Neuromorphic computing using nonlinear ring resonators on a Silicon photonic chip

Florian Denis-Le Coarer

N° d'ordre : 2020-CSUP-0001

CentraleSupélec

Ecole Doctorale C2MP

« Chimie Mécanique. Matériaux Physique »

Laboratoire Matériaux Optiques, Photonique et Systèmes LMOPS EA-4423

THÈSE DE DOCTORAT

Spécialité de doctorat : PHYSIQUE

Soutenue le 22 Janvier 2020

par :

Florian DENIS-LE COARER

Neuromorphic computing using nonlinear ring resonators on a Silicon photonic chip

Composition du jury :

Directeur de thèse :	Marc SCIAMANNA	Professeur (CentraleSupélec)
Co-directeur de thèse :	Damien RONTANI	Maître de conférences (CentraleSupélec)
Président du jury :	Sylvain GIGAN	Professeur des Universités (Université la Sorbonne)
	Serge MASSAR	Professeur (Université Libre de Bruxelles)
Examinateurs :	Peter BIENSTMAN	Professeur (Ghent University)
	Sylvie MENEZO	CEO (SCINTIL Photonics)

Remerciements

A l'issue de la rédaction de ce manuscrit de thèse, je suis convaincu que la thèse est loin d'être un travail solitaire. En effet, je n'aurais jamais pu réaliser ce travail doctoral sans le soutien d'un grand nombre de personnes dont la générosité, la bonne humeur et l'intérêt manifestés àl 'égard de ma recherche m'ont permis de progresser dans cette phase délicate de « l'apprenti-chercheur ».

En premier lieu, je tiens à remercier mes directeurs de thèse, Marc SCIAMANNA et Damien RONTANI, pour la confiance qu'il m'ont accordé en acceptant d'encadrer ce travail doctoral, pour leurs multiples conseils et pour toutes les heures qu'il a consacrées à diriger cette recherche. Je tiens en particulier à souligner leur disponibilité, et leur capacité à la fois à me remonter le moral dans les moments moins faciles, et à me mettre la pression nécessaire à la réussite de ces recherches.

C'est avec une grande sincérité que je souhaite remercier tous les membres de mon jury, Mr Sylvain Gigan son président, Mr Serge Massar pour sa relecture attentive en tant que rapporteur, Mr Peter Bienstman et Mme Sylvie Menezo en tant qu'examinateurs, pour leurs retours positifs et leurs questions pertinentes lors de la soutenance.

Cette thèse s'est inscrite dans le contexte absolument passionnant d'un projet Européen H2020. Je souhaite remercier les membres du consortium, et en particulier nos proches collaborateurs de l'Université de Gand, en Belgique. Il me faut remercier une nouvelle fois Mr Peter Bienstman ainsi que Mme Joni Dambre, qui m'ont accueilli à de nombreuses reprises dans leur équipe pour des formations, ainsi que des campagnes de mesures expérimentales, et sans qui ce travail de thèse n'aurait pas été possible. Mais aussi d'autres membres de leur équipe, en particulier Andrew Katumba, Matthias Freiberger en particulier pour leurs remarques pertinentes et leur disponibilité lors de mes visites.

Qu'aurait été cette thèse sans mes collègues de l'équipe Photonique du LMOPS ? Je n'ose pas l'imaginer. Pour ces moments de complicité, de craquages, mais aussi de nombreuses pauses café à débattre de tout, de rien, mais surtout

de science, je tiens à remercier Thomas Bouchet, Chi-Hak Uy, Lonel Weicker, Alban Maertens, Lamyae Drouzi, Guillaume Bouchez, Jérémy Vatin, Nacera Bouldja, Yaya Doumbia, Stefan Bittner, Piotr Antonik, Jean-Louis Gutzwiller, Nicolas Marsal et Delphine Wolfersberger, ainsi que mes deux directeurs de thèse Marc et Damien. Bien évidemment, je ne peux oublier de mentionner d'autres collègues de CentraleSupélec qui ont permis à ce travail d'être un succès, par leur aide technique, ou leur bonne humeur. Je cite ici Mme Fabienne Munier, Mr Mario Fernandez, Maryvonne Fall, Gilles Berlin.

Je tiens aussi à remercier les générations d'étudiants ingénieurs que j'ai pu côtoyer pendant ces années. Si je devais remercier chacun d'entre eux, il me faudrait plusieurs dizaines de pages. Mais ils m'ont beaucoup apporté, soutenus moralement. Je ne peux m'empêcher de remercier les membres de certaines associations qui m'ont donné la chance de pouvoir continuer à m'investir de manière plus sporadique. En particulier la Sono Supélec Metz et la Coopé Technopôle Metz.

Je peux aussi remercier les agents de la SNCF, qui m'ont donné l'occasion d'avoir quelques semaines de plus pour peaufiner ma soutenance, qui a été reportée la veille de celle-ci à cause d'une grève, et ma maman qui a préparé le buffet de thèse deux fois.

En enfin. Que dire, si ce n'est MERCI. Merci à vous mes très chers parents, et famille. D'avoir cru en moi. De m'avoir soutenu, de m'avoir tant donné. Toujours là lorsque j'en avais besoin, mais surtout là tout le reste du temps.

RÉSUMÉ EN FRANÇAIS

Avec la quantité exponentielle de données générées chaque jour, un besoin de pouvoir traiter les données en temps réel et de manière économe en énergie est né. Ces défis ont motivé la recherche de moyens de traitements non conventionnels de l'information. Parmi les techniques existantes, l'apprentissage machine est un paradigme très efficace d'informatique non conventionnelle. Il fournit, à travers de nombreuses implémentations possibles, un ensemble de techniques pour apprendre à un ordinateur à effectuer des tâches complexes, telles que la classification, la reconnaissance de formes ou la génération de signaux.

Parmi les approches pour mettre en œuvre l'apprentissage machine, on trouve le réseau de neurones artificiels. Il s'inspire du mécanisme de traitement de l'information du cerveau humain, qui consiste en l'interconnexion de petites unités de calcul appelées neurones. Un sous-groupe de réseaux de neurones artificiels, appelés réseaux de neurones récurrents, comprend des cycles de récurrence dans les interconnexions, ce qui permet au réseau d'exécuter efficacement des tâches qui nécessitent de la mémoire tel que la reconnaissance vocale et la reconnaissance de formes ou la génération de signaux, mais qui implique un entraînement difficile du réseau neuronal artificiel.

Le reservoir computing a été proposé il y a une dizaine d'années [1, 2] comme une extension des echo state network [3] et des liquid-state machine [4], et a attiré beaucoup d'attention en raison de l'universalité de ses concepts [5–7]. Ce nouveau paradigme a été suggéré comme un moyen de simplifier la procédure d'entraînement du réseau de neurones. En effet, le réseau neuronal récurrent est maintenu fixe et seules les connexions entre la couche de lecture et la sortie sont entraînées au moyen d'une simple régression linéaire. L'architecture interne du réseau neuronal - c'est-à-dire un réseau fixe récurrent avec une couche de lecture - permet des implémentations sur substrat physique, et des implémentations de réservoirs ont été proposées sur diverses plates-formes matérielles, y compris la photonique [8–11], including photonics [12–18].

Les implémentations de réservoirs sur puce photoniques sont très prometteuses pour les applications de télécommunications optiques, car elles fonc-

tionnent à grande vitesse avec une faible consommation d'énergie [12, 14, 15, 19, 20]. En particulier, un projet européen H2020 nommé Phresco (**PH**otonic **RES**ervoir **CO**mputing) [21, 22] a été proposé dans le prolongement des travaux réalisés par nos proches collaborateurs de l'Université de Gand sur un reservoir computer étendu composé de 16 nœuds linéaires interconnectés. Ce projet est le fruit d'une collaboration entre la Katholieke Universiteit Leuven (KUL), Universiteit Gent (Université de Gand), IBM Research GMBH à Zurich, IHP GMBH (Innovations for High Performance Microelectronics à Leibniz) et CentraleSupélec, et vise à concevoir et à expérimenter un réservoir sur puces photonique à 64 noeuds pour les télécommunications à 32 Gb/s.

Les travaux présentés dans cette thèse sont à mettre en perspective avec ce projet européen et s'inscrivent dans le cadre d'une étude exploratoire sur les avantages et les inconvénients de l'ajout de nonlinéarités dans la structure interne du réservoir. En effet, dans le travail de nos collaborateurs [15], les neurones sont des composants photoniques linéaires (petits guides d'ondes intégrés), et la nonlinéarité du système se trouve à la photo-détection. Cependant, il est souvent affirmé que la présence de nœuds nonlinéaires dans la structure du réservoir est nécessaire à de meilleures performances. Ainsi, une partie de cette thèse consiste en l'étude de la performance des architectures de réservoirs dans lesquelles nous avons remplacé les nœuds linéaires de [15] par des éléments nonlinéaires.

Nous avons basé nos architectures sur un composant très couramment utilisé, à savoir le résonateur en anneau nonlinéaire. Cet élément intégré présente des comportements dynamiques nonlinéaires riches [23–29]. Les résonateurs en anneaux intégrés sur silicium sont principalement utilisés comme filtres optiques [30], mais peuvent également être intégrés dans des architectures plus complexes et effectuer d'autres types de traitement tout optique de l'information tels que les fonctions booléennes [31], le seuillage [32] ou la restauration des impulsions [33].

Dans cette thèse, nous avons étudié trois architectures réservoirs sur puce photonique, en utilisant le résonateur nonlinéaire en anneau comme noeud primaire.

Ordinateur à réservoir composé de 16 résonateurs en anneau nonlinéaires : Plus spécifiquement, nous avons proposé de remplacer les 16 noeuds linéaires des références [15, 20, 34] par 16 résonateurs en anneau non linéaires, également interconnectés selon la topologie SWIRL. Nous avons montré numériquement que cette structure peut être utilisée comme un réservoir, et qu'elle peut fonctionner à des niveaux de performance de pointe sur la tâche typique du XOR retardé. Pour cette tâche, nous atteignons un taux d'erreur de 2.5×10^{-4} à 20 Gb/s, et pour un large ensemble de valeurs de

paramètres.

Nous avons relié les propriétés intrinsèques de l'élément constitutif du réservoir avec (i) la conception optimale en termes de délai d'entrelacement des noeuds, et (ii) les paramètres d'injection optimaux du réservoir. En effet, pour les applications réservoir, une technique d'injection couramment utilisée consiste à faire fonctionner le système à la limite de l'instabilité [35]. Dans notre cas, nous avons fait l'hypothèse que l'ensemble du réseau reste stable lorsque chaque élément nonlinéaire est stable, et que lorsque nous injectons les données proche de l'instabilité de chaque résonateur en anneau, le réseau entier sera proche de sa région d'instabilités. Cette hypothèse reste cohérente en raison des pertes de connexion entre deux nœuds consécutifs induites par les séparateurs, les combinateurs et le long guide d'ondes d'interconnexion. Nous avons donc cartographié les zones de stabilité d'un résonateur en anneau soumis à une injection optique, dans un plan (désaccord optique, puissance d'injection), et trouvé un ensemble de valeurs de paramètres optimales pour faire fonctionner le réservoir. De plus, nous avons montré par des simulations approfondies que ce type de structure est relativement robuste par rapport aux procédés de fabrication, en particulier en ce qui concerne les fréquences de résonance des anneaux.

Cependant, cette architecture composée de 16 micro-anneaux nonlinéaires interconnectés ne surpasse pas clairement en terme de performance la structure utilisant uniquement des éléments linéaires comme nœuds. En effet, le niveau de performance et la consommation énergétique de ce réservoir ne sont que très légèrement meilleurs que ceux d'un réservoir composé uniquement d'éléments linéaires. Ceci est principalement dû aux pertes dans la structure interne du réservoir, dues aux séparateurs, aux combinateurs et aux longs guides d'ondes d'interconnexion.

Calculateur de réservoir composé de 16 noeuds non-identiques : Calculateur de réservoir composé de 16 noeuds non-identiques : Ainsi, afin d'améliorer l'architecture du réservoir constituée de 16 résonateurs en anneau nonlinéaires interconnectés, et dans le prolongement des travaux de nos collaborateurs [15, 20, 34] avec des nœuds linéaires, et des premiers travaux introduisant la topologie SWIRL pour le reservoir computing utilisant les SOA comme nœuds [19], nous avons proposé de mettre en œuvre une architecture utilisant la topologie SWIRL à nœuds non identiques dans la structure du réservoir. Les nœuds ont été choisis pour être soit des résonateurs en anneau non linéaires, soit des éléments linéaires (par exemple des petits guides d'ondes), soit des amplificateurs optiques à semi-conducteur. Nous avons suggéré cette structure pour ajouter du gain dans la structure interne du réservoir, afin de compenser les pertes induites par les séparateurs, les combinateurs et les

guides d'ondes d'interconnexion.

Nous avons démontré par des simulations numériques intensives que cette structure fonctionne avec des niveaux de performance satisfaisant sur deux tâches typiques : la tâche XOR retardée, et la tâche de reconnaissance de formes à 3 bits, à 20Gb/s et 30Gb/s. Nous avons également étudié deux stratégies d'entrée, et avons confirmé le travail présenté par Andrew Katumba et ses collaborateurs [20] sur la meilleure stratégie d'injection dans un réseau 4×4 SWIRL utilisé comme réservoir. Avec nos paramètres d'injection, nous atteignons des niveaux de performance de pointe sur la tâche de reconnaissance de formes à 3 bits lorsque nous injectons uniquement sur les quatre nœuds centraux, avec une puissance d'entrée optique moyenne de 0,6 mW et une pompe électrique de 0,0675 A par SOA.

De notre étude numérique, nous avons trouvé un phénomène intéressant. Lors de l'injection des données sur tous les nœuds, le système fonctionne bien sur la tâche XOR retardée et moins bien sur la tâche de reconnaissance de formes à 3 bits, mais lorsque nous injectons sur un plus petit nombre de nœuds (les quatre nœuds centraux) le système fonctionne bien pour la tâche de reconnaissance de formes, et ne fonctionne pas pour la tâche XOR retardée. Nous expliquons ce phénomène par l'arbitrage entre la nonlinéarité des nœuds et la capacité de mémoire linéaire d'un réservoir [3, 36]. En effet, lorsque nous injectons sur tous les nœuds, les SOAs sont excités avec plus de puissance, et donc se comportent de manière plus nonlinéaire que lorsque nous injectons uniquement sur les quatre nœuds centraux. Par conséquent, le système, lorsqu'il est injecté sur tous les nœuds, fonctionne mieux sur la tâche XOR qui est très nonlinéaire, et ne peut pas effectuer correctement la tâche de reconnaissance de formes qui nécessite une mémoire linéaire. Et au contraire, lorsque nous injectons uniquement sur les quatre nœuds centraux, les SOA reçoivent moins de puissance et se comportent donc de manière plus linéaire. Le système est donc plus performant sur la tâche qui nécessite une mémoire linéaire mais moins performant sur une tâche qui demande plus de nonlinéarité.

Cependant, les résultats numériques sur cette architecture de réservoir ne montrent pas d'amélioration nette des performances d'une telle structure par rapport (i) au travail de nos collaborateurs [20], et (ii) au travail avec seulement des résonateurs nonlinéaires en anneau. Nous pensons que dans cette structure, la nonlinéarité imposée par la photo-détection écrase la nonlinéarité de la structure interne du réservoir.

Réservoir retard sur puce photonique utilisant un anneau nonlinéaire comme noeud physique : Après avoir étudié ces deux structures étendues, nous sommes arrivés à la conclusion que la structure SWIRL étendue

peut bien s'acquitter de tâches très simples à très grande vitesse, mais qu'elle est limitée en termes d'évolutivité du nombre de nœuds, principalement en raison des pertes dans la structure. Cette limite dans le nombre de nœuds que nous pouvons utiliser dans le réservoir réduit considérablement l'étendue des possibilités en termes de complexité des tâches. C'est pourquoi nous avons proposé une implémentation sur puce photonique du très étudié réservoir à retard temporel, en utilisant un résonateur en anneau nonlinéaire comme nœud physique, et de multiples aller-retour dans une ligne à retard relativement courte pour distribuer les nœuds virtuels. Cette idée d'utiliser plusieurs allers-retours dans une ligne à retard pour distribuer les nœuds virtuels est née à la fois de la modélisation mathématique d'un tel système et des travaux de Takano [37].

Nous avons démontré numériquement qu'il est possible d'atteindre des niveaux de performance à l'état de l'art sur des tâches plus complexes comme la prédiction de séries temporelles chaotiques, en utilisant un système très simple, composé d'un anneau nonlinéaire, d'un séparateur, d'un combineur, d'un SOA et d'un guide d'ondes. Plus spécifiquement, avant de tester cette architecture de réservoir sur des tâches de benchmark, nous avons étudié la capacité mémoire du système, qui est la capacité du système à reconstruire les entrées passées. Nous avons trouvé pour une valeur inter-nœuds et des paramètres d'injection optimisés une capacité de mémoire de 15, ce qui signifie qu'à un moment donné, le système contient des informations sur les 15 bits précédemment injectés. Ensuite, afin de comparer la performance de cette architecture avec notre réservoir étendu composé de 16 résonateurs en anneau non linéaire interconnectés, nous étudions par simulations numériques la performance du réservoir sur la tâche XOR retardée, et constatons que ce type de réservoir fonctionne avec la même précision sur cette tâche booléenne. Cependant, en raison du débit d'un réservoir à retard, cette application n'est pas pertinente car ce calculateur à retard ne fonctionne qu'à 0,3 Giga symboles par seconde. Cependant, ce type d'architecture est pertinent pour les tâches exigeantes en mémoire, telles que la tâche de prédiction de séries temporelles chaotiques de Santa Fe. Nous montrons par des simulations numériques intensives que cette structure peut résoudre avec succès cette tâche, avec une erreur quadratique moyenne normalisée de 1.12×10^{-2}, à 0,3 Giga symboles par seconde. Ce niveau de performance peut être comparé de manière cohérente à des systèmes basés sur le retard utilisant des lasers à semi-conducteurs comme noeuds non linéaires, et un grand nombre de noeuds virtuels distribués dans une ligne à retard constituée d'une longue fibre optique [37–39], avec des vitesses de traitement plus faibles (de l'ordre de dizaines de méga symboles par seconde).

En plus de la simplicité de conception de ce circuit photonique intégré,

cette structure offre une évolutivité très simple du nombre de nœuds. En effet, l'augmentation du nombre de nœuds peut se faire sans modifier l'architecture du circuit mais seulement en répartissant les nœuds sur un plus grand nombre d'allers-retours dans la ligne à retard, ce qui induit l'inconvénient typique du réservoir à retard temporel, qui est une réduction de la vitesse du traitement.

Conclusion : Dans cette thèse, nous avons suggéré, présenté et étudié numériquement trois nouveaux concepts de réservoir computer sur puce optique. Ces architectures reposent sur la mise en œuvre de systèmes très compacts intégrés sur une puce photonique en silicium. Dans toutes ces architectures, l'élément constitutif du réseau de neurones artificiels est le résonateur en anneau nonlinéaire, qui est un élément bien connu et couramment utilisé. Dans le prolongement des travaux de nos proches collaborateurs de l'Université de Gand, nous avons proposé un réservoir à 16 nœuds composé de 16 anneaux interconnectés selon la topologie SWIRL. Ensuite, nous avons proposé deux architectures pour améliorer les performances de ce réservoir. La première est basée sur la même topologie SWIRL, et la mise en œuvre d'un réseau neuronal avec des nœuds non identiques. La deuxième architecture est basée sur le paradigme du réservoir à retard, en utilisant un anneau comme nœud physique. Cette architecture permet une virtualisation des nœuds sur plusieurs allers-retours dans une ligne de retard relativement petite, donc cette conception est très compacte et permet une évolutivité du nombre de nœuds.

References

[1] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[2] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[3] H. Jaeger, *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001, vol. 5.

[4] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[5] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[6] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[7] L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Preprint*, pp. 1–25, 2018.

[8] F. Schürmann, K. Meier, and J. Schemmel, "Edge of chaos computation in mixed-mode vlsi-a hard liquid," *Advances in neural information processing systems*, pp. 1201–1208, 2005.

[9] D. Nikolić, S. Haeusler, W. Singer, and W. Maass, "Temporal dynamics of information content carried by neurons in the primary visual cortex," *Advances in neural information processing systems*, pp. 1041–1048, 2007.

[10] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, pp. 1–6, 2012.

[11] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[12] 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*, vol. 16, no. 15, p. 11182, 2008.

[13] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

[14] C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," *JOSA B*, vol. 30, no. 11, pp. 3048–3055, 2013.

[15] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[16] R. Modeste Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van Der Sande, "Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

[17] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[18] X. Bao, Q. Zhao, H. Yin, and J. Qin, "Recognition of the optical packet header for two channels utilizing the parallel reservoir computing

based on a semiconductor ring laser," *Modern Physics Letters B*, vol. 32, no. 14, p. 1850150, 2018.

[19] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[20] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[21] www.phresco.eu, accessed: 2019.

[22] https://cordis.europa.eu/project/rcn/198823/factsheet/en, accessed: 2019.

[23] T. Van Vaerenbergh, "All-optical spiking neurons integrated on a photonic chip," *PhD Thesis*, 2014.

[24] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[25] L. Zhang, Y. Fei, Y. Cao, X. Lei, and S. Chen, "Experimental observations of thermo-optical bistability and self-pulsation in silicon microring resonators," *JOSA B*, vol. 31, no. 2, pp. 201–206, 2014.

[26] W. H. Pernice, M. Li, and H. X. Tang, "Time-domain measurement of optical transport in silicon micro-ring resonators," *Optics express*, vol. 18, no. 17, pp. 18 438–18 452, 2010.

[27] L. Zhang, Y. Fei, T. Cao, Y. Cao, Q. Xu, and S. Chen, "Multibistability and self-pulsation in nonlinear high- q silicon microring resonators considering thermo-optical effect," *Physical Review A*, vol. 87, no. 5, p. 053805, 2013.

[28] S. Chen, L. Zhang, Y. Fei, and T. Cao, "Bistability and self-pulsation phenomena in silicon microring resonators based on nonlinear optical effects," *Optics Express*, vol. 20, no. 7, pp. 7454–7468, 2012.

[29] I. S. Amiri, R. Ahsan, A. Shahidinejad, J. Ali, and P. P. Yupapin, "Characterisation of bifurcation and chaos in silicon microring resonator," *IET Communications*, vol. 6, no. 16, pp. 2671–2675, 2012.

[30] J. Hryniewicz, P. Absil, B. Little, R. Wilson, and P.-T. Ho, "Higher order filter response in coupled microring resonators," *IEEE Photonics Technology Letters*, vol. 12, no. 3, pp. 320–322, 2000.

[31] H. J. Caulfield, R. A. Soref, and C. S. Vikram, "Universal reconfigurable optical logic with silicon-on-insulator resonant structures," *Photonics and Nanostructures-Fundamentals and Applications*, vol. 5, no. 1, pp. 14–20, 2007.

[32] H. Kishikawa, T. Kondo, N. Goto, and S. Talabattula, "Optical thresholder consisting of two cascaded mach–zehnder interferometers with nonlinear microring resonators," *Optical Engineering*, vol. 56, no. 8, p. 086101, 2017.

[33] Y. Dumeige, L. Ghisa, and P. Féron, "Integrated all-optical pulse restoration with coupled nonlinear microring resonators," *Optics letters*, vol. 31, no. 14, pp. 2187–2189, 2006.

[34] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[35] T. Yamane, S. Takeda, D. Nakano, G. Tanaka, R. Nakane, S. Nakagawa, and A. Hirose, "Dynamics of reservoir computing at the edge of stability," *International Conference on Neural Information Processing*, pp. 205–212, 2016.

[36] M. Inubushi and K. Yoshimura, "Reservoir Computing beyond Memory-Nonlinearity Trade-off," *Scientific Reports*, vol. 7, no. 1, pp. 1–10, 2017.

[37] K. Takano, C. Sugano, M. Inubushi, K. Yoshimura, S. Sunada, K. Kanno, and A. Uchida, "Compact reservoir computing with a photonic integrated circuit," *Optics express*, vol. 26, no. 22, pp. 29 424–29 439, 2018.

[38] L. Appeltant *et al.*, "Reservoir computing based on delay-dynamical systems," *These de Doctorat, Vrije Universiteit Brussel / Universitat de les Illes Balears*, 2012.

[39] K. Hicke, M. A. Escalona-Morán, D. Brunner, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Information processing using transient dynamics of semiconductor lasers subject to delayed feedback," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 4, pp. 1 501 610–1 501 610, 2013.

ENGLISH SUMMARY

With the exponential amount of data generated everyday has emerged a need for real-time, energy-efficient data processing. These challenges have motivated research in non-conventional information processing. Among the existing techniques, machine learning is a very efficient paradigm for non-conventional computing. It provides, through many possible implementations, a set of techniques to teach a computer to perform complex tasks, such as classification, pattern recognition, or signal generation.

Among the approaches to implement machine learning is the so-called artificial neural network. It is inspired by the human brain information processing mechanism, which consists in the interconnection of small computational units called neurons. A subgroup of artificial neural networks, called recurrent neural networks, includes recurrence cycles in the interconnections, thus allowing the network to perform efficiently on tasks that require memory like speech and pattern recognition or signal generation, but resulting in a difficult training of the artificial neural network.

Reservoir computing was proposed about a decade ago [1, 2] as an extension of echo state networks [3] and liquid state machines [4], and has attracted a lot of attention due to the universality of its concepts [5–7]. This new paradigm was suggested as a way to simplify the training procedure of the neural network. Indeed the recurrent neural network is kept fixed and only the connections between the readout layer and the output are trained by a simple linear regression. The inner architecture of the neural network - that is a fixed recurrent network with a readout layer - allows implementations at the physical layer, and implementations of reservoir computing have been suggested on various hardware platforms [8–11], including photonics [12–18].

On-chip implementations of photonic reservoir computing are very promising candidates for optical telecom applications as they operate at high-speed with low power consumption [12, 14, 15, 19, 20]. In particular, a European H2020 Project named Phresco (**PH**otonic **RES**ervoir **CO**mputing) [21, 22] was proposed as an extension of the work done by our close collaborators at the Ghent University on extended reservoir computer made of 16 inter-connected

linear nodes [15]. This project is in collaboration between the Katholieke Universiteit Leuven (KUL), the Universiteit Gent (Ghent University), IBM Research GMBH in Zurich, IHP GMBH (Innovations for High Performance Microelectronics in Leibniz), and CentraleSupélec, and aims at the design and the experimental demonstration of an on-chip 64-nodes reservoir computer for telecommunication applications at 32 Gb/s.

The work presented in this dissertation should be put in perspective with this European project, and is part of an exploratory study on the advantages and drawbacks of adding non-linearity in the inner structure of the reservoir. Indeed, in the work of [15], the neurons are linear photonics components (small waveguides), and the non-linearity of the system is left to the photo-detection. However, it is often stated that having nonlinear nodes in the structure of the reservoir is necessary for better performance. Hence, part of this dissertation consists in the investigation on the performance of reservoir computer architectures in which we have replaced the linear nodes of [15] by nonlinear elements.

We have based our architectures on a very commonly used component, namely the nonlinear ring resonator. This integrated element exhibits rich nonlinear dynamical behaviors [23–29]. Silicon-on-insulator microrings resonators are mostly used as optical filters [30], but can also be integrated in more complex architectures and perform other types of all-optical information processing such as Boolean functions [31], thresholding [32], pulse restoration [33], or ASK-to-PSK conversion [34].

In this thesis, we have investigated three architectures for on-chip reservoir computing, using the nonlinear ring resonator as primary node.

Reservoir computer made of 16 nonlinear ring resonators : More specifically, we have suggested to replace the 16 linear nodes of [15, 20, 35] by 16 nonlinear ring resonators, also interconnected according to the *SWIRL* topology. We have shown numerically that this structure can be used as a reservoir computer, and can perform at state-of-the-art levels of performance on the typical delayed-XOR task. For this task, we attain a Bit Error Rate of 2.5×10^{-4} at 20 Gb/s, and for a large set of parameters values.

We have also connected the intrinsic properties of the building block of the reservoir with (i) the optimum design in terms of node-interdelay, and (ii) the optimum injection parameters for reservoir computing. Indeed for reservoir computing applications, a commonly used injection technique is to operate the system at the edge of instability [36]. In our case, we have made the assumption that the whole network remains stable when each nonlinear element is stable, and that when we inject the data close to the instability of each nonlinear

ring resonator, the whole network will be close to its instable region. This assumption remains consistent due to the losses in the connection between two consecutive nodes induced by the splitters, the combiners, and the long interconnection waveguide. Hence, we have mapped the stability regions of a nonlinear ring resonator submitted to optical injection, in a (optical detuning, injection power)-plane, and found a set of optimum parameter values to operate the reservoir. Moreover, we have shown through extensive simulations that this kind of structure is relatively robust with regards to the fabrication process, in particular for the ring resonance frequencies.

However, this architecture made of 16 interconnected nonlinear ring resonators does not clearly outperform the structure using only linear elements as nodes. Indeed, the level of performance and the power consumption of this reservoir are only better by a very small factor in comparison with the reservoir made only of linear elements. This is mostly due to the losses in the inner structure of the reservoir, due to the splitters, the combiners, and the long interconnection waveguides.

Reservoir computer made of 16 non-identical nodes : Thus, as a way to improve the reservoir architecture made of 16 interconnected nonlinear ring resonators, and as an extension of the work of [15, 20, 35] with linear nodes, and the first work introducing the *SWIRL* topology for reservoir computing using SOAs as nodes [19], we have suggested to implement an architecture using the *SWIRL* topology with non-identical nodes in the structure of the reservoir. Nodes have been chosen to be either nonlinear ring resonators, linear elements (*e.g.* small waveguides), or semiconductor optical amplifiers. We have suggested this structure to add gain in the inner structure of the reservoir, in order to compensate for the losses induced by the splitters, combiners, and interconnection waveguides.

We have demonstrated through intensive numerical simulations that this structure can perform well on two typical tasks: the delayed XOR task, and the 3-bit pattern recognition task, at 20Gb/s and 30Gb/s. We have also investigated two input strategies, and have confirmed the work presented by A. Katumba *et al.* [20] on the best injection strategy in a 4×4 *SWIRL* network used as a reservoir computer. With our injection parameters, we attain state-of-the-art levels of performance on the 3-bit pattern recognition task when we inject only on the four central nodes, with an averaged optical input power of 0.6 mW and an electrical pump of 0.0675 A per SOA.

From our numerical investigation, we have found an interesting phenomenon. When injecting the data on all nodes, the system performs well on the delayed-XOR task and badly on the 3-bit pattern recognition task,

but when we inject on a smaller number of nodes (the four central nodes) the system performs well for the pattern recognition task, and badly for the delayed-XOR task. Our explanation for this phenomenon is the trade-off between the non-linearity of the nodes and the linear memory capacity of a reservoir computer [3, 37]. Indeed, when we inject on all nodes, the SOAs are excited with more power, and thus behave more non-linearly than when we inject only on the four central nodes. Hence the system, when injected on all nodes, performs better on the very non-linear XOR task, and cannot perform well the pattern recognition task that needs more linear memory. And on the contrary, when we inject only on the four central nodes, the SOAs receive less power, and thus behave more linearly. Hence the system performs better on the task that needs linear memory but worse on a task that demands more non-linearity.

However, the numerical results on this reservoir computing architecture do not show a clear improvement of the performance of such a structure in comparison with (i) the work of [20], and (ii) the work with only nonlinear ring resonators. We think that in this structure, the non-linearity imposed by the photo-detection overwrites the non-linearity in the inner structure of the reservoir.

On-chip time-delayed reservoir computer using a nonlinear ring resonator as physical node : After investigating these two extended structure, we have come to the conclusion that the extended *SWIRL* structure can perform well on very simple tasks at very high speed, but is limited in terms of scalability of the number of nodes, mostly due to the losses in the structure. This limit in the number of nodes we can use in the reservoir drastically reduce the scope of possibilities in terms of complexity of the tasks. Hence we have suggested an on-chip implementation of the well-studied time-delay reservoir computing, using a nonlinear ring resonator as physical nodes, and multiple round-trip in a relatively small delay line to distribute the virtual nodes. This idea of using multiple round-trips in a delay line to distribute the virtual nodes arose from both the mathematical modelling of such a system, and the work of K. Takano *et al.* in [38].

We have numerically shown that it is possible to attain state-of-the-art levels of performance on more complex tasks like chaotic time series prediction, using a very simple system, made only of one nonlinear ring resonator, one splitter, one combiner, one SOA, and one waveguide. More specifically, before testing this reservoir architecture on benchmark tasks, we have investigated the memory capacity of the system, which is the capacity of the system to reconstruct past inputs. We have found for optimized inter-node value and

injection parameters a memory capacity of 15, meaning that at a given time, the system holds information on the 15 precedent injected bits. Then, in order to compare the performance of this architecture with our extended reservoir made of 16 interconnected nonlinear ring resonators, we investigate through numerical simulations the performance of the reservoir on the delayed-XOR task, and find that this kind of reservoir performs with the same accuracy on this Boolean task. However, due to the drawback of time-delay reservoir, this application is not relevant as this delay-based reservoir computer only operates at 0.3 Giga symbols per second. However, this kind of architecture is relevant for memory demanding tasks, such as the Santa Fe chaotic time series prediction task. We show through intensive numerical simulations that this structure can successfully resolve this task, with a normalized mean square error of 1.12×10^{-2}, at 0.3 Giga symbols per second. This level of performance can be consistently compared to delay-based systems using semiconductor lasers as nonlinear nodes, and a large number of virtual nodes distributed in a delay line made of optical fibers [38–40], with lower processing speeds (of the order of tens of Mega symbols per second).

In addition to the simplicity of design for this integrated photonic circuit, this structure offers a very straightforward scalability of the number of nodes. Indeed, increasing the number of nodes can be done without changing the architecture of the circuit but only in distributing the nodes on a larger number of round-trips in the delay line, inducing the typical drawback of time-delay reservoir computing, which is a reduction of the processing speed.

Conclusion : In this dissertation, we have suggested, presented and numerically investigated three new designs for all-optical on-chip reservoir computing. These architectures rely on the implementation of very compact systems integrated on a silicon photonic chip. In all of these architecture, the building block of the artificial neural network is the nonlinear ring resonator, which is a well-known and commonly used element. We have suggested a 16-node reservoir computer made of 16 ring resonators interconnected according to the *SWIRL* topology as an extension of the work by our close collaborators at the Ghent University. Then we have suggested two architectures to improve the performance of this reservoir computer. The first is based on the same *SWIRL* topology, and the implementation of a neural network with non-identical nodes. The second architecture is based on the time-delayed paradigm for reservoir computer with a ring resonator as physical nodes. This architecture allows for a virtualization of the nodes on multiple round-trips in a relatively small delay line, hence this design is very compact and allows for a scalability of the number of nodes.

References

[1] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[2] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[3] H. Jaeger, *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001, vol. 5.

[4] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[5] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[6] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[7] L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Preprint*, pp. 1–25, 2018.

[8] F. Schürmann, K. Meier, and J. Schemmel, "Edge of chaos computation in mixed-mode vlsi-a hard liquid," *Advances in neural information processing systems*, pp. 1201–1208, 2005.

[9] D. Nikolić, S. Haeusler, W. Singer, and W. Maass, "Temporal dynamics of information content carried by neurons in the primary visual cortex," *Advances in neural information processing systems*, pp. 1041–1048, 2007.

[10] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, pp. 1–6, 2012.

[11] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[12] 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*, vol. 16, no. 15, p. 11182, 2008.

[13] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

[14] C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," *JOSA B*, vol. 30, no. 11, pp. 3048–3055, 2013.

[15] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[16] R. Modeste Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van Der Sande, "Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

[17] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[18] X. Bao, Q. Zhao, H. Yin, and J. Qin, "Recognition of the optical packet header for two channels utilizing the parallel reservoir computing

based on a semiconductor ring laser," *Modern Physics Letters B*, vol. 32, no. 14, p. 1850150, 2018.

[19] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[20] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[21] www.phresco.eu, accessed: 2019.

[22] https://cordis.europa.eu/project/rcn/198823/factsheet/en, accessed: 2019.

[23] T. Van Vaerenbergh, "All-optical spiking neurons integrated on a photonic chip," *PhD Thesis*, 2014.

[24] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[25] L. Zhang, Y. Fei, Y. Cao, X. Lei, and S. Chen, "Experimental observations of thermo-optical bistability and self-pulsation in silicon microring resonators," *JOSA B*, vol. 31, no. 2, pp. 201–206, 2014.

[26] W. H. Pernice, M. Li, and H. X. Tang, "Time-domain measurement of optical transport in silicon micro-ring resonators," *Optics express*, vol. 18, no. 17, pp. 18 438–18 452, 2010.

[27] L. Zhang, Y. Fei, T. Cao, Y. Cao, Q. Xu, and S. Chen, "Multibistability and self-pulsation in nonlinear high- q silicon microring resonators considering thermo-optical effect," *Physical Review A*, vol. 87, no. 5, p. 053805, 2013.

[28] S. Chen, L. Zhang, Y. Fei, and T. Cao, "Bistability and self-pulsation phenomena in silicon microring resonators based on nonlinear optical effects," *Optics Express*, vol. 20, no. 7, pp. 7454–7468, 2012.

[29] I. S. Amiri, R. Ahsan, A. Shahidinejad, J. Ali, and P. P. Yupapin, "Characterisation of bifurcation and chaos in silicon microring resonator," *IET Communications*, vol. 6, no. 16, pp. 2671–2675, 2012.

[30] J. Hryniewicz, P. Absil, B. Little, R. Wilson, and P.-T. Ho, "Higher order filter response in coupled microring resonators," *IEEE Photonics Technology Letters*, vol. 12, no. 3, pp. 320–322, 2000.

[31] H. J. Caulfield, R. A. Soref, and C. S. Vikram, "Universal reconfigurable optical logic with silicon-on-insulator resonant structures," *Photonics and Nanostructures-Fundamentals and Applications*, vol. 5, no. 1, pp. 14–20, 2007.

[32] H. Kishikawa, T. Kondo, N. Goto, and S. Talabattula, "Optical thresholder consisting of two cascaded mach–zehnder interferometers with nonlinear microring resonators," *Optical Engineering*, vol. 56, no. 8, p. 086101, 2017.

[33] Y. Dumeige, L. Ghisa, and P. Féron, "Integrated all-optical pulse restoration with coupled nonlinear microring resonators," *Optics letters*, vol. 31, no. 14, pp. 2187–2189, 2006.

[34] C. Tanaram, C. Teeka, R. Jomtarak, P. Yupapin, M. Jalil, I. Amiri, and J. Ali, "Ask-to-psk generation based on nonlinear microring resonators coupled to one mzi arm," *Procedia Engineering*, vol. 8, pp. 432–435, 2011.

[35] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[36] T. Yamane, S. Takeda, D. Nakano, G. Tanaka, R. Nakane, S. Nakagawa, and A. Hirose, "Dynamics of reservoir computing at the edge of stability," *International Conference on Neural Information Processing*, pp. 205–212, 2016.

[37] M. Inubushi and K. Yoshimura, "Reservoir Computing beyond Memory-Nonlinearity Trade-off," *Scientific Reports*, vol. 7, no. 1, pp. 1–10, 2017.

[38] K. Takano, C. Sugano, M. Inubushi, K. Yoshimura, S. Sunada, K. Kanno, and A. Uchida, "Compact reservoir computing with a photonic integrated circuit," *Optics express*, vol. 26, no. 22, pp. 29 424–29 439, 2018.

[39] L. Appeltant *et al.*, "Reservoir computing based on delay-dynamical systems," *These de Doctorat, Vrije Universiteit Brussel / Universitat de les Illes Balears*, 2012.

[40] K. Hicke, M. A. Escalona-Morán, D. Brunner, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Information processing using transient dynamics of semiconductor lasers subject to delayed feedback," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 4, pp. 1 501 610–1 501 610, 2013.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

GENERAL INTRODUCTION

"Telephone did not come into existence from the persistent improvement of the postcard."

— Amit Kalantri, *Wealth of Words*

When we look at any object around us, we can immediately recognise this object, with no confusion between a chair and a table for example. However, if we want to write a software for a computer to perform this exact human-like task, it is way more difficult. One could extract the typical properties of both objects (*e.g.* the size, the shape, if there is a backrest or not), and try to write an algorithm to recognise these properties in an image. But if we want to add a new object to this image classification software, let's say a sofa, we will have to extract the particular properties of this new *class*, and add it to our software. This would become an incredibly unsatisfying solution for a large number of objects.

This image classification task is a very good example of human-like task the human brain is able to perform perfectly on, while an explicit software on a computer would not be efficient at all. Among those human-like tasks, we can mention speech or hand-writing recognition, car steering, medical diagnosis and so on. With the purpose of trying to replicate the ease of the human

brain to perform on this kind of tasks has emerged a new field of research in computer science based on cognitive computing or neuromorphic computing, with concepts like *Artificial Intelligence* or *Machine Learning*. The amount of research related to this new field of computer science has grown exponentially over the past twenty years, in particular with the exponential increasing of the computational power of data centers.

Nowadays, we find machine learning techniques in our everyday life, with applications far away from typical classification tasks. Say we go to our favorite Internet browser, and send a search request for a few words. The algorithm used to return the best results is based on machine learning algorithm taking into account the most interesting articles according to other users, your personal preferences, and so on. Moreover, your inquiry will be saved to sharpen the algorithm for your upcoming search request, or to perform targeted marketing according to your interests.

These machine learning algorithms are based on a statistical learning of the features allowing the computer to perform the task we want it to perform. If we go back to the example of classification of house furniture (chairs, tables), the typical procedure for the computer to learn how to classify picture will be to feed the machine with thousands of pictures of chairs and tables, with or without the correct label on it, and let the computer learn the features that distinguish one furniture from the other. If this stage, called the training procedure, is successful, the computer should be able to statistically associate unseen images with the correct label. Moreover, the algorithm written to classify the images will not be an explicit algorithm.

Among the approaches to implement machine learning is the so-called artificial neural network. It is inspired by the human brain information processing mechanism, which consists in the interconnection of small computational units called neurons. The training procedure for this kind of cognitive computer consists in the modification of the strength of the connections between consecutive neurons (*weights*), based on a large database of examples. A subgroup of artificial neural networks, called recurrent neural networks, is opposed to the architecture for feedforward neural network as it includes recurrence cycles in the interconnections, thus allowing the network to perform efficiently on tasks

that require memory like speech and pattern recognition, or signal generation, but resulting in a difficult training of the artificial neural network.

Indeed, for typical software-based solutions for machine learning, the training procedure will consists in determining the weights of all the connections of the network. On the contrary, a new paradigm for recurrent neural network was suggested a decade ago. This neural network architecture, called reservoir computing, relies on a fixed neural network, and a simple training procedure which only involves the modification of the weights between a readout layer and the output of the reservoir. This is done using a simple linear regression. Finally, the fixed structure of the recurrent neural network allows for hardware implementations, and many physical systems have been suggested for reservoir computing on various platforms [1–4], including photonic systems [5–8].

In this thesis manuscript, we will investigate various implementations for reservoir computing at the hardware level, and we will specifically focus on implementations using nanophotonics, as this platform offers many advantages over electronic implementations for example. Indeed, light is described by an amplitude and a phase, and considering both amplitude and phase offers a new degree of freedom in the network, which hopefully would improve the computational power of the system. Secondly, the bandwidth of optical systems is several orders of magnitude over the bandwidth of electronic systems. Furthermore, the timescales for the non-linearity of optical circuits can be of the order of the ps, or even fs. Finally, the power consumption of optical systems is several orders of magnitude below the energy consumption of electrical circuits, as the thermal losses are much lower.

However, the design of nanophotonics circuits remains a technological challenge, and it attracts a lot of studies, with the objective of replacing electronic circuits by photonic circuits in telecommunication networks for simple operations like information routing. Many of the tasks performed by electronic circuits in telecommunication networks involve decision making, which could be done using artificial neural networks. Hence, the implementation at the hardware level of machine learning techniques, using photonic reservoir computing for example, could be a solution for some of today's challenges, including working at telecommunication data rates, with low energy consumption, and

an easy compatibility with the existing optical fiber network.

