Influence of the input signal's phase modulation on the performance of optical delay-based reservoir computing using semiconductor lasers

IAN BAUWENS,[1,*] ⓘ KRISHAN HARKHOE,[1] ⓘ PETER BIENSTMAN,[2] ⓘ GUY VERSCHAFFELT,[1] ⓘ AND GUY VAN DER SANDE[1]

[1]*Applied Physics Research Group, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium*
[2]*Photonics Research Group, Department of Information Technology, Ghent University-IMEC, Technologiepark Zwijnaarde 126, 9052 Ghent, Belgium*
[*]ian.bauwens@vub.be

Abstract: In photonic reservoir computing, semiconductor lasers with delayed feedback have shown to be suited to efficiently solve difficult and time-consuming problems. The input data in this system is often optically injected into the reservoir. Based on numerical simulations, we show that the performance depends heavily on the way that information is encoded in this optical injection signal. In our simulations we compare different input configurations consisting of Mach-Zehnder modulators and phase modulators for injecting the signal. We observe far better performance on a one-step ahead time-series prediction task when modulating the phase of the injected signal rather than only modulating its amplitude.

1. Introduction

With current developments in machine learning, we are becoming increasingly able to process and analyze information in our technological society [1]. The concept of reservoir computing (RC) within the machine learning field offers a simple, yet powerful technique to use recurrent networks for computing. RC systems have shown good performance in various benchmark tasks, such as speech recognition [2–4], non-linear channel equalization [5] or time-series predictions [6–8]. An RC system consists of a large recurrent neural network with fixed interconnections. Its topology can be described in three separate components: an input layer, the reservoir, and an output layer. In the input layer, the data is injected into the system and is sent to the reservoir, which consists of a recurrently connected network of non-linear nodes (i.e. neurons). The processed information is then sent to the output layer, where the output weights are optimized to match the output with a corresponding target output. The optimization of weights, and thus the training phase, occurs only in the output layer, whereas the internal weights of the reservoir itself are not altered. This makes training much more straightforward compared to other artificial neural networks (such as deep artificial networks, that also require training of the network's internal nodes) and simplifies reservoir computing in its implementation. Interesting implementations of RC systems can be found in the emerging field of neuromorphic photonics. The advantages of using photonic systems are abundant, ranging from a low-energy consumption, high-speed performance and the possibility of high inherent parallelism [9,10].

There already exist several successful implementations of photonic RC systems, e.g. based on a network of passive elements or semiconductor optical amplifiers [11–14], by using a diffractive optical element coupled to a network of vertical cavity surface emitting lasers [15] or by using excitable photonic systems (also referred to as spiking systems) [16,17]. In this paper, we focus on a single mode semiconductor laser with delay-based RC [5,18–23]. The injection of input data into this reservoir can be performed via several methods. The input data can e.g. be injected

electronically by direct modulation of the injection current [24]. In this work, however, we will focus on optically injected data, which has the advantage of allowing higher data injection rates [25]. This latter method can be performed by modulating the phase of the injected electric field by using a phase modulator or by modulating the amplitude of the electric field [19,26–29]. Although these various injection schemes will have an effect on the final performance of RC, their influence on RC performance has not yet been studied and compared in detail. In this work, we numerically investigate the effect of the optical data injection configurations on the performance of delay-based reservoir computing system.

The paper is organised as follows: in Section 2, we give a short introduction to the specific reservoir computing method which is used: delay-based RC. We also discuss the model used to perform our numerical study of the RC systems and the different input configurations considered in this paper. In Section 3, we discuss the RC performance of the different input configurations when changing the strength of the amplitude and/or phase modulation of the input signal. In this Section, we also investigate the role of the number of nodes and their separation on the performance and study the effect of phase modulation on the memory capacity of delay-based RC systems, before drawing conclusions in Section 4 of this paper.

2. Numerical implementation of reservoir computing

2.1. Delay-based reservoir computing

Semiconductor lasers with delayed feedback rely on a time-multiplexing approach to implement reservoir computing [19]. This delay-based technique has been implemented in several types of electronic or photonic reservoirs [19,21–23]. Figure 1 shows the topology of a photonic delay-based RC using a single mode semiconductor laser as non-linear node, which will be studied in this paper, and consists of an input layer, reservoir and output layer. In the input layer, we optically inject the discrete input data u_k, with k the index of the data sample, via an input configuration which we will vary in this work. Due to the time-multiplexing, we need to make use of a preprocessing mask $m(t)$ before injecting the input data into the reservoir. In this paper, this mask is created by randomly choosing N values, from following 5 sublevels: $\left[0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right]$. The mask is kept piecewise constant over the randomly selected sublevels, with each interval having a duration θ, the node separation. This piecewise constant segment is then repeated such that the mask is periodic with period $N\theta$, with N the number of virtual nodes in the system. The injection of the input samples u_k is handled numerically in the following procedure. Every data sample is first stretched to a time interval equal to $N\theta$ resulting in a piecewise constant continuous signal $u(t)$. Subsequently, we multiply this signal $u(t)$ with the mask $m(t)$ resulting in a masked data signal $S(t)$. After multiplying this masked data signal with an amplitude (and possibly a bias), it is sent to the reservoir.

The reservoir itself consists of a semiconductor laser (SL) with optical feedback with a delay τ. Note that in this paper, we assume τ and the period of the mask matched, i.e. $\tau = N\theta$. A mismatch between these two can be introduced, as in e.g. Reference [5,30–32], which can improve the performance of RCs. However, in this work, we do not consider such a mismatch here for simplicity. The reason for this is that the value of such a mismatch would need to be scanned to further optimize performance. This improvement in performance is expected to be equally applicable to any of the systems which we consider here and would not change the relative differences between them. After the signal passes through the reservoir, the output state, **A**, is sent to the output layer. The RC's output is then calculated as a linear combination of the node states using (trained) output weights **w** [9,18].

