IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 43(7):532–543, July 1996.

[24] S. Arik. *Global asymptotic stability of a larger class of neural networks with constant time delay.* Physics Letters A, 311(6):504–511, May 2003.

[25] Z. Wang, Y. Liu, M. Li, and X. Liu. *Stability analysis for stochastic Cohen-Grossberg neural networks with mixed time delays.* IEEE Transactions on Neural Networks, 17(3):814–820, May 2006.

[26] Y. He, G. Liu, and D. Rees. *New delay-dependent stability criteria for neural networks with time-varying delay.* IEEE Transactions on Neural Networks, 18(1):310–4, January 2007.

[27] F. Atay, J. Jost, and A. Wende. *Delays, Connection Topology, and Synchronization of Coupled Chaotic Maps.* Physical Review Letters, 92(14):144101, April 2004.

[28] J Cao, P Li, and W Wang. *Global synchronization in arrays of delayed neural networks with constant and delayed coupling.* Physics Letters A, 353(4):318–325, May 2006.

[29] M. Dhamala, V. Jirsa, and M. Ding. *Enhancement of Neural Synchrony by Time Delay.* Physical Review Letters, 92(7):074104, February 2004.

[30] X. Liang, M. Tang, M. Dhamala, and Z. Liu. *Phase synchronization of inhibitory bursting neurons induced by distributed time delays in chemical coupling.* Physical Review E, 80(6):066202, December 2009.

[31] H. Wang and Q. Song. *Synchronization for an array of coupled stochastic discrete-time neural networks with mixed delays.* Neurocomputing, 74(10):1572–1584, May 2011.

[32] T. Pérez, G. C. Garcia, V. M. Eguíluz, R. Vicente, G. Pipa, and C. Mirasso. *Effect of the topology and delayed interactions in neuronal networks synchronization.* PloS one, 6(5):e19900, May 2011.

[33] W. Bogaerts, S. K. Selvaraja, P. Dumon, J. Brouckaert, K. De Vos, D. Van Thourhout, and R. Baets. *Silicon-on-Insulator Spectral Filters Fabricated With CMOS Technology.* IEEE Journal of Selected Topics in Quantum Electronics, 16(1):33–44, 2010.

[34] W. Bogaerts and S. K. Selvaraja. *Compact Single-Mode Silicon Hybrid Rib/Strip Waveguide With Adiabatic Bends.* IEEE Photonics Journal, 3(3):422–432, June 2011.

[35] M. Osinski and J. Buus. *Linewidth Broadening Factor in Semiconductor-Lasers - an Overview.* IEEE Journal of Quantum Electronics, 23(1):9–29, 1987.

[36] B. Schrauwen, J. Defour, D. Verstraeten, and J. Van Campenhout. *The Introduction of Time-Scales in Reservoir Computing , Applied to Isolated Digits Recognition.* In Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN), volume 4668 of *Lecture Notes in Computer Science*, pages 471–479, Porto, Portugal, September 2007.

[37] S. Jarvis, S. Rotter, and U. Egert. *Increased robustness and intermittent dynamics in structured Reservoir Networks with feedback.* In M. Verleysen, editor, Proceedings of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, April 2011.

[38] B. J. Luff, D. Z. Feng, J. Fong, C. C. Kung, W. Qian, H. Liang, M. Asghari, M. Boudreau, X. Hong, D. Hui, R. Narayan, S. Rangarajan, and M. Gokhale. *Silicon Photonic Filters for High Speed Data Transmission Applications.* In European Conference on Optical Communication (ECOC), page Tu.5.C.3, Torino, Italy, 2010.

[39] J. Brouckaert, W. Bogaerts, S. K. Selvaraja, P. Dumon, R. Baets, and D. Van Thourhout. *Planar concave grating demultiplexer with high reflective bragg reflector facets.* IEEE Photonics Technology Letters, 20(1-4):309–311, 2008.

[40] J. Schrauwen, D. Van Thourhout, and R. Baets. *Trimming of silicon ring resonator by electron beam induced compaction and strain.* Optics Express, 16(6):3738–3743, March 2008.

[41] P. Dong, N. N. Feng, D. Z. Feng, W. Qian, H. Liang, D. C. Lee, B. J. Luff, T. Banwell, A. Agarwal, P. Toliver, R. Menendez, T. K. Woodward, and M. Asghari. *GHz-bandwidth optical filters based on high-order silicon ring resonators.* Optics Express, 18(23):23784–23789, 2010.

6

Toward a hardware implementation

The goal of this work has always been to demonstrate the concept of photonic reservoir computing both through simulations and an integrated hardware implementation. This chapter deals with the attempts to realize a hardware implementation. We explored two different routes: one with a pure III-V platform (Section 6.1) and one with III-V on Silicon-On-Insulator (SOI), which is the core technology used in our group (Section 6.2). Neither have thus far resulted in an actual working hardware implementation with experimental results. For the first route we designed a small reservoir on a pure III-V platform (JePPIX), and the chips are at the time of writing fabricated. Future work involves building an appropriate measurement setup. For the second route, we are in the process of getting one SOA working in Continuous Wave regime (CW) after having achieved gain in pulsed regime [1].

6.1 III-V platform

This first part describes the progress in making an integrated reservoir of coupled SOAs on a III-V platform. The design and fabrication have been completed at the time of writing. Future work involves building a setup to characterize and measure these chips.

6.1.1 Platform

In contrast with electronics, where one material system (silicon) has been dominating the entire semiconductor industry for decades, photonics lacks a dominant material platform. Several materials are used next to each other, depending on the application and each has its advantages and disadvantages. III-V compounds[1] are the dominant material system for making lasers and detectors for telecom wavelengths (0.8 μm, 1.3 μm, 1.55 μm). Binary, ternary and even quaternary compounds are commonly used and depending on the materials and their composition ratios, one can choose the lattice constant, band gap energy E_g and corresponding wavelength [2]. The lattice constant is especially important since these materials are epitaxially grown monocrystalline layer after monocrystalline layer. If there is a mismatch between the lattice constant of two materials, defects occur at the interface and they degrade the material quality.

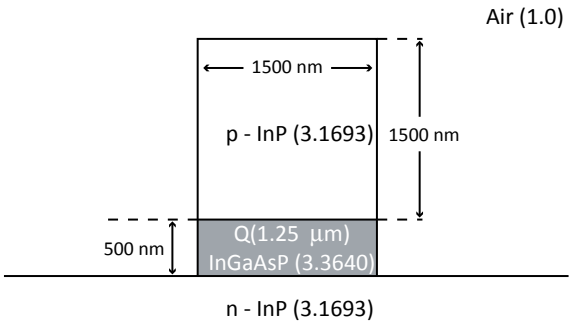
The usefulness of a material system is partly determined by its ability to combine active and passive functionality. Active functions include generating, detecting and modulating light, while passive functions include waveguiding and filtering. Active components require a material with a band gap energy the same as that of the involved photons (Figure 3.1 and 3.2). Passive components require a material with a band gap larger than the photon energy, because in that case photons cannot be absorbed (Figure 3.2(b)) and the material is transparent at that wavelength. Since these two functions have to be combined on a chip, at least two different materials with a different band gap should be present on the same chip. For a III-V platform this is done through etching and regrowing layers with a different band gap. The two basic compounds we used are indium and phosphorus, and we will therefore from here on refer to it as an indium phosphide (InP) platform.

JePPIX is the Joint European Platform for Photonic Integration of InP-based Components and Circuits[2] offering the possibility for third-party researchers to send in designs on a shared run. A small number of basic components, which can be combined and used to design complex circuits are available. The basic buildings blocks are [3, 4]:

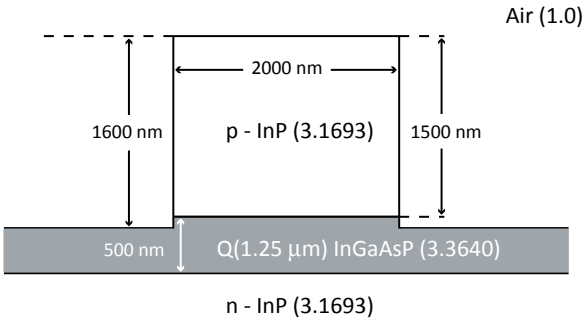
- *Passive components* such as waveguides and filters. Waveguides are either shallow or deep etched. Shallow edged waveguides have a low-index contrast which means that losses are low but it is impossible to make sharp bends, so the total size needed for structures is larger (Figure 6.1(b)). Deep etched waveguides on the other hand have a stronger index contrast which makes them ideal for sharp bends (with radii standard around 100 μm),

[1] This means that an element from group III in the periodic table is combined with an element from group V.

