Time-delay reservoir computers: nonlinear stability of functional differential systems and optimal nonlinear information processing capacity. Applications to stochastic nonlinear time series forecasting.

Lyudmila Grigoryeva[1], Julie Henriques[2], Laurent Larger[2], **Juan-Pablo Ortega**[3,4]

[1]Universität Konstanz, Germany

[2]Université Bourgogne Franche-Comté, France

[3]Universität Sankt Gallen, Switzerland

[4]CNRS, France

University of St.Gallen

Outline

1. Machine learning in a nutshell:
   - Discrete vs continuous time
   - Deterministic vs stochastic
2. Static problems, neural networks, and approximation theorems
3. Dynamic problems and reservoir computing
4. Universality theorems
   - The control theoretical approach
   - The filter/operators approach
5. Time-delay reservoir computers
   - Hardware realizations, scalability, and big data compatibility
   - Models and performance estimations
6. Application examples

University of St.Gallen

Machine learning in a nutshell

- We approach to machine learning as an input/output problem.
  - Input: it is denoted by the character **z**. It contains available information for the solution of the problem (historical data, explanatory factors, features of the individuals that need to be classified).
  - Output: denoted generically by **y**. Contains the solution of the problem (forecasted data, explained variables, classification results).
- Purely empirical approach not based on first principles but on a training/testing routine.
- We distinguish between static/discrete-time and continuous-time setups and between deterministic and stochastic situations since they lead to very different levels of mathematical complexity.

University of St.Gallen

Examples

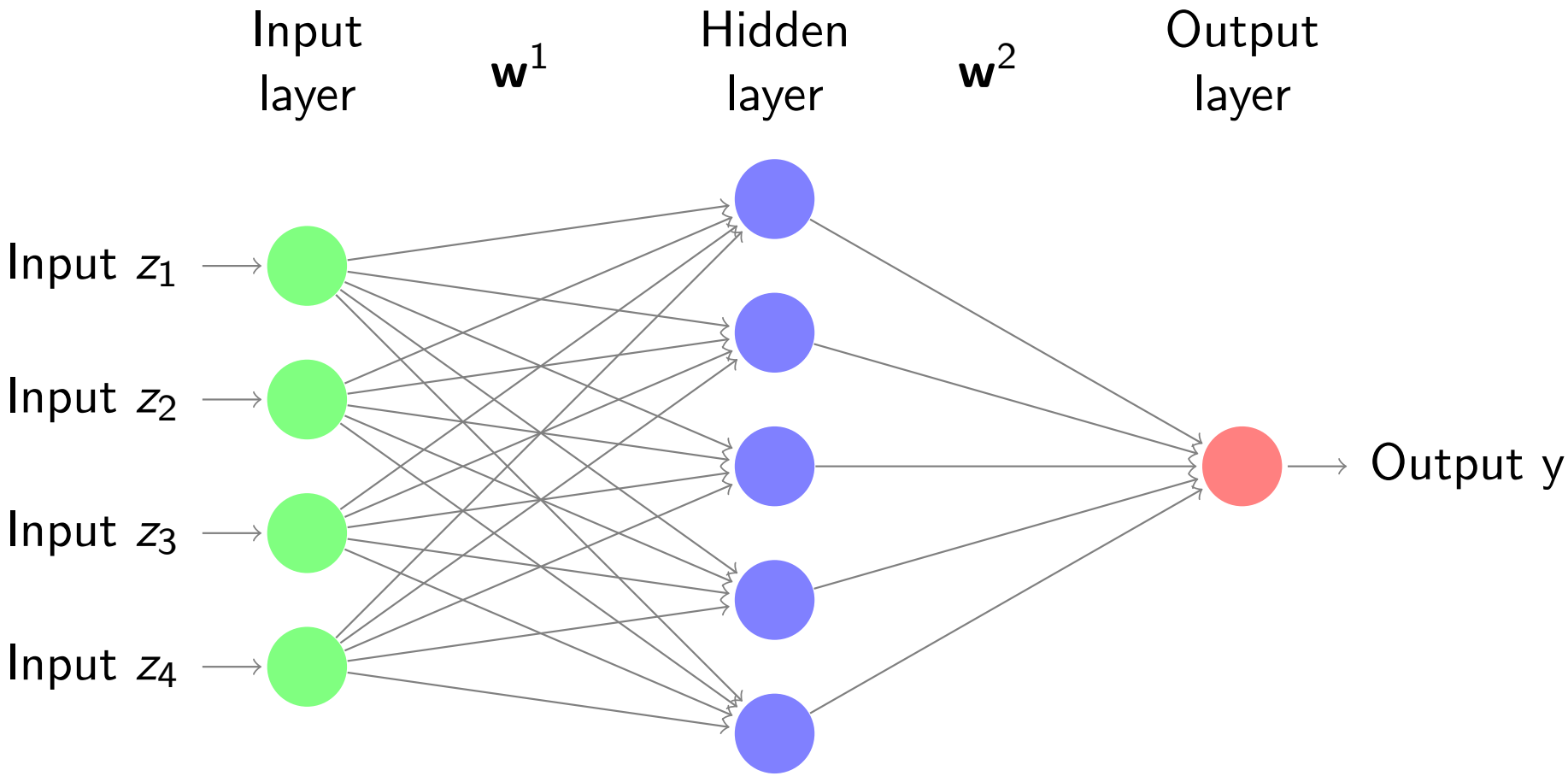
- **Deterministic setup**: an explicit functional relation (via a just measurable function) is assumed between input and output.
  - Static/Discrete-time: observables or diagnostics variables in complex physical or noiseless engineering systems (domotics), translators, memory tasks, games.
  - Continuous time: integration or path continuation of (chaotic) differential equations: molecular dynamics, structural mechanics, vibration analysis, space mission design. Autopilot systems, robotics.
- **Stochastic setup**: the input and the output are random variables or processes and only probabilistic dependence is assumed between them.
  - Static/Discrete-time: image classification, speech recognition, time series forecasting, volatility filtering, factor analysis.
  - Continuous time: physiological time series classification, financial bubble detection.

Setups considered

	Static/Discrete time		Continuous time	
	Deterministic	**Stochastic**	**Deterministic**	**Stochastic**
Characterization of ingredients	$\mathbf{z} \in \mathbb{R}^n$ $\mathbf{y} \in \mathbb{R}^q$	$\mathbf{z} \in (L^2(\Omega, \mathcal{F}, \mathbb{P}))^n$ $\mathbf{y} \in (L^2(\Omega, \mathcal{F}, \mathbb{P}))^q$	$\mathbf{z} \in C^\infty([a, b], \mathbb{R}^n)$ $\mathbf{y} \in C^\infty([a, b], \mathbb{R}^n)$	$\mathbf{z}$ and $\mathbf{y}$ are $\mathbb{R}^n$ and $\mathbb{R}^q$-valued processes adapted with respect to a given filtration $\mathcal{F}$
Problem to be solved	$\mathbf{y} = f(\mathbf{z})$ f measurable	$\mathrm{E}[\mathbf{y} \mid \mathbf{z}]$	$\mathbf{y}(\cdot) = F(\mathbf{z}(\cdot))$	$\mathrm{E}[\mathbf{y}(\cdot) \mid \mathbf{z}(\cdot)]$
Object to be trained	Real/complex function	Conditional expectation	Functional/Operator Causal Filter	Stochastic Causal Filter
Approach and source of Universality	Approximation theory	(Semi)-parametric statistics Kalman filter	Control theory Stone-Weierstraß	Functional data analysis and Stochastic control theory

University of St.Gallen

Neural networks



$$y = \psi\left(\sum_{i=1}^{5} w_i^2 \psi\left(\sum_{j=1}^{4} w_{ij}^1 z_j\right)\right), \ \psi \text{ sigmoid function.} \qquad (1)$$

University of St.Gallen

Universality in neural networks and approximation theorems

- Neural networks are implemented as a machine learning device by tuning the weights $\mathbf{w}^i$ using a gradient descent algorithm (backpropagation) that minimizes the approximation error based on a training set.
- In the deterministic case, the objective is to recover an explicit functional relation between input and output.
- In the absence of noise there is not danger of overfitting.