We use the intensity of the nodes as output of the reservoir. We can find the output weights **w** corresponding to the N nodes of the reservoir in the training phase. In order to achieve this, we use the output of the RC system **A**, which represents the node responses to the training input data, and the expected target data $\mathbf{y}^{train}$. In practice, the weights **w** can be retrieved by minimizing the
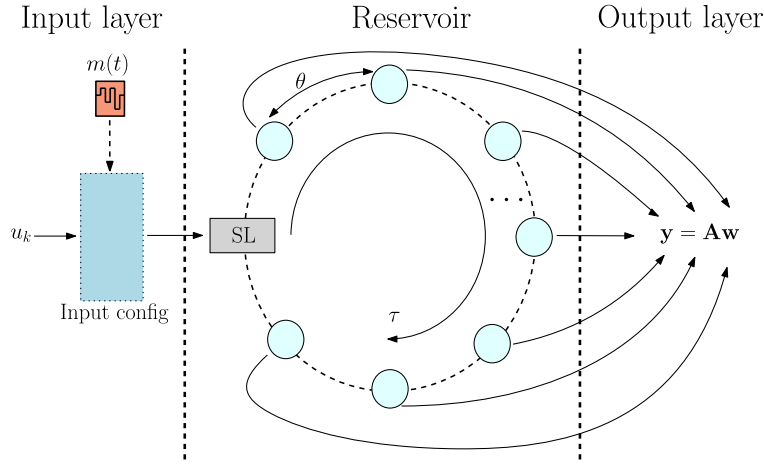
Fig. 1. Illustration of a delay-based RC system using a semiconductor laser (SL), with input data u_k, mask $m(t)$, nodal separation θ and delay time τ. The light blue circles represent the virtual nodes and the output layer is defined by the reservoir output $\mathbf{A}$ and output weights $\mathbf{w}$.

squared error $E_{sq}(\mathbf{w})$ between the predicted value for the target data $\hat{\mathbf{y}}^{train}$ and the expected target data points $\mathbf{y}^{train}$. We have used the real Moore–Penrose pseudoinverse (denoted by the symbol †)

$$\mathbf{w} = \mathbf{A}^{\dagger}\mathbf{y}^{train}. \tag{1}$$

Because of the presence of internal noise in the reservoir, we do not use any regularization methods. Once the weights have been found in the training phase, we test how the RC performs on unseen data, which is referred to as the testing phase. In order to quantify this performance, we use the normalized mean squared error (NMSE) between the expected output $\mathbf{y}^{test}$ and predicted output $\hat{\mathbf{y}}^{test}$, unless indicated differently.

2.2. Numerical implementation of our RC system and the input configurations

The delay-based RC system with a single-mode semiconductor laser as non-linear node, can be accurately modeled using rate-equations [33]

$$\frac{dE(t)}{dt} = \frac{1}{2}(1 + i\alpha)\xi n(t)\,E(t) + \eta E(t - \tau)e^{-i\Omega_0 \tau} + \tilde{F}_{\beta} + \mu E_{inj}(t) \tag{2}$$

$$\frac{dn(t)}{dt} = \Delta J - \frac{n(t)}{\tau_c} - [g + \xi n(t)]\,|E(t)|^2\,, \tag{3}$$

where $E(t)$ and $n(t)$ are the complex valued electric field of the laser and the excess amount of available carriers (both dimensionless). α represents the linewidth enhancement factor, and ξ and g the differential gain and threshold gain. Parameters η and μ are the feedback rate and the injection rate. ΔJ represents the excess pump current rate, and is defined as $\Delta J = I_{thr}\Delta I/e$, where I_{thr} is the threshold pump current, e the elementary charge and ΔI the dimensionless pump current excess, $\Delta I = (I - I_{thr})/I_{thr}$. We use a single feedback phase, which is not varied in our work, $\Omega_0 \tau = 0$. $\tilde{F}_{\beta}$ represents complex Gaussian white noise to simulate the spontaneous emission noise strength. $\tilde{F}_{\beta}$ has a zero mean and autocorrelation equal to $\langle \tilde{F}(t)\tilde{F}(t')^* \rangle = \beta/\tau_c \,\delta(t - t')$, where β controls the spontaneous emission noise and where τ_c is the carrier lifetime. Furthermore, the input data is injected through an optical input signal $E_{inj}(t)$ with the same wavelength as the free running laser, i.e. the injection frequency detuning is therefore equal to zero and not varied in this work. Following previous work in Ref. [34], we have chosen a value of 20 ps as the

standard value for the node separation [26]. Table 1 contains a summary of the parameters of above equations, with their respective standard values. Unless stated otherwise, we keep these parameters fixed to their standard value.

Table 1. Parameters, and their respective values, which are used in the simulations, unless stated otherwise.