[2] http://www.jeppix.eu/

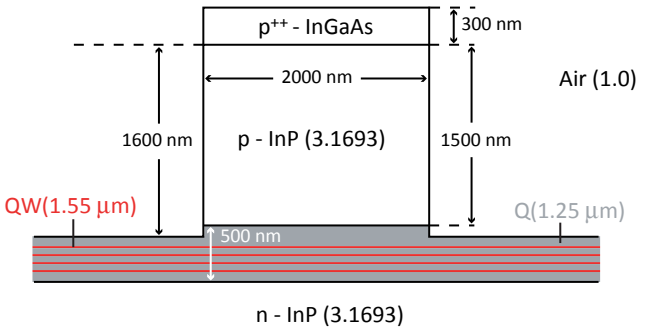


(a) deep etched WG

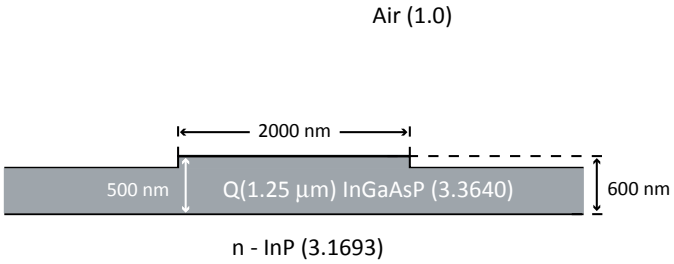


(b) shallow etched WG

Figure 6.1: Profile for the two different kind of waveguides of the JePPIX platform. The layer-stack is the same for both: on an n-doped InP substrate a 500 nm quaternary InGaAsP layer is grown with a band gap wavelength of 1.25 μm, which means that the material is transparent for light at 1.55 μm (InP itself has a band gap wavelength of 0.92 μm [2]). On top p-doped InP is grown. The refractive indices at 1.55 μm are shown between brackets for the different materials. (a) Shows the profile for a deep etched waveguide where the core is completely etched through, which causes a higher index contrast between core and cladding, which in turn allows for sharper bends and narrower waveguide widths. (b) Shows the profile of a shallow edged waveguide or so-called rib waveguide. The index contrast between core and cladding is lower in this case, so bending radii should be larger.



(a) shallow etched SOA



(b) electronically isolation section

Figure 6.2: (a) Shows the profile of the active SOA sections. It is the same as for shallow edge waveguides (chosen because its losses are lower than deep etched waveguides), with two essential differences: first of all, small quantum well layers are present in the core layer and their band gap wavelength is tuned to 1.55 μm. When carriers are supplied they act as SOAs, when carriers are extracted as detectors. To allow an easy flow of current through this stack, a highly conductive InGaAs layer is added on top of the waveguide — although in practice it is standard there, but removed for all passive sections. On top of this contact layer, Ti–Pt–Au metal is deposited for electrical contacting. The n-doped substrate is mounted on metal to allow contacting from the bottom. (b) When both detectors and SOAs are present on one chip and waveguides run between them, it is important to have a short electronically insulating section by removing the lightly p-doped InP layer.

but they have higher losses as more surface is etched, becoming a source of losses due to surface roughness. In practice all waveguides will be shallow edged, apart from bends (and between these bends and waveguides, there are low-loss shallow-edge junctions). Other passive devices include Multimode Interference couplers (MMI with standard 1×2 and 2×2 configurations available), Arrayed Waveguide Grating demultiplexers (AWGs) and tapers.

- *Phase modulators* have the same structure as the shallow edged waveguides, but with a heavily doped contact layer and electrical contacts. Injected carriers in the core InGaAsP layer alter the refractive index and allow for phase tuning of the light. We have not used these components in our design.

- *Semiconductor optical amplifiers* amplify light if forward biased and detect light when reverse biased[3]. The waveguide is shallow edged to have lower losses and there is a highly conductive contact layer on top of the stack where metal is deposited for electrical contacting (Figure 6.2(a)). In the core layer, small so-called Quantum Well (QW) layers are added, in this case a quaternary InGaAsP compound with a band gap wavelength at $1.55\,\mu$m (i.e., the desired emission wavelength). In the past bulk Q($1.55\,\mu$m) material was also used as active layer, but sandwitching very thin layers (QWs) in between Q($1.25\,\mu$m) material offers superior performance [5] and is standard with JePPIX.

  When both SOAs and detectors are desired on the same chip, and waveguides run between them, it is important to assure electrical insulation and therefore the top p-doped InP layer is removed over a short section (profile as in Figure 6.2(b)).

The platform combines the designs from different researchers on one chip and for every run 6 cells are available[4]. The first three runs have led to successful demonstrations of devices such as a mode-locked ring laser [6], a pulse-compressor [7], a fast tunable laser [4] and a nanosecond reconfigurable cyclic router [8]. In the next section we will discuss our JePPIX design.

[3] Forward biased means that the n-doped layer is connected with the negative terminal of the source. In this regime current flows easily through the stack. When reverse biased, current does not flow, unless carriers are created in the core layer. Light can create carriers and hence the resulting current is proportional to the light intensity (light detection).

[4] There are actually 9 cells, but 3 are used for testing.

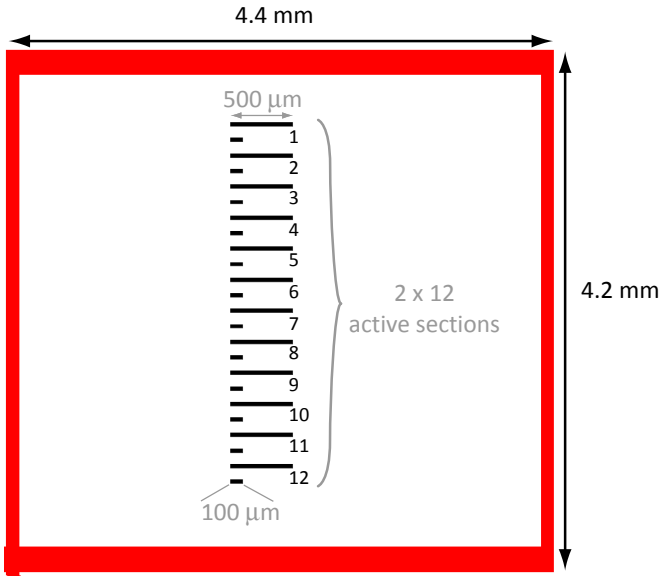


Figure 6.3: JePPIX cell for our design with a size of 4.4 mm×4.2 mm. The active regions of 500 μm and 100 μm length are predefined and there are 12 of each. The 500 μm sections will be used as SOAs, the 100 μm sections as detectors. The outer red square marks the boundaries of the chip.

6.1.2 Design

We participated on the fourth JePPIX run (called Jeppix4 from here on), and sent a design for one of the 9 cells on the chip. The cell itself had already predefined active regions. In this case 12 sections with length 500 μm and width 30 μm, interleaved with 12 sections with length 100 μm and the same width (Figure 6.3). We will use the 500 μm sections as SOAs and the short ones as detectors. The total chip size is around 16 mm^2.

Chapter 5 showed that delay is an important design parameter and since we always had the same delay length for all connections in our simulations, it was something we tried to incorporate in this design. A first decision was to make every node identical as shown in Figure 6.4, which means that the splitters and couplers are the same for every SOA. On the left side we placed one 1×2 MMI splitter, while on the right side we reserved a fourth of the power for the detector by using two 1×2 MMIs. To assure electrical isolation between the SOA and the detector, a small section will have the profile of Figure 6.2(b) with the top p-InP layer removed (green section (D) in Figure 6.4). The detector will only be useful when optical input for the SOAs is coming from the left side. The waveguides with a gray border in Figure 6.4 are deep etched waveguides and are only used

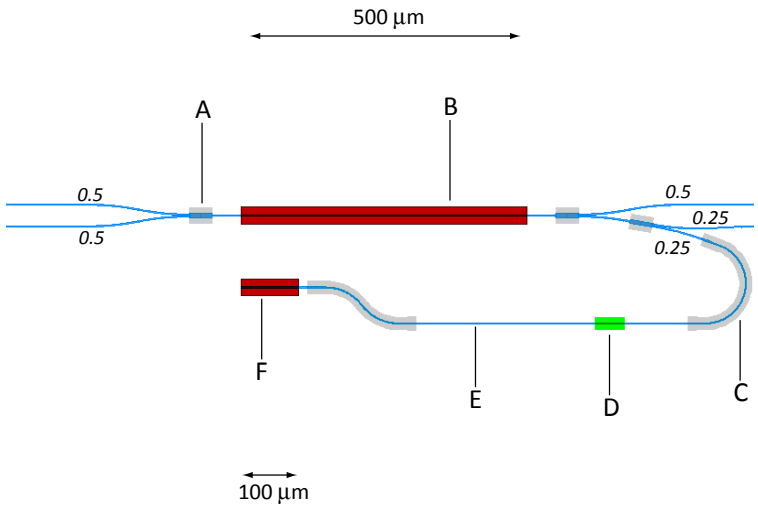


Figure 6.4: Structure of one SOA-detector pair. This will be the same for all 12 active sections of the cell (Figure 6.3). (A): a 1×2 MMI coupler/splitter with a power ratio of 0.5/0.5. The same splitter is also used at the right side and the corresponding power ratios are shown. (B): 500 μm active region, used as an SOA with a cross-section as in Figure 6.2(a). (C) deep etched waveguide for making a sharp turn. Here, the radius is 70 μm) and the profile is as in Figure 6.1(a). (D): isolation section to ensure electrical isolation between the SOA and the detector (profile as in Figure 6.2(b)). (E): shallow etched waveguide with a profile as in Figure 6.1(b). (F): active section with a length of 100 μm, used for detecting light.

when a sharp bend is required.

