POST-DIGITAL - European Training Network on Post-Digital Computing [GA860360]

Document Details

Title	Deliverable 4.2 Simulation-based evaluation of multi-reservoir design methodology for standard and advanced benchmarks
Deliverable number	D4.2
Deliverable Type	Report (public)
Deliverable title	Simulation-based evaluation of multi-reservoir design methodology for standard and advanced benchmarks
Work Package	WP4- Benchmarking and applications
Description	Simulation-based evaluation of multi-reservoir design methodology for standard and advanced benchmarks
Deliverable due date	30/09/2023
Actual date of submission	
Lead beneficiary	IMEC
Version number	V1.0
Status	Draft

Dissemination level

PU	Public	X
CO	Confidential, only for members of the consortium (including Commission Services	

Project Details

Grant Agreement	860360
Project Acronym	POST-DIGITAL
Project Title	POST-DIGITAL - European Training Network on Post-Digital Computing
Call Identifier	H2020-MSCA-ITN-2019
Project Website	https://postdigital.astonphotonics.uk/
Start of the Project	1 April 2020
Project Duration	48 months

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860360

Table of Contents

List of Figures

Executive Summary

This deliverable reports on research related to the design and simulation-based evaluation of multi-reservoir systems by 4 ESRs in the PostDigital project. Each contribution reflects (a part of) the respective ESRs personal research project. As a consequence, these contributions discuss multi-reservoir systems of different types and use different evaluation strategy. Together, these different contributions illustrate that the PostDigital project's research topics stretch from mature technologies and application domains, to innovative theoretical concepts and design methodologies.

The first contribution (Section 1), the work of Sarah Masaad (ESR9, UGent-imec), focuses on integrated (parallel) photonic reservoir computing systems. She proposes a hybrid serially coupled multi-module system, consisting of (i) a photonic, on-chip reservoir, (ii) a KKK receiver and (iii) digital post-processing. The tunable parameters in this system are jointly optimised with backpropagation. Numerical simulation results demonstrate the successful reduction of errors from a fiber-based telecommunication system by up to 4 times.

The second and third contributions report progress towards the realisation of multi-reservoir systems with delayed feedback optoelectronic reservoir systems. The work of Tigers Jonuzi (ESR11 - VLC Photonics) in Section 2 is focused on enabling technology for such devices. He proposes an architecture for an analog readout layer for time-multiplexed reservoir computing. The simulation results show how this architecture supports the performance of the reservoir enabling real-time, low-latency signal processing beyond GHz range. Section 3 reports on the work of Mirko Goldmann (ESR5, IFISC, UIB-CSIC), characterising the computational properties of a (simulated) system of two coupled optoelectronic delayed-feedback reservoirs. He uses the theoretical information processing capacity benchmarking tool Dambre et al. [2012] to analyse the impact of different coupling schemes between the two reservoirs on computation. The results show that, by changing the way the modules are coupled, one can boost two contrasting properties, either long-lasting linear memory or high nonlinearity, adapting the overall system to tasks with different demands.

Finally, the fourth contribution (work of Benedikt Vettelschoss, ESR10, UGent-imec) in section 4 focuses on novel approaches to open up new universal computation formalisms with reservoir computing. In particular, he re-investigates the finite state machine (FSM) paradigm and a way to embed its switchable behaviour into the traditional reservoir model system: an ESN. The long-term view is to create multi-reservoir systems in which one reservoir provides the switchable behaviour (the state-transition graph of the FSM), while two (or more) other reservoirs translate the input into the desired excitation signals for this module, and transform the input and state into the desired output. Since the last two functionalities could be provided by previous work on multi-reservoir systems performed at UGent Freiberger et al. [2020], the current work focuses on the missing (and most difficult) part of this system: embedding a state transition graph into an ESN. Benedikt proposes to embed the desired network attractor into an ESN by driving it towards a multiple Hopf bifurcation by means of unsupervised synaptic plasticity. Given the theoretical nature of this approach, the deliverable text first introduces the stepping stones that are necessary for understanding the solution. Proof of concept simulation results for a two-state system are then presented. Simulations show how a simple ESN is indeed driven towards a multiple Hopf bifurcation and next steps are outlined.

1 Multi-Reservoir Network for Transceiver Nonlinearity Equalization

1.1 Introduction

Multi-reservoir design is the use of a hierarchical topology to advance the performance of individualistically operating reservoirs. In this section, we present simulation results pertaining to such a network, used for the equalization of telecom transceiver errors. Our design of the multi-reservoir topology involves a photonic recurrent structure that is cascaded with a feed-forward electronic one. This work, done by Sarah Masaad at Ugent-imec, numerically demonstrates the successful reduction of errors from a fiber-based telecommunication system by up to 4 times.

To meet current and provisioned data requirements, including those generated by cloud services and Internet of Things devices, new technologies targeting short-haul systems and data center interconnects are required. To address some of these challenges, self-coherent receivers like the Kramers-Kronig (KK) receiver allow the detection of both the phase and amplitude of light without the high costs associated with coherent detection. However, this comes at the expense of operational constraints that are required to ensure linearity of the receiver. Namely, an unmodulated carrier must be transmitted with the modulated signal and should have a significantly higher power. The power ratio between the unmodulated carrier and the message signal is termed CSPR and should be at least 9 dB. In addition, the receiver processes must operate at 6 samples/symbol (sps) instead of the standard 2 sps required for coherent receivers. These constraints impose additional power and processing costs but are important for correct operation. If these constraints are not respected, the receiver becomes nonlinearly distorting, which introduces very high errors.

1.2 Methods

The proposed system operates based on a multi-network design, where a linear photonic reservoir cascaded with the KK receiver is a network followed by a feed-forward electronic network. The entire system is trained through backpropagation such that an optimum set of weights is found to best approximates the distorted signal to the target symbols. Furthermore, the system is trained in a back-to-back setup (i.e., without a fiber), which allows targeting the nonlinear transceiver errors. These errors originate from the receiver being operated at 3 sps instead of the required 6 and at CSPRs below 9 dB. Training in this manner facilitates the operation of the equalization network as a “plug-and-play” equalizer. This means that once trained, the solution is tested in systems deploying standard single-mode fiber links varying between 10 to 60 km. The network performs consistently well and reduces the overall BER by up to 4 times. Note that since the network only addresses transceiver nonlinearity, the fiber-caused impairments are addressed through Digital Signal Processing (DSP).

The proposed solution includes a photonic on-chip network trained following a machine learning paradigm known as reservoir computing. Importantly, the reservoir replaces a bandpass filter before the receiver, which is typically required in a self-coherent systems, and thus is not an additional optical component. Since chromatic dispersion compensation should happen electronically, the combined behavior of the reservoir and receiver should be linear. By performing nonlinear transceiver equalization under the condition of a linear output, cascading the optical processing with digital processing is achievable.

The overall system is shown in Figure 1, where the training setup is denoted as pipeline b (blue)

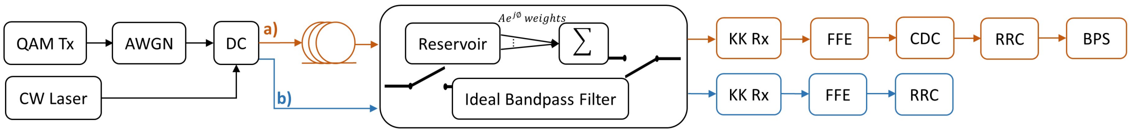

Figure 1: Schematic of a. (orange) system under study and b. (blue) training system. Black indicates common blocks. Both systems are simulated with a reservoir for proposed method and an ideal bandpass filter for benchmarking.