Parameter	Symbol	Standard value
Amount of virtual nodes (i.e. neurons)	N	200
Node separation	θ	20 ps
Linewidth enhancement factor	α	3
Threshold gain	g	$1\,\mathrm{ps}^{-1}$
Differential gain	ξ	$5 \times 10^{-9}\,\mathrm{ps}^{-1}$
Spontaneous emission noise factor	β	$\approx 10^2$
Carrier lifetime	τ_c	1 ns
Threshold pump current	I_{thr}	16 mA
Excess pump current rate	ΔJ	$1.02 \times 10^5\,\mathrm{ps}^{-1}$
Feedback rate	η	$7.8\,\mathrm{ns}^{-1}$
Injection rate	μ	$98.1\,\mathrm{ns}^{-1}$
Amplitude of injected field	ϵ	100
Feedback phase mismatch	$\Omega_0 \tau$	0
Modulation amplitude of MZM	A_{MZM}	$\frac{\pi}{2}$
Bias voltage of MZM	Φ_{MZM}	$\frac{\pi}{4}$
Modulation amplitude of PM	A_{PM}	Variable
Bias voltage of PM	Φ_{PM}	0

In Fig. 2, we show the different input configurations which we have considered and which were numerically implemented in the input layer of Fig. 1. The first configuration is shown in Fig. 2(a) and consists of a Mach-Zehnder modulator (MZM), which is used to modulate the output beam of a semiconductor laser using the masked data. The second configuration consists of the combination of an MZM together with a phase modulator (PM) to inject the data and is shown in Fig. 2(b). Both of these configurations use an MZM, where it can be either balanced or unbalanced. With the balanced type, we apply a positive voltage to one arm and an equal, but negative voltage to the other arm. With the unbalanced type, we only apply a voltage to one arm. This is shown in Fig. 2, where $V_{MZM,1}$ and $V_{MZM,2}$ represent the two voltages that are applied to the two arms of the MZM, and V_{PM} is the voltage of the PM. The third and last configuration uses only a phase modulator to inject the data, as shown in Fig. 2(c). The optical input signal $E_{inj}(t)$ generated by each of the input configurations shown in Fig. 2 is subsequently sent to the reservoir.

In order to simulate the injection of data in Eq. (2), we specify the term $E_{inj}(t)$:

$$
E_{inj}(t) = \begin{cases}
\epsilon \left(1 + e^{iB_{MZM}(t)}\right) & \text{for unbalanced MZM} & (4) \\[2mm]
\epsilon \left(e^{\frac{i}{2}B_{MZM}(t)} + e^{-\frac{i}{2}B_{MZM}(t)}\right) & \text{for balanced MZM} & (5) \\[2mm]
\epsilon \left(e^{\frac{i}{2}B_{MZM}(t)} + e^{-\frac{i}{2}B_{MZM}(t)}\right) e^{iB_{PM}(t)} & \text{for balanced MZM with PM} & (6) \\[2mm]
\tilde{\epsilon}\, e^{iB_{PM}(t)} & \text{for PM} & (7)
\end{cases}
$$

The terms $B_j(t)$ ($j \in \{MZM, PM\}$) represent the masked time-dependent modulator signal which is used as input for the different input configurations of the RC system. These terms are directly related to the voltages $V_j(t)$ in Fig. 2 through $B_j(t) = \pi V_j(t)/(2V_\pi)$, where V_π is the
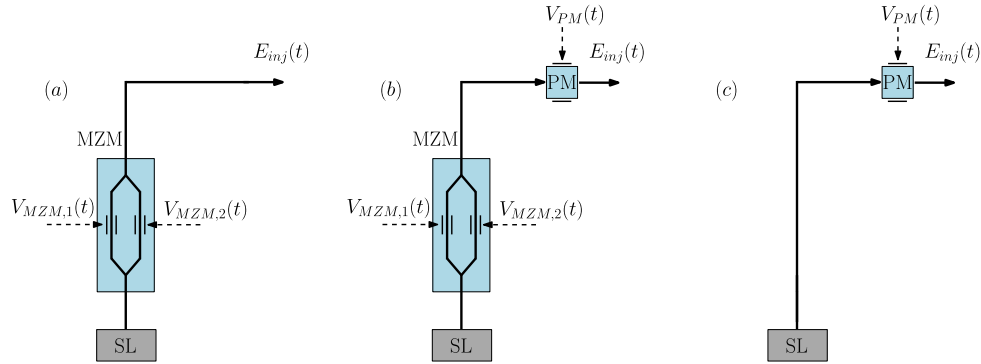
Fig. 2. Illustrations of the different input configurations which are used to optically inject the input data u_k, starting from the beam emitted by a semiconductor laser (SL): the (un)balanced MZM only (a), the (un)balanced MZM combined with a phase modulator (b) and a phase modulator (c), with their corresponding voltages $V_j(t)$.

voltage that results in a π phase-shift in the arms of the modulators. Note that the injected intensity $|E_{inj}(t)|^2$ in Eq. (4)–(6) is time-dependent, while this is not the case for Eq. (7). Therefore, we have replaced ϵ by $\tilde{\epsilon}$ in Eq. (7), such that we can set the time-averaged energy in $E_{inj}(t)$ equal for all configurations. This $\tilde{\epsilon}$ factor is calculated as the product of ϵ and the time average of the modulus of $E_{inj}(t)$, as present in Eq. (4)–(6) and allows for a fair comparison between the different input configurations.

We define the modulator signal $B_j(t)$, corresponding to input configuration j, from the masked data signal $S_j(t)$ using an amplitude A_j and bias Φ_j,

$$B_j(t) = A_j \, S_j(t) + \Phi_j. \tag{8}$$

In order to compare our results for tasks which have different ranges for their input data, we will introduce in our discussions the range of B_j, marked by ΔB_j.

For the simulations of our delay-based RC system we numerically integrate the rate-Eqs. (2)–(3), with the input configurations defined in Eq. (4)–(7).

3. Numerical results

3.1. RC performance on the Santa Fe task for different input configurations

In order to compare the performance of the different input configurations, we use a one-step ahead time-series prediction task. The input dataset used for this task is the Santa Fe dataset, which consists of just over 9000 data points sampled from a far-IR laser in a chaotic regime [35]. The goal is to find the input configuration which results in the lowest error, and thus the best performance for this particular task with the given reservoir parameters. Typical values for the NMSE for the Santa Fe one-step ahead predictions via simulations of RC systems are around 0.01 [26,36].