At the time of designing, we started using the swirl topology for our simulations (Chapter 5) and therefore we chose to implement this topology in our design. Figure 6.5 shows the swirl topology for 12 nodes in a 3×4 configuration. The numbering of the nodes corresponds with the one in Figure 6.3. As we will see in the actual design, it was not possible to have input and output waveguides from and to the chip to provide optical inputs and outputs for every SOA and this is already indicated in Figure 6.5. However, every SOA's power can be monitored electrically with the detector (F) in Figure 6.4. Although we use our network always in one direction in simulations, waveguides and SOAs are bidirectional in nature, which means that the same topology can be used in two opposite directions depending if the input is coming from the left side of the

chip (Figure 6.5(a)) or the right side (Figure 6.5(b)). As mentioned before, the detectors cannot be used when the input is coming from the right side.

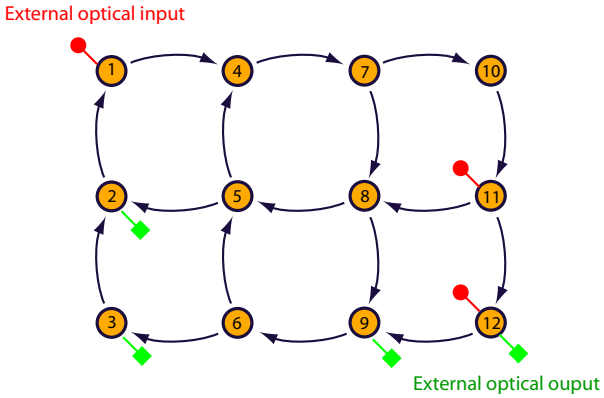
The actual implementation is shown in Figure 6.6. A first thing that is striking is the complex shape of all the connections. This was necessary for two reasons: first of all, the rectangular grid had to be translated to a different structure because of the predefined active regions and second to make the lengths of all the connections the same, some careful routing was required. The length of each connection is around 4.6 mm or 50 ps, which means that it is rather short in comparison to the optimal delays in Figure 5.17. It was however very difficult to have longer connections with the predefined active regions and minimal bending radii of 80 μm. We did not only try to have equal physical lengths but also optical ones by having as many bends with deep etched waveguides for every connection (6 times 90°). This is probably not very useful, since in reality there can even be a gradient for the refractive index over the entire chip and combined with process variations, this means that the time delay for all the connections will be almost the same, but the phase differences will be random. However, Section 5.3.1 showed that this kind of variation should not be a problem.

A consequence of the fixed configuration of every node (Figure 6.4), is that some optical inputs and outputs are not needed for the swirl topology. If these inputs and outputs are just terminated, light will reflect at their interface and it will interfere with the rest of the circuit. To avoid this, it is necessary that this light is absorbed or leaks away. We implemented this by sending the light into a structure with ever smaller bending radii, so the light is not guided anymore (e.g., a spiral with decreasing radius) or in an ever widening taper. By also keeping the highly doped and absorbing top InGaAs contact layer (Figure 6.2(a)) we get the structures in yellow (C in Figure 6.6).

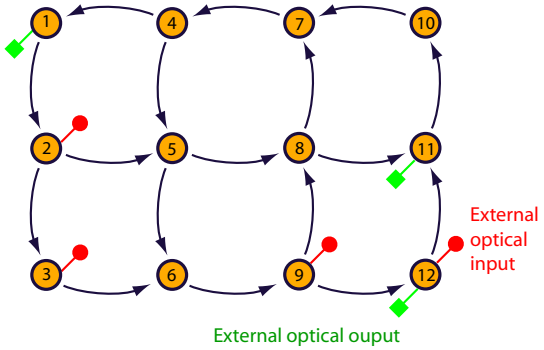
Reflections are avoided at the chip interface by having angled facets and an anti-reflection coating at these facets (as shown in Figure 3.3). Without this suppression of reflection, the SOAs would no longer be traveling wave SOAs and they could even begin to lase when the gain in the SOAs is high enough to overcome all the losses. The restriction on the number of SOAs that can be optically connected to the edge of the chip by waveguides can clearly be seen in Figure 6.6 and they are also marked in Figure 6.5.

The contacts on top of the chip for the detectors and the SOAs are p-contacts because of the JePPIX stack structure (Figure 6.2(a)). The n-contact is located at the bottom of the chip. Since applying current to 12 SOAs simultaneously is not that easy with individual probes or even with a probe card, the SOA contacts are routed towards the upper part of the chip (E in Figure 6.7), to allow them to be wire-bonded (D in Figure 6.7) to a Printed Circuit Board (PCB) and driven by a dedicated current source.

For the detectors, this wire-bonding is not always desirable when measuring



(a) external optical input from left



(b) external optical input from right

Figure 6.5: This shows a 3×4 swirl topology that can be made with the 12 SOAs available in our cell (Figur 6.3). The numbering of the SOAs corresponds to Figure 6.3. The number of external optical inputs and outputs are also indicated (since we avoided crossings, not every SOA could be reached as the actual design shows (Figure 6.6)). In reality, SOAs and waveguides are bidirectional, which means that our design corresponds to two topologies if only optical inputs from one side are considered: (a) with external optical inputs coming from the left side of the chip and external optical outputs going to the right, (b) with external optical inputs from the right side of the chip, and external optical outputs to the left. All the connections in (b) are oriented in the opposite direction as in (a). (a), as opposed to (b), allows to monitor each SOA's output power electrically with the 12 detectors.

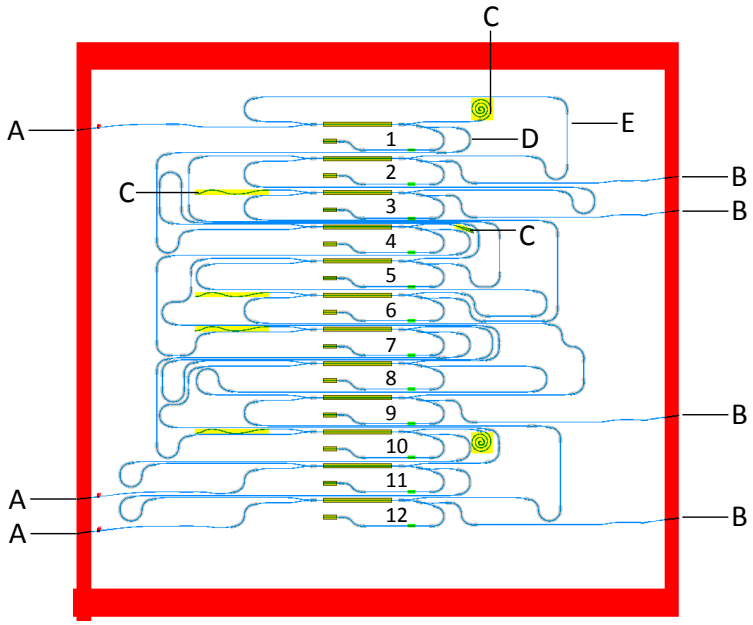


Figure 6.6: The actual chip design of the 3 × 4 swirl topology from Figure 6.5. All connections are designed to have the same length (4.6 mm), but also the same number of bends (6 times 90°) to make them as equal as possible. (A): angled facets (7°) to couple light in and out the chip from the left side with AR coatings (B): angled facets (7°) to couple light in and out the chip from the right side with AR coatings (C): Since the number of inputs and outputs are the same for every SOA (Figure 6.4), some inputs and outputs are not needed in the swirl topology. To avoid any reflections from them, light should 'disappear' and to accomplish this the highly doped and therefore absorbing contact layer (Figure 6.2(a)) remains (yellow rectangles) and the light is guided in increasingly sharper bends or a widening taper. (D) The deep etched bends for all connections have a bending radius of 80 μm (as opposed to the 70 μm for the waveguide from the SOA to the detector). (E) an example of a connection and in this case between SOA 2 and SOA 1.

high-speed signals, since the wire-bonding puts an upper limit on the signal speed (depending on the wire-bond length, ...). Special high-speed probes are used instead, and they have a special configuration with a spacing of 75–100 μm between the two pins (one for the positive and one for the negative contact, and they are often called Signal and Ground and hence SG-probes). That means we need a contact pad for the negative contact within a very close spacing of those for the positive contacts (A in Figure 6.7). This can be done in practice by wire-bonding from the bottom of the chip (n-contact) to the top (contact C in Figure 6.7). On the lower right side of the chip, test contact pads for testing the high-speed SG-probes with varying loads are present (F in Figure 6.7).

6.1.3 Measurements

At the time of writing, the fabrication of the chips was finished. Building a setup — including cooling the chip, generating and measuring high-speed input signals and driving of the SOAs by wire bonding them to a PCB[5] — to demonstrate the first photonic reservoir computing measurements on an integrated chip are a topic of future research.