This chapter is organised as follows. First, we present in Sec. 1.1 the technological challenges brought by the exponentially growing amount of data generated, processed, and transmitted everyday in today's ultra-connected world. Then, we present in Sec. 1.2 why nanophotonics, and in particular photonic integrated circuits, are very good candidates for the implementation of optical processing units, and in particular reservoir computing. We also present in this section the European H2020 Phresco project, of which this thesis is part. Finally, in Sec. 1.3 and 1.4 we respectively present the context and goal of the thesis on the one hand, and the outline of the dissertation on the other hand.

1.1 Information processing for telecommunications

The amount of data we have at our disposal is exponentially growing. Mainly due to the Internet, but also to local networks based on other telecommunication protocols like Bluetooth or NFC. Nowadays, everyone has a computer, a smartphone, a tablets, and the number of connected objects from the Internet of Things (IoT) is also growing at a very fast rate, with all the home automation environment, the smart watches, smart glasses, the autonomous vehicles, etc. And with the very foreseeable arrival of the 5G technology, this exponential growth of the IoT will explode, with the creation of network of inter-connected objects that will not only communicate one with another, but also be the core of the telecommunication network of tomorrow.

All of these technologies relies on the possibility and the capacity to process this huge amount of data generated and transmitted everyday, hence the emergence of a need for real-time, energy-efficient data processing. These challenges have motivated research in non-conventional information processing, that is a set of techniques for computers and data-centers to perform a variety of new tasks relying not only on explicit computing, but also on cognitive computing for image and speech recognition applications, targeted marketing applications

based on the behaviour of visitors on web-sites, decision making for financial applications, etc. Among the existing techniques, machine learning is a very efficient paradigm for non-conventional computing. It provides, through many possible implementations, a set of techniques to teach a computer to perform complex tasks, such as classification, pattern recognition, or signal generation.

On the other hand, the deployment of optical fiber networks for telecommunication applications as a replacement for copper based networks has attracted a lot of applications for optical communications, hence engaging a lot of research and studies on this subject. In particular has emerged a need to also replace the network core by all-optical elements, that is the information processing unit. Indeed, using photonics components to process the data (for instance routers, switches, etc) could allow for (i) an improvement of the data rate and (ii) an improvement of the energy consumption balance.

1.2 Nanophotonics and photonic integrated circuits

In photonic, we study the interaction between light and matter in various media, through studies on the generation, on the propagation, and the space-time dynamics resulting from these interactions. From these studies have emerged many applications in various domain such as sensing, telecommunications, lighting, photovoltaics, and so on. For most of these applications, the wavelengths of interest are comprised in the visible (from 380 nm to 780 nm) and the near infrared (from 780 nm to 2,000 nm).

Recently, in order to miniaturise the components, a lot of studies have been done on the integration of optical circuits on photonic integrated circuits, allowing for the integration of many components on a single photonic chip. These chips are cheaper, more robust, consume less power than free-space components, and now target application in the terahertz region [9], in quantum information processing [10, 11], or re-configurable core for signal processing [12].

In particular, silicon photonics has drown a lot of attention as it is a very

good light guider in the near infrared at telecommunication application wavelengths (1,300 nm and 1,500 nm). Moreover, the process to create miniaturised circuits in Silicon are very well-understood and robust as it benefits from the electronics industry expertise. Hence, the field of silicon photonics begins to have a bit of maturity and many applications have already been developed thanks to the recent advances in the integration of light sources on silicon photonics [13].

1.2.1 Optical reservoir computing

Reservoir computing implementation using photonic systems has been the subject of a very large number of studies. Indeed, thanks to the universality of the concept of reservoir computing, any system can be operated as a cognitive hardware computer if driven by a sufficiently well formatted data stream. And as stated beforehand, photonics systems are very fast systems for information transmission and processing, hence a large variety of photonics systems for reservoir computing have emerged. We can cite optoelectronic oscillators [14–18], laser with optical feedback [19–24], semiconductor laser networks [25, 26], large-scale extended networks [27] and photonic integrated circuits [7, 28–33]. Silicon-on-insulator on-chip reservoir computers are very good candidate for telecommunications, as they can operate at telecommunication data rates, and at low energy consumption [30]. Moreover, this technology is inherently compatible with existing optical telecommunication networks.

Other paradigms for optical cognitive computing have emerged very recently, like spiking optical networks [34, 35], or the very interesting project of the startup LightOn [36], which is to implement an Optical Processing Unit (OPU) for neuromorphic computing and image processing, with a cloud solution and a python interface for online research. Their technology for image recognition is based on matrix multiplications using a dispersive media as a random matrix. More info on their technology can be found on their website [37].

1.2.2 Phresco H2020 Project

The Phresco (acronym for **PH**otonic **RES**ervoir **CO**mputing) project is an European H2020 Project [38, 39] in collaboration between the Katholieke Universiteit Leuven (KUL), the Universiteit Gent (Ghent University), IBM Research GMBH in Zurich, IHP GMBH (Innovations for High Performance Microelectronics in Leibniz), and CentraleSupélec. The goals of the project are depicted below, according to the official description of the project [39].

New computing paradigms are required to feed the next revolution in Information Technology. Machines need to be invented that can learn, but also handle vast amount of data. In order to achieve this goal and still reduce the energy footprint of Information and Communication Technology, fundamental hardware innovations must be done. A physical implementation natively supporting new computing methods is required. Most of the time, CMOS is used to emulate *e.g.* neuronal behavior, and is intrinsically limited in power efficiency and speed.

Reservoir computing is one of the concepts that has proven its efficiency to perform tasks where traditional approaches fail. It is also one of the rare concepts of an efficient hardware realization of cognitive computing into a specific, silicon-based technology. Small RC systems have been demonstrated using optical fibers and bulk components. In 2014, optical RC networks based integrated photonic circuits were demonstrated. The PHRESCO project aims to bring photonic reservoir computing to the next level of maturity. A new RC chip will be co-designed, including innovative electronic and photonic component that will enable major breakthrough in the field. We will (i) Scale optical RC systems up to 60 nodes (ii) build an all-optical chip based on the unique electro-optical properties of new materials (iii) Implement new learning algorithms to exploit the capabilities of the RC chip.

The hardware integration of beyond state-of-the-art components with novel system and algorithm design will pave the way towards a new era of optical, cognitive systems capable of handling huge amount of data at ultra-low power consumption.

1.3 Context and goal of the thesis

The work presented in this dissertation should be put in perspective with the European H2020 Project Phresco (**PH**otonic **RES**ervoir **CO**mputing) [38, 39]. This project was suggested as an extension of the work done by our close collaborators at the Ghent University on extended reservoir computer made of 16 inter-connected linear nodes [29], and the mains goals of this project have

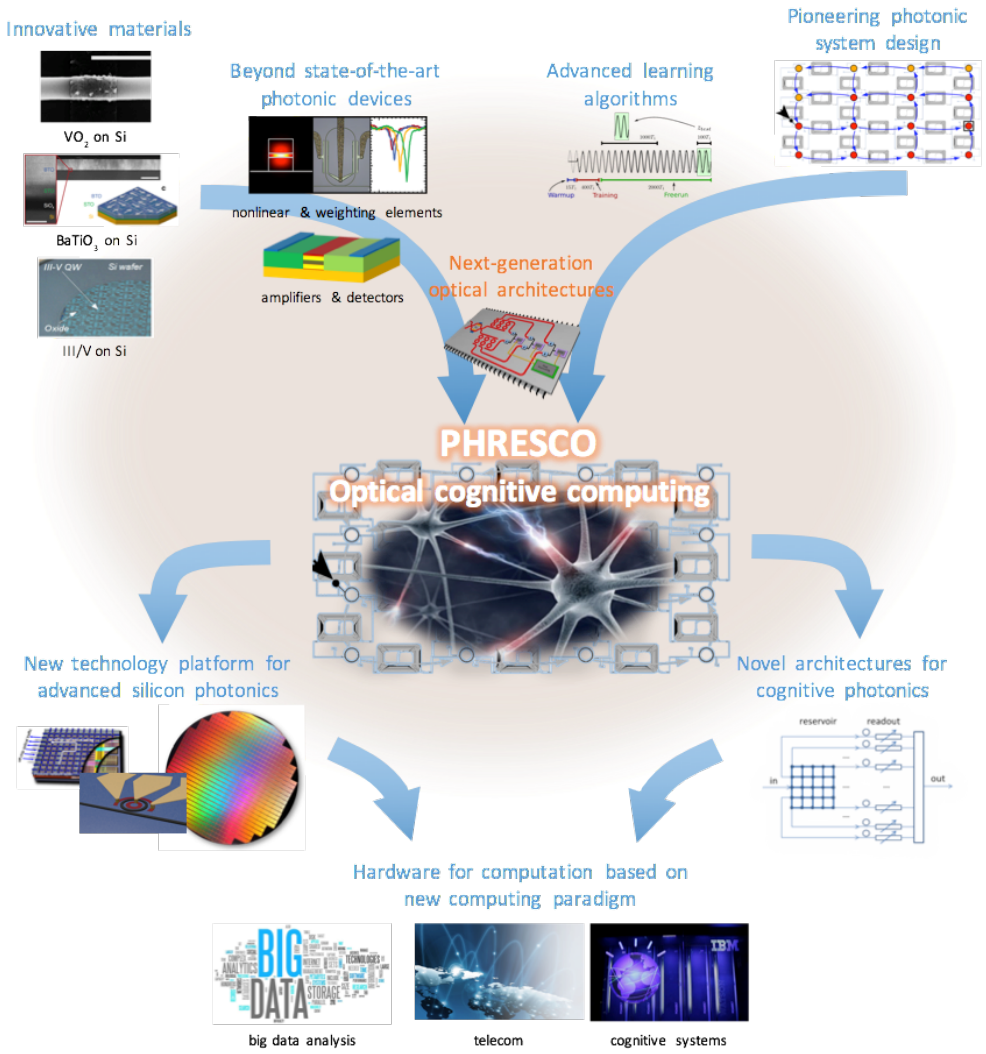

Figure 1.1: Official poster of the Phresco project, presenting graphically the general concepts and goals of the project.

been presented in Sec. 1.2.2.

The work of this thesis was part of an exploratory study on the advantages and drawbacks of adding non-linearity in the inner structure of the reservoir. Indeed, in the work of [29], the neurons are linear photonics components (small waveguides), and the non-linearity of the system is left to the photo-detection. However, it is often suggested that having nonlinear nodes in the structure of the reservoir is necessary for better performance. Hence, in this work, we have replaced the linear nodes by (i) only nonlinear nodes, and (ii) non-identical nodes with a panel of linear and nonlinear nodes.

We have based our architectures on a very commonly used component, that is the nonlinear ring resonator. This integrated element exhibits rich nonlinear dynamical behaviors [34, 35, 40–44]. Silicon-on-insulator microrings resonators are mostly used as optical filters [45], but can also be integrated in more complex architectures and perform other types of all-optical information processing such as Boolean functions [46], thresholding [47], pulse restoration [48], or ASK-to-PSK conversion [49].

In this context of proposal for new structures for all-optical neuro-inspired computers, we have also suggested a new architecture for on-chip reservoir computing based on the well-known time-delay paradigm [50] with the possibility to distribute the nonlinear nodes of the architecture in multiple round-trips in the delay line. The physical nonlinear node is still a ring resonator in this architecture, and the model of this component submitted to optical feedback allows this distribution of the nodes.

This thesis was mostly a numerical simulation work, using a python library to model the optical components, developed by Luceda Photonics [51]. And a few experimental measurement campaigns on the Phresco Chip have been realised at our collaborator's lab at the Ghent University for the benefit of the project.

1.4 Thesis outline

This document is organised according to the following plan.

We introduce in Chap. 2 the general concepts of silicon-based integrated optics, and in particular we focus on the modelling techniques. We also present in this chapter the models of usual integrated components that we will use throughout this dissertation, and we concentrate our attention on the integrated component used as a building block of our structures : the nonlinear ring resonator.

We then introduce in Chap. 3 the field of reservoir computing, and give general information on the training and testing methods for this particular artificial neural network.

Chapter 4 is devoted to the in-depth analysis of the design and the performance of a reservoir computing made of 16 inter-connected nonlinear ring resonators. We will present the structure of the reservoir, test through numerical simulations its performance, and discuss the correlation between the internal dynamics of the building block of the reservoir, and the performance of the aforementioned reservoir when modifying the injection parameters.

We extend in Chap. 5 the work of the precedent chapter by modifying the internal structure of the reservoir. More specifically, we suggest to introduce non-identical nodes in the inner structure of the reservoir, resulting in a reservoir made of 16 inter-connected nodes being either (i) nonlinear ring resonators, (ii) semiconductor optical amplifier, or (iii) small waveguides (linear nodes, as in [29, 30]. We numerically test the performance of such a reservoir computer on typical tasks and attain state-of-the-art levels of performance at very high speed (20Gb/s to 30 Gb/s) with low power consumption.

Chapter 6 is dedicated to the numerical analysis of the performance of a new on-chip architecture based on the well known time-delay paradigm for reservoir computing. Indeed, we suggest in this chapter a very simple structure for on-chip delay-based reservoir computing, made only of a ring resonator, one splitter and one combiner, a relatively small waveguide as delay line, and a semiconductor optical amplifier to compensate for part of the losses in the structure. This structure benefits from its mathematical modelling to be kept relatively small while allowing a very straightforward scale-up of the number of nodes, through the distribution of the virtual nodes in multiple round-trips in the aforesaid delay line. We numerically resolve benchmark

tasks at state-of-the-art levels of performance with this structure.

Finally, chapter 7 presents our conclusions and future perspectives on this work.

References

[1] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[2] L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Preprint*, pp. 1–25, 2018.

[3] D. Nikolić, S. Haeusler, W. Singer, and W. Maass, "Temporal dynamics of information content carried by neurons in the primary visual cortex," *Advances in neural information processing systems*, pp. 1041–1048, 2007.

[4] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, pp. 1–6, 2012.

[5] 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*, vol. 16, no. 15, p. 11182, 2008.

[6] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

[7] C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," *JOSA B*, vol. 30, no. 11, pp. 3048–3055, 2013.

[8] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[9] G. Ducournau, "Silicon photonics targets terahertz region," *Nature Photonics*, vol. 12, no. 10, p. 574, 2018.

[10] X. Qiang, X. Zhou, J. Wang, C. M. Wilkes, T. Loke, S. O'Gara, L. Kling, G. D. Marshall, R. Santagati, T. C. Ralph *et al.*, "Large-scale silicon quan-

tum photonics implementing arbitrary two-qubit processing," *Nature photonics*, vol. 12, no. 9, p. 534, 2018.

[11] D. Bunandar, A. Lentine, C. Lee, H. Cai, C. M. Long, N. Boynton, N. Martinez, C. DeRose, C. Chen, M. Grein *et al.*, "Metropolitan quantum key distribution with silicon photonics," *Physical Review X*, vol. 8, no. 2, p. 021009, 2018.

[12] D. Pérez, I. Gasulla, L. Crudgington, D. J. Thomson, A. Z. Khokhar, K. Li, W. Cao, G. Z. Mashanovich, and J. Capmany, "Multipurpose silicon photonics signal processor core," *Nature communications*, vol. 8, no. 1, p. 636, 2017.

[13] Z. Wang, A. Abbasi, U. Dave, A. De Groote, S. Kumari, B. Kunert, C. Merckling, M. Pantouvaki, Y. Shi, B. Tian *et al.*, "Novel light source integration approaches for silicon photonics," *Laser & Photonics Reviews*, vol. 11, no. 4, p. 1700063, 2017.

[14] R. Martinenghi, S. Rybalko, M. Jacquot, Y. K. Chembo, and L. Larger, "Photonic nonlinear transient computing with multiple-delay wavelength dynamics," *Physical review letters*, vol. 108, no. 24, p. 244101, 2012.

[15] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[16] P. Antonik, F. Duport, M. Hermans, A. Smerieri, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer applied to real-time channel equalization," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2686–2698, 2016.

[17] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," *Scientific reports*, vol. 6, p. 22381, 2016.

[18] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Physical Review X*, vol. 7, no. 1, p. 011015, 2017.

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

[20] K. Hicke, M. A. Escalona-Morán, D. Brunner, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Information processing using transient dynamics of semiconductor lasers subject to delayed feedback," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 4, pp. 1 501 610–1 501 610, 2013.

[21] R. Modeste Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van Der Sande, "Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

[22] J. Nakayama, K. Kanno, and A. Uchida, "Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal," *Optics express*, vol. 24, no. 8, pp. 8679–8692, 2016.

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

[24] ——, "Experimental reservoir computing using vcsel polarization dynamics," *Optics Express*, vol. 27, no. 13, pp. 18 579–18 584, 2019.

[25] D. Brunner and I. Fischer, "Reconfigurable semiconductor laser networks based on diffractive coupling," *Optics letters*, vol. 40, no. 16, pp. 3854–3857, 2015.

[26] D. Brunner, M. Jacquot, L. Larger, and I. Fischer, "A complex network of 1600 holographically coupled opto-electronic oscillators: network dynamics and utilisation for reservoir computing," in *The European*

Conference on Lasers and Electro-Optics. Optical Society of America, 2017, p. CD_10_4.

[27] P. Antonik, N. Marsal, and D. Rontani, "Large-scale spatiotemporal photonic reservoir computer for image classification," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–12, 2019.

[28] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[29] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[30] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[31] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[32] K. Takano, C. Sugano, M. Inubushi, K. Yoshimura, S. Sunada, K. Kanno, and A. Uchida, "Compact reservoir computing with a photonic integrated circuit," *Optics express*, vol. 26, no. 22, pp. 29 424–29 439, 2018.

[33] K. Harkhoe, G. Verschaffelt, A. Katumba, P. Bienstman, and G. V. der Sande, "Demonstrating delay-based reservoir computing using a compact photonic integrated chip," 2019.

[34] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[35] T. Van Vaerenbergh, "All-optical spiking neurons integrated on a photonic chip," *PhD Thesis*, 2014.

[36] https://www.lighton.ai/, accessed: 2019.

[37] https://docs.lighton.ai/notes/opu.html, accessed: 2019.

[38] www.phresco.eu, accessed: 2019.

[39] https://cordis.europa.eu/project/rcn/198823/factsheet/en, accessed: 2019.

[40] L. Zhang, Y. Fei, Y. Cao, X. Lei, and S. Chen, "Experimental observations of thermo-optical bistability and self-pulsation in silicon microring resonators," *JOSA B*, vol. 31, no. 2, pp. 201–206, 2014.

[41] W. H. Pernice, M. Li, and H. X. Tang, "Time-domain measurement of optical transport in silicon micro-ring resonators," *Optics express*, vol. 18, no. 17, pp. 18 438–18 452, 2010.

[42] L. Zhang, Y. Fei, T. Cao, Y. Cao, Q. Xu, and S. Chen, "Multibistability and self-pulsation in nonlinear high- q silicon microring resonators considering thermo-optical effect," *Physical Review A*, vol. 87, no. 5, p. 053805, 2013.

[43] S. Chen, L. Zhang, Y. Fei, and T. Cao, "Bistability and self-pulsation phenomena in silicon microring resonators based on nonlinear optical effects," *Optics Express*, vol. 20, no. 7, pp. 7454–7468, 2012.

[44] I. S. Amiri, R. Ahsan, A. Shahidinejad, J. Ali, and P. P. Yupapin, "Characterisation of bifurcation and chaos in silicon microring resonator," *IET Communications*, vol. 6, no. 16, pp. 2671–2675, 2012.

[45] J. Hryniewicz, P. Absil, B. Little, R. Wilson, and P.-T. Ho, "Higher order filter response in coupled microring resonators," *IEEE Photonics Technology Letters*, vol. 12, no. 3, pp. 320–322, 2000.

[46] H. J. Caulfield, R. A. Soref, and C. S. Vikram, "Universal reconfigurable optical logic with silicon-on-insulator resonant structures," *Photonics and Nanostructures-Fundamentals and Applications*, vol. 5, no. 1, pp. 14–20, 2007.

[47] H. Kishikawa, T. Kondo, N. Goto, and S. Talabattula, "Optical thresholder consisting of two cascaded mach–zehnder interferometers with

nonlinear microring resonators," *Optical Engineering*, vol. 56, no. 8, p. 086101, 2017.

[48] Y. Dumeige, L. Ghisa, and P. Féron, "Integrated all-optical pulse restoration with coupled nonlinear microring resonators," *Optics letters*, vol. 31, no. 14, pp. 2187–2189, 2006.

[49] C. Tanaram, C. Teeka, R. Jomtarak, P. Yupapin, M. Jalil, I. Amiri, and J. Ali, "Ask-to-psk generation based on nonlinear microring resonators coupled to one mzi arm," *Procedia Engineering*, vol. 8, pp. 432–435, 2011.

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

[51] http://www.lucedaphotonics.com/, accessed: 2016.

CHAPTER 2

SILICON-ON-INSULATOR PHOTONICS

"Music is the arithmetic of sounds as optics is the geometry of light."

— C. Debussy

The integration of optical technologies on silicon platforms has attracted a lot of attention lately, through the study and the industrial exploitation of photonic integrated circuits (PIC), resulting from a interesting combination of technological and economical reasons. Indeed, the technological challenge of silicon-on-insulator integrated optics are limited because silicon is a well-understood and robust material that benefits from the electronics industry expertise. Moreover, integrated optics in silicon remains attractive because of the cost of bulk silicon, along with the reduced costs of processing the wafers due to the maturity of the silicon processing technology.

The attractiveness of integrated photonics on the one hand, and on the other hand the very recent development of hardware solutions for machine learning applications, through the implementation of reservoir computing for instance, have led to the field of silicon photonic reservoir computing that we explore in this thesis. However, before we can start to analyse the performance of on-chip reservoir computing architectures, we need to introduce some basics about

integrated optics, and the modelling of usual silicon-on-insulator components.

Hence, we dedicate the chapter to the presentation of the integrated components we are going to use as building blocks of our reservoir architectures. More specifically, after a brief introduction on light guiding and silicon photonic circuits in Sec. 2.1, we present in Sec. 2.2 the modelling strategy, and the *Caphe* photonic circuit simulator library [1] that we use throughout this thesis to run the numerical simulations. We then present in Sec. 2.3 some very common components that we use in all our designs, like optical sources, photodetectors, SOI waveguides, and directional couplers. We dedicate Sec. 2.4 to the presentation of the nonlinear element that will be used as nodes in our neural network architectures, namely the nonlinear ring resonator. More specifically, we will describe its model, and the various nonlinear behaviours that can arise when subjected to optical injection. Finally, we conclude this chapter in Sec. 2.5.

2.1 Introduction to SOI photonic circuits

We give in this section a brief introduction to the light guiding theory, then we give some insight on the experimental study of photonic integrated circuits, and finally we suggest some applications of this technology.

2.1.1 Light guiding on a silicon photonic chip

This section is inspired from the interesting book : "Silicon Photonics, an introduction", by G.T. Reed and A.P. Knights [2]. We suggest the reader to refer to this book for a more complete overview of the guiding theory, the theoretical analysis of the guiding properties of an on-chip waveguide, and the mode propagation in these waveguides.

Light propagation theory is mostly studied using the classical electromagnetic theory, and in particular the rigorous Maxwell's equations. However, these equations are not always necessary to study some basic concepts of light propagation, and the ray optics approach can sometimes allow a reader to understand a propagation phenomenon. In particular, the concept of light guiding

can be easily understood using the typical Snell's law and the total reflection idea.

The idea of using silicon as a substrate for on-chip optical wave guiding applications raised from the transparency of silicon to telecommunications wavelengths (around 1.3 to 1.6 μm), and from the technological knowledge of the standard CMOS-processing techniques, as presented in the introduction of this chapter. Figures 2.1(a) and 2.1(b) present respectively the geometry of a silicon-on-insulator planar waveguide and the geometry of a SOI rib waveguide. In Fig. 2.1(a), the guiding layer in silicon ($n = 3.5$) is constrained between the cladding layer in silicon dioxide (SiO_2, $n = 1.45$), and the air ($n = 1$). In Fig. 2.1(b) the structure of a rib waveguide allows to achieve optical confinement in two dimensions. Indeed, vertically the guiding layer is also constrained between the cladding layer in silicon dioxide and the air, and horizontally the waveguide is surrounded by air.

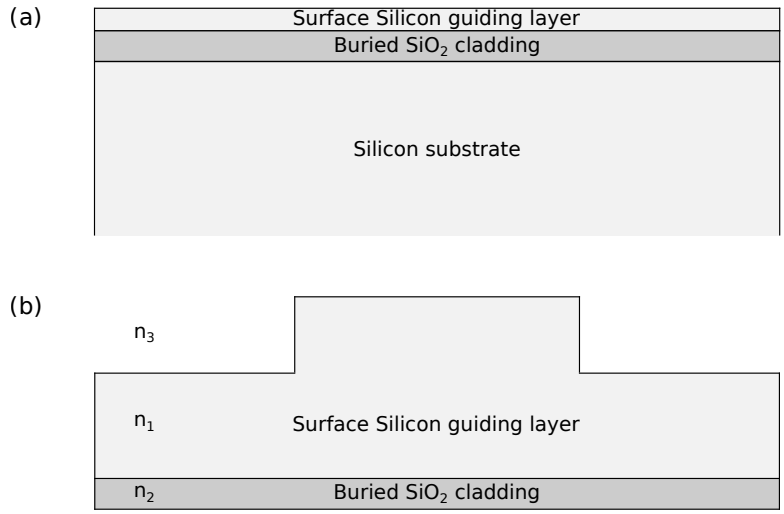

Figure 2.1: (a) Silicon-on-Insulator planar waveguide. The silicon guiding layer is confined between the buried SiO_2 cladding layer and the air. (b) Rib waveguide geometry, used to achieve optical confinement in two dimensions. Figures taken and reproduced from [2].

2.1.2 Experimental study of PICs

The usual process to design an integrated chip is to (i) choose a general integrated circuit according to the optical function we want to perform, (ii) to define a specific design and perform numerical simulations of this design using typical methods like Finite Difference Time Domain (FDTD), and finally (iii) to create the chip and experimentally test it. In this thesis, we have limited ourselves to the first point, but we also performed some experimental studies of photonic integrated chips in collaboration with the Ghent University. We present here the usual experimental setup used to test a photonic chip, and the various coupling techniques between an optical fiber and an integrated waveguide.

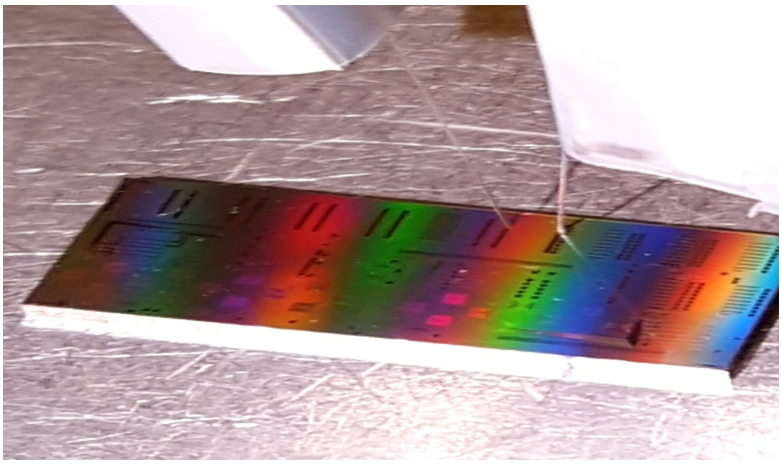

Figure 2.2: Picture of a photonic silicon chip, with the input and output coupling optical fibers.

2.1.2.1 Usual experimental setup

We present in Fig. 2.2 a picture of a silicon photonic chip with the input and output optical fibers. Depending on the study we need to perform on a chip, various experimental setups can be suggested, and we present two different

experimental setups in Fig. 2.3. First, a very simple experimental setup consists of a laser, a polarization controller, the chip, and a power meter. This kind of setup is mostly used for the study the transmission curve of a circuit. Secondly, for more complex studies like information processing, it is necessary to use a more complex experimental setup, as presented in Fig. 2.3(b). This setup is an example and was used with our collaborators at Ghent University to measure the output of an on-chip reservoir computer.

From this experimental setup, we can highlight a very challenging issue of the study of photonic silicon chips : the losses. Indeed, we see that we need one amplifier before the chip, and one after the chip.

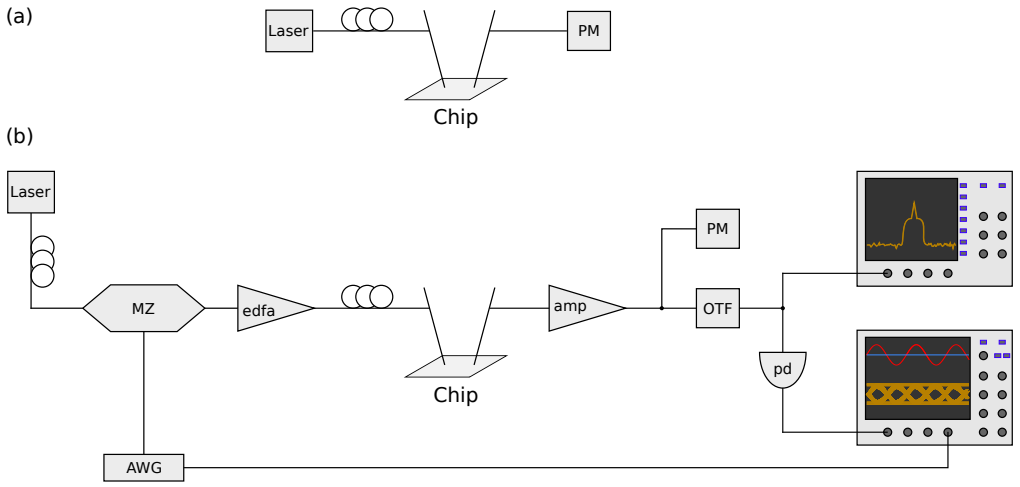

Figure 2.3: (a) Simple experimental setup for the study of a silicon photonic chip. The light from a tunable laser is coupled to the optical circuit after a polarisation controller, and the light is coupled back in an optical fiber for further analysis. (b) Experimental setup for information processing analysis on a photonic chip. The light from a tunable laser is send to a Mach-Zehnder modulator through a polarisation controller. We amplify the light using an erbium doped fiber amplifier (edfa), then couple this light after a polarization controller to the optical chip. The light is coupled back to an optical fiber, amplified, and filtered thanks to an optical tunable filter before it is analysed (optical spectrum analyser and ultra fast photodiode/oscilloscope).

2.1.2.2 Coupling to the optical circuit

Among the losses when studying a silicon-on-insulator photonic chip, the coupling between the chip and the optical fibers at the input and the output are both responsible of 6 dB of losses at best. The best strategy to inject the light from an optical fiber to a chip depends on the packaging of the chip, the test we need to perform, etc. Hence, multiple techniques exist, and we present the four main coupling strategies in Fig. 2.4.

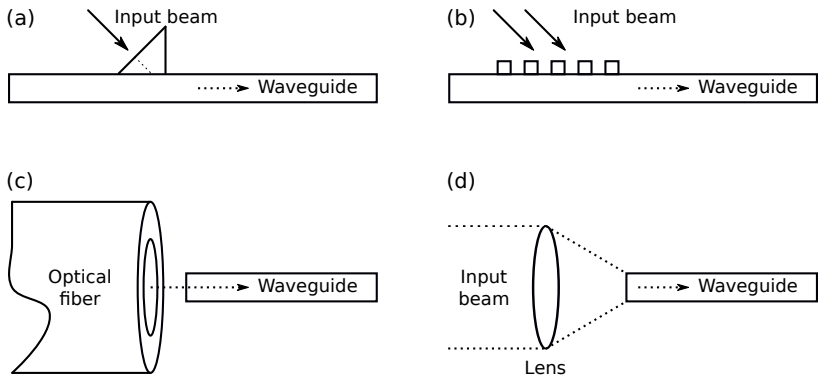

Figure 2.4: Techniques for coupling light to an optical waveguide. (a) Prism coupling, (b) grating coupling, (c) butt coupling, and (d) end-fire coupling. The most usual coupling strategy is to use grating couplers. Figure taken from [2].

The strategies presented in Fig. 2.4(a) and (b) are respectively called the prism coupling and the grating coupling, and are mostly used in testing procedures. The grating coupling is usually implemented as it reduce the losses. The strategies presented in Fig. 2.4(c) and (d) are respectively called the butt-coupling and the end-fire coupling technique, and are implemented for packaged photonic chips.

2.1.3 Example of circuits and components

Silicon photonics is attracting a lot of attention lately. We present here a few example of components and an example of picture of a silicon wafer. These figures are coming from the paper of M. Hochberg *et al.* called "Silicon photonics,

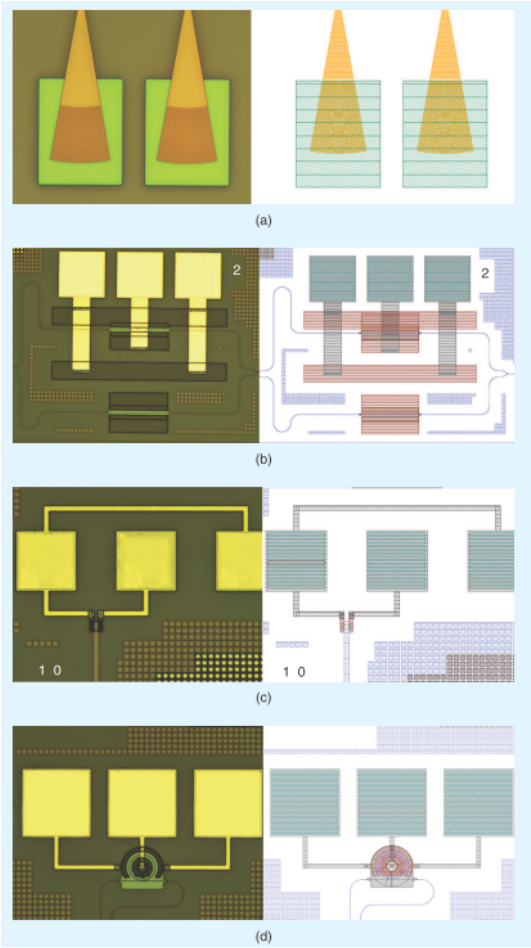

Figure 2.5: Optical micrograph and the picture of the layout of (a) grating couplers, (b) a modulator, (c) a photodetector, and (d) a tunable ring resonator. Figure taken from [3].

the next fabless semiconductor industry" [3], and we suggest the reader to refer to this reference for more information.

We present in Fig. 2.5 the optical micrograph and the picture of the layout of a few silicon photonic components it is now possible to fabricate. Respectively grating couplers (as presented in Sec. 2.1.2.2), a modulator, a photodetector, and a tunable ring resonator. These components were fabricated at IME A'STAR.

We also present in Fig. 2.6 the picture of a silicon wafer with various integrated circuits. This figure was also taken from [3].

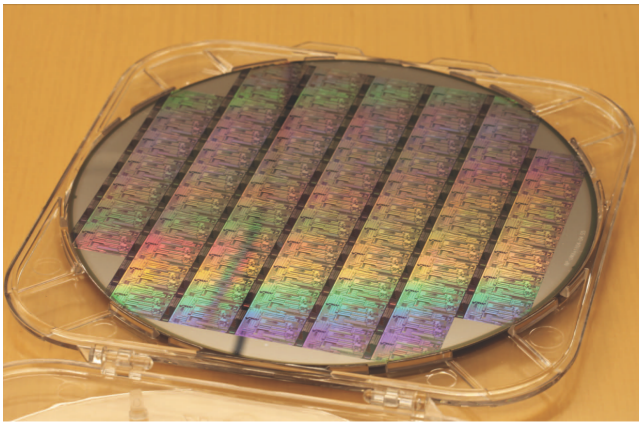

Figure 2.6: Photograph of a SOI wafer with various integrated circuits and components. Figure taken from [3].

2.2 Phenomenological modelling of photonic integrated components

We present in this section the simulation framework deployed to model the integrated components we use in our reservoir computing architectures throughout this thesis. After a brief introduction on ordinary differential equations, we present the *Caphe* photonic circuit simulator [1] used to perform our numerical simulations. More specifically, we present the modelling strategy of this python library, namely the coupled mode theory framework. We also present in Sec. 2.2.4 the concept of hierarchical photonic integrated circuit, allowing to create complex integrated systems from simple building unitary components through the *Caphe* library.

2.2.1 Ordinary differential equations

Natural phenomena are usually theoretically described through the expression of the time evolution of physical variables, called the state variables. The relevant state variables used to model a phenomenon differ from one problem to another, and may vary from fundamental properties like mass and position, to ensemble variables like temperature, or relative properties like phase dif-

ferences. These state variables are usually either real or complex valued, and are described by time-dependant functions measuring those physical variables. Finally, the evolution of these relevant physical variables can often be described with ordinary differential equations (ODEs), whose general expression is given in Eq. (2.1) :

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = F\left(\mathbf{X}(t), t\right), \tag{2.1}$$

where $\mathbf{X}$ is the state variable vector, in which each coordinate correspond to one time-dependant physical quantity describing the phenomenon, and F is the evolution function that embodies the changes of the system with the time t and is determined from fundamental laws and approximations. While F can be relatively simple and thus analytically studied for a large variety of physical phenomena like the typical pendulum, F can also be too complicated, hence requiring numerical tools to study the evolution of the phenomenon.

2.2.2 Simulation methods

For the study of the light propagation in silicon-on-insulator integrated components, the evolution function F is often too complicated for analytical resolution, hence we must use numerical algorithms to study the time evolution of the propagation. Depending on the purpose of the simulation, one can use different numerical approaches to simulate an integrated circuit. Ref. [4] gives a very recent overview of the techniques of the design of silicon photonic devices.

A common family of simulation techniques used to design components is based on discretizing Maxwell's equations on a finite grid both in space and in time. One of these techniques is called Finite Difference Time Domain (FDTD), and can be used both for 2D and 3D simulations. This kind of simulation is very general and can be applied with no restriction of geometry or material properties. However, for complex structures in which light can do multiple round trips in the circuit, the computation time can explode and take hours to days.

At the other end of the range of existing simulation techniques, we find a technique based on modelling integrated components using black boxes without taking into account the spacial dependencies of the system. This technique consists of interpreting the circuit as a collection of waveguides, and use an eigenmode solver to calculate the mode profile of the waveguides. Finally, using dynamical and temporal dependencies of the component through the Couple-Mode Theory (CMT) framework for instance, it is possible to simulate the phenomenological aspects of the circuit, while drastically decreasing the computation time of the simulation.

We give in Fig. 2.7 a graphical example of the difference between those two techniques on a simple ring resonator. In this example, one can simulate the ring using a full vectorial FDTD, or the ring can be considered to be a

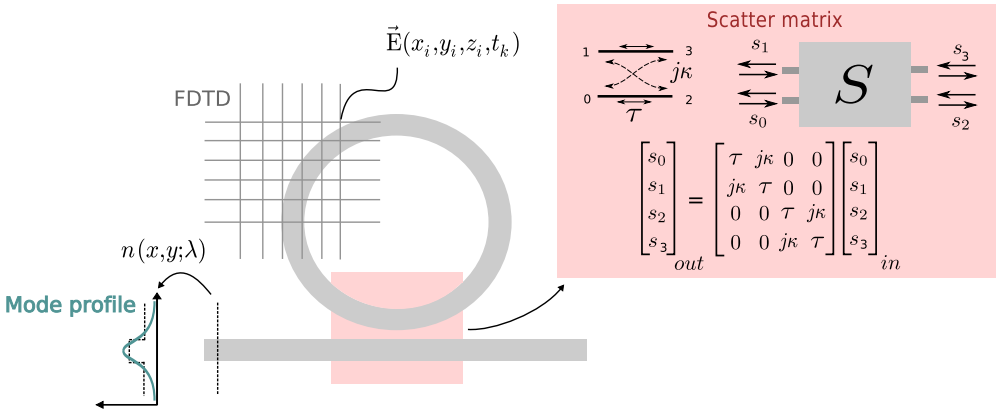

Figure 2.7: A microring can be investigated using different simulation algorithms. A full vectorial FDTD, by discretizing the Maxwell's equations both in space and time will return a detailed distribution of the electromagnetic fields. However, to reduce simulation time, one can also use an eigenmode solver to calculate the mode profile of the waveguides. One can then calculate effective refractive indices based on this mode profile, and use this information in black box models that do not incorporate any spatial information of the field distribution. For instance, the ring can be considered to be a combination of a directional coupler and a waveguide, which can be modelled using a scatter-matrix. Figure taken from [5]

combination of a directional coupler and a waveguide, which can be modelled using a scatter-matrix, and adding nonlinear effects through the CMT model of a ring resonator.

Depending on the purpose of the numerical simulation, one will choose the full FDTD simulation, or the phenomenological representation.In this thesis, we aim to design complex systems for reservoir computing applications, hence we focus on the "black box" approach, and in particular, we use a very accurate python library developed by Luceda Photonics [1] to perform our simulations on a phenomenological level. We present this library, called *Caphe*, in Sec. 2.2.3.

2.2.3 The Caphe library

The *Caphe* library is a python extension of the *Ipkiss* library developed by Luceda Photonics [1] as a toolbox to help researchers and industrial companies to design photonic integrated circuits. The company started as spin-off from imec, the photonics group of the UGent and the VUB, hence benefited from the expertise if the research group.

The very realistic *Caphe* libraries relies on the implementation of the couple mode theory description of integrated components (see Sec. 2.2.4) and allows for the design of the layouts of a component, the frequency-domain and the time-domain numerical simulation of the component, and the integration of single components in much more complex structures (see Sec. 2.2.4.2 on hierarchical integrated circuits).

The main idea of this library is to be able to define arbitrary building block with their own arbitrary number of ports or their own set of ordinary differential equations, and then to couple these components according to the desired topology, thus resulting in a very powerful tool to design and simulate integrated circuits both through frequency-domain time-domain analysis [6].

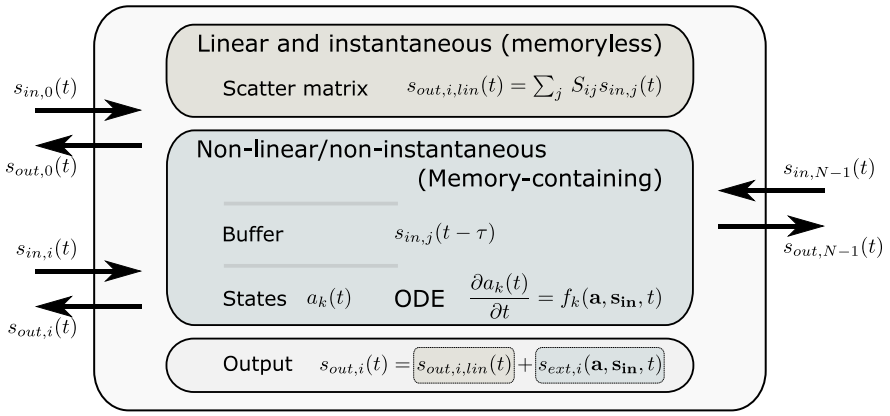

Figure 2.8: Typical description of a node with N ports in *Caphe*. A linear and instantaneous node is described by its scatter-matrix $\mathbf{S}$. State variables (*e.g.* temperature and free carriers), accompanied by the corresponding (nonlinear) ODEs, can be added to this description. This makes the node non-instantaneous. Figure reproduced from [5, 6].

2.2.4 Node description of a photonic integrated component

The implementation of a photonic integrated component in the *Caphe* library is based on its node description. This description is a way to implement the Coupled Mode Theory (CMT, [7, 8]) model of a component. The node description takes into account on the one hand the passive and linear part of the model, embodied in the scatter matrix, and the nonlinear or time-dependant part of the CMT-model of the optical component on the other hand. The node description of a given component in *Caphe* is often summarized in the very general figure given in Fig. 2.8.

More specifically, in the CMT-framework, a node is described by its input-output relationship, and the different state-variables and their respective differential equations. The input-output relationship is defined as a combination of a linear, instantaneous, and memory-less part embodied in a scatter matrix (see Sec. 2.2.4.1), and a nonlinear, non-instantaneous, and memory-containing

part, taking into account the possible time delay induced by a node.

The "node" framework, with this separation of the different parts of the model of a component, allows for a speed-up of the time-domain simulations of networks, as the linear and instantaneous sections of the circuit will be computed only by matrix operations, and the nonlinear and non-instantaneous parts will be computed separately.

