A Multiple-Input Strategy to Efficient Integrated Photonic Reservoir Computing

Andrew Katumba · Matthias Freiberger · Peter Bienstman · Joni Dambre

R

Abstract

Introduction

Photonic reservoir computing has evolved into a viable contender for the next generation of analog computing platforms as industry looks beyond standard transistor-based computing architectures. Integrated photonics reservoir computing, particularly on the Silicon-on-Insulator platform, presents a CMOS-compatible, wide-bandwidth, parallel platform for implementation of optical reservoirs. A number of demonstrations of the applicability of this platform for processing optical telecommunications signals have been made in the recent past. In this work, we take it a stage further by performing an architectural search for designs that yield the best performance while maintaining power efficiency.

Methods

We present numerical simulations for an optical circuit model of a 16 node integrated photonic reservoir with the input signal injected in combinations of 2, 4, and 8 nodes, or into all 16 nodes. The reservoir is composed of a network of passive photonic integrated circuit components with the required nonlinearity introduced at the readout point with a photodetector.

The resulting error performance on the temporal XOR task for these multiple input cases is compared with that of the typical case of input to a single node. We additionally introduce for the first time in our simulations a realistic model of a photodetector. Based on this, we carry out a full power-level exploration for each of the above input strategies.

Results and Conclusions

Multiple-input reservoirs achieve better performance and power efficiency than single-input reservoirs. For the same input power level, multiple-input reservoirs yield lower error rates. The best multiple-input reservoir designs can achieve the error rates of single-input ones with at least 2 orders of magnitude less total input power. These results can be generally attributed to the increase in richness of the reservoir dynamics and the fact that signals stay longer within the reservoir.

If we account for all loss and noise contributions, the minimum input power for error free performance for the optimal design is found to be in the $\approx$ 1mW range.

Keywords Reservoir Computing · Integrated Photonics · Photonic Reservoir Computing · Reservoir Architectures

1 Introduction

The persistent increase in demand for systems that can process the massive amounts of data available today has strained the currently employed transistor-based von Neumann architectures. Simultaneously, the growing demand for high-throughput, high-fidelity telecommunications systems has generated significant implementation hurdles for the associated signal processing systems.

To address the compounding challenges for these computation and communication systems, a major design revolution is underway for the next generations of these systems in the IT research world. The frantic search for potential solutions has initiated a revisit to analog computation platforms but with the aim of combining them with the state-of-the-art in large-scale integration technology. These platforms exploit the inherent dynamics of certain physical systems for processing and/or computing. Of these, prominently under consideration are biologically inspired techniques, and particularly brain-inspired computing approaches that employ artificial structures that mimic the brain's neural computational semantics.

Reservoir computing (RC) is a brain-inspired computing approach that initially emerged as a way around the intricacies associated with correctly training recurrent neural networks [1–3]. Classical software RC involves setting up a large randomly initialized nonlinear dynamical system (*the reservoir*) – usually an artificial neural network – that is tuned into a specific dynamical regime to allow for the following three conditions: separability of the inputs, generation of similar outputs for similar inputs and some form of finite memory of the previous inputs. Under these circumstances, the states of the reservoir can be linearly combined, following task-imposed optimization criteria, to extract the desired outputs for the specified inputs.

Beyond the initial software implementations, RC has evolved into a way to enable computing with physical nonlinear dynamical systems. Examples of the concept applied to mechanical systems, memristive systems, atomic switch networks, boolean logic elements and photonic systems can be found in [4–8]. Photonic RC particularly presents a number of benefits compared to e.g. electronics, as it offers a large bandwidth and is inherently massively parallel.

To date, experimental demonstrations of photonic reservoirs routinely achieve state of the art performance on various information processing tasks. Implementations based on a single nonlinear node with a delayed feedback architecture have proven that photonic RC is competitive for analog information processing [9–17]. Moreover, integrated photonic reservoirs can push computation speeds even higher for digital information processing. The performance of integrated photonic reservoirs has been studied numerically for networks of ring resonators [18–22], networks of SOAs [7], and experimentally with networks of delay lines and splitters in [23]. Integrated photonic reservoirs are particularly compelling, especially when implemented in the CMOS platform as they can take advantage of its associated benefits for technology reuse and mass production.

A recent development in the design of RC systems is the realization that for certain tasks that are not strongly nonlinear, it is possible to achieve state-of-the-art performance using a completely passive linear network, i.e., one without amplification or nonlinear elements. The required nonlinearity is introduced at the readout point, typically with a photodetector [23]. The work discussed in this paper is also based on this architecture. Aside from the integrated implementation introduced in [23], the passive architecture has been adapted to the single node with delayed feedback architecture in form of a coherently driven passive cavity [9].