Most high-speed equipment for generating and detecting high-speed signals is used for telecom applications and hence they are developed for digital experiments where all signals are periodic bit patterns. Signals such as the speech signals from Chapter 5 require an arbitrary waveform generator and a very high-speed real-time scope, which are very expensive and currently not present in our group. Therefore the first experiments will be digital and include for example training the reservoir so that it remembers the previous bit, or the two previous bits, and then also manipulations on those bits, such as taking the XOR of the two previous bits. Initial simulations have shown that a network of SOAs could be used for such a task [9].

Generating high-speed sequences of bit patterns and transferring them to the optical domain with modulators is possible up to 40 Gbit/s in our group. The inputs can be fed from both sides of the chip (A and B in Figure 6.6), although the 12 detectors are only useful when the inputs come from the left side. In that case the output of the 12 SOAs can be monitored by measuring the detectors with a high-speed probe (A and C in Figure 6.7). In principle 3 optical inputs could be used at the same time, although initial experiments will only use one as the alignment is a lot easier. The high-speed probes only allow measuring the output of one detector at a time, which means we will have to repeat the experiment 12 times with the same input to capture the response of every SOA. We have a high-speed sampling oscilloscope for periodic signals with speeds up

[5] Personal communication with K. Williams from Eindhoven, indicates that we possibly could use their printed circuit board design.

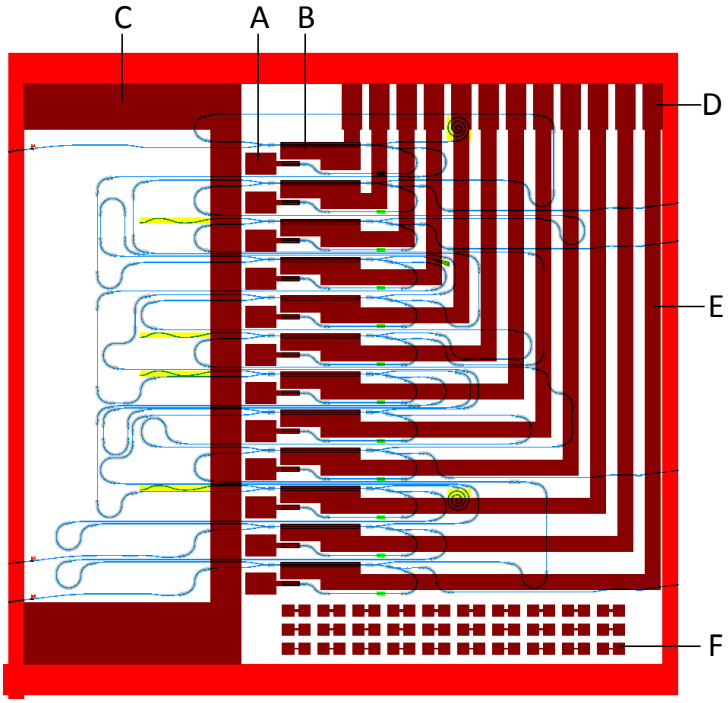


Figure 6.7: The electrical contact pads for our design. The contacts on top are p-contacts when they are connected to the active sections such as (A) for the detectors and (B) for the SOAs. The n-contact is at the bottom of the substrate. To allow contacting from the top for the n-contact as well, a large contact is placed on the left side (C) and it can be used as ground, when there are wire-bonds between the bottom of the chip and this large contact. (D) SOA contact pads to allow for wire bonding from the chip to a printed circuit board (size: 300 μm×130 μm and a pitch of 180 μm). (E) 100 μm wide contact stripes between the SOA contacts and their respective upper contact pads. (F) Test contact pads with different loads for the high speed probes.

to 100 Gbit/s. A sampling oscilloscope allows to measure signals with a higher bandwidth than with a real-time scope. A real-time scope captures the signal in one time and allows to investigate one-time transients. A sampling scope takes advantage of the periodicity of the signal by measuring only part of the signal in one period, but each period the exact sampling moment within the period is slightly shifted and hence gradually over time the entire signal is constructed repetition after repetition [10]. Our input has to be a certain bit pattern, repeated until the scope has captured the entire response to that bit pattern. The SOAs are wirebonded to a dedicated PCB and driven simultaneously with a dedicated current source and all this combined constitutes the general outline of our future setup and experiments.

6.2 Silicon platform

This section describes our efforts to make an integrated photonic reservoir with coupled SOAs on an SOI platform (one year was spent at Arai-Nishiyama lab, Tokyo Institute of Technology (TIT) in Japan for that reason). Although designs were made with networks of 4 SOAs, this research is now concentrated on making one SOA on SOI and is mostly carried out by Martijn Tassaert from the Photonics Research Group (INTEC, Ghent University-imec). At the time of writing an SOA was fabricated with a gain of around 8 dB but only in pulsed regime [1].

6.2.1 Platform

Silicon photonics has become an increasingly important platform for photonics in recent years and it has become the core focus of our research group. One reason is that silicon is the dominant material for the electronics world and the same (very expensive) CMOS processing can be used. Additionally this allows for mass-fabrication and hence lower costs, which shows for example through the typical wafer diameter of 300 mm (usually referred to as 12 inch) in the commercial state-of-the-art silicon world, while the JePPIX chips from the previous section are 2 inch. Our group coordinates a project called ePIXfab[6], similar in concept to JePPIX, but here different users can subscribe a few times a year and submit designs for silicon-on-insulator wafers (in this case 8 inch). The fabrication itself happens in either imec in Leuven (A famous electronics research center that has a pilot line that is used also for silicon photonics research) or a similar lab called LETI in Grenoble. Details about this technology can be found in the PhD thesis of Shankar Selvaraja [11].

The wafers have a 2 μm thick oxide layer which is sandwiched between a silicon substrate and a thin 220 nm high silicon layer where waveguiding struc-

[6] http://www.epixfab.eu/

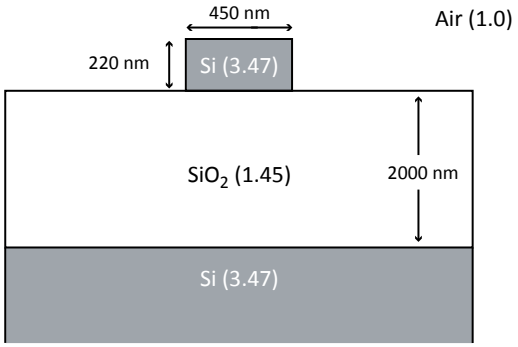


Figure 6.8: Profile of a typical (deep etched) silicon waveguide with dimensions 450 nm×220 nm for the ePIXfab platform. It rests on a 2 μm thick buried oxide cladding layer and a silicon substrate. The refractive indices for light at 1.55 μm are shown between brackets.

tures are fabricated (Figure 6.8). A second advantage of silicon-on-insulator is its high vertical index contrast in comparison to InP (Figure 6.1), since silicon has an index of 3.47 around 1.55 μm and its oxide 1.45. This means that structures are smaller (a typical deep etched silicon waveguide is 450 nm×220 nm) and bending radii up to 5 μm without significant additional loss are possible [12]. With a typical waveguide loss of 3 dB/cm, it is at least as good as typical JePPIX shallow etched waveguides. Deep etched waveguides have long been the standard waveguide on the ePIXfab platform, but recently shallow edged waveguides have been used to reduce waveguide losses tenfold to around 0.3 dB/cm, while the bends are still deep etched with low-loss taper transitions in between [13]. Shallow edged structures which reduce locally the index contrast allow also for low-loss crossings [14] and lower insertion losses in structures such as AWGs [12]. This indicates that the silicon platform allows for a dramatic reduction in size in comparison to the InP platform.

Silicon has, however, one important disadvantage in comparison to III-V materials: its indirect band gap. In Section 3.1.1 we described the importance of the band gap in semiconductors for detection, amplification and generation of light. Next to the energy of the carriers and photons however, their momentum also plays a role. When a transition over the band gap does not involve a momentum change, the band gap is called direct. Only direct band gaps allow for efficient light emission. The search for a laser and amplifier compatible with wafer-scale CMOS processes has been the holy grail for silicon photonics research this past decade. The two main approaches are heterogeneous and monolithic integration.

Monolithic integration means that different materials (with different band gaps) are grown and regrown (on wafer-scale) on top of each other . Due to lattice mismatches it has been impossible so far to grow high quality III-V materials on silicon, although this approach is still investigated [15]. Other materials have been tried such as germanium and despite the fact that successful modulators and detectors have been fabricated with that material [16–20], the laser quality remains well below that of III-V lasers [21].

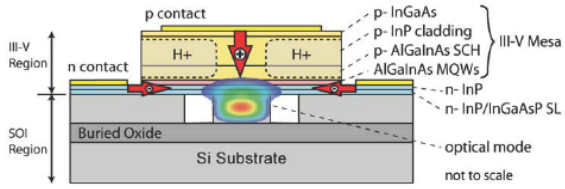
Heterogeneous integration is sticking two materials together and the two dominant methods to achieve this in silicon photonics are molecular bonding and adhesive bonding. All the passive functionality happens in the silicon, while most of the active components are fabricated in regions where the active material (mostly III-V) is bonded. Molecular bonding uses two extremely clean surfaces and brings them close enough so van der Waals forces hold them together and it has been used to make lasers [23, 27–30] , detectors [23, 31], modulators [32] and amplifiers [22] on silicon (Figure 6.9(a)).