2.2.4.1 Scatter matrices

The scatter matrix is a rather elegant way to describe and embody the linear and instantaneous properties of an optical component. With this description, the component is considered to be a black box, with a certain number of ports. A port can account for an optical waveguide mode, or an electromagnetic beam. We assume each port corresponds to an optical mode, and we will define as many ports as the number of modes in the case of a multi-mode simulation.

We can define $s_{in,i}$ (resp. $s_{out,i}$) the complex amplitude of the ingoing (resp. outgoing) normalized electromagnetic mode at each port. In the case of a linear component, the relationship between the output $\mathbf{s_{out}} = (s_{out,0}, ..., s_{out,N-1})$ and the input $\mathbf{s_{in}} = (s_{in,0}, ..., s_{in,N-1})$ is :

$$\mathbf{s_{out}} = \mathbf{S} \times \mathbf{s_{in}}, \tag{2.2}$$

where $\mathbf{S}$ is the scatter matrix of the component, and $\mathbf{S}_{ij}$ is the linear and instantaneous transmission between port $s_{in,i}$ and port $s_{out,j}$.

2.2.4.2 Hierarchical photonic integrated circuits

The node description of an optical component can be extended for circuits with multiple nodes, with the creation of hierarchical nodes. Indeed, as represented in the sketch of Fig. 2.9, when multiple nodes are inter-connected, it is possible to define a global node called hierarchical node.

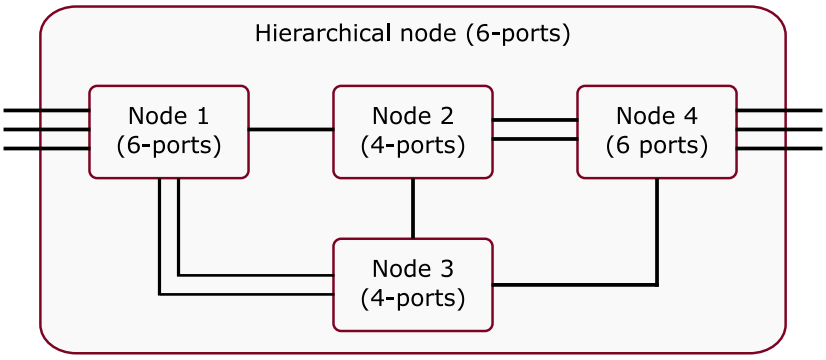

Figure 2.9: Sketch of a hierarchical node made of 4 nodes. The node thus created is now considered as a single 6-port node.

2.3 Usual photonic integrated components

We present in this section some usual integrated photonic components that are used in this thesis. More specifically, we present the photodetector model used in our simulations, and finally the models of optical waveguides, and directional couplers. Note that the main component studied in this thesis is described separately in Sec. 2.4.

In *Caphe*, optical sources are modelled as perfect optical sources and it is possible to feed a component with any arbitrary signal.

2.3.1 Photodetectors

The detector used in our simulations is the same as the one used in previous work [9] and its model is based on the Alphalas UPD-15-IR2-FC photodetector. This takes into account the bandwidth limitation of the detector (modelled by a low-pass filter with a 3 dB cutoff), the response-time limitations, the responsivity, and various noise contributions, including shot noise and thermal noise. The total noise σ_n^2 is given by Eq. (2.3) :

$$\sigma_n^2 = 2qB(\langle I \rangle + \langle I_d \rangle) + \frac{4k_B T B}{R_L}, \tag{2.3}$$

where B is the bandwidth (B = 25 GHz), $\langle I \rangle$ and $\langle I_d \rangle$ are respectively the mean value of the photocurrent and the dark current ($\langle I_d \rangle$ = 0.1 nA), q is the elementary particle charge, k_B is Boltzmann's constant, T is the temperature (in K), and R_L is the load impedance (R_L = 50 Ω). The mean value of the photocurrent is calculated from $\langle I \rangle = r \cdot NEP \cdot \sqrt{B}$ and the values given in the data-sheet of the photodetector : the responsivity r = 0.5 A/W, and the noise equivalent power ($NEP = 10^{-15}$ W/$\sqrt{\text{Hz}}$).

2.3.2 Modelling of an optical waveguide

We have presented in Sec. 2.1 the cross section of a waveguide (see Fig. 2.1(b)). When we inject light in a waveguide of length L, the signal will be delayed of $\tau = n_{eff}(\lambda) \times L/c$, with $n_{eff}(\lambda)$ the effective index of the bulk silicon, and c the speed of light. The amplitude of the signal will also be decreased with a loss coefficient $\alpha(\lambda)$, taking into account the phase shift induced by the light propagation. Finally, the input-output relationship of an optical waveguide is given in Eq. (2.4).

$$s_{out}(t) = \alpha(\lambda) \times s_{in}(t - \tau), \tag{2.4}$$

where $\alpha(\lambda)$ is defined by :

$$\alpha(\lambda) = A(\lambda) \exp\left(-j \frac{2\pi}{\lambda} L n_{eff}(\lambda)\right). \tag{2.5}$$

In this equation, A is the loss factor, of the order of 0.1 to 3.0 dB per centimeter [10]. This high loss factor is the reason most integrated circuits must remain compact.

2.3.3 Modelling of directional couplers

A directional coupler is a three-port component that can be used either as a splitter, or a combiner. It is also known as a Y-junction. This optical component is linear and is considered as instantaneous in our simulations, hence is only described by its scatter matrix, given in Eq. (2.6).

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix}_{out} = \begin{pmatrix} 0 & \sqrt{\beta} & \sqrt{1-\beta} \\ \sqrt{\beta} & 0 & 0 \\ \sqrt{1-\beta} & 0 & 0 \end{pmatrix} \times \begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix}_{in}, \tag{2.6}$$

where $\mathbf{s}_out = (s_{0,out}, s_{1,out}, s_{2,out})$ is the output vector, and $\mathbf{s}_in = (s_{0,in}, s_{1,in}, s_{2,in})$, and β the splitting parameter.

The typical 3dB coupler is a symmetrical coupler, the scatter matrix $\mathbf{S}$ is given in Eq. (2.7). This kind of coupler induce a typical loss of 3dB.

$$\mathbf{S} = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \end{pmatrix}. \tag{2.7}$$

2.3.4 Modelling of a semiconductor optical amplifier

In this dissertation, we will use semiconductor optical amplifiers (SOA). The model of the component we use is the typical model of 1989 from Agrawal *et al.* [11]. We have used this model as it is already implemented in the python library we use for our simulations. Still, it could be interesting to use the model of very recent advanced SOA in future studies.

The model of a SOA - with an input signal s_{in} and an output signal s_{out} - is given in the following equations. The input/output relationship is given by Eq (2.8), and the evolution of the state-variable y, called the integrated gain of the SOA, is given in Eq. (2.9).

$$s_{out}(t) = s_{in}(t-\tau) \times \exp\left(j \times \phi_{offset} - \frac{\alpha \times h}{2}\right), \tag{2.8}$$

where τ is the delay time induced by the propagation of the light in the SOA, ϕ_{offset} is the phase-shift induced by the light propagation in the SOA as a function of its length L and the effective index n_{eff} of the material used to

Table 2.1: Parameters values used in the simulations of the SOA model, from the *Caphe* library.

Parameter	Value
L	500×10^{-6} m
soa_{width}	2×10^{-6} m
soa_{height}	0.2×10^{-6} m
N_0	1×10^{24} m^{-3}
a	2.7×10^{-20}
n_{eff}	3.75
$\tau_{carrier}$	300×10^{-12} s
γ	0.3
α	5

fabricate the SOA, α the linewidth enhancement, and h the real part of the integrated gain of the SOA.

$$\frac{\mathrm{d}y}{\mathrm{d}t}(t) = \frac{1}{\tau_{carrier}} \times \left((g_0 L - h) - \frac{P_{in}}{P_{sat}} \times (\exp h - 1) \right), \tag{2.9}$$

where $\tau_{carrier}$ is the lifetime of the carriers, g_0 is the small-signal gain (defined by $g_0 = \gamma a N_0 (I_{bias}/I_{th} - 1)$, where γ is the confinement factor, a is the differential gain coefficient, N_0 is transparency carrier density, I_{bias} and I_{th} respectively the bias current and the threshold current depending on the geometry of the SOA), L the length of the SOA, $P_{in} = |s_{in}|^2$, and P_{sat} the saturation power of the SOA. In our studies, the parameter we can tune to operate the SOA is the bias current, and the other parameter values are given in Table 2.1.

2.4 Silicon-on-insulator microring resonator

We present in this section the component used as the most important building blocks of the reservoir architectures presented in this thesis. We show in Fig. 2.10 a schematic of a nonlinear ring resonator. This component is a two-

port element where a ring is coupled through evanescent field to a small waveguide with a coupling coefficient κ. In all our simulations, we neglect the back-propagation.

2.4.1 Phenomenological description of a SOI ring resonator

SOI ring resonators are mostly used as optical filters [12], due to the transmission curve of a nonlinear ring resonator (see Fig. 2.11). Still, it can also be integrated in more complex architectures and perform other types of all-optical information processing such as boolean functions [13], thresholding [14], pulse restoration [15], or ASK-to-PSK conversion [16]. Finally, this integrated element exhibits rich nonlinear dynamical behaviors [5, 17–22].

In previous research done by our collaborators at the Ghent University, optical bistability and self-pulsation in a SOI-microring has experimentally been demonstrated [23]. Hence, it is necessary to use a model with enough richness to explain these behaviours. Hopefully, these behaviours can be found when using the CMT-description of a microring, in which we can include several physical effects in a limited number of rate equations.

The typical effects taken into account in the CMT-model of a ring resonator are presented here, with a small phenomenological description of these effects, as presented in [5]. A first important effect in bulk silicon is two photon

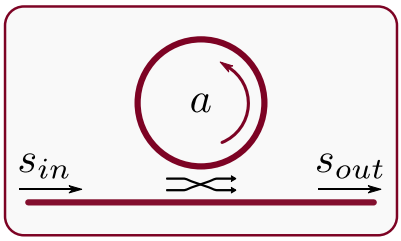

Figure 2.10: Schematic of a nonlinear ring resonator, where a ring is coupled through evanescent waves to a small waveguide.

absorption (TPA), which generates free carriers. These free carriers are then able to absorb light by free carriers absorption (FCA). In addition, the presence of free carriers causes a blueshift in the wavelength by free carriers dispersion (FCD). In SOI ring resonators also surface state absorption (SSA) at the silicon-silica interface is present, and at the same time some light is lost due to surface scattering and radiation loss [23].

The absorbed optical energy is mainly lost by thermo-optic effect, and the heat induced by these effects results in a redshift in the resonance wavelength. The free carriers typically relax at least one order of magnitude faster than the temperature.

2.4.2 Model of a nonlinear microring resonator

The theoretical framework we use is based on the well-established coupled-mode theory (CMT). The model described in our work has already been proposed and was able to correctly describe for the SOI microrings a wide range of dynamical behaviors observed experimentally [5, 17].

The input/output relation is given in Eq. (2.10), in which s_{in} is the input signal (with $P_{in} = |s_{in}|^2$ the input power), s_{out} the output signal (with $P_{out} = |s_{out}|^2$ the output power), ϕ_c the phase propagation in the bus waveguide, κ the coupling between the bus waveguide and the microring, and a the complex

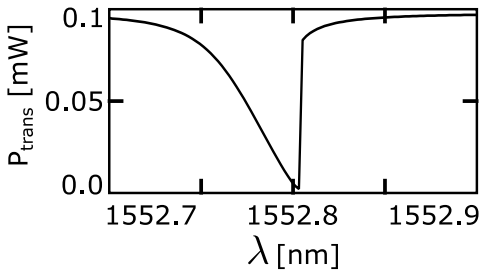

Figure 2.11: Transmission curve of a nonlinear ring resonator, obtained by the numerical simulation of the CMT model of a ring resonator using a Runge-Kutta 4 integration algorithm.

amplitude of the optical mode in the cavity (with $|a|^2$ the energy in the cavity).

$$s_{out} = e^{j\phi_c} s_{in} + \kappa a. \tag{2.10}$$

The state variables of the SOI nonlinear microring resonator within the CMT-framework are : a the complex amplitude of the optical mode, ΔT the mode-averaged temperature difference between the circular waveguide of the microring and its surroundings, and N the number of free carriers. These variables account for the physical effects taking place in a nonlinear microring resonator : specifically, (i) the two-photon absorption (TPA), which generates free carriers; (ii) the free carrier absorption (FCA) (*i.e.*) absorption of light by the free carriers; (iii) the free carrier dispersion (FCD) and (iv) losses.

The nonlinear dynamical equations controlling the temporal evolution of the three state variables are given in Eqs. (2.11)-(2.13), with typical time scales $\tau_a \approx 21$ ps, $\tau_{th} = 65$ ns, and $\tau_{fc} = 5.3$ ns.

$$\frac{\mathrm{d}a}{\mathrm{d}t} = \left[j(\omega_r + \delta\omega_{nl} - \omega) - \frac{\gamma_{loss}}{2} \right] a + \kappa s_{in}, \tag{2.11}$$

$$\frac{\mathrm{d}\Delta T}{\mathrm{d}t} = -\frac{\Delta T}{\tau_{th}} + \frac{\Gamma_{th}\gamma_{abs}|a|^2}{\rho_{Si} c_{p,Si} V_{th}}, \tag{2.12}$$

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\frac{N}{\tau_{fc}} + \frac{\Gamma_{FCA}\beta_{Si} c^2 |a|^4}{2\hbar\omega V_{FCA}^2 n_g^2}, \tag{2.13}$$

where $\omega = 2\pi c/\lambda$ and $\omega_r = 2\pi c/\lambda_r$ with $\lambda_r = 1552.770$ nm are the frequency of the input light and the resonance frequency of the ring, respectively. The relaxation times for the temperature variations and the free carriers are respectively given by τ_{th} and τ_{fc}. TPA in silicon is governed by the constant β_{Si}. n_{Si}, $c_{p,Si}$, and ρ_{Si}, which are the refractive index, the thermal capacity, and the density of the bulk silicon, respectively. We neglect dispersion, thus the group index n_g is equal to n_{Si}. We also define the effective volumes and confinements for each nonlinear effect : V_{FCA}, Γ_{FCA}, V_{TPA}, and Γ_{TPA}.

Losses also play an important role, as they introduce coupling between the three state variables. The total loss γ_{loss} results from the sum of absorption losses γ_{abs}, coupling losses into the waveguide γ_{coup} (with $\kappa = j\sqrt{\gamma_{coup} e^{j\phi_c}}$), and radiation losses γ_{rad}. The absorption losses in the ring are due to linear

surface absorption, TPA, and FCA, as presented in Eq. (2.14) :

$$\gamma_{abs} = \gamma_{abs,lin} + \Gamma_{TPA}\frac{\beta_{Si}c^2|a|^2}{n_g^2 V_{TPA}} + \Gamma_{FCA}\frac{\sigma_{Si}c}{n_g}N, \tag{2.14}$$

where $\gamma_{abs,lin}$ is the linear absorption constant, and σ_{Si} is the absorption cross section of FCA in silicon. In the case of a critically coupled ring, we also have $\gamma_{coup} = \gamma_{abs,lin} + \gamma_{rad}$.

Table 2.2: Parameters values used in the simulations of the microring model, adapted from [5, 17].

Parameter	Value
β_{Si}	8.4×10^{-12} m·W^{-1}
$^{dn_{si}}/_{dT}$	1.86×10^{-4} K^{-1}
$^{dn_{si}}/_{dN}$	-1.73×10^{-27} m^3
σ_{Si}	10^{-21} m^2
ρ_{Si}	2.33 g·cm^{-3}
$c_{p,Si}$	0.7 J·g^{-1}·K^{-1}
$n_g = n_{Si}$	3.476
η_{lin}	0.4
$\gamma_{abs,lin}$	$^{2\eta_{lin}}/_{205}$ ps^{-1}
γ_{coup}	$^{2}/_{205}$ ps^{-1}
τ_{th}	65 ns
τ_{fc}	5.3 ns
Γ_{th}	0.9355
Γ_{TPA}	0.9964
Γ_{FCA}	0.9996
V_{th}	3.19 μm^3
V_{TPA}	2.59 μm^3
V_{FCA}	2.36 μm^3

Finally, we give in Eq. (2.15) the expression of the nonlinear detuning $\delta\omega_{nl}$, that is caused by the thermo-optic effect and FCD, while the Kerr-effect is here neglected :

$$\delta\omega_{nl} = -\frac{\omega_r}{n_g}\left(\frac{dn_{Si}}{dT}\Delta T + \frac{dn_{Si}}{dN}N\right). \tag{2.15}$$

As for any optical injection study, the two parameters of interest are the input power $P_{in} = |s_{in}|^2$, and the wavelength difference between the injected light and the resonance wavelength of the nonlinear microring resonator, that is the optical detuning $\delta\lambda = \lambda - \lambda_r$. For the other parameters of the model, we use the typical numerical values listed in table 2.2([5, 17]). These values will be later used in all our numerical simulations.

2.4.3 Nonlinear dynamics in a microring resonator

2.4.3.1 Introduction to nonlinear dynamics

Investigations on the dynamical behaviours that can happen in a system can lead to very rich studies, when one is interested in the nonlinear properties of a system. When we study the evolution of a system using its ordinary differential equation (see Sec. 2.2.1) while modifying the value of one or multiple parameters, it may exhibit richer behaviours than the typical stable solution. Some systems can show multi-stable solutions, a periodic evolution, an excitable behaviour, or erratic dynamics (chaos). There are some tools existing for the study of nonlinear systems. We present in this section the two main indicators we use in this dissertation, namely the bifurcation diagram, and the continuation technique.

When we study the dynamical response of a system while modifying one particular parameter, it is possible to plot what we usually call a bifurcation diagram. This diagram is constructed according to the following procedure: we numerically integrate the ordinary differential equation describing the system for each value of the parameter of interest (called bifurcation parameter), and we report on a graph the extrema of the response of the system for each value of this parameter (after deleting the transient). We give in Fig. 2.12 the example of a typical bifurcation diagram obtained by the integration of the logistic map [24].

In this figure, we can see different regions of the bifurcation parameter. Before about 3.0, the output of the system has only one extremum, hence the system is stable. From 3.0 to 3.45, the output of the logistic map has two extrema, hence the population is oscillating between two values. Then from

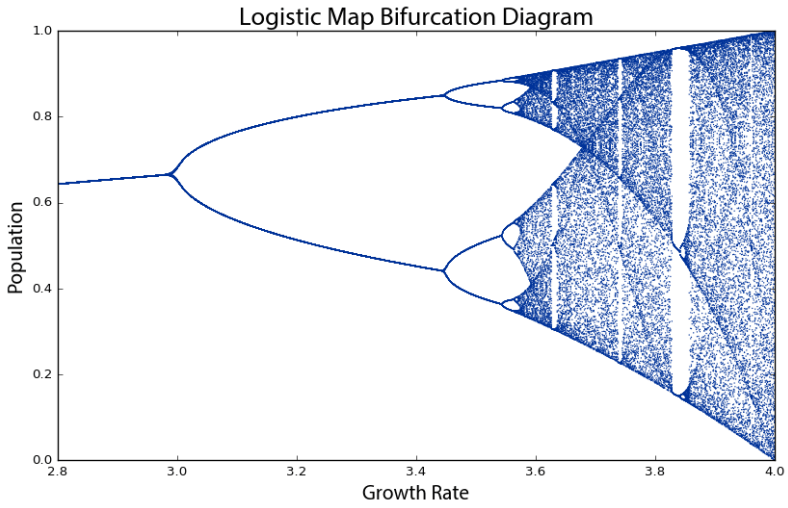

Figure 2.12: Example of a typical bifurcation diagram obtained by the integration of the logistic map [24]. The equation of the logistic map is $x_{n+1} = rx_n(1-x_n)$. The bifurcation parameter is the growth rate r, and this parameter varies from 2.8 to 4.0.

3.45 to 3.55, the output has 4 extrema then 8 extrema. We can deduce that the system is still periodic, but with a more complex dynamics than just simple oscillations. And because after 3.55, the output of the system has a very large number of extrema, we can deduce that (i) after 3.55 the system exhibits chaotic behaviours and (ii) we have between 3.0 and 3.55 a period doubling bifurcation cascade to chaos.

This tool is very interesting for the study of the behaviour of a system when modifying one parameter. Indeed, it can show the values of the bifurcation parameter for which the system is stable, periodic, or erratic. Moreover, the values of dynamical changes in this diagram are called bifurcations. It is possible, using continuation techniques, to follow a bifurcation point in a plane of parameters [25]. This method is used to find the regions where the system is stable, periodic, or chaotic in a parameter plane. These techniques allow for the highlighting of various kind of bifurcations, to study their stability, etc.

The study of the bifurcations of a system is a whole field of expertise, and in this dissertation, we only use this tools as a way to find the stable region of parameters of our nonlinear elements.

2.4.3.2 Nonlinear behaviours of a microring resonator

A nonlinear ring resonator, when submitted to optical injection, can exhibit rich dynamical behaviours, as already presented in a few studies [5, 17–22]. In particular, the nonlinear dynamical properties of the particular model of nonlinear ring resonator we use in our thesis was the subject of an in-depth study by T. van Vaerenbergh and collaborators in the context of the study of excitability of a ring resonator [5, 17], and we suggest the reader to refer to these documents for a complete overview of the dynamical richness of this particular on-chip optical component.

Nevertheless, we summarize here some of the main conclusions of these studies, that we have recomputed with the authorisation of our collaborators at the Ghent University. In particular, we give insights on the mapping of the dynamical states of a ring resonator submitted to optical injection, since the stability is a key information for reservoir computing applications.

We present in Figs. 2.13(a)-(c) three bifurcation diagrams of a nonlinear ring resonator submitted to optical injection, obtained by the simulation of a single, uncoupled, nonlinear microring resonator subjected to steps of optical power between $P_{in_0} = 0$ mW and several maximum values P_{in_1}. The simulations are performed as follow : we integrate the CMT-model of the nonlinear microring resonator (see [5, 17] for the equations and the parameters values) over 2.5 μs with a power step from $P_{in_0} = 0$ mW to the value of P_{in_1} at $t = 100$ ns. We use an Euler integration method with a 1.0 ps integration time step, and a 10.0 ps sampling time. These simulations are performed using the *Caphe* software environment [1].

We then extract from the time series the consecutive extrema for each value of the maximum input power, after deleting the transients. We plot the extrema for each value of the maximum input power at different values of the optical detuning, and obtain the bifurcation diagrams shown in Figs. 2.13(a)-(c), for respectively (a) $\delta\lambda = -50$ pm, (b) $\delta\lambda = 0$ pm, and (c) $\delta\lambda = 50$ pm.

Figure 2.13(a) shows the output power of a microring resonator with an optical detuning $\delta\lambda = -50$ pm, which is always a fixed point for $P_{in_1} < 2.0$ mW. For an optical detuning $\delta\lambda = 0$ pm (see Fig. 2.13(b)), the output power is a fixed

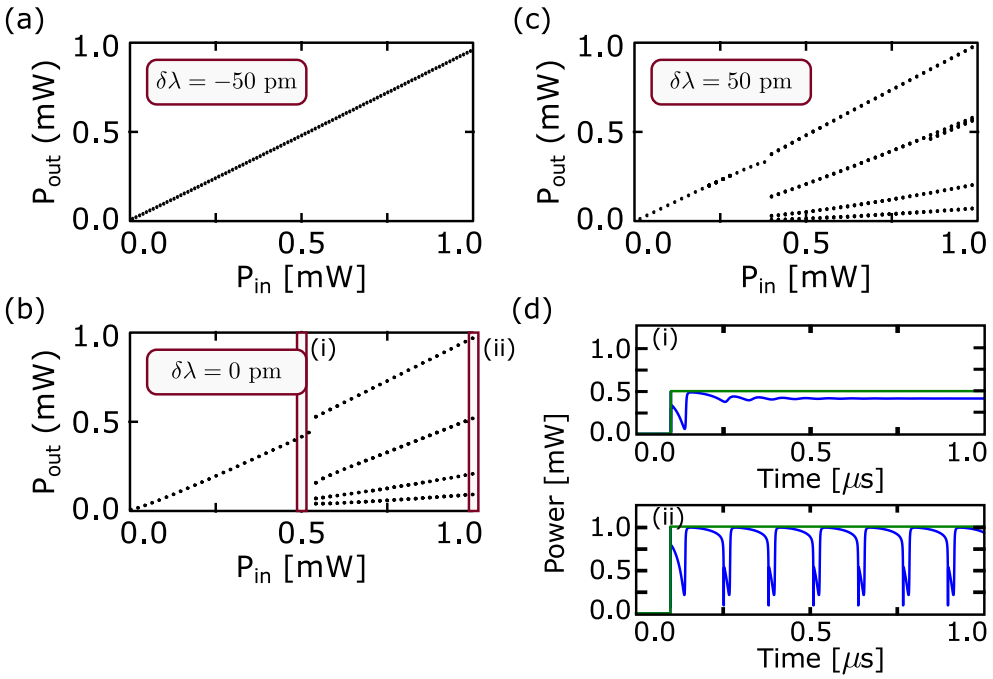

Figure 2.13: Nonlinear dynamics of a nonlinear ring resonator. (a)-(c) Bifurcation diagrams of a single nonlinear microring resonator. The bifurcation parameter is the injected power P_{in} (in mW), and we give the diagrams for various values of the optical detuning : (a) $\delta\lambda = -50$ pm, (b) $\delta\lambda = 0$ pm, and (c) $\delta\lambda = 50$ pm. (d) Times series corresponding to the highlighted points of (b), respectively (i) a fixed point obtained for injection parameters $\delta\lambda = 0$ pm and $P_{in_1} < 0.5$ mW, and (ii) a self-pulsating dynamics obtained for $\delta\lambda = 0$ pm and $P_{in_1} < 1.0$ mW.

point for $P_{in_1} < 0.52$ mW, and a self-pulsation (SP) for $P_{in_1} > 0.54$ mW. Finally, we see in Fig. 2.13(c) that, for an optical detuning $\delta\lambda = 50$ pm, the output power is stationary for $P_{in_1} < 0.38$ mW, and a self-pulsating for $P_{in_1} > 0.40$ mW.

We also present in Fig. 2.13(d) two examples of typical time series at the output of a ring resonator, respectively for (i) a fixed point obtained for injection parameters $\delta\lambda = 0$ pm and $P_{in_1} < 0.5$ mW, and (ii) a self-pulsating dynamics obtained for $\delta\lambda = 0$ pm and $P_{in_1} < 1.0$ mW. These time traces have been used to plot the two highlighted points (i) and (ii) of Fig. 2.13(b).

Finally, in Fig. 2.14, we present a theoretically obtained stability map of a nonlinear microring resonator. This shows the ring's dynamical behavior in the optical detuning/injected power plane, and for a given set of injection parameters. We find three different regions associated to stable fixed points, self-pulsing, and bistability when two different states can be reached depending of the initial conditions. Each region is delimited by bifurcation points, two saddle-node and a supercritical Hopf bifurcation for the bistable and self-pulsing region, respectively. Note that this map was originally presented in a normalized parameter plane [5, 17], but we have recomputed it with continuation techniques [26] and reformatted it with respect to our parameters of interest.

With Fig. 2.14, it is possible to extract the information of Figs. 2.13(a)-(c) for any optical detuning; thus finding the value of injected power for which the microring is on a fixed point close to self-pulsing. This allows to set an optimal operating parameter conditions for the reservoir.

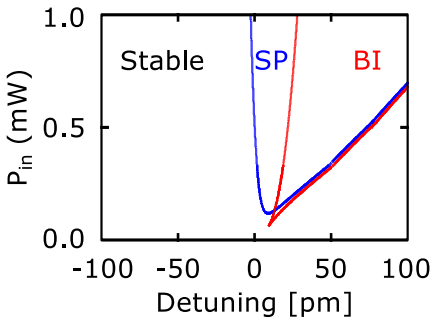

Figure 2.14: Stability map of a nonlinear microring resonator in the ($\delta\lambda$, P_{in}) plane. Figure adapted from [5, 17], using continuation techniques and the *Auto* software [26]

2.5 Conclusions

In this chapter, we have given a small introduction to the concept of light guiding, and its application for on-chip circuits. We have also presented the experimental challenges for photonic integrated circuit testing, that can be summarized in a simple word : losses. These losses are due to the coupling from the input optical fiber to the chip, to the coupling from the chip to the output optical fiber, and on-chip losses in waveguides, and directional couplers for instance.

We have presented our numerical simulation methods for the upcoming chapters. In particular, we have presented the Coupled Mode Theory framework of our simulations, and how it is possible to speed up the simulations using the node description of a component in the *Caphe* library from Luceda Photonics [1]. This description is based on the separation between the linear and instantaneous part of the model on the one hand (embodied in the scatter matrix), and the nonlinear and non-instantaneous part of the model on the other hand. Using this description, and the *Caphe* numerical solver, the linear part of the model is computed only using matrix operations, and the nonlinear part is computed on its own, thus reducing the simulation time. This description also allows for the implementation of hierarchical nodes, which is an interesting tool for the design of complex structures.

Finally, we have presented the models of various integrated components in the CMT framework, and when possible by their scatter matrices. More specifically, in this thesis we need optical waveguides, directional couplers, and nonlinear microring resonators. This last component has been widely investigated as it will be the key component of our reservoir architectures (see Chapters 4, 5, and 6), and we expect for the performance of these reservoir architectures to have a strong correlation with the internal dynamics of their building blocs.

References

[1] http://www.lucedaphotonics.com/, accessed: 2016.

[2] G. T. Reed and A. P. Knights, *Silicon photonics: an introduction*. John Wiley & Sons, 2004.

[3] M. Hochberg, N. C. Harris, R. Ding, Y. Zhang, A. Novack, Z. Xuan, and T. Baehr-Jones, "Silicon photonics: the next fabless semiconductor industry," *IEEE Solid-State Circuits Magazine*, vol. 5, no. 1, pp. 48–58, 2013.

[4] W. Bogaerts and L. Chrostowski, "Silicon photonics circuit design: methods, tools and challenges," *Laser & Photonics Reviews*, vol. 12, no. 4, p. 1700237, 2018.

[5] T. Van Vaerenbergh, "All-optical spiking neurons integrated on a photonic chip," *PhD Thesis*, 2014.

[6] M. Fiers, "Nanophotonic reservoir computing using photonic crystal cavities," Ph.D. dissertation, Ghent University, 2013.

[7] H. Haus, W. Huang, S. Kawakami, and N. Whitaker, "Coupled-mode theory of optical waveguides," *Journal of Lightwave Technology*, vol. 5, no. 1, pp. 16–23, 1987.

[8] H. A. Haus and W. Huang, "Coupled-mode theory," *Proceedings of the IEEE*, vol. 79, no. 10, pp. 1505–1518, 1991.

[9] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

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

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

[12] J. Hryniewicz, P. Absil, B. Little, R. Wilson, and P.-T. Ho, "Higher order filter response in coupled microring resonators," *IEEE Photonics Technology Letters*, vol. 12, no. 3, pp. 320–322, 2000.

[13] H. J. Caulfield, R. A. Soref, and C. S. Vikram, "Universal reconfigurable optical logic with silicon-on-insulator resonant structures," *Photonics and Nanostructures-Fundamentals and Applications*, vol. 5, no. 1, pp. 14–20, 2007.

[14] H. Kishikawa, T. Kondo, N. Goto, and S. Talabattula, "Optical thresholder consisting of two cascaded mach–zehnder interferometers with nonlinear microring resonators," *Optical Engineering*, vol. 56, no. 8, p. 086101, 2017.

[15] Y. Dumeige, L. Ghisa, and P. Féron, "Integrated all-optical pulse restoration with coupled nonlinear microring resonators," *Optics letters*, vol. 31, no. 14, pp. 2187–2189, 2006.

[16] C. Tanaram, C. Teeka, R. Jomtarak, P. Yupapin, M. Jalil, I. Amiri, and J. Ali, "Ask-to-psk generation based on nonlinear microring resonators coupled to one mzi arm," *Procedia Engineering*, vol. 8, pp. 432–435, 2011.

[17] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[18] L. Zhang, Y. Fei, Y. Cao, X. Lei, and S. Chen, "Experimental observations of thermo-optical bistability and self-pulsation in silicon microring resonators," *JOSA B*, vol. 31, no. 2, pp. 201–206, 2014.

[19] W. H. Pernice, M. Li, and H. X. Tang, "Time-domain measurement of optical transport in silicon micro-ring resonators," *Optics express*, vol. 18, no. 17, pp. 18 438–18 452, 2010.

[20] L. Zhang, Y. Fei, T. Cao, Y. Cao, Q. Xu, and S. Chen, "Multibistability and self-pulsation in nonlinear high- q silicon microring resonators

considering thermo-optical effect," *Physical Review A*, vol. 87, no. 5, p. 053805, 2013.

[21] S. Chen, L. Zhang, Y. Fei, and T. Cao, "Bistability and self-pulsation phenomena in silicon microring resonators based on nonlinear optical effects," *Optics Express*, vol. 20, no. 7, pp. 7454–7468, 2012.

[22] I. S. Amiri, R. Ahsan, A. Shahidinejad, J. Ali, and P. P. Yupapin, "Characterisation of bifurcation and chaos in silicon microring resonator," *IET Communications*, vol. 6, no. 16, pp. 2671–2675, 2012.

[23] G. Priem, P. Dumon, W. Bogaerts, D. Van Thourhout, G. Morthier, and R. Baets, "Optical bistability and pulsating behaviour in silicon-on-insulator ring resonator structures." *Optics express*, vol. 13, no. 23, pp. 9623–9628, 2005.

[24] https://en.wikipedia.org/wiki/Logistic_map, accessed: 2019.

[25] E. L. Allgower and K. Georg, *Numerical continuation methods: an introduction*. Springer Science & Business Media, 2012, vol. 13.

[26] http://indy.cs.concordia.ca/auto/, accessed: 2016.

CHAPTER 3

RESERVOIR COMPUTING

"To improve oneself you must be as persistent as the drip, drip, drip of water filling a bucket. Do a little bit, every day."

— Jeffrey Fry

As stated in the general introduction of this dissertation, this thesis aims at designing hardware implementations of machine learning techniques. And reservoir computing is the perfect paradigm to do so, thanks to the general concepts of this particular variation of artificial neural networks. Indeed, a reservoir computing architecture relies on a fixed recurrent neural network and a simplified training procedure of the network, based only on the weighting of the connections between the readout layer and the output of the reservoir.

This chapter introduce the various concepts, mathematical models, and examples to understand the reservoir computing paradigm. However, this dissertation only relies on this particular architecture of neural networks, hence we do not investigate exhaustively machine learning methods, nor mathematical proofs of the properties of these systems. However, we suggest the reader to refer to the recent work of A.E. Hassanien [1] for a complete and up-to-date

overview on machine learning techniques.

This chapter is organised as follows. We briefly introduce in Sec. 3.1 the general concepts of artificial neural networks. Then in Sec. 3.2 we present the particular architecture of artificial neural network called reservoir computing that we investigate in this dissertation. We also introduce in this section a few architectures for reservoir computing that have been investigated, and focus alternatively on architectures relying on various physical platforms on one hand, and on photonic platforms on the other hand. We investigate in Sec. 3.3 the various benchmark tasks we will use to measure the level of performance of our architectures throughout this dissertation. Finally, we conclude this chapter in Sec. 3.4.

3.1 Artificial neural networks

Artificial neural networks have attracted a lot of attention lately, as it provides a straightforward implementation for machine learning techniques. Machine learning is a non-conventional information processing technique which aims at creating a system that is able to perform on specific tasks without *a priori* knowledge of a given input. Instead of an exhaustive mapping of the input/output, the system has learned to generalise from the training examples, and is able to extrapolate on unknown input data. Machine learning is closely related to statistical computing as the output of a trained system is the most statistically probable answer for a given input.

Machine learning and conventional programming are based on very different approaches. Indeed, in machine learning, we do not write an explicit software to solve a specific task, but we let the system modify its own parameters during the training phase in order to find the relation between the input to the output. There are many different methods for the system to find this relationship (*i.e.*) to train the system, and we present here the three main training procedure families. Note that it is also possible to combine those methods for a better accuracy.

- Unsupervised training methods [2] : the system is fed with unlabelled

examples, and has no clue of which kind of results it must get. The main goal of this kind of algorithm is to discover structures or regularities in a data set, without having to explicitly program them.

- Reinforcement learning methods [3] : the system is fed with data, and gets information on how well it behaves. However, the system does not get information on what the exact output should be. The system is given a score for each new input, but never the exact wanted output. The system then learns from rewards and penalties. This training methods is popular for applications which we do not know the exact result of, and the only knowledge is if a given result is good or not. Robotics is a good example of application for these learning methods.

- Supervised learning methods [4] : this set of methods is the most traditional and used way of training a cognitive computer. We feed the system with labelled data *i.e.* with a set of known input data for which we know the desired output. This set of data is called the training set. Knowing the exact desired output for several examples greatly improves the accuracy of the statistical model found by the neuromorphic computer.

As aforementioned, artificial neural networks have obviously emerged as good candidate for solving machine learning problems, as they rely on brain inspired structures. We present in the rest of this section the necessary concepts for understanding neural network architectures.

3.1.1 Introduction to artificial neural networks

An artificial neural network is is in simple words a system that tries to mimic the human brain information processing mechanisms. More specifically, an artificial neural network consists of the interconnection of neuron-like computational units, and we teach the neural network to solve the desired task by modifying the weights of synapse-like interconnections during the training procedure. We show in Fig. 3.1 an usual representation of two neurons interconnected through a synapse [5]. We present in the remaining parts of this

section how we model the different components of this scheme in an artificial neural network.

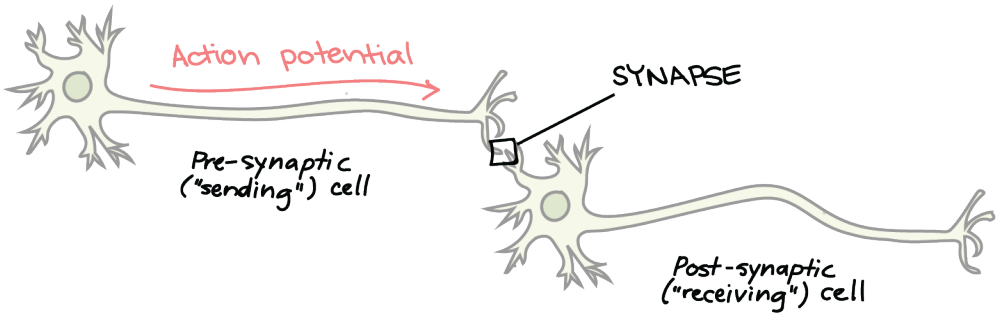

Figure 3.1: Usual representation of two neurons interconnected through a synapse. In an artificial neural network, the neuron-like computational unit are also connected one to another through synapse-like interconnections. Figure taken from [5].

3.1.1.1 Neuron-like computational unit

In an artificial neural network, what we call a neuron is basically a mathematical function that non-linearly transform an input signal to an output signal. We depict in Fig. 3.2 the model of a neuron. A neuron is submitted to a set of inputs, and has one output. And the neuron apply its mathematical model to the sum of the inputs to create the output. Depending on the application, different models for an artificial neuron have been suggested.

- Threshold gate (or perceptrons) are based on the model of McCulloch-Pitts for neurons. They only produce digital outputs. This very simple model take the sum of the inputs, and if this sum is above a certain threshold, the output is "1", "0" otherwise. See panel (i) of Fig. 3.2.

- A second class of model represents analog neurons based on an activation function. The computational unit perform a nonlinear transformation of the weighted sum of the inputs, and give an output according to a continuous activation functions. See panel (ii) of Fig. 3.2. The most

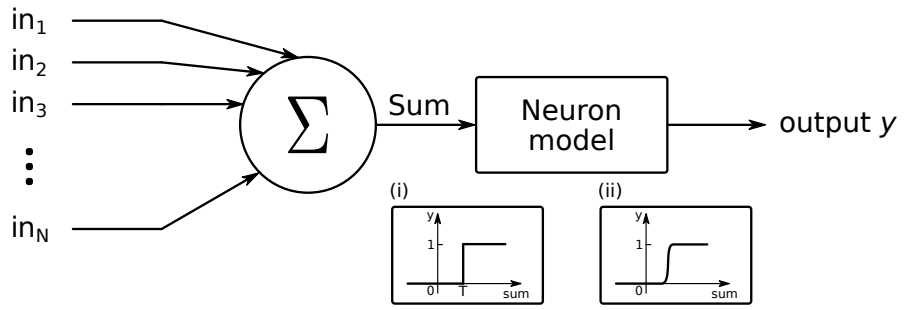

Figure 3.2: Symbolic illustration of the model of a neuron. A neuron is submitted to a set of inputs, and has one output. And the neuron apply its mathematical model to the sum of the inputs to create the output. Various models for the neuron function have been suggested, and we show respectively in (i) and (ii) a digital threshold function, and an analog activation function.

common functions are sigmoid-type functions, such as the logistic (or fermi) function (see Eq. (3.1)), or the tanh function.

$$\text{fermi}(x) = \frac{1}{1+\exp(x)}. \tag{3.1}$$

- A third type of neuron model is called the spiking model. This modelling is biologically more realistic, but the resulting spiking neural network is more complex to model and use in practice, due to the difficulties brought by the data encoding.

In this dissertation, we investigate various neuron-like computational units in our architectures for a hardware implementation of cognitive computing. Since we use physical components as neurons, they are analogically modelled.

3.1.1.2 Synapse-like interconnection unit

In an artificial neural network, the synapse-like connection is ensured only by saying the the output of a neuron is linked to the input of another neuron. This connection is often modelled by its strength, called the weight. Hence, a neuron will receive as an input a weighted sum of the output of the neurons that are connected to him.

Moreover, the training procedure for an artificial neural network consists of finding the appropriate weights for the connections between the neurons of the network in order to perform the desired task.

3.1.2 Neural networks architectures

In an artificial neural network, the neuron-like computational unit are connected one to another through the synapse-like interconnection units. The way they are interconnected defines a topology, that defines both which neuron is connected to which, and which strength of the connection (weight). All of this connectivity information is reported in the weight matrix $\mathbf{W}$, where $\mathbf{W}_{ij}$ determines both the connectivity (a non-zero value means neuron j is connected to neuron i), and the weight of the connection.

Usually, artificial neural networks are divided into two different sub-types of networks, depending on their connectivity. We present these typologies here, which are called feedforward neural networks (see Sec. 3.1.2.1) and recurrent neural network (see Sec. 3.1.2.2).

3.1.2.1 Feedforward neural network

Feedforward neural network are a sub-type of artificial neural network in which there is no recurrent connections in the structure of the network. The nodes (neurons) are distributed according to a layered topology, and the information

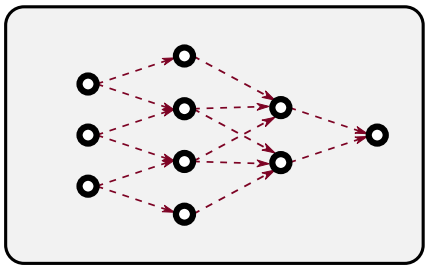

Figure 3.3: Illustration of a feedforward neural network. The information only goes from one layer to the next, and the data only propagate in one direction.

flow unidirectionally from the input layer, to the intermediates layers (also called hidden layers), to the output layer, as presented in the sketch of Fig. 3.3. This topology can also be called multi-layer perceptron.

One fundamental consequence of this topology is that such a neural network cannot store temporal information as the data only flow in one direction. Hence, this architecture is not suited for temporal memory demanding tasks like speech recognition or signal generation for example.

The training of a neural network with this topology is considered to be relatively easy, and the preferred method relies on the error-backpropagation method, in which we adjust the weights of the network through an iterative minimisation of the error between the actual output and the desired output.

This type of neural network is often implemented as it is a very good architecture for decision making and generalisation, in particular with the deep neural networks in which people implement feedforward neural networks with hundreds to thousands of hidden layers.

3.1.2.2 Recurrent neural network

On the other hand, the topology of recurrent neural networks contains recurrence in the inner structure of the neural network, as shown in the illustration given in Fig. 3.4. Hence, the information does not flow unidirectionally, and can go trough a neuron on multiple occasions, thus creating feedback loops

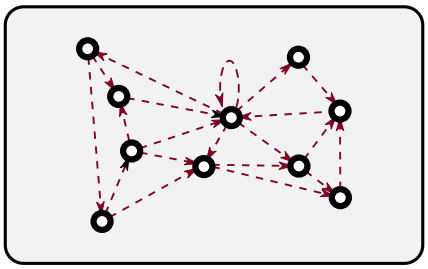

Figure 3.4: Illustration of a recurrent neural network. The information can do recurrent loops in the inner structure of the network.

involving a temporal delay in the network.

