Using digital masks to enhance the bandwidth tolerance and improve performance in on-chip reservoir computing systems

Bendix Schneider, Joni Dambre, *Member, IEEE* and Peter Bienstman, *Member, IEEE*

***Abstract*— Reservoir Computing (RC) is a computing scheme related to recurrent neural network theory. As a model for neural activity in the brain it attracts a lot of attention, especially because of its very simple training method. However, building a functional, on-chip, photonic implementation of RC remains a challenge. Scaling delay lines down from optical fibre scale to chip scale results in RC systems that compute faster, but at the same time require that the input signals are also scaled up in speed, which might be impractical or expensive. In this paper, we show that this problem can be alleviated by a masked RC system in which the amplitude of the input signal is modulated by a binary-valued mask. For a speech recognition task we demonstrate that the necessary input sample rate can be a factor of 40 smaller than in a conventional RC system. Additionally, we also show that linear discriminant analysis and input matrix optimisation is a well-performing alternative to linear regression for reservoir training.**

***Index Terms*—Photonic reservoir computing, Optical neural network, Supervised learning, Photonic integrated circuits**

I. Introduction

Reservoir Computing (RC) emerged as a machine learning concept a little more than a decade ago. In their seminal papers [1–3] on Echo State Networks and Liquid State Machines respectively, H. Jaeger and W. Maas introduced the basics of RC as a method that combines the strength of recurrent neural networks with the ease of a linear readout as the only part that is trained in a supervised manner. Research efforts have been made ever since, both towards applications and towards a better theoretical understanding [4–7]. Powerful RC systems have since been designed and applied successfully to tasks such as robot control, speech recognition or channel equalisation [2], [8–10]. The field of RC has also opened new perspectives on analogue computing, leading to the first hardware reservoir implementations, using hybrid analogue-digital technologies [11–13]. In this work the neural network computation is performed by analogue components, but the preprocessing and learning stages are still performed offline on conventional computers. A special role is attributed to all-optical RC set-ups due to their inherent high bandwidth which allows for a tremendous increase in computing speed. A network of integrated SOAs (semiconductor optical amplifiers) [14], [15], a single non-linear node in a delayed feedback loop [16], and a semiconductor laser diode subject to delayed self-feedback [17] have been proposed among others.

The on-chip photonic integration of a reservoir offers the advantage of miniaturisation, cost-effectiveness and the possibility for mass fabrication. However, it can only perform useful computation if its internal timescales match the timescale of the input signal at hand. It is desirable to preserve the inherently high bandwidth that is offered by the photonic implementation. Yet, this choice compels timescale compression for many real-time applications, e.g. speech recognition, and implies the use of high-speed, analogue waveform generators as prerequisites for RC. Given that these are very costly, leading-edge technology for working frequencies of modern optical networks, we endeavour to enlarge the range of input signal bandwidths supported by the integrated reservoir system.

In this article, we will show that this can be achieved by means of an earlier introduced data-processing scheme, called ‘masking’ [16], in which the input signal is multiplied by a much faster periodic signal called the mask. It can be shown that this essentially realises a non-trivial mapping of a large recurrent neural network onto a smaller one and is accomplished by serialising the computations, i.e. rather than unfolding calculations in space in a large network, the calculations are unfolded in time in a smaller network. In the extreme case, this even renders reservoir computing with a single non-linear node possible as has been demonstrated in [16]. Because of this serialisation the single node system emulates a much bigger network which greatly enhances its computational power. Our goal is not only to translate the masking approach from a single-node network to an all-optical, multi-node SOA network, but also to adapt the technique to relax the input signal’s bandwidth requirements. This masking scheme is not only restricted to integrated photonic devices but is equally pertaining to extended optical system solutions opening a wider range of applications.

The article is structured as follows. We review the concept of RC and explain the masking in more detail in Section II. Section III describes the way the reservoir is modeled and how it is used to numerically solve a speech recognition task which consists in classifying digits. In Section IV we discuss the main simulation results and compare it to previous research that was conducted using the same task.

This work is supported by the Interuniversity Attraction Pole IAP 7-35, Photonics@be, of the Belgian Federal Science Policy Office (belspo) and the ERC starting grant NARESCO.

B. Schneider, and P. Bienstman are with the Photonics Research Group (INTEC), Ghent University–imec, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium, and with the Center for Nano- and Biophotonics (NB-Photonics), Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium (e-mail: bendix.schneider@intec.ugent.be; peter.bienstman@ugent.be).

J. Dambre is with the Electronics and Information Systems (ELIS), Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium (email: joni.dambre@ugent.be).

II. Review of the Reservoir Computing and masking concept

In 2001, Jaeger et al. [1] published a paper proposing a novel way of training recurrent neural networks in an easy manner. Recurrent neural networks are known for their excellent performance with respect to language modeling, real-time computing, and handwritten character recognition, but they are difficult to train [19–21].