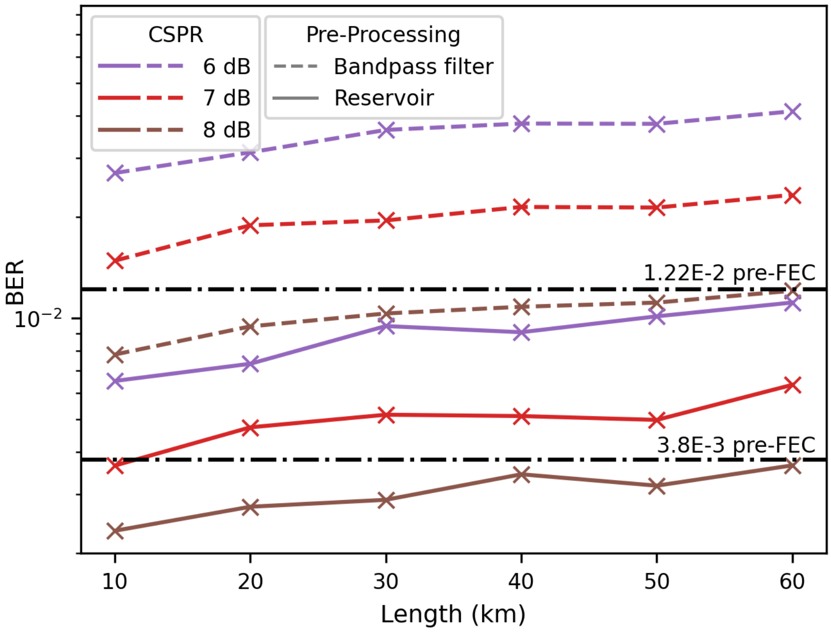

Figure 2: BER results vs link length for test set.

while the testing setup is pipeline a (orange). The blocks common to both pipelines are shown in black. Furthermore, to benchmark the performance of the system, an ideal bandpass filter is used as opposed to the use of the optical reservoir. The feed-forward network is kept for both the benchmark and the proposed solution.

In the testing pipeline, ideal chromatic dispersion compensation (CDC) and blind-phase search (BPS) are used to address chromatic dispersion and laser phase noise respectively.

1.3 Results

Figure 2 shows the achieved BER on 3 different transceiver conditions, where the CSPR is varied between 6 and 8 dB. The benchmark deploying the bandpass filter is shown as dotted lines whereas the multi-network proposed solution is shown as solid lines. The overall performance shows up to 4 times improvement in BER across all length studied.

1.4 Conclusion

This work will facilitate the deployment of self-coherent receivers like the KK receiver in short-haul systems, since their operational constraints were considered debilitating. By relieving some of these constraints through a simple and integrated solution, the achieved performance falls below the required BER and as such can be used in, for example, data centers.

We are in the process of preparing these results for publication.

2 Real-time delay reservoir computing for sequential optical signal processing

2.1 Introduction

The field of analog machine learning holds significant promise in addressing the limitations of traditional computing methods. As a result, there has been a growing focus on merging brain-inspired concepts with the benefits of optical computing. Photonic integration stands out as a formidable challenge in enabling broader applications in this domain. Reservoir computing (RC) emerges as one of the most appealing approaches for hardware implementations. It offers a conceptually simple yet powerful technique for training recurrent neural networks, overcoming the need for optimizing connections between input and network nodes or among the network's nodes. This simplification dramatically reduces hardware complexity, allowing for a concentrated effort on optimizing only the output weights. For these reasons, Reservoir Computing proves to be a well-suited method for processing sequential data, and numerous electronic and photonic implementations have demonstrated exceptional performance in tasks like time series prediction.

However, in most existing hardware implementations of RC, the output information is digitally post-processed. This involves storing the output, and applying the necessary weighting and summation offline, which results in the loss of the rapid real-time analog optical computing capabilities. In response to this challenge, we propose the development of an integrated readout layer for a photonic implementation of RC. This readout layer will be capable of applying weights and performing summations in the optical analog domain, enabling low latency and ultrafast real-time processing. Our focus is on designing a readout layer suitable for delay-based reservoir computers, which rely on a single hardware node and a feedback delay loop. The following design might not be central in realizing what is promised in the deliverable, but we believe it is of crucial importance at the time of realizing in hardware multi-reservoir designs.

2.2 Methods

Photonic integrated neural networks hold the promise of delivering exceptionally high data bandwidth while consuming minimal power and capitalizing on the inherent parallel processing capabilities of light-based computing. Researchers have explored different methods of integrated time-delay reservoir computing, which involve simulating the inner workings of microring resonators or experimenting with semiconductor lasers, with and without external feedback loops. However, it's worth noting that all of these implementations necessitate post-processing of the output signal in digital form.

The field of integrated photonics has matured to a point where various passive components can be designed to fulfill the requirements of machine learning techniques. A simplified representation of the necessary components for performing real-time operations like weighting and summing of the output signal in a delay-based reservoir (with 4 virtual nodes) is depicted in Figure 3(a). The optical analog readout system consists of an amplitude modulator to apply trained optical weights, a cascaded setup of multimode interferometers (MMIs) for splitting and combining signals, and delay lines to temporally align the delayed virtual nodes. Phase shifters have also been introduced to either match phase for coherent summation or apply phase-based weights.

Figure 3(b) provides a closer look at the optical signal after the delay lines in the proposed scheme. The output signal is divided into various branches, each characterized by a delay line

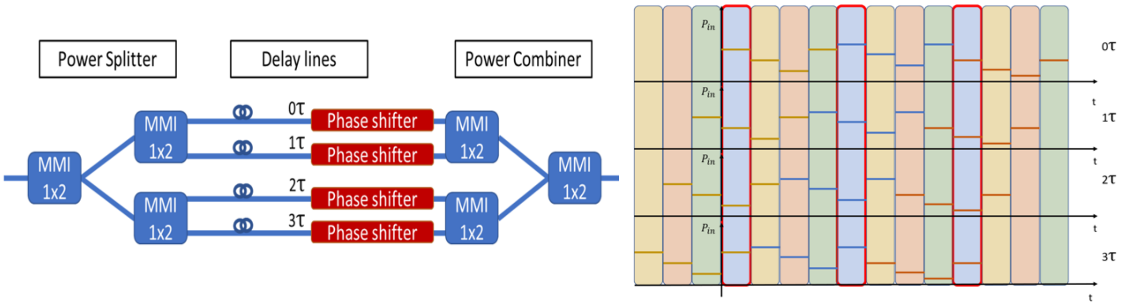

Figure 3: a) Simplified scheme of the integrated optical readout layer. b) Optical signal flow of three inputs over each branch in the delay line stage. We define a time slot (blue) that corresponds to the superposition of the virtual nodes.

multiple of θ, leading to a temporal overlap of the different nodes within the highlighted blue time slot.

2.3 Results

To evaluate the performance of the introduced readout layer, we exploit a 3-bit-header recognition task, fed with a sequence of binary inputs (bits) with an on-top Gaussian noise. The task is to classify the last three input signals into eight different classes. The simulated delay-based reservoir can be implemented with a Mach-Zehnder modulator for the non-linearity and an optical fiber for the delay. The Readout scheme used consists of the previously introduced 16 virtual nodes. The dynamics of this setup are described by the Ikeda-equation. The input period is defined by $T = N \cdot \theta$, where $N = 16$ is the number of nodes and θ = 1 is the normalized temporal separation of the virtual nodes. The delay τ of the reservoir is set equal to the input period T. For the time-multiplexing we use a random mask with values drawn from a uniform distribution u[0,1].

Fig. 4(a) depicts several steps of signal processing using a delay-based reservoir with the proposed integrated readout layer. In Fig.4(a), we show the input signal for the 3-bit header recognition task which will be double-modulated as typical in time-multiplexed reservoir computers. In Fig. 4(b), we display the response of the reservoir for the 10 consecutive input bits. The reservoir output is multiplied by the learned output weights w(t) resulting in the weighted signal shown in c). The weighted states w(t)x(t) enter the simulated integrated PIC and the nodes states are summed up to generate the continuous output signal $\hat{o}$(t). Negative phase weights can be set by applying a π shift with the integrated thermal phase shifters in the PIC, surrounded with deep trenches to maximize the efficiency. Fig.4(e) shows the output sampled signal after summation, which indicates the reservoir prediction $\hat{o}$(k) with respect to the targeted signal $\tilde{o}$(k). For this example, we obtain a bit error rate of 0.004, in line with the state of the art of Reservoir for 16 nodes.