The consequence of these feedback loops in the network is that the state of the network now depends not only on the current input, but also of past inputs and past states of the network. Hence, this topology can partially solve the issue raised in the presentation of feedforward neural networks, and this kind of neural network can solve temporal problems, like speech recognition, signal generation, or time series forecasting.

However, even if recurrent neural networks seem to solve the issue raised by the feedforward topology, this is at the expense of the training ease of the network. Indeed, the are few to none learning methods for this kind of network, and they converge slowly. Hence, recurrent neural networks are rarely implemented.

3.2 Reservoir computing

Reservoir computing was proposed about a decade ago [6, 7] as an extension of recurrent neural networks. This new paradigm was suggested as a way to simplify the training procedure of the neural network. Indeed the recurrent neural network is kept fixed and only the connections between the readout layer and the output are trained by a simple linear regression. This new paradigm has attracted a lot of attention recently due to the universality of its concepts [8–10], and is the subject of this dissertation.

3.2.1 General structure of a reservoir computer

Reservoir computing was suggested as an extension of echo-state networks [11] and liquid-state machines [12] - two subgroups of recurrent neural networks - allowing for a simplification of the training procedure. We present here both of these concepts, and the typical topology of a reservoir computer. These three particular architectures for recurrent neural network rely on the same concept that a dynamical system maps the problem to a higher dimensional feature space, and it become easier to separate the features using an hyperplane. We

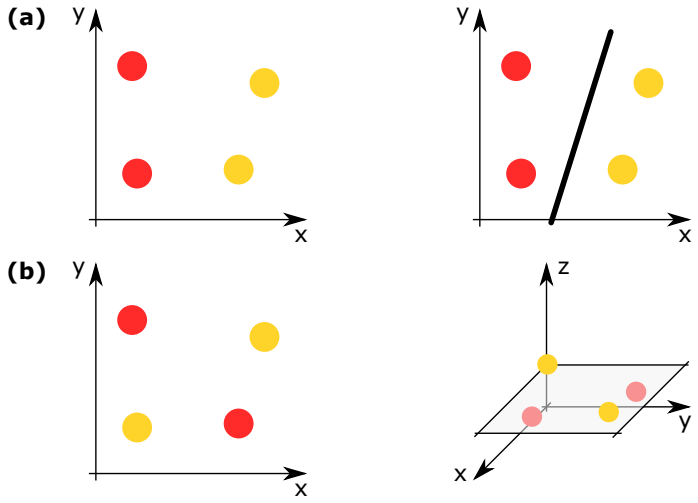

Figure 3.5: In this very simple example, we want to separate in dimension 2 the red disks from the yellow ones. In (a), the task is easy an can be done only by a hyperplane of dimension 1 (a straight line). However in (b), it is not possible to use a straight line to separate the features. But, if we map the features in a higher dimensional space (dimension 3 in this case), it might become possible to use a hyperplane of dimension 2 to separate the red spheres from the yellow ones.

give in Fig. 3.5 a typical illustration of a problem shifted to a higher dimensional space.

Echo State Network : Echo state networks have been introduced in 2001 by H. Jaeger [11]. This network is a recurrent neural network with a random topology and randoms weights describing the connection between the neurons. The data is fed to the network, and we use the states of the system to simply train the system to perform on the desired task, through a simple linear regression. Such a neuro-inspired machine possess fading memory, meaning that after a while the network returns to its stable state and somehow forgets the information it has seen. This kind of network is very attractive as it is quite easy to implement, and successfully resolve a variety of difficult benchmark tasks. However, the optimal dynamical regime for the network to perform well depends on the application, and can be a laborious search of the optimal scaling parameter of the weight matrix.

Liquid State Machine : The liquid state machine concept was introduced by W. Maass in 2002 [12], and is an architecture for recurrent neural networks with a lot of similarities to echo state networks. This architecture is inspired from the biologically realistic spiking neural networks, but with two separated parts. The liquid state machine is formed by a dynamical spiking network, and output neurons that transform the high dimensional transient states given by the spiking network to stable readout values tat can be used for the training of the neural network.

This kind of network relies on its ability to separate different input streams into different trajectories of internal states (the separation property), and on its ability to distinguish and transform different dynamical trajectories into distinguishable outputs (approximation property). While the simplicity of the readout makes it easy to evaluate the properties of this kind of system, it remains difficult to encode and decode the data into train of spikes.

Reservoir Computing : As stated before, reservoir computing was introduce about a decade ago [6, 7] as an extension of these two concepts. It implements a mapping of the features in a higher dimensional space by a fixed recurrent neural network with readout neurons trained to extract those differ-

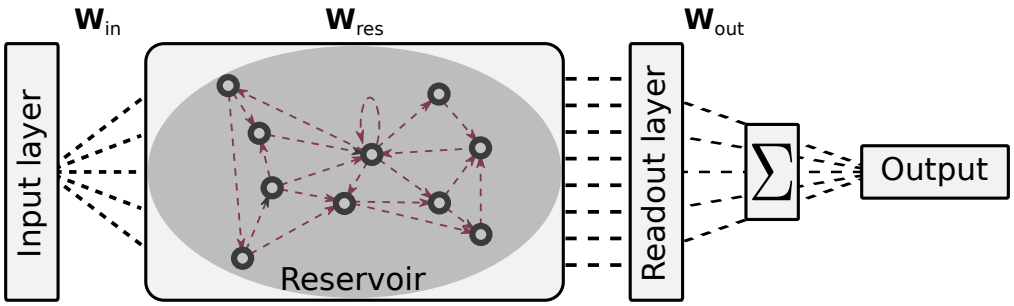

Figure 3.6: General structure of a reservoir computer. The data stream is fed to the system through the input layer. The reservoir is a fixed recurrent neural network, and the connection matrix is defined by the topology of the reservoir. The readout layer consists of the different information we can extract from the network. And finally, the output of the reservoir is constructed as a weighted sum of the readout layer elements.

ent features. Hence, the general structure of a reservoir computer is composed of an input layer, a fixed recurrent neural network called the reservoir, and a readout layer, as depicted in Fig. 3.6.

Again, depending on the task we want to solve, the optimal values for the parameters influencing the dynamical behaviour of the reservoir will change. In particular, the topology of the reservoir, the type of non-linearity, the number of neurons, etc. Nevertheless, for an optimal use of the reservoir, fading memory remains an important feature of a system. Hence, it is necessary to operate the reservoir in a stable dynamical state. However, to improve the memory of the system, it can be interesting for this stable state to have long and rich transient dynamics, thus it can be interesting to operate the reservoir both on a stable state, and close to an unstable state.

3.2.2 Mathematical description of a reservoir computer

We present in this section a more rigorous mathematical description of a reservoir computing system, as the one illustrated in Fig. 3.6. We present in Eq. (3.2) the general state-update equation of a reservoir computer :

$$\mathbf{x}[k+1] = f\left(\mathbf{x}[k], \mathbf{W}_{res}\mathbf{x}[k] + \mathbf{W}_{in}(\mathbf{u}[k+1] + u_{bias})\right), \tag{3.2}$$

where $\mathbf{x}$ is the state of the reservoir, f is a nonlinear vector field that accounts for the nonlinear transformations induced by the reservoir nodes, $\mathbf{u}$ is the input signal to the reservoir, u_{bias} is a bias signal that can be added to the input signal, $\mathbf{W}_{in}$ is the input matrix, representing the input weights, and finally $\mathbf{W}_{res}$ is the interconnection matrix to account for the inner design of the reservoir.

The output $\mathbf{y}_{out}$ of the reservoir is given by Eq. (3.3), and is a linear combination of the states $\mathbf{x}_{readout}$ of the reservoir at the output layer. The output weights of the reservoir are collected in the readout matrix $\mathbf{W}_{out}$, and are determined during the training by a simple ridge regression.

$$\mathbf{y}_{out}[k] = \mathbf{W}_{out}\mathbf{x}_{readout}[k]. \tag{3.3}$$

3.2.3 Training of a reservoir computer

There are two main approaches to train a reservoir computer, either on-line learning or off-line learning. We briefly present in this section the on-line and off-line learning procedure and concepts.

On-line learning : The on-line learning procedure is an incremental technique, which consists of modifying the output weights immediately after the recording of a new sample [13].

Off-line learning : The off-line learning technique for the training of a reservoir computer consists of feeding the system with the whole train of input data, and record the dynamical state of the network at the readout layer $\mathbf{x}_{readout}$. Afterwards, the output weights are determined and stored in the readout matrix $\mathbf{W}_{out}$. The weights are found through a simple ridge regression using the training data and the corresponding desired output [14].

In this thesis manuscript, we will only focus on off-line learning for the training of our simulated reservoir computers. And, in practice for our numerical simulations, the training procedure is done using typical regression functions implemented in Python libraries for machine learning, like *scikit learn* [15].

3.2.4 Hardware implementations of reservoir computing

As stated previously, the main feature of a reservoir computing is that the inner structure of the recurrent neural network is kept fixed during the training procedure. This particular property allows implementations at the physical layer, and indeed implementations of reservoir computing have been suggested on various hardware platforms, including photonics. We present here a few studies on physical implementations of reservoir computing. Nevertheless, we also suggest to the reader to refer to the recent work of G. Tanaka *et al.* [16] for a very complete overview of physical implementations of reservoir computing.

3.2.4.1 Reservoir computing on various hardware platforms

Hardware implementations of reservoir computing have attracted, and still attract a lot of studies on a large variety of substrates, due to the universality of its concepts [8–10]. And a very recent study is even trying to present an unified framework for the characterisation of reservoir computers, with no dependency in the substrate [17].

Among the existing hardware platforms for reservoir computing, implementations using electronic systems have been actively studied. On this substrate, the implementations aim at the design of high-speed, energy-efficient architectures in the electronic domain. Reservoir computer systems using *FPGAs* are widely studied as this substrate is a common hardware device with configurable logic blocks and interconnection structures, and it allows to implement either the artificial neural network [18, 19] or the readout layer [20]. Memristive systems as reservoir computing implementations have also been investigated as they are good candidate for hardware implementations of neural networks [21–28].

Spintronics is also a very promising technology that has been used to implement reservoir computing, in particular through the implementation spin-based reservoir computers with (i) spin oscillations [29, 30], (ii) spin waves [31], and (iii) skyrmions [32].

Another class of substrates widely investigated in recent studies are the mechanical architectures for reservoir computing, with applications in robotics using soft bodies (inspired of octopus limbs) [33–35] or compliant robots [36–40]. The idea of using a physical system to perform the computations in the field of robotics is known as *morphological computing* [41].

Finally, as a way to go further in the idea of using physical bodies to compute, some studies have focus on the implementation of reservoir computing using a biological substrate, such as regions of the brain [42–44], or *in vitro* cultured cells [45–47].

A few studies on the possibility to implement quantum reservoir computing have been done very recently on the theoretical level [48].

3.2.4.2 Photonic reservoir computing

The implementation of reservoir computing techniques using photonic devices is a whole field of research, and has attracted a lot of attention lately, since the photonic substrate allows for very high speed data processing, with very low energy consumption compared to electronic devices for instance. Hence, many studies have arisen using very different architectures and paradigms. We summarise the rich literature of photonic reservoir computing here, and we categorise the implementations in two main fields, namely spatially distributed array reservoirs (or extended reservoirs), and reservoir with delayed feedback. We suggest very complete reviews on this subject in [49, 50], and a tutorial on the implementation of delay-based reservoirs in [51].

Extended designs for reservoir computing : A first category of photonic implementations of the reservoir computing paradigm is based on spatially distributed array reservoirs, in which each neuron-like computational unit in the network is a physical component.

Amongst the implementations, a few designs based on the principles of free-space optics have been suggested. Interactions between a laser beam and a diffrative element can lead to rich spatial nonlinear dynamics, and have been used by several groups to implement reservoir computers [52–55]. This family of implementations is close in terms of concepts to the work done by the startup LightOn on optical processing units [56, 57].

Secondly, the main family of designs for spatially distributed array reservoir computers is based on the on-chip implementation of extended optical circuits. The first implementations have been suggested by the group of the Ghent University in 2008 [58], and this group has been working on this kind of devices ever since [59–62]. Various components have been used as nodes for on-chip extended reservoir computing, including SOAs [59], InGaAsP ring resonators [63, 64], or silicon-on-insulator (SOI) linear nodes [60–62], or crystal cavities [65, 66]. In this thesis manuscript, we investigate on this kind of architectures as an extension of the work done by our collaborators at the Ghent University, and focus in Chap. 4 and 5 on the numerical simulation of

on-chip reservoir computers using respectively (i) SOI ring resonators and (ii) non-identical SOI components.

Time-delayed architectures for reservoir computing : A solution to have a large number of nodes in a reservoir computer was suggested in 2011 by L. Appeltant *et al.* [67], and consist of having only one nonlinear node and distributing a large number of virtual nodes in a long delay line. We focus particularly on this kind of implementations in Chap. 6 of this thesis manuscript.

This kind of delay-based architecture for reservoir computing has attracted a lot of attention lately. In fact, it is the most extensively studied implementation for optical reservoir computers. Systems using opto-electronic systems have been initially suggested [13, 68–76] as a way to implement the paradigm suggested by L. Appeltant *et al.*. In these implementations, either the feedback loop or the nonlinear node can be in the electronic domain. For instance in [68, 69], the nonlinear node is continuous emitting laser, and the feedback loop is and opto-electronic device, while in [72] the feedback loop is in the optical domain and the nonlinear node is opto-electronic. Finally, a few recent works have demonstrated the possibility to implement online learning techniques for opto-electronic reservoir computer [13, 74–76]. Finally, an interesting opto-electronic multimode design based on a Vertical Cavity Surface Emitting Laser (VCSEL) chip array was numerically investigated in [77, 78].

All-optical systems have also been suggested [79–88], using either a continuous emitting laser and an optical gain as nonlinear node in the feedback loop [79, 89], or a laser diodes directly submitted to optical delayed feedback [81, 82, 84, 87, 88]. In the study of J. Vatin *et al.* [87, 88], an improvement of the levels of performance attained by a time-delayed reservoir computer is shown when we use the polarisation dynamics of a VCSEL submitted to optical delayed feedback.

3.3 Tasks and metrics

Multiple indicators are used to benchmark the performance of a reservoir computer. The memory capacity [90] is a good indicator of the capacity of a system to perform on memory-demanding tasks. We also present in this section a few temporal tasks, namely the delayed exclusive-OR task, the 3-bit pattern recognition task, and the one-step Santa Fe chaotic time series forecasting task.

3.3.1 Memory capacity

Many applications of machine learning require a mixing of previous inputs. Time series forecasting and pattern or speech recognition are good examples of these memory demanding applications, and the absence of memory in the system can drastically degrade the performance of the neural network on such tasks. The ability of an artificial neural network, and in particular a reservoir computer, to reconstruct past input signals is measured by the evaluation of the memory capacity. The memory capacity was introduced in 2002 by H. Jaeger in [90], and gives a good insight on how much of past input signals is still in the network at a given time.

Mathematically, the memory capacity is measured according to the following procedure. We first generate a random input time series u, where the u_k are drawn from a uniform distribution in the interval $[-1,1]$. This input signal is then fed to the reservoir, and we successively train the reservoir to construct an infinite number of output series y_i, each being copies of the input time series u shifted by i steps in the past. With these notations, $y_i[k]$ is the reconstruction of $u[k-i]$ for $i = 1...\infty$. Finally, the memory capacity μ_c is defined as the sum of the normalised correlations m_i between the approximation of the targets $\hat{y}_i$ at the output of the reservoir and their associated target y_i, as presented in Eq. (3.4):

$$\mu_c = \sum_{i=1}^{\infty} m_i = \sum_{i=1}^{\infty} \mathrm{corr}[y_i, \hat{y}_i]. \tag{3.4}$$

In practice, it is not necessary to compute this sum for an infinite number of output time series, as it was theoretically demonstrated in [90] that the maximum possible memory capacity is limited by the number of nodes in the neural network. Hence we train our reservoir to successively reconstruct the $u[k-i]$ for $i = 1...N$, with N the number of nodes in the reservoir.

Figure 3.7 is an example of the plot of the normalised correlation m_i as a function of the number of the delayed input steps for the 16-nodes *SWIRL* reservoir computer presented later in Chap. 4. In this example, the measured memory capacity is the sum of the m_i, and is equal to 6.4. We can see from this figure that the reservoir can reconstruct the six past inputs with a normalised correlation m_i greater than 0.6, so we can easily conclude that at a given time, the current signal in the reservoir incorporate information from the six past inputs in the system.

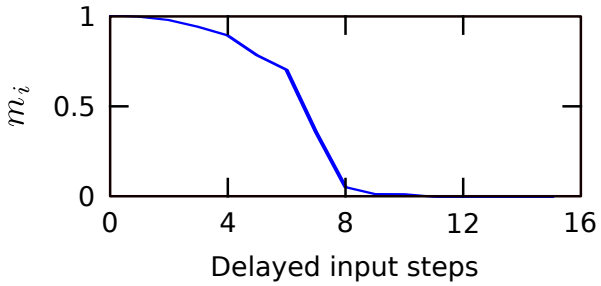

Figure 3.7: Example of the evaluation of the memory capacity of a 16-nodes swirl reservoir computer with nonlinear microring resonators as nodes. We plot the normalised correlation m_i as a function of the delayed input steps. The measured memory capacity is the sum of the m_i, and is equal to 6.4 in this example.

This quantity gives simultaneous indications on the properties (presented in Sec. 3.2) that define a good system as a reservoir computer, namely the consistency and the separability properties. Indeed, the ability to reconstruct a past input comes from these two properties, hence the evaluation of the memory capacity is a good metric to analyse the relevance of using a system as a reservoir computer.

3.3.2 Delayed exclusive OR task

In this dissertation, we benchmark the performance various on-chip architectures for reservoir computer on the binary delayed exclusive OR task (XOR task), defined in Eq. (3.5) and depicted in Fig. 3.8. This task is considered as the most difficult two-bits binary delayed task, due to the nonlinear separability in machine learning terms [61], and is commonly used to benchmark the performance of similar architectures [60–62].

The current output bit $y[k]$ for this task is the Boolean XOR operation between the current input bit $u[k]$ with the bit that is n_{delay} bits in the past $u[k-n_{delay}]$.

$$y[k] = u[k] \oplus u[k-n_{delay}]. \tag{3.5}$$

The level of performance of the reservoir on this task is measured by the evaluation of the bit error rate (BER), defined by Eq. (3.6) as the number of

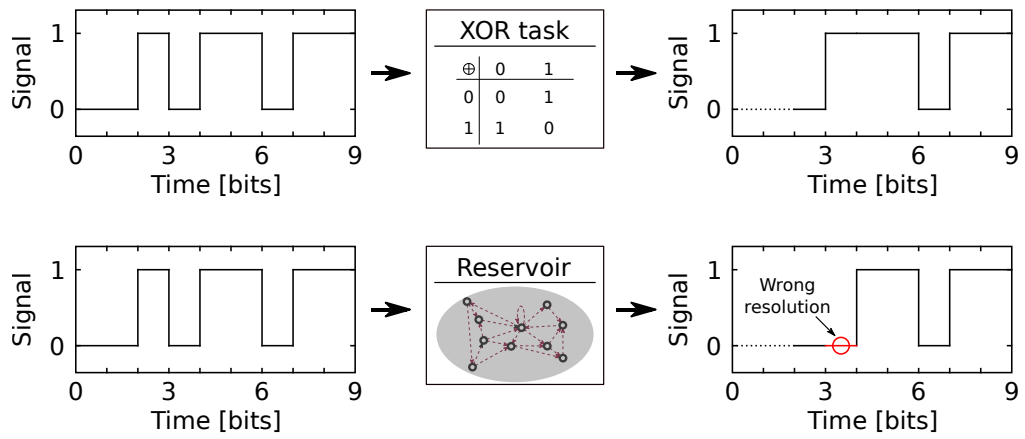

Figure 3.8: Illustration of the delayed XOR task for $n_{delay} = 1$. The target $y[k]$ is the Boolean XOR operation between the current input bit $u[k]$ with the bit that is n_{delay} bits in the past $u[k-n_{delay}]$. The top panel shows the actual output of the 1-bit delayed Boolean XOR operation, and the lower panel shows the resolution of the task by a reservoir computer, with an example of an error.

times the output of the reservoir is different of the target, divided by the total number of tested bits.

$$BER = \frac{\#\ times\ \hat{y}[k] \neq y[k]}{total\ \#\ of\ y[k]}. \tag{3.6}$$

3.3.3 3-bit pattern recognition task

We also benchmark the performance of the reservoir computer architectures on a task with direct telecommunication applications, the binary pattern recognition task. The ability to recognise a bit pattern is essential for data processing, in particular the header recognition process is necessary for information routing.

This task is defined as follows: the current output bit $y[k]$ is equal to one if the 3 (respectively 4) last bits of the input series are forming the pattern we intend to recognise, and is null in any other case. In our simulations, the 3-bit pattern (respectively the 4-bit pattern) we intend to recognise is the pattern [1.0.1] (respectively [1.0.1.1]), hence the current output bit $y[k]$ of the reservoir is given by Eq. (3.7) (respectively Eq. (3.8)) and we illustrate this task in Fig. 3.9.

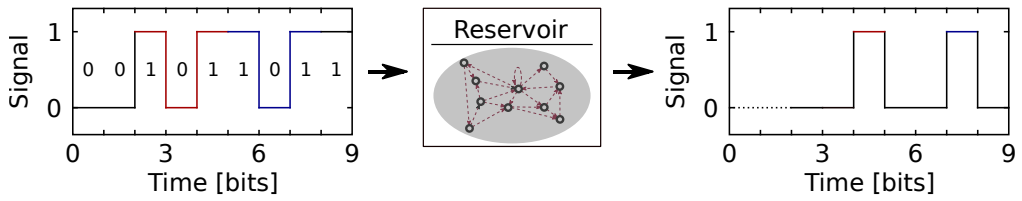

Figure 3.9: Illustration of the 3-bit pattern recognition task. We intend to recognise the pattern [1.0.1]. In this small time series, the pattern can be found two times, highlighted in red and in blue. The output of the reservoir is then one when the last 3 bits of the input series are forming the pattern [1.0.1], and null in any other case.

$$y[k] = \begin{cases} 1 & \text{if } u[k] = 1 \;\&\; u[k-1] = 0 \;\&\; u[k-2] = 1, \\ 0 & \text{else.} \end{cases} \tag{3.7}$$

$$y[k] = \begin{cases} 1 & \text{if } u[k] = 1 \;\&\; u[k-1] = 1 \;\&\; u[k-2] = 0 \;\&\; u[k-3] = 1, \\ 0 & \text{else.} \end{cases} \tag{3.8}$$

The level of performance of the reservoir on this task is also measured by the evaluation of the bit error rate (BER), defined by Eq. (3.6) as the number of times the output of the reservoir is different of the target, divided by the total number of tested bits.

3.3.4 The Santa Fe Task

Finally in Chap. 6 of this dissertation, we test the performance of an on-chip time-delayed reservoir computer. We show through the study of the memory capacity of such a system that this reservoir architecture can perform on more challenging tasks. Hence, we will benchmark our system on the Santa Fe chaotic time series one-step prediction task. This task is usually used to benchmark time-delayed reservoir computers [82, 87, 91–94]. The time series we aim to forecast is composed of a clean low-dimensional nonlinear and stationary time series derived from a laser-generated data set, recorded from a far-infrared laser in a chaotic state. This time series is the first part of the Santa Fe time series prediction competition [95]. We plot in Fig. 3.10 1,000 samples of the *Santa Fe A* time series.

For this task, we feed the Santa Fe times series to the system, and we train the reservoir to predict the next step of the time series. The performance of the reservoir is assessed by measuring the normalised mean square error (NMSE) between the output of the trained reservoir and the actual Santa Fe time series, using the equation given in Eq. (3.9) :

$$NMSE = \frac{1}{N} \times \frac{\sum_{i=1}^{N} (y(i) - \tilde{y}(i))^2}{\sigma_y}, \tag{3.9}$$

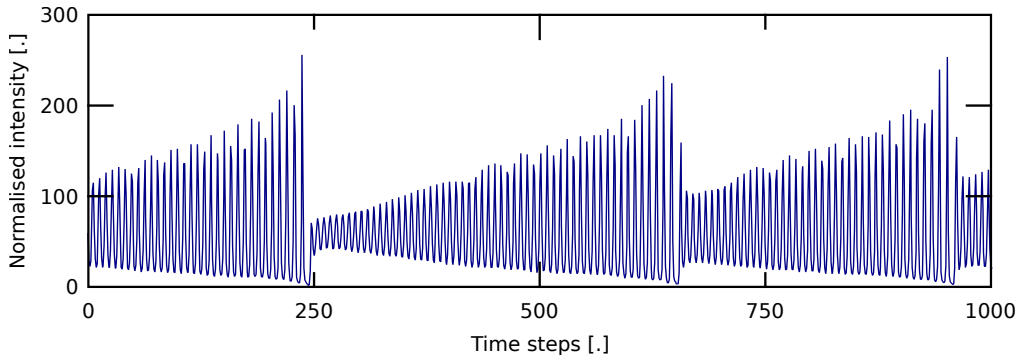

Figure 3.10: Plot of 1,000 samples of the Santa Fe chaotic time series. This temporal sequence is part of the Santa Fe time series prediction contest [95], and is usually referred as the *Santa Fe A* time series.

Delay-based systems can successfully resolve the one-step Santa Fe prediction task with good levels of performance. More specifically, L. Appeltant was able to resolve the task with a NMSE of 0.04 numerically, and 0.124 experimentally [94], K. Hicke resolved the Santa Fe prediction task numerically and attained a NMSE of 0.02 [82], H. Zhang did resolve the task using an on-chip time-multiplexing solution based on multiple nonlinear ring resonators coupled to a single waveguide, and obtained numerically a NMSE of 0.027. Very recently on another on-chip delay-based reservoir, K. Takano experimentally resolved the Santa Fe task with a NMSE of 0.109, and J. Vatin suggested to use the polarization dynamics of a VCSEL to improve the performance and numerically obtained a NMSE of 10^{-3} on a classical fiber optics system with 400 nodes.

3.4 Conclusion

We have presented in this chapter the general framework of reservoir computing as a particular sub-type of recurrent neural network. We have explained why this paradigm is well suited for an implementation at the hardware level, and presented various implementations on different hardware platforms, in-

cluding photonics. In this dissertation, we will focus on implementations using nanophotonic components, and in particular photonic integrated circuits. The various architecture will be presented alternatively in chapters 4, 5, and 6. The benchmark tasks used to asses the level of performance of these various architectures have been presented in this section.

In this dissertation, we have only focused ourselves on a few benchmark tasks, mostly for computational time issues. There are a few other typical benchmark tasks it could interesting to test our reservoir computing architectures on, like the nonlinear autoregressive moving average (NARMA) task, or the nonlinear channel equalisation task, with direct telecommunication applications.

References

[1] A. E. Hassanien *et al.*, *Machine learning paradigms: Theory and application*. Springer, 2019.

[2] H. B. Barlow, "Unsupervised learning," *Neural computation*, vol. 1, no. 3, pp. 295–311, 1989.

[3] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 2, no. 4.

[4] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited„ 2016.

[5] https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse, accessed: 2019.

[6] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[7] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[8] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[9] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[10] L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Preprint*, pp. 1–25, 2018.

[11] H. Jaeger, *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001, vol. 5.

[12] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on

perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[13] P. Antonik, F. Duport, A. Smerieri, M. Hermans, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer," in *International Conference on Neural Information Processing*. Springer, 2015, pp. 233–240.

[14] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[15] http://scikit-learn.org/, accessed: 2016.

[16] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, 2019.

[17] M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer, "A substrate-independent framework to characterize reservoir computers," *Proceedings of the Royal Society A*, vol. 475, no. 2226, p. 20180723, 2019.

[18] J. Zhu and P. Sutton, "Fpga implementations of neural networks–a survey of a decade of progress," in *International Conference on Field Programmable Logic and Applications*. Springer, 2003, pp. 1062–1066.

[19] N. D. Haynes, M. C. Soriano, D. P. Rosin, I. Fischer, and D. J. Gauthier, "Reservoir computing with a single time-delay autonomous boolean node," *Physical Review E*, vol. 91, no. 2, p. 020801, 2015.

[20] P. Antonik, *Application of FPGA to Real-Time Machine Learning: Hardware Reservoir Computers and Software Image Processing*. Springer, 2018.

[21] M. S. Kulkarni and C. Teuscher, "Memristor-based reservoir computing," *Nanoscale Architectures (NANOARCH), 2012 IEEE/ACM International Symposium on*, pp. 226–232, 2012.

[22] J. Bürger and C. Teuscher, "Variation-tolerant computing with memristive reservoirs," in *Proceedings of the 2013 IEEE/ACM International Symposium on Nanoscale Architectures*. IEEE Press, 2013, pp. 1–6.

[23] J. Bürger, A. Goudarzi, D. Stefanovic, and C. Teuscher, "Hierarchical composition of memristive networks for real-time computing," in *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH´ 15)*. IEEE, 2015, pp. 33–38.

[24] J. Burger, A. Goudarzi, D. Stefanovic, and C. Teuscher, "Computational capacity and energy consumption of complex resistive switch networks," *arXiv preprint arXiv:1507.03716*, 2015.

[25] X. Yang, W. Chen, and F. Z. Wang, "Investigations of the staircase memristor model and applications of memristor-based local connections," *Analog Integrated Circuits and Signal Processing*, vol. 87, no. 2, pp. 263–273, 2016.

[26] C. Donahue, C. Merkel, Q. Saleh, L. Dolgovs, Y. K. Ooi, D. Kudithipudi, and B. Wysocki, "Design and analysis of neuromemristive echo state networks with limited-precision synapses," in *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*. IEEE, 2015, pp. 1–6.

[27] C. H. Bennett, D. Querlioz, and J.-O. Klein, "Spatio-temporal learning with arrays of analog nanosynapses," in *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. IEEE, 2017, pp. 125–130.

[28] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, "Reservoir computing using dynamic memristors for temporal information processing," *Nature communications*, vol. 8, no. 1, p. 2204, 2017.

[29] T. Furuta, K. Fujii, K. Nakajima, S. Tsunegi, H. Kubota, Y. Suzuki, and S. Miwa, "Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions," *Physical Review Applied*, vol. 10, no. 3, p. 034063, 2018.

[30] J. Williame, A. Difini Accioly, D. Rontani, M. Sciamanna, and J.-V. Kim, "Chaotic dynamics in a macrospin spin-torque nano-oscillator with delayed feedback," *Applied Physics Letters*, vol. 114, no. 23, p. 232405, 2019.

[31] R. Nakane, G. Tanaka, and A. Hirose, "Reservoir computing with spin waves excited in a garnet film," *IEEE Access*, vol. 6, pp. 4462–4469, 2018.

[32] G. Bourianoff, D. Pinna, M. Sitte, and K. Everschor-Sitte, "Potential implementation of reservoir computing models based on magnetic skyrmions," *AIP Advances*, vol. 8, no. 5, p. 055602, 2018.

[33] J. Kuwabara, K. Nakajima, R. Kang, D. T. Branson, E. Guglielmino, D. G. Caldwell, and R. Pfeifer, "Timing-based control via echo state network for soft robotic arm," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 1–8.

[34] T. Li, K. Nakajima, M. Cianchetti, C. Laschi, and R. Pfeifer, "Behavior switching using reservoir computing for a soft robotic arm," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 4918–4924.

[35] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Information processing via physical soft body," *Scientific reports*, vol. 5, p. 10487, 2015.

[36] K. Caluwaerts and B. Schrauwen, "The body as a reservoir: locomotion and sensing with linear feedback," in *2nd International conference on Morphological Computation (ICMC 2011)*, 2011.

[37] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "The role of feedback in morphological computation with compliant bodies," *Biological cybernetics*, vol. 106, no. 10, pp. 595–613, 2012.

[38] K. Caluwaerts, J. Despraz, A. Işçen, A. P. Sabelhaus, J. Bruce, B. Schrauwen, and V. SunSpiral, "Design and control of compliant tensegrity robots through simulation and hardware validation," *Journal of the royal society interface*, vol. 11, no. 98, p. 20140520, 2014.

[39] G. Urbain, J. Degrave, B. Carette, J. Dambre, and F. Wyffels, "Morphological properties of mass–spring networks for optimal locomotion learning," *Frontiers in neurorobotics*, vol. 11, p. 16, 2017.

[40] K. Fujita, S. Yonekura, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "Environmental and structural effects on physical reservoir computing with tensegrity," *Proceedings of ICISIP*, pp. 197–204, 2017.

[41] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[42] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio, "A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex," 2005.

[43] D. Nikolić, S. Haeusler, W. Singer, and W. Maass, "Temporal dynamics of information content carried by neurons in the primary visual cortex," *Advances in neural information processing systems*, pp. 1041–1048, 2007.

[44] D. Nikolić, S. Häusler, W. Singer, and W. Maass, "Distributed fading memory for stimulus properties in the primary visual cortex," *PLoS biology*, vol. 7, no. 12, p. e1000260, 2009.

[45] B. Jones, D. Stekel, J. Rowe, and C. Fernando, "Is there a liquid state machine in the bacterium escherichia coli?" in *2007 IEEE Symposium on Artificial Life*. Ieee, 2007, pp. 187–191.

[46] R. L. Ortman, K. Venayagamoorthy, and S. M. Potter, "Input separability in living liquid state machines," in *International Conference on Adaptive and Natural Computing Algorithms*. Springer, 2011, pp. 220–229.

[47] A. Didovyk, O. I. Kanakov, M. V. Ivanchenko, J. Hasty, R. Huerta, and L. Tsimring, "Distributed classifier based on genetically engineered bacterial cell cultures," *ACS synthetic biology*, vol. 4, no. 1, pp. 72–82, 2014.

[48] R. Alicki, M. Horodecki, P. Horodecki, and R. Horodecki, "Dynamical description of quantum computing: generic nonlocality of quantum noise," *Physical Review A*, vol. 65, no. 6, p. 062101, 2002.

[49] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[50] P. R. Prucnal and B. J. Shastri, *Neuromorphic photonics*. CRC Press, 2017.

[51] D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger, "Tutorial: Photonic neural networks in delay systems," *Journal of Applied Physics*, vol. 124, no. 15, p. 152004, 2018.

[52] D. Brunner and I. Fischer, "Reconfigurable semiconductor laser networks based on diffractive coupling," *Optics letters*, vol. 40, no. 16, pp. 3854–3857, 2015.

[53] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, "Reinforcement learning in a large-scale photonic recurrent neural network," *Optica*, vol. 5, no. 6, pp. 756–760, 2018.

[54] J. Dong, S. Gigan, F. Krzakala, and G. Wainrib, "Scaling up echo-state networks with multiple light scattering," in *2018 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2018, pp. 448–452.

[55] P. Antonik, N. Marsal, and D. Rontani, "Large-scale spatiotemporal photonic reservoir computer for image classification," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–12, 2019.

[56] https://www.lighton.ai/, accessed: 2019.

[57] https://docs.lighton.ai/notes/opu.html, accessed: 2019.

[58] 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*, vol. 16, no. 15, p. 11182, 2008.

[59] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[60] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[61] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[62] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[63] C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," *JOSA B*, vol. 30, no. 11, pp. 3048–3055, 2013.

[64] C. Mesaritakis, A. Kapsalis, and D. Syvridis, "All-optical reservoir computing system based on ingaasp ring resonators for high-speed identification and optical routing in optical networks," *SPIE OPTO*, pp. 937 033–937 033, 2015.

[65] M. A. A. Fiers, T. Van Vaerenbergh, F. Wyffels, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 344–355, 2014.

[66] F. Laporte, A. Katumba, J. Dambre, and P. Bienstman, "Numerical demonstration of neuromorphic computing with photonic crystal cavities," *Optics express*, vol. 26, no. 7, pp. 7955–7964, 2018.

[67] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information

processing using a single dynamical node as complex system," *Nature communications*, vol. 2, p. 468, 2011.

[68] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing," *Optics express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[69] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, pp. 1–6, 2012.

[70] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[71] S. Ortín, M. C. Soriano, L. Pesquera, D. Brunner, D. San-Martín, I. Fischer, C. Mirasso, and J. Gutiérrez, "A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron," *Scientific reports*, vol. 5, p. 14945, 2015.

[72] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Physical Review X*, vol. 7, no. 1, p. 011015, 2017.

[73] J. Qin, Q. Zhao, H. Yin, Y. Jin, and C. Liu, "Numerical simulation and experiment on optical packet header recognition utilizing reservoir computing based on optoelectronic feedback," *IEEE Photonics Journal*, vol. 9, no. 1, pp. 1–11, 2017.

[74] P. Antonik, M. Hermans, F. Duport, M. Haelterman, and S. Massar, "Towards pattern generation and chaotic series prediction with photonic reservoir computers," *Proc. of SPIE Vol*, vol. 9732, pp. 97 320B–1, 2016.

[75] P. Antonik, F. Duport, M. Hermans, A. Smerieri, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer

applied to real-time channel equalization," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2686–2698, 2016.

[76] P. Antonik, M. Gulina, J. Pauwels, and S. Massar, "Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography," *Physical Review E*, vol. 98, no. 1, p. 012215, 2018.

[77] J. B. Héroux, H. Numata, and D. Nakano, "Polymer waveguide-based reservoir computing," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 840–848.

[78] J. B. Héeroux, H. Numata, N. Kanazawa, and D. Nakano, "Optoelectronic reservoir computing with vcsel," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–6.

[79] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

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

[81] D. Brunner, M. C. Soriano, and I. Fischer, "High-speed optical vector and matrix operations using a semiconductor laser," *IEEE Photonics Technology Letters*, vol. 25, no. 17, pp. 1680–1683, 2013.

[82] K. Hicke, M. A. Escalona-Morán, D. Brunner, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Information processing using transient dynamics of semiconductor lasers subject to delayed feedback," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 4, pp. 1 501 610–1 501 610, 2013.

[83] A. Dejonckheere, F. Duport, A. Smerieri, L. Fang, J.-L. Oudar, M. Haelterman, and S. Massar, "All-optical reservoir computer based on saturation of absorption," *Optics express*, vol. 22, no. 9, pp. 10 868–10 881, 2014.

[84] R. Modeste Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van Der Sande, "Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

[85] J. Nakayama, K. Kanno, and A. Uchida, "Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal," *Optics express*, vol. 24, no. 8, pp. 8679–8692, 2016.

[86] J. Bueno, D. Brunner, M. C. Soriano, and I. Fischer, "Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback," *Optics express*, vol. 25, no. 3, pp. 2401–2412, 2017.

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

[88] ——, "Experimental reservoir computing using vcsel polarization dynamics," *Optics Express*, vol. 27, no. 13, pp. 18 579–18 584, 2019.

[89] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," *Scientific reports*, vol. 6, p. 22381, 2016.

[90] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. GMD-Forschungszentrum Informationstechnik Bonn, 2002, vol. 5.

[91] Y. Kuriki, J. Nakayama, K. Takano, and A. Uchida, "Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers," *Optics express*, vol. 26, no. 5, pp. 5777–5788, 2018.

[92] Y. Hou, G. Xia, W. Yang, D. Wang, E. Jayaprasath, Z. Jiang, C. Hu, and Z. Wu, "Prediction performance of reservoir computing system based on a semiconductor laser subject to double optical feedback and optical injection," *Optics express*, vol. 26, no. 8, pp. 10 211–10 219, 2018.

[93] H. Zhang, X. Feng, B. Li, Y. Wang, K. Cui, F. Liu, W. Dou, and Y. Huang, "Integrated photonic reservoir computing based on hierarchical time-

multiplexing structure," *Optics express*, vol. 22, no. 25, pp. 31 356–31 370, 2014.

[94] L. Appeltant *et al.*, "Reservoir computing based on delay-dynamical systems," *These de Doctorat, Vrije Universiteit Brussel / Universitat de les Illes Balears*, 2012.

[95] A. S. Weigend, *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.

CHAPTER 4

SILICON-BASED RING RESONATORS AS NODES OF AN ON-CHIP RESERVOIR COMPUTER

"I seem to see ahead, in a kind of way. I know we are going to take a very long road, into darkness; but I know I can't turn back."

— J. R. R. Tolkien, *The Lord of the Rings*

The aim of this PhD is to design new architectures of Si-based on-chip reservoir computers for all-optical signal processing applications. In this chapter, we numerically investigate the properties of a 16-nodes reservoir computer in which every neuron-like computational unit is a nonlinear microring resonator, as introduced in Sec. 2.4. In this architecture, the 16 nonlinear nodes are interconnected according to the *SWIRL* topology using integrated splitters, combiners, and waveguides. We suggest this design as an extension of the work previously done by K. Vandoorne *et al.* [1, 2] and A. Katumba *et al.* [3, 4] on *SWIRL* architectures. This previous work done by our collaborators at the Ghent University investigated the performance of a network strictly made of linear nodes, and only using the non-linearity of the photodetectors at the readout layer of the reservoir.

In this work, we investigate the advantages and drawbacks of adding an-

other level of non-linearity in the inner structure of the reservoir, through the implementation of network using nonlinear ring resonators as nodes. This commonly used component [5–15] can exhibit rich nonlinear dynamical behaviours, hence we expect an improvement of the overall performance of the reservoir. We perform an in-depth numerical analysis of the performance of such a reservoir and investigate the impact of typical parameters, namely the injected power, the optical detuning, and possible resonance mismatches between the microring resonators. The performance of our reservoir on benchmark tasks will be compared to the performance of the purely linear reservoir from [2–4] in order to analyse the advantages and the drawbacks of such an architecture.

The performance of the reservoir architecture is assessed on typical benchmark tasks. We compute the memory capacity as an indication of how well the reservoir will perform on memory-demanding tasks such as pattern recognition or time series forecasting. We analyse the performance on the typical binary delayed-XOR task, which is a commonly used task to benchmark this kind of integrated architectures as it demands a strong nonlinear separability in machine learning terms [3]. The performance on this task is measured using the bit error rate (BER), and we demonstrate that our reservoir computer can reach error rate levels lower than 10^{-3} at a data rate of 20 Gb/s, and over a wide range of design parameters. Furthermore, we show that the power consumption required to reach this level of performance is only 0.15 mW per node. We also test our architecture on a task with immediate telecommunication applications, namely the header recognition task. We show that we can successfully resolve the 3-bit and 4-bit pattern recognition task at a data rate of 20 Gb/s.

The rest of this chapter is organised as follows. We first present in Sec. 4.1 the architecture of our 16-nodes reservoir computer. More specifically, we give a detailed overview of the input layer to the reservoir, the inner structure of the reservoir with a complete description of the *SWIRL* topology, and finally the output layer of the reservoir. In Sec. 4.2, we introduce our numerical simulation strategies. In particular, we give an in-depth presentation of the library PhotRC, developed by our collaborators at the Ghent University, and modified to match our needs. We also present the various tasks used to measure the performance

of the reservoir computer, and their corresponding metrics. We conclude this Sec. 4.2 with a description of our training and testing procedures. We devote the section 4.3 to the search of an optimal operating point for our reservoir computer. We demonstrate that this optimal operating point can be empirically determined from the static and dynamical properties of the building block of the reservoir, namely the nonlinear microring resonator. In Sec. 4.4, we investigate through intensive numerical simulations the performance of the reservoir. We first evaluate the memory capacity of our reservoir. Then we study the performance of this architecture on the typical binary delayed-XOR task and examine the impact of various parameters on the reachable levels of performance. Finally, we evaluate the performance of the reservoir on a telecom-oriented task, that is the pattern recognition task. We give in Sec. 4.5 the conclusions of this chapter.

The results presented in this chapter have been published in an article in a peer-reviewed journal and a proceeding of a SPIE Photonics Europe conference [16, 17].

4.1 Architecture of the reservoir computer

This section is dedicated to the description and the design of the architecture of the reservoir computer presented in this chapter. The general structure of a reservoir computer [18, 19] has been investigated in Sec. 3.2.1, and consists of an input layer, the so-called reservoir that is a fixed artificial neural network, and a readout layer. We reproduce in Fig. 4.1 the scheme of the general structure of a reservoir computer.