Adhesive bonding on the other hand uses some kind of 'glue' (adhesive) between the two materials. Our group was one of the pioneers using BCB as adhesive [33] and a very good description can be found in the PhD thesis of Gunther Roelkens [34]. A definite advantage over molecular bonding is that the tolerances for surface cleaning are more relaxed and it has been used to make lasers and detectors [24, 26, 35–37] (Figures 6.9(b) and (c)). A nice overview of both bonding methods and their applications can be found in [25].

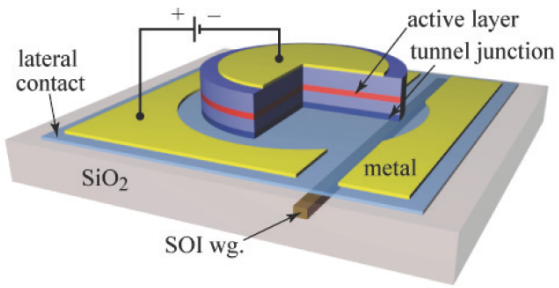
The main research focus in the past has been on demonstrating a laser, although one SOA was fabricated [22]. The overlap between the mode in the silicon waveguide and the III-V above is rather limited, so a long device length was required to achieve sufficient amplification (around 1.36 mm for 13 dB amplification at 1550 nm). Our group came up with a new concept described in the next section, to have a higher confinement in the III-V and hence a shorter device length.

6.2.2 Concept

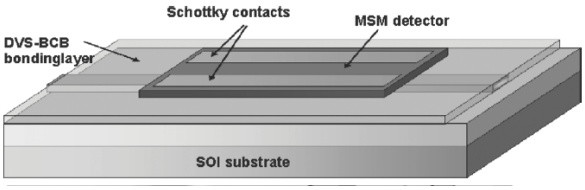
A new idea was conceived to implement SOAs integrated on the silicon platform and it is shown in Figure 6.10. Locally the silicon waveguide is first tapered broader (450 nm to 700 nm) to reduce the reflections on the interface with the III-V membrane as discussed in Section 3.1.1. To further reduce reflections and ensure a traveling wave SOA, the III-V membrane can be made with an angled facet. Next, the light is tapered back to a narrow tip with a width of 100 nm. This will squeeze the guided mode out of the silicon waveguide and into the III-V waveguide, where it will be amplified. At the other side the light is tapered back into the silicon waveguide. This ensures that all the light in the silicon gets



(a) Molecular bonded III-V on silicon



(b) Microdisk laser with BCB bonding



(c) Detectors with BCB bonding

Figure 6.9: Three examples of components made with III-V chips bonded on silicon: (a) profile of III-V chips molecularly bonded on silicon and used to make lasers and amplifiers [22, 23], (b) schematic representation of a III-V microdisk bonded with BCB on silicon [24, 25], (c) integrated detector bonded with BCB on a silicon waveguide [26].

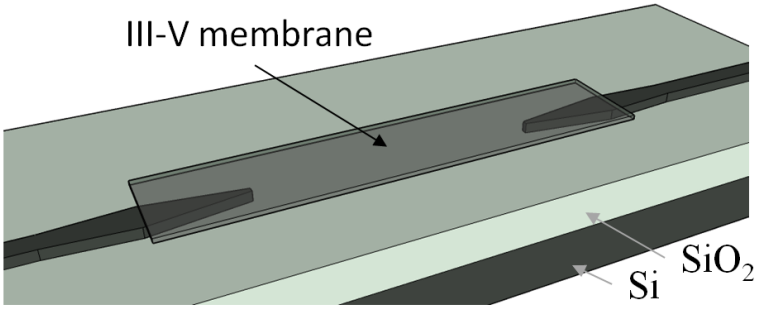


Figure 6.10: Layout of our concept for an integrated SOA on SOI [1]. The profile of the silicon waveguides is the same as in Figure 6.8. The waveguide is first broadened to a width of 700 nm, and then tapering into a small 100 nm tip. This pushes both the signal and pump light out of the silicon waveguide and into the III-V bonded on top, where the signal light gets amplified. When the III-V stripe is narrow enough (1–2 μm) it acts as a waveguide, when it is wider, a ridge waveguide structure has to be made such as in Figure 6.1(b). (picture courtesy of Martijn Tassaert)

transferred to the III-V and due to the high index contrast (there is air all around), the confinement in the III-V is high and the device length is shorter to obtain a similar gain as in [22].

The amplification is done optically with a strong pump at a wavelength shorter than the signal wavelength. These pump photons have a higher energy and hence they excite carriers through absorption (Figure 3.2(b)) that are used to amplify the signal light through stimulated emission (Figure 3.2(a)). In practice the choice of the wavelengths initially depends on the availability of sources, mostly at typical telecom wavelengths such as 1550 nm (lowest losses in optical fibers), 1300 nm (lowest dispersion in standard optical fibers) and 1480 nm (typical optical pump wavelength in EDFAs).

In the next section we will discuss the different fabrication routes we explored to implement this concept on an SOI platform.

6.2.3 Fabrication in Japan

From November 2008 till November 2009 I spent a year in Japan in the Arai-Nishiyama lab at Tokyo Institute of Technology (TIT) to make a network of SOAs on a silicon chip. Originally their focus was on III-V lasers, but in recent years they have also been investigating the possibility to bond them onto SOI and in 2007 they had a paper where they made a III-V laser on silicon with molecular

bonding [30].

They first clean both the silicon and the InGaAsP/InP chip and dip them into H_2SO_4:H_2O_2:H_2O (sulphuric acid - hydrogen peroxide - water) to make their surfaces hydrophilic (Figure 6.11). After bringing them together, they are put into an oven for 3 hours at 250°. The silicon chip was not structured and the defining of the III-V laser and the silicon waveguide was done in one step and the laser is pumped from the top with a 980 nm laser. This is different from the approach we envision in Gent where bonding of III-V is done on a structured silicon chip and the pumping happens in-plane and not from the top.

The initial plan was quite simple. I would design both the passive structures in silicon fabricated with ePIXfab and the length and structure of the III-V regions as in Figure 6.10. They would take the fabrication of the III-V chips upon them and the bonding to the silicon and I would measure the chips, compare it with the designs and simulations and the cycle would be repeated. Shortly before my arrival, the group had acquired a new bonding tool that allowed for Plasma-Assisted Surface-Activated Direct Bonding, which means that the surfaces are cleaned and activated by exposure to a plasma. The surfaces are contacted in high vacuum and room temperature and to achieve a firm bonding a high pressure is applied on the chips together with an increase in temperature [38, 39]. The temperatures needed for a strong bonding should be smaller than with hydrophilic bonding and therefore reduce thermal stress in the chips.

That year they did mostly a development of this new bonding technique and after 10 months of small incremental progress but also repeated failure to achieve a useful bonding (detachment of the III-V die, cracks, damage of the active layers), the old process of their original paper was employed again [30] with the result of Figure 6.12 when I was back in Ghent. The bonding was mostly successful, but one region detached. At the time, the communication between both groups was limited as the group from TIT focused on their main goal of bonding a III-V laser with lateral current injection on a silicon chip [37]. One problem was that our approach of Figure 6.10 requires very thin membranes (order of 100 nm thickness) which were damaged due to the high pressures during bonding with their plasma-assisted bonding tool. Adhesive bonding does not need these high pressures and therefore the focus was shifted away from molecular bonding to adhesive bonding when I was back in Gent and this is next section's topic.

6.2.4 Fabrication in Gent

Although the very thin bonding thickness required for the BCB — below 100 nm to couple the light between the silicon and III-V waveguides with high efficiency — poses a challenge, the bonding process itself can be done in Gent with success as shown in Figure 6.13(a). One problem during operation is that BCB is a

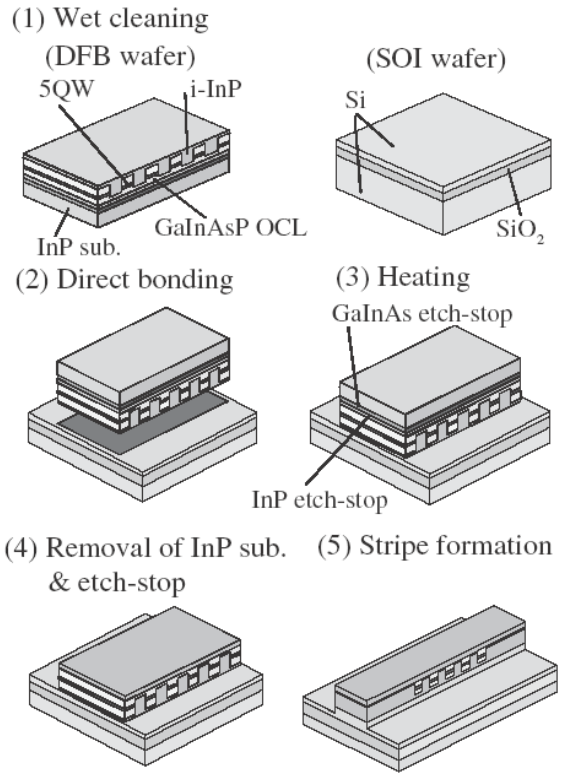


Figure 6.11: The fabrication process for the optically pumped laser bonded with molecular bonding from [30]. In the first step a InGaAsP/InP is grown with Organic Metal Vapor Phase Epitaxy (OMVPE), with 5 active QW layers. The group is famous for making wire-like DFB lasers, which means that they etch a periodic structure in the active region itself acting as a mirror, instead of making the periodic structure close to the active region. The following steps include cleaning both unstructured silicon and III-V chips and bonding them. Finally the InP substrate is removed and the waveguide structure is formed with lithography and etching.