RC copes with that problem by leaving the recurrent weights of the network untrained. Indeed, a random set of weights in a reservoir of large size provides a large variety of complex oscillator dynamics in different node subsets. If this variety is rich enough to represent the target signal faithfully by linearly combining the outputs from the reservoir nodes, RC equips us with a simple but powerful machine learning method.

Interference effects and complex-valued interconnections enrich the reservoir behaviour [15], [22] in a coherent optical reservoir. It is often convenient to use the optical power at the readout, which is a non-linear mapping of the node state variables and hence a non-linear effect by itself.

A quite recent breakthrough in experimental RC is based on the masking concept [16] where a periodic mask – a set of weights – modulates the input signal. We now briefly explain how this mechanism works in an experimental setup (Fig.1) for the binary-valued mask and which conditions must be satisfied for useful computations. The successive steps involved in the preprocessing and masking scheme are part of Fig. 1 (green and blue dotted-dashed box). A more stringent mathematical notation of all the subsequent steps is summarised in Table I.

First of all, during a preprocessing step, the discrete-time input signal *s*, which can be multi-dimensional, is projected onto the reservoir's state space via a projection matrix $W_{inp2res}$. The reservoir consists of 16 coupled SOAs (reservoir nodes), so the state space spanned by the respective SOA integrated gain variables is 16-dimensional. An affine transformation is applied to the resulting input vector $\underline{s}_{16}$ to ensure that all its components are positive-definite functions of time. This is a necessary step as we encode inputs to the reservoir as optical power values. The preprocessing stage leads to an effective input vector u which varies with time in discrete time steps defined by the signal sample time T_{sample}, which we allow to range from 20 ps to 10 ns. Each of the 16 components of the effective input vector u is now running through the masking stage as indicated by the blue dotted-dashed box of Fig. 1.

At the start of the masking stage every input vector component is converted to continuous-time with the help of a digital-to-analogue converter (DAC) implemented as a zero-order hold or a non-ideal DAC followed by a sample-and-hold circuit. The resulting piecewise-constant signal can be thought of as output by a fast arbitrary waveform generator and is used for direct modulation of a laser diode. During each hold segment the laser light intensity is constant too. The critical part of the masking procedure consists of the periodic amplitude modulation of those piecewise-constant segments by the mask signal, in which the mask period is equal to the sampling time, i.e. $T_{sample} = T_{mask}$. Another important fact is the piecewise-constant nature of the periodic mask signal itself. Indeed, one period of the mask signal is composed of N non-return-to-zero bits. Consequently, the mask signal changes at a N-times higher rate than the input signal sampling rate. As a consequence of the masking procedure, i.e. the multiplication of two piecewise-constant signals (the effective input vector component and the periodic mask), the final, masked version of the effective input vector component presents itself again as a piecewise-constant function. Time bins of constant light intensity and duration T_{sample}/N occur naturally and are termed 'virtual nodes' [16] in accordance to the RC literature. As shown in Fig. 1, any masked input signal has a fast component, switching it on and off (for a binary mask signal only) at a freqeuncy N/T_{sample}, and a slower component where the masked signal's amplitude is changed. Eventually each of the 16 masked components of the effective input vector u is injected into a different reservoir node (SOA). We included a low-pass filter effect at this point in order to take into account the physical upper bandwidth limits of modulator and DAC. In principle it is possible to choose a different periodic bit sequence as a mask for each component of u. In all our simulations, however, we decided to use a common mask signal that is applied to all the components of u. This reduces the otherwise exponentially growing parameter space of different mask signal combinations and avoids an excess of time-consuming simulations.