In this section, we first describe the input layer to the reservoir that represents how the data is injected to the reservoir. Then, we present the inner structure of the reservoir. More specifically, we introduce the *SWIRL* topology, and show how this topology can be integrated on a silicon photonic chip using silicon-on-insulator ring resonators as nonlinear nodes. Finally, we depict the readout layer of our integrated reservoir computer.

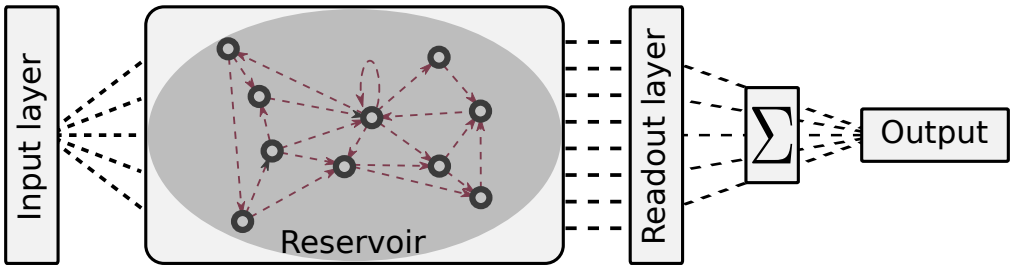

Figure 4.1: General structure of a reservoir computer. The data is send to the neural network from the input layer. The fixed recurrent neural network is called the reservoir. We read the state of the reservoir at the readout layer, and we construct the output of the reservoir as a linear combination of the readout states of the reservoir.

4.1.1 Input layer to the reservoir

The input layer to the reservoir is a representation of how the input signal is fed to the so-called reservoir. In the mathematical framework of reservoir computing, given in Sec. 3.2.2 by Eq. (3.2), this input layer is embodied by the input matrix $\mathbf{W}_{in}$. This input matrix represents the input weights on each node. In our simulations, we inject the same power modulation in all the active nodes with random phase shifts, hence $\mathbf{W}_{in}$ is a 16×16 diagonal matrix with random elements sampled from a uniform distribution over the interval $[-\pi, \pi]$.

4.1.2 Inner structure of the reservoir

We investigate in this section the inner structure of the fixed neural network used as the reservoir of our implementation. Our design is integrated on a silicon photonic chip, hence the structure must to be planar but must also allow maximum signal mixing. Hence, we structure our reservoir computer according to the *SWIRL* topology.

The *SWIRL* approach for the distribution of the nodes in a planar architecture was introduced by K. Vandoorne in [1], and we show in Fig. 4.2(a) a scheme of a 4×4 *SWIRL* network. This topology is one of the few structures that

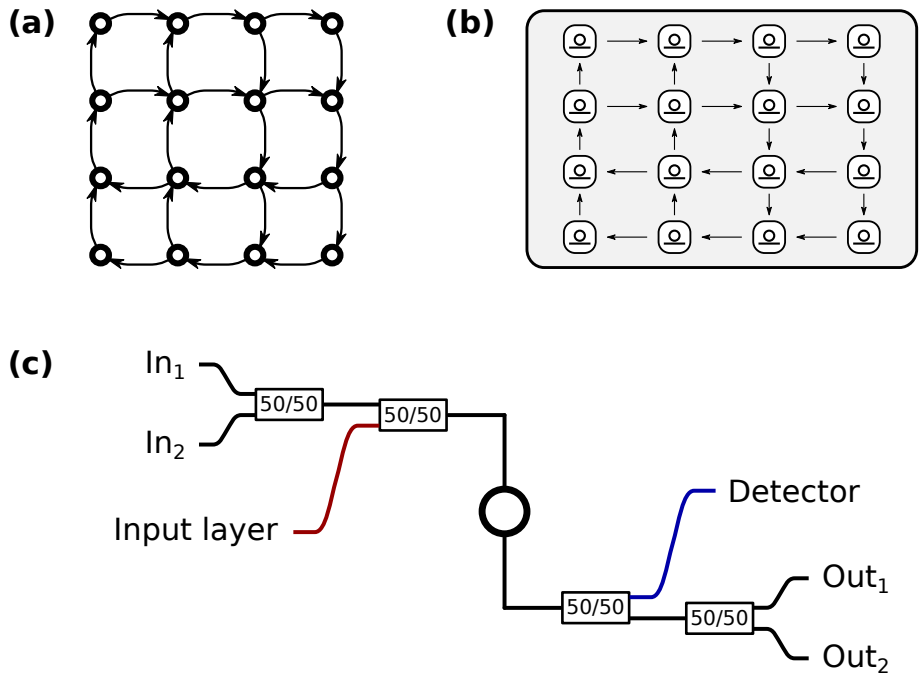

Figure 4.2: (a) Typical scheme of a 4×4 *SWIRL* network. (b) Scheme of the 4×4 *SWIRL* network with nonlinear ring resonators as neuron-like computational units. (c) Illustration of a fully connected node in the *SWIRL* topology, including the 3 dB splitters and combiners. The input to a node is a sum of the signals coming from two previous neighbouring nodes at ports In_1 and In_2 and the input signal. Similarly, the signal at the output of a node is split between the signal going to the readout layer (detector, and the signals going to the two next neighbouring nodes at ports Out_1 and Out_2.

allows sufficient mixing of the input signals while satisfying to the planarity constraint of the integrated implementation and minimizing the power losses in the structure. In this specific work, we implement the *SWIRL* topology on a silicon photonic chip, and we use nonlinear microring as neuron-like computational units, as represented in Fig. 4.2(b). This nonlinear integrated component is commonly used, and has been extensively studied in Sec. 2.4 and in [6, 8].

The schematic illustration of Fig. 4.2(c) presents a complete overview of a fully connected node in the *SWIRL* network. In the *SWIRL* topology, a 2-port

node receives a signal that is a sum of the signals coming from two previous neighbouring nodes and the input signal. Similarly, the signal at the output of this 2-port node is split between the signal going to the readout layer, and the signals going to the two next neighbouring nodes. The synapse-like connections between neighbouring nodes are ensured by long waveguides, which introduce a non-negligible inter-delay due to the finite-time propagation of optical signals.

In the framework of reservoir computing, given in Sec. 3.2.2 by Eq. (3.2), the inner structure of the reservoir is described by the interconnection matrix $\mathbf{W}_{res}$. The coefficient of this matrix represent the connections between the nodes of the reservoir, and takes into account splitting ratios, the losses induced by the synapse-like connections, and random phase shifts uniformly distributed on $[-\pi, \pi]$ to acknowledge for random difference in the manufacture of the interconnection waveguides.

4.1.3 Readout layer and output of the reservoir

We measure the state of the reservoir node as the output power coming out of a node, as represented in Fig. 4.2(c). This measure is performed using a photodetector, whose model is depicted in Sec. 2.3.2 [20]. In the mathematical framework of reservoir computing presented in (3.3) of Sec. 3.2.2, we report the output power at each node in the readout vector $\mathbf{x}_{readout}$, and we construct the output $\mathbf{y}_{out}$ as a linear combination of the readout vector coordinates. The coefficients of this linear combination are reported in the readout matrix $\mathbf{W}_{out}$, and are called the output weights of the reservoir. They are determined during the training procedure using a ridge regression.

4.2 Numerical simulation methods

We examine in this section the numerical simulation methods used to measure the performance of our 16-node reservoir computer. We obtain the reservoir states through the simulation of the 4×4 *SWIRL* reservoir depicted in Fig. 4.2(c). For the simulations, we modify the library PhotRC, developed by our collaborators at the Ghent University, to allow the use of a new type of nodes. Hence, we

first present this library as well as the changes made to match our needs, and we present in a second paragraph the respective training and testing procedure for the different tasks.

4.2.1 The PhotRC library

To obtain the reservoir states, we run numerical simulations of the 4×4 *SWIRL* network illustrated in Fig. 4.2(b). We use a Python library named PhotRC to perform our simulations. This library was developed by our collaborators at the Ghent University as an extension of the Caphe library [21] for the simulation of complex networks. More precisely, this library creates all the objects necessary to create and simulate a *SWIRL* network when given the number and the type of the nodes. In particular, the library generates the interconnection matrix $\mathbf{W}_{res}$, accounting for the splitting ratios of a fully connected node (see Fig. 4.2(c)), the losses induced by the waveguides used as synapses, and random phase shifts due to light propagation. The library also generates the input matrix $\mathbf{W}_{in}$ resulting from the definition of the input layer to the reservoir. In our case, we inject the same power modulation in all the active nodes with random phase shifts, but the library also allows for the generation of more complex input matrix with different input powers at the nodes, injection only on some of the nodes, the addition of a bias signal, etc.

In this particular work, we analyse the performance of a 4×4 network in which the nodes are nonlinear microring resonator. It was then necessary to create a new type of node, being the nonlinear microring resonator described in Sec. 2.4. As a reminder, the microring is a 2-port node described by its input/output relationship, and three state variables: the optical mode a, the temperature variations ΔT, and the free carrier concentration N. Finally, once created with the PhotRC library, the reservoir states of this reservoir computer are obtained by the simulation of the system using the numerical Euler algorithm, and the post processing (training and testing of the reservoir) can be performed as described in Sec. 4.2.2, according to the desired task.

4.2.2 Numerical simulations and post processing

This section is devoted to the description of the numerical simulations, and the post processing operations. In particular, the post processing procedures include the training and the testing of the reservoir, and can vary according to the desired task. We will describe the numerical simulation strategy and the post processing operation for the three benchmark tasks we use in this chapter, namely the evaluation of the memory capacity, the delayed-XOR task, and the N-bit pattern recognition task. These tasks have been presented in Sec. 3.3.

In the case of the evaluation of the memory capacity, the input signal fed into the reservoir consists of 5,000 randomly chosen samples drawn from a uniform distribution in the interval $[-1,1]$, scaled down and shifted to match the optimal operating point presented in Sec. 4.3.2. We perform our simulations with a sampling rate of 200 Gb/s, and the same input stream is injected simultaneously on all nodes of the reservoir at 10 Gb/s, though the input matrix $\mathbf{W}_{in}$, which is a 16×16 diagonal matrix with random elements sampled from a uniform distribution over the interval $[-\pi,\pi]$ accounting for random phase shifts at the input layer. The reservoir states are obtained through the simulation of the 4×4 *SWIRL* network using the PhotRC library. For the readout layer, we use the discrete states $x_{detector}$ of all 16-nodes to perform the training and the testing of the reservoir. The training of the linear readouts is performed using a simple linear regression on 1,400 bits using the scikit-learn library [22], and the testing is done on the 3,600 remaining bits.

In the case of the 1-bit delayed-XOR task, the bit stream fed into the reservoir consists of 20,000 randomly chosen bits. We use in our simulations a sampling rate of 160 Gb/s. Multiple-input simulations are performed with the same bit stream injected simultaneously with the same input power weights on all 16-nodes, also through the input matrix $\mathbf{W}_{in}$. For the readout layer, we also use the discrete states $x_{detector}$ of all 16-nodes to perform the training and the testing of the reservoir. The training of the linear readouts is performed using regularized ridge regression on 16,000 bits, using the scikit-learn library [22]. The testing is done on the 4,000 remaining bits, for a regularization parameter chosen using the best case from a five-fold cross-validation. We report the error

rates on the test data, hence the minimum measurable error rate is 2.5×10^{-4}.

And finally, in the case of the 3-bit and 4-bit pattern recognition task, the simulation strategy and the post processing procedure are the same than what has been presented for the delayed XOR task.

4.3 Operating point of the reservoir

J. Dambre highlighted in 2012 [23] that any dynamical system having a fading memory [24] and linearly independent internal variables has in principle the capacity to process information, in particular in the framework of reservoir computing. A proof of this assertion is provided by the large number of systems that have been investigated as implementation of reservoir computing. To name a few, the very simple bucket of water [25], a physical soft body [26], the brain of a living cat [27], memristor-based systems [28, 29], and obviously opto-electronic [30–33] and photonic [1, 2, 34–41] systems. This profusion of very different systems used for the same application demonstrates the universality of the concept of reservoir computing.

From this observation, it was said in [42] that one the most challenging task of the study of a reservoir computer is to find the optimal operating point for the resolution of the desired task. The identification of the best operating parameters can be done in a very exhaustive way, thus requiring a large amount of time and resources to scan all the parameters of the phase space. However, it is also possible to intuit a close range of parameters in which the reservoir computer optimal operating point will be from the dynamical properties of the system.

In our work, we suggest an operating point for the reservoir computer thanks to the knowledge of the intrinsic properties of the building block of the reservoir, namely the nonlinear ring resonator.In particular, we both take into account the static and dynamical properties of this nonlinear integrated component to intuit a range of space parameters (injection power and optical detuning) for an optimal use of the network for reservoir computing applications.

4.3.1 Edge of instabilities

The fading memory [24] is a very important property of a system for reservoir computing applications [23]. In order to maximise the fading memory in a reservoir, a commonly used technique is to operate the system at the edge of instability [43]. Typically, the system should operate in a stable regime, but close to instabilities in order for the transient dynamics to be as complex as possible.

For some simple dynamical systems, it is possible to mathematically map the regions of the phase space where the system is stable through stability analysis procedures and continuation techniques. From this information, it is very simple to suggest an range of parameters at the edge of instability. However, our system is made of 16 nonlinear microring resonators interconnected in a complex network through splitters, combiners, and long waveguides inducing time delays. As presented in Sec. 2.4.2, a nonlinear ring resonator is described in the coupled-mode theory framework by one static equation, and three coupled complex ordinary differential equations. Hence, such a 4×4 network would be described by four static delayed equations and $16 \times 3 = 48$ complex delayed differential equations. The theoretical stability analysis of such a system cannot be performed, hence we use a pragmatic approach to find the optimal operating point of our reservoir. This approach is described in the following section.

4.3.2 Pragmatic approach to find the best reservoir operating point

We describe in this section our approach for a pragmatic search of the injection parameters to operate our network as a reservoir computer. As explained previously, we cannot search for the regions of stability of the network theoretically due to the complexity of the system model. However, the dynamical and static properties of the neuron-like computational unit are well known and can be used to suggest an operating point.

Indeed, we can make the simple assumption that the reservoir will be in

a steady state if a single node of the reservoir is on a fixed point. This is a reasonable assumption because of the weak linear optical coupling due to the losses induced by the waveguides (3dB/cm), the splitters (3dB for each splitter), and the combiners (3dB for each combiner). This study is performed below.

4.3.2.1 Stability of a ring resonator

We study in this section the stability of a nonlinear ring resonator. We first simulate using the Caphe software environment [21] a single, uncoupled, nonlinear microring resonator subjected to steps of optical power between $P_{in,0}$ = 0 mW and several maximum values $P_{in,1}$, for various values of the optical detuning $\delta\lambda$ as defined in Sec. 2.4.2. The simulations are performed as follow : we integrate the CMT-model of the nonlinear microring resonator (see Sec. 2.4.2 and Ref. [6, 8] for the equations and the parameters values) over 2.5 μs with a power step from $P_{in,0} = 0$ mW to the value of $P_{in,1}$ at $t = 100$ ns. We use an Euler integration method with a 1.0 ps integration time step, and a 10.0 ps sampling time. This nonlinear element can exhibit mostly two different dynamical behaviours: a stable equilibrium and a self-pulsating dynamics. We illustrate these two dynamical behaviours in Fig. 4.3 respectively with injection parameters $P_{in,1} = 0.5$mW and $\delta\lambda = 0$ pm for Fig. 4.3(a) and $P_{in,1} = 1.0$mW and $\delta\lambda = 0$ pm for Fig. 4.3(b).

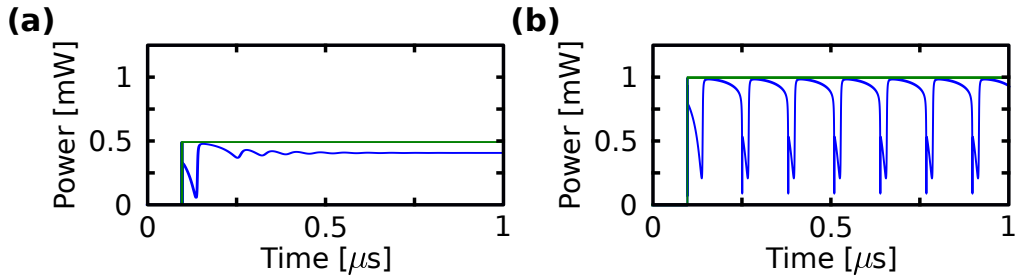

Figure 4.3: Illustration of the dynamical behaviours of a single, uncoupled, nonlinear microring resonator. (a) Stable equilibrium and (b) self pulsating dynamics. The injection parameters are respectively (a) $P_{in,1} = 0.5$mW and $\delta\lambda = 0$ pm, and (b) $P_{in,1} = 1.0$mW and $\delta\lambda = 0$ pm.

Then, from the time series obtained through these simulations, we extract the consecutive extrema for each value of the maximum input power, after deleting the transients. We plot the extrema for each value of the maximum input power $P_{in,1}$ at different values of the optical detuning $\delta\lambda$, and obtain the bifurcation diagrams shown in Figs. 4.4(a)-(c), for respectively (a) $\delta\lambda = -50$ pm, (b) $\delta\lambda = 0$ pm, and (c) $\delta\lambda = 50$ pm. For an optical detuning $\delta\lambda = -50$ pm, the output power is always a fixed point for $P_{in,1} < 2.0$ mW. For an optical detuning $\delta\lambda = 0$ pm, the system is in a stable state for $P_{in,1} < 0.52$ mW, and is self-pulsating for $P_{in,1} > 0.54$ mW. Finally, we see in Fig. 4.4(c) that, for an optical detuning $\delta\lambda = 50$ pm, the output power is stationary for $P_{in,1} < 0.38$ mW, and self-pulsating for $P_{in,1} > 0.40$ mW.

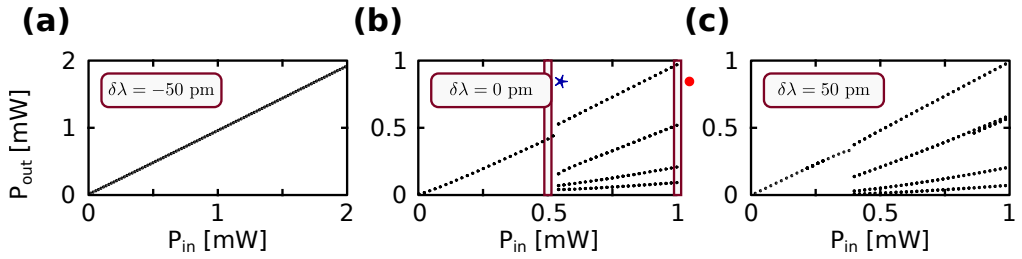

Figure 4.4: Bifurcation diagrams of a single, uncoupled, nonlinear microring resonator for respectively (a) $\delta\lambda = 0$ pm, (b) $\delta\lambda = -50$ pm, and (c) $\delta\lambda = 50$ pm. The bifurcation parameter is the high value of the power step $P_{in,1}$. Note that the we have highlighted two particular injection parameters in (b), corresponding respectively to the times series obtained in Fig. 4.3(a) (blue star), and Fig. 4.3(b) (red dot).

From these bifurcation diagrams, we can identify an operating point for the reservoir in terms of power amplitude modulation for each value of the optical detuning. For $\delta\lambda = 0$ pm, we choose $P_{in,1} = 0.5$ mW (close to the self-pulsation bifurcation point). For $\delta\lambda = 50$ pm, we choose $P_{in,1} = 0.3$ mW. And finally, for $\delta\lambda = -50$ pm, a single nonlinear ring resonator does not exhibit unstable dynamics, hence this value of optical detuning does not allow the ring resonator to be close to instabilities.

Finally, in order to have a complete overview of the stability of a single, uncoupled, nonlinear ring resonator, we present in Fig. 4.5 a theoretically obtained stability map of a ring resonator. This map was originally presented in a normalized parameter plane in [6], and was recomputed and reformatted with continuation techniques using Auto [44] with respect to our parameters of interest (see Sec. 2.4.3.1). This map shows the ring's dynamical behavior in the optical detuning/injected power plane, and for any given set of injection parameters.

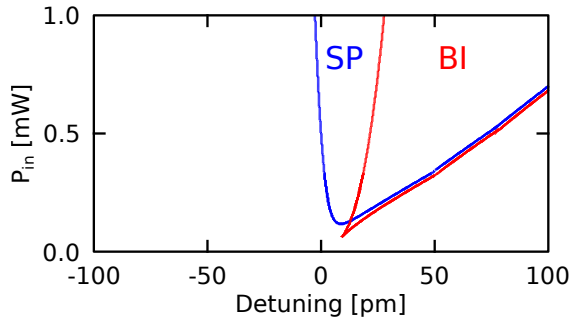

Figure 4.5: Theoretically obtained stability map of a ring resonator in the $(\delta\lambda, P_{in})$ plane, adapted from [6] with continuation techniques. Each region is delimited by bifurcation lines: respectively two saddle-node and a supercritical Hopf bifurcation for the bistable (BI) and self-pulsing (SP) region.

We find three different regions associated respectively to stable fixed points, self-pulsation, and bistability. We call bistable region a region where two different states can be reached depending of the initial conditions. Each region is delimited by bifurcation lines: respectively two saddle-node and a supercritical Hopf bifurcation for the bistable and self-pulsing region. With Fig. 4.5, it is possible to extract the information of Figs. 4.4 for any optical detuning. This map gives us a good indication to set the optimal operating parameters for the reservoir.

4.3.2.2 Filtering properties of a ring resonator

We have analysed in the previous section the dynamical properties of a ring resonator, in order to find the best injection parameters to use our system as a reservoir computer. However, we also need to take into account the static properties of a ring resonator. Indeed, this integrated component is mostly used as an stop-band filter, due to its typical transmission curve. Hence, if we inject the signal with a frequency close to the resonance frequency of the ring resonators, we could filter the signal, thus reducing the interactions between the nodes. We give in Fig. 4.6 the typical transmission curve of a single, uncoupled, nonlinear ring resonator obtained through the simulation of the building block of our reservoir.

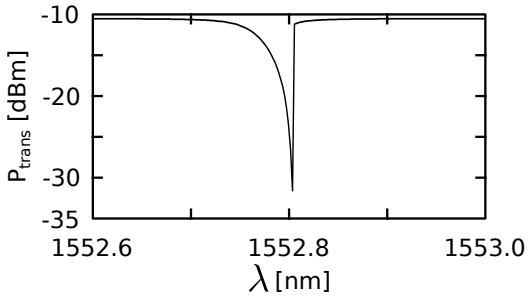

Figure 4.6: Numerically obtained transmission curve of a nonlinear microring resonator with a resonance at $\lambda_r = 1552.770$. We inject a constant input power $P_{in} = 1.0$ mW and record the output value after deleting the transients. This curve shows the stop-band filtering properties of this integrated element.

4.3.2.3 Operating point of the reservoir

We choose in this section the injection parameters to operate our 4×4 *SWIRL* network as a reservoir computer. Our pragmatic approach for the choice of the operating parameters was to study the static and dynamical properties of a ring resonator, and to choose the injection parameters so that the ring resonator is stable, but at the edge of stability. Then, due to the weak linear optical coupling due to the losses induced by the waveguides and splitters/combiners, we make

the reasonable hypothesis that the whole network will remain in steady state if the dynamics of a single node is steady. We choose the injection parameters to be $\delta\lambda = 50$ pm and $P_{in,1} = 0.3$ mW since it allow us to be relatively far from the resonance frequency of the ring (thus the signal is not filtered by the rings), close to the instability bifurcation, and at a relatively limited optical injection power.

We verify our hypothesis that the *SWIRL* network is still stable by injecting a power step from $P_{in,0} = 0.0$ mW to $P_{in,1} = 0.3$ mW on all nodes, with an optical detuning with regards to the resonance frequency of the ring resonators $\delta\lambda = 50$ pm. We plot the output of one node recorded by a photodetector in Fig. 4.7. We see in this figure that indeed, the network is stable for these injection parameters.

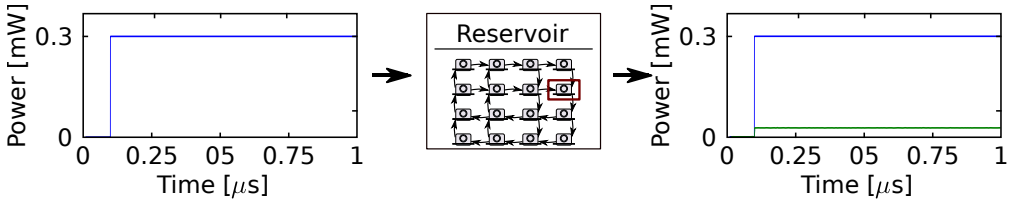

Figure 4.7: Output of one node (highlighted in the network) recorded by a photodetector when we inject a power step from $P_{in,0} = 0.0$ mW to $P_{in,1} = 0.3$ mW on all nodes, with an optical detuning with regards to the resonance frequency of the ring resonators $\delta\lambda = 50$ pm.

4.4 Performance of the reservoir

We analyse now the performance of our reservoir computer. We first evaluate the memory capacity of the system, then test the performance on the delayed XOR task and the 3-bit and 4-bit pattern recognition task. Finally, we study the parametric influence of injection parameters (bit rate, injection power, optical detuning) in order to study the correlation between the performance of the reservoir with the intrinsic properties of its building block. We also study the

robustness of the performance with regards to fabrication uncertainties by introducing a resonance frequency mismatch between the nodes.

4.4.1 Memory capacity of the reservoir

We evaluate in this section the memory capacity of the reservoir computer. We measure the memory capacity using the simulation methods described in Sec. 4.2 for the 4×4 *SWIRL* reservoir with nonlinear microring resonators as nodes. In this work, we investigate the optimal design of the reservoir in terms of interconnection lengths, for a fixed data rate. Hence, we investigate the evolution of the memory capacity when changing the node interdelay at 10 Gb/s. The node interdelay t_{delay} represents the time the light needs to travel in the waveguide from one node to its closest neighbour, and can be calculated from intrinsic properties of the system by Eq. (4.1):

$$t_{delay} = \frac{L \times n_{Si}}{c}, \tag{4.1}$$

where L is the length of the interconnection waveguide, $n_{Si} = 3.476$ is the refractive index of the bulk silicon, and c is the speed of light.

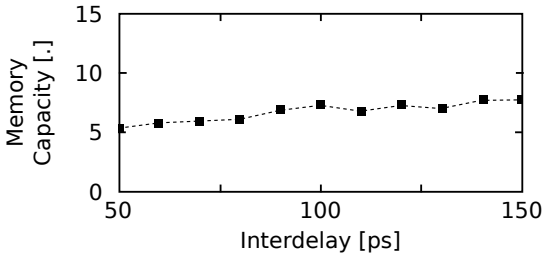

Figure 4.8: memory capacity as a function of the node interdelay for the 4×4 *SWIRL* reservoir made of nonlinear ring resonators at 10 Gb/s. The best measured memory capacity is 7.3.

We present the results of our numerical simulations on the evaluation of the memory capacity in Fig. 4.8. The best measured memory capacity is

7.3, so we can easily conclude that at a given time, the current signal in the reservoir incorporate information from the six or seven past inputs in the system. With this value of memory capacity, we can expect to be able to resolve pattern recognition up to 6-bit pattern recognition at maximum, however this value is not adequate for more complex and memory demanding tasks as NARMA10 [45] or chaotic time series forecasting like the Santa Fe task [46].

4.4.2 Delayed-XOR task

We investigate in this section the performance of our reservoir computer on the 1-bit delayed XOR task, and we compare the results to the performance of the network from [3] with linear nodes. We study the influence of various injection parameters, namely the processing rate, the injection power, and the optical detuning. Finally, from the results of our simulations, we study the correlation between the performance and the intrinsic properties of the microring resonator used as node.

4.4.2.1 Influence of the bit rate

The reservoir performance is plotted in Fig. 4.9. We focus on the influence of the data rate, and give the performance as a function of the node interdelay at 10 Gb/s (black dots), 15 Gb/s (red squares), 20 Gb/s (blue triangles), and 30 Gb/s (green diamonds), respectively. In order to compare with previous work, we give in Fig. 4.9(a) the performance of the fully passive reservoir of [2, 3], and in Fig. 4.9(b) the performance of the reservoir using nonlinear microring resonators as nodes (called MR-reservoir for clarity purposes). In the case of the passive reservoir (Fig. 4.9(a)), the bit stream is fed on all nodes through a power modulation from $P_{in,0} = 0.1$ mW and $P_{in,1} = 0.2$ mW. In the case of the MR-reservoir (Fig. 4.9(b)), we fix the optical detuning at $\delta\lambda = 50.0$ pm, and we modulate the injected power between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW, according to the optimal injection parameter conditions determined in Sec. 4.3. In this reservoir, all the ring resonators have the same resonance frequency.

The results presented in this figure suggest that the reservoir with nonlinear microring resonators as nodes can perform the typical 1-bit delayed

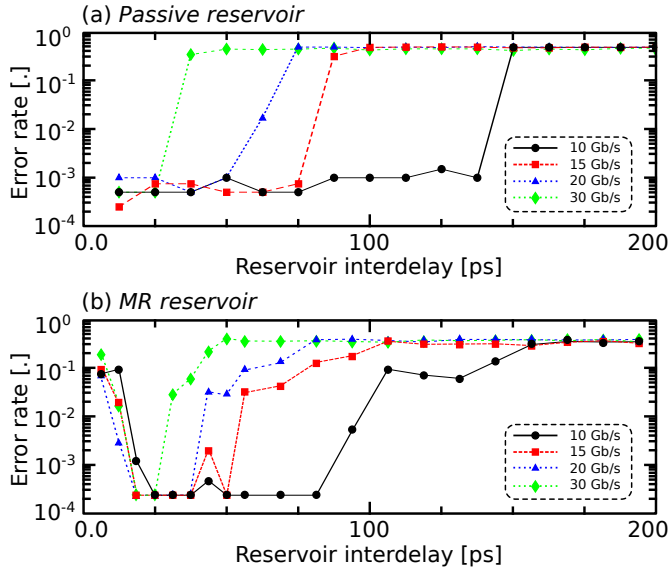

Figure 4.9: Error rate - for the XOR task - as a function of the node interdelay for various bit rates : 10 Gb/s (black dots), 15 Gb/s (red squares), 20 Gb/s (blue triangles), and 30 Gb/s (green diamonds). Comparison between (a) the passive reservoir of [2, 3], and (b) the MR-reservoir. (a) : we modulate the injected power between $P_{in,0} = 0.1$ mW and $P_{in,1} = 0.2$ mW. (b) : the optical detuning is $\delta\lambda = 50.0$ pm, and we modulate the injected power between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW. The minimum acceptable error rate is 2.5×10^{-4}.

XOR task with error rates about 2.5×10^{-4} (lowest achievable value with the number of bit used in testing) for various values of the interdelay at high bit rates. We also see that the range of interdelay values, where the reservoir performs at its best, is slightly greater for lower bit rates. This is similar to the passive reservoir (see Fig. 4.9(a)), but our architecture can achieve lower error rates. We notice also a reduced range of interdelay values for the best performance compared to a passive reservoir. This is most likely due to the internal time scale of the optical mode $\tau_a \approx 20$ ps in the microring resonator model. This time scale is close to the optimal value of interdelay in term of reservoir performance.

We present also in Fig. 4.10(a) normalized time series at four nodes at the

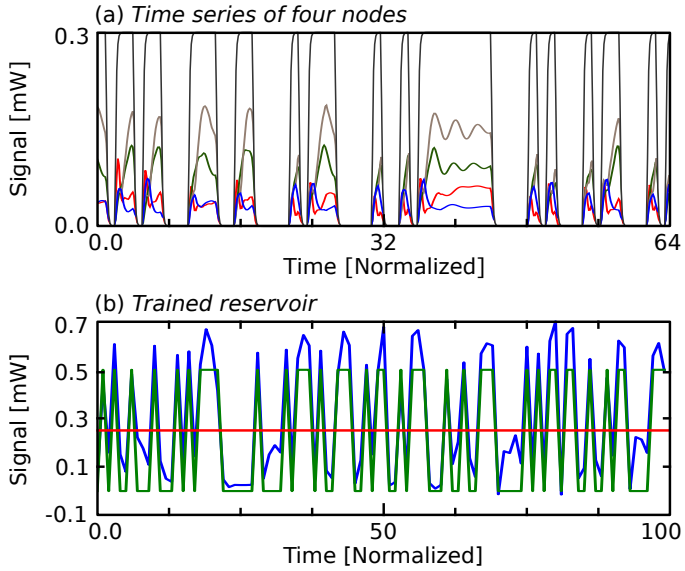

Figure 4.10: (a) Time series at the readout layer of four different nodes of the reservoir when injecting the bit stream on all nodes with an optical detuning $\delta\lambda = 50.0$ pm, a power modulation between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW at 20 Gb/s, and a node interdelay $t_{delay} = 18.75$ ps. (b) Desired output (green curve), trained output of the reservoir (blue curve), and decision threshold (red line) for the same injection parameters as in (a). These parameters correspond to an optimal value of the error rate of Fig. 4.9(b).

readout layer of the reservoir, along with the input power in each node. These time series are obtained for the simulation of the MR-reservoir for an optical detuning $\delta\lambda = 50.0$ pm, a power modulation comprised between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW, and an interdelay $t_{delay} = 18.75$ ps. This injection point corresponds to an optimal value of the error rate in Fig. 4.9(b). Finally, in Fig. 4.10(b), we show the output of the trained reservoir for the same injection parameters as in Fig. 4.10(a). The green curve is the desired output, the blue curve is the output of the trained reservoir, and the red line is the decision threshold. For both Fig. 4.10(a) and Fig. 4.10(b), the time is normalized with respect to the duration of one bit.

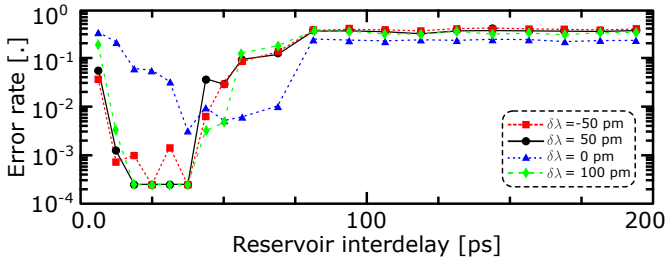

Figure 4.11: Error rate - for the XOR task - as a function of the reservoir interdelay for various values of the optical detuning at 20 Gb/s. The power modulation is chosen so that a microring alone is in a stationary state, but close to the instabilities, with $P_{in,0} = 0.0$ mW and respectively $\delta\lambda = -50$ pm and $P_{in,1} = 0.5$ mW (red squares), $\delta\lambda = 0.0$ pm and $P_{in,1} = 0.5$ mW (blue triangles), $\delta\lambda = 50$ pm and $P_{in,1} = 0.3$ mW (black dots), and $\delta\lambda = 100$ pm and $P_{in,1} = 0.5$ mW (green diamonds). The minimum acceptable error rate is 2.5×10^{-4}.

4.4.2.2 Influence of the optical detuning

In this section, we focus on the influence of a new degree of freedom introduced by the use of nonlinear ring resonators, namely the optical detuning between the injected light and the resonance frequency of the ring resonators. We investigate the changes in the levels of performance that can be attained with such a reservoir when introducing this nonlinear resonant element as node, and find the correlation between the intrinsic properties of the building block of the reservoir and the performance. In Fig. 4.11 is plotted the performance of the reservoir as a function of the node interdelay for four different values of the optical detuning : $\delta\lambda = -50$ pm (red squares), $\delta\lambda = 0.0$ pm (blue triangles), $\delta\lambda = 50$ pm (black dots), and $\delta\lambda = 100$ pm (green diamonds). The Return-to-Zero power modulation is chosen so that a microring alone is in a stationary state, but close to a bifurcation point. Referring to the stability map of a ring resonator given in Fig. 4.5, the high value of the power modulation is $P_{in,1} = 0.3$ mW for $\delta\lambda = 50$ pm, and $P_{in,1} = 0.5$ mW for $\delta\lambda = 0$ pm, $\delta\lambda = 100$ pm, and $\delta\lambda = -50$ pm.

Figure 4.11 shows that the reservoir performs better when the value of the optical frequency of the injected light is shifted with respect to the resonance

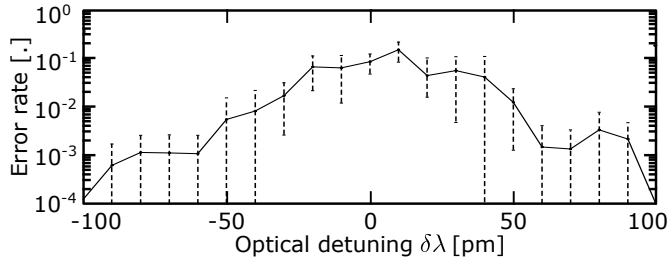

Figure 4.12: Error rate - for the XOR task - as a function of the optical detuning for a power modulation between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.5$ mW, a node interdelay of 18.75 ps, and a bit rate 20 Gb/s. Error bars are given for seven series of simulations. The minimum acceptable error rate is 2.5×10^{-4}.

frequency of the nonlinear node (typically in our study $\delta\lambda \in \{-50, 50, 100\}$ pm), than when the light is injected at the resonance frequency of the microring resonator (*i.e.* $\delta\lambda = 0$ pm). Note that the performance is similar for those three different values of the optical detuning $\delta\lambda \in \{-50, 50, 100\}$ pm. Intuitively, this can be understood by looking at the filtering properties of a microring resonator presented in Sec. 4.3.2.2: The nonlinear rings absorb more optical power if the frequency of the injected light is close to their resonance. As a result, the wave-mixing between the nodes in the network is reduced, thus impeding the reservoir computer performance.

In order to corroborate the results of Fig. 4.11, we plot in Fig. 4.12 the performance of the reservoir as a function of the optical detuning. More specifically, we have set the interdelay $t_{delay} = 18.75$ ps, which corresponds to the best choice in terms of interconnection length, as it ensures relatively small connection waveguides. And, we inject a power modulation between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.5$ mW. We realize seven experiments for each optical detuning, and averaged the results that are given with the error bars.

Figure 4.12 unveils a better level of performance when the frequency of the injected light is far from the frequency resonance of the nonlinear microring resonators. We see also that the performance are better for negative values of the frequency detuning. This can be understood by looking at the stability map

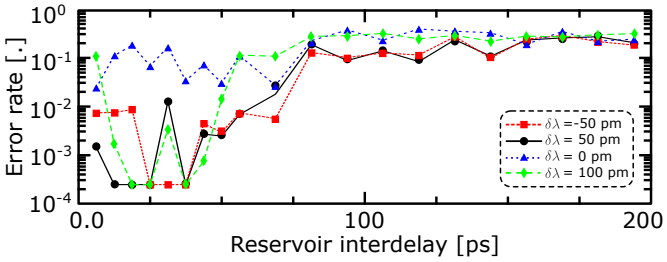

Figure 4.13: Error rate - for the XOR task - as a function of the reservoir interdelay for various values of the optical detuning at 20 Gb/s. The power modulation is chosen so that a microring alone is in a stationary state, but close to the instabilities, with $P_{in,0} = 0.0$ mW and respectively $\delta\lambda = -50$ pm and $P_{in,1} = 0.5$ mW (red squares), $\delta\lambda = 0.0$ pm and $P_{in,1} = 0.5$ mW (blue triangles), $\delta\lambda = 50$ pm and $P_{in,1} = 0.3$ mW (black dots), and $\delta\lambda = 100$ pm and $P_{in,1} = 0.5$ mW (green diamonds). in addition, we have introduced a frequency mismatch from one ring to another, through a 10 pm standard deviation centred on respectively $\delta\lambda \in \{-50, 0, 50, 100\}$ pm. The minimum acceptable error rate is 2.5×10^{-4}.

of Fig. 4.5. For positive values of the optical detuning from 0 pm and 75 pm, a nonlinear microring resonator is self-pulsing for an injected power of 0.5 mW, thus meaning the reservoir is not on a steady state and consequently reducing its performance.

4.4.2.3 Inhomogeneities of the nonlinear nodes

In the precedent section, we have considered that all the nodes are strictly the same, meaning that all 16 nonlinear microring resonators have the same resonance frequency. For more realistic simulations, we give in Fig. 4.13 - and for the same input conditions as in Fig 4.11 - the performance of our reservoir when the resonance frequencies of the ring resonators are different. The resonance frequencies of the 16 microring resonators follow a Gaussian distribution centred on respectively $\delta\lambda \in \{-50, 0, 50, 100\}$ pm, with a 10 pm standard deviation, that is a rather pessimistic value with regard to the current technology.

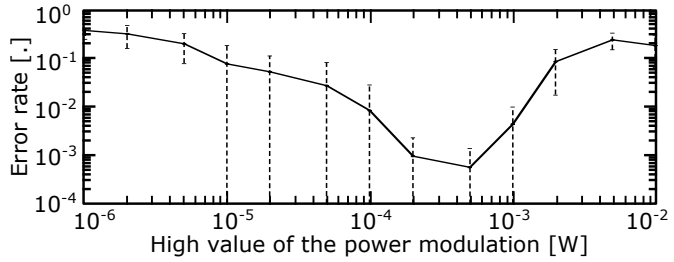

Figure 4.14: Error rate - for the XOR task - as a function of the high value of the power modulation for an optical detuning $\delta\lambda = 50.0$ pm, at 20 Gb/s, and with a node interdelay $t_{delay} = 18.75$ ps. The low value of the Return-to-Zero power modulation is set to $P_{in,0} = 0.0$ mW. We give error bars for an average on seven series of simulations. The minimum acceptable error rate is 2.5×10^{-4}.

Figure 4.13 shows a good robustness of the reservoir with regards to heterogeneities in the frequency resonance between the nodes, providing that the detuning of the injected light is larger than the standard deviation of the heterogeneities in the oscillator resonance frequencies. This information is of great importance for manufacturing purposes.

4.4.2.4 Influence of the injection power

We investigate here the influence of the injection power on the level of performance that can be attained by our reservoir computer, as a way to focus on the energy consumption. In order to quantify the influence of this injection parameter, we evaluate the bit error rate for Return-to-Zero modulations of the injected power at each node for an optical detuning $\delta\lambda = 50.0$ pm, at 20 Gb/s, and with a node interdelay $t_{delay} = 18.75$ ps. We plot in Fig. 4.14 the error rate of the reservoir as a function of the high value of the power modulation $P_{in,1}$. We give the average and the error bars for seven series of simulations. Note that we have also introduced mismatches in the resonance frequency between the rings, similarly to previous studies.

Figure 4.14 shows that values of the injected power lower than 0.1 mW result in a degradation of the performance, due to a reduction of total power

on each node and wave-mixing between the nodes through losses in the other integrated elements (waveguides, splitters, combiners). We also see that the performance of the reservoir are low for high values of the modulation ($P_{in,1} > 1$ mW). This is due the fact that each node is self-pulsing for these injection parameters, as previously shown in the stability map of a single, uncoupled, nonlinear ring resonator in Fig. 4.5. Finally, the optimal operating condition at this particular optical detuning is when the high value of the power modulation leads a single microring resonator to be in a steady state, but close to instabilities. However, we also find a very good performance obtained for the high value of the power modulation $P_{in,1} = 0.5$ mW, where a single microring resonator alone is self-pulsing.

In the simulations performed in the previous sections, we have set the optical detuning at $\delta\lambda = 50.0$ pm, with a power modulation comprised between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW, which is in the interval of best performance at this particular detuning. From this information, we can evaluate the total power budget of our architecture for an optimal use, and find it to be very low. Indeed the chip is only powered by the optical power using the same bit stream input on each node, thus the averaged power needed for the reservoir to perform is $N_{nodes} \times 0.5 \times (P_{in_1} - P_{in_0}) = 2.4$ mW (with $N_{nodes} = 16$, the number of nodes in the reservoir). Moreover, unlike the purely passive reservoir of [2, 3], where a bias power was necessary to perform optimally, the MR-reservoir has the best performance when there is no power bias, thus reducing the mean power consumption.