With regards to general task suitability, photonic RC is particularly beneficial when the signals to be processed are already in the optical domain. This is for example true for tasks oriented towards fiber-optic based telecommunication systems as is the case for bit-sequence processing tasks such as logical temporal XOR, AND, OR; header recognition; and equalization. For these scenarios, the reservoir manipulates the light signals directly without the need for any extra electrical-optical and/or optical-electrical conversions. This setup could lead to processing speed-ups and overall reduction in system complexity. Furthermore, without the extra EO conversions, as is the case with passive reservoirs, there is a potential power consumption advantage since the computation itself does not require external energy.

Aside from performance characterizations, full adoption of an RC scheme for a particular application requires a study of the power efficiency benefits of such a deployment. The most complete energy efficiency calculation for an optical reservoir can be found in [10] for a fully nonlinear reservoir based on a laser with feedback. The authors reported a power consumption of 10mJ per bit for the speech processing task. In [9], a minimum input power of 0.57mW at the input is reported for the coherently driven passive cavity reservoir with a fiber loop. Our analysis shows that the total input power requirements of the optimal multiple-input reservoir is also the $\approx$ 1mW regime. However, a full determination of the power requirements is strictly tied to the implementation substrate, and there is no straightforward way to make a one-to-one comparison between the different realizations.

While the majority of our recent work on passive integrated photonic RC focused on single-input reservoirs, our previous paper on passive integrated photonics [23] already introduced the idea that it may be beneficial to inject multiple copies of the input signal into the reservoir. However, only a very specific case

of presenting the input to all nodes with different random phases is discussed. The work presented here is a detailed investigation of the impact of the choice of the number and configuration of the input nodes on the robustness of the reservoir. Equally important, we introduce in our numerical simulations a photodetector model at each readout node that takes into account bandwidth limitations, as well as optical and electrical noise properties encountered in real-world detectors. With this model in place, we are able to examine for the first time the impact of the input power level on the performance and make conclusions about the energy efficiency of various reservoir designs.

2 Methods

2.1 Passive Integrated Photonic Reservoir Computing

The integrated photonic reservoirs typically studied in the past are limited to planar architectures in a bid to minimize crossings that manifest as a source of signal cross-talk and extra losses. This constrains the design space from which reservoir configurations can be chosen. The swirl reservoir architecture, as is used in this work, was introduced in [18] as a way to satisfy planarity constraints while allowing for a reasonable mixing of the input signals. A 16-node photonic swirl reservoir is shown in Figure 1. Passive integrated photonic reservoir computing is a special form of photonic reservoir computing that consists of a linear network of passive photonic integrated circuit (PIC) components with the required nonlinearity typically provided by the readout system (an optical nonlinearity is also an alternative). In current passive photonic RC implementations, the photodetector, required to convert the complex-valued reservoir states to real-valued intensities, suitably serves this purpose [23].

2.2 Reservoir Model

The reservoir state update equation is given as:

$$\boldsymbol{x}[k+1] = \boldsymbol{W}_{res}\boldsymbol{x}[k] + \boldsymbol{w}_{in}(\boldsymbol{u}[k+1] + u_{bias}) \tag{1}$$

where $\boldsymbol{u}$ is the input to the reservoir and u_{bias} is a fixed scalar bias applied to the inputs of the reservoir. For an N-node reservoir, $\boldsymbol{W_{res}}$ is an $N \times N$ matrix representing the interconnections between reservoir components taking into account splitting ratios and losses, with phases drawn from a random uniform distribution on $[-\pi, \pi]$, $U(-\pi, \pi)$. $\boldsymbol{w}_{in}$ is an N-dimensional column vector whose elements are nonzero for each active input node. The input weights are similarly chosen from $U(-\pi, \pi)$.

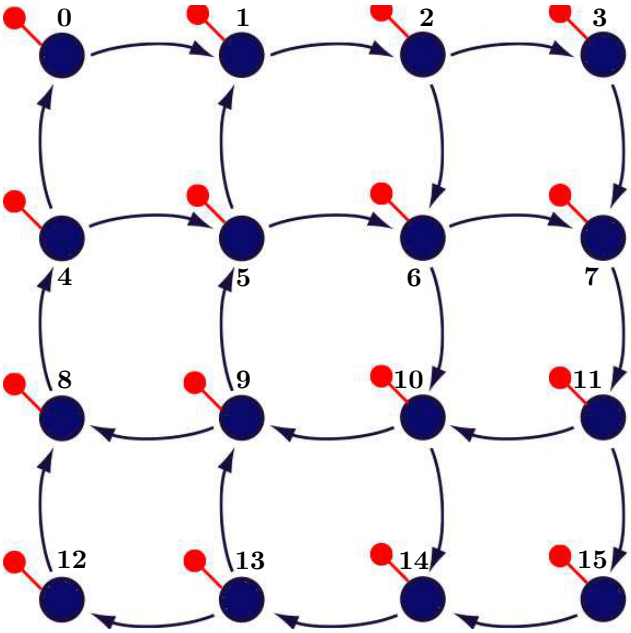

Fig. 1 16-node swirl reservoir schematic. From here on, nodes will be referenced following the labels displayed here. In this particular implementation, the nodes are the locations at which states are appropriately combined and split. They also serve as input and detection points.