TABLE I
MATHEMATICAL DESCRIPTION OF THE SIGNAL TRANSFORMATIONS APPEARING IN THE PIPELINE OF FIGURE 1

Type of signal transformation	Mathematical Description
Projection	$s_{16}\colon \mathbb{R}^n \times \mathbb{Z} \to \mathbb{R}^{16} \times \mathbb{Z}$ $s[n] \mapsto W_{inp2res} s[n]$
Affine transformation (peak signal amplitude s_0)	$u\colon \mathbb{R}^{16} \times \mathbb{Z} \to \mathbb{R}^{16} \times \mathbb{Z}$ $s_{16}[n] \mapsto s_0 \cdot \left(s_{16}[n] - \min_{(i,n)} s_{16,i}[n]\right)$
Sample&hold (DAC with zero-order hold)	$u_{zoh}(t)\colon u[n] \mapsto \sum_n u[n] \cdot rect\left(\frac{t}{T_{sample}} - \frac{1}{2} - n\right)$
Masking (e.g. bit = (1,1,0,1,0) with N=5)	$u_{mask}\colon u_{zoh}(t) \mapsto u_{zoh}(t) \cdot m(t)$ $m(t) = \sum_k \sum_{n=0}^{N-1} bit_n \cdot rect\left(\frac{t}{T_{sample}/N} - \frac{1}{2} - n - Nk\right)$
Low-pass filter of time constant τ ($\tau = 4T_{sample}$)	$u_{mask}(t) \mapsto \mathcal{L}^{-1}\{H(s)\mathcal{L}(s)\{u_{mask}\}\}; H(s) = \frac{1}{1+s\tau}$
Demasking	$s_{SOA}\colon \mathbb{C}^{16} \times \mathbb{R} \to \mathbb{C}^{16xN} \times \mathbb{Z}$ $s_{SOA,m(N-1)+i}[n] \mapsto \delta_{T_{sample}(i+0.5+Nn)/N}[s_{SOA,m}(t)]$
Detection (PD) and centroids	$s_{PD}\colon \mathbb{C}^{16xN} \times \mathbb{Z} \to \mathbb{R}^{16xN}$ $s_{PD}[n] \mapsto \frac{1}{N_{frame}} \sum_{n=0}^{N_{frame}-1} \lvert s_{PD}[n] \rvert^2$

Virtual nodes arise in the context of 'masking'. They are not physical in the sense that they form well-defined components of the reservoir, but are of dynamical nature. They are the artificial result of a serialised input; the well-defined constant power levels play the role of an increased signal state space upon which the non-linearity of the physical reservoir node acts. The virtual nodes undergo an update rule similar to the neurons in a formal recurrent network, and the coupling among them is a consequence of the finite relaxation time of the latent (physical) node variable [16]. This sets an upper bound on the duration of each virtual node, since coupling only occurs if is less than the relaxation time of the physical variables that determine the state space. In contrast, choosing the piecewise-constant sections too short leads to increasing inertial effects and the reservoir system is unable to follow the fast input changes. These bounds on the virtual node duration constitute the aforementioned constraints on the masking mechanism. For the case of a single non-linear node with delayed feedback comprising 400 virtual nodes [16], the authors found an optimal relaxation time-to-virtual node duration ratio of 1:5 which we will also adopt as a starting point.

The masking procedure is always matched with a 'demasking' procedure at the readout stage. For each masking period, it extracts the N virtual nodes and offers them in parallel (i.e. at the same time) to the linear readout function. This is a crucial step for the subsequent application of the machine learning algorithm which operates on the enlarged node set and maps it back to a discrete-time output signal. Experimentally, the demasking procedure can be designed and carried out as shown schematically in the orange, dotted-dashed box of Fig. 1. At every reservoir node, a small portion of the SOA output light power is branched out and used to implement an N-fold readout function. There are as many virtual nodes per masking period as there are bits in the masking bit sequence. The branched optical signal is fed through an 1xN splitter and forked into N distinct delay lines of increasing length. Each length increment in the delay lines corresponds to a readout delay of one virtual node duration. Therefore the demasking procedure mimics a deinterleaver that parallelises the sequential order of appearance of the virtual nodes in one masking cycle. If the signals at the individual photodetectors are gated using the clock output of the sample-and-hold unit as a trigger, a lower-speed implementation of the photodetector array is conceivable which makes the output units available for training at once. Alternatively, one could use a single higher-speed photo detector for readout of each reservoir node, sample at a rate N/T_{sample}, and perform the parallelisation electronically.

Thus the masking-demasking pair can be understood as an efficient mapping between a given input space and an abstract, increased, implicit feature space that eventually boosts the reservoir system's computational power. The combined system of masking-demasking and RC forms a building block for non-linear, adaptive filters.

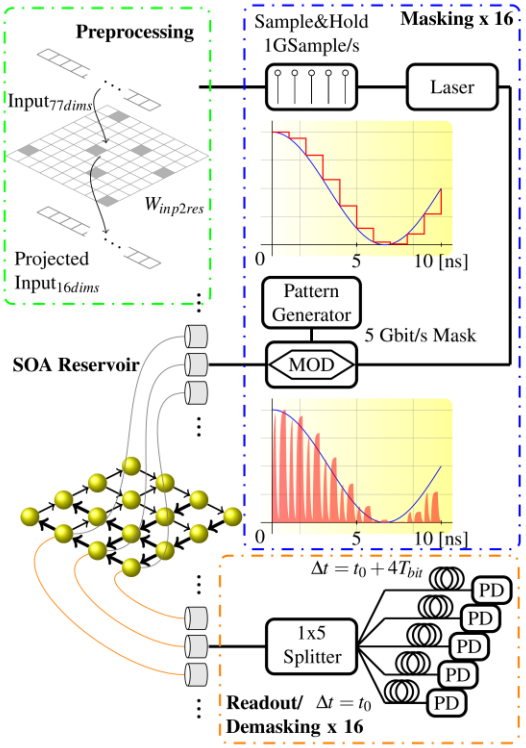

Fig. 1. Simulation flow diagram: preprocessing step (green box, offline) - the input dimensions of 77 are matched to the reservoir dimension of 16 using a projection matrix; masking step (blue box) using a sample & hold circuit and a modulator driven by a bit pattern generator which multiplies a fast binary mask onto a slower input signal; signal processing by the SOA reservoir; readout and demasking stage (orange box) where the signal is temporally deinterleaved and detected. For clarity only the connections of 3 reservoir nodes are shown.