4.4.2.5 Mapping of the performance in the optical detuning, injection power plane

We present in Fig. 4.15 a mapping of the performance of the reservoir for various optical detuning and high values of the power modulation. For all the simulations, the design of the reservoir if fixed with an interdelay of 18.75 ps. In order to study the robustness of the reservoir, and as in Sec. 4.4.2.3, we have introduced heterogeneities in the resonance frequency of the ring resonators. Typically, the resonance frequencies of the 16 microring resonators follow a

Gaussian distribution centred at λ_r, with a 10 pm standard deviation, which is a rather pessimistic value with respect to the current technology. For all the simulations, the same input stream of 15,000 bits is fed at 20 Gb/s on all 16-nodes of the reservoir. We realise the training on 12,000 bits and the testing on the 3,000 remaining bits.

We see in this map that the reservoir can perform at acceptable levels of performance for various injection parameters, and for low power consumption. Indeed the average power consumption of the reservoir is given by Eq. (4.2) :

$$< P_{total} >= N_{nodes} < P_{in} >= \frac{P_{in,0} + P_{in,1}}{2} N_{nodes}, \tag{4.2}$$

where N_{nodes} is the number of nodes, and $P_{in,1}$ the high value of the power modulation. Hence, we can find a set of injection parameters where the reservoir computer performs at best with very low power consumption. For instance for a power modulation between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.1$ mW, at $\delta\lambda = 50$ pm, the BER is lower than 10^{-3} and the total averaged power is 0.8 mW.

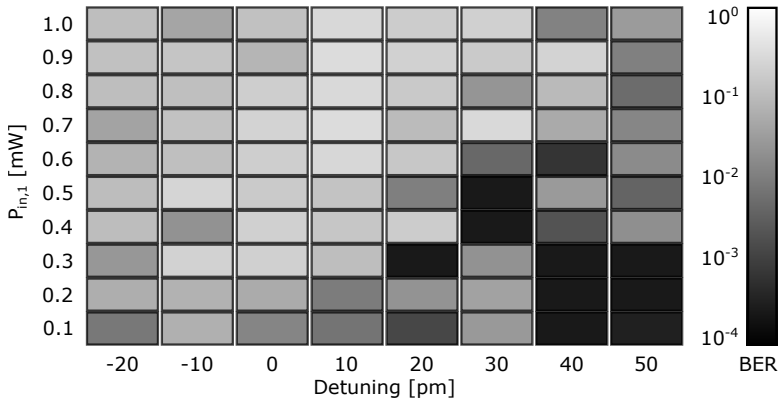

Figure 4.15: Mapping - in the optical detuning/power modulation plane - of the performance of the reservoir computer on the delayed XOR task. The interdelay is 18.75 ps. The minimum error rate is 2.5×10^{-4}.

4.4.3 Header recognition

Finally, we analyse the performance of our reservoir computer on a typical task with telecommunication applications, namely the pattern recognition task. We have extensively studied in the precedent section the best set of parameters for the injection of a binary signal on the 16-node reservoir. More specifically, for a node interdelay is $t_{delay} = 18.75$ ps, the best injection parameters at 20 Gb/s are an optical detuning $\delta\lambda = 50.0$ pm, and a power modulation comprised between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW. These parameters give the best measurable performance on the XOR task, while ensuring the lowest power

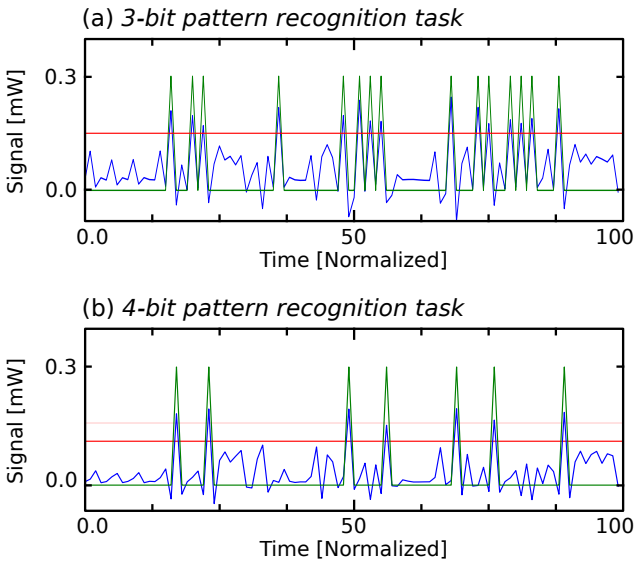

Figure 4.16: Desired output (green curve), trained output of the reservoir (blue curve), and decision threshold (red line) for respectively (a) the 3-bit pattern recognition task and (b) the 4-bit pattern recognition task at 20 Gb/s. The pattern we intend to recognise is respectively (a) [1.0.1], and (b) [1.0.1.1]. The binary stream is injected on all nodes according to the following parameters: $\delta\lambda = 50.0$ pm, and $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW, with a node interdelay $t_{delay} = 18.75$ ps. These parameters correspond to the best measurable performance 2.5×10^{-4}. Note that in (b), we have lowered the threshold value to attain this level of performance.

budget.

Hence, it is reasonable to inject our binary signal for the pattern recognition task according to the same parameters. We attain with these injection parameters the same bit error rate of 2.5×10^{-4} on the 3-bit pattern recognition task, corresponding to the lowest measurable error with this number of testing bits. It is also possible to attain this level of performance for the 4-bit pattern recognition task with these injection parameters, but it is necessary to lower the value of the threshold discriminating a **1** from a **0**. We plot in Fig. 4.16 the output of the trained reservoir for the best injection parameters, respectively for (a) the 3-bit pattern ([1.0.1]) recognition task, and (b) the 4-bit pattern ([1.0.1.1]) recognition task. The green curve is the desired output, the blue curve is the output of the trained reservoir, and the red line is the decision threshold. For both Fig. 4.16(a) and Fig. 4.16(b), the time is normalized with respect to the duration of one bit.

4.5 Conclusion

In this chapter, we have suggested a novel integrated reservoir architecture using microring resonators as nonlinear nodes, that can perform at state-of-the-art level of performance [1, 3, 4] on a nonlinear Boolean task and on a memory demanding task for various operating parameter conditions. We also have related the performance of the reservoir computer to the nonlinear properties of the nodes stability with respect to injected power and frequency detuning between the injected light and the resonance of the rings. More specifically, we have studied the influence of the data rate, and shown that the intrinsic presence of three distinct time scales in the model of the nonlinear nodes leads to the need to carefully design the reservoir in terms of the length of the interconnections between the nodes. We have also investigated the influence of two critical operational parameters in the network dynamics : (i) the injected power and (ii) the optical detuning. We have found that a large variety of operating conditions can lead to optimal performance of the reservoir on the typical delayed XOR task, when some important conditions are fulfilled. First,

each node should be in a steady state, close to instabilities. This condition, along with a stability map of a single node, allows us to choose the operating condition of the complete reservoir parameters for optimal performance. We have also found a good robustness when we introduce heterogeneities in the properties of the nonlinear nodes, for example in the resonance frequencies of the different ring resonators.

We have demonstrated that this integrated reservoir can perform very well and with very low power consumption on a typical Boolean task, namely the 1-bit delayed XOR task, and on a memory demanding task with direct telecommunication applications, *i.e.* the pattern recognition task. Considering the Return-to-Zero power modulation between $P_{in,0} = 0.0$ mW and $P_{in,1} = 0.3$ mW with the same bit stream input on each node, the power budget is very good, and could be further improved in future work by reducing the number of injected nodes, for instance by injecting the data only on the four central nodes, as suggested by A. Katumba *et al.* in [47]. Moreover, from an experimental point of view, it is simpler to inject the data on fewer nodes, as it reduces the routing density on the chip.

Contrary to the passive reservoir of [2, 3] in which the non linearity is in the readout (*i.e.* the detector), we have integrated nonlinear elements (the microring resonators) in the recurrence of the network. This work shows that the performance on these particular tasks in terms of error rate and power consumption are very similar with the previous design. This is mainly due to the losses limiting the mixing in both architectures with or without embedded nonlinear elements. A different internal architecture with better loss management would probably enhance our performance in presence of microring resonators. The current results motivate further investigations on the performance of this kind of structure, and in particular we investigate a structure with heterogeneous nodes in Chap. 5, in which some of the nodes are the nonlinear microring resonator from this architecture, some nodes are the linear nodes from [2], and finally, some nodes are integrated semiconductor optical amplifier, as in the work of K. Vandoorne in [1].

References

[1] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[2] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[3] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[4] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[5] H. J. Caulfield, R. A. Soref, and C. S. Vikram, "Universal reconfigurable optical logic with silicon-on-insulator resonant structures," *Photonics and Nanostructures-Fundamentals and Applications*, vol. 5, no. 1, pp. 14–20, 2007.

[6] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[7] S. Chen, L. Zhang, Y. Fei, and T. Cao, "Bistability and self-pulsation phenomena in silicon microring resonators based on nonlinear optical effects," *Optics Express*, vol. 20, no. 7, pp. 7454–7468, 2012.

[8] T. Van Vaerenbergh, "All-optical spiking neurons integrated on a photonic chip," *PhD Thesis*, 2014.

[9] L. Zhang, Y. Fei, Y. Cao, X. Lei, and S. Chen, "Experimental observations of thermo-optical bistability and self-pulsation in silicon microring resonators," *JOSA B*, vol. 31, no. 2, pp. 201–206, 2014.

[10] W. H. Pernice, M. Li, and H. X. Tang, "Time-domain measurement of optical transport in silicon micro-ring resonators," *Optics express*, vol. 18, no. 17, pp. 18 438–18 452, 2010.

[11] I. S. Amiri, R. Ahsan, A. Shahidinejad, J. Ali, and P. P. Yupapin, "Characterisation of bifurcation and chaos in silicon microring resonator," *IET Communications*, vol. 6, no. 16, pp. 2671–2675, 2012.

[12] J. Hryniewicz, P. Absil, B. Little, R. Wilson, and P.-T. Ho, "Higher order filter response in coupled microring resonators," *IEEE Photonics Technology Letters*, vol. 12, no. 3, pp. 320–322, 2000.

[13] H. Kishikawa, T. Kondo, N. Goto, and S. Talabattula, "Optical thresholder consisting of two cascaded mach–zehnder interferometers with nonlinear microring resonators," *Optical Engineering*, vol. 56, no. 8, p. 086101, 2017.

[14] Y. Dumeige, L. Ghisa, and P. Féron, "Integrated all-optical pulse restoration with coupled nonlinear microring resonators," *Optics letters*, vol. 31, no. 14, pp. 2187–2189, 2006.

[15] C. Tanaram, C. Teeka, R. Jomtarak, P. Yupapin, M. Jalil, I. Amiri, and J. Ali, "Ask-to-psk generation based on nonlinear microring resonators coupled to one mzi arm," *Procedia Engineering*, vol. 8, pp. 432–435, 2011.

[16] F. Denis-Le Coarer, M. Sciamanna, A. Katumba, M. Freiberger, J. Dambre, P. Bienstman, and D. Rontani, "All-optical reservoir computing on a photonic chip using silicon-based ring resonators," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 24, no. 6, pp. 1–8, 2018.

[17] F. Denis-le Coarer, D. Rontani, A. Katumba, M. Freiberger, J. Dambre, P. Bienstman, and M. Sciamanna, "Toward neuro-inspired computing using a small network of micro-ring resonators on an integrated photonic chip," vol. 10689, p. 1068908, 2018.

[18] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[19] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[20] F. R. Times, W. Ranges, and W. W. Ranges, "Ultrafast Photodetectors," *Data Sheet*, 2000.

[21] http://www.lucedaphotonics.com/, accessed: 2016.

[22] http://scikit-learn.org/, accessed: 2016.

[23] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[24] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[25] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *European conference on artificial life*. Springer, 2003, pp. 588–597.

[26] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Information processing via physical soft body," *Scientific reports*, vol. 5, p. 10487, 2015.

[27] D. Nikolić, S. Haeusler, W. Singer, and W. Maass, "Temporal dynamics of information content carried by neurons in the primary visual cortex," *Advances in neural information processing systems*, pp. 1041–1048, 2007.

[28] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, "Reservoir computing using dynamic memristors for temporal information processing," *Nature communications*, vol. 8, no. 1, p. 2204, 2017.

[29] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," *Frontiers in neuroscience*, vol. 9, p. 502, 2016.

[30] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, pp. 1–6, 2012.

[31] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing," *Optics express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[32] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics Express*, vol. 21, no. 1, p. 12, 2013.

[33] P. Antonik, F. Duport, M. Hermans, A. Smerieri, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer applied to real-time channel equalization," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2686–2698, 2016.

[34] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

[35] C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," *JOSA B*, vol. 30, no. 11, pp. 3048–3055, 2013.

[36] C. Mesaritakis, A. Kapsalis, and D. Syvridis, "All-optical reservoir computing system based on ingaasp ring resonators for high-speed identification and optical routing in optical networks," *SPIE OPTO*, pp. 937 033–937 033, 2015.

[37] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[38] R. Modeste Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van Der Sande, "Simultaneous computation of two independent tasks using reservoir

computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

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

[40] X. Bao, Q. Zhao, H. Yin, and J. Qin, "Recognition of the optical packet header for two channels utilizing the parallel reservoir computing based on a semiconductor ring laser," *Modern Physics Letters B*, vol. 32, no. 14, p. 1850150, 2018.

[41] J. Vatin, D. Rontani, and M. Sciamanna, "Experimental reservoir computing using vcsel polarization dynamics," *Optics Express*, vol. 27, no. 13, pp. 18 579–18 584, 2019.

[42] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega, "Optimal nonlinear information processing capacity in delay-based reservoir computers," *Scientific reports*, vol. 5, p. 12858, 2015.

[43] T. Yamane, S. Takeda, D. Nakano, G. Tanaka, R. Nakane, S. Nakagawa, and A. Hirose, "Dynamics of reservoir computing at the edge of stability," *International Conference on Neural Information Processing*, pp. 205–212, 2016.

[44] http://indy.cs.concordia.ca/auto/, accessed: 2016.

[45] R. M. Hirsch, J. R. Slack, and R. A. Smith, "Techniques of trend analysis for monthly water quality data," *Water resources research*, vol. 18, no. 1, pp. 107–121, 1982.

[46] A. S. Weigend, *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.

[47] L. Zhang, Y. Fei, T. Cao, Y. Cao, Q. Xu, and S. Chen, "Multibistability and self-pulsation in nonlinear high- q silicon microring resonators considering thermo-optical effect," *Physical Review A*, vol. 87, no. 5, p. 053805, 2013.

CHAPTER 5

HETEROGENEOUS *SWIRL* RESERVOIR COMPUTER ON A SILICON PHOTONIC CHIP

"Evolution is an integration of matter and concomitant dissipation of motion; during which the matter passes from an indefinite, incoherent homogeneity to a definite, coherent heterogeneity."

— H. Spencer, *First Principles*

We have presented in Chapter 4 a 16-node network made of nonlinear microring resonators, and studied the performance of such a system on typical tasks, namely the delayed-XOR task and the pattern recognition task. We have compared our numerical results to a similar *SWIRL* architecture, introduced by K. Vandoorne in [1], and studied extensively with linear components as nodes by our collaborators at the Ghent University [2–4]. The results from Chapter 4 suggest that the losses induced by the interconnecting waveguides and the splitters/combiners are playing a very important role in limiting the wave-mixing between the nodes, and therefore the performance that can be attained with such a system.

A classical way to amplify an optical signal on a silicon photonic chip is to integrate a semiconductor optical amplifier (SOA) [5]. This integrated compo-

nent allows for an amplification of the signal, but is also a nonlinear element that can be used as a nonlinear node in a reservoir computing architecture, as suggested by K. Vandoorne [1]. In this chapter, we suggest a new architecture for on-chip high-speed reservoir computing, relying on the implementation of a 4×4 *SWIRL* network with heterogeneous nodes. More specifically, some of the nodes of our architectures are nonlinear ring resonators as in the architecture presented in Chapter 4 and in [6–8], some nodes are the purely passive components as in the architecture presented in [2, 3], and finally a few carefully chosen nodes are semiconductor optical amplifier, as in the work presented in [1].

We numerically analyse in this chapter the performance of this architecture on the two typical tasks already presented in the previous chapter, namely the delayed-XOR task. and the pattern recognition task, at telecommunication application processing rate, *i.e.* 20 Gb/s and 30 Gb/s. Through the extensive numerical simulation campaign, we show that our hybrid on-chip reservoir computer can operate at state-of-the-art levels of performance on those tasks up to 30 Gb/s when we inject the input data on all nodes (bit error rate below 10^{-3}). We also demonstrate that the reservoir performs at state-of-the-art levels of performance on the 3-bit pattern recognition task when the injection is done only on the four central nodes, thus reducing the total optical power consumption for this telecommunication oriented task to 0.6 mW. This injection strategy on this kind of *SWIRL* network was suggested in the work of A. Katumba in [3], and we confirm using our novel architecture that this way of injecting the data is indeed a good strategy.

This chapter is organized as follows. We give in Sec. 5.1 an in-dept description of the structure of the reservoir computer, and in particular we present the input layer, the inner structure, and the readout layer of the reservoir. Section 5.2 is dedicated to the presentation of our simulation methods and the search for the optimal operating parameters. We investigate the reservoir performance in Sec. 5.3 on both the delayed-XOR task and the 3-bit pattern recognition task at 20 Gb/s and 30 Gb/s, and we focus alternatively on two injection strategies, namely an injection on all nodes, and an injection on the four central nodes. Finally, a small conclusion to this chapter is given in Sec. 5.4.

The results presented in this chapter have been submitted for publication in a peer-review journal, and the article is still in the review procedure.

5.1 Architecture of the reservoir computer

We present in this section the detailed architecture of the reservoir studied in this chapter. We give once again in Fig. 5.1 the general structure of a typical reservoir computer [9, 10], as given in Sec. 3.2.1. A reservoir computer is made of an input layer to the reservoir, a fixed recurrent neural network named the reservoir, and a readout layer. In this section, we give a detailed description of the three layers of our heterogeneous reservoir computer, with a particular focus on the inner structure of the recurrent neural network with heterogeneous nodes.

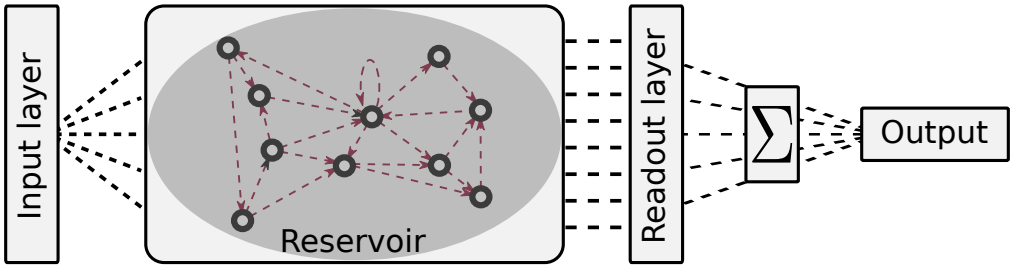

Figure 5.1: General structure of a reservoir computer. The data is send to the neural network from the input layer. The fixed recurrent neural network is called the reservoir. We read the state of the reservoir at the readout layer, and we construct the output of the reservoir as a linear combination of the readout states of the reservoir.

5.1.1 Input layer to the reservoir

In the mathematical framework of reservoir computing, extensively studied in Sec. 3.2.2, the input layer is embodied in the input matrix $\mathbf{W}_{in}$, whose coefficients are called input weights to the reservoir. In this chapter, we investigate

the reservoir performance for two injection strategies, and we describe below the corresponding input matrices.

5.1.1.1 Input to all nodes

The first input strategy is to inject the same power modulation in all the active nodes with random phase shifts, exactly as in chapter 4. Hence, since our reservoir is a 4×4 *SWIRL* network, $\mathbf{W}_{in}$ is a 16×16 diagonal matrix with random elements sampled from a uniform distribution over the interval $[-\pi, \pi]$.

5.1.1.2 Input to the four central nodes

The second strategy to inject the data on our 4×4 *SWIRL* reservoir computer was suggested in [3], and consist on injecting the same power modulation on only the four central nodes of the network, with random phase shifts. Hence, the input matrix $\mathbf{W}_{in}$ is a 16×16 diagonal matrix where only the four coefficients corresponding to the central nodes are non-zero, and are random elements sampled from a uniform distribution over the interval $[-\pi, \pi]$.

5.1.2 *SWIRL* reservoir with heterogeneous nodes

Our implementation relies on the *SWIRL* topology (see Fig. 5.2(a)), already presented in Sec. 4.1.2, however in this structure, we suggest the use of heterogeneous neuron-like computational units. This idea of using neurons with non-identical activation functions in an artificial neural networks has been investigated in other fields of artificial intelligence as a way to improve our understanding of natural nervous systems [11, 12].

We give an in-depth description of the inner structure of our *SWIRL* network in Fig. 5.2(b). In our 4×4 network, the four central nodes (in red) are nonlinear microring resonators, as in the work presented in chapter 4 [6], the outer nodes are either linear nodes (small waveguides, in white) as in the work by our collaborators at the Ghent University [2, 3], or semiconductor optical amplifier (in blue) as in [1].

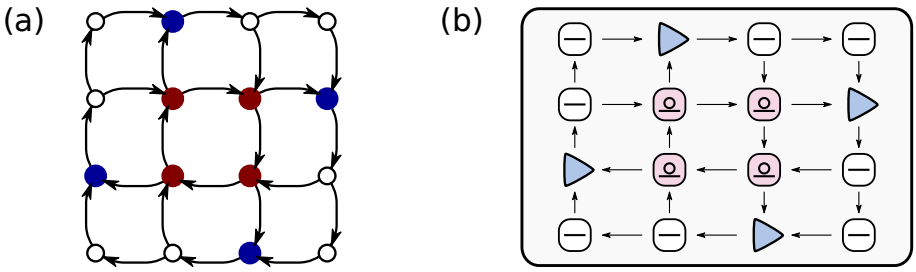

Figure 5.2: (a) Typical scheme of a 4×4 *SWIRL* network. (b) Scheme of the 4×4 *SWIRL* network with heterogeneous nodes as neuron-like computational units. The four central nodes (in red) are nonlinear microring resonators, as in the work presented in chapter 4, the outer nodes are either linear nodes (small waveguides, in white) as in the work by our collaborators at the Ghent University [2, 3], or semiconductor optical amplifier (in blue) as in [1].

We have carefully chosen the location of the SOA in such a way that they are immediately after a nonlinear ring resonator in the *SWIRL* topology. This choice is motivated by the input strategy presented in Sec. 5.1.1.2, namely to inject the data stream on the four central nodes. We expect that the signal coming from the nonlinear microring resonators to be amplified before being sent to the following linear nodes, thereby maximizing the signal-mixing of the whole structure.

5.1.3 Readout layer of the reservoir

This section is devoted to the presentation of the readout layer and the actual output of the reservoir computer. Similarly to the case of the 16-node *SWIRL* reservoir presented in Chap. 4, we measure the state of all reservoir nodes as the output power coming out each node. This measure is performed using a photodetector [13], whose model is depicted in Sec. 2.3.2. In the mathematical framework of reservoir computing presented in (3.3) of Sec. 3.2.2, we report the output power at each node in the readout vector $\mathbf{x}_{readout}$, and we construct the output $\mathbf{y}_{out}$ as a linear combination of the readout vector coordinates. The coefficients of this linear combination are reported in the readout matrix $\mathbf{W}_{out}$,

and are called the output weights of the reservoir. They are determined during the training procedure by a ridge regression.

5.2 Numerical simulation strategy

This section is devoted to the description of the numerical simulation methodology. In particular, we explain in a first part how we modified the library PhotRC to simulate our specific network. In a second part, we present the actual numerical simulation strategy and the post processing methods. Finally, a last part is dedicated to the search of the operating parameters in order for our reservoir to operate at best levels of performance.

5.2.1 Modification of the PhotRC library

We have already introduced the PhotRC library in Sec. 4.2.1. This python library was developed by our collaborators at the Ghent University as an extension of the Caphe photonic circuit simulator [14] in order to simulate complex networks of interconnected elements in the *SWIRL* topology. In particular, the library creates all the necessary objects ($\mathbf{W}_{in}$, $\mathbf{W}_{res}$) to design and simulate a *SWIRL* network when given the number and the type of the nodes.

This library was originally developed to simulate homogeneous on-chip networks, in which all the nodes are the same elements. In this chapter, we investigate the properties of a network made of non-identical nodes. Hence, we modify the library in order to achieve the creation of this particular design of reservoir computer. We created a framework for the heterogeneous distribution of the nodes in a 4×4 network, according to our design presented in Fig. 5.2. More specifically, the four central nodes are 2-port nonlinear microring resonators, as presented in [6, 15, 16]. Four of the outer nodes are 2-port semiconductor optical amplifiers [1, 5], and their location is set in such a way that they are immediately after a nonlinear ring resonator in the *SWIRL* topology. Finally, the other outer nodes are 2-port small waveguides introducing no time delay, as in [2–4].

5.2.2 Data stream and post processing

The reservoir states are obtained through the simulation, using the PhotRC and the Caphe photonic circuit simulator [14] libraries, of the 4×4 *SWIRL* network, where nodes are non-identical, as presented in Fig. 5.2(b). For this reservoir architecture, we measure the level of performance on the delayed-XOR task and the 3-bit pattern recognition task,which have both been presented in Sec. 3.3.

The input data stream is constructed as follows. We feed the system with a signal stream consisting of 17,000 randomly chosen bits. We use in our simulations a sampling rate of 160 Gb/s, and a fourth order Runge Kutta algorithm. The training of the reservoir (i.e the identification of the readout matrix $\mathbf{W}_{out}$) is performed using the regularized ridge regression on 15,000 bits, and the testing is done on the 2,000 remaining bits. We average the performance on either 20 or 50 sets of simulations, and the number of sets of simulations will be specified for each given result.

For the XOR task, the regularization parameter is chosen using the best case from a five-fold cross-validation, and the threshold value separating a "0" and a "1" is at half the amplitude. For the 3-bit pattern recognition task, we

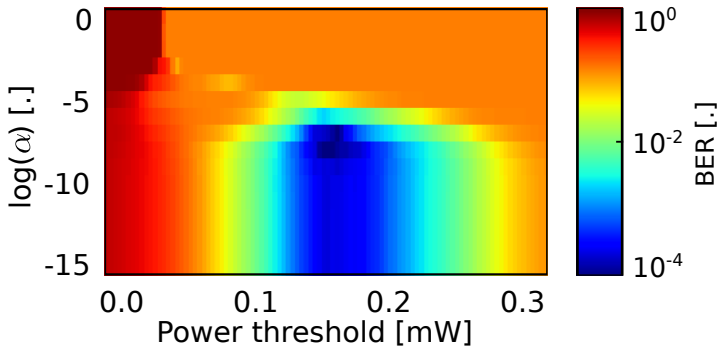

Figure 5.3: Typical map of the error rate in the regularization parameter, threshold value (α, th) plane. The map is averaged on 50 simulations, and maps the error rate for the 3-bit pattern recognition task at 20 Gb/s, with an input on all nodes. The measured error rate is the minimum value of this map, which is 9×10^{-5}.

determine for each value of interdelay the regularization parameter (α) and the threshold value (th) separating a "0" and a "1" using the map presented in Fig. 5.3. This map is obtained through the simulations of the reservoir states for 20 (resp. 50) sets of input signals. We train the reservoir for each value of α, and test the performance of the reservoir for every value of the power threshold th. We then map the error rate for each simulation in this (α, th) plane, and we average on the 20 (resp. 50) sets of simulation the error rate for each point of the map. Finally, the reported error rate is chosen as the minimum error rate of the averaged map, presented in Fig. 5.3.

5.2.3 Operating point of the reservoir

In Sec. 4.3, we highlighted that the search for the optimal injection parameters to operate a reservoir computer are one of the most challenging part of the design of an architecture for reservoir computing applications [17]. Hence, we devote this section to the investigation of optimal operating parameters.

As stated in [18], the fading memory [19] of a system is a very important property for its use as a reservoir computer. A way to maximise the fading memory of a dynamical system was proposed by [20] and consists for the system to be stable, but close to instabilities. This strategy for the driving of the system maximise the length of the transient dynamics, hence maximising the signal mixing in the network for reservoir applications.

We have already found in Sec. 4.3 the optimal injection parameters in terms of power modulation and optical detuning in a network of ring resonators for reservoir computing applications. In the system presented in this chapter, the four central nodes are the same ring resonators as in Chapter 4, hence the information found in Sec. 4.3 is still relevant for the strategy corresponding to an input on the four central nodes. In all the simulations, the bit stream is fed on the injected nodes (either the four central nodes or all the nodes of the network) through a power modulation between $P_{in,0}$ = 0.0 mW and $P_{in,1}$ = 0.3 mW, with a random phase shift. The injected light wavelength is set to λ = 1552.820 nm, corresponding to an optical detuning $\delta\lambda$ = 50 pm with regards to the resonance frequency of the ring resonators.

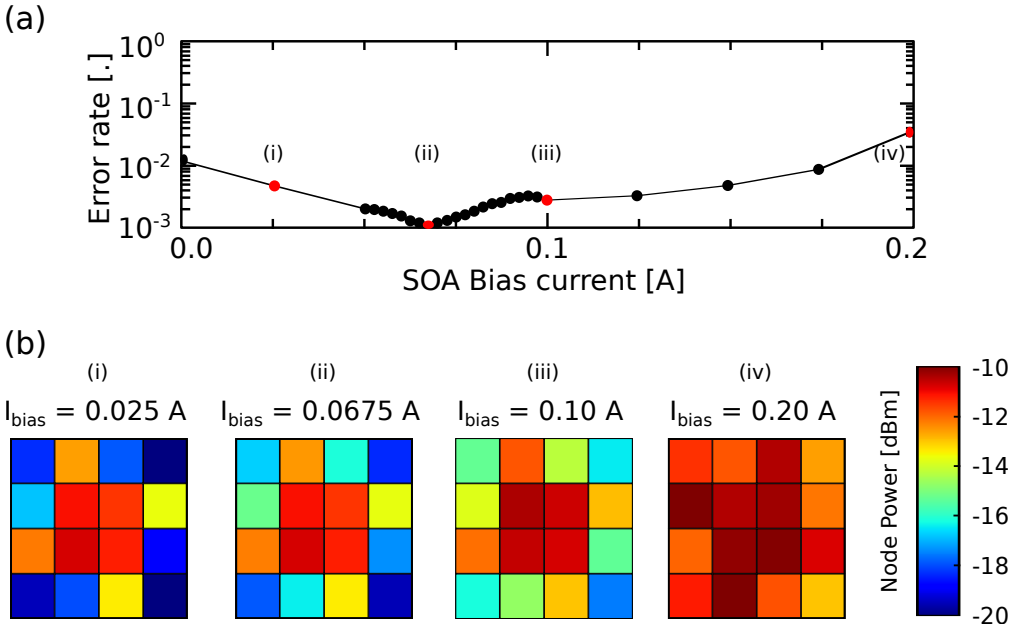

Figure 5.4: (a) Performance of the *SWIRL* 4×4 reservoir computer on the 3-bit pattern recognition task at 30 Gb/s as a function of the bias current of the four SOAs. The pattern we intend to recognise is "101". The performance is averaged on 20 sets of simulations, and the input stream is fed to the four central nodes. (b) Averaged power per node in the same conditions for (i) $I_{bias} = 25.0$ mA, (ii) $I_{bias} = 67.5$ mA, (iii) $I_{bias} = 100$ mA, and (iv) $I_{bias} = 200$ mA. For each case, the corresponding performance of the reservoir is highlighted (red dots) in the upper panel of the figure.

Let us now identify the best operating point for the reservoir in terms of bias current of the semiconductor optical amplifiers. We aim for our reservoir to be used for telecommunication applications, hence we focus our research on one use case. We want to perform the 3-bit pattern recognition task at 30 Gb/s, with an averaged input power as low as possible. This low average power consumption is attained with the injection strategy based on a data stream only fed to the four central nodes of the 4×4 *SWIRL* reservoir.

In order to set the bias current of the optical amplifiers in our network, we plot in Fig. 5.4(a) the level of performance of the hybrid on-chip reservoir

computer 3-bit pattern recognition task at 30 Gb/s, when the input data stream is fed to the four central nodes, as a function of the bias current of the SOAs. We find the best level of performance for a bias current $I_{bias} = 67.5$ mA. We also give in Fig. 5.4(b) (i)-(iv) the averaged power at each node for various bias currents of the SOAs. We show in Fig. 5.4 that the best level of performance is not obtained, as one could expect, when the averaged power is the highest at each node, but for an intermediate value $I_{bias} = 67.5$ mA. This can be explained by a saturation at the SOAs nodes, or also the amplification of non-meaningful signals (*i.e.* simulation noise).

We have determined in this section the operating parameters for our reservoir. These parameters are reported in Table 5.1.

Parameter	Strategy (i)	Strategy (ii)
Input nodes	All nodes	Four central nodes
Light wavelength λ	$\lambda = 1552.820$ nm	$\lambda = 1552.820$ nm
Optical detuning $\delta\lambda$	$\delta\lambda = 50$ pm	$\delta\lambda = 50$ pm
Power modulation	0.0 to 0.3 mW	0.0 to 0.3 mW
Bias current SOA	$I_{bias} = 67.5$ mA	$I_{bias} = 67.5$ mA

Table 5.1: Operating injection parameters for the heterogeneous 4×4 *SWIRL* network used as a reservoir computer.

5.3 Performance of the reservoir

We investigate in this section the level of performance of our heterogeneous reservoir computer. We evaluate the performance on the typical delayed XOR task in Sec. 5.3.1, and on the 3-bit pattern recognition task in Sec. 5.3.2. For both those tasks, we try our two input strategies, namely the input on all nodes and the input on the four central nodes. The injection parameters have been presented for the two strategies in Table 5.1.

5.3.1 Delayed-XOR task

We present in this section the numerically obtained measure of the performance of the 4×4 heterogeneous *SWIRL* network on the delayed-XOR task, and focus alternatively on the performance when we feed the input bit stream on (i) all nodes, and (ii) only on the four central nodes.

5.3.1.1 Input to all nodes

We plot in Fig. 5.5(a) and Fig. 5.5(b) the performance of the hybrid reservoir on the 1-bit delayed XOR task at respectively 20 Gb/s and 30 Gb/s. We see that when we inject the data stream on all active nodes, the level of performance of the reservoir computer on this task at 20 Gb/s and 30 Gb/s is below 10^{-3} for a large set of interdelay values, which corresponds to a large set of possible designs in terms of interconnection waveguide lengths. Indeed, the node interdelay is directly related to the length of the interconnection waveguide by the relation (5.1) :

$$t_{delay} = \frac{L \times n_{Si}}{c}, \tag{5.1}$$

where L is the length of the interconnection waveguide, $n_{Si} = 3.476$ is the refractive index of the bulk silicon, and c is the speed of light.

These numerical results are consistent with the results on the same task for the 4×4 *SWIRL* network made of homogeneous nodes presented in Chapter 4 in which nodes are nonlinear ring resonators. Even the shrinking of the range of node interdelay values that are optimal for using the network as a reservoir computer with the increasing of the data rate is consistent with what was observed in Fig. 4.9(b). However, for both 20 Gb/s and 30 Gb/s data rates, the range of interdelay values for which we attain the best level of performance in improved with the heterogeneous design. At 20 Gb/s, the best interdelay values are in the interval $[18.75, 35]$ ps for the reservoir of Chapter 4 and are in the interval $[18.75, 50]$ for the reservoir presented in this chapter. And similarly at 30 Gb/s, the best interdelay values are in the interval $[18.75, 25]$ ps for the homogeneous reservoir and are in the interval $[12.5, 25]$ for the heterogeneous

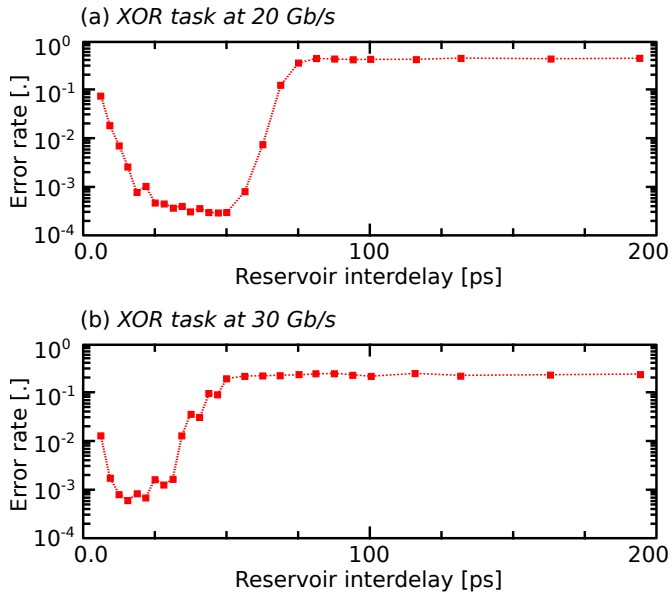

Figure 5.5: Performance of the 4×4 reservoir computer on the delayed XOR task at (a) 20 Gb/s and (b) 30 Gb/s. The input bit stream is fed on all nodes, and the performance is averaged over 50 sets of data.

reservoir. For this task, we benefit from the non-linearity and the reduction of the losses introduced by the SOAs.

5.3.1.2 Input to the four central nodes

We present in Fig. 5.5(a) and Fig. 5.5(b) the performance of the hybrid reservoir on the 1-bit delayed XOR task at respectively 20 Gb/s and 30 Gb/s when we feed the input signal only on the four central nodes. We clearly see in these figures that when we inject the signal only on the four central nodes, this reservoir with non-identical neuron-like computational units cannot resolve the 1-bit delayed XOR task.

The 1-bit delayed XOR task is a very nonlinear binary task. The fact that our heterogeneous reservoir cannot resolve this task when we inject the data only on the four central nodes could be due to the fact that this injection strategy does not trigger enough non linearity. Indeed, when we inject only

to the four central nodes, the SOAs are driven in a very linear amplification regime, while they behave more non-linearly when they are excited with more power, for instance when we inject the data strem on all nodes, including the SOAs.

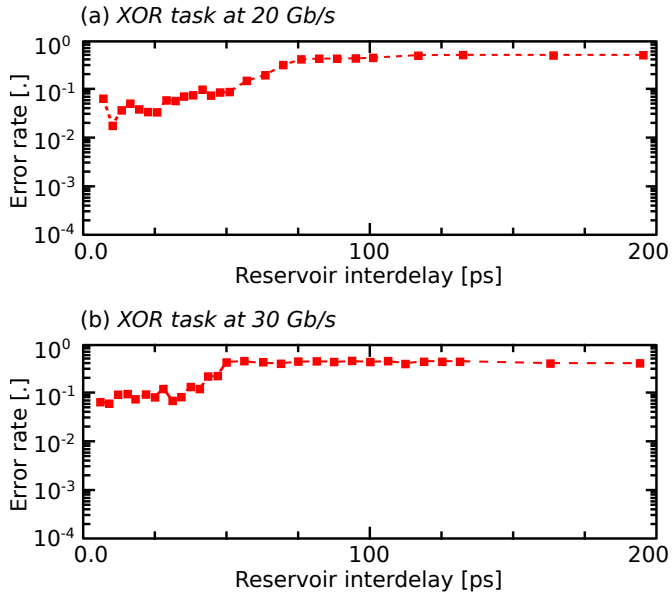

Figure 5.6: Performance of the 4×4 reservoir computer on the delayed XOR task at (a) 20 Gb/s and (b) 30 Gb/s. The input bit stream is fed on the four central nodes, and the performance is averaged over 50 sets of data.

5.3.2 Header recognition

We now study the level of performance that can be attained by our heterogeneous *SWIRL* reservoir on a more telecommunication oriented task, namely the pattern recognition task. Indeed, we target for this kind of architecture to by able to perform header recognition at telecommunication data processing rates, with sufficient level of performance, *i.e.* about 10^{-3}. We also focus alternatively on our two input strategy (i) input on all nodes and (ii) input on the four central nodes.

5.3.2.1 Input to all nodes

We plot in Fig. 5.7 the level of performance attained by this reservoir computer on the 3-bit pattern recognition task at respectively (a) 20 Gb/s and (b) 30 Gb/s. We train our reservoir to recognise the pattern "101". Our reservoir is able to resolve this task at 20 Gb/s for a range of interdelay values in the interval $[25,50]$ ps with error rates below 10^{-3}, and we attain a $BER = 10^{-4}$ for three values of node interdelay : $t_{delay} = 34,4$ ps, $t_{delay} = 37.5$ ps, and $t_{delay} = 46.9$ ps. For this last value $t_{delay} = 46.9$ ps, using the post processing strategy presented in Sec. 5.2.2, we attain our best performance $BER = 9 \times 10^{-5}$.

On the contrary, Fig. 5.7(b) shows that this heterogeneous reservoir computer cannot resolve the 3-bit pattern recognition task at 30 Gb/s when we inject on all nodes. In order to explain this, we have measured the memory capacity of this system at 30 Gb/s. The methodology used to evaluate the memory

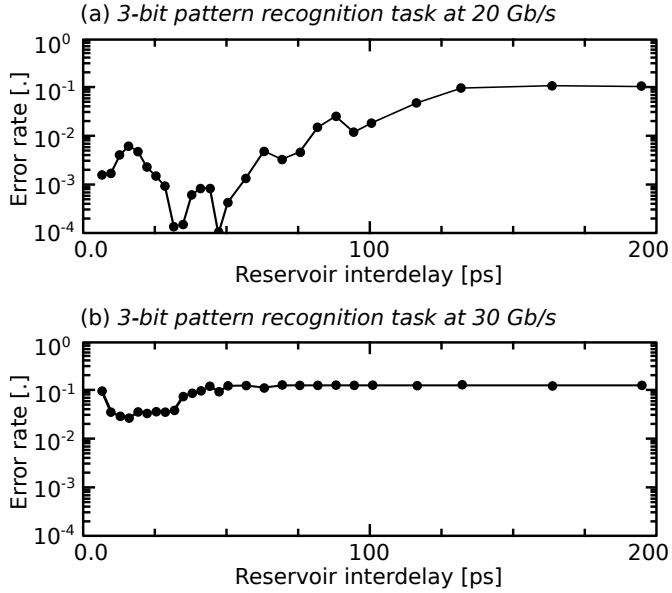

Figure 5.7: Performance of the 4×4 reservoir computer on the 3-bit patter recognition task task at (a) 20 Gb/s and (b) 30 Gb/s. The 3-bit pattern we intend to recognise is [1.0.1]. The input bit stream is fed on all nodes, and the performance is averaged over 50 sets of data.

capacity of a system has been previously presented in Sec. 3.3.1. We measure a memory capacity of 2.29 when we inject the data stream on all nodes at 30 Gb/s. Hence, with this value of memory capacity, we understand that our system cannot resolve a task demanding a memory depth greater than two at 30 Gb/s when we inject on all nodes. This can explain why we could resolve the 1-bit delayed XOR task (memory depth equal 2) in the same injection conditions, but not the 3-bit pattern recognition task (memory depth equal 3).

We finally plot in Fig. 5.8(a) the time series obtained at the output of each node in grey when we inject the data stream in blue on all nodes. Note that the input data stream has been scaled down in this figure for clarity purpose, and is a power modulation between $P_{in,0} = 0.0$ and $P_{in,1} = 0.3$ mW. Figure 5.8(b) shows the output of the trained reservoir (blue), desired output (green), and decision threshold (red) for the 3-bit pattern recognition task at the best performance for an input on all nodes at 20 Gb/s. In both plots, the time is normalized with respect to the duration of one bit.

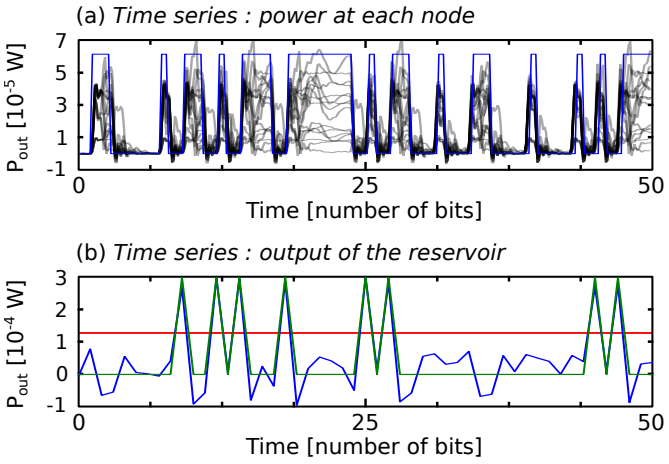

Figure 5.8: (a) Output power at each node (grey). The input stream (blue) is scaled down for clarity purpose. (b) Output of the trained reservoir (blue), desired output (green), and decision threshold (red) for the 3-bit pattern recognition task at the best performance for an input on all nodes at 20 Gb/s. In both plots, the time is normalized so that one bit is equal to one unit of time.