University of St.Gallen

- **Universality problem:** how large is the class of input-output functions that can be generated using feedforward neural networks as in (1)?
- **Hilbert's 13th problem** on multivariate functions: can any continuous function of three variables be expressed as a composition of finitely many continuous functions of two variables? This question is a generalization of the original problem for algebraic functions posed in the 1900 ICM in Paris and in [Hil27]

The Kolmogorov-Arnold representation theorem and Kolmogorov-Sprecher networks

Theorem (Kolmogorov-Arnold [Kol56, Arn57])

There exist fixed continuous increasing functions $\varphi_{p,q}(x)$ *on* $I = [0,1]$ *such that each continuous function* f *on* I^n *can be written as*

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} g_q \left(\sum_{p=1}^{n} \varphi_{pq}(x_p) \right)$$

where the g_q *are properly chosen continuous functions of one variable.*

- This amounts to saying that the only genuinely multivariate function is the sum!
- This is a representation and not an approximation theorem

University of St.Gallen

Theorem (Sprecher [Spr65, Spr96, Spr97])

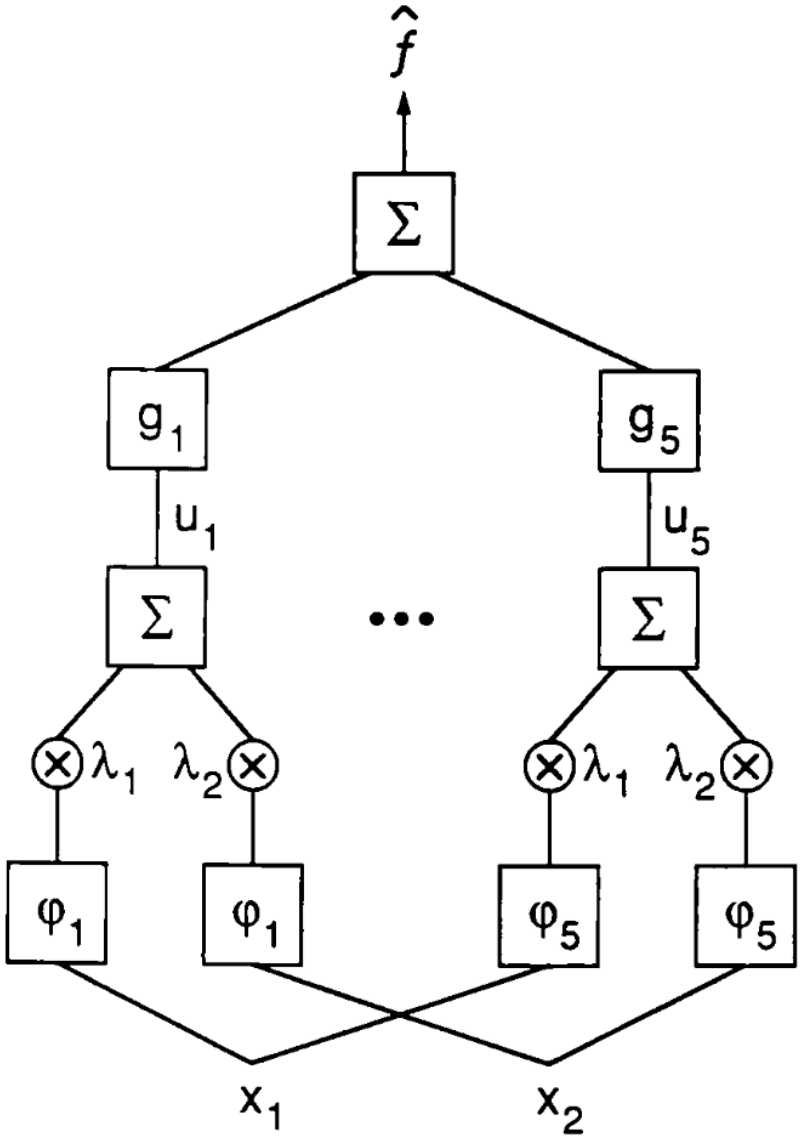
There exist constants λ_p and fixed continuous increasing functions $\varphi_q(x)$ on $I = [0, 1]$ such that each continuous function f on I^n can be written as

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} g_q \left(\sum_{p=1}^{n} \lambda_p \varphi_q(x_p) \right)$$

where the g_q are properly chosen continuous functions of one variable.

- The g_q functions depend on f but not λ_p and φ_q. All the information contained in the multivariable continuous function f is contained in the single variable continuous functions g_q.
- This is not ideal for machine learning applications because we would need to train the g_q functions. It still can be done (see the CMAC in [CG92])

University of St.Gallen



The Kolmogorov-Sprecher network (taken from [CG92])

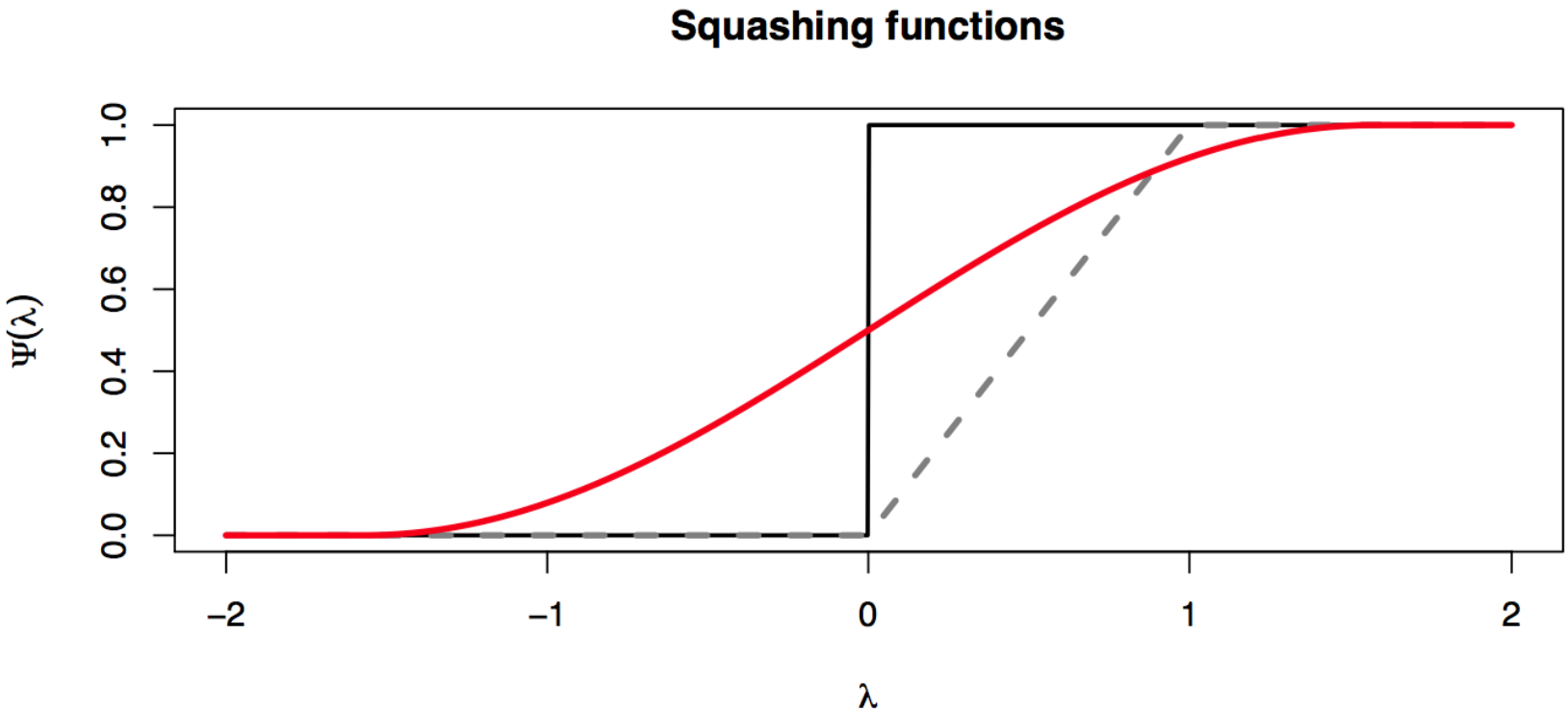
University of St.Gallen

The Cybenko and the Hornik *et al.* theorems

Definition

A squashing function is a map $\psi : \mathbb{R} \to [0, 1]$ that is non-decreasing and that

$$\lim_{\lambda \to -\infty} \psi(\lambda) = 0 \quad \text{and} \quad \lim_{\lambda \to \infty} \psi(\lambda) = 1$$