Whether a masked RC system is suitable for a certain task depends on how many virtual nodes it employs. One is thus eager to design a tunable delay line as interconnect between two physical nodes, which is amenable to host a varying number of virtual nodes and which scales linearly with the network size. This was done experimentally for the RC based on single-node dynamics in an electronic set-up [16]. However, the operational bandwidth therein is limited by the node's intrinsic timescale of ~ 10 ms. In this paper, we try to compensate for the deleterious bandwidth effects and shift to an integrated photonic platform, using the faster semiconductor's gain dynamics (~ 100 ps) as source of non-linearity. Unfortunately, this transition to integrated, all-optical reservoirs has to face two major difficulties that are not present in the electronic set-up. Firstly, integrated optical delay lines are lossy waveguides that cannot be tuned easily over a wide range. Especially when hundreds of virtual nodes are needed, this solution is quickly loss-limited. Secondly, the benefit of high intrinsic bandwidth in an all-optical reservoir system (~ 10 GHz) poses stringent requirements on the speed of the analogue, effective input vector u which might stem from a lower speed physical process (e.g. speech) or from readily available low-speed analogue waveform generators (~ 100 MHz) and thus introduces several orders of timescale compression. Hereafter, we show that the first difficulty is overcome by a RC device with more than one physical node –

in our case a swirl network comprising 16 integrated SOAs that is easily scalable due to its integrated solution. Furthermore, we demonstrate numerically that the masking approach still allows one to use lower-speed input signals when modulating them with higher-speed binary masks.

III. Software implementation

A. *Reservoir and integration method*

We have simulated the spoken digit task (recognising the spoken digits 'zero' through 'nine'), a standard benchmark task in the RC community, for a 4-by-4 network of SOAs connected according to a swirl topology (see Fig. 2). Each of the SOAs obeys the state space rate equation indicated in (1), where the dimensionless variables h represent the integrated SOA gain and G_{ss} (13.2 dB) the integrated small signal gain [23]. The incoming light field is denoted A_{in} and its square magnitude is written in units of the SOA saturation power P_{sat} (21.1 mW). There is one free parameter, α, which depends on the ratio between delay and SOA carrier lifetime (0.3 ns) under the assumption that all the photonic wire connections have equal length. We therefore refer to time in units of the interconnection delay whenever the reservoir response is solved numerically. The nodes influence each other via the incoming light waves which are modeled as a coherent superposition of the outgoing waves of all the neighbouring nodes plus the external driving signal. Due to the presence of splitters, combiners and the waveguide interconnects, losses as well as phase shifts will be imparted to the outgoing waves. Hence, we model the update rule for the incoming light fields according to (2). The complex coefficient $\gamma_{i,j}$ describes the attenuation and the phase shift experienced by the wave travelling from the j-th node to the i-th node, and ß (a typical value of which is 5) is the linewidth enhancement factor. The outgoing wave is related to the incoming wave as stated in (3a, 3b), t_{SOA} being the group delay inside each amplifier.

To decrease the large parameter space and to save computational resources, we decided to leave the reservoir size unchanged and chose the same SOA settings as in [15], [22].

$$\dot{h} = -\alpha h - \alpha(G_{ss}e^{h} - 1)|A_{in}(t)|^2 \quad (1)$$

$$\boldsymbol{A}_{in}(t+1) = \hat{\boldsymbol{K}}\boldsymbol{A}_{in}(t) + \boldsymbol{A}_{ext}(t+1) \quad (2)$$

$$\hat{\boldsymbol{K}}_{i,j} = \gamma_{i,j}\sqrt{G_{ss}}e^{\frac{1}{2}(1-i\beta)h_j(t)} \quad (3a)$$

$$A_{out,i}(t) = A_{in,i}(t - t_{SOA})\sqrt{G_{ss}}e^{\frac{1}{2}(1-i\beta)h_i(t)} \quad (3b)$$

We solve the problem stated above, (1)-(3), which can be rewritten as a delay differential equation by the method of steps: Equation (1) is integrated on the interval t = [0,1] given the input history $\boldsymbol{s}(t) = 0, t \leq 0$, the field amplitudes are updated according to (2) and used as the new input history for the interval t = [1,2] as we go on. This procedure is repeated until the solution is constructed for the complete interval of interest, t = [0,t_{final}]. A commercial MATLAB™ ode23 solver was used to carry out the integration (relative accuracy 10^{-4}) in which all signals are interpolated to the required order for dense output.