All our previous work on integrated photonic reservoir computing has assumed perfect reconstruction of the states at the readout nodes. The absolute square value of the reservoir states (electric field values) was used as the input for the machine learning model. In this work, we introduce a detector model that takes into account the responsivity, as well as various noise contributions and the response-time limitation encountered in real photodetectors. The total noise σ_n^2 of the photodetector has shot noise and thermal noise contributions as follows:

$$\sigma_n^2 = 2qB(\langle I \rangle + \langle I_d \rangle) + 4k_BTB/R_L \tag{2}$$

where B is the bandwidth of the detector, $\langle I \rangle$ is the photocurrent, I_d is the dark current, q is the elementary particle charge charge, k_B is Boltzmann's constant, R_L is the load impedance and T is the temperature (in K).

The first part of equation 2 represents shot noise terms due to the input signal and the dark current, while the last part is the thermal noise contribution due to the detector load resistor. The bandwidth limitation of the detector is approximated by a low-pass filter with 3dB cutoff corresponding to the detector bandwidth.

The output from the reservoir is then given as:

$$\boldsymbol{y}_{out} = \boldsymbol{W_{out}}\boldsymbol{x_{pd}} \tag{3}$$

where $\boldsymbol{W}_{out}$ are the linear output (readout) weights to be determined through training with ridge regression, and $\boldsymbol{x_{pd}}$ are the reservoir states after the photodetector.

Introducing this model for the detector dictates that we pay extra attention to the receiver power levels and in general the overall power budget of our systems, to prefer designs that not only yield acceptable performance, but are also energy efficient.

2.3 Single-Input RC

The most obvious way to get the signal into the planar integrated photonic reservoir is to inject it at a single node, for example with a fiber grating coupler, and allow it to propagate throughout the network. This reservoir design paradigm is attractive due its straightforward implementation and the fact that it does not require the use of crossings. The states for the machine learning phase are obtained by reading out each input-output node combination. The single-input passive reservoir has been shown to reach state-of-the-art performance for speech signal processing and bit-sequence processing tasks [23, 7]. With the same strategy, we have more recently demonstrated signal equalization for metro links [24].

2.4 Multiple-Input RC

While the reservoir architecture in section 2.3 is amenable to the bit-level tasks outlined above, it suffers from major drawbacks due to the inherent limitations of an integrated photonics platform. Particularly, the losses increase with the size of the architecture. This work therefore seeks to look at how such an architecture could be extended to simultaneously achieve power efficiency and performance benefits. To this end, we study architectures that seek to support these ideals. We compare the performance of an architecture with the same size as in [23], with the same total input power injected into the reservoir but distributed over different nodes. The experimental section will show that even when the same power is injected into the reservoir, the increased variation between the reservoir states contributes considerably to the computing power of the architecture.

3 Simulation Results and Analysis

The reservoir states are obtained as per equation 1 by propagating the inputs through a photonic reservoir model implemented in Caphe photonics circuit simulator [25]. The photodetector used in the simulations is modeled based on the Alphalas UPD-15-IR2-FC photodector [26] that is available in our lab. The specific parameters used are a bandwidth of 25 GHz, a responsivity of 0.5 A/W (a pessimistic value as the datasheet value is 0.75 A/W), a dark current of 0.1 nA and a Noise Equivalent Power (NEP) of $1 \times 10^{-15}\,\mathrm{W}/\sqrt{\mathrm{Hz}}$. This NEP corresponds to an average signal power of 1.6 nW at an SNR of 10. It should be mentioned that the ultimate minimum power at the reservoir input will be set by the requirements of the downstream processing electronics.

In this work, each considered combination of reservoir initialization and input configuration was tasked to solve the delayed XOR task. The current output bit for this task is the XOR of the current input bit with one n_{delay} bits in the past. Here we express it as:

$$y[n] = x[n] \oplus x[n - n_{delay}], \tag{4}$$

where $x[n]$ is the bit-level representation of the input data stream and $y[n]$ is the bit-level representation of the output. Before injection into the reservoir, the inputs ($x[n]$) are converted from logical levels to discrete sampled data by upsampling and pulse shaping steps.