2.4 Conclusion

The results based on numerical simulation show that the PIC readout supports the performance of the reservoir enabling real-time, low-latency signal processing beyond GHz range. Future studies will focus on improving the scalability limited by the modulation bandwidth. Further improvements will be dedicated to replacing thermal phase shifters with optical MEMS or phase change materials to minimize power consumption. Realizing a real-time, scalable, compact,

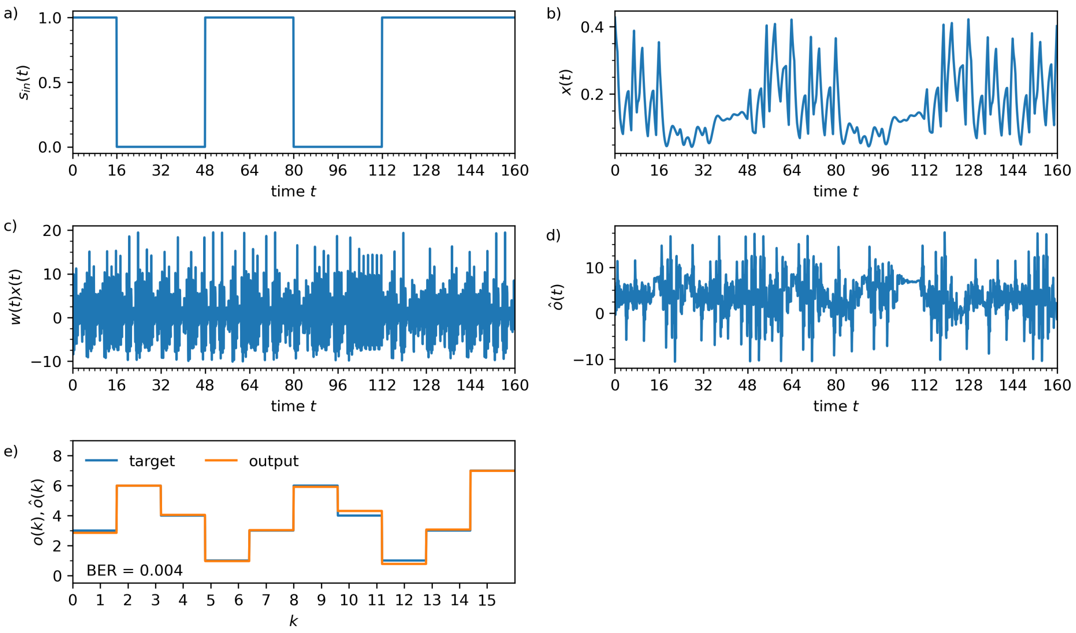

Figure 4: Several steps in the signal processing using a delay-based reservoir and the proposed integrated optical readout layer. In a) the input signal for a 3-bit-header recognition task. B) The reservoir response for 10 consecutive bits. c) The weighted states. d) The PIC output. e) The targeted signal (blue) and the reservoir prediction (orange).

and near-zero power readout interface is of crucial importance in the realization of analog multi-reservoir design.

This work has been published in the conference proceedings of the European Conference on Integrated Optics (ECIO) 2022 Jonuzi et al. [2022] and a second publication is in preparation.

3 Memory engineering using a reversible deep reservoir

3.1 Introduction

Reservoir computing (RC) enables unconventional computing using untrained recurrent neural networks where the training is based on a linear regression of the readout layer only Jaeger [2001]. By applying time multiplexing on an input signal, RC can be implemented using a single nonlinear node with delayed feedback, which enables a straightforward implementation into hardware Appeltant et al. [2011]. Motivated by the success of deep learning, there is an increasing interest in deep architectures relying on the RC scheme. Hierarchical coupled recurrent neural networks can enhance the computation of signals exhibiting multiple time scales, benefiting time series processing Gallicchio et al. [2017,0], Manneschi et al. [2021]. Recently, hardware implementation-friendly deep reservoirs were introduced relying on the delay-based reservoir approach. These systems show promising performance and increased computational speed compared to setups relying on a single layer Penkovsky et al. [2019], Goldmann et al. [2020].

3.2 Methods

In the following, ESR5 analyzes the information processing capabilities of a two-layer architecture that can be arranged in three different configurations. As shown in Figure 5, we consider a delay-based reservoir for each layer. The three configurations arise by using different coupling schemes accomplished by three simple switches S$_1$, S$_2$, and S$_3$. Depending on the state of the switches, we obtain two parallel and independent reservoirs (S$_1$, S$_2$ closed, S$_3$ open) or a deep reservoir, which considers a unidirectional connection between the layers such that the information can flow either from layer 1 to layer 2 or in the reverse direction (S$_1$ closed, S$_3$ closed or S$_2$ closed, S$_3$ closed).

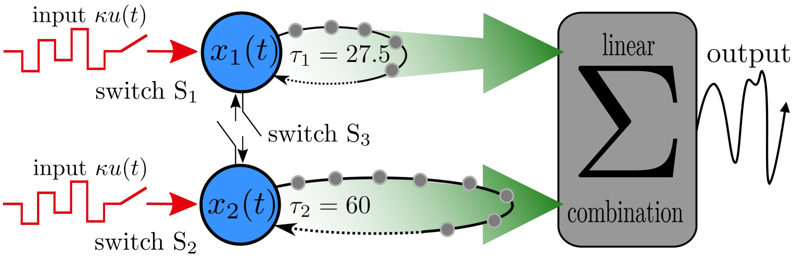

Figure 5: Scheme of two delay-based reservoirs that can be decoupled or unidirectionally coupled by flipping the switches S_1, S_2 and S_3. Each layer contains $N = 80$ virtual nodes, which are all read out and used to train the output layer.

The two layers in Figure 5 might be implemented using, e.g., two optoelectronic Mach-Zehnder modulators with delayed feedback loops. The dynamics of each layer can be described using the following Ikeda equation:

$$\dot{x}_i(t) = -x_i(t) + \beta \sin^2(x_i(t-\tau_i) + \kappa J_i(t) + b), \; i = 1, 2. \tag{1}$$

Without loss of generality, we chose the feedback gain $\beta = 1.6$, the phase offset $b = 0.2$ and the input gain $\kappa = 0.2$, equal in both layers. The only difference between the layers is the length of the delays, which in the first layer ($\tau_1 = 27.5$) is close to the modulation time of the input signal ($T = 25$), and in the second layer $\tau_2 = 60$, more than twice of that. As shown in Goldmann et al. [2020], combining different delay lengths in a deep architecture influences the information processing and, e.g, can be used to increase the performance of the reservoir in forecasting a chaotic time series.

The hierarchical order in the deep reservoir can be rearranged via the switch S_3, such that the first layer drives the second or the second layer drives the first.

$$J_1(t) = \begin{cases} u(t) \text{ for parallel RC \& deep RC 1-2} \\ x_2(t) \text{ for deep RC 2-1} \end{cases} \qquad J_2(t) = \begin{cases} u(t) \text{ for parallel RC \& deep RC 2-1} \\ x_1(t) \text{ for deep RC 1-2} \end{cases}$$

The reservoir is fed by inputs that are randomly drawn from a uniform distribution $u(k) \in [-1, 1]$. Using the response of $N = 80$ nodes per layer, we compute the information processing capacity (IPC) of the three architectures Dambre et al. [2012]. The IPC gives insight into the linear and nonlinear transformations performed by the reservoir and it provides information on how well a reservoir can reproduce past inputs while being driven with uncorrelated data.