St.Gallen

Approximation of continuous functions

Theorem (Cybenko [Cyb89])

Let ψ be a continuous squashing function. Then, the functions $G_{\psi,N} : I^n \to \mathbb{R}$ of the form

$$G_{\psi,N}(\mathbf{z};\boldsymbol{\theta}) = \left(\sum_{1=1}^{N} w_j^2 \psi\left(\langle \mathbf{w}_j^1, \mathbf{z}\rangle + \theta_j\right) \right), \quad \mathbf{w}_j^1, \mathbf{z} \in \mathbb{R}^n, \mathbf{w}^2 \in \mathbb{R}^N, \theta_j \in \mathbb{R},$$

are dense in $C(I^n)$, that is, given any function $f \in C(I^n)$ and $\epsilon > 0$, there is a sum of this type for which

$$|G_{\psi,N}(\mathbf{z};\boldsymbol{\theta}) - f(\mathbf{z})| < \epsilon, \quad \text{for all} \quad \mathbf{z} \in I^n.$$

- This result proves that any continuous function can be **approximated** using a feedforward neural network with a single hidden layer if we use a given continuous activation function.

University of St.Gallen

Approximation of measurable functions and functions with finite support

The Hornik, Stinchcombe, and White [HSW89] theorems:

- The previous theorem holds even if the function f is only measurable and the activation function is a not necessarily continuous squashing function.
- Functions with finite support can be exactly attained using a feedforward neural network with a single hidden layer if the activation function attains 0 and 1: let $\{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ be a set of distinct points in $\mathbb{R}^n$ and let $f : \mathbb{R}^n \to \mathbb{R}$ be an arbitrary function, then there exists a feedforward neural network with k neurons in its hidden layer and transfer function $G_{\psi,N}$ such that $G_{\psi,N}(\mathbf{z}_i; \boldsymbol{\theta}) = f(\mathbf{z}_i)$.
- The network can be trained so that it learns not only the function but also its derivatives [HSW90, GW92].

This result has been extended to backpropagation (as opposed to feedforward) neural networks by Hecht-Nielsen [HN89].

The Maurey-Jones-Barron Theorem

- Let G be a set of approximating functions: splines with free nodes, trigonometric polynomials with free frequencies, feedforward neural networks.
- The **variable basis approximation** consists of using the set

$$\operatorname{span}_n G := \left\{ \sum_{i=1}^{n} w_i g_i \mid w_i \in \mathbb{R}^n,\ g_i \in G \right\}.$$

- When G is a subset of a normed linear space $(X, \|\cdot\|)$, we use the G-variation

$$\|f\|_G := \inf\{c > 0 \mid f/c \in \operatorname{cl}\operatorname{conv}(G \cup -G)\}.$$

University of St.Gallen

Theorem

$(X, \|\cdot\|)$ a Hilbert space, G a bounded subset and $s_G = \sup_{g\in G} \|g\|$. For every $f \in X$ and every positive integer n

$$\|f - \mathrm{span}_n G\| \leq \sqrt{\frac{(s_G\|f\|_G)^2 - \|f\|^2}{n}}.$$

- Any function in a ball of radius r in G-variation can be approximated by a neural network with n hidden units computing functions from G within accuracy $r/\sqrt{n}$.
- This estimate holds for any number of variables: **no curse of dimensionality**.

University of St.Gallen

Implementation

It involves three main issues:

1. Choice of an architecture: squashing function, number of layers, number of neurons in each layer, and connectivity between them.
2. Estimation of the connectivity weights: a supervised learning approach is taken. Realizations of the input and the output are used to minimize an error function via a gradient descent method.
   Potential problems:
   - Local minima
   - Flat gradients in deep structures
3. Cross validation and regularization: a posteriori verification of the goodness of the architecture choice in the first point regarding:
   - Deterministic case: is this the most economic structure for a prescribed accuracy level in the approximation problem?
   - Stochastic case: are we overfitting?

   In both cases the solution is obtained using new architectures selected via cross-validation or pruning techniques (see [KvD03] for references and [SX99, SCHU16] for Lasso related approaches).

Non-linear regressions

The deterministic universal approximation properties of neural networks yield non-parametric estimators for non-linear regression functions. Consider the following heteroscedastic regression model:

$$y_t = f(\mathbf{z}_t) + \varepsilon_t, \quad \{\mathbf{z}_t\} \sim \mathrm{IID}(p(\mathbf{z})), \quad \varepsilon_t | (\mathbf{z}_t = \mathbf{z}) \sim \mathrm{IID}(0, s_\varepsilon^2(\mathbf{z}) < \infty), \quad t = 1, \ldots, T, \tag{2}$$

and assume that the functions $f, s_\varepsilon^2 : \mathbb{R}^n \to \mathbb{R}$ are continuous and bounded. Notice that hypotheses (2) imply that, in this case,

$$\mathrm{E}\left[y_t | \mathbf{z}_t\right] = f(\mathbf{z}_t).$$

In order to estimate the regression function f, we fit a neural network with a hidden layer and a sufficiently large number N of neurons using two realizations $\{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ and $\{y_1, \ldots, y_T\}$ of the input and the output, which yields the following estimator $\widehat{\boldsymbol{\theta}}_T$ of the weights vector $\boldsymbol{\theta}$:

$$\widehat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \left\{y_t - G_{\psi,N}(\mathbf{z}_t; \boldsymbol{\theta})\right\}^2. \tag{3}$$

University of St.Gallen

Under appropriate conditions $\widehat{\boldsymbol{\theta}}_N$ converges in probability for $T \to \infty$ and for a fixed N to the parameter vector $\boldsymbol{\theta}_N$ which corresponds to the best approximation of $f(\mathbf{z})$ by a function of type $G_{\psi,N}(\mathbf{z};\boldsymbol{\theta})$, that is,

$$\boldsymbol{\theta}_N = \arg\min_{\boldsymbol{\theta}} = \mathrm{E}[\{f(\mathbf{z}_t) - G_{\psi,N}(\mathbf{z}_t;\boldsymbol{\theta})\}^2]. \tag{4}$$

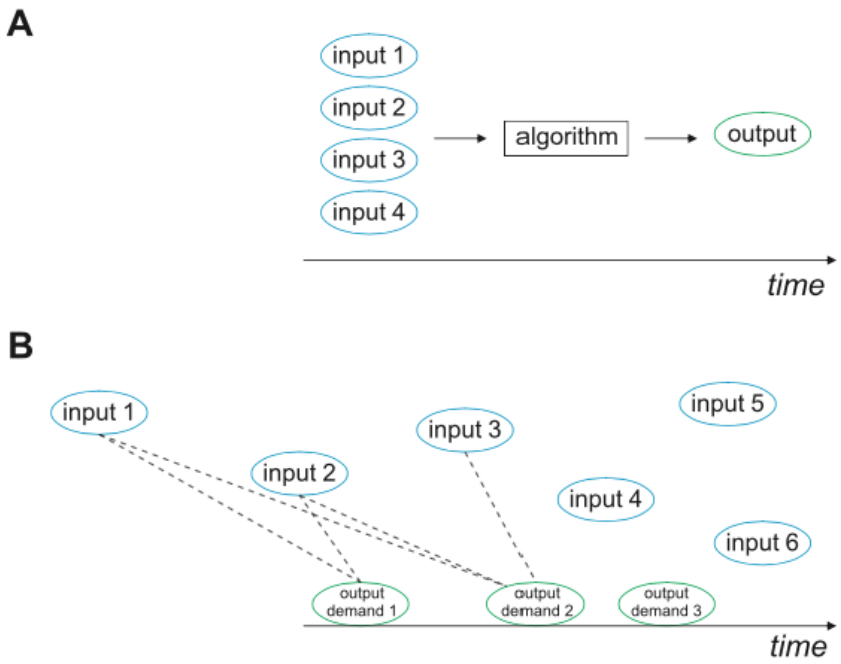
Under somewhat stronger assumptions, it can be shown the asymptotic normality of the estimator $\widehat{\boldsymbol{\theta}}_N$ [FN00].