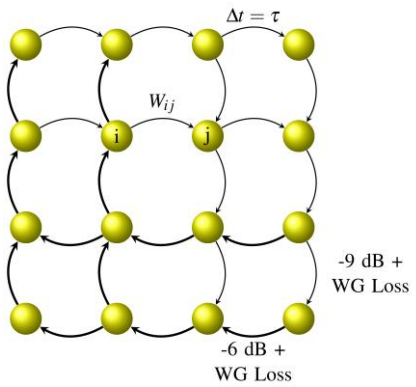

Fig. 2. Reservoir exhibiting a swirl topology. The nodes consist of identical SOAs and are represented as circles. Each arrow corresponds to a waveguide wire connecting node i and j, and confers a unit delay Δt to the signal. Thinner arrows impart an additional loss of 3 dB to the connections as compared to thicker ones.

B. *Data set and preprocessing*

The data set consists of the TI 46 corpus [14] wherein 500 spoken digits were selected (five speakers uttering the digits 'zero' through 'nine' ten times). In order to make the classification task harder, 3 dB of babble noise was added from the NOISEX-92 database [24]. The noise-corrupted speech signal is then down-sampled by a factor of 128 and subsequently preprocessed by the Lyon Ear Model [15]. This model emulates the response of the human cochlea and outputs 77 predictive features. These features are combined into a 16-dimensional input vector via the reservoir's input weight matrix, with weights drawn independently from the discrete set {-1, 1}. Given that we encode our signal as optical power values, we have to apply a bias term to ensure positive values. We also rescale each sample drawn from the data set so that the overall peak power equals the input scaling.

As mentioned in the introduction, in contrast to the standard approach to RC where the weighted and rescaled signal is directly fed to the numerous reservoir nodes, we add another preprocessing step up front. It consists of modulating the input streams with a periodic NRZ (non-return-to-zero) bit sequence of length N common to all the reservoir nodes (referred to as the shared digital mask, see also Fig. 1).

The digital masking is carried out by chirp-free amplitude modulation. We approximate the time continuity of the signal by upsampling the input by a factor of 16 and include a low-pass filtering effect of the modulator.

C. *Post-processing and training*

The output power is monitored at all the reservoir SOA nodes, including the demasking step (cf. Fig. 1), as long as an input is applied. We call this time span the observation frame which is necessarily proportional to the input length. In all our simulations the observation frame exceeds the intrinsic timescales of the reservoir. Note that by the virtue of demasking, we increased the number of different output signals by a factor of N, resulting in a total of 80 for a 4-by-4 network with a 5 bit binary mask. This increases the number of signals available for the classifier by a factor of N, which is

beneficial for performance.

Next, the time dependence of all the demasked output signals within each observation frame is averaged out, so as to calculate their centroids. The same was done in [15].

Having post-processed the output as described above, contrary to earlier work [15] we do not perform a least-square optimisation to determine the optimal readout weights such that a linear combination of the output signals matches as closely as possible a desired output signal (e.g. 1 if the word to be recognised is present, 0 otherwise). Instead, we apply the more sophisticated lower-rank multiclass linear discriminant analysis (LDA) as supervised training method [18], which tries to maximise class separability, i.e., the distance between the lower-rank class means, by taking into account the within-class scatter. As it is characterised by linear decision boundaries, we are still true to the philosophy of classical RC to implement only linear readout weights. In this LDA, the kth discriminant function δ_k is calculated from the class mean μ_k, covariance matrix Σ, and class prior probability π_k.

The magnitude of δ_k determines the probability of assigning the observation x at the output of the reservoir to the class of spoken digit k. The class means μ_k are calculated as the average position in parameter space of all the training observations belonging to the digit k, and the class prior π_k reflects the frequency with which digit k occurs in the training set. The covariance matrix Σ is estimated in the same way as in the normal form of least squares.

A convenient performance measure is the Word Error Rate (WER), i.e. the fraction of misclassified digits. Since we have worked with a rather small data set of 500 observations, we used leave-one-out cross validation (LOOCV) to obtain better model statistics for the generalization error. For LOOCV the ith data point is removed from the data set and the retained samples are used to train the classification algorithm. A model prediction produced for the ith input is then compared to the correct class label. This procedure is repeated for each data point in the data set and the overall fraction of misclassified samples constitutes the cross-validation error estimate.