We have taken the first 3000 data samples from the discrete Santa Fe dataset, u_k^{train}, where $k \in \{1, \ldots, 3000\}$, as the training set in the RC system. As test set u_k^{test}, we have taken the next 1000 data samples, as done in Ref. [37]. Before injecting the signals, all of the Santa Fe data was normalized over the whole dataset, so that for both training and testing, $u_k \in [0,1]$. We have repeated each numerical experiment 10 times, each with different mask realizations, from which we calculate the average NMSE and its standard deviation. We use the standard deviation as the error bars in subsequent figures.

If we use an unbalanced MZM as input configuration, we find a value for the NMSE = 0.0194 ± 0.0047 for the testing set. If we consider a balanced MZM as input configuration, we find an NMSE = 0.134 ± 0.044. Both results are in agreement with typical NMSE values found in literature [26,36], however, it is also immediately clear that the unbalanced MZM outperforms the balanced MZM as input configuration. Our conjecture is that this difference is due to the fact that the unbalanced MZM inherently modulates the injected signal $E_{inj}(t)$ both in amplitude and phase, whereas the balanced MZM is not performing any phase modulation. To investigate this, we consider the input configuration of Fig. 2(b), an unbalanced MZM combined with a PM. We alter no parameters with regards to the MZM, but now have complete freedom over the modulation in phase. Our results are shown in Fig. 3, where we show the NMSE versus the total range of the phase modulator signal ΔB_{PM} for different input configurations. In this figure, we have indicated the performance of the balanced and unbalanced MZM (without PM) as horizontal lines. For the balanced MZM combined with a PM, we have made the distinction between a case where we reuse the mask of the MZM for the PM, shown in blue, and a case in which we use a different mask for the PM and the MZM, shown in red. Note that we do not show the performance of the unbalanced MZM combined with a phase modulator, as this configuration can always be replaced by an equivalent balanced MZM combined with a phase modulator.
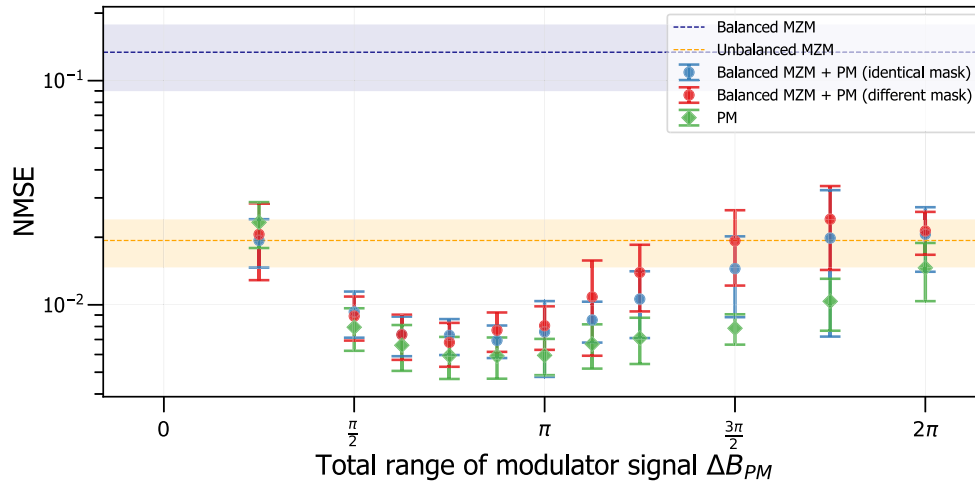
Fig. 3. NMSE in function of the total range ΔB_{PM} of the phase modulator signal for one-step ahead prediction of Santa Fe data.

We observe for all input configurations with a PM that the NMSE initially decreases when increasing ΔB_{PM}, reaches an optimal point (lowest) and then again increases for larger ΔB_{PM}.

When investigating the balanced MZM combined with a phase modulator, we observe no significant difference in performance between using the same mask for the PM and the MZM or opting for a different mask. Furthermore, we observe that the input configuration of a balanced MZM combined with a phase modulator, with an identical mask, reaches around $\Delta B_{PM} = \frac{\pi}{4}$ an NMSE which is the same as the situation where we use only an unbalanced MZM. This can be understood from the fact that the optical input signal $E_{inj}(t)$ is actually identical in both cases, which can be verified by setting Eq. (6) equal to Eq. (4).

As we achieve a large improvement by adding a phase modulator, the question can be raised whether an MZM is required at all to obtain good RC performance. Ultimately, this allows for a simpler input configuration that only uses a phase modulator. We observe in Fig. 3 that the input configuration where only a PM is used results in the best mean NMSE for the given reservoir. We do, however, want to stress that the standard deviation error bars of this configuration overlap

with the configuration of the balanced MZM combined with a phase modulator. Nonetheless, this shows that by removing the MZM from the input configuration, and thus reducing the complexity of the input system, a similarly good performance compared to input configurations with an MZM can be achieved.