Remark: Many important examples like nonlinear state space models (ARSV for example) do not satisfy the independence hypothesis on the input signal $\mathbf{z}_t$ or there is just no function f which makes necessary the use of other tools like the nonlinear Kalman filter.

University of St.Gallen

Offline and online computing

- Turing machines compute sequentially and offline batches of information. Computations have a beginning and an end.
- Neuronal and "behaving" systems compute online as information arrives (probably desynchronized and with different sampling frequencies) and reuse the result of previous computations



Offline and online computations (taken from [Maa11])

Mathematical formulation of reservoir computing

Reservoir computing is based on three main principles:

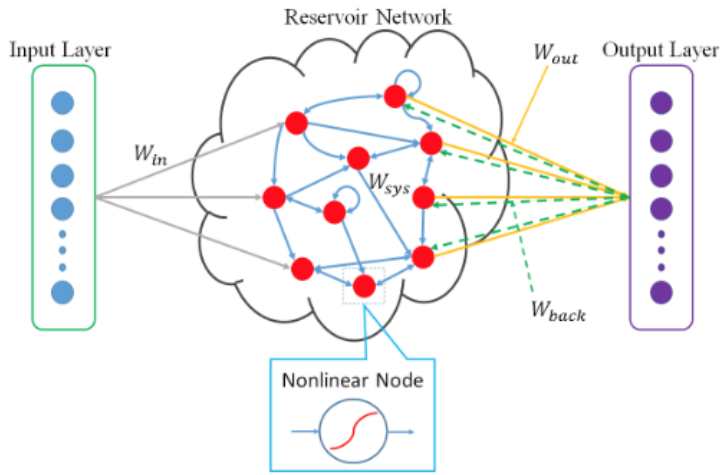
- The input signal $\mathbf{z}(t) \in \mathbb{R}^n$ is inserted as the external forcing of the flow $F_t : \mathbb{R}^N \times \mathbb{R}^n \to \mathbb{R}^N$ of a non-autonomous dynamical system (the **reservoir**):

$$\mathbf{x}(t) = F_t(\mathbf{x}_0, \mathbf{z}(t)). \tag{5}$$

  The value $\mathbf{x}(t)$ is the **reservoir state** at time t.
- A static readout $h : \mathbb{R}^N \to \mathbb{R}^q$ is trained in order to obtain the desired output $\mathbf{y}(t)$ out of the input $\mathbf{z}(t)$:

$$\mathbf{y}(t) = h(\mathbf{x}(t)).$$

- **Multitasking:** different readouts can be trained on the same reservoir output in order to extract different pieces of information about the input.



- Fundamentally new approach to neural computing [Jae01, JH04, MNM02, VSDS07, LJ09]; defining features of RC: the fading-memory, separation, and approximation properties [LJ09]
- Modification of the traditional RNN in which the architecture and the neuron weights of the network are created in advance (for example randomly) and remain unchanged during the training stage
- If readout layer is linear then inference and theoretical performance evaluation becomes possible!!

University of St.Gallen

Reservoir computing and neural processes

- The reservoir computing approach approach resembles neural processes in which sensory inputs (input signals) are pre-processed by the neural microcircuits of a cortical column and then various single neurons (readouts) extract information from it and send it to other brain areas.
- The use of different readouts serves different computational goals. Example: in the case of the visual cortex determine, size, direction of motion, identity of objects.
- The division of information processing between reservoir and readout is very efficient (one processing serves several computational goals) and helps explaining the energy efficiency of the brain.
- Neurophisiology evidence: spike trains coming from different projection neurons from the same cortical column tend to be weakly correlated.

Universality result for neural circuits [MJS07]

Suppose that we are given an external continuous-time input $z(t)$ and a solution $u(t)$ of a non-autonomous nth-order differential equation of the form

$$y^{(n)}(t) = G(y(t), y'(t), y''(t), \ldots, y^{(n-1)}(t)) + z(t). \tag{6}$$

Then, for any non-autonomous dynamical system of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t)) \cdot v(t), \qquad f, g : \mathbb{R}^n \to \mathbb{R}^n, \tag{7}$$

that has the fading memory property (see below), there exist a feedback $K : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ and a smooth readout $h : \mathbb{R}^n \to \mathbb{R}$ such that any solution $y(t)$ of (6) can be written as $y(t) = h(\mathbf{x}(t))$ with $\mathbf{x}(t)$ the solution of the system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t)) \cdot K(\mathbf{x}(t), z(t) + z_0(t)), \qquad \mathbf{x}(0) = \mathbf{0},$$

with $z_0(t)$ a fixed input that satisfies that $z_0(t) = 0$ for all $t \geq 1$.

University of St.Gallen

- The proof of this result is control-theory based.
- The feedback K and the readout h depend only on the function G that characterizes the system that needs to be simulated but not on the external output $z(t)$ that needs to be processed.
- Since these two functions are static, they are ideal targets for learning.
- The function h is chosen in many situations to be just linear and training is carried out by solving a simple (regularized) regression problem.
- This result shows that RCs constructed using the solutions of dynamical systems of the form (7) have the computational power of a universal Turing machine when put together with suitable feedback and readout functions. This follows from the fact that every Turing machine can be simulated by systems of equations of the form (6) (see [Bra95, SS94, SS92, Orp97]).
- The dynamical systems (7) include as a particular case the standard systems of nonlinear differential equations that are used to model the dynamics of firing rates in recurrent circuits of neurons.

The filtering/operator point of view [MNM02]

We now use operators $L : C^0(\mathbb{R}^n) \to C^0(R^N)$ instead of flows F_t like in (5) in order to transform the input signal into the reservoir state curve:

$$L(\mathbf{z}(\cdot)) = \mathbf{x}(\cdot).$$

- **Time invariance:** Let $U^n_{t_0}$ the time-shift operator for curves in $\mathbb{R}^n$, that is $U^n_{t_0}(\mathbf{z}(\cdot))(t) = \mathbf{z}(t + t_0)$. The filter L is called time invariant if

$$L \circ U^n_{t_0} = U^N_{t_0} \circ L$$

- **Causality:** L is causal if $L(\mathbf{z}(\cdot))(t)$ does not depend on $z(s)$ for $s > t$.
- **Fading memory property:** $L(\mathbf{z}(\cdot))(0)$ can be approximated by the outputs $L(\mathbf{u}(\cdot))(0)$ for any other input $\mathbf{u}$ that approximates $\mathbf{z}$ on a sufficiently long time interval $[-T, 0]$ going back into the past.

- Equivalently, in order to compute the most significant bits of $L(\mathbf{z}(\cdot))(0)$ it is not necessary to know the precise value of the input function $\mathbf{z}$ for any time s and it is also not necessary to know anything about the values of $\mathbf{z}$ for more than a finite time interval back into the past.
- Fading memory filters are automatically causal.
- The category of time-invariant fading memory filters is large and includes well-known examples like Volterra series; it can actually be shown [BC85, MS00] that any time-invariant fading memory filter can be approximated by a (possibly infinite) Volterra series.
- **Separation property:** a class $\mathcal{L}$ of filters has the separation property if for any two inputs $\mathbf{z}$ and $\mathbf{u}$ such that $\mathbf{z}(s) \neq \mathbf{u}(s)$, for some $s \leq t$, there exists $L \in \mathcal{L}$ such that $L(\mathbf{z})(t) \neq L(\mathbf{u})(t)$.