This task was considered as it is the most difficult of all delayed binary tasks involving only two bits. This is the case because, in machine learning terms, XOR is not linearly separable (see for example [27]).

For all considered input cases, the 4x4 (16 node) reservoir architecture was used to generate the states. This number of nodes was chosen as it is a design that is both cost-effective to produce with multi-project wafer runs, but also has a good performance on a number of tasks. In all cases, the length of the interconnections between the reservoir translates to a propagation time of 62.5 ps, matching the current generation of available chips.

Once the states were obtained and transformed with the detector model, the readout was trained with a combination of the Oger machine learning toolbox [28] and the scikit-learn library [29].

3.1 Simulation Methods

We feed 10000 randomly chosen bits into the reservoir and use the resulting states for training with 5-fold cross validation to optimise the design parameters and yet another 10000 for testing. We use regularized ridge regression to train the linear readout. Testing is done on the best case resulting from the cross-validation. All reported error rates are relate to the test data. With 10000 bits for testing, error rates are reported at a confidence level of about 90% [30].

3.1.1 Data Rate Studies

For the cases of single-input and multiple-input reservoirs, we studied the error rate of the reservoir across multiple data rates. To match the limitations of currently available measurement equipment in our lab, we restrict the maximal data rate to 32 Gbps. The data stream is a NRK OOK modulated signal, which for simulation purposes is over-sampled 24 times to achieve sufficient simulation accuracy.

For a fair comparison between the different cases, the same aggregate input power accross all input nodes was used: 100 mW. Where the input was fed into more than one node, the power was equally divided between the nodes. Results are reported as averages accross 30 different random initialisations of the input weights and reservoir waveguide phases (each using different randomly generated bit streams.

For plotting and interpreting the results, we make use of the the reservoir interdelay parameter r_{id}, which is defined as:

$$r_{id} = \frac{\tau_{bit}}{\tau_{id}}, \tag{5}$$

where τ_{bit} is the bit duration for the given data rate and τ_{id} is the interconnection delay time, corresponding to the the time it takes signals to propagate between reservoir nodes. The reservoir inter-delay parameter can be directly interpreted as the number of times the bit duration fits into the reservoir interconnection delay and can be used to identify under which regime the current computation is being carried out.

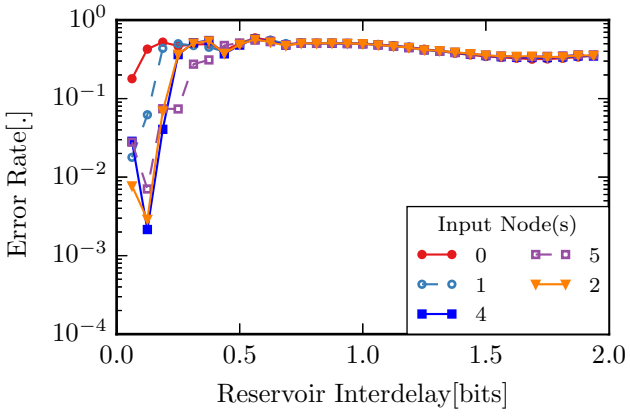

Fig. 2 Error rate vs. reservoir interdelay for various nodes for the input to single node case. The minimum acceptable error rate is 10^-3

For the single-input simulations, we chose a representative sample of the available nodes as dictated by the symmetry of the swirl architecture relative to the central loop. The error rates for different reservoir interdelays are given in Figure 2 for input to nodes 0, 1, 2, 4 and 5. The results show the typical single sharp minimum that translates into the reservoir only being able to process signals at a single data rate. We can also conclude that proximity of the node to the central loop (nodes 5, 6, 9 and 10) is important for realizing low error rates on the task. Nodes 0 and 1, which are furthest away from the central loop, have the worst error performance while 4, 2, and 5, which inject either directly into the central loop or are only one hop away, yield the best performance.

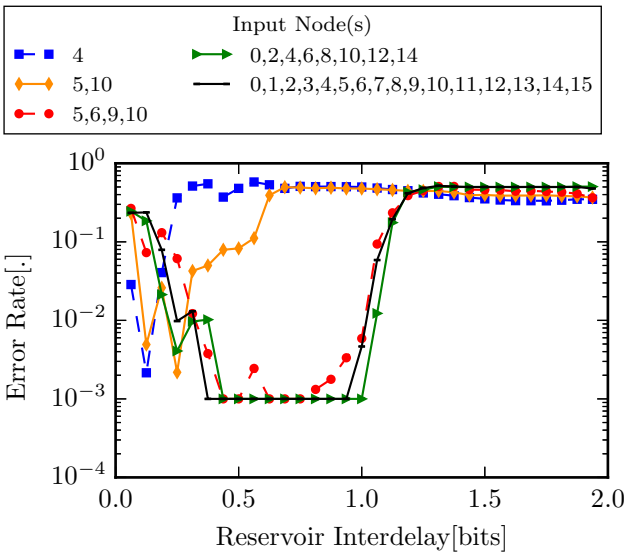

Fig. 3 Error rate vs. reservoir interdelay for the different injection strategies. Minimum acceptable error rate is 10^-3