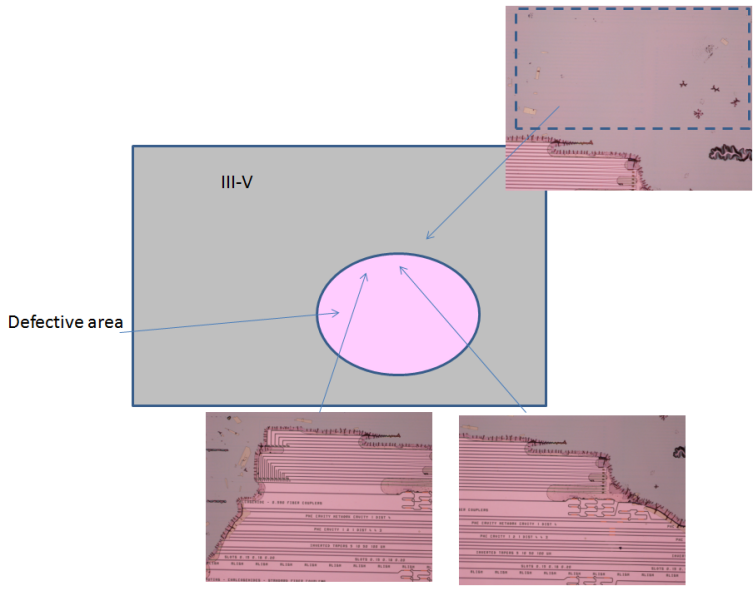


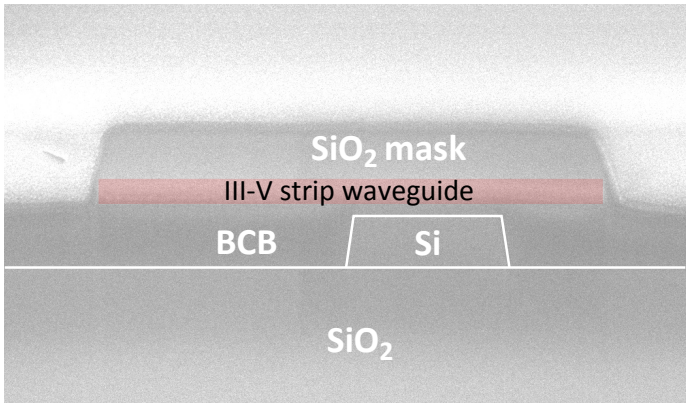
Figure 6.12: Bonding of a III-V chip from TIT with a silicon chip from ePIXfab. Part of the bonded region detached during bonding.

bad conductor for heat, so the III-V is thermally isolated and heats up which degrades the performance of the amplifier or even leads to material breakdown (Figure 6.13(b)). Research is ongoing to alleviate those problems, but in pulsed regime a gain of over 8 dB has been achieved [1]. For now the research is still focused on fabricating one SOA working in continuous wave regime before fabrication and experiments can start on a network of coupled SOAs on this platform.

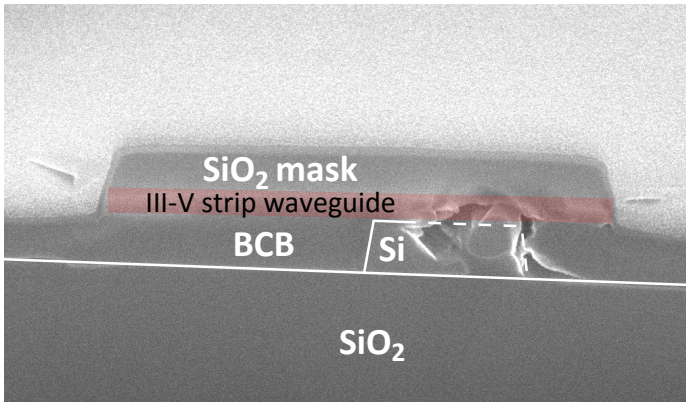
6.3 Conclusions

This chapter described the progress made to fabricate and measure an integrated chip of coupled SOAs. The route with the pure III-V platform of JePPIX looks most promising in the short term as the fabrication has been finished and the work to be done involves building a measurement setup. The silicon route with bonded III-V is still in an experimental phase as the SOAs so far only work in pulsed regime.

Recently integrated optical amplification was shown in silicon waveguides with a strong optical pump and the nonlinear process of four-wave mixing [41]. This approach requires a very high pump power, but otherwise everything can be done in silicon and heterogeneous integration is not necessary, which makes



(a) BCB bonded III-V on Si



(b) Thermal effects

Figure 6.13: (a) shows a Scanning Electron Microscope (SEM) picture of a III-V chip bonded with BCB onto a silicon waveguide [40]. On top of the III-V strip waveguide there is still a hard mask used for defining the strip (or deep etched) waveguide. (b) shows the problem of thermal effects as a part of the waveguide and the III-V has burst into pieces. (picture courtesy of Martijn Tassaert).

it very interesting for future research.

References

[1] M. Tassaert, S. Keyvaninia, D. Van Thourhout, W. M. J. Green, Y. Vlasov, and G. Roelkens. *A nanophotonic InP/InGaAlAs optical amplifier integrated on a Silicon-On-Insulator waveguide circuit.* In Proceedings of Information Photonics, Ottawa, Canada, May 2011.

[2] B. E. A. Saleh and M. V. Teich. *Fundamentals of Photonics.* John Wiley & Sons, Inc., New York, NY, USA, 1991.

[3] X. Leijtens. *JePPIX: the platform for InP-based photonics.* In Proceedings of the 15th European Conference on Integrated Optics (ECIO), page ThG3, Cambridge, UK, April 2010.

[4] M. J. R. Heck, A. La Porta, X. J. M. Leijtens, L. M. Augustin, T. de Vries, B. Smalbrugge, Y. S. Oei, R. Nötzel, R. Gaudino, D. J. Robbins, and M. K. Smit. *Monolithic AWG-based Discretely Tunable Laser Diode With Nanosecond Switching Speed.* IEEE Photonics Technology Letters, 21(13):905–907, July 2009.

[5] E. Kapon. *Semiconductor lasers, Part I and II.* Academic Press, San Diego, CA, USA, 1999.

[6] Y. Barbarin, E. A. J. M. Bente, M. J. R. Heck, J. H. den Besten, G. Guidi, Y. S. Oei, J. J. M. Binsma, and M. K. Smit. *Realization and modeling of a 27-GHz integrated passively mode-locked ring laser.* IEEE Photonics Technology Letters, 17(11):2277–2279, November 2005.

[7] M. J. R. Heck, P. Muñoz, B. W. Tilma, E. A. J. M. Bente, Y. Barbarin, Y. S. Oei, R. Nötzel, and M. K. Smit. *Design, Fabrication and Characterization of an InP-Based Tunable Integrated Optical Pulse Shaper.* IEEE Journal of Quantum Electronics, 44(4):370–377, April 2008.

[8] A. Rohit, K. A. Williams, X. J. M. Leijtens, T. de Vries, Y. S. Oei, M. J. R. Heck, L. M. Augustin, R. Nötzel, D. J. Robbins, and M. K. Smit. *Monolithic Multi-band Nanosecond Programmable Wavelength Router.* IEEE Photonics Journal, 2(1):29–35, February 2010.

[9] K. Caluwaerts. *Semiconductor Optical Amplifiers as a basis for Photonic Reservoir Computing.* Technical report, PARIS, ELIS, Ghent University, 2011.

[10] *What is the difference between an equivalent time sampling oscilloscope and a real-time oscilloscope?* Technical report, Agilent Technologies, Inc., 2008. Application Note 1608.

[11] S. K. Selvaraja. *Wafer-Scale Fabrication Technology for Silicon Photonic Integrated Circuits*. PhD thesis, Ghent University, 2011.

[12] W. Bogaerts, S. K. Selvaraja, P. Dumon, J. Brouckaert, K. De Vos, D. Van Thourhout, and R. Baets. *Silicon-on-Insulator Spectral Filters Fabricated With CMOS Technology*. IEEE Journal of Selected Topics in Quantum Electronics, 16(1):33–44, 2010.