IV. Results

As benchmark, the same data set of 500 observations and leave-one-out cross-validation was simulated by reservoir-free LDA, regularised least-squares (LS), and k-Nearest Neighbours (kNN), a common instance-based machine learning algorithm [18]. This has two advantages. Firstly, it allows us to validate the use of LDA as classification algorithm given that its scores are indeed located between to those of kNN and LS. Secondly, the benchmark test aids us in putting the results from the RC techniques into the broader context of classification problems in the field of machine learning.

A closer look on the results displayed in Table II clearly shows an average WER below 5 % when training is performed on the full 77-dimensional feature vector. In order to make the benchmark results more comparable to the RC techniques we employed, the same input-to-reservoir weight matrix $W_{inp2res}$ that the reservoir requires was applied to every training vector before presenting it to RC-free LDA or kNN. This projects the 77 dimensions of the speech task down to a lower-dimensional representation, i.e. 16 dimensions for the 4x4 reservoir. Obviously the choice of $W_{inp2res}$ is far from optimal because the average WER increases significantly in this case: above 20 % WER for both LDA and LS, and more than 10 % for kNN. We try to remedy the increase in the WER by choosing a particular input-to-reservoir projection matrix W_{PCA} that was constructed from the data set's 16 largest principal components. As expected, we notice an important improvement of the WER in the third column of Table II.

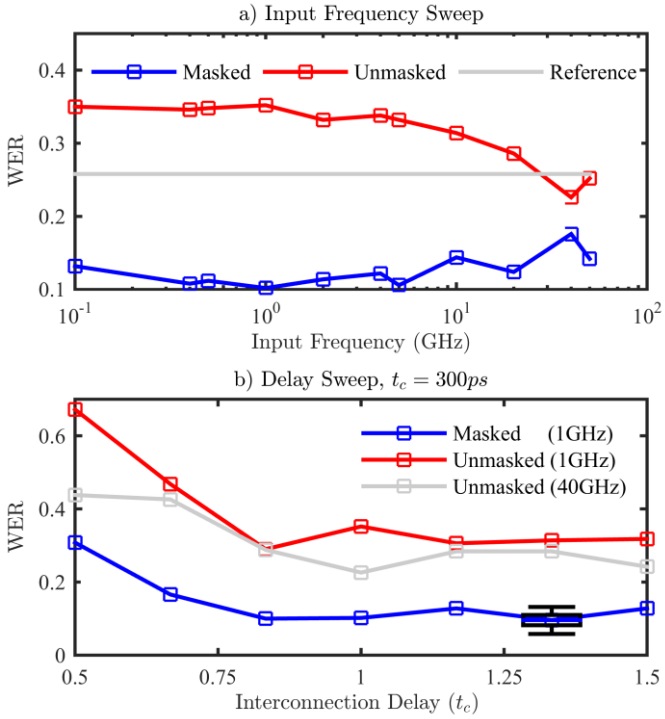

Fig. 3. WER performance plots for the 4x4 swirl network of SOAs trained by LDA. The red curves correspond to the conventional reservoir, the blue ones to the masked reservoirs (using an '11010' sequence). In a) the input sampling frequency is swept. The gray reference line shows the WER achieved by reservoir-free LDA. Subplot b) displays a sweep of the interconnection delay at the optimal input sampling frequency point (1GHz) found in a). This time we included a gray reference curve for the best sampling frequency of the unmasked reservoir (at 40 GHz). For the lowest WER the statistical error due to the random choice of $W_{inp2res}$ is indicated by error bars (bars: min/max, box: 1st/3rd quartile).

Next, we simulated the 16-node reservoir without masking the input. In all our simulations, we swept the most important reservoir parameters: the input sample rate, the interconnection delay τ, and the input (power) scaling. The minimally achieved WER of 22.6 % (settings: 6 dB/cm additional waveguide loss, 40 GHz input sample rate ($1/T_{sample}$), 300 ps interconnection delay, and 0.5 mW input peak power, see also Fig. 3) is comparable to the benchmark results of reservoir-free LDA and $W_{inp2res}$. Clearly, the use of a conventional reservoir is of little interest because it only adds complexity inside the RC layer without improving the WER any further.

Then we repeated the same parameter sweeps for the

masked reservoir. Therefore we chose an arbitrary, 5 bit long mask, i.e. the yet non-optimal sequence '11010'. Later on we will investigate the model behaviour when the particular bit mask is changed. The presence of 5 bits in one mask period satisfies the constraints on the virtual node duration, the piecewise-constant parts of the input signal: in integrated photonics we are working with short delay lines (~300 ps) that are of the same order of magnitude as the SOA's time constant (300 ps), hence 5 bits (virtual nodes) per delay line constitute a good balance between the relaxation and the inertial effects exerted on them when passing from one SOA node to the next one. The reservoir nodes constantly stay in a transient regime where computations are carried out.