5.3.2.2 Input to the four central nodes

We plot the averaged performance in Fig. 5.9 at respectively (a) 20 Gb/s and (b) 30 Gb/s for 50 sets of simulations. We see that for both data rates presented, the heterogeneous reservoir can resolve this task with error rate levels of the order of 10^{-3} for at least one node interdelay value, which corresponds to state-of-the-art levels of performance on this task for similar architectures [3].

Interestingly, our system is able to resolve the 3-bit pattern recognition task at 30 Gb/s when we inject the data stream only on the four central nodes at state-of-the-art level of performance $BER = 10^{-3}$, but cannot resolve the same task when we inject the signal on all 16 nodes. Our explanation for this phenomenon is the trade-off between the non-linearity of the nodes and the linear memory capacity of a reservoir computer [21, 22]. Indeed, when we inject on all nodes, the SOAs are excited with more power, and thus behave more

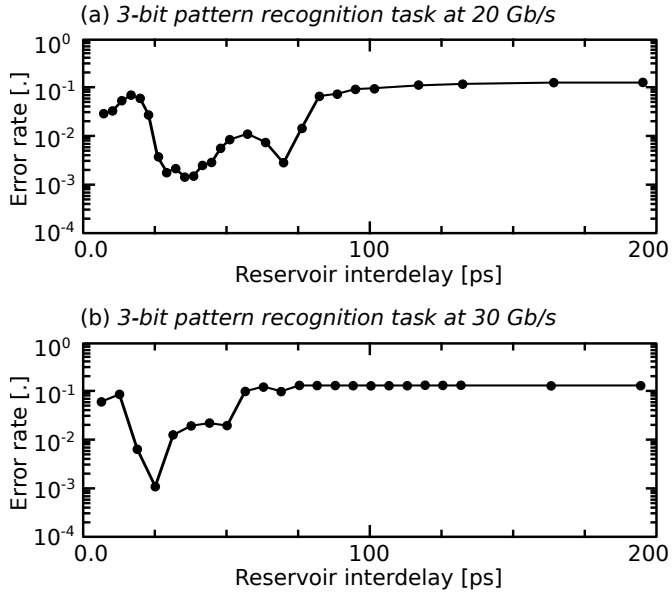

Figure 5.9: Performance of the 4×4 reservoir computer on the 3-bit patter recognition task task at (a) 20 Gb/s and (b) 30 Gb/s. The 3-bit pattern we intend to recognise is [1.0.1]. The input bit stream is fed on the four central nodes, and the performance is averaged over 50 sets of data.

non-linearly than when we inject only on the four central nodes. Hence the system, when injected on all nodes, performs better on the very non-linear XOR task (see Sec. 5.3.1), and cannot perform well the pattern recognition task that needs more linear memory. And on the contrary, when we inject only on the four central nodes, the SOAs receive less power, and thus behave more linearly. Hence the system performs better on the task that needs linear memory like the 3-bit pattern recognition task.

We finally plot in Fig. 5.10(a) the time series obtained at the output of each node in grey when we inject the data stream in blue on the four central nodes. Note that the input data stream has been scaled down in this figure for clarity purpose, and is a power modulation between $P_{in,0} = 0.0$ and $P_{in,1} = 0.3$ mW. Figure 5.10(b) shows the output of the trained reservoir (blue), desired output (green), and decision threshold (red) for the 3-bit pattern recognition task at the best performance for an input on all nodes at 30 Gb/s. In both plots, the time is normalized so that one bit is equal to one unit of time. In comparison

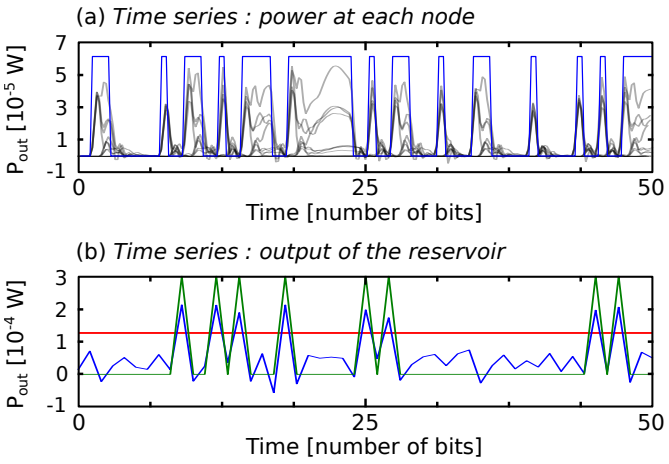

Figure 5.10: (a) Output power at each node (grey). The input stream (blue) is scaled down for clarity purpose. (b) Output of the trained reservoir (blue), desired output (green), and decision threshold (red) for the 3-bit pattern recognition task at the best performance for an input on the four nodes at 30 Gb/s. In both plots, the time is normalized so that one bit is equal to one unit of time.

to Fig. 5.8, we see in the upper panel that there is less power at each node which is consistent with the fact that there is less optical power in the whole structure, and we see in the lower panel that even if we can resolve the 3-bit pattern recognition task, the decision threshold must be carefully chosen as the discrimination between a "1" and a "0" is really narrow.

5.3.3 Summary of the performance levels

We have presented in Sec. 5.3.1 and 5.3.2 the performance of our 4×4 *SWIRL* reservoir computer with non-identical neuron-like computational units on two typical tasks used to benchmark this kind of reservoir architectures. We summarise in Table. 5.2 the best performance levels attained by our reservoir. From this table, it is interesting to note that it is possible to adapt the input strategy and the speed to be able to resolve these two tasks with good levels of performance.

5.4 Conclusion

To conclude, we have designed an on-chip all-optical reservoir computer with non-identical nodes that can perform at very high speed on two benchmark tasks : the Boolean delayed-XOR task, and the 3-bit pattern recognition task.

	1-bit delayed XOR		3-bit pattern recognition	
Data rate	St (i)	St (ii)	St (i)	St (ii)
20 Gb/s	2.5×10^{-4}	2×10^{-2}	9×10^{-5}	10^{-3}
30 Gb/s	5×10^{-4}	10^{-1}	2×10^{-2}	10^{-3}

Table 5.2: Performance of the heterogeneous reservoir computer on the delayed XOR task and the 3-bit pattern recognition task. We report in this table the best error rate attained in each case. St (i) and St (ii) correspond respectively to the two injection strategies, *i.e.* feeding the input signal to all nodes and feeding the input signal to the four central nodes.

The nodes can be microring resonators, waveguides, or SOAs. We have found the best operating point in terms of SOA bias current to be 0.0675 A, while we inject optically the data as a power modulation between 0 and 0.3 mW, and an optical detuning of 50 pm with regards to the ring resonators according to previous work [6]. With these injection parameters, we attain state-of-the-art levels of performance on the 3-bit pattern recognition task when we inject only on the four central nodes, with an averaged optical input power of 0.6 mW and an electrical pump of 0.0675 A per SOA. This already low power consumption could be further improved by using more recent SOA technologies like quantum dots SOAs [23].

This work opens new research venues aiming at integrated, high-speed, energy-efficient, all-optical data processing for telecommunication applications. More specifically, our photonic reservoir approach allows for scalability, *i.e* one could add more nodes in the reservoir in order to perform the pattern recognition task for longer patterns, or for all-optical routing applications. Moreover, the fact that this hybrid reservoir computer performs better on a pattern classification than a reservoir with identical nodes could suggest that having heterogeneous nodes in a reservoir computing architecture, or more generally having neurons with non-identical activation functions in an artificial neural network, could improve our understanding of natural nervous systems, as suggested in [24] in which the authors study heterogeneous artificial neural networks (using natural animal neural systems as examples) to improve the performance of a cognitive computer on an adaptation task.

References

[1] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[2] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[3] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[4] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

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

[6] F. Denis-Le Coarer, M. Sciamanna, A. Katumba, M. Freiberger, J. Dambre, P. Bienstman, and D. Rontani, "All-optical reservoir computing on a photonic chip using silicon-based ring resonators," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 24, no. 6, pp. 1–8, 2018.

[7] C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," *JOSA B*, vol. 30, no. 11, pp. 3048–3055, 2013.

[8] C. Mesaritakis, A. Kapsalis, and D. Syvridis, "All-optical reservoir computing system based on ingaasp ring resonators for high-speed identification and optical routing in optical networks," *SPIE OPTO*, pp. 937 033–937 033, 2015.

[9] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[10] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[11] R. D. Beer, H. J. Chiel, and L. S. Sterling, "Heterogeneous neural networks for adaptive behavior in dynamic environments," in *Advances in neural information processing systems*, 1989, pp. 577–585.

[12] R. A. Stefanescu and V. K. Jirsa, "A low dimensional description of globally coupled heterogeneous neural networks of excitatory and inhibitory neurons," *PLoS computational biology*, vol. 4, no. 11, p. e1000219, 2008.

[13] F. R. Times, W. Ranges, and W. W. Ranges, "Ultrafast Photodetectors," *Data Sheet*, 2000.

[14] http://www.lucedaphotonics.com/, accessed: 2016.

[15] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[16] T. Van Vaerenbergh, "All-optical spiking neurons integrated on a photonic chip," *PhD Thesis*, 2014.

[17] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega, "Optimal nonlinear information processing capacity in delay-based reservoir computers," *Scientific reports*, vol. 5, p. 12858, 2015.

[18] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[19] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[20] T. Yamane, S. Takeda, D. Nakano, G. Tanaka, R. Nakane, S. Nakagawa, and A. Hirose, "Dynamics of reservoir computing at the edge of stability," *International Conference on Neural Information Processing*, pp. 205–212, 2016.

[21] H. Jaeger, *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001, vol. 5.

[22] M. Inubushi and K. Yoshimura, "Reservoir Computing beyond Memory-Nonlinearity Trade-off," *Scientific Reports*, vol. 7, no. 1, pp. 1–10, 2017.

[23] G. Contestabile, Y. Yoshida, A. Maruta, and K. Kitayama, "Ultra-broad band, low power, highly efficient coherent wavelength conversion in quantum dot SOA," *Optics Express*, vol. 20, no. 25, p. 27902, 2012.

[24] R. D. Beer, H. J. Chiel, and L. Sterling, "Heterogeneous Neural Networks for Adaptive Behavior in Dynamic Environments," *Advances in Neural Information Processing Systems 1*, pp. 577–585, 1989.

CHAPTER 6

DELAY-BASED RESERVOIR COMPUTING ON A SILICON PHOTONIC CHIP

"Never do today what you can put off till tomorrow. Delay may give clearer light as to what is best to be done."

— A. Burr*, The Complete Art of Public Speaking*

We have presented in Chap. 4 and Chap. 5 two architectures for on-chip reservoir computing relying on the implementation of an extended network of 16 physical nodes, respectively using only nonlinear ring resonators or non-identical nodes. These architectures can perform very well on simple binary tasks, like the delayed XOR task or the 3-bit pattern recognition task. However, the limited number of nodes does not allow more complex tasks to be resolved with a satisfactory level of performance. Moreover, having a large number of nodes in integrated photonic reservoir computing is a significant technological challenge, and most of the reservoir computing systems have been investigated for a small number of nodes, for example 16 nodes in [1–4]. Indeed, the losses in the many splitters, couplers, and long waveguides do not allow for sufficient signal mixing in the reservoir, thus reducing significantly the performance on challenging tasks like chaotic time series prediction.

A quite convincing solution to scale up the number of nodes in a reservoir computer was suggested in 2011 by L. Appeltant [5], and consists of having only one nonlinear node and distributing a large number of virtual nodes in a long delay line. This kind of time-delayed architectures has attracted a lot of focus lately, as it can attain very good levels of performance using opto-electronics systems [6–9], or all-optical systems [10–15]. However, implementing such an architecture on a silicon photonic chip remains a very challenging technological issue. Indeed, the length of the delayed feedback loop usually range from a few meters (for example about 12 m in [15]) to a few kilometers (for example 1.6 km in [16]), and the typical losses in an on-chip silicon waveguide is between 0.1 dB/cm to 3 dB/cm [17], thus implementing a delay based reservoir computer on a photonic chip was only suggested a few times [18–20], mostly on InP chips, hence allowing to integrate laser sources directly on the photonic chip.

We suggest in this chapter a novel time-delayed architecture integrated on a silicon photonic chip using a ring resonator as nonlinear node. In our approach and similarly to [19], the virtual nodes are distributed along multiple round-trips in the delay line, hence the length of the delay waveguide is kept small and the corresponding losses remain small. The idea of distributing the virtual nodes in multiple round-trip in the delay line naturally emerges from the model of a nonlinear ring resonator submitted to delayed optical feedback, and can successfully be implemented by changing the masking procedure for delay-based reservoir computing. In this chapter, we numerically study this novel design for on-chip neuromorphic computing, and we demonstrate that it is possible to attain state-of-the-art levels of performance with this very compact structure. In particular, we successfully resolve the Santa Fe chaotic time series prediction task with a normalised mean square error of 1.12×10^{-2}, and at a processing speed of 0.3 Giga symbols per second. We also show that with this architecture, the scalability of the number of nodes is very straightforward. Indeed, we do not need to change the design of the chip to add more nodes, but only to distribute the nodes on more round-trip.

This chapter is organised as follows. We present in Sec. 6.1 the general structure of a delay-based architecture for reservoir computing, along with the masking and reading procedures. We devote Sec. 6.2 to the presentation of

our on-chip design for time-delayed reservoir computing. More specifically, we give the full model of a ring resonator submitted to delayed optical feedback, and then explain the modified masking procedure. We present in Sec. 6.3 the numerical simulation methods, and in particular the training and testing procedures, and how we choose the best reservoir operating point for our simulations. Finally we numerically analyse in Sec. 6.4 the performance of the on-chip delay-based reservoir computer. More specifically, we measure the memory capacity of the system, the performance on the delayed-XOR task in order to compare the performance with the reservoir presented in Chap. 4, and the performance on the Santa Fe chaotic time series prediction task. We report our conclusions in Sec. 6.5.

The results presented in this chapter have been submitted for publication in a peer-review journal, and the article is still in the review procedure.

6.1 Time-delayed reservoir computing

The paradigm of delay-based reservoir computing was introduced in 2011 by L. Appeltant as a solution to reduce the complexity of the system used for reservoir computer applications [5]. In this kind of architecture, a nonlinear node is submitted to delayed feedback, and an arbitrary number of virtual nodes are distributed in the delay line of length L and duration τ. We present in Fig. 6.1 a figure taken from the supplementary materials of [5] showing the typical architecture as presented by L. Appeltant *et al.*. This figure is the first representation of the concept of time-delay reservoir computing. Along the delay line of duration τ are distributed N virtual nodes, separated by a temporal distance θ from each other. A mask of duration τ is superimposed on the input data (see Sec. 6.1.2.1), and the output of the reservoir is constructed as the weighted sum of the readout of the virtual time-multiplexed nodes, as presented in Sec. 6.1.2.2.

The rest of this section is organised as follows. We present in Sec. 6.1.1 the typical architecture for delay-based reservoir computing, and the various experimental systems that have been recently investigated. We finally devote

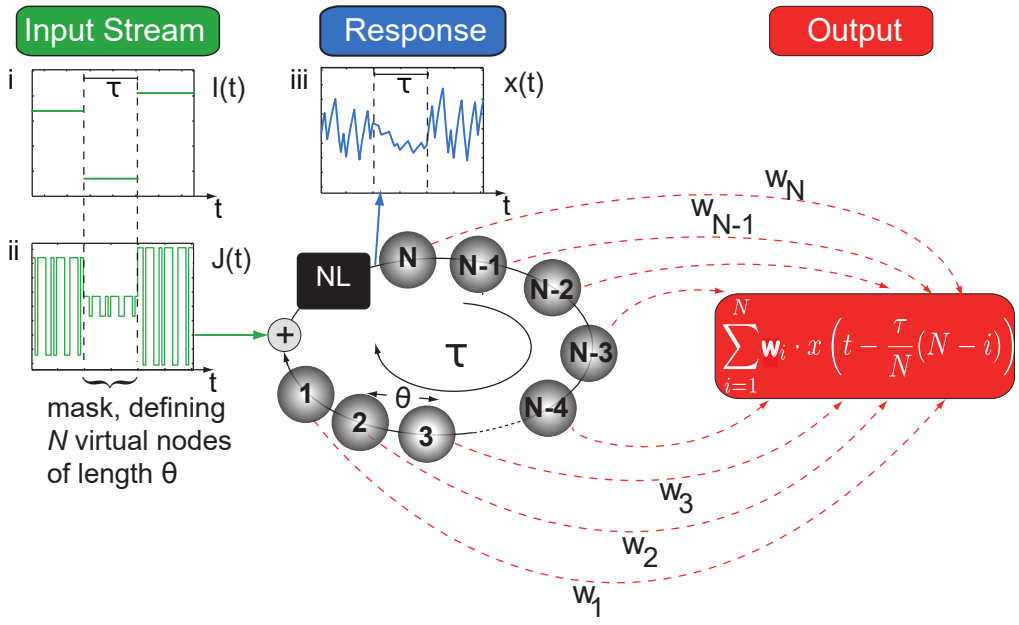

Figure 6.1: Scheme of single nonlinear node reservoir computer. Along the delay line are distributed N virtual nodes, separated a distance θ from each other. (i)-(ii) To create more diversity in the states a mask is superimposed on the input. (iii) The transient response of the node. Figure taken from supplementary material of Appeltant *et al.* [5]. This figure is the first introducing the concept of time-delay reservoir computing. We introduce with more details the different inserts of this figure in our own figures in this section.

Sec. 6.1.2 to the presentation of the masking and reading procedures that defines the interconnection structure of the reservoir.

6.1.1 Single node with delayed feedback

In the framework of liquid-state machines [21] and echo-state network [22], the reservoir is driven by an input signal and generates high-dimensional transient responses using a network of interacting neurons. The output of the reservoir is eventually constructed as a weighted sum of the readout layer vector coordinates. Another technique to generate high-dimensional patterns from an input signal would be to use a time-delayed dynamical system [23].

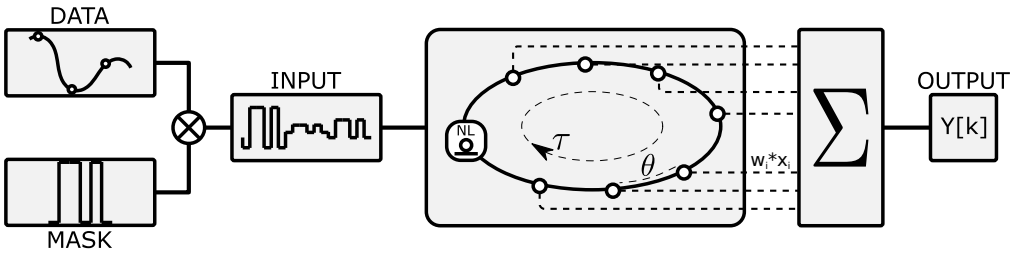

Figure 6.2: Typical scheme of a delay-based reservoir computer. The input data is masked through the procedure described in Sec. 6.1.2.1. The reservoir is made of a nonlinear node submitted to delayed feedback, and N virtual nodes are distributed every θ in the delay line of length τ. The output is the weighed sum of the component of the readout vector, constructed according to Sec. 6.1.2.2.

Such a system can be described by Eq. (6.1) :

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = F(t, \mathbf{x}(t), \mathbf{x}(t-\tau)), \tag{6.1}$$

where t is the continuous time, $\mathbf{x}$ is the state vector, F is a nonlinear function representing the flow of the system, and τ is the time delay. This kind of delayed systems can exhibit rich dynamical behaviours including periodic oscillations [24] or deterministic chaos [25].

The idea of using a time-delayed dynamical system as a reservoir computer consists of having only one nonlinear node submitted to delayed feedback. We give in Fig. 6.2 the typical scheme of a delayed based reservoir computer. In this delay-based architecture for reservoir computer, the nonlinear node is described by the state vector $\mathbf{x}$ of Eq. (6.1), and the flow of the system is represented by F. The time delay τ is therefore the length of the delay line, and N virtual nodes are distributed every θ in this delay line. Equation (6.2) gives the relation between the number of nodes N, the node interdelay θ, and the length of the delay line τ.

$$\tau = N \times \theta. \tag{6.2}$$

This kind of delay-based architecture for reservoir computing has attracted a lot of attention lately. In fact, it is the most extensively studied implementation for optical reservoir computers, using either opto-electronic systems [6–9, 16, 26–30], or all-optical systems [10–12, 14, 15, 31–35]. We report in Table 6.1 some important results on this subject.

Paper	Important result
L. Appeltant *et al.* [5]	Suggestion of the architecture.
L. Larger *et al.* [6] Y. Paquot *et al.* [7]	First experimental demonstration with an opto-electronic feedback loop.
F. Duport *et al.* [10]	All-optical system with an optical amplifier and a fiber coupler.
M. Soriano *et al.* [8]	Importance of the pre-processing mask to limit the impact of noise on the performance.
R. Nguimdo *et al.* [12]	Chaotic time series prediction and nonlinear channel equalisation tasks.
Y. Kuriki *et al.* [36]	Colored-noise or chaotic mask can improve the performance.
J. Vatin *et al.* [14, 15]	Improvement of the performance when using the polarization dynamics of a VCSEL.

Table 6.1: Important results on delay-based architectures for reservoir computing. This table is not exhaustive.

However, the implementation of such a delay-based architecture with a long delay line on a photonic chip has only been suggested a few times yet [19, 20], due to the typical losses in an integrated waveguide (typical losses between 0.1 and 3.0 dB/cm [17]). In our approach for Si-based on-chip delay-based reservoir computing, we use a nonlinear ring resonator as a nonlinear node and an relatively small integrated waveguide (2.85 cm) as feedback loop, and

we distribute the virtual nodes along multiple round-trips in this feedback loop (similarly to Ref. [19]), hence the length of the delay waveguide is kept small and the corresponding losses remain limitted.

6.1.2 Time multiplexed virtual nodes

We present in this section the masking and reading procedures for the time multiplexing of the virtual nodes of the network. These procedures were introduced by L. Appeltant in 2011 [5] as the key concept to implement this kind of delay-based architectures for reservoir computing.

6.1.2.1 Masking procedure

The masking procedure is a necessary step for time-delayed architectures to work efficiently, since it keep the reservoir in a dynamically rich regime. As presented in Fig. 6.1 and Fig. 6.2, the input data undergoes a series of transformations before being fed to the reservoir, called the masking procedure, and presented in Fig. 6.3. First, the time-continuous $u(t)$ or time-discrete $u[k]$ is converted to an input stream that is constant during one delay interval τ through a sample-and-hold operation. In principle the reservoir could be fed with this piece-wise constant stream, but this would lead to a low-dimensional response in the transient time series of the delayed system. Hence, a τ-periodic

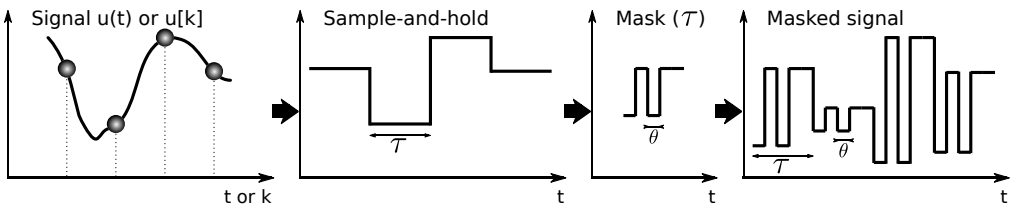

Figure 6.3: Description of the masking procedure. The time-continuous $u(t)$ or time-discrete $u[k]$ is converted to an input stream that is constant during one delay interval τ through a sample-and-hold operation. This stream is convoluted to a mask of length τ, divided in N constant intervals of length θ. This procedure defines the N virtual nodes, and set the node interdelay θ.

function is superimposed to the input, with N piece-wise constant values of length θ. This procedure defines the N virtual nodes, and set the node interdelay θ. The choice of the node interdelay value is of great importance as it sets the dynamical richness in the transient response of the reservoir.

In Fig. 6.3, we have shown an example of binary mask, as most implementations are using this kind of masks. Nevertheless, more complex masks have been explored recently by Kuriki *et al.* in [36] and they demonstrated an improvement of the performance of the reservoir when using a chaotic or colored-noise mask, provided a proper selection of the cut-off frequency of the signal used as mask. Moreover and as described in Sec. 2.1.2 of L. Appletant thesis [37], the mask is a natural representation of the input matrix $\mathbf{W}_{in}$ of Eq. (3.2), as it sets the input weight at each node.

6.1.2.2 Readout of the reservoir

The construction of the readout of a delay-based reservoir deserves some further explanation. Indeed, we only record one continuous time trace at the output of the system, and construct sequentially the coordinates of the readout vector $x_{readout}$. We present in Fig. 6.4 the construction of the readout vector : the output signal is sampled down every θ, and for each input i, the k-th

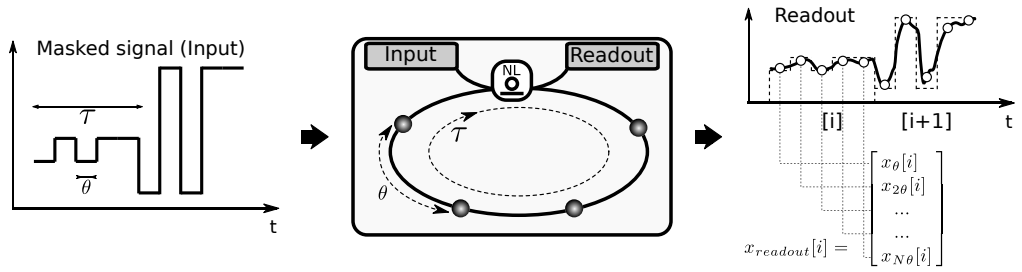

Figure 6.4: Readout layer of the reservoir. The masked input signal is fed to the neural network, and the signal coming out from the system is processed according to the following procedure to reconstruct the readout layer of the reservoir. The output signal is sampled every θ, and for each input i, the k-th component of the readout vector $x_{readout}[i]$ is $x_{k\theta}[i]$, as presented in the right panel of the figure.

component of the readout vector $x_{readout}[i]$ is $x_{k\theta}[i]$, as presented in the right panel of Fig. 6.4. In this figure, the sampling position is chosen in the middle of θ. However, other choices of sampling position can also yields good results, and we have here a degree of freedom to slightly improve the performance.

A very important issue with delay-based reservoir computing architectures is the processing speed. Indeed, to reconstruct the output of the reservoir $y[i]$ as the weighted sum of the coordinates of the readout vector $x_{readout}[i]$, we must have recorded all the $x_{k\theta}[i]$, hence it takes exactly the time τ to construct the output of one sample. Finally, the processing speed will be set by the length of the delay line and is limited to $1/\tau$.

6.2 On-chip time-delayed reservoir computer

We present in this section the design of our silicon on-chip time-delayed reservoir computer. As mentioned previously, the on-chip implementation of a delay-based architecture for reservoir computing is a technological challenge, as the losses in the long feedback loop would drastically reduce the signal mixing in the reservoir. Also, the on-chip integration of a waveguide with a length in the range of a few meters would considerably reduce the compactness of the chip, hence questioning the relevance of the implementation. However, we suggest in this chapter to distribute the virtual nodes in multiple round-trips in the delay line, quite similarly to K. Takano *et al.* in [19]. Hence, the waveguide used for the delay line remains short, thus reducing the corresponding losses, and improving the compactness of the system.

6.2.1 System used as a reservoir computer

This section is devoted to the presentation of the system used as an on-chip delay-based reservoir computer, that is a nonlinear ring resonator submitted to delayed optical feedback. The models of each component of the system have been extensively studied in Secs. 2.3 and 2.4, and we investigate in this section the modelling of the whole structure.

6.2.1.1 Ring resonator with delayed feedback

Our system for time-delayed reservoir computing on a silicon photonic chip is presented in Fig. 6.5. The nonlinear node is a ring resonator, as presented in Sec. 2.4 and already used as nonlinear node of our previous structures presented respectively in Chaps. 4 and 5. We create a relatively small feedback loop through a 2.85 cm-long waveguide, with a typical loss factor 1 dB/cm. The data stream s_{in} is fed to the reservoir through a 3-port combiner (see Sec. 2.3.3) with a 90/10% splitting ratio. Similarly, the output signal s_{out} is read though a 3-port splitter (see Sec. 2.3.3), also with a 90/10% splitting ratio. This output is then connected to a photodetector whose model is based on an experimental equipment in the Ghent University laboratory, the Alphalas UPD-15-IR2-FC photodetector (see Sec. 2.3.2 and [38]). Finally, in order to compensate for the losses in (i) the long waveguide and (ii) the splitter and the combiner, we put a semiconductor optical amplifier [39] in the delay line that improve the quality factor of the whole loop.

Figure 6.6 shows a very accurate scheme of the physical implementation of

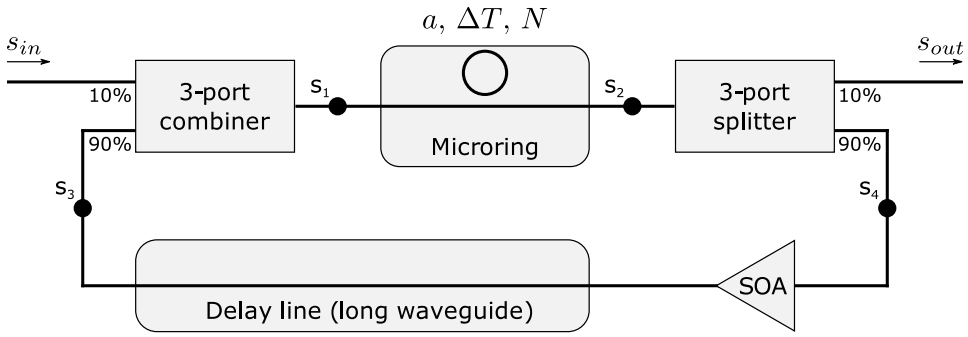

Figure 6.5: Scheme of the on-chip system we intend to use as a time-delayed reservoir computer. A nonlinear microring resonator is submitted to delayed optical feedback through a 2.85 cm-long waveguide. The input and the output are connected through a 90/10% splitter and a 90/10% combiners. Finally, a semiconductor optical amplifier is added in the feedback loop to compensate for the losses in the delay line, the splitter, and the combiner. s_1, s_2, s_3, and s_4 are virtual points of interest used for the calculations.

this kind of architecture. The design of the spiral waveguide allows a reduction of the spatial congestion, hence the structure used as a reservoir computer is very compact. Moreover, as presented in Sec. 6.2.2, the virtual nodes are distributed in multiple round-trips in the reservoir. Hence, once the design is fixed, we still can tune the node interdelay and the number of nodes, just by doing a different number of round-trips in the delay line. Hence, such a design allows great flexibility.

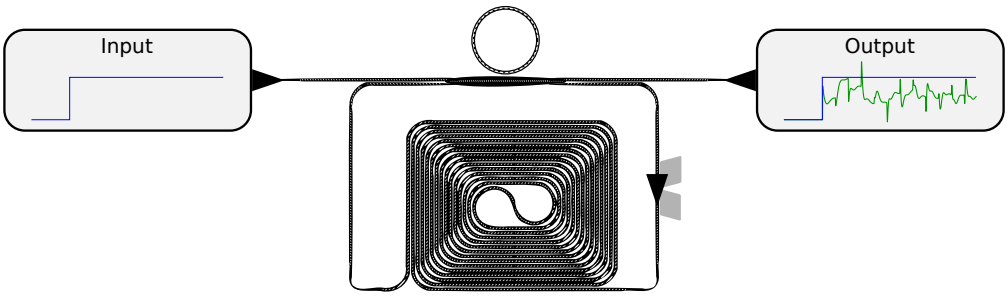

Figure 6.6: Scheme of the physical design of the system. We use a spiral waveguide to reduce the spacial congestion. Hence, the design is very compact.

6.2.1.2 CMT-model of the system

We present in this section the coupled mode theory model of a ring resonator submitted to optical feedback as depicted in Fig. 6.5. We give in particular the calculations used to determine the expression of the output signal s_{out} as a function of the state variables of a ring resonator, the waveguide model, and the 3-port directional couplers with a splitting ratio $\beta = 90\%$. The models of these components have been presented in Sec. 2.3 and 2.4. In our calculations, we model the optical amplifier by its gain G in order to simplify our initial analysis, however in our numerical simulations, we use the SOA model suggested in 1989 by Govind P. Agrawal [39].

We calculate the output signal s_{out}, and the power can be calculated by $P_{out} = |s_{out}|^2$. From the model of a 3-port combiner with a spitting ratio β, we can deduce

$$s_{out}(t) = \sqrt{1-\beta^2}\, s_2(t), \tag{6.3}$$

where s_2 has been defined in Fig. 6.5 as a point of interest. We can link the signal s_2 coming out of a nonlinear ring resonator to the input signal to the ring resonator and the optical mode a of the ring resonator through the input/output relationship of the coupled mode theory model of a ring resonator. Hence, we have

$$s_2(t) = e^{j\phi_c} s_1(t) + \kappa a(t), \tag{6.4}$$

where j is the imaginary unit, ϕ_c is the phase shift induced by the very small waveguide coupled to the ring resonator, and κ is the coupling coefficient of the ring resonator. Once again, from the model of a 3-port combiner used as a directional coupler, we get the expression of s_1 as a function of the input signal to the system s_{in} (with $P_{in} = |s_{in}|^2$), and the signal at the point s_3, given in Eq. (6.5) :

$$s_1(t) = \sqrt{1-\beta^2}\, s_{in}(t) + \beta s_3(t). \tag{6.5}$$

The expression of s_3 is deduced from the model of a waveguide already investigated in Sec. 2.3.3, and the phenomenological gain G of the semiconductor optical amplifier. The signal at the output of a waveguide is the signal at the input with a loss factor $\alpha(\lambda)$, and a delay τ depending on the length of the waveguide. Hence, we have the following expression :

$$s_3(t) = G\alpha(\lambda) s_4(t-\tau). \tag{6.6}$$

We use once again the model of a 3-port directional coupler to extract the expression of s_4 as a function of s_2 :

$$s_4(t-\tau) = \beta s_2(t-\tau), \tag{6.7}$$

with β the splitting ratio of the 3-port splitter. Finally, from Eqs. (6.4) to (6.7), we can deduce a first expression of the signal at the point s_2 at any time t. This equation is given in Eq. (6.8).

$$s_2(t) = \kappa a(t) + e^{j\phi_c}\left[\sqrt{1-\beta^2}\, s_{in}(t) + \beta^2 G\alpha(\lambda) s_2(t-\tau)\right]. \tag{6.8}$$

We can follow the same logic to get the expression of $s_2(t-\tau)$ as a function of the variables. The expression is very similar and is given in Eq. (6.9).

$$s_2(t-\tau) = \kappa a(t-\tau) + e^{j\phi_c}\left[\sqrt{1-\beta^2}\, s_{in}(t-\tau) + \beta^2 G\alpha(\lambda) s_2(t-2\tau)\right]. \tag{6.9}$$

Th expression of $s_2(t-k\tau)$ can be generalised for $k \in \mathbb{N}$. We can then combine all these equations with Eq. (6.3) to get the following expression of the output signal s_{out} :

$$\begin{aligned} s_{out}(t) = \sqrt{1-\beta}\Bigg[\sqrt{1-\beta}\, e^{j\phi_c} \sum_{n=0}^{\infty} G^n \alpha(\lambda)^n e^{jn\phi_c} \beta^n s_{in}(t-n\tau) \\ + \kappa \sum_{n=0}^{\infty} G^n \alpha(\lambda)^n e^{jn\phi_c} \beta^n a(t-n\tau)\Bigg]. \end{aligned} \tag{6.10}$$

As expressed in Sec. 2.4, a ring resonator is not only described in the coupled mode theory framework by its input/output relationship, but also by three state variables : the optical mode a, the temperature variations ΔT, and the free carriers concentration N. The ordinary differential equations governing the evolution of the temperature variations and the free carriers concentration are not modified with regards to the equations of Sec. 2.4, however the state-update equation of the optical mode a is modified in a very similar way to the input/output relationship of the whole system, and according to the following calculations.

We give again in Eq. (6.11) the ordinary differential equations governing the evolution of the optical mode a in our system.

$$\frac{da}{dt}(t) = \left[j(\omega_r + \delta\omega_{nl} - \omega) - \frac{\gamma_{loss}}{2}\right] a(t) + \kappa s_1(t), \tag{6.11}$$

where $\omega_r - \omega$ is the optical detuning, $\delta\omega_{nl}(\Delta T, N)$ is a nonlinear detuning induced by thermo-optic effects and free carrier dispersion, $\gamma_{loss}(a, N)$ the total coefficient loss resulting from absorption losses, coupling losses, and radiation losses. The detailed expression of these parameters can be found in Sec. 2.4 in which we investigate the coupled mode theory of the ring resonators. The other parameters of this equation have already been presented in this section.

From Eqs. (6.4) to (6.7), we can extract the expression of the signal at the point s_1 at time t :

$$s_1(t) = \sqrt{1-\beta^2}\, s_{in}(t) + \beta^2 G a(\lambda)\left[e^{j\phi_c} s_1(t-\tau) + \kappa a(t-\tau)\right]. \tag{6.12}$$

Again, and very similarly to our previous calculations, we can express $s_1(t-\tau)$ as :

$$s_1(t-\tau) = \sqrt{1-\beta^2}\, s_{in}(t-\tau) + \beta^2 G a(\lambda)\left[e^{j\phi_c} s_1(t-2\tau) + \kappa a(t-2\tau)\right]. \tag{6.13}$$

And finally, if we express $s_1(t-k\tau)$ for k in $0, 1, ..., \infty$ in a similar way to Eq. (6.13), we can then combine all these equations with Eq. (6.11) to get the very nice expression of the ordinary differential equations governing the evolution of the optical mode a given in Eq. (6.14) :

$$\begin{aligned} \frac{\mathrm{d}a}{\mathrm{d}t}(t) = &\left[j(\omega_r + \delta\omega_{nl} - \omega) - \frac{\gamma_{loss}}{2}\right] a(t) \\ &+ \kappa \left[\sqrt{1-\beta}\sum_{n=0}^{\infty} G^n \alpha(\lambda)^n \mathrm{e}^{jn\phi_c} \beta^n s_{in}(t-n\tau)\right. \\ &\left. + \kappa \sum_{n=1}^{\infty} G^n \alpha(\lambda)^n \mathrm{e}^{j(n-1)\phi_c} \beta^n a(t-n\tau)\right]. \end{aligned} \tag{6.14}$$

These simplified modelling equations for the nonlinear ring resonator submitted to delayed optical and expressed in the coupled mode theory framework are interesting, because we see that the output power of the whole system is function of an infinite sum of past inputs and past values of the state variable a. Hence, we can carefully set the value of the gain G of the optical amplifier

in order for the information to stay virtually forever in the system. Of course in our simulations, and moreover in an experiment, this conclusion must be restricted due to the nonlinear behaviour of the SOA and the losses in the substrate. Still, from this simplified model naturally has emerged the idea of distributing the virtual nodes on multiple round-trips in the delay line.

6.2.2 Multiple round-trips in the feedback loop

In this section, we specify our strategy to distribute the virtual nodes on multiple round-trips in the small delay line. In particular, we present the time scale τ' corresponding to the total length of the mask, and the corresponding modified masking procedure.

6.2.2.1 Node distribution

As already mentioned before, the idea of distributing the virtual nodes naturally emerges from the model of a ring resonator in the coupled mode theory framework. In particular, the expression of the output signal s_{out} is function of infinite sums of past inputs to the systems $s_{in}(t-k\tau)$ and past states of the reservoir $a(t-k\tau)$.

The N virtual nodes are distributed along multiple round-trips in the delay line. We set the node interdelay θ, and we distribute the N virtual nodes on p round-trips in the delay line of length τ. Hence, the total delay we use to

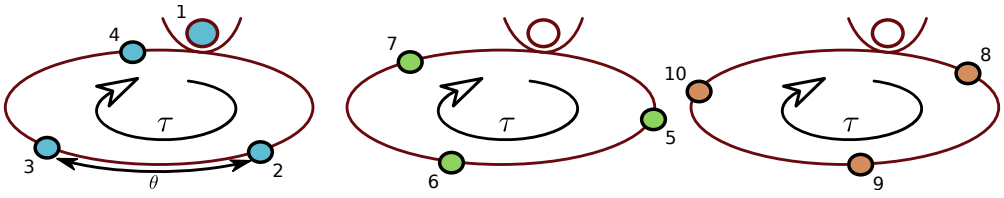

Figure 6.7: Sketch of the distribution of the virtual nodes in the first three round-trips in the delay line. The N nodes are distributed every θ in p round-trips in the delay line of length τ. The total delay we use to distribute the virtual nodes is $\tau' = p \times \tau = N \times \theta$.

distribute the virtual nodes is $\tau' = p \times \tau = N \times \theta$. We show in Fig. 6.7 an example of distribution of the first ten virtual nodes in three round-trips in the delay line of length τ.

6.2.2.2 Modified masking procedure

The masking procedure is modified in order to distribute the virtual nodes on multiple round-trips in the delay line of length τ. Unlike the masking procedure described in Sec. 6.1.2, the length of the mask is not equal to the length of the delay line. The new length of the mask is τ', and is equal to the number of round-trips p times the length τ of the delay line. The total length of the mask is also still equal to the node interdelay θ times the number of nodes N.

We show in Fig. 6.8 an example of the masking procedure used to distribute 33 virtual nodes on ten round trips in a delay line of length $\tau' = 3.3$ ns with an interdelay $\theta = 100$ ps. The colors on top of the node number correspond to the colors in Fig. 6.7 for the distribution of the ten first virtual nodes in three round-trips in the delay line.

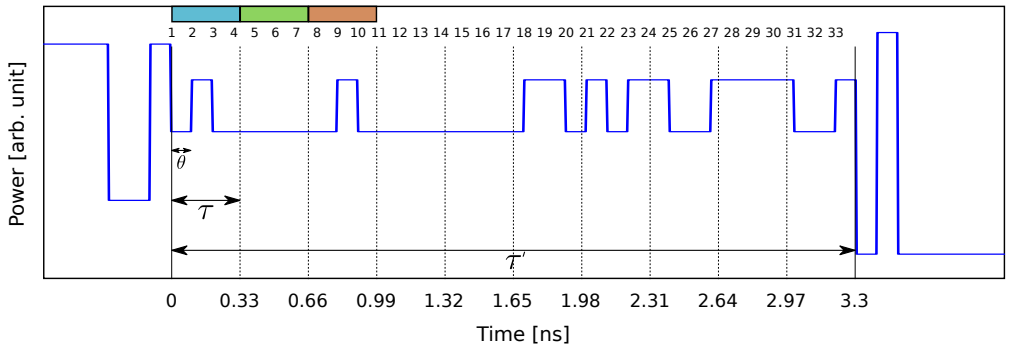

Figure 6.8: Masking procedure on multiple round-trips in the delay line. In this example, the 33 time-multiplexed virtual nodes are distributed along ten round-trips in the delay line. The interdelay between two consecutive nodes is θ, the time of one round-trip in the delay waveguide is τ, and the total length of the mask is $\tau' = 33 \times \theta = 10 \times \tau$. The colors on top of the node number correspond to the colors in Fig. 6.7.

6.3 Numerical simulation strategy

We investigate in this section the numerical simulation strategy used to measure the performance of the system presented in Sec. 6.2. More specifically, we present in Sec. 6.3.1 the tools used to perform the numerical simulations. Sec. 6.3.2 is devoted to the description of the training and testing procedures, and finally Sec. 6.3.3 is dedicated to the search of an optimal operating point.

6.3.1 Simulation of the system

The simulation of the system is done using the *Caphe* photonic circuit simulator [40]. *Caphe* is a realistic on-chip circuit simulator that gives results very close to what can be experimentally obtained [41]. More specifically, we use a typical source term representing the input signal either from a laser source or the coupled light from an optical fiber to the chip. We create a hierarchical structure of the whole system from the typical component models presented in Sec. 2.3 and 2.4. And finally, we use a photodetector to construct the output power based on the Alphalas UPD-15-IR2-FC [38], with a 25 Gb/s bandwidth.