In Fig. 3, we observe that the lowest NMSE values occur for input configurations with PMs around the broad range of $\Delta B_{PM} = \frac{\pi}{2}$ to $\Delta B_{PM} = \pi$. Therefore, we achieve an improvement in the performance within this large ΔB_{PM} range. This optimal NMSE can be explained by two factors. For small ΔB_{PM} (around $\frac{\pi}{4}$), the system is limited by noise that obstructs the masked data, so that it becomes difficult for the system to distinguish noise from different sublevel mask values. For large ΔB_{PM}, the phase modulated signal will stand to wrap on itself. Both of these phenomena will have a negative effect on the achieved performance and explain the existence of the optimal range in the modulator signal range ΔB_{PM}. Note that it is possible to change some of the internal parameters, such as e.g. the linewidth enhancement factor α or the injection frequency detuning. However, we do not change these internal system parameters, as we are interested here in the influence of the input configuration rather than in the influence of these internal parameters on RC performance. Moreover, investigating the role of these internal parameters in detail would require to scan a large multi-dimensional parameter space in order to find their optimal values, which is outside of the scope of this paper. However, we have performed a quick scan (not shown) with a small detuning of $\Delta \omega = \pm 1$GHz and found qualitatively the same results. Additionally, we have redone the simulations for an α factor of 5, and also found a broad maximum in RC performance around $\Delta B_{PM} = \frac{\pi}{2}$, which indicates that phase modulation remains important to achieve optimum RC performance.

3.2. Effect of number of nodes

In the previous section, we have fixed the number of neurons $N = 200$. In this Section, we investigate how far N can be reduced because of the better performance due to the inclusion of a phase modulator in the input configuration.

In Fig. 4, the NMSE is shown in function of the amount of virtual nodes N for both the input configuration with only a PM (in green) and the input configuration with the balanced MZM and a phase modulator (with identical mask, in blue). Additionally, the NMSE in function of the number of neurons for the unbalanced MZM is also shown (in orange). For clarity, the balanced MZM is left out because of its poor performance. The total modulator range ΔB_{PM} is kept fixed, and corresponds to the minimum NMSE achieved in Fig. 3. Therefore, for the configuration with only a phase modulator we have chosen $\Delta B_{PM} = \pi$ and for the balanced MZM with PM, we use $\Delta B_{PM} = \frac{3\pi}{4}$.

We observe in Fig. 4 that for low N we find a poor performance, because of the high NMSE, for all input configurations. The performance increases for increasing N for all input configurations, until it eventually saturates. This is because, for large N, several nodes will become increasingly correlated, thus resulting in a stagnating performance when increasing N. We observe that the saturating point, where the NMSE remains approximately constant, can be found around $N = 125$ for the unbalanced MZM, while this is around $N = 75$ for the phase modulator. As a result, by choosing a phase modulator at the input of the RC system, we can vastly reduce the amount of virtual nodes and increase the data injection rate, while retaining the same NMSE. For example, we achieve the same performance with a PM as input configuration for an RC system with $N = 50$ as an unbalanced MZM for an RC system with $N = 125$. This reduction in N improves the computational performance, due to a lower computation time.
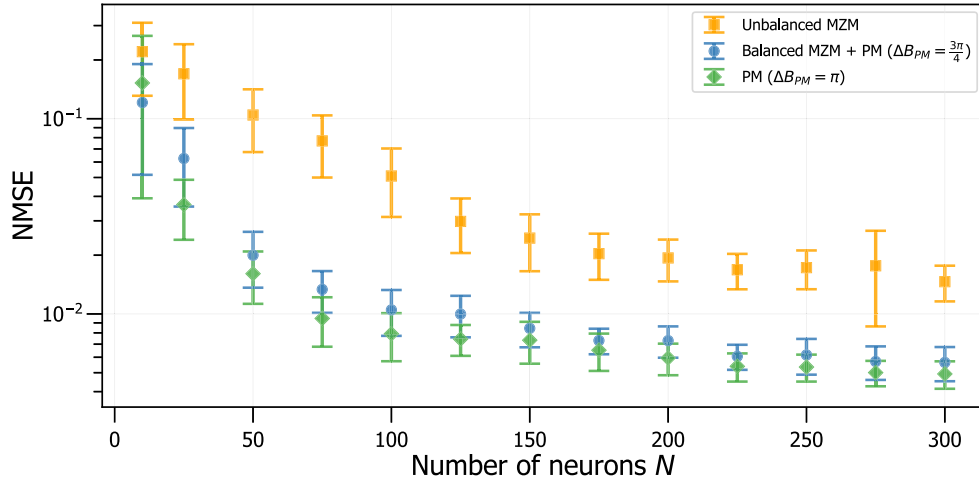
Fig. 4. NMSE in function of the number of virtual nodes for one-step ahead prediction of Santa Fe data.

### 3.3. Effect of node separation

In the previous results, we have fixed the node separation at 20 ps, as was done in Ref. [34]. Varying this value can give insight into which timescales are relevant in the RC system, which is what we will study in this Section.

In Fig. 5, we show the NMSE as a function of the node separation θ for the unbalanced MZM (in orange), for the balanced MZM combined with a PM (with identical mask, in blue) and for the input configuration consisting of only a PM (in green). The values for ΔB_{PM} of the PMs are taken equal to the most optimal values found in Section 3.1, i.e. $\Delta B_{PM} = \frac{3\pi}{4}$ for the balanced MZM with a PM and $\Delta B_{PM} = \pi$ for the PM only. For computational reasons, the amount of neurons for all these cases has been kept constant at $N = 50$ nodes, so that θ is the only varying parameter, with values scanned between 2.5 ps and 1 ns. This implies that we are probing both the short timescales related to phase dynamics as well as long timescales related to relaxation oscillations. We want to remark that the optimum node distance depends on the laser's inherent timescale and thus also on the laser parameters.