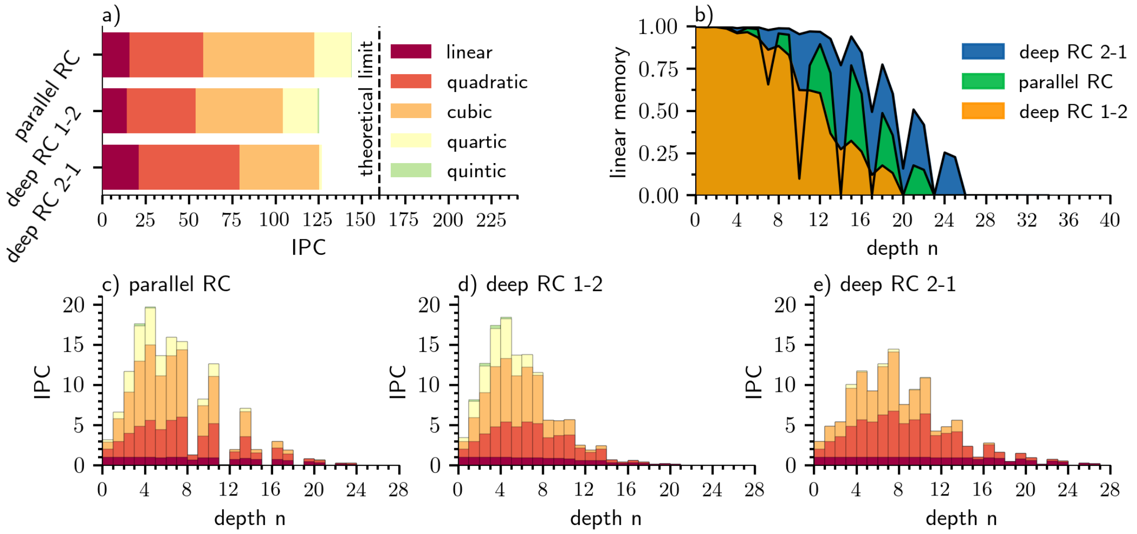

Figure 6: The information processing capacity (IPC) of parallel RC and the two deep configurations. In a) the different degrees of the IPC are color-coded, while the length of the bar represents the total IPC. The red color indicates the linear memory, while from red to green the nonlinearity of the memory increases. In b) the temporal profile of the linear memory is shown for all three configurations. Hereby, a linear memory of one indicates the reservoir can fully reproduce the input $u(k-n)$. In panel c), d) and e), the temporal profile of the different degrees of the IPC is shown. During the computation of the nonlinear IPC, the reservoir also has to reproduce products of inputs at different times into the past, e.g $u(k-3)u(k-6)$. Here, we refer the depth n to the largest distance into the past.

3.3 Results

In Figure 6 a), we show the distribution of the IPC split into linear and nonlinear degrees for the three architectures mentioned above. We find that the IPC of the parallel RC gets close to the theoretical limit that is given by the total number of virtual nodes ($\text{IPC}_{\text{max}} = 160$), whereas the two possible deep reservoirs exhibit a reduction of their total IPC. Nevertheless, it can be seen that the deep RC 2-1 reaches the highest linear memory. By analyzing the temporal profile of the linear memory, as shown in Figure 6 b), we obtain that the deep RC 1-2 has a fast degrading linear memory, and the parallel RC exhibits several gaps in the linear memory arising from the long delay τ_2. The deep RC 2-1, due to the interplay of the layers, can close these linear memory gaps up to $n = 13$ and further the linear memory decays slower than in the other two configurations. In panels c), d), and e) of Figure 6, we show the distribution of all degrees of the IPC over the temporal depth n. In Figure 6 c), the parallel uncoupled RC reveals several specific gaps in the profile where the memory strongly reduces, e.g., at $n = 8, 12$. These gaps can be overcome by the deep RC 1-2, which provides a dense IPC distribution up to $n = 16$ in Figure 6 d). The IPC profile can be extended to longer depths in the case of the deep RC 2-1 at the cost of reducing the presence of nonlinear memory contributions beyond the cubic terms, as shown in Figure 6 e).

3.4 Conclusion

In this study, we demonstrate that reservoir configurations containing two nodes with delayed feedback can be used to engineer different memory profiles. By reversing the hierarchical order of the deep reservoir, one can boost two contrasting properties, either long-lasting linear memory or high nonlinearity. We show how the configurations of the proposed reservoir architectures are modified using simple switches and how the reservoirs can be adapted for tasks with different demands.

The results of our study where presented (online) at the Deep Learning in Unconventional Neuromorphic Hardware workshop of the (virtual) IJCNN 2021 conference (Goldmann et al. [2021]).

4 Finite state computation at the edge of stability

4.1 Introduction

Any theory of post-digital, co-digital or "somehow more general than"-digital computation must give a unifying account of the symbolic processing of digital computers and analog computation associated with dynamical systems. In fact, all available rigorous definitions of computation are to be found in the digital realm, whereas the computational properties of dynamical systems remain characterized as *real-time* and *online* but no structured account of what actually *happens* while a system computes is given.

Finite state machines and network attractors The reason for this discrepancy is undoubtedly the availability of a theoretical model for all symbolic computation - the Turing machine (Turing et al. [1936]), while no such overarching framework exist for dynamical systems in general. While the Turing machine relies on an infinite tape that serves as storage for memory and instruction sets, its finite resource analogue - and hence the model underlying all physically realized computing machinery - is the finite state machine. A finite state machine accepts or rejects a given input string composed of the elements of a finite alphabet. It does so by traversing a finite set of states, with transitions triggered by reading the symbols that make up the string one at a time. A finite state machine is usually visualized as a graph, whose nodes are the states of the machine and transitions are indicated by directed edges with the associated symbols annotated.

A number of studies have attempted to identify principles that allow to carry out the discrete time symbolic computations achieved by digital computers using the dynamics of a continuous dynamical system. They all aim at embedding the graph structure of an FSM into a system's state space. Most notably, the theory of network attractors has been able to provide a mechanism that implements a given finite state machine graph in a type of continuous time recurrent neural network (Ashwin and Postlethwaite [2021]). However, their method still requires a very specific form of equation, while a theory of finite state computation in general dynamical systems should be applicable to any given system.

In this report we connect a certain dynamical regime - the edge of stability - rather than a specific equation to the emergence of network structures possibly supporting symbolic computation.

Self-organization and the edge of stability While the computational power of algorithmic computation devices - and hence finite state machines - increases with the number of possible states, for the sort of cybernetical computation carried out by continuously evolving dynamical systems no clear-cut quantifications of their computational power exist. While attempts have been made to develop measures that assess a system's information processing capacity independently of an input signal (Dambre et al. [2012]), most studies resort to heuristically pointing out dynamical features of the undriven system and make more or less vague claims about their computational benefits. One of the proposed features is the *edge of stability*, a configuration in which a system is close to changing the number and/or stability of its equilibria - a bifurcation point. Much confusion has been spread concerning the general benefits of this dynamical regime. Some clarity has been provided in Yildiz et al. [2012], pointing out that the dynamical regime of an autonomous system is never beneficial in general but always must be considered in relation to the driving input. A system with a fixed point that is barely stable takes a long time to relax back to equilibrium upon a perturbation. It thus has a long linear memory span. Consequentially, for tasks requiring such long memory spans this configuration may be desirable while for others that necessitate strong nonlinear transformations it may not, due to a fundamental trade-off between memory and nonlinearity (Dambre et al. [2012]).

In this work we propose a scenario in which a graph structure is embedded in the state space of a dynamical system at the edge of stability. We do not therefore claim that this is a generally advantageous dynamical regime, but rather that a system near the edge of stability might be able to utilize this graph structure to carry out a certain kind of tasks - namely symbolic processing by means of finite state computation.

4.2 Methods

Here we define the necessary concepts, taken mainly from dynamical systems theory. We briefly treat stability of equilibirum points and elementary bifurcation theory. Then we introduce the notion of a network attractor. Our interest will be particularly in the multiple Hopf bifurcation and associated amplitude equations derived by the method of normal forms. These two concepts will be explained in detail and subsequently a principle of self-organization will be laid out, which may be used to drive a given dynamical system to the neighbourhood of a multiple Hopf bifurcation point.