TABLE II
LOWEST WER AND 95% CONFIDENCE INTERVAL OF BENCHMARK LEARNING METHODS AND METHODS INVOLVING RC

Learning Method[a]	WER all input dimensions (77 channels)	WER $W_{inp2res}$	WER W_{PCA}
Reservoir-free Methods			
LS	7.2 ± 2.3 %	31.4 ± 4.1 %	12.0 ± 2.9 %
LDA	4.4 ± 1.8 %	25.8 ± 3.8 %	6.4 ± 2.2 %
k-NN	2.2 ± 1.3 %	11.6 ± 2.8 %	2.4 ± 1.3 %
Reservoir Methods			
LS + RC		27.4 ± 3.9 %	21.0 ± 3.6 %
LDA + RC		22.6 ± 3.7 %	14.0 ± 3.0 %
LS + MRC [11010]		13.2 ± 3.0 %	6.0 ± 2.1 %
LDA + MRC [11010]		9.6 ± 2.6 %	3.6 ± 1.6 %
LDA + opt. MRC [01101]		8.2 ± 2.4 %	3.6 ± 1.6 %

[a]Method acronyms: LDA – Linear Discriminant Analysis (9 dimensions); LS – Least Squares; k-NN – k Nearest Neighbour clustering (k = 4); RC – Reservoir Computing (16 nodes); MRC – Masked Reservoir Computing (bit mask in brackets, 16 nodes)

Figure 3 shows the model behavior depending on the two most critical network parameters, viz. the input sample rate $1/T_{sample}$ and the internodal delay. Their optimal values are located at 1 GHz and 400ps for the input sample rate and the interconnection delay, respectively. In order to find the optimal working point for the reservoir, we first minimised the WER with respect to the input sample rate and then, in a second step, with respect to the internodal delay. We omitted to include the dependence on the input (power) scaling because, except for very large input peak powers of tens of mW the WER does not deviate strongly from its optimal value at 0.5 mW. From Fig. 3 we conclude that upon applying the masking scheme a significant decrease of the WER is obtained throughout the parameter space. The best achieved WER in this case (LDA, $W_{inp2res}$) is as low as 9.6 %. This result highlights not only the superior performance of a masked reservoir over an unmasked one, but also proves to outperform basic, off-the-shelf classification algorithms for even small network sizes. Crucially, it also shows that good performance is achieved over a much wider frequency band of input sampling rates of the input signals. We want to emphasise that our result differs significantly from the findings in [15], [22], in which ridge regression was applied to the output time traces before time-averaging and decision making. Therein the performance of small network sizes is much worse (26.4 % for the 4x4 SOA network) and the required input sampling frequency much higher (~80 GSample/s for 400 ps delay lines). This means that the use of LDA is beneficial from a machine-learning perspective since it provides a more robust training algorithm decreasing the WER by several percent points in both the masked and unmasked case (cf. LS + RC vs. LDA + RC and LS + MRC vs. LDA + MRC in Table II). Additionally, the masking scheme helps further improving the WER, lowering it to 3.6 % when combined with a judicious choice of the projection matrix. The key result, however, resides in the tremendously decreased input sample rate to and below 1 GSample/s.

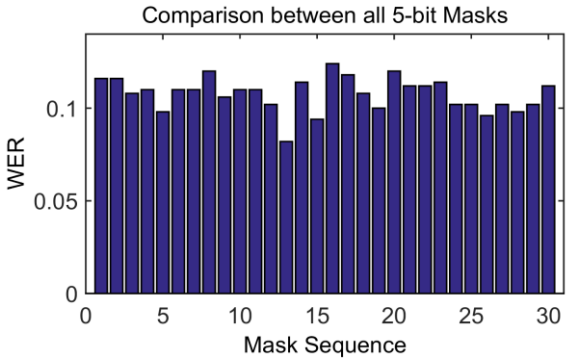

Fig. 4. Bar plots showing the average WER for different masking sequences of the input signal. The digital masks are ordered and represented by their equivalent decimal notation.

We finish this section by optimising the 5-bit mask. Recently, a similar study on binary masks has been carried out for the delayed-feedback, single-node reservoir system [25]. To this end we have studied the dependence of the average WER on the specific bit sequence in the mask at the previously found optimum for the particular 5-bit mask '10110'. The overall minimum WER determined in such a way is 8.2 % (cf. Table II, LDA + opt. MRC), obtained with the bit-mask pattern '01101'. Fig. 4 shows all the possible mask patterns of length 5, except the all-zeros and all-ones sequences which we omitted for the reason that they either produce no input at all or just repeat the input signal without masking. We labelled the remaining 30 patterns according to their decimal representation, e.g. '00011' maps to 3. According to Fig. 4, there exists only little variation in performance for the whole spectrum of possible masking sequences. Eventually, a ceiling analysis based on the entries in Table II makes clear that designing a better input weight matrix, e.g. via principal component analysis (PCA), contributes in a much better way to the overall system performance than improving the specific binary masking sequence: 1.4 % yield when optimising the binary mask vs. 6.0 % when choosing a better input weight matrix W_{PCA}. This corresponds to the findings in [26].