In Fig. 5, we observe a sharp decrease in the NMSE when increasing the node separation θ for both the balanced MZM combined with a PM and for the PM only. This behaviour holds until around $\theta = 25$ ps. For the unbalanced MZM, there is a somewhat smaller decrease in the NMSE. At $\theta = 25$ ps, we find a global minimum for all three input configurations, after which the NMSE increases with increasing θ. Additionally, we observe that for the PM and for the balanced MZM combined with the PM configurations, we have similar NMSE values for θ below 100 ps. For θ above 100 ps, we observe that the performance of all configurations is different. For node separations between 100 ps and 400 ps, we observe that the NMSE remains approximately constant for the unbalanced MZM and the balanced MZM combined with a PM. For the input configuration where we only use a PM, we observe that in this region the NMSE to some extent slightly decreases, around 150 ps. For node separations higher than 400 ps, all three input configurations show an increasing NMSE when increasing the node separation. Furthermore, over nearly all values of θ, we observe a better performance for both the PM and balanced MZM with a PM compared to the input configuration consisting of only an unbalanced MZM.

The global NMSE minimum in Fig. 5, occurs around the same value for all input configurations, $\theta = 25$ ps, indicating that the initial value of $\theta = 20$ ps from Ref. [34], is well chosen for the RC systems studied in this paper. This can be explained by the fact that this timescale corresponds to
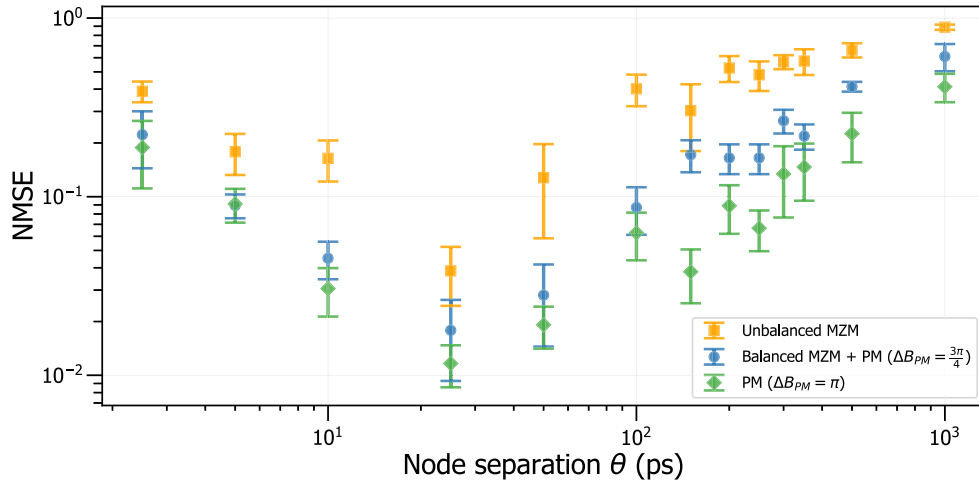
Fig. 5. NMSE in function of the node separation for one-step ahead prediction of Santa Fe data.

the typical timescales of phase dynamics in this system [26]. The slight decrease around 150 ps can be found for the input configuration consisting of only a PM, although this decrease is less pronounced here. The corresponding timescale is around the same characteristic timescale of that of relaxation oscillations (with typical frequencies of several GHz).

3.4. Memory capacity of PM input configurations

In Section 3.1, we have found that ΔB_{PM} reaches an optimal value which results in the best performance on the Santa Fe task. In this Section, we calculate the non-linear memory capacity of different input configurations in order to quantify the memory of our reservoir system, which is independent of any specific task and useful to better understand the performance differences. The procedure for calculating the memory capacity is based on the techniques from Ref. [38]. One injects discrete inputs u_k, drawn from a uniform distribution between $[-1, +1]$, into a reservoir and trains the system to reconstruct a set of products of normalized Legendre polynomials via l-delayed past inputs u_{k-l} [37,38], which is limited in our work to $l = 10$.

Reproducing a certain target input $\hat{y}_k$ by a set of Legendre polynomials $P_{d_l}(\cdot)$, of degree d_l, from previous inputs is performed by

$$\hat{y}_{\{d_l\}}(k) = \prod_l P_{d_l}(u_{k-l}).\tag{9}$$

The mean squared error between the signal calculated by the RC $\hat{y}(k)$ and expected signal $y(k)$ over all input samples can be calculated. This value is then used for the definition of the memory capacity, which is defined as

$$C = 1 - \frac{\left\langle (\mathbf{y} - \hat{\mathbf{y}})^2 \right\rangle}{\left\langle \mathbf{y}^2 \right\rangle},\tag{10}$$

with values for C between 0 and 1 and where the average is taken over all input samples and combinations of Legendre polynomials.

The sum of all memory capacities C, will result in the total computational capacity of the system, CC. The theoretical upper limit of this total computational capacity in our system is given by the amount of neurons N. In practice, we can only use a finite amount of input data, due to computational reasons, which can potentially lead to overestimating the value for the

memory capacity. To counter this effect, we utilise a threshold capacity C_{thr} so that values below this threshold are not counted, as discussed in Ref. [38]. This is because memory capacities below this number are not statistically relevant and are thus discarded. The value for this memory capacity threshold is calculated by a generalised χ^2.

Finally, to better understand the dependence of the results of Section 3.1 on ΔB_{PM}, we calculate the task independent memory capacities C of the various input configurations. We perform the simulations for 25 virtual nodes, with a node separation $\theta = 20$ ps on uniformly distributed data with a size of $T = 5 \times 10^5$ data points and values between -1 and 1. The threshold memory capacity is approximately $C_{thr} \approx 2.7 \times 10^{-4}$ for the configurations in this Section.