4.2.1 Equilibria, stability and bifurcation in dynamical systems

Dynamical systems theory is - as the name suggests - concerned with the mathematical description of systems evolving in time. For ordinary differential equations,

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{2}$$

that describe the dynamics of a finite number of state variables collected in the vector $\mathbf{x}$, long-term behaviours are of particular interest because they determine a systems dynamics once initial transients have died out. The simplest such long-term dynamics are those in which the system approaches a steady state, i.e. where

$$\dot{\mathbf{x}^*} = f(\mathbf{x}^*) = \mathbf{0}. \tag{3}$$

The points for which Eq.(3) holds are fixed points of a system. All fixed points have in common that a system, initialized directly at the point will stay there indefinitely. Three different types of fixed points may be distinguished, though, according to what happens when our system gets slightly perturbed from the equilibrium solution. For stable fixed points, perturbations decrease as time increases, hence there exists a neighbourhood of the point in which it attracts all trajectories. The converse is true for unstable fixed points, where an infinitesimal perturbation away from the fixed point grows in magnitude and trajectories are repelled either to a different equilibrium or they diverge to infinity. Finally, a fixed point may be stable to some perturbations and unstable to others. In that case, starting from the point there are directions in which it is attracting, and others in which it repels and we speak of a saddle point. Figure 7 illustrates the three kinds of stabilities discussed for one-dimensional systems. In fact, if there is only a single state variable, these are the only long-term dynamics that can be observed in systems in continuous time. With increasing dimension of the state space, the range of possible dynamics becomes broader and more complicated. Dimension two introduces limit cycles, i.e. periodic orbits through state space that smoothly connect back to themselves. Systems with a limit cycle in state space may display oscillatory behaviour. Similar to fixed points, limit cycles come in stable and unstable variants. A limit cycle, however, can only exist around a fixed point (or around other limit cycles which ultimately surround a fixed point) and stability of the cycle is determined by the stability of the fixed point. Figure 8 illustrates the two possible scenarios: a

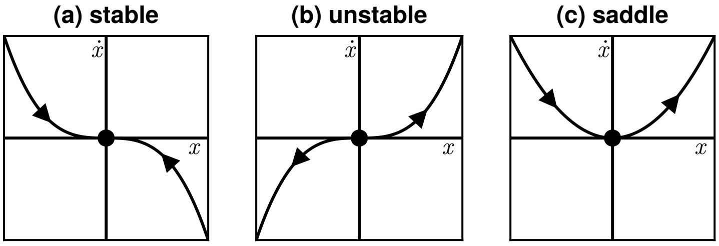

Figure 7: Three different types of stability in one-dimensional systems. (a) Stable fixed point (b) Unstable fixed point. (c) Saddle point.

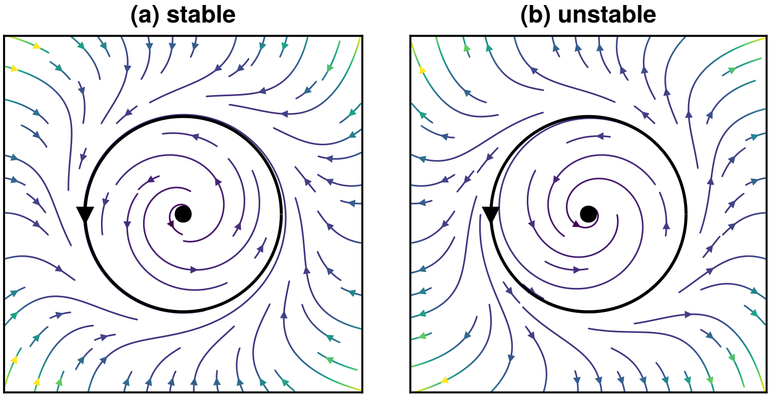

Figure 8: Stability of limit cycles in a 2D system. (a) Stable limit cycle surrounding an unstable fixed point. (b) Unstable limit cycle surrounding a stable fixed point.

single limit cycle must enclose an unstable fixed point if it is itself stable (Fig.8(a)). Conversely, an unstable limit cycle must have a stable fixed point in its interior (8(b)).

In dimension 3 and higher systems might display deterministic chaos. Since in the context of our work chaos is not of immediate relevance, the interested reader is referred to the standard literature, e.g. Strogatz [2018], Guckenheimer and Holmes [2013], Ott [2002].

Linear stability analysis How then, is stability of fixed points assessed? Suppose that for a given system we have found a fixed point $\mathbf{x}^*$ satisfying Eq.(2). To determine its stability properties we want to know how the system reacts to infinitesimal perturbations away from the point. We can find out by zooming in closely to the immediate neighbourhood of the fixed point in question. Whereas the original Equation may possibly be nonlinear and trajectories wind through state space in curves, the further we zoom in around a point, the closer the appearance of these curves will get to straight lines. That means our system will be well approximated by a linear equation for which it is easy to assert whether perturbations grow or decay. Mathematically, the zooming in operation is achieved as follows:

1. Taylor-expand the governing equations around the fixed point of interest
2. Truncate the Taylor expansion to involve only linear terms. Hence, the truncated system takes the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, where $\mathbf{x}$ is the n-dimensional state vector and $\mathbf{A}$ is an $n \times n$ matrix. This is called the local linearization of the system.
3. For a one-dimensional system, the matrix $\mathbf{A}$ is in fact a scalar and we may read off stability from its sign. For higher dimensional systems, however, we have to

4. compute the eigenvalues λ_i of the matrix $\mathbf{A}$. The fixed point is classified as
   (a) **stable** when all eigenvalue real parts $Re(\lambda_i) < 0$.
   (b) **unstable** when all eigenvalue real parts $Re(\lambda_i) > 0$.
   (c) a **saddle-point** when some eigenvalues have negative and others have positive real parts.

Elementary bifurcations So far the state of the systems we considered changed with time along the contours of its state space. Now, we will consider a system that additionally depends on a set of parameters θ. Hence,

$$\dot{\mathbf{x}} = f(\mathbf{x}, \theta). \tag{4}$$

This additional dependence allows state space itself to change as a parameter is varied. In particular, the number and stability of equilibria may change. When this happens we say that our system, or one of its fixed points, undergoes a bifurcation. While there is a whole zoo of bifurcations involving the collision of two fixed points, fixed points and limit cycles or multiple limit cycles, we will for now focus on the elementary bifurcations only, that is those in which a single fixed point loses stability to a novel structure appearing in state space. The appearance of new fixed points or limit cycles in an elementary bifurcation in fact always entails a change of stability for a previous equilibrium state.

From the classification scheme above we can hence already infer a method to detect elementary bifurcations. Since during a bifurcation an equilibrium changes stability, a fixed point bifurcates when, by varying a parameter, an eigenvalue of its local linearization goes from having negative real part to having positive real part. Since eigenvalues are either real or come in complex-conjugate pairs we may differentiate two bifurcations that will be relevant later on. In a saddle-node bifurcation, a stable and an unstable fixed point collide and annihilate each other. Linearizing around the stable of the two fixed points involved, we observe a single eigenvalue changing sign from negative to positive by going through zero on the real line in the complex plane. At the same time an eigenvalue of the unstable fixed point becomes negative. This scenario will be important in the next Section 4.2.2. Of particular importance in the present context is the Hopf bifurcation. In a Hopf bifurcation a pair of complex-conjugate eigenvalues crosses the imaginary line in the complex plane and a fixed point changes stability while a periodic orbit emerges around it. This case will be treated in all detail in Section (4.2.3) below.

4.2.2 Heteroclinic and excitable network attractors

In this Section we give an introduction to the concept of network attractors. Network attractors have been previously introduced to identify and embed network structures in neural networks Ceni et al. [2020], population ecology Hofbauer and Sigmund [1998] and pattern formation Küppers and Lortz [1969]. In Ashwin and Postlethwaite [2021] they have been identified as a mechanism enabling sensible finite state computation in a type of continuous time RNNs.