V. Conclusion

In this work we have demonstrated that a binary masking and demasking scheme can be used in combination with the RC concept in order to relax its input sample rate requirements: by multiplying a slow input signal with a fast mask, a new input signal is created that is compatible with the fast, intrinsic dynamics of the reservoir without sacrificing performance. A secondary time-multiplexing scheme, the demasking procedure, is applied to the readout nodes of the reservoir as described by Zhang et alii [27]. This additional demultiplexing stage provides the machine learning algorithm with a valuable extra set of predictors and results in significantly improved classification performances. Relaxed constraints on the input sample rate are of paramount importance for conducting experimental RC studies because low-bandwidth function generators are readily available. It is noteworthy that an experimental study is still very elaborate because of the large amount of optical and electronic components and devices destined to carry out the various signal processing steps. From our simulations we conclude that the necessary input sample rate can be reduced by a factor of 40 compared to conventional RC – that is from 40 GSample/s down to 1GSample/s – without any loss of performance. This processing speed is similar in magnitude to the one reported by Zhang et alii [27], in which a passive linear system with a non-linear readout was used. In contrast to their work that combines masking and hierarchical time-multiplexing to increase the effective node number (prediction accuracy) at constant processing speed, our approach introduces the masking and demasking concept to lower the input sample rate to practical values. Moreover, by using more advanced training mechanism like LDA and optimised input masks, we can achieve word error rates as low as 3.6 % in a small-sized reservoir consisting only of 16 reservoir nodes. Finally, we remark that the SOA network is operating at low input powers compared to the SOA's saturation power which efficiently removes the node non-linearity. This implies a more power-efficient solution in form of cleverly designed passive reservoirs should be possible, provided that their loss management is properly addressed.

References

[1] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," vol. 148, no. GMD Report 148, p. 43, 2001.

[2] 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.

[3] 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.

[4] H. Jaeger, "Short term memory in echo state networks," German National Research Center for Information, vol. 152, no. GMD Report 152, 2002.

[5] A. Rodan and P. Tino, "Minimum complexity echo state network.," IEEE Transactions on Neural Networks, vol. 22, no. 1, pp. 131–144, 2011.

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

[7] 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.

[8] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Isolated word recognition using a Liquid State Machine," Neural Networks, no. April, pp. 27–29, 2005.

[9] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme Recognition with Large Hierarchical Reservoirs," Advances in Neural Information Processing Systems 23, vol. 23, pp. 1–9, 2010.

[10] P. Joshi and W. Maass, "Movement generation and control with generic neural microcircuits," in Control, vol. 3141, Springer Verlag, 2004, pp. 258–273.

[11] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, 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, p. 3241, 2012.

[12] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic Reservoir Computing," Scientific reports, vol. 2, no. 287, p. 287, 2012.

[13] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," Optics Express, vol. 20, no. 20, pp. 22783–22795, 2012.

[14] 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, pp. 11182–11192, 2008.

[15] 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.

[16] 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, no. 13 September 2011, p. 468, 2011.

[17] 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.

[18] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, vol. 27, no. 2. Springer, 2009, pp. 113-119, pp. 463-468.

[19] S.-W. Lee and E.-J. Lee, "Integrated segmentation and recognition of connected handwritten characters with recurrent neural network," in Document Recognition III, 1996, vol. 1, pp. 251–261.

[20] S. Lawrence, S. Fong, and C. L. Giles, "Natural Language Grammatical Inference: A Comparison of Recurrent Neural Networks and Machine Learning Methods," in Connectionist Statistical and Symbolic Approaches to Learning for Natural Language Processing, vol. 1040, S. Wermter, E. Riloff, and G. Scheler, Eds. Springer Berlin / Heidelberg, 1996, pp. 33–47.

[21] J. Wang and G. Wu, "A multilayer recurrent neural network for solving continuous-time algebraic Riccati equations.," Neural Networks, vol. 11, no. 5, pp. 939–950, 1998.

[22] 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," Nat Commun, vol. 5, Mar. 2014.

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

[24] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," Speech Communication, vol. 12, no. 3, pp. 247–251, 1993.

[25] L. Appeltant, G. Van der Sande, J. Danckaert, and I. Fischer, "Constructing optimized binary masks for reservoir computing with delay systems," Sci. Rep., vol. 4, Jan. 2014.

[26] M. Hermans, M. C. Soriano, J. Dambre, P. Bienstman, and I. Fischer, "Photonic Delay Systems as Machine Learning Implementations," Accepted for publication in JLMR in Dec. 2014.

[27] 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, p. 31356, Dec. 2014.