For the multiple-input reservoir case, we consider input configurations involving simultaneous injection of the input bit stream into: 2 nodes, 4 nodes, 8 nodes or all 16 nodes of the reservoir. The input node combinations with best error rates in each of the groupings are plotted together in Figure 3. From the plot, we observe that in general the multiple-input reservoirs perform better than their single-input counterparts. As more reservoir nodes are driven, we discern the emergence of an increasingly wider basin in which the error is at or below the measurable minimum (10^{-3} in this case). The all-input case provides the widest basin. A wide basin implies more flexible architectures that can operate at multiple data rates. To change the data rate of operation, one simply has to re-train the reservoir readout for that data rate.

We further checked the influence of moving to multiple input reservoir configurations on the computational power of the reservoir, more specifically its memory.

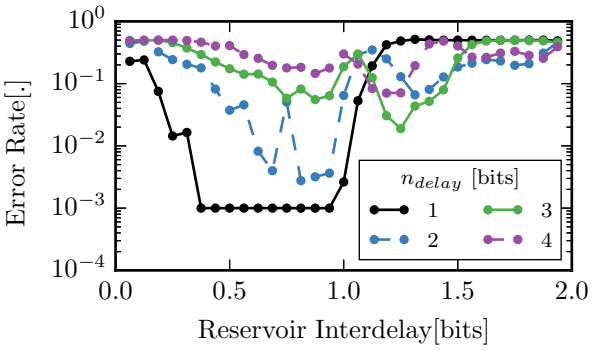

Fig. 4 Error rate vs reservoir interdelay for the input to all nodes case. n_{delay} specifies the separation,in number of bits, of the two bits used for the XOR computation.

Here we present Figure 4 which depicts the error rates corresponding to the single-input versus the all-input case for various values of n_{delay}. In the plots, a larger n_{delay} corresponds to a task that requires more memory. For example for the temporal XOR task, this simply means the current output bit is the XOR of the current input bit with a bit much further back in time.

For the single input case, no error rates below 0.1 can be obtained for $n_{delay} > 1$. Even though for multiple inputs reservoirs the performance similarly deteriorates with increasing n_{delay}, it is clear that they can be operated for longer values of n_{delay}. This is because the useful signal (with a level significantly above the noise floor) remains present in the reservoir for a longer time.

3.1.2 Power level analysis

A key design guideline for signal processing systems for fiber-optic telecommunications systems is to keep the energy consumption as low as possible. In all our previous works, simulations assumed idealized detection of the reservoir states at each detection point for the readout nodes. In this work, on top of the search for the lowest error rate and robust reservoir designs, we now also look at how power efficiency maps to the different choices.

The data rates for the power sweeps were chosen at the minima of the error rate versus reservoir interdelay sweep curves (like the ones in Figure 3). The simulations were repeated 10 times for each reservoir design with different initializations.

Figure 5 shows averaged error rates plotted against total input power.

We observe that as we increase the number of the input nodes, the minimum power requirements for error-free performance also go down. The most significant jump in power efficiency is an approximately 2 orders

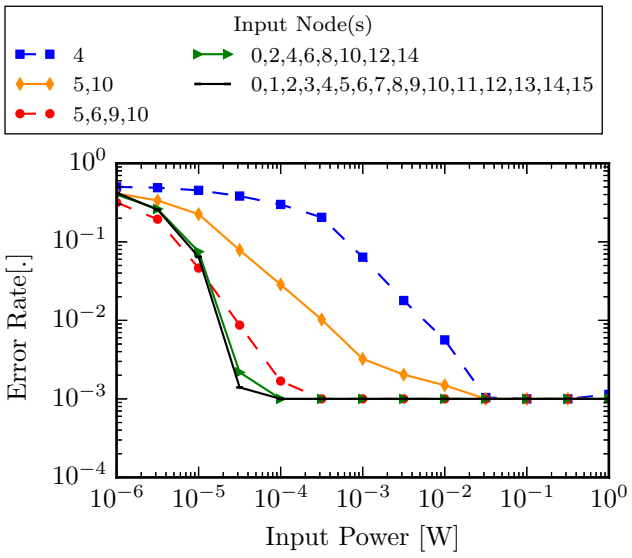

Fig. 5 Error rate vs total input power for different injection scenarios. The minimum measurable error, given the number of bits used for testing, is 10^{-3}.

of magnitude decrease for the best 4-input node combination as compared to the 1 or 2 node input combinations. This can be attributed to the fact that the [5, 6, 9, 10] combination is the central loop in the swirl architecture which allows for significant signal distribution for a small number of inputs. We also observe that increasing the number of input nodes beyond 4 does not significantly impact the power efficiency. Since each input that needs to be driven incurs an additional hardware cost, we can conclude that driving the central four nodes is the most cost- and power-efficient solution.