[13] W. Bogaerts and S. K. Selvaraja. *Compact Single-Mode Silicon Hybrid Rib/Strip Waveguide With Adiabatic Bends*. IEEE Photonics Journal, 3(3):422–432, June 2011.

[14] W. Bogaerts, P. Dumon, D. Van Thourhout, and R. Baets. *Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides*. Optics Letters, 32(19):2801–2803, 2007.

[15] T. Wang, H. Liu, A. Lee, F. Pozzi, and A. Seeds. *1.3-μm InAs/GaAs quantum-dot lasers monolithically grown on Si substrates*. Optics Express, 19(12):11381–11386, May 2011.

[16] A. E. J. Lim, T. Y. Liow, F. Qing, N. Duan, L. Ding, M. Yu, G. Q. Lo, and D. L. Kwong. *Novel evanescent-coupled germanium electro-absorption modulator featuring monolithic integration with germanium p-i-n photodetector*. Optics Express, 19(6):5040–5046, March 2011.

[17] J. Liu, J. Michel, W. Giziewicz, D. Pan, K. Wada, D. D. Cannon, S. Jongthammanurak, D. T. Danielson, L. C. Kimerling, J. Chen, F. O. Ilday, F. X. Kartner, and J. Yasaitis. *High-performance, tensile-strained Ge p-i-n photodetectors on a Si platform*. Applied Physics Letters, 87(10):103501, 2005.

[18] M. Rouvière, L. Vivien, X. Le Roux, J. Mangeney, P. Crozat, C. Hoarau, E. Cassan, D. Pascal, S. Laval, J. M. Fédéli, J. F. Damlencourt, J. M. Hartmann, and S. Kolev. *Ultrahigh speed germanium-on-silicon-on-insulator photodetectors for 1.31 and 1.55 μm operation*. Applied Physics Letters, 87(23):231109, 2005.

[19] J. Michel, J. Liu, and L. C. Kimerling. *High-performance Ge-on-Si photodetectors*. Nature Photonics, 4(8):527–534, August 2010.

[20] T. Y. Liow, K. W. Ang, Q. Fang, J. F. Song, Y. Z. Xiong, M. B. Yu, C. Q. Lo, and D. L. Kwong. *Silicon Modulators and Germanium Photodetectors on SOI: Monolithic Integration, Compatibility, and Performance Optimization*. IEEE Journal of Selected Topics in Quantum Electronics, 16(1):307–315, 2010.

[21] J. Liu, X. Sun, R. Camacho-Aguilera, L. C. Kimerling, and J. Michel. *Ge-on-Si laser operating at room temperature.* Optics Letters, 35(5):679–681, March 2010.

[22] H. Park, A. W. Fang, O. Cohen, R. Jones, M. J. Paniccia, and J. E. Bowers. *A hybrid AlGaInAs-silicon evanescent amplifier.* IEEE Photonics Technology Letters, 19(2-4):230–232, 2007.

[23] A. W. Fang, R. Jones, H. Park, O. Cohen, O. Raday, M. J. Paniccia, and J. E. Bowers. *Integrated AlGaInAs-silicon evanescent racetrack laser and photodetector.* Optics Express, 15(5):2315–2322, 2007.

[24] J. Van Campenhout, P. Rojo-Romeo, P. Regreny, C. Seassal, D. Van Thourhout, S. Verstuyft, L. Di Cioccio, J. M. Fedeli, C. Lagahe, and R. Baets. *Electrically pumped InP-based microdisk lasers integrated with a nanophotonic silicon-on-insulator waveguide circuit.* Optics Express, 15(11):6744–6749, May 2007.

[25] G. Roelkens, L. Liu, D. Liang, R. Jones, A. Fang, B. Koch, and J. Bowers. *III-V/silicon photonics for on-chip and intra-chip optical interconnects.* Laser & Photonics Reviews, 4(6):751–779, November 2010.

[26] J. Brouckaert, G. Roelkens, D. Van Thourhout, and R. Baets. *Compact InAlAs-InGaAs Metal-Semiconductor-Metal photodetectors integrated on silicon-on-insulator waveguides.* IEEE Photonics Technology Letters, 19(17-20):1484–1486, 2007.

[27] A. W. Fang, B. R. Koch, K. G. Gan, H. Park, R. Jones, O. Cohen, M. J. Paniccia, D. J. Blumenthal, and J. E. Bowers. *A racetrack mode-locked silicon evanescent laser.* Optics Express, 16(2):1393–1398, 2008.

[28] A. W. Fang, E. Lively, Y. H. Kuo, D. Liang, and J. E. Bowers. *A distributed feedback silicon evanescent laser.* Optics Express, 16(7):4413–4419, March 2008.

[29] T. Maruyama, T. Okumura, S. Sakamoto, K. Miura, Y. Nishimoto, and S. Arai. *GaInAsP/InP membrane BH-DFB lasers directly bonded on SOI substrate.* Optics Express, 14(18):8184–8188, 2006.

[30] T. Okumura, T. Maruyama, M. Kanemaru, S. Sakamoto, and S. Arai. *Single-mode operation of GaInAsP/InP-membrane distributed feedback lasers bonded on silicon-on-insulator substrate with rib-waveguide structure.* Japanese Journal of Applied Physics Part 2-Letters & Express Letters, 46(45-49):L1206–L1208, 2007.

[31] H. Park, A. W. Fang, R. Jones, O. Cohen, O. Raday, M. N. Sysak, M. J. Paniccia, and J. E. Bowers. *A hybrid AlGaInAs-silicon evanescent waveguide photodetector.* Optics Express, 15(10):6044–52, May 2007.

[32] H. W. Chen, Y. H. Kuo, and J. E. Bowers. *High speed hybrid silicon evanescent Mach-Zehnder modulator and switch.* Optics Express, 16(25):20571–20576, December 2008.

[33] G. Roelkens, J. Brouckaert, D. Van Thourhout, R. Baets, R. Nötzel, and M. Smit. *Adhesive Bonding of InP/InGaAsP Dies to Processed Silicon-On-Insulator Wafers using DVS-bis-Benzocyclobutene.* Journal of The Electrochemical Society, 153(12):G1015, 2006.

[34] G. Roelkens. *Heterogeneous III-V/Silicon Photonics: Bonding Technology and Integrated Devices.* PhD thesis, Ghent University, 2007.

[35] Y. Halioua, A. Bazin, P. Monnier, T. J. Karle, G. Roelkens, I. Sagnes, R. Raj, and F. Raineri. *Hybrid III-V semiconductor / silicon nanolaser.* Optics Express, 19(10):9221–9231, 2011.

[36] G Roelkens, D Van Thourhout, R Baets, R Nötzel, and M Smit. *Laser emission and photodetection in an InP/InGaAsP layer integrated on and coupled to a Silicon-on-Insulator waveguide circuit.* Optics express, 14(18):8154–8159, October 2006.

[37] T. Okumura, T. Koguchi, H. Ito, N. Nishiyama, and S. Arai. *Injection-Type GaInAsP/InP Membrane Buried Heterostructure Distributed Feedback Laser with Wirelike Active Regions.* Applied Physics Express, 4(4):042101, March 2011.

[38] R. Takei, K. Yoshida, and T. Mizumoto. *Effects of Wafer Precleaning and Plasma Irradiation to Wafer Surfaces on Plasma-Assisted Surface-Activated Direct Bonding.* Japanese Journal of Applied Physics, 49(8):086204, August 2010.

[39] D. Pasquariello and K. Hjort. *Plasma-assisted InP-to-Si low temperature wafer bonding.* IEEE Journal of Selected Topics in Quantum Electronics, 8(1):118–131, 2002.

[40] G. Roelkens, O. Raz, W. M. J. Green, S. Assefa, M. Tassaert, S. Keyvaninia, K. Vandoorne, D. Van Thourhout, R. Baets, and Y. Vlasov. *Towards a low-power nanophotonic semiconductor amplifier heterogeneously integrated with SOI waveguides.* In Proceedings of the 7th IEEE International Conference on Group IV Photonics (GFP), pages 16–18, September 2010.

[41] B. Kuyken, S. Clemmen, S. K. Selvaraja, W. Bogaerts, D. Van Thourhout, P. Emplit, S. Massar, G. Roelkens, and R. Baets. *On-chip parametric amplification with 26.5 dB gain at telecommunication wavelengths using CMOS-compatible hydrogenated amorphous silicon waveguides.* Optics Letters, 36(4):552–554, February 2011.

7 Conclusions and perspectives

In this chapter we first discuss the conclusions from this work, and finish with an outlook on future research possibilities.

7.1 Conclusions

In this thesis we explored the usefulness of integrated photonics for a hardware implementation of reservoir computing. Since this research is at the intersection of two research fields, photonics and machine learning, the target reader group consists of engineers from both fields. Chapter 2 gave an introduction to reservoir computing with some historical background to give a broader perspective regarding the problems RC offers a solution to. Machine learning involves systems that are trained to achieve a certain functionality, instead of using explicitly programmed solutions. Training dynamical recurrent systems proved to be very hard, but reservoir computing circumvents most problems by bringing on the one hand the dynamical recurrent reservoir in the right regime without training it, while on the other hand the training of the (often linear) readout is done with simple methods. The chapter also described a typical reservoir implementation with a recurrent network of simple tanh nodes with leak rate, which adds internal memory to each node by feeding it with its own past state as input.