University of St.Gallen

Universality theorem [MM04]

Theorem

Let F be an arbitrary time-invariant filter that satisfies the fading memory property. Assume the availability of a space of fading memory filters $\mathcal{L}$ that satisfies the pointwise separation property.
Then, for any chosen accuracy, there exists $m \in \mathbb{N}$, filters $L_1, \ldots, L_m$ in the space $\mathcal{L}$, and a readout function $h : \mathbb{R}^m \to \mathbb{R}$ such that F can be approximated by the composition $h \circ (B_1, \ldots, B_m)$.

University of St.Gallen

Observations

- **Examples**
  - Dynamic networks [MS00]: feedforward neural networks with time-varying weights.
  - Synaptic dynamical models by Tsodyks, Pawelzik, and Markram [TPM98].
- Remarkable consequence: for a large variety of classes $\mathcal{L}$ of basis filters (such as delay lines, linear filters, dynamic synapses, or circuits with fading memory) the pointwise separation property, in combination with sufficiently "flexible" readout maps, endows the resulting RC with universal computational power in the giant class of filters $\mathcal{F}$ that are time-invariant and have fading memory.
- Fading memory filters only generate fading memory filters. Not the case using the neural circuit approach. Major computational jump.
- Proof goes via the Stone-Weierstrass theorem.

University of St.Gallen

Time-delay reservoir computers

TDRs are based on the interaction of the discrete input signal $z(t) \in \mathbb{R}$ with the solution space of a TDDE of the form

$$\dot{x}(t) = -x(t) + f(x(t-\tau), I(t), \boldsymbol{\theta}), \tag{8}$$

where f is a nonlinear smooth function (**nonlinear kernel**), $\boldsymbol{\theta} \in \mathbb{R}^K$ is the parameter vector, $\tau > 0$ is the **delay**, $x(t) \in \mathbb{R}$, and $I(t) \in \mathbb{R}$ is obtained via temporal multiplexing of the input signal $z(t)$ over the delay period.

The choice of nonlinear kernel f is determined by the physical implementation; consider two parametric sets of kernels:

- Mackey-Glass [MG77]: $f(x, I, \boldsymbol{\theta}) = \frac{\eta(x+\gamma I)}{1+(x+\gamma I)^p}$, $\boldsymbol{\theta} = (\eta, \gamma, p)$
- Ikeda [Ike79]: $f(x, I, \boldsymbol{\theta}) = \eta \sin^2(x + \gamma I + \phi)$, $\boldsymbol{\theta} = (\eta, \gamma, \phi)$

Used in the RC electronic [ASV+11] and optoelectronic [LSB+12] realizations.

Discrete time model of TDR

Consider the Euler time-discretization of (8) with integration step $d := \tau/N$:

$$(x(t) - x(t-d))/d = -x(t) + f(x(t-\tau), I(t), \boldsymbol{\theta}). \tag{9}$$

Define **neuron layers** $\mathbf{x}(t)$ and **input layers** $\mathbf{I}(t) := C\mathbf{z}(t) \in \mathbb{R}^N$ by setting

$$x_i(t) := x(t\tau-(N-i)d), \quad I_i(t) := I(t\tau-(N-i)d), \quad i \in \{1,\dots,N\}, \quad t \in \mathbb{Z},$$

where $x_i(t)$ is the i**th neuron value of the** t**th layer of the reservoir**. Then the solutions of (9) are given by

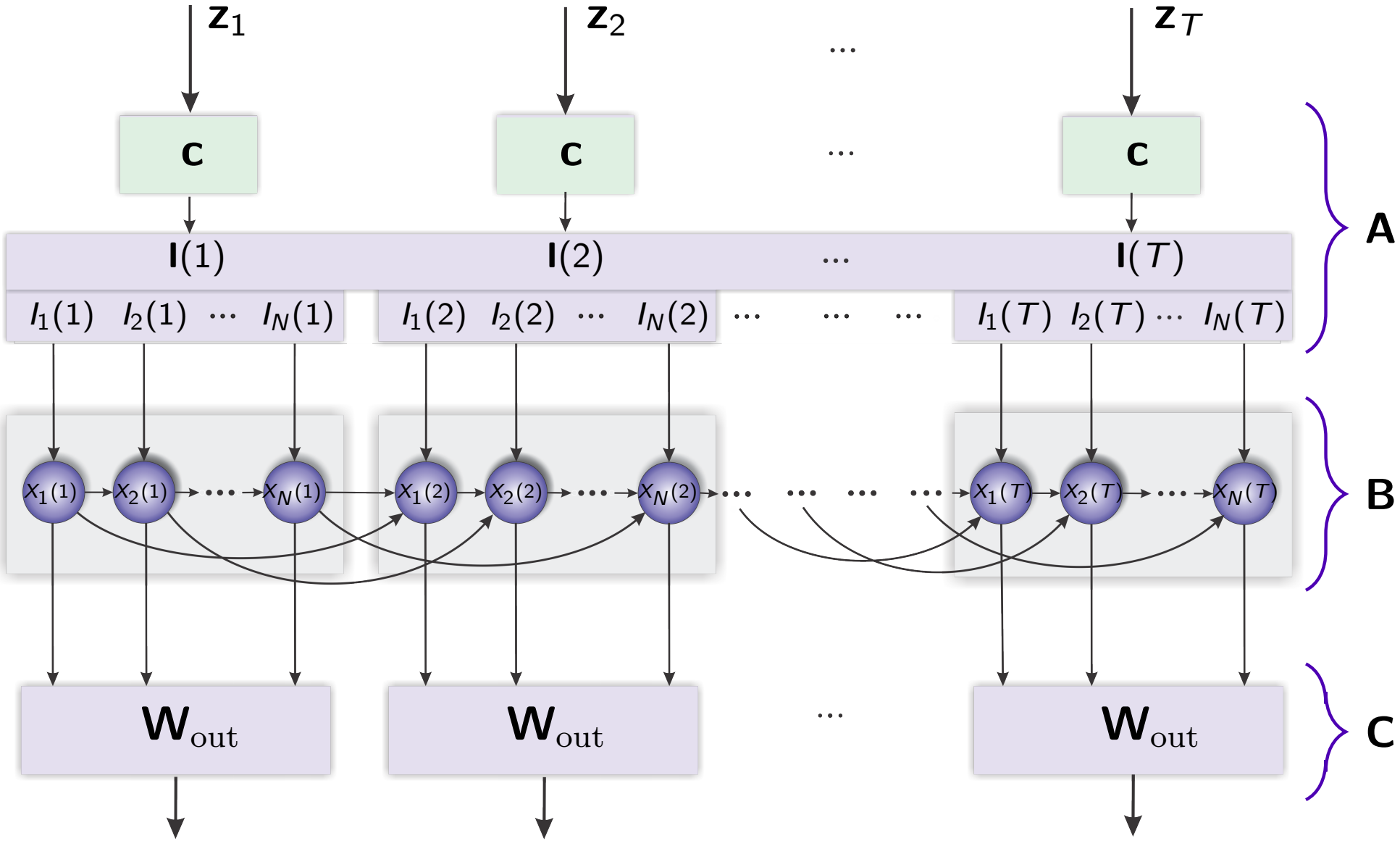
$$x_i(t) := e^{-\xi}x_{i-1}(t)+(1-e^{-\xi})f(x_i(t-1), I_i(t), \boldsymbol{\theta}), \;\; x_0(t) := x_N(t-1), \;\; \xi := \log(1+d),$$

A smooth map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$ specifies the neuron values as a recursion via

$$\mathbf{x}(t) = F(\mathbf{x}(t-1), \mathbf{I}(t), \boldsymbol{\theta}), \tag{10}$$

where F is constructed out of the nonlinear kernel map f; F is referred to as the **reservoir map**.

University of St.Gallen



Architecture of the time-delay reservoir (TDR) and the three modules of the reservoir computer (RC): the input layer A, the time-delay reservoir B, and the readout layer C.