A network attractor consists of a set of fixed points and dynamical connections among them. Based on the asymptotic stability properties of the set of fixed points one may classify them into heteroclinic and excitable networks (cf. 13 below). In a heteroclinic network all fixed points are of saddle type. Hence, if we concern ourselves with the systems local linearization around each of the points, we find stable subspaces in which the fixed point is attracting and unstable subspaces in which it is repelling. To forge a collection of saddle points into a heteroclinic network, we now require that for every fixed point all of its unstable subspaces are the stable subspace of another. Thereby, the system may move from one fixed point to another along an invariant subspace. The flow in the invariant subspaces of a cyclic heteroclinic network is shown

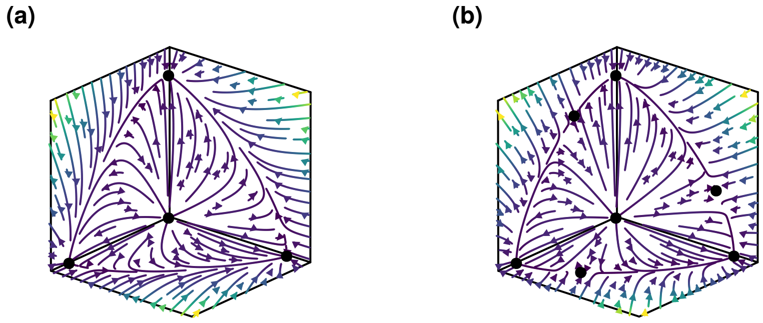

Figure 9: Flow along coordinate planes of a network attractor. (a) A cyclic heteroclinic network. (b) A cyclic excitable network.

in Fig. 9(a). Trajectories will approach each of the saddle points in turn, with rapid transitions among them via the invariant coordinate planes. As the system evolves, trajectories approach each of the fixed points closer and closer and the time spent in their vicinity diverges.

An excitable network consists of fixed points that are locally stable, hence there is no dynamical connection between equilibria as in the heteroclinic case. Thus, left on its own, trajectories from a random initial state will approach one of the fixed points and remain in its vicinity indefinitely. However, a switch may be induced by subjecting the system's evolution to an external signal, which pushes the system's state over the line separating two equilibria's basins of attraction (Fig.9(b)). We speak of an excitable connection from point $\mathbf{x}_i^*$ to $\mathbf{x}_j^*$, when the distance of $\mathbf{x}_i^*$ to the border separating the two point's basins of attraction subceeds a predetermined excitablity threshold. Consequentially, in both the heteroclinic and excitable case only one directed connection is possible per invariant subspace.

The existence of invariant subspaces among which transitions between fixed points may take place is itself not common. However, systems whose governing equations possess symmetries naturally possess invariant subspaces. For example, the system described by Guckenheimer and Holmes [1988] is of the form,

$$\dot{x}_i = x_i \left(l_i + \sum_{j=1}^{n} A_{ij} x_j^2 \right) . \tag{5}$$

This equation is symmetric under permutation of its variables and reflection about the origin. Guckenheimer and Holmes show that this allows for a heteroclinic cycle between three equilibria by imposing constraints on the coefficients A_{ij}.

Heteroclinic and excitable connections are closely related as one may turn into the other via a saddle-node bifurcation. Correspondingly, the same type of Equation with different parameters may implement an excitable network, as demonstrated in Ashwin and Postlethwaite [2021].

4.2.3 The multiple Hopf bifurcation

Of particular interest for the study at hand is a bifurcation scenario in which a stable fixed point becomes unstable and a single or multiple stable periodic orbits emerge around it. The situation in which a single periodic orbit is born is referred to as a supercritical Hopf bifurcation

(Strogatz [2018]). Consequently, we will call a configuration in which multiple periodic orbits arise the multiple Hopf bifurcation. The Hopf bifurcation is characterized by a complex conjugate pair of eigenvalues crossing the imaginary line in the complex plane. Figs.10 and 11 show examples of the ensuing dynamics. There are two cases to be differentiated. In the supercritical case (Figure 10) a stable fixed point loses stability in favor of a stable oscillation around it. Conversely, in the subcritical case (Figure 11 an unstable fixed point becomes stable and an unstable periodic orbit is born around it.

Here we are interested only in the supercritical case. However, given a system at a Hopf bifurcation, how do we determine the stability of the limit cycle and hence if the bifurcation is sub- or supercritical? The classification procedure described in Section 4.2.1 is only applicable for hyperbolic fixed points, i.e. when eigenvalues have nonzero real parts. At the bifurcation point the leading terms of the Taylor expansion carry no information regarding stability. Thus determining the bifurcating limit cycle's stability requires taking into account higher order terms. A systematic way of treating higher order terms is provided by the method of normal forms (Murdock [2003]). For the single Hopf bifurcation it allows to derive equations for the phase and amplitude of the periodic solution. In particular, the resulting amplitude dynamics are described by

$$\dot{r} = r\left(\mu + A_{11} r^2\right) + \mathcal{O}(5). \tag{6}$$

Here, μ denotes the real part of the pair of complex conjugate eigenvalues crossing the imaginary line. The stability of the limit cycle can now be read off the coefficient A_{11}, as the associated term in the equation serves as a nonlinear damping or amplification depending on its sign.

For $A_{11} < 0$ the amplitude dynamics have a stable nonzero fixed point, the limit cycle is thus itself stable and the bifurcation supercritical. The converse is true in the subcritical case. For $A_{11} = 0$, next higher order terms must be consulted.

A similar procedure might be carried out in the case of multiple simultaneously occurring Hopf bifurcations. The resulting amplitude equations then become slightly more involved, as in addition to the self-damping term A_{ii} found already in Equation 6, we find interaction terms A_{ij} :

$$\dot{r}_i = r_i\left(\mu_i + \sum_{j=1}^{n} A_{ij} r_j^2\right) + \text{h.o.t.} \tag{7}$$

The interaction terms couple the amplitude dynamics of the i-th and j-th bifurcating limit cycles. We note that this is essentially equivalent to Eq. (5) introduced in Section 4.2.2. It possesses the same symmetries allowing heteroclinic or excitable connections of fixed points along invariant subspaces of the dynamics. Stable fixed points in the amplitude dynamics, however, translate into stable oscillations of the full system close to the bifurcation point. This mechanism has been proposed in Melbourne [1989] as producing intermittency: approximated to third order, the original system transitions among oscillations according to a network structure encoded into the matrix of interaction terms A_{ij}. Adapting a system to the multiple Hopf bifurcation point with specific values for the interaction terms hence seems like a viable path to embed a graph structure as a network attractor onto a dynamical system at the edge of stability. Furthermore, if the network attractor is excitable, it may allow sensitive finite state computation, where states correspond to oscillations. A self-organized feedback mechanism that drives a given system into the vicinity of a multiple Hopf bifurcation point will be introduced in Section 4.2.4.

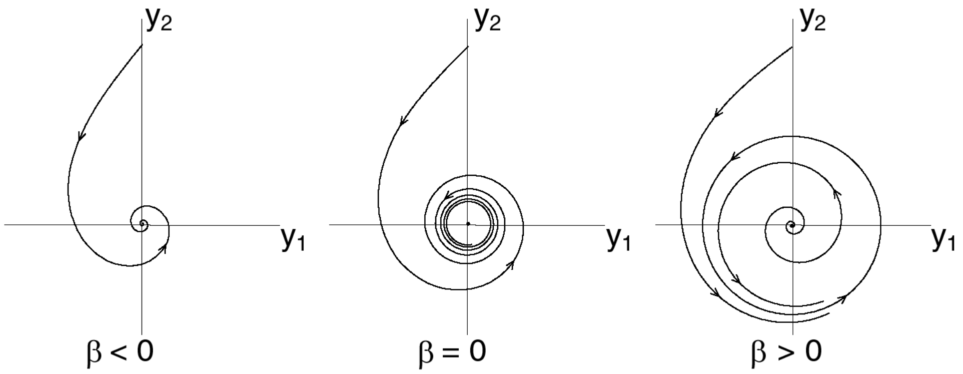

Figure 10: Dynamics in the course of a supercritical Hopf bifurcation. Figure from Kuznetsov [2006].