Looking in more detail at what happens inside the reservoir, Figures 6, 7 and 8 show the average power intensity in all reservoir nodes for the cases of single-node input, input to the central loop and input to all nodes respectively. For the single-node input case, the power decreases significantly within a few hops from the driving node. As an example, node 8, which is just below node 4, has more than 10 dB less power than node 4. When all nodes are driven, the power is most evenly distributed across all the nodes. This scenario also corresponds to the best power efficiency (3 orders of magnitude higher than the best single input case) obtained in our simulations. With the power injected in the central loop nodes only, the power efficiency lies between the two extreme cases. In this instance, there is still a significant subset of the reservoir nodes with similar power levels and only the furthest nodes exhibit a power drop of more than 5dB compared to the input nodes.

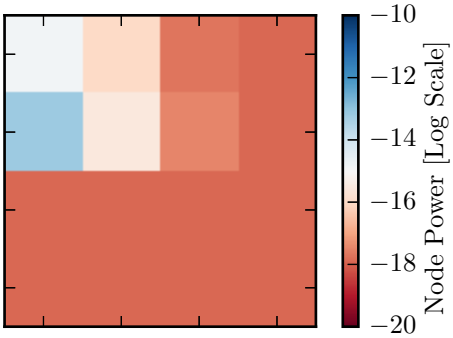

Fig. 6 Average power distribution over the reservoir nodes for input to node 4.

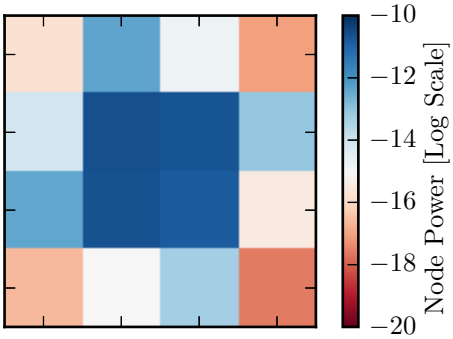

Fig. 7 Average power distribution over the reservoir nodes for input to the central loop .

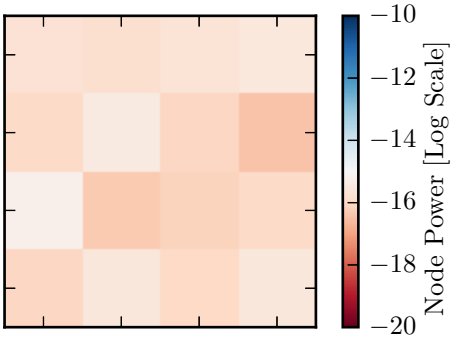

Fig. 8 Average power distribution over the reservoir nodes for input to to all nodes.

3.1.3 Discussion for optimal design

Simulation results from Sections 3.1.1 and 3.1.2 above indicate that injection of power into the central nodes of the reservoir, [5, 6, 9, 10], provides the best combination of performance and energy efficiency.

Figures 9 and 10 illustrate the bounds of the errors for the results within 1 standard deviation of average over the repetitions for error rate studies and power level studies respectively. Unsurprisingly, the transition

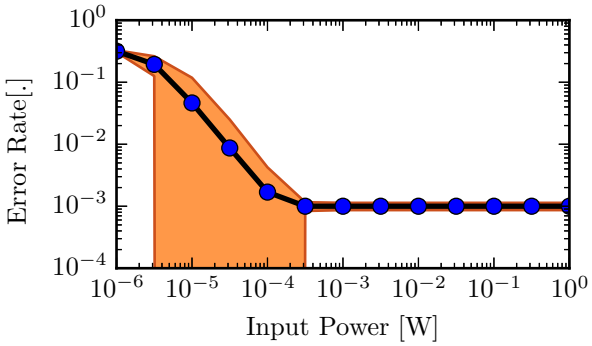

Fig. 9 Error rate vs total input power for input to the central swirl loop (nodes [5, 6, 9, 10]). The solid line indicates the mean value over all repetitions while the shaded areas indicate the error bounds within 1 standard deviation of the mean.