University of St.Gallen

Input and output modules

Input: Take a multi-dimensional time series $\mathbf{z}(t) \in \mathbb{R}^n$ as the input signal. For each t define $\mathbf{I}(t) := C\mathbf{z}(t) \in \mathbb{R}^N$, where $C \in \mathbb{M}_{N,n}$ is the so called **input mask** that takes care of the dimensional and temporal multiplexing.
Output: Let the training be carried out with a **teaching signal** $\mathbf{y}(t) \in \mathbb{R}^n$ that is used to construct a readout W_{out} out of the solution of the ridge regression:

$$W_{\text{out}} := \underset{W \in \mathbb{M}_{N,n}}{\arg\min} \left(\sum_{t=1}^{T^*} \| W^\top \cdot \mathbf{x}(t) - \mathbf{y}(t) \|^2 + \lambda \| W \|^2_{\text{Frob}} \right), \tag{11}$$

whose solution is

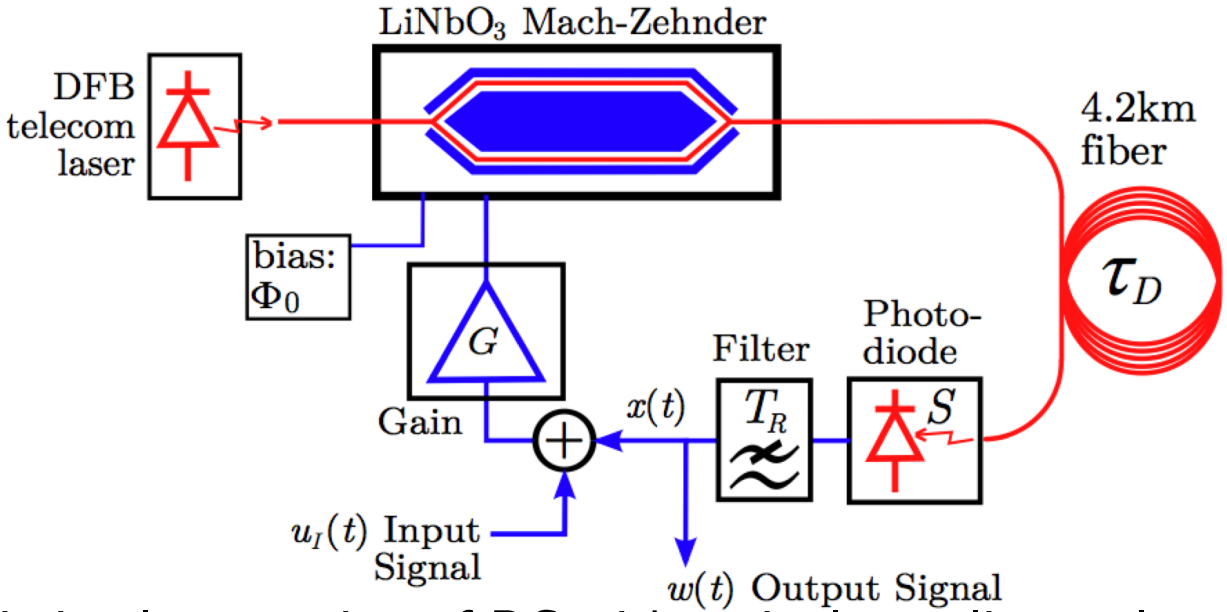
$$W_{\text{out}} = (XX^T + \lambda \mathbb{I}_N)^{-1} XY, \tag{12}$$

where $X \in \mathbb{M}_{N,T^*}$ is the reservoir output given by $X_{i,j} := x_i(j)$ and $Y \in \mathbb{M}_{T^*,n}$ is the teaching matrix containing the vectors $\mathbf{y}(t)$, $t \in \{1, \ldots, T^*\}$, organized by rows, $\lambda \in \mathbb{R}$ is a regularization parameter (usually obtained via cross-validation).

University of St.Gallen

Physical implementation: reservoir computing (RC) devices

- A major feature of the RC is the possibility of constructing physical realizations of reservoirs instead of simulating them using a computer
- Chaotic dynamical systems can be used to construct reservoirs that exhibit the RC features: in [ASV+11] using chaotic electronic oscillators or using optoelectronic devices like in [LSB+12]



Optoelectronic implementation of RC with a single nonlinear element subject to delayed feedback [LSB+12]

University of St.Gallen

nature COMMUNICATIONS

OPEN

ARTICLE

Received 23 Dec 2010 | Accepted 12 Aug 2011 | Published 13 Sep 2011

DOI: 10.1038/ncomms1476

Information processing using a single dynamical node as complex system

L. Appeltant[1], M.C. Soriano[2], G. Van der Sande[1], J. Danckaert[1], S. Massar[3], J. Dambre[4], B. Schrauwen[4], C.R. Mirasso[2] & I. Fischer[2]

nature COMMUNICATIONS

ARTICLE

Received 16 Aug 2012 | Accepted 10 Dec 2012 | Published 15 Jan 2013

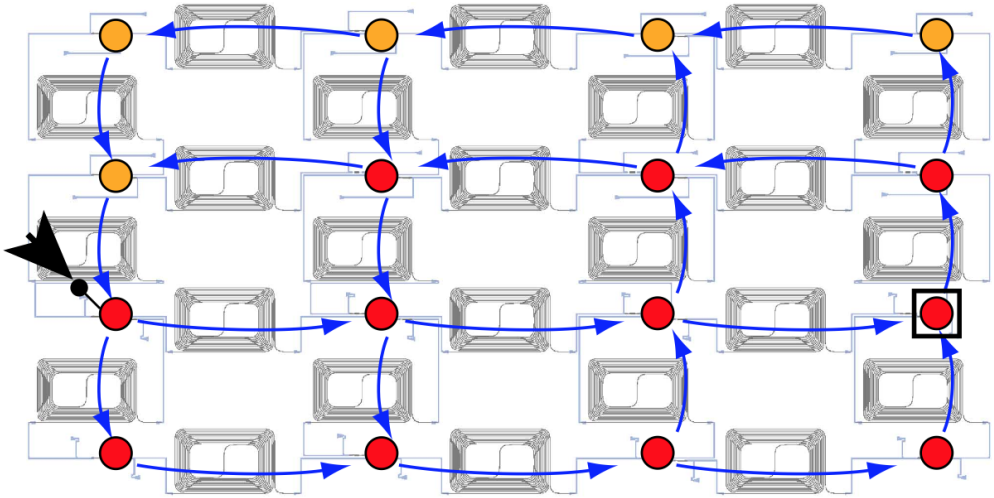
DOI: 10.1038/ncomms2368

OPEN

Parallel photonic information processing at gigabyte per second data rates using transient states

Daniel Brunner[1], Miguel C. Soriano[1], Claudio R. Mirasso[1] & Ingo Fischer[1]

University of St.Gallen



nature COMMUNICATIONS

ARTICLE

Received 13 Aug 2013 | Accepted 4 Mar 2014 | Published 24 Mar 2014

DOI: 10.1038/ncomms4541 OPEN

Experimental demonstration of reservoir computing on a silicon photonics chip

Kristof Vandoorne[1,2], Pauline Mechet[1,2], Thomas Van Vaerenbergh[1,2], Martin Fiers[1,2], Geert Morthier[1,2], David Verstraeten[3], Benjamin Schrauwen[3], Joni Dambre[3] & Peter Bienstman[1,2]

...sity of St.Gallen

Universality vs performance optimization

Universality is a reassuring feature but in practice there are architecture restrictions on the:

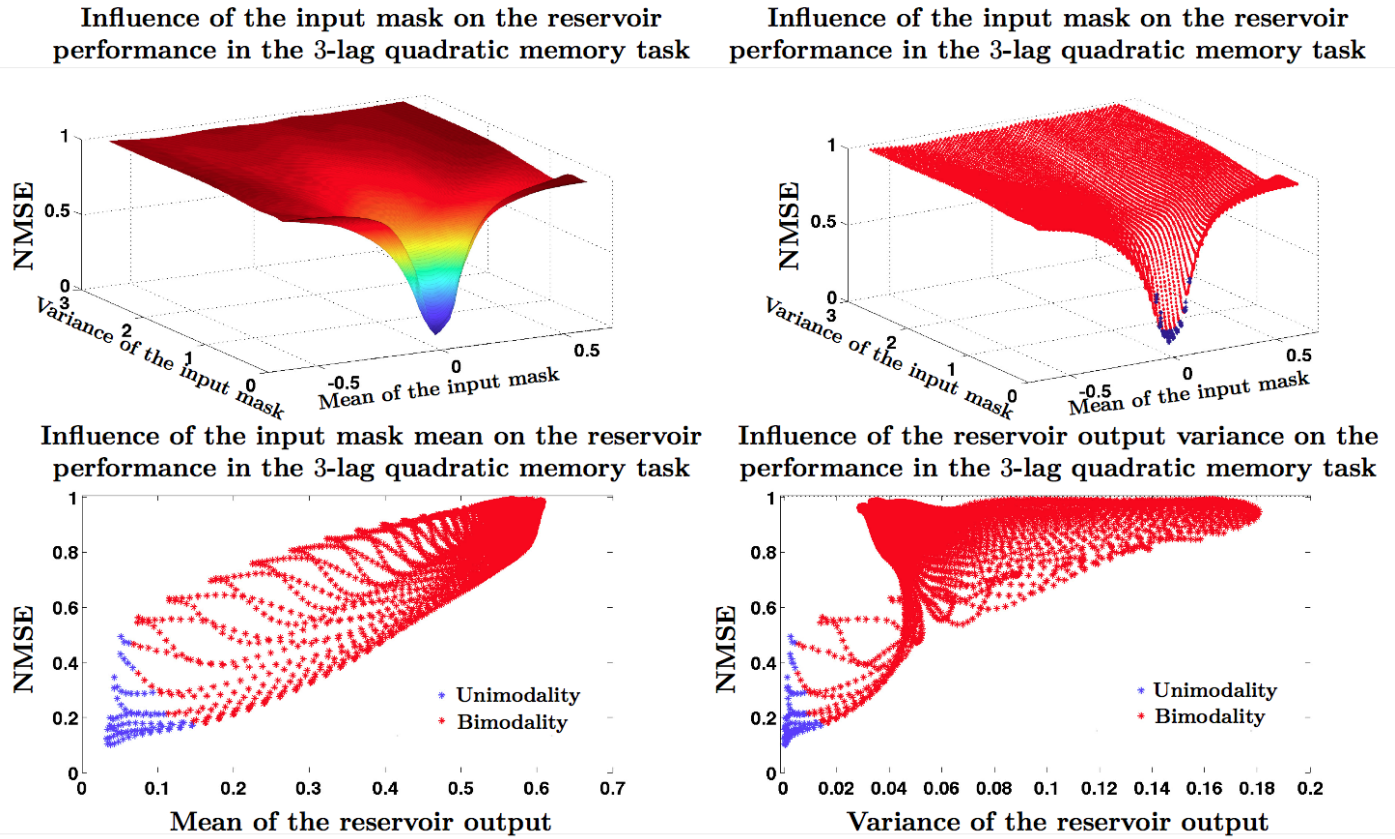
- Basis filters available
- Functional form of the readout

It is hence important to be able to evaluate RC performance for a given architecture and a given task so that it can be optimized by tuning the available parameters in the setup.

We do so in what follows in the setup of TDRs so that we can test the robustness of a given setup with respect to modifications in the task and the parameters.

University of St.Gallen

Optimal performance: stability and unimodality



Behavior of the reservoir performance in a quadratic memory task as a function of the $\bar{c}$ and $\mathrm{var}(\mathbf{c})$. The top panels show how the performance degrades very quickly as soon as $\bar{c}$ and $\mathrm{var}(\mathbf{c})$ separate from zero. The bottom panels depict the reservoir performance as a function of the various output means and variances. We have indicated with red markers the cases in which the reservoir visits the stability basin of a contiguous stable equilibrium, hence showing how unimodality is associated to optimal performance.

Stability analysis

Theorem (Grigoryeva, Henriques, Larger, JPO, 2015)

Let x_0 be an equilibrium of the reservoir time-delay differential equation in autonomous regime, that is, when $I(t) = 0$, and suppose that there exists $\varepsilon > 0$ and $k_\varepsilon \in \mathbb{R}$ such that one of the following conditions holds

$$\text{(i)} \quad f(x + x_0, 0, \boldsymbol{\theta}) \leq k_\varepsilon x + x_0 \text{ for all } x \in (-\varepsilon, \varepsilon)$$

$$\text{(ii)} \quad \frac{f(x + x_0, 0, \boldsymbol{\theta}) - x_0}{x} \leq k_\varepsilon \text{ for all } x \in (-\varepsilon, \varepsilon).$$

If $|k_\varepsilon| < 1$ then x_0 is **asymptotically stable**. *If $|k_\varepsilon| \leq 1$ then x_0 is* **stable**.

Corollary (Grigoryeva, Henriques, Larger, JPO, 2015)

Let x_0 be an equilibrium of the reservoir TDDE and suppose that the nonlinear reservoir kernel function f is continuously differentiable at x_0. If $|\partial_x f(x_0, 0, \boldsymbol{\theta})| < 1$ (respectively, $|\partial_x f(x_0, 0, \boldsymbol{\theta})| \leq 1$), then x_0 is asymptotically stable (respectively, stable).

Corollary (Stability of the equilibria of the Ikeda TDDE; Grigoryeva, Henriques, Larger, JPO, 2015)

Consider the reservoir TDDE in autonomous regime based on the Ikeda kernel,

$$f(x, 0, \boldsymbol{\theta}) = \eta \sin^2(x + \phi). \tag{13}$$

The Ikeda nonlinear TDDE exhibits two families of equilibria:

(i) *The trivial solution $x_0 = 0$ for any $\eta \in \mathbb{R}$ and $\phi = \pi n$, $n \in \mathbb{Z}$. The equilibium $x_0 = 0$ is asymptotically stable for any $\eta \in \mathbb{R}$.*

(ii) *The non-trivial equilibria x_0 are obtained as solutions of the equation $x_0 = \eta \sin^2(x_0 + \phi)$, for any $\eta \in \mathbb{R}$ and $\phi \neq \pi n$, $n \in \mathbb{Z}$. These equilibria are asymptotically stable (respectively, stable) if*

$$|\sin(2x_0 + 2\phi)| < \frac{1}{|\eta|} \quad (\textit{respectively, } |\sin(2x_0 + 2\phi)| \leq \frac{1}{|\eta|}). \tag{14}$$

When $|\eta| < 1$ (respectively, $|\eta| \leq 1$), there exists only one non-trivial equilibrium that is always asymptotically stable (respectively, stable).

Stability of the TDR: discrete time approximation

Proposition (Grigoryeva, Henriques, Larger, JPO, 2015)

The point $x_0 \in \mathbb{R}$ is an equilibrium of the reservoir time-delay differential equation in autonomous regime, that is when $I(t) = 0$, if and only if the vector $\mathbf{x}_0 := x_0 \mathbf{i}_N$ is a fixed point of the N-dimensional discretized nonlinear time-delay reservoir

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t-1), \mathbf{I}(t), \boldsymbol{\theta}) \tag{15}$$

in autonomous regime, that is, when $\mathbf{I}(t) = \mathbf{0}_N$.

Theorem (Grigoryeva, Henriques, Larger, JPO, 2015)

Let $\mathbf{x}_0 = x_0 \mathbf{i}_N$ be a fixed point of the N-dimensional recursion $\mathbf{x}(t) = F(\mathbf{x}(t-1), \mathbf{I}(t), \boldsymbol{\theta})$ in autonomous regime. Then, $\mathbf{x}_0 \in \mathbb{R}^N$ is asymptotically stable (respectively stable) if $|\partial_x f(x_0, 0, \boldsymbol{\theta})| < 1$ (respectively, $|\partial_x f(x_0, 0, \boldsymbol{\theta})| \leq 1$).