In Fig. 6, we show the memory capacity of the systems in function of ΔB_{PM}. The memory capacities are represented as coloured bars per degree. We have opted to only calculate the memory capacity for the input configurations consisting of only a phase modulator, because it is the simplest input configuration we considered that results in the best performance.
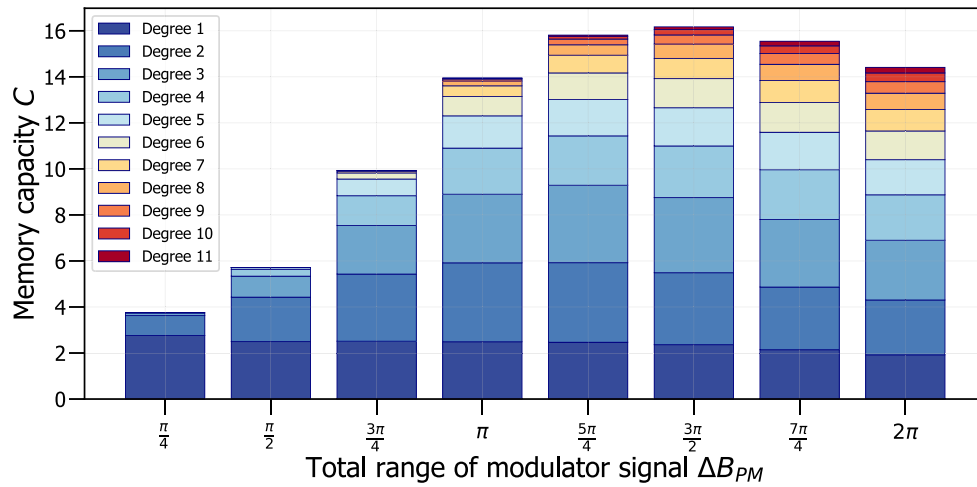
Fig. 6. Memory capacity per degree of linearly independent basis functions for input configurations with only a PM.

In the same figure, we observe that the total memory capacity of the RC system combined with a PM increases with increasing ΔB_{PM}, up to $\Delta B_{PM} = \frac{3\pi}{2}$, after which it slightly decreases again for larger ΔB_{PM}. Nonetheless, there exists a broad region, situated at large ΔB_{PM} values, for which the total memory capacity remains fairly high and where it does not vary much with ΔB_{PM}. For small ΔB_{PM}, we observe that the total memory capacity consists of mostly degree 1 and 2. For larger ΔB_{PM}, the larger degrees will become more important and have a larger contribution to the total memory capacity. We also observe that for large ΔB_{PM}, the memory capacity reaches its largest value for degree 3 and decreases for larger degrees.

Furthermore, the highest memory capacities for degree 1 and 2 (i.e. linear and quadratic capacities) are given by the phase modulators where $\Delta B_{PM} = \pi$ and $\Delta B_{PM} = \frac{5\pi}{4}$. These values for ΔB_{PM} agree with the best performing ranges found in Fig. 3 for the Santa Fe task, so that the systems with the highest linear and quadratic memory capacities result in the best NMSE for the Santa Fe dataset. The fact that we reach an optimal value for the linear and quadratic memory capacity for certain ΔB_{PM} is because we can make the same argument as in Section 3.1.

For further studies, we hypothesise that for tasks which require large degrees of memory capacities, and thus also large total memory capacities, setting $\Delta B_{PM} = \frac{3\pi}{2}$ will result in the most optimal performance.

We remark that the theoretical total computational capacity is bounded above by the number of nodes [38], which is in this case 25. The fact that we do not reach this upper limit can be explained by correlations present between the nodes and by the existence of noise in the systems, which effectively reduce the memory capacity of the system.

4. Conclusion

We have numerically investigated the effect of several input configurations used for optically injecting data, with the goal of improving delay-based reservoir computing with semiconductor lasers. We have found that using an unbalanced Mach-Zehnder modulator as input configuration outperforms the balanced Mach-Zehnder modulator in the one-step ahead prediction Santa Fe task. This performance difference can be explained due to the unbalanced Mach-Zehnder modulator inherently modulating the injected signal both in amplitude and phase, whereas the balanced Mach-Zehnder modulator is not performing any phase modulation. This has led us to the investigation of using a phase modulator to inject the signal, which resulted in an improved performance compared to the systems using a Mach-Zehnder modulator, and also an improved performance compared to literature. We therefore conclude that modulating the phase of the injected signal strongly increases the performance of optical reservoir computing. As a result, we were able to drastically decrease the required amount of virtual nodes of our reservoir computer, while retaining the same performance. For example, we could achieve the same performance using a phase modulator combined with a reservoir computer with only 50 nodes when compared with an unbalanced Mach-Zehnder modulator combined with a reservoir computer with 125 nodes. We conclude that using only a phase modulator, with well-adjusted modulation amplitude, as input configuration is ideal for the one-step ahead prediction Santa Fe task, both performance-wise as well as in simplicity for implementation. Ultimately, from our results on the task-independent memory capacity we see that many non-linear capacities are available which indicates that a phase modulator as input configuration will have a beneficial effect on the performance for other tasks.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," IEEE Network **30**(3), 22–29 (2016).
2. D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, "Isolated word recognition with the liquid state machine: a case study," Inf. Process. Lett. **95**(6), 521–528 (2005).
3. M. R. Salehi, E. Abiri, and L. Dehyadegari, "An analytical approach to photonic reservoir computing–a network of SOA's–for noisy speech recognition," Opt. Commun. **306**, 135–139 (2013).
4. D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Reservoir-based techniques for speech recognition," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, (IEEE, 2006), pp. 1050–1053.
5. Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," Sci. Rep. **2**(1), 287 (2012).
6. H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," Science **304**(5667), 78–80 (2004).
7. E. S. Skibinsky-Gitlin, M. L. Alomar, E. Isern, M. Roca, V. Canals, and J. L. Rossello, "Reservoir computing hardware for time series forecasting," in *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, (IEEE, 2018), pp. 133–139.
8. D. Canaday, A. Griffith, and D. J. Gauthier, "Rapid time series prediction with a hardware-based reservoir computer," Chaos: An Interdiscip. J. Nonlinear Sci. **28**(12), 123119 (2018).
9. G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," Nanophotonics **6**(3), 561–576 (2017).