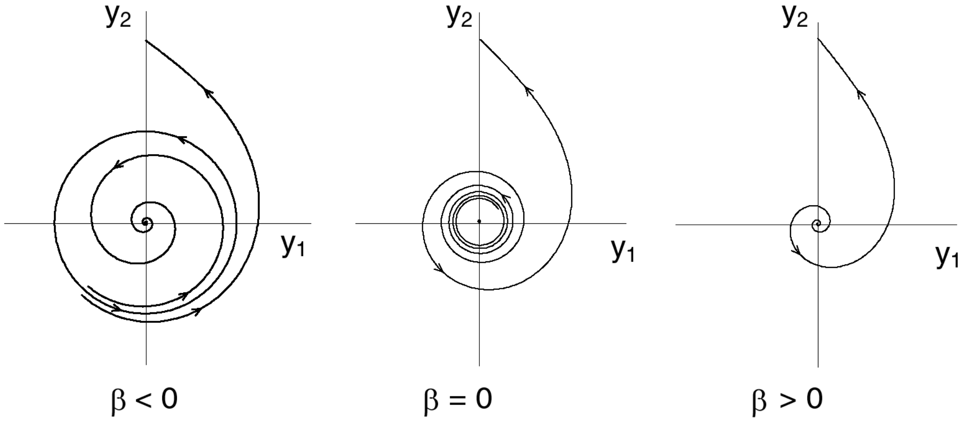

Figure 11: Dynamics in the course of a subcritical Hopf bifurcation. Figure from Kuznetsov [2006].

4.2.4 Self-organization towards the edge of stability

In this Section we will review a mechanism proposed by Magnasco et al. [2009] to drive a simple linear model of a neural network to a critical state. As an instance of unsupervised anti-Hebbian 'synaptic' adaptation the proposed learning rule in conjunction with the neural network model exhibits self-organized criticality. Concretely, the proposed learning rule is set to work on the system

$$\dot{\mathbf{x}} = \mathbf{W}\mathbf{x}, \tag{8}$$

where $\mathbf{W}$ is the matrix of synaptic connection strengths. The learning rule consists of a constant driving term and a balancing anti-Hebbian term:

$$\dot{\mathbf{W}} = \alpha \left(\mathbf{I} - \mathbf{x}\mathbf{x}^T \right). \tag{9}$$

In Magnasco et al. [2009] the authors argue that the constant driving term drives the eigenvalues of $\mathbf{W}$ towards and beyond the critical line where their real part is zero - hence through a bifurcation point. After the associated bifurcation occurred, spontaneous dynamics emerge in the subspace spanned by the corresponding eigenvectors. Subsequently, the anti-Hebbian term will pull the eigenvalues real part back to the subcritical regime and the process begins again. In this manner, after some time all eigenvalues real parts will oscillate in a small strip around 0 with the strips magnitude determined by the time scale parameter α. This entails 'breakout' dynamics where the different modes of the system become unstable in turn and are quickly damped again.

We adopt the learning rule Eq.(9) and adjust it to modify the feedback weights of a small recurrent neural network (RNN) in continuous time. The RNN's update equation are given by

$$\frac{1}{\tau}\dot{\mathbf{x}} = -\mathbf{x} + \tanh\left[(\mathbf{W} + \mathbf{W}_{fb})\mathbf{x}\right] \tag{10}$$

The entries of the weight matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ are drawn from the standard normal distribution and rescaled such that all eigenvalues of $\mathbf{W} - \mathbf{I}$, the Jacobian matrix of Eq.(10) evaluated at the origin, have negative real part. The feedback weights $\mathbf{W}_{fb} \in \mathbb{R}^{n\times n}$ are initialized to be all zero. Since in contrast to Magnasco et al. [2009] our system is nonlinear, a careful tuning of the relations between timescales in Eqs.(10) and (9) is in order. To this end we introduce separate scaling parameters for the two terms in Eq.(9) which now reads

$$\dot{\mathbf{W}} = \alpha\mathbf{I} - \beta\mathbf{x}\mathbf{x}^T. \tag{11}$$

We find empirically that separating the timescales τ, β and α by one order of magnitude respectively keeps the eigenvalue oscillations within reasonable bounds.

4.3 Results

In this Section we report on the dynamics evoked by the self-organized criticality mechanism described above. We first show how our method drives a four neuron RNN in continuous time to the critical regime, where the double Hopf bifurcation occurs. Then we report on some of the structures found in the associated amplitude dynamics and give examples of heteroclinic and excitable connections that we believe provide potential for embeddings of finite state machine graphs.

Figure 12 shows the dynamics of a four neuron RNN in continuous time as described be Eq.(10), where feedback weights are adapted according to the discussed learning rule Eq.(11). While panel (b) depicts the evoked dynamics, panel (a) shows the time evolution of the Jacobian's eigenvalues separated into real and imaginary parts. Similar to what has been discussed for the linear case in Section 4.2.4, we find that real parts approach the critical zero line from below beyond the bifurcation point. Sufficiently greater than zero the plasticity rule's anti-Hebbian term pulls the eigenvalues back to the critical line where the process begins again and hence real parts oscillate in a narrow strip around zero. In contrast to the linear case reported in Magnasco et al. [2009], imaginary parts do not remain unchanged during adaptation, which is undoubtedly attributable to the nonlinearity in our system. In particular, it may happen that complex-conjugate eigenvalues collide on the real line and split off into two real eigenvalues. If this happens our analysis of the amplitude equations becomes invalid as we are not in the vicinity of a Hopf bifurcation point anymore. While this phenomenon deserves some attention in the future, for now we restrict ourselves to the case where eigenvalues stay complex-conjugate for all times.

During adaptation, our system traverses through a multitude of parameterisations as indicated by the time evolution of eigenvalues. As long as two complex-conjugate pairs remain in the neighbourhood of the critical line, the amplitude equations derived via the method of normal forms vary continuously and we may plot the resulting amplitude vector fields. This is done in Figure 13 for the time steps I and II as indicated in Figure 12. Panel (I) shows an excitable connection from the fixed point lying on the r_2-axis to the point lying on the r_1-axis, whereas panel (II) shows a heteroclinic connection in the opposite direction.

Through adaptation of feedback weights we have thus driven our system into a regime where it contains a (very simple, for now) network attractor.

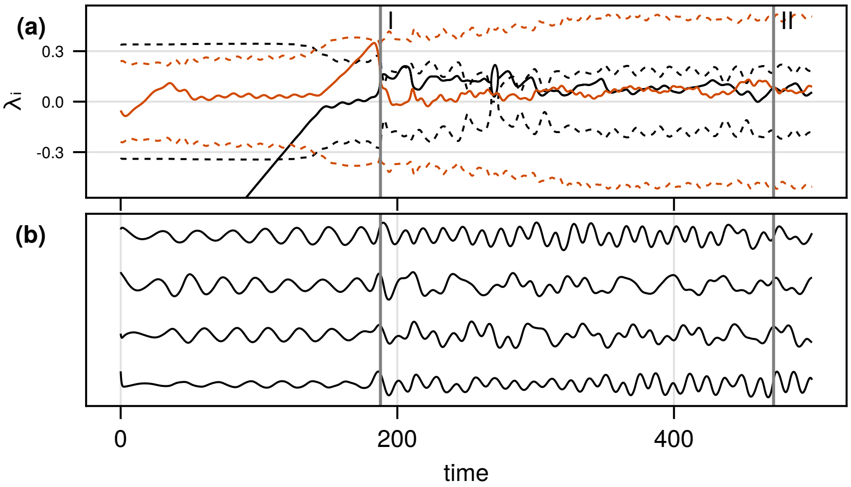

Figure 12: Dynamics under the self-organized plasticity rule Eq.(11)*. (a) Eigenvalues* λ_i *of the Jacobian, real parts as solid lines, imaginary parts as dashed lines. Color code indicates complex-conjugate pairs. (b) Ensuing neural dynamics. Vertical lines mark timesteps for which amplitude dynamics will be investigated further on.*