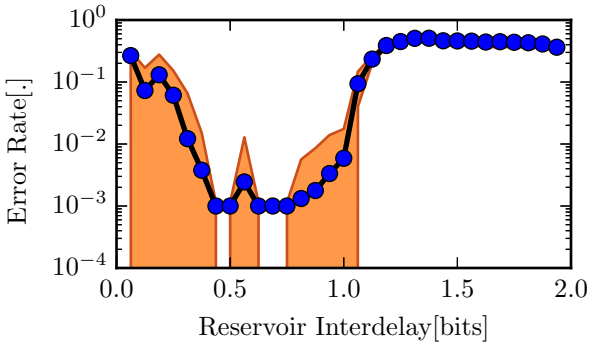

Fig. 10 Error rate vs reservoir interdelay for input to the central swirl loop (nodes [5, 6, 9, 10]). The solid line indicates the mean value over all repetitions while the shaded areas indicate the error bounds within 1 standard deviation of the mean.

regions between the zones of best performance and those with the highest error rates have the highest uncertainty. The width of these regions can be shrunk by, for example, considering a larger number of bits in the test dataset. Concerning the minimum input power for this design, and since the voltage required for the subsequent machine learning electronics is on the order of a few mV, the equivalent power at the input of the reservoir is on the order of a few mW.

3.1.4 Summary

The multiple-input case performs better in terms of error rate and power efficiency. For the error rate performance results, it can be argued that having power injected at multiple locations increases the number of possible mixing combinations of the signals. This mixing is important for computation as there is a richer

signal from which the machine learning algorithm can extract useful features.

Another equally important aspect is that with the multiple input case a much lower power budget suffices to reach the same performance. This is because the power is more evenly spread out throughout the reservoir which is crucial to the correct recovery of the reservoir states as it ensures that the signal is sufficiently higher than the noise at for all readout nodes.

4 Conclusions

We have presented an architectural exploration for planar, passive integrated photonic reservoir computing systems. Error rates obtained from circuit simulations of reservoir designs with various input configurations establish that multiple-input reservoirs perform better than single-input reservoirs for a larger number of data rates. The varied mixing between the multiple copies of the input signals with different phases translates into increased computational power of the reservoir.

Additionally, reservoirs with multiple inputs allow a more even power distribution landscape. This creates a larger usable richness in the reservoir since more signals with roughly similar amplitudes are mixed. Moreover, multiple input-designs present a better power efficiency and so present better odds for correct extraction of all reservoir states, since there are more nodes that have power that is higher than the noise floor. An added benefit is that with more input points, the signal tends to stick around longer in the reservoir which increases the reservoir memory.

However, driving more nodes comes at an additional hardware cost, because the optical signals need to be distributed to all nodes. Since most of the improvement in robustness and power efficiency is obtained by driving the four central nodes instead of just one, we consider this to be the most promising and cost effective solution for small reservoirs. In its current state, this optimal design requires a few mW of input power. We are currently investigating ways of bringing this value down, for example, by reducing the the internal losses in the reservoir, or by using more compact architectures in which losses do not scale directly with reservoir sizes.

In future work, we will explore how to use such a 16 node reservoir as a tile to create larger reservoirs. This way, the lessons learned from this work's architectural exploration exercises will drive the design of the next generation of reservoir computing chips to tackle faster, more complex optical telecommunications signal processing applications.

Acknowledgements This research was funded by the EU Horizon 2020 PHRESCO Grant (Grant No 688579), and the BELSPO IAP P7-35 program Photonics@be.

Compliance with Ethical Standards

Conflict of Interest

Andrew Katumba, Matthias Freiberger, Peter Bienstman and Joni Dambre declare that they have no conflict of interest.

Informed Consent

Informed consent was not required as no human or animals were involved.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Maass W, Natschläger T, Markram H. Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural computation. 2002;2560:2531–2560.
2. Jaeger H, Haas H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science (New York, NY). 2004;304:78–80.
3. Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D. An experimental unification of reservoir computing methods. Neural Networks. 2007 4;20(3):391–403.
4. Hauser H, Ijspeert A, Füchslin R, Pfeifer R, Maass W. Towards a theoretical foundation for morphological computation with compliant bodies. Biological Cybernetics. 2011 12;105(2011):355–370.
5. Sillin HO, Aguilera R, Shieh HH, Avizienis AV, Aono M, Stieg AZ, et al. A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. Nanotechnology. 2013;24:384004.
6. Kulkarni MS, Teuscher C. Memristor-based reservoir computing. In: Proceedings of the 2012 IEEE/ACM International Symposium on Nanoscale Architectures - NANOARCH '12. New York, New York, USA: ACM Press; 2012. p. 226–232.
7. Vandoorne K. Photonic reservoir computing with a network of coupled semiconductor optical amplifiers; 2011.
8. Paquot Y, Duport F, Smerieri A, Dambre J, Schrauwen B, Haelterman M, et al. Optoelectronic Reservoir Computing. Scientific Reports. 2012 2;2:287.
9. Vinckier Q, Duport F, Smerieri A, Vandoorne K, Bienstman P, Haelterman M, et al. High-performance photonic reservoir computer based on a coherently driven passive cavity. Optica. 2015;2(5):438–446.
10. Brunner D, Soriano MC, Mirasso CR, Fischer I. Parallel photonic information processing at gigabyte per second data rates using transient states. Nature communications. 2013 1;4:1364.