The data stream is created according to the task, and fed to the system with a sampling rate of 200 Giga sample per second.The states of the reservoir are obtained by the simulation using a four order Runge Kutta algorithm of the system using the *Caphe* photonic circuit simulator.

6.3.2 Training and testing procedures

We present in this section the training and testing procedures for the measurement of the performance of our on-chip system used as a time-delayed reservoir computer. More specifically, for each benchmark task we test our reservoir computer on, we describe how we construct the time series, and the procedures for the training and testing of the performance. For this particular design, we measure the memory capacity of the system, as an indicator of the capacity of the system to reconstruct past inputs. Then in order to compare this design with the extended architecture presented in Chap. 4 we asses the level of performance of this reservoir computer on the delayed-XOR task. And finally,

as the memory capacity of this system is sufficiently high, we benchmark our reservoir on the typical and memory-demanding one-step Santa Fe chaotic time series prediction task. The detailed description of these tasks is given in Sec. 3.3.

For the measure of the memory capacity of the system, the input time series is a randomly generated succession of 5,000 samples chosen from a uniform distribution on $]0, 0.5[$. This time series is then masked according to the procedure described in Sec. 6.2.2.2, and the resulting signal is scaled in order for the amplitude modulation of the input signal to be within the $0.5-1.0$ mW power range. We train the system on the first 1,500 samples using a simple linear regression, and we measure the memory capacity using the remaining 3,500 samples.

In the case of the delayed-XOR task, the input is a succession of 20,000 randomly chosen binary bits, masked using the procedure described in Sec. 6.2.2.2, and scaled to be a power modulation between 0.5 and 1.0 mW. The reservoir computer is trained on 16,000 bits using a simple linear regression, and tested on the remaining 4,000 bits, hence the smallest measurable error is 2.5×10^{-4}, and we set any lower BER to this value.

Finally, for the one-step Santa Fe chaotic time series prediction task, we have taken the typical 9,000-samples data set used in the Santa Fe competitions [42], that we have masked using the same process as previously, and scaled down as a power modulation between 0.5 and 1.0 mW. We train the reservoir to predict the time series on the first 8,000 samples using a simple linear regression, and test the performance on the remaining 1,000 samples.

6.3.3 Operating point of the reservoir

In Sec. 4.3, we already highlighted that the search for the optimal injection parameters to operate a reservoir computer is one of the most challenging part of the design of an architecture for reservoir computing applications [43]. Hence, we devote this section to the investigation of optimal operating parameters.

As stated in [44], the fading memory [45] of a system is a very important property for its use as a reservoir computer. A way to maximise the fading

memory of a dynamical system was proposed by [46]. It consists of operating the system in a steady state but also to inject the system such that it maximises the time for the transient dynamics to vanish, therefore maximising the signal mixing in the network for reservoir applications.

In order to inject the data in the system while maximizing the length of the transient dynamics, we can tune three main parameters, namely (i) the optical detuning between the resonance frequency of the ring resonator and the frequency of the injected light, (ii) the injected power range for the modulation of the injected data, and (iii) the bias current of the semiconductor optical amplifier used to improve the quality factor of the whole system. We plot the step response for various injection parameters in Fig. 6.9. The injection parameters for cases (1) to (7) are given in Table. 6.2.

These parameters are only a few amongst the vast exploration of injection parameters, and account for most interesting cases we can find while searching for the operating point. Indeed, for most values of the injection parameters, either the system is oscillating or unstable, or the output power is not sufficient to perform reservoir computing. Hence, in order for the system to be in a linear regime, but still close to instabilities, we remain close to the natural resonance frequency of the ring, and inject a step of power. We have found that the power modulation comprised between 0.5 and 1.0 mW is the most representative, and gives the richest transient.

In Fig. 6.9, we see that for some sets of injection parameters, the system is self pulsing at a very low rate, due to the internal time scales of N and ΔT of the nonlinear ring resonator. For those injection cases (3) and (7), we cannot use this system since a reservoir computer as the system is not stable. However, for the other injection cases, the system is stable when we inject a power step. We therefore focus on cases (1,2,4-6) in Fig. 6.9(b)-(f). We first see that for cases (1) and (5), the transient state is not rich enough compared to the other cases. Finally, among the three last cases, we computed the memory capacity of the system, and saw the case (4) was the injection parameter set giving the best results. Finally, Fig. 6.9(g) shows a zoom of the step response for the best injection parameter set, with an insight of the relevant time scales θ, τ and τ'.

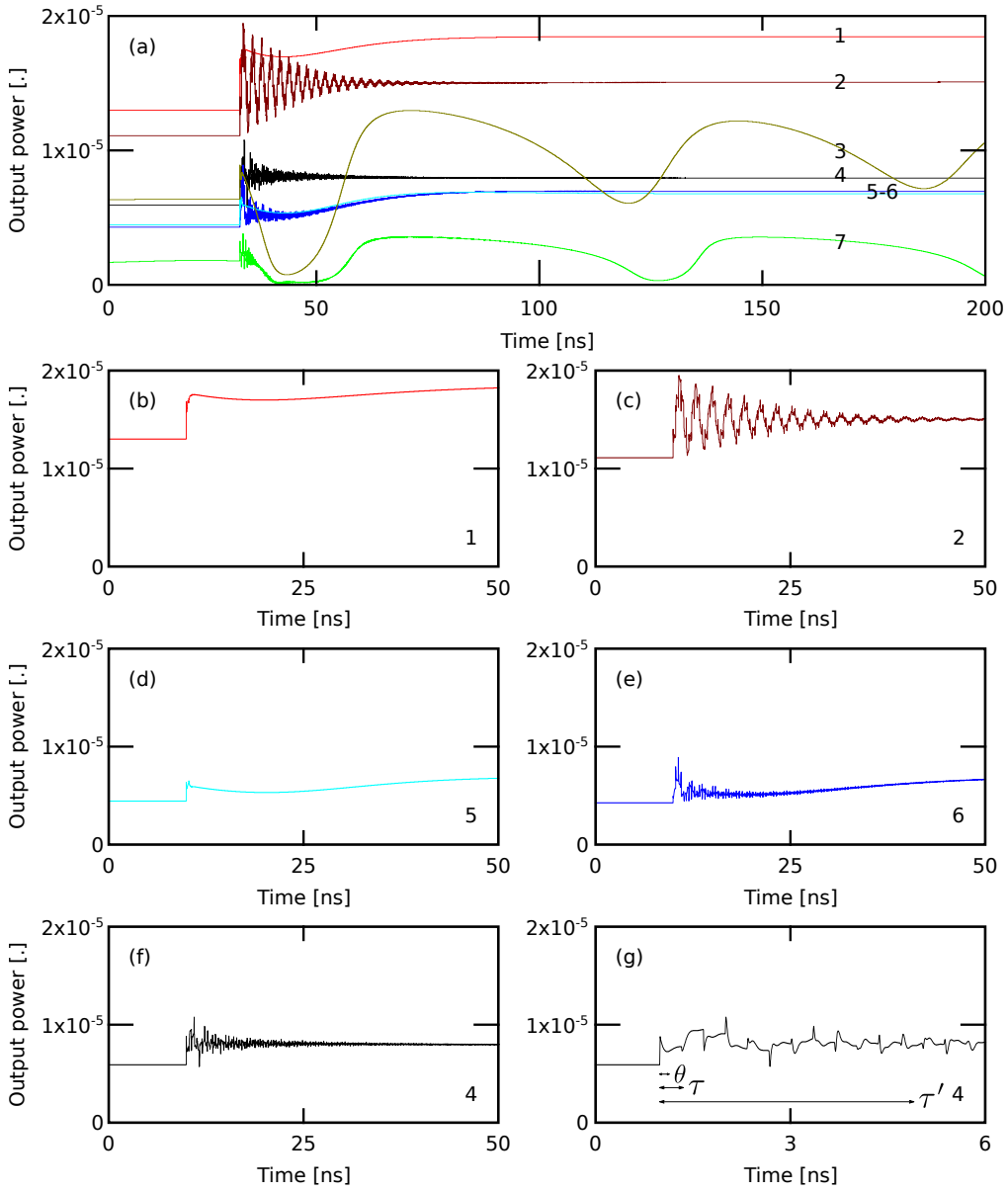

Figure 6.9: step response for various injection parameters. The injection parameters are given in Table. 6.2 for cases (1) to (7). (b) to (f) correspond to zooms of (a) for each set of injection parameters, and (g) is a zoom of (f). We do not show zooms of cases (3) and (7) as the pulsating dynamics does not allow for reservoir computing applications.

Case	$\delta\lambda$ (pm)	P_0 (mW)	P_1 (mW)	I_{bias} (mA)
(1)	-10	0.5	1	50.0
(2)	50	0.5	1	129.5
(3)	10	0.5	1	50
(4)	-10	0.5	1	129.5
(5)	0	0.5	1	50
(6)	0	0.5	1	129.5
(7)	10	0.5	1	129.5

Table 6.2: Injection parameters for cases (1) to (7) of Fig. 6.9.

6.4 Performance of the reservoir

We present in this section the performance of the on-chip delay-based reservoir computer. In Sec. 6.4.1 we discuss on the processing speed of such a delay-based reservoir computer. Then, we measure in Sec. 6.4.2 the memory capacity of the system. And finally in Sec. 6.4.3 and 6.4.4, we present successively the levels of performance of this reservoir computer on the delayed-XOR task and the Santa Fe chaotic time series one-step prediction task. For each task, we compare the results with the reservoir computer made of 16 interconnected nonlinear ring resonators, presented in Chap. 4.

6.4.1 On the processing speed

The delay-based architecture for reservoir computing shows usually good results on a large variety of tasks [10–15]. However, the processing speed that can be attained with this approach is still a drawback of this kind of implementations. Indeed, the processing speed is fixed by the multiplicative inverse of the total length τ' of the mask we use to distribute our virtual nodes in the feedback loop, or in our case in multiple round-trips in the feedback loop.

We take the example of the best case for the Santa Fe task, attained for $N = 56$ virtual nodes with a node interdelay of $\theta = 50$ ps, the processing speed

is given by :

$$Rate = \frac{1}{\tau'} = \frac{1}{N \times \theta} = 0.36 \text{ Giga symbols per second.} \tag{6.15}$$

The data rate we can achieve with this kind of architecture is quite far from the data rate that can be processed by the reservoir computers presented in Chaps. 4 and 5. Actually, the limiting factor to improve the processing speed is not really the chip but the speed limit of our classical measuring equipment.

6.4.2 Memory capacity of the reservoir

We measure in this section the memory capacity of the system, according to the numerical simulation procedure presented in Sec. 6.3. The memory capacity of the system tells us about the ability of the system to reconstruct past inputs, therefore giving us a good insight on the performance we can expect on memory demanding tasks as the Santa Fe chaotic time series one-step prediction task. We mostly analyse the memory capacity when changing the node interdelay θ, that is the time between two consecutive virtual nodes.

We present in Fig. 6.10 a mapping of the memory capacity of a 33-nodes on-chip reservoir computer in the plane (θ, I_{bias}), where I_{bias} is the bias current controlling the gain of the SOA in the feedback loop of the reservoir. We explore in Fig. 6.10(a-b) two strategies for the parametric study of the node interdelay. In Fig. 6.10(a), the design is kept fixed, (in particular we fix the length of the long waveguide), and we distribute the 33 virtual nodes on a variable number of round trips in the feedback loop. In Fig. 6.10(b), we change the length of the long waveguide, and we always distribute the 33 virtual nodes along 10 round-trips in the feedback loop. Note that the 100 ps-interdelay column is the same for the two strategies.

From Fig. 6.10(a-b), we can extract the following information. First, we can attain a memory capacity of about 13 with a 33-nodes reservoir computer for carefully chosen values of node interdelay and bias current of the SOA. Secondly, for any value of node interdelay θ, the maximal bias current of the SOA for which we attain an acceptable value of memory capacity is $I_{bias} = 129.5$

mA. Hence, we will choose this particular value for the following simulations. Finally, if we compare strategies (a) and (b) for the modification of the node interdelay, we see that the strategy (a) - that consists in keeping a fixed length

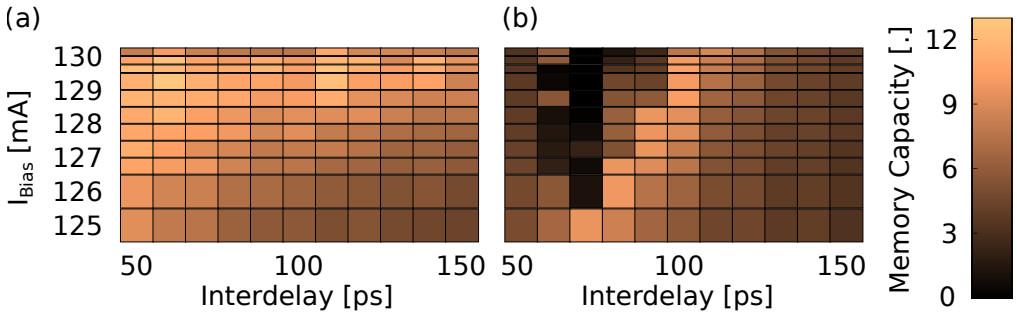

Figure 6.10: Measure of the memory capacity. (a-b) Mapping of the memory capacity of the 33-nodes delay-based reservoir computer in a parametric plane : node interdelay θ and bias current I_{bias} of the SOA. We explore two strategies for the parametric study of the node interdelay. (a) the design is kept fixed, and we distribute the 33 virtual nodes on a variable number of round trips in the feedback loop, and (b) we change the length of the long waveguide, and we always distribute the 33 virtual nodes along 10 round-trips in the feedback loop.

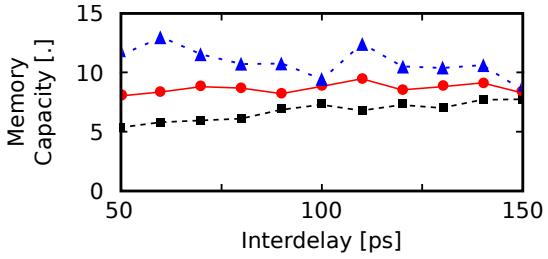

Figure 6.11: Measure of the memory capacity. Comparison of the memory capacity of the parallel 16-node *SWIRL* network from Chap. 4 at 10 Gb/s (black squares with dashed lines), with the memory capacity of the 16-nodes (resp. 33-nodes) delay-based reservoir computer (red dots with plain lines (resp. blue triangles with dashed lines)) when changing the the node interdelay.

of the waveguide, and distributing the virtual nodes on a variable number of round-trips in the delay line - allows for better performance in terms of memory capacity. We will keep this strategy for further sweeps in node interdelay, hence we fix the length of the waveguide to 2.85 cm, inducing a time delay of 330 ps.

We compare in Fig. 6.11 the memory capacity of a 16-nodes delay-based reservoir computer made of one non-linear ring resonator submitted to delayed optical feedback (red dots with plain lines), to the memory capacity of the 16-nodes extended reservoir computer made of 16 nonlinear ring resonators of Chap. 4 at 10 Gb/s (black squares with dashed lines). We see that the memory capacity of the delay-based reservoir is always greater than the memory capacity of the *SWIRL* reservoir. Hence if we compare the extended design with the time-delayed architecture, not only the physical implementation is simpler with the delay-based reservoir computer, but the memory capacity with the same number of nodes is also better.

We also plot in Fig. 6.11 the memory capacity of a 33-nodes delay-based reservoir computer as a function of the node interdelay (blue triangles with dashed lines) without changing the inner design of the reservoir computer. We achieve higher levels of memory capacity compared to the 16-nodes on chip delay-based reservoir computer.

In Fig. 6.12, we plot the memory capacity as a function of the number of

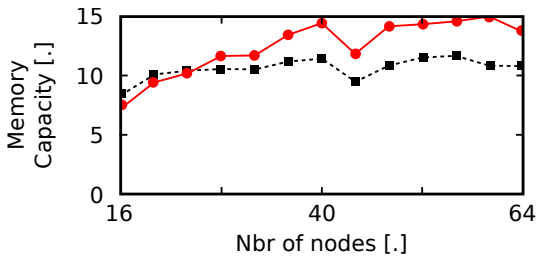

Figure 6.12: Measure of the memory capacity. Memory capacity as a function of the number of nodes for the delay-based reservoir for a node interdelay of 50 ps (resp. 100 ps) in red dots with plain lines (resp. black squares with dashed lines).

nodes (without changing the design of the reservoir computer) for $\theta = 50$ ps (resp. $\theta = 100$ ps) in red dots with plain lines (resp. black squares with dashed lines). The best memory capacity we can attain is 15 for various number of nodes with $\theta = 50$ ps, hence we can expect this system to perform with good levels of performance on memory-demanding tasks. These plots shows that it is possible and very straightforward to scale up the number of virtual nodes without modifying the physical system, and that it allows for better performance, at the cost of a reduced processing speed. Indeed, the processing speed is given by $1/(n_{nodes} \times \theta)$, *i.e.* 500 Mega symbol per second for 40 nodes and $\theta = 50$ ps.

6.4.3 Delayed-XOR task

We present in this section the numerical results on the measure of the level of performance of our on-chip delay-based architecture for reservoir computing on the delayed-XOR task. This task is not usually used to measure the performance of delay-based architectures, as it is not a memory demanding task, and that there are other ways of performing this Boolean task at higher processing speed. However, we test our reservoir on this task in order to compare the performance of this reservoir with the reservoir presented in Chap. 4.

6.4.3.1 Influence of the node interdelay

We give in Fig. 6.13 a comparison of the performance of the parallel 16-node *SWIRL* network from Chap. 4 [4] at 10 Gb/s (black squares with dashed lines), with the performance of the 16-nodes delay-based reservoir computer (red dots with plain lines) when changing the the node interdelay on the 1-bit delayed XOR task. Our reservoir with 16 physical nodes presented in Chap. 4 can perform at the best measurable performance for a large number of interdelay values.

We see that the 16-nodes delay-based reservoir can also perform at the best measurable level of performance on this task for some node interdelay values (for instance, $\theta = 120$ ps, giving a processing speed of 520 Mbits/s), similarly to the 16-nodes physical network. However, in the last plot of Fig. 6.13 (blue

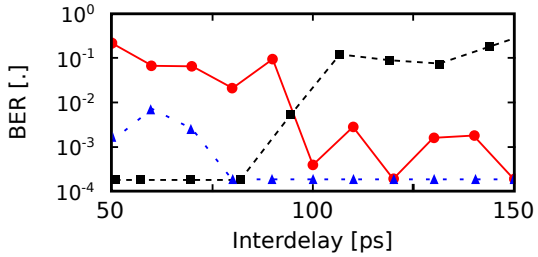

Figure 6.13: Reservoir performance on the delayed XOR task. Comparison of the performance of the parallel 16-node *SWIRL* network from Chap. 4 at 10 Gb/s (black squares with dashed lines), with the performance of the 16-nodes (resp. 33-nodes) delay-based reservoir computer (red dots with plain lines, resp. blue triangles with dashed lines) when changing the the node interdelay on the delayed XOR task.

triangles with dashed lines), we increase the number of nodes in the delay-based reservoir to 33, without modifying the inner design of the reservoir, and we attain this best measurable level of performance on this task for a large range of node interdelay values $\theta \in [80:150]$ ps, giving a processing speed in [202:378] Mbits/s.

We give in Fig. 6.14 an example of time series of the output of the trained

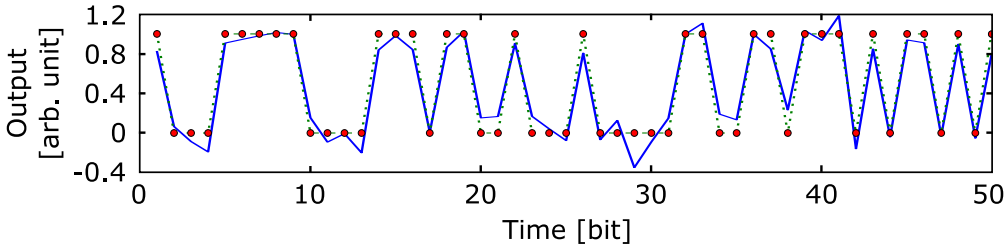

Figure 6.14: Reservoir performance on the delayed XOR task. Example of time trace of the output of the trained 33-nodes delay-based RC on the delayed XOR task. Green dashed line is the target, blue plain line is the actual output of the trained reservoir, and red dots are the thresholded outputs. This figure shows no errors, and the error rate is 2.5×10^{-4}.

33-nodes delay-based reservoir on the 1-bit delayed XOR task. Green dashed line is the target, blue plain line is the actual output of the trained reservoir, and red dots are the thresholded outputs. This figure shows no errors : the error rate reaches 2.5×10^{-4} which the lowest error rate one can compute considering the number of testing bits.

6.4.3.2 2-bits and 3-bits delayed XOR task

Figure 6.15 shows the evolution of the level of performance of the 16-node *SWIRL* network from Chap. 4, the 16-nodes delay-based reservoir computer, and the 33-nodes delay-based reservoir when changing n delay , the memory in the delayed XOR task. In Fig. 6.15, we choose for each reservoir an interdelay value for which it performs at best for $n_{delay} = 1$ (see Fig. 6.13), and we modify n_{delay} . We come to the conclusion that the 16-nodes extended reservoir cannot resolve the delayed XOR task for $n_{delay} > 1$, the 16-nodes delay-based reservoir cannot resolve the delayed XOR task for $n_{delay} > 2$, and the 33-nodes delay-based RC can resolve the delayed XOR task for $n_{delay} = 3$. This figure can be linked to the comparison of memory capacity of Fig. 6.11, and the results are very consistent with the conclusions from this figure: the delay-based reservoir performs better on memory demanding tasks compared to the *SWIRL* network.

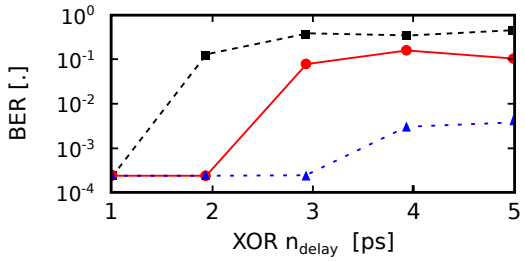

Figure 6.15: Reservoir performance on the delayed XOR task. Comparison of the performance of the parallel RC (black squares with dashed lines), the delay-based RC with 16 nodes (red dots with plain lines), and the delay-based RC with 33 nodes (blue triangles with dashed lines) on the 1-bit delayed XOR task, when changing n_{delay}.

6.4.4 Santa Fe task

We present in this section the performance of our on-chip delay-based reservoir computer on a chaotic time series forecasting. More specifically, we train the reservoir to perform the one-step prediction of the *Santa Fe A* time series presented in Fig. 3.10. Being able to predict chaotic behaviours can be very interesting as it can help to predict financial fluctuations, as financial time series are governed by deterministic chaos [47]. This is why the Santa Fe task is a typical benchmark task to test the performance of artificial neural networks.

6.4.4.1 Influence of the node interdelay

We compare in Fig. 6.16 the performance of the parallel 16-node *SWIRL* network from Chap. 4 and [4] at 10 Gb/s (black squares with dashed lines), with the performance of the 16-nodes delay-based reservoir computer (red dots with plain lines). We see that the two 16-nodes reservoir computers have relatively low levels of performance on this task, with measured NMSE larger than 1×10^{-1}. However, if we train a 33-nodes delay-based reservoir computer to

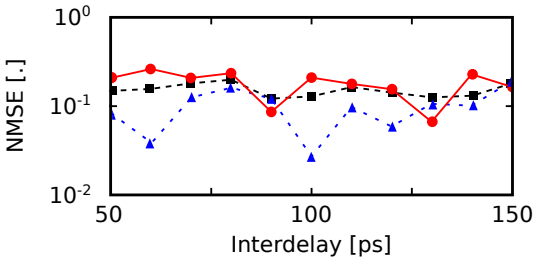

Figure 6.16: Reservoir performance on the one-step Santa Fe time series forecasting task. Comparison of the performance of the parallel 16-node *SWIRL* network presented in Chap. 4 at 10 Gb/s (black squares with dashed lines), with the performance of the 16-nodes (resp. 33-nodes) delay-based reservoir computer (red dots with plain lines, resp. blue triangles with dashed lines) when changing the the node interdelay on the one-step Santa Fe time series forecasting task.

perform the one-step Santa Fe time series forecasting (blue triangles with dashed lines in Fig. 6.16), we see that we manage to attain an acceptable level of performance, with a $NMSE = 1.2 \times 10^{-2}$ for an interdelay value $\theta = 100$ ps.

6.4.4.2 Influence of the number of nodes

We then investigate the performance of the delay-based reservoir computer when increasing the number of nodes without changing the inner design of the system, for interdelay values of 50 ps (red dots with plain lines) and 100 ps (black squares with dashed lines). The results are reported in Fig. 6.17, and show that we successfully reach a NMSE equal to 1.2×10^{-2} for $\theta = 100$ ps and 32 nodes, and we reach a NMSE equal to 1.12×10^{-2} for $\theta = 50$ ps and 56 nodes. The levels of performance of our very compact all-optical on-chip reservoir computer are consistent with state-of-the-art levels of performance on this task by comparable systems [19, 32, 37], but with a lower number of nodes. Usually, the reservoir from Ref. [19, 32, 37] resolve this task for a number of nodes greater than 100, and a NMSE of 1×10^{-3} was numerically obtained in Ref. [14] for 400 nodes.

Finally, we present in Fig. 6.18 an example of time trace of the output of the trained 56-nodes delay-based reservoir on the one-step Santa Fe time series prediction task. Black dashed line is the target, and red dots is the

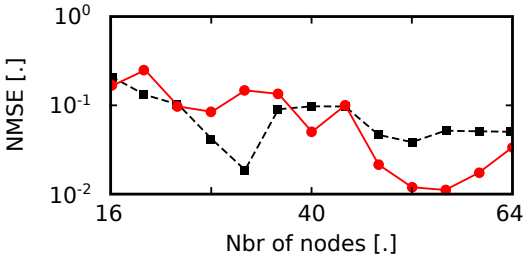

Figure 6.17: Performance on the Santa Fe prediction task as a function of the number of nodes for the delay-based reservoir for a node interdelay of 50 ps (resp. 100 ps) in red dots with plain lines (resp. black squares with dashed lines).

actual output of the trained reservoir. The measured NMSE in this example is 1.12×10^{-2}.

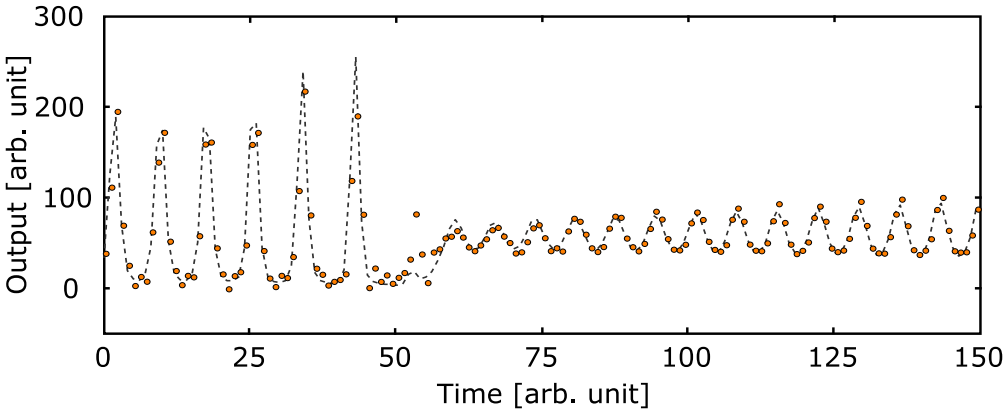

Figure 6.18: Example of time trace of the output of the trained 56-nodes delay-based reservoir on the one-step Santa Fe time series prediction task. Black dashed line is the target, and red dots is the actual output of the trained reservoir. The measured NMSE in this example is 1.12×10^{-2}.

6.5 Conclusion

In this chapter, we have suggested, presented and numerically investigated a new design for all-optical on-chip reservoir computing. This architecture relies on the implementation of a very compact system integrated on a silicon photonic chip, made of a nonlinear ring resonator submitted to delayed optical feedback, used as a time-delay feedback reservoir computer. In this particular design, the nonlinear node is a ring resonator, and virtual nodes are distributed along multiple round-trips in the delay line.

We have studied through intensive numerical simulations the memory capacity of such a system used as a reservoir computer, and compared the performance of this system to our previous design made 16 interconnected nodes in which the nodes were also nonlinear ring resonators, presented in Chap. 4 and in Ref. [4]. More specifically, we have shown that this new design

performs either as good for strongly non-linear tasks or better on memory demanding tasks than our previous architecture for the same number of nodes, however with a more compact and simplified architecture. Moreover, it is possible to scale-up the number of nodes in this design without modifying the inner structure of the reservoir, leading to an out-performance of this architecture on typical memory demanding tasks, for example the one-step Santa Fe time series forecasting task.

We have shown through our simulations that this system can perform at state-of-the-art levels of performance on various benchmark tasks for optimized parameters including the number of nodes, node interdelay, and bias current of the SOA positioned in the feedback loop. In particular, on the non-trivial one-step chaotic Santa Fe time series forecasting, we attain a NMSE equal to 1.12×10^{-2} for 56 nodes separated by a node interdelay $\theta = 50$ ps, distributed along ten round-trips in the delay line, at 0.3 Giga symbols per second. This level of performance can be consistently compared to other optical delay-based reservoir systems that use semiconductor laser with optical or optoelectronic feedback [19, 32, 37] . These other delay-based approaches require typically a larger number of nodes to achieve the same performance, hence lowering the processing speeds (of the order of tens of Mega symbols per second). By comparison, our reservoir computer is a very compact Silicon-based on-chip system, hence allowing for a large scale manufacturing at low cost.

These results open new research venues aiming at compact on-chip implementations of machine learning techniques for all-optical data processing applications. More specifically, our new architecture for reservoir computing allows for a straightforward scalability in terms of number of nodes, without modifying the inner structure of the very compact system. This work could also motivate for an experimental validation of these numerically obtained results.

References

[1] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[2] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[3] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[4] F. Denis-Le Coarer, M. Sciamanna, A. Katumba, M. Freiberger, J. Dambre, P. Bienstman, and D. Rontani, "All-optical reservoir computing on a photonic chip using silicon-based ring resonators," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 24, no. 6, pp. 1–8, 2018.

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

[6] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing," *Optics express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[7] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific Reports*, vol. 2, pp. 1–6, 2012.

[8] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[9] P. Antonik, F. Duport, M. Hermans, A. Smerieri, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer applied to real-time channel equalization," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2686–2698, 2016.

[10] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

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

[12] R. Modeste Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van Der Sande, "Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

[13] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Information processing via physical soft body," *Scientific reports*, vol. 5, p. 10487, 2015.

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

[15] ——, "Experimental reservoir computing using vcsel polarization dynamics," *Optics Express*, vol. 27, no. 13, pp. 18 579–18 584, 2019.

[16] P. Antonik, M. Hermans, F. Duport, M. Haelterman, and S. Massar, "Towards pattern generation and chaotic series prediction with photonic reservoir computers," *Proc. of SPIE Vol*, vol. 9732, pp. 97 320B–1, 2016.

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

[18] H. Zhang, X. Feng, B. Li, Y. Wang, K. Cui, F. Liu, W. Dou, and Y. Huang, "Integrated photonic reservoir computing based on hierarchical time-

multiplexing structure," *Optics express*, vol. 22, no. 25, pp. 31 356–31 370, 2014.

[19] K. Takano, C. Sugano, M. Inubushi, K. Yoshimura, S. Sunada, K. Kanno, and A. Uchida, "Compact reservoir computing with a photonic integrated circuit," *Optics express*, vol. 26, no. 22, pp. 29 424–29 439, 2018.

[20] K. Harkhoe, G. Verschaffelt, A. Katumba, P. Bienstman, and G. V. der Sande, "Demonstrating delay-based reservoir computing using a compact photonic integrated chip," 2019.

[21] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[22] H. Jaeger, *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001, vol. 5.

[23] S. Lepri, G. Giacomelli, A. Politi, and F. Arecchi, "High-dimensional chaos in delayed dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 70, no. 3, pp. 235–249, 1994.

[24] L. Weicker, C.-H. Uy, D. Wolfersberger, and M. Sciamanna, "Mapping of external cavity modes for a laser diode subject to phase-conjugate feedback," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 11, p. 114314, 2017.

[25] G. Bouchez, C.-H. Uy, B. Macias, D. Wolfersberger, and M. Sciamanna, "Wideband chaos from a laser diode with phase-conjugate feedback," *Optics letters*, vol. 44, no. 4, pp. 975–978, 2019.

[26] S. Ortín, M. C. Soriano, L. Pesquera, D. Brunner, D. San-Martín, I. Fischer, C. Mirasso, and J. Gutiérrez, "A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron," *Scientific reports*, vol. 5, p. 14945, 2015.

[27] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir comput-

ing using a time-delay-based architecture: Million words per second classification," *Physical Review X*, vol. 7, no. 1, p. 011015, 2017.

[28] J. Qin, Q. Zhao, H. Yin, Y. Jin, and C. Liu, "Numerical simulation and experiment on optical packet header recognition utilizing reservoir computing based on optoelectronic feedback," *IEEE Photonics Journal*, vol. 9, no. 1, pp. 1–11, 2017.

[29] P. Antonik, F. Duport, A. Smerieri, M. Hermans, M. Haelterman, and S. Massar, "Online training of an opto-electronic reservoir computer," in *International Conference on Neural Information Processing*. Springer, 2015, pp. 233–240.

[30] P. Antonik, M. Gulina, J. Pauwels, and S. Massar, "Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography," *Physical Review E*, vol. 98, no. 1, p. 012215, 2018.

[31] D. Brunner, M. C. Soriano, and I. Fischer, "High-speed optical vector and matrix operations using a semiconductor laser," *IEEE Photonics Technology Letters*, vol. 25, no. 17, pp. 1680–1683, 2013.

[32] K. Hicke, M. A. Escalona-Morán, D. Brunner, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Information processing using transient dynamics of semiconductor lasers subject to delayed feedback," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 4, pp. 1 501 610–1 501 610, 2013.

[33] A. Dejonckheere, F. Duport, A. Smerieri, L. Fang, J.-L. Oudar, M. Haelterman, and S. Massar, "All-optical reservoir computer based on saturation of absorption," *Optics express*, vol. 22, no. 9, pp. 10 868–10 881, 2014.

[34] J. Nakayama, K. Kanno, and A. Uchida, "Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal," *Optics express*, vol. 24, no. 8, pp. 8679–8692, 2016.

[35] J. Bueno, D. Brunner, M. C. Soriano, and I. Fischer, "Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback," *Optics express*, vol. 25, no. 3, pp. 2401–2412, 2017.

[36] Y. Kuriki, J. Nakayama, K. Takano, and A. Uchida, "Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers," *Optics express*, vol. 26, no. 5, pp. 5777–5788, 2018.

[37] L. Appeltant *et al.*, "Reservoir computing based on delay-dynamical systems," *These de Doctorat, Vrije Universiteit Brussel / Universitat de les Illes Balears*, 2012.

[38] F. R. Times, W. Ranges, and W. W. Ranges, "Ultrafast Photodetectors," *Data Sheet*, 2000.

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

[40] http://www.lucedaphotonics.com/, accessed: 2016.

[41] T. Van Vaerenbergh, M. Fiers, P. Mechet, T. Spuesens, R. Kumar, G. Morthier, B. Schrauwen, J. Dambre, and P. Bienstman, "Cascadable excitability in microrings," *Optics express*, vol. 20, no. 18, pp. 20 292–20 308, 2012.

[42] A. S. Weigend, *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.

[43] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega, "Optimal nonlinear information processing capacity in delay-based reservoir computers," *Scientific reports*, vol. 5, p. 12858, 2015.

[44] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, p. 514, 2012.

[45] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[46] T. Yamane, S. Takeda, D. Nakano, G. Tanaka, R. Nakane, S. Nakagawa, and A. Hirose, "Dynamics of reservoir computing at the edge of stability," *International Conference on Neural Information Processing*, pp. 205–212, 2016.

[47] J. Hołyst, M. Żebrowska, and K. Urbanowicz, "Observations of deterministic chaos in financial time series by recurrence plots, can one control chaotic economy?" *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 20, no. 4, pp. 531–535, 2001.

CHAPTER 7

GENERAL CONCLUSION AND PERSPECTIVES

"In literature and in life we ultimately pursue, not conclusions, but beginnings."

— Sam Tanenhaus, *Literature Unbound*

In this chapter, we summarize the work that has been presented in this thesis. We also suggest some perspective of this work for the implementation at the hardware level of reservoir computing techniques for all-optical communications. In particular, we point out the lack of experimental validation of this work, and suggest for future studies to experimentally test the conclusions of this work.

7.1 Summary

This dissertation has focus alternatively on the numerical study of three new architectures for on-chip reservoir computing. This work was done in the context of the European H2020 Phresco project [1, 2], which aim at the design, and the experimental demonstration of an on-chip 64-nodes reservoir computing for telecommunication applications at 32 Gb/s.

The main strategy for this project was to extend the passive structure already studied by our collaborators at the Ghent University for all-optical on-chip reservoir computing based on a small network of 16 linear nodes (small waveguides) interconnected according to the *SWIRL* topology [3–5], leaving the readout layer do the nonlinear transformation through photo-detection of the optical signal. Hence, we suggested to work on the addition of nonlinearity in the intrinsic architecture of the reservoir, that is to replace the linear nodes of [3–5] by nonlinear elements. In this thesis, we have investigated three architectures for on-chip reservoir computing, using the nonlinear ring resonator as primary node.

More specifically, we have suggested in Chap. 4 to replace the 16 linear nodes of [3–5] by 16 nonlinear ring resonators, also interconnected according to the *SWIRL* topology. We have shown numerically that this structure can be used as a reservoir computer, and can perform at state-of-the-art levels of performance on typical tasks, such as the delayed-XOR task at 20 Gb/s, and for a large set of parameters values. We have also connected the intrinsic properties of the building block of the reservoir with the optimum injection parameters for reservoir computing. Moreover, we have shown through extensive simulations that this kind of structure is relatively robust with regards to the fabrication process, in particular for the ring resonance frequencies. However, this architecture made of 16 interconnected nonlinear ring resonators does not clearly outperform the structure using only linear elements as nodes. This is mostly due to the losses in the inner structure of the reservoir, due to the splitters, the combiners, and the long interconnection waveguides.

Thus, as a way to improve the design presented in Chap. 4, and as an extension of the work of [3–5] with passive nodes and the first work introducing the *SWIRL* topology for reservoir computing using SOAs as nodes [6], we have suggested in Chap. 5 to implement an architecture using the *SWIRL* topology with non-identical nodes in the structure of the reservoir. Nodes can be either nonlinear ring resonators, linear elements, or semiconductor optical amplifiers. We have suggested this structure to add gain in the inner structure of the reservoir, in order to compensate for the losses induced by the splitters, combiners, and interconnection waveguides. We have demonstrated through

intensive numerical simulations that this structure can perform well on two typical tasks : the delayed XOR task, and the 3-bit pattern recognition task, at 20Gb/s and 30Gb/s. We have also investigated two input strategies, and have confirmed the work presented by A. Katumba *et al.* [4] on the best injection strategy in a 4×4 *SWIRL* network used as a reservoir computer. However, the numerical results of this chapter do not show a clear improvement of the performance of such a structure in comparison with (i) the work of [4], and (ii) the work presented in Chap. 4. We think that in this structure, the non-linearity imposed by the photo-detection overwrites the non-linearity in the inner structure of the reservoir.

After investigating the two extended structures of Chap. 4 and 5, we have come to the conclusion that the extended *SWIRL* structure can perform well on very simple tasks at very high speed, but is limited in terms of scalability of the number of nodes, mostly due to the losses in the structure. This limit in the number of nodes we can use in the reservoir drastically reduce the scope of possibilities in terms of complexity of the tasks. Hence we have suggested in Chap. 6 an on-chip implementation of the well studied time-delay reservoir computing, using a nonlinear ring resonator as physical nodes, and multiple round-trip in a relatively small delay line to distribute the virtual nodes. This idea of using multiple round-trips in a delay line to distribute the virtual nodes arose from both the mathematical modelling of such a system, that has been investigated in Sec. 6.2, and the work of K. Takano *et al.* in [7]. We have numerically shown in this chapter that it is possible to attain state-of-the-art levels of performance on more complex tasks like chaotic time series prediction, using a very simple system, made only of one nonlinear ring resonator, one splitter, one combiner, one SOA, and one waveguide. In addition to the simplicity of design for this integrated photonic circuit, this structure offers a very straightforward scalability of the number of nodes Indeed, increasing the number of nodes can be done without changing the architecture of the circuit but only in distributing the nodes on a larger number of round-trips in the delay line, inducing the typical drawback of time-delay reservoir computing that is a reduction of the processing speed.

7.2 Perspectives for future work

All-optical on-chip reservoir computing is still a very recent and promising field of research. We suggest in this section a few ideas that could extend this work.

7.2.1 New benchmark tasks and applications

We have restrained ourselves on a limited number of applications, and we have trained our different reservoir architectures to perform on relatively simple tasks, that have no direct applications except for the pattern recognition task for telecommunication applications, and the chaotic time series prediction task for applications in finance or extreme event prediction (natural disasters for instance).

Reservoir computing is a very general concept in the scope of machine learning, and the results obtained throughout this dissertation suggest that we could train our neural network to perform on other, more complex tasks for telecommunication, or classification applications. For instance, our system presented in Chap. 6 can attain a memory capacity of 15, which can be associated with the possibility for the system to reconstruct around 15 bits in the past. With such memory in the system, it could be possible to use such reservoir for nonlinear channel equalization (*i.e* compensation of the signal distortion induced by the communication channel), speech recognition, or other memory demanding tasks.

Another scope of application of machine learning is image classification. It would be an interesting application for this kind of structure to be able to perform well on image recognition tasks, like the widely used handwritten digit database MNIST [8] for instance. The difficulty with this kind of task would be to the data pre-processing and encoding the information in such a way that the reservoir could process it. In particular, we inject information sequentially in the reservoir, and for image recognition, it could be interesting to either inject all the information simultaneously, or to reduce the sequential nature of the injected data through projection of 2D information to 1D information.

Finally, an interesting future work would be to implement online learning

concepts, and to add on-chip weights for all-optical applications. Some studies have been done in this direction by our collaborators at the Ghent University [9] for extended structures with great architectural similarities with the reservoir presented in Chap. 4 and 5. Hence, we can extend this work and apply the conclusions to our structures and expect good results.

7.2.2 Experimental validation

The whole dissertation has addressed the numerical investigation of various architectures for reservoir computing, with a very realistic component library developed by Luceda Photonics [10]. However, for various reasons, it has not been possible to experimentally test these structures in order to validate the numerical results. Due to the accuracy of the library, we can expect the experimental results to be close to our conclusions, but it would still be interesting to get this validation, as experimental and numerical results can easily vary due to the importance of the noise levels, mostly at the detection but also when we have amplification, and thermal variations.

References

[1] www.phresco.eu, accessed: 2019.

[2] https://cordis.europa.eu/project/rcn/198823/factsheet/en, accessed: 2019.

[3] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, 2014.

[4] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cognitive Computation*, pp. 1–8, 2017.

[5] A. Katumba, J. Heyvaert, B. Schneider, S. Uvin, J. Dambre, and P. Bienstman, "Low-loss photonic reservoir computing with multimode photonic integrated circuits," *Scientific reports*, vol. 8, no. 1, p. 2653, 2018.

[6] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[7] K. Takano, C. Sugano, M. Inubushi, K. Yoshimura, S. Sunada, K. Kanno, and A. Uchida, "Compact reservoir computing with a photonic integrated circuit," *Optics express*, vol. 26, no. 22, pp. 29 424–29 439, 2018.

[8] https://en.wikipedia.org/wiki/MNIST_database, accessed: 2019.

[9] M. Freiberger, A. Katumba, P. Bienstman, and J. Dambre, "On-chip passive photonic reservoir computing with integrated optical readout," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2017, pp. 1–4.

[10] http://www.lucedaphotonics.com/, accessed: 2016.