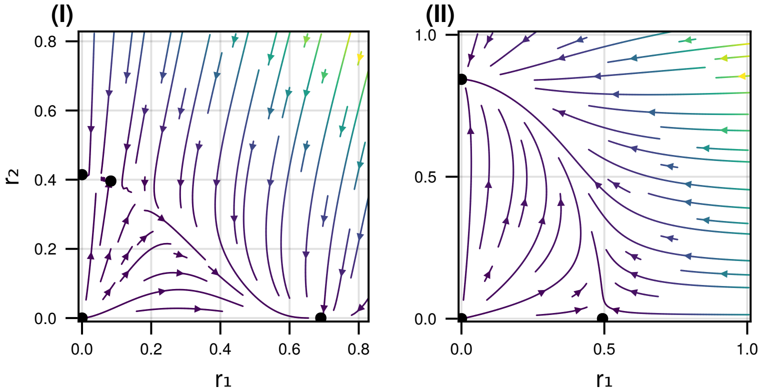

Figure 13: Amplitude vector fields for selected time steps in Fig. 12. (I) Example of an excitable connection. (II) Example of a heteroclinic connection.

4.4 Conclusion

This study constitutes a first proof of concept, that dynamics in networks may arise naturally in a system driven to the edge of stability by means of self-organization. Note, however, that what is happening in the amplitude equations so far is relying on a third order approximation of the original system. A peculiarity of the (multiple) Hopf bifurcation normal form is, that it contains only odd-powered terms. Hence, the equation's symmetry properties allowing for heteroclinic or excitable connections are not lost, but additional constraints must be met when introducing higher order terms.

Furthermore, so far we have successfully driven a system into the vicinity of a multiple Hopf bifurcation point and then merely detected some occurring heteroclinic and excitable connections in the associated amplitude equations. Future work may concern the guided self-organization of a dynamically critical state that yields the desired structure in the interaction terms. At present tuning interaction terms by means of backpropagation and automatic differentiation are investigated. Finally, any useful finite state machine has more than two states and dynamical systems considered to be computationally powerful most definitely have more than four state variables (think: neurons in the brain). For systems with more degrees of freedom it is moreover unrealistic to require that all modes of the linearization around one of its equilibria be critical. Eigenvalues well in the stable regime may not be associated directly to bifurcating solutions of the dynamical system. They do, however, have an influence on the interaction terms in the amplitude equations.

In short, the presented mechanism is a first attempt at shedding light on the potential symbolic computational power of dynamical systems. It remains to be revealed how this mechanism integrates into the zoo of fascinating phenomena that constitute complexity.

This work has been accepted as a poster presentation in the session on *Theory: new concepts and mathematical foundations* at the International conference on neuromorphic, natural and physical computing (Vettelschoss and Dambre [2023]).

References

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*, 2:466–468, 2011.

P. Ashwin and C. Postlethwaite. Excitable networks for finite state computation with continuous time recurrent neural networks. *Biological cybernetics*, 115(5):519–538, 2021.

A. Ceni, P. Ashwin, and L. Livi. Interpreting recurrent neural networks behaviour via excitable network attractors. *Cognitive Computation*, 12:330–356, 2020.

J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar. Information processing capacity of dynamical systems. *Scientific reports*, 2:514, 2012.

M. Freiberger, P. Bienstman, and J. Dambre. A training algorithm for networks of high-variability reservoirs. *Scientific Reports*, 10:14451, 2020.

C. Gallicchio, A. Micheli, and L. Pedrelli. Deep reservoir computing: A critical experimental analysis. *Neurocomputing*, 268:87–99, 2017.

C. Gallicchio, A. Micheli, and L. Silvestri. Local Lyapunov exponents of deep echo state networks. *Neurocomputing*, 298(March):34–45, 2018.

M. Goldmann, F. Köster, K. Lüdge, and S. Yanchuk. Deep time-delay reservoir computing: Dynamics and memory capacity. *Chaos*, 30(9), 2020.

M. Goldmann, C. R. Mirasso, I. Fischer, and M. C. Soriano. Memory engineering using a reconfigurable deep reservoir architecture. In *Deep Learning in Unconventional Neuromorphic Hardware, IJCNN2021 workshop (online presentation)*, 2021. URL https://events.femto-st.fr/DLUNH/.

J. Guckenheimer and P. Holmes. Structurally stable heteroclinic cycles. *Mathematical Proceedings of the Cambridge Philosophical Society*, 103:189–192, 1988.

J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. Springer Science & Business Media, 2013.

J. Hofbauer and K. Sigmund. *Evolutionary games and population dynamics*. Cambridge university press, 1998.

H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report 148, Fraunhofer Institute for Autonomous Intelligent Systems, 2001.

T. Jonuzi, M. Goldmann, A. Argyris, I. Fischer, M. C. Soriano, and D. Doménech. Integrated optical readout layer for ultrafast real-time delay reservoir computing. In *European Conference on Integrated Optics (ECIO)*, page 3. Politecnico di Milano, 2022.

G. Küppers and D. Lortz. Transition from laminar convection to thermal turbulence in a rotating fluid layer. *Journal of Fluid Mechanics*, 35(3):609–620, 1969.

Y. A. Kuznetsov. Andronov-Hopf bifurcation, 2006.

M. O. Magnasco, O. Piro, and G. A. Cecchi. Self-tuned critical anti-hebbian networks. *Physical review letters*, 102(25):258102, 2009.

L. Manneschi, M. O. Ellis, G. Gigante, A. C. Lin, P. Del Giudice, and E. Vasilaki. Exploiting Multiple Timescales in Hierarchical Echo State Networks. *Frontiers in Applied Mathematics and Statistics*, 6(February):1–15, 2021.

I. Melbourne. Intermittency as a codimension-three phenomenon. *Journal of Dynamics and Differential Equations*, 1:347–367, 1989.

J. A. Murdock. *Normal forms and unfoldings for local dynamical systems*. Springer, 2003.

E. Ott. *Chaos in dynamical systems*. Cambridge university press, 2002.

B. Penkovsky, X. Porte, M. Jacquot, L. Larger, and D. Brunner. Coupled Nonlinear Delay Systems as Deep Convolutional Neural Networks. *Physical Review Letters*, 123(5), 2019.

S. H. Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

A. M. Turing et al. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.

B. Vettelschoss and J. Dambre. Finite state computation at the edge of stability. In *Proceedings of the International conference on neuromorphic, natural and physical computing*, page to appear, 2023.

I. B. Yildiz, H. Jaeger, and S. J. Kiebel. Re-visiting the echo state property. *Neural networks*, 35:1–9, 2012.

Consortium

AU	Aston University	United Kingdom
CSIC	Agencia Estatal de Consejo Superior de Investigaciones Cientificas	Spain
UB	Université Bourgogne Franche-Comté	France
IBM	IBM Research Labs Zurich	Switzerland
IMEC	IMEC	Belgium
LO	LightOn	France
ULB	Université Libre de Bruxelles	Belgium
VLC	VLC Photonics	Spain
TH	Thales SA	France
RUG	University of Groningen	The Netherlands
IL	IniLabs	Switzerland
UG	University of Gent	Belgium
UIB	Universitat de les Illes Balears	Spain

Disclaimer

All information provided reflects the status of the Post-Digital project at the time of writing and may be subject to change.

Neither the Post-Digital Consortium as a whole, nor any single party within the Post-Digital Consortium warrant that the information contained in this document is capable of use, nor that the use of such information is free from risk. Neither the Post-Digital Consortium as a whole, nor any single party within the Post-Digital Consortium accepts any liability for loss or damage suffered by any person using the information.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

Copyright Notice