11. Appeltant L, Soriano MC, Van der Sande G, Danckaert J, Massar S, Dambre J, et al. Information processing using a single dynamical node as complex system. Nature communications. 2011 9;2:468.
12. Larger L, Soriano MC, Brunner D, Appeltant L, Gutierrez JM, Pesquera L, et al. Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. Optics Express. 2012 1;20(3):3241.
13. Duport F, Schneider B, Smerieri A, Haelterman M, Massar S. All-optical reservoir computing. Optics Express. 2012 9;20(20):22783.
14. Dejonckheere A, Duport F, Smerieri A, Fang L, Oudar JL, Haelterman M, et al. All-optical reservoir computer based on saturation of absorption. Optics Express. 2014 5;22(9):10868.
15. Soriano MC, Ortín S, Brunner D, Larger L, Mirasso CR, Fischer I, et al. Optoelectronic reservoir computing: tackling noise-induced performance degradation. Optics Express. 2013 1;21(1):12.
16. Nguimdo RM, Verschaffelt G, Danckaert J, Van der Sande G. Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics. Optics Express. 2014 4;22(7):8672.
17. Hicke K, Escalona-Morán M, Brunner D, Soriano MC, Fischer I, Mirasso CR. Information Processing Using Transient Dynamics of Semiconductor Lasers Subject to Delayed Feedback. IEEE Journal of Selected Topics in Quantum Electronics. 2013 7;19(4):1501610–1501610.
18. Vandoorne K, Dambre J, Verstraeten D, Schrauwen B, Bienstman P. Parallel reservoir computing using optical amplifiers. IEEE transactions on neural networks. 2011 9;22(9):1469–81.
19. Mesaritakis C, Papataxiarhis V, Syvridis D. Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system. JOSA B. 2013;(October).
20. Fiers MAA, Van Vaerenbergh T, Wyffels F, Verstraeten D, Schrauwen B, Dambre J, et al. Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns. IEEE Transactions on Neural Networks and Learning Systems. 2014;25(2):344–355.
21. Zhang H, Feng X, Li B, Wang Y, Cui K, Liu F, et al. Integrated photonic reservoir computing based on hierarchical time-multiplexing structure. Opt Express. 2014 12;22(25):31356–31370.
22. Mesaritakis C, Kapsalis A, Syvridis D. All-Optical Reservoir Computing system based on InGaAsP Ring Resonators for High-Speed Identification and Optical Routing in Optical Networks. 2015;9370:1–7.
23. Vandoorne K, Mechet P, Van Vaerenbergh T, Fiers M, Morthier G, Verstraeten D, et al. Experimental demonstration of reservoir computing on a silicon photonics chip. Nature communications. 2014 1;5:3541.
24. Katumba A, Bienstman P, Dambre J. Photonic Reservoir Computing Approaches to Nanoscale Computation. In: Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication. NANOCOM' 15. New York, NY, USA: ACM; 2015. p. 23:1–23:2.
25. Fiers M, Vaerenbergh TV, Caluwaerts K, Ginste DV, Schrauwen B, Dambre J, et al. Time-domain and frequency-domain modeling of nonlinear optical components at the circuit-level using a node-based approach. J Opt Soc Am B. 2012 5;29(5):896–900.
26. Ultrafast InGaAs PIN photodetector UPD-15-IR2-FC. ALPHALAS GmbH; 2012.
27. Theodoridis S, Koutroumbas K. 04 Nonlinear Classifiers. Pattern Recognition (Fourth Edition). 2009;p. 151–260.
28. Verstraeten D, Schrauwen B, Dieleman S, Brakel P, Buteneers P, Pecevski D. Oger: Modular Learning Architectures For Large-Scale Sequential Processing. Journal of Machine Learning Research;.
29. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in {P}ython. Journal of Machine Learning Research. 2011;12:2825–2830.
30. Jeruchim M. Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems. IEEE Journal on Selected Areas in Communications. 1984 1;2(1):153–170.