10. T. F. De Lima, B. J. Shastri, A. N. Tait, M. A. Nahmias, and P. R. Prucnal, "Progress in neuromorphic photonics," Nanophotonics **6**(3), 577–599 (2017).

11. S. Sackesyn, C. Ma, J. Dambre, and P. Bienstman, "Experimental realization of integrated photonic reservoir computing for nonlinear fiber distortion compensation," Opt. Express **29**(20), 30991–30997 (2021).

12. K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," IEEE Trans. Neural Netw. **22**(9), 1469–1481 (2011).

13. K. Vandoorne, W. Dierckx, B. Schrauwen, D. Verstraeten, R. Baets, P. Bienstman, and J. Van Campenhout, "Toward optical signal processing using photonic reservoir computing," Opt. Express **16**(15), 11182–11192 (2008).

14. K. Takano, C. Sugano, M. Inubushi, K. Yoshimura, S. Sunada, K. Kanno, and A. Uchida, "Compact reservoir computing with a photonic integrated circuit," Opt. Express **26**(22), 29424–29439 (2018).

15. D. Brunner and I. Fischer, "Reconfigurable semiconductor laser networks based on diffractive coupling," Opt. Lett. **40**(16), 3854–3857 (2015).

16. J. Robertson, M. Hejda, J. Bueno, and A. Hurtado, "Ultrafast optical integration and pattern classification for neuromorphic photonics based on spiking VCSEL neurons," Sci. Rep. **10**(1), 6098 (2020).

17. A. N. Tait, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Broadcast and weight: an integrated network for scalable photonic spike processing," J. Lightwave Technol. **32**(21), 4029–4041 (2014).

18. 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," Nat. Commun. **2**(1), 468 (2011).

19. K. Harkhoe, G. Verschaffelt, A. Katumba, P. Bienstman, and G. Van der Sande, "Demonstrating delay-based reservoir computing using a compact photonic integrated chip," Opt. Express **28**(3), 3086–3096 (2020).

20. G. Van der Sande, K. Harkhoe, A. Katumba, P. Bienstman, and G. Verschaffelt, "Integrated photonic delay-lasers for reservoir computing," in *Physics and Simulation of Optoelectronic Devices XXVIII*, vol. 11274 (International Society for Optics and Photonics, 2020), p. 112740D.

21. M. C. Soriano, S. Ortín, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. Van der Sande, "Delay-based reservoir computing: noise effects in a combined analog and digital implementation," IEEE Trans. Neural Netw. Learning Syst. **26**(2), 388–393 (2015).

22. H. Toutounji, J. Schumacher, and G. Pipa, "Optimized temporal multiplexing for reservoir computing with a single delay-coupled node," in *The 2012 International Symposium on Nonlinear Theory and its Applications (NOLTA 2012)*, (2012).

23. 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," Phys. Rev. X **7**(1), 011015 (2017).

24. Q. Zeng, Z. Wu, D. Yue, X. Tan, J. Tao, and G. Xia, "Performance optimization of a reservoir computing system based on a solitary semiconductor laser under electrical-message injection," Appl. Opt. **59**(23), 6932–6938 (2020).

25. D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, "Parallel photonic information processing at gigabyte per second data rates using transient states," Nat. Commun. **4**(1), 1364 (2013).

26. R. M. Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van der Sande, "Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics," Opt. Express **22**(7), 8672–8686 (2014).

27. K. Sozos, C. Mesaritakis, and A. Bogris, "Reservoir computing based on mutually injected phase modulated lasers: A monolithic integration approach suitable for short-reach communication systems," in *Optical Fiber Communication Conference*, (Optical Society of America, 2021), pp. W6A–4.

28. J. Nakayama, K. Kanno, and A. Uchida, "Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal," Opt. Express **24**(8), 8679–8692 (2016).

29. Q. Cai, Y. Guo, P. Li, A. Bogris, K. A. Shore, Y. Zhang, and Y. Wang, "Modulation format identification in fiber communications using single dynamical node-based photonic reservoir computing," Photonics Res. **9**(1), B1–B8 (2021).

30. F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," Opt. Express **20**(20), 22783–22795 (2012).

31. F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," Sci. Rep. **6**(1), 22381 (2016).

32. F. Stelzer, A. Röhm, K. Lüdge, and S. Yanchuk, "Performance boost of time-delay reservoir computing by non-resonant clock cycle," Neural Networks **124**, 158–169 (2020).

33. D. Lenstra and M. Yousefi, "Rate-equation model for multi-mode semiconductor lasers with spatial hole burning," Opt. Express **22**(7), 8143–8149 (2014).

34. K. Harkhoe and G. Van der Sande, "Delay-based reservoir computing using multimode semiconductor lasers: Exploiting the rich carrier dynamics," IEEE J. Sel. Top. Quantum Electron. **25**(6), 1–9 (2019).

35. A. S. Weigend and N. A. Gershenfeld, "The Santa Fe time series competition data," (1991).

36. 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," Opt. Express **21**(1), 12–20 (2013).

37. K. Harkhoe and G. Van der Sande, "Task-independent computational abilities of semiconductor lasers with delayed optical feedback for reservoir computing," in *Photonics*, vol. 6 (Multidisciplinary Digital Publishing Institute, 2019), p. 124.

38. J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," Sci. Rep. **2**(1), 514 (2012).