Since the requirements the dynamical system has to fulfill are quite general, integrated photonics was investigated as a hardware implementation of reser-

voir computing. Photonics is a broad field in itself, and in Chapter 3 we mainly focused on semiconductor optical amplifiers as our choice for the nodes in the recurrent network. We described the SOA model we used and the properties where it differs from classical reservoir implementations with tanh nodes. Important differences include the internal dynamics due to the carrier lifetime (the carriers provide amplification), the coherence of the light (which can be represented by complex-valued amplitudes instead of real-valued with classical reservoirs) and the limitations on the topology since we want to avoid crossing waveguides on planar chips.

Chapter 4 described the results of photonic SOA reservoirs on a simple signal classification task. It showed that despite the differences between classical and SOA reservoirs, the latter can obtain good results on such a simple task. Furthermore it revealed that working with complex amplitudes is important as the results can be very sensitive to the phase of the light and in the connections. Together with coherence, it turned out that the interconnection delay itself, a parameter usually ignored for classical reservoirs, has a very strong influence on the performance.

With these insights we turned to a much more difficult task (recognizing isolated spoken digits with babble noise) in Chapter 5 to investigate in detail the difference between classical and SOA reservoirs. We have systematically linked design parameters and behavioral properties of our implementation to the observed performance trends. As in Chapter 4, the delay of the inter-node connections turned out to be very important and we have shown that its impact is also found in traditional reservoirs without leak rate. A second important parameter is the phase shift in the interconnections. If this can be controlled, a significant performance improvement is achieved. When both interconnection delay and phase shift can be optimized, our SOA implementation outperforms optimized traditional tanh reservoirs with leak rate. If the phase shift can not be tuned, we still find an optimal performance similar to that of traditional simulated reservoirs. Other design parameters were found to have less impact on performance, although for larger reservoirs using a swirl topology has an advantage over random topologies as the performance for random topologies levels off.

The insights from this chapter can be used to formulate a rule of thumb of how to obtain good performance with SOA reservoirs when dealing with new tasks. First, choose a swirl topology (as the swirl topology did it usually as good or better than random topologies) with an appropriate size (at least more than the number of input channels). Second, make sure that the relevant time scales of the input signals are in the same order of the time scales in the photonic reservoir by resampling. Third, take a value for the input current of the SOAs so the losses in the connections are compensated and a value for the input scaling to stay in

the linear regime of the SOAs. Fourth, search for an optimal value for the delay through $(\rho, \Delta\Phi)$ sweeps. If such an optimal delay is found, an optimal value for the input scaling can also be obtained through $(\rho, \Delta\Phi)$ sweeps. Finally, larger reservoirs can further improve the performance.

The influence of noise in the amplifiers and fabrication variations were also investigated. While process variations do not turn out to be very important, careful considerations to limit ASE will be necessary. Although these experiments offered an in-depth analysis for one specific task, confirming the insights of the previous chapter, it indicates that similar conclusions would follow for other tasks. The importance of our work is not only in evaluating this specific SOA architecture. Indeed, our conclusions can be generalized to other coherent photonic implementations with similar node properties.

Finally, Chapter 6 discussed our progress in building a photonic hardware implementation and validating our simulation results. Although we never considered integrating several SOAs on a single chip to be easy, it has proved to be a lot harder than expected. E.g., an amplifier working in continuous wave regime still has to be demonstrated with our in-house silicon-on-insulator technology. Another photonic hardware platform based on III-V materials has so far resulted in a successful fabrication of a design with 12 SOAs at the time of writing.

7.2 Perspectives

This work about photonic reservoir computing was the first in (hopefully) a long train of successive research projects. This is already ongoing as two people in our group have started working on photonic reservoir computing: Martin Fiers and Thomas Van Vaerenbergh. Their work is more focused on integrated nanoscale implementations that are optimized for speed and/or power consumption, like rings and photonic crystal cavities. These passive structures have a very narrow band response and interesting nonlinear dynamics like bistability, self-pulsing and chaos [1]. They stray farther away from the path of classical reservoir computing which makes it simultaneously more complicated and promising. Although we have made the claim that photonics offers speed and power consumption advantages, we do not envision this for SOA reservoirs, which were mostly chosen because of their close resemblance to classical reservoir nodes (but also because they provide gain and are broadband).

As our work was only the first step, calling the scope for further research vast would be an understatement. The fabricated JePPIX chip with 12 SOAs is a perfect small-scale implementation to validate all the simulation results from this PhD thesis. But 12 nodes are still quite limited as all the results from Chapter 5 were with networks of 81 nodes and results improve further for larger networks. An SOA on the silicon platform offers the potential to not only integrate more

SOAs on a single chip, but also longer delays because spiraling delay lines with a small footprint can be made easier in silicon. Filters to remove the influence of ASE have already been intensively investigated [2], although some tuning and trimming of the rings is probably required to let their resonances overlap [3–5], despite a demonstrated linewidth control of 0.6 nm [6].

The amplification in this work has been done with semiconductor optical amplifiers. However, that is not the only possibility to provide amplification as nonlinear processes can be used as well. Very recently an amplification of 26.5 dB in SOI based on four-wave mixing has been demonstrated [7]. The advantage is that no heterogeneous integration is required as everything happens in the silicon waveguides under influence of a strong pump.

The tasks used in this work, signal classification and speech recognition, are benchmark tasks from the world of reservoir computing. However, it would be interesting to have an application specifically geared toward a photonic application. One example could be the monitoring of nonlinear signal degradation [8]. Digital tasks such as signal regeneration and header recognition and processing are also attractive as optical digital solutions are still in a research phase. However, it remains to be seen if reservoir computing is really the most suitable approach to digital problems. This might seem as if photonic reservoir computing is a solution looking for a problem, but when it comes to analyzing huge amounts of very fast time series of optical data, it definitely has an edge as the success of machine learning techniques have shown for all kinds of classification and recognition problems.

Reservoir computing uses a reservoir and a linear readout. The focus in this work has been on making a photonic reservoir, with the assumption that the readout and training would be done electronically. The readout itself, where different output streams are multiplied by a weight and summed together, could be translated to photonics, especially when the training would happen with complex weights, as opposed to real weights used in this dissertation. However, the training which involves matrix inversion, is for the moment not straightforward optically. A first approach is using a different kind of training with easier manipulations than matrix inversion, a second is tightly bound with a breakthrough in digital photonics which would fulfill the same functionality as digital electronics does now.

Finally, the field of reservoir computing itself is evolving as well, and will inspire new ideas and approaches. So, possibilities abound for this new and exciting research field.

References

[1] B. Maes, M. Fiers, and P. Bienstman. *Self-pulsing and chaos in short chains of coupled nonlinear microcavities.* Physical Review A, 80(3):033805, September 2009.

[2] W. Bogaerts, S. K. Selvaraja, P. Dumon, J. Brouckaert, K. De Vos, D. Van Thourhout, and R. Baets. *Silicon-on-Insulator Spectral Filters Fabricated With CMOS Technology.* IEEE Journal of Selected Topics in Quantum Electronics, 16(1):33–44, 2010.

[3] P. Dong, N. N. Feng, D. Z. Feng, W. Qian, H. Liang, D. C. Lee, B. J. Luff, T. Banwell, A. Agarwal, P. Toliver, R. Menendez, T. K. Woodward, and M. Asghari. *GHz-bandwidth optical filters based on high-order silicon ring resonators.* Optics Express, 18(23):23784–23789, 2010.

[4] W. De Cort, J. Beeckman, R. James, F. A. Fernández, R. Baets, and K. Neyts. *Tuning silicon-on-insulator ring resonators with in-plane switching liquid crystals.* Journal of the Optical Society of America B, 28(1):79–85, January 2011.

[5] J. Schrauwen, D. Van Thourhout, and R. Baets. *Trimming of silicon ring resonator by electron beam induced compaction and strain.* Optics Express, 16(6):3738–3743, March 2008.

[6] S. K. Selvaraja, W. Bogaerts, P. Dumon, D. Van Thourhout, and R. Baets. *Subnanometer Linewidth Uniformity in Silicon Nanophotonic Waveguide Devices Using CMOS Fabrication Technology.* IEEE Journal of Selected Topics in Quantum Electronics, 16(1):316–324, 2010.

[7] B. Kuyken, S. Clemmen, S. K. Selvaraja, W. Bogaerts, D. Van Thourhout, P. Emplit, S. Massar, G. Roelkens, and R. Baets. *On-chip parametric amplification with 26.5 dB gain at telecommunication wavelengths using CMOS-compatible hydrogenated amorphous silicon waveguides.* Optics Letters, 36(4):552–554, February 2011.

[8] X. Wu, J. A. Jargon, R. A. Skoog, L. Paraschis, and A. E. Willner. *Applications of Artificial Neural Networks in Optical Performance Monitoring.* Journal of Lightwave Technology, 27(16):3580–3589, August 2009.