The approximating model and nonlinear memory capacity

Consider a stable equilibrium $x_0 \in \mathbb{R}$ of the autonomous system associated to (8) or, equivalently, a stable fixed point $\mathbf{x}_0 := (x_0, \ldots, x_0)^\top \in \mathbb{R}^N$ of (10). We construct the approximation of (10) by using its linearization at $\mathbf{x}_0$ with respect to the delayed self-feedback and its Rth-order Taylor expansion with respect to its dependence on the signal injection:

$$\mathbf{x}(t) = F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) + A(\mathbf{x}_0, \boldsymbol{\theta})(\mathbf{x}(t-1) - \mathbf{x}_0) + \boldsymbol{\varepsilon}(t), \tag{16}$$

where $A(\mathbf{x}_0, \boldsymbol{\theta}) := D_{\mathbf{x}} F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ and $\boldsymbol{\varepsilon}(t)$ is given by:

$$\boldsymbol{\varepsilon}(t) = (1 - e^{-\xi}) \left(q_R\left(z(t), c_1\right), \ldots, q_R\left(z(t), c_1, \ldots, c_N\right) \right)^\top,$$

with

$$q_R\left(z(t), c_1, \ldots, c_r\right) := \sum_{i=1}^{R} \frac{z(t)^i}{i!} (\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta}) \sum_{j=1}^{r} e^{-(r-j)\xi} c_j^i,$$

and $(\partial_I^{(i)} f)(x_0, 0, \boldsymbol{\theta})$ the ith order partial derivative of the nonlinear kernel f with respect to $I(t)$ evaluated at $(x_0, 0, \boldsymbol{\theta})$.

University of St.Gallen

Let the input signal be $\{z(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(0,\sigma_z^2)$, then $\{\mathbf{I}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\mathbf{0}_N,\Sigma_I)$, with $\Sigma_I := \sigma_z^2\mathbf{c}^\top\mathbf{c}$, and $\{\boldsymbol{\varepsilon}(t)\}_{t\in\mathbb{Z}} \sim \mathrm{IID}(\boldsymbol{\mu}_\varepsilon,\Sigma_\varepsilon)$ with

$$\boldsymbol{\mu}_\varepsilon = (1-e^{-\xi})\left(q_R\left(\mu_z,c_1\right),\ldots,q_R\left(\mu_z,c_1,\ldots,c_N\right)\right)^\top,$$

where $\mu_z^i := \mathrm{E}\left[z(t)^i\right]$ and $\Sigma_\varepsilon := \mathrm{E}\left[(\boldsymbol{\varepsilon}(t)-\boldsymbol{\mu}_\varepsilon)(\boldsymbol{\varepsilon}(t)-\boldsymbol{\mu}_\varepsilon)^\top\right] \in \mathbb{S}_N$ with the entries given by:

$$\begin{aligned}(\Sigma_\varepsilon)_{ij} =&(1-e^{-\xi})^2((q_R(\cdot,c_1,\ldots,c_i)\cdot q_R(\cdot,c_1,\ldots,c_j))(\mu_z)\\ &- q_R(\mu_z,c_1,\ldots,c_i)q_R(\mu_z,c_1,\ldots,c_j)),\quad i,j=1,\ldots,N.\end{aligned}$$

The process (16) is a VAR(1) model

$$\mathbf{x}(t)-\boldsymbol{\mu}_x = A(\mathbf{x}_0,\boldsymbol{\theta})(\mathbf{x}(t-1)-\boldsymbol{\mu}_x)+(\boldsymbol{\varepsilon}(t)-\boldsymbol{\mu}_\varepsilon) \tag{17}$$

with $\boldsymbol{\mu}_x = (I_N - A(\mathbf{x}_0,\boldsymbol{\theta}))^{-1}(F(\mathbf{x}_0,\mathbf{0}_N,\boldsymbol{\theta})-A(\mathbf{x}_0,\boldsymbol{\theta})\mathbf{x}_0+\boldsymbol{\mu}_\varepsilon)$ and an autocovariance function $\Gamma(k) := \mathrm{E}\left[(\mathbf{x}(t)-\boldsymbol{\mu}_x)\left(\mathbf{x}(t-k)-\boldsymbol{\mu}_x\right)^\top\right]$, $k\in\mathbb{Z}$, recursively determined by the Yule-Walker equations [Lüt05]:

$$\begin{aligned}\mathrm{vec}(\Gamma(0)) &= \left(\mathbb{I}_{N^2}-A(\mathbf{x}_0,\boldsymbol{\theta})\otimes A(\mathbf{x}_0,\boldsymbol{\theta})\right)^{-1}\mathrm{vec}(\Sigma_\varepsilon),\\ \Gamma(k) &= A(\mathbf{x}_0,\boldsymbol{\theta})\Gamma(k-1),\quad \Gamma(-k)=\Gamma(k)^\top.\end{aligned}$$

University of St.Gallen

The nonlinear memory capacity estimations

A h-**lag memory task** is determined by a function $H : \mathbb{R}^{h+1} \to \mathbb{R}$ (in general nonlinear) that is used to generate $y(t) := H(z(t), z(t-1), \ldots, z(t-h)) \in \mathbb{R}$ out of the reservoir input $\{z(t)\}_{t\in\mathbb{Z}}$.

Recall, that the optimal linear readout $\mathbf{W}_{\text{out}}$ adapted to the memory task H is given by the solution of a ridge (or Tikhonov [Tik43]) linear regression problem

$$(\mathbf{W}_{\text{out}}, a_{\text{out}}) := \underset{\mathbf{W}\in\mathbb{R}^N, a\in\mathbb{R}}{\arg\min} \left(\mathrm{E}\left[(\mathbf{W}^\top \cdot \mathbf{x}(t) + a - y(t))^2\right] + \lambda\|\mathbf{W}\|^2\right). \tag{18}$$

Using the fact that $\{\mathbf{x}(t)\}_{t\in\mathbb{Z}}$ is the unique stationary solution of VAR(1) approximating system (17) for the TDR (17) obtain

$$\mathbf{W}_{\text{out}} = (\Gamma(0) + \lambda\mathbb{I}_N)^{-1}\mathrm{Cov}(y(t), \mathbf{x}(t)), \tag{19}$$

$$a_{\text{out}} = \mathrm{E}[y(t)] - \mathbf{W}_{\text{out}}^\top \boldsymbol{\mu}_x, \tag{20}$$

where $\boldsymbol{\mu}_x$, $\Gamma(0) \in \mathbb{S}_N$ are provided in (17), and $\mathrm{Cov}(y(t), \mathbf{x}(t))$ is a vector in $\mathbb{R}^N$ that has to be determined for every specific memory task H.

The error committed by the reservoir when using the optimal readout is

$$\begin{aligned}\mathrm{MSE}_H = &\operatorname{var}(y(t)) - \operatorname{Cov}(y(t), \mathbf{x}(t))^\top (\Gamma(0) + \lambda \mathbb{I}_N)^{-1} (\Gamma(0) + 2\lambda \mathbb{I}_N) \\ &\times (\Gamma(0) + \lambda \mathbb{I}_N)^{-1} \operatorname{Cov}(y(t), \mathbf{x}(t)).\end{aligned}$$

Using the VAR(1) approximating model (17) of RC, the corresponding H-**memory capacity** is

$$\begin{aligned}C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda) = &\operatorname{Cov}(y(t), \mathbf{x}(t))^\top (\Gamma(0) + \lambda \mathbb{I}_N)^{-1} (\Gamma(0) + 2\lambda \mathbb{I}_N) \qquad (21) \\ &\times (\Gamma(0) + \lambda \mathbb{I}_N)^{-1} \operatorname{Cov}(y(t), \mathbf{x}(t)) / \operatorname{var}(y(t)). \qquad (22)\end{aligned}$$

Additionally,

$$0 \leq C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda) \leq 1.$$

Once a specific reservoir and task H have been fixed, the capacity function $C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda)$ can be explicitly written down and it can hence be used to find reservoir parameters $\boldsymbol{\theta}_{\mathrm{opt}}$ and an input mask $\mathbf{c}_{\mathrm{opt}}$ that maximize it, by solving the optimization problem

$$(\boldsymbol{\theta}_{\mathrm{opt}}, \mathbf{c}_{\mathrm{opt}}) := \underset{\boldsymbol{\theta} \in \mathbb{R}^K, \mathbf{c} \in \mathbb{R}^N}{\arg\max}\ C_H(\boldsymbol{\theta}, \mathbf{c}, \lambda). \qquad (23)$$

University of St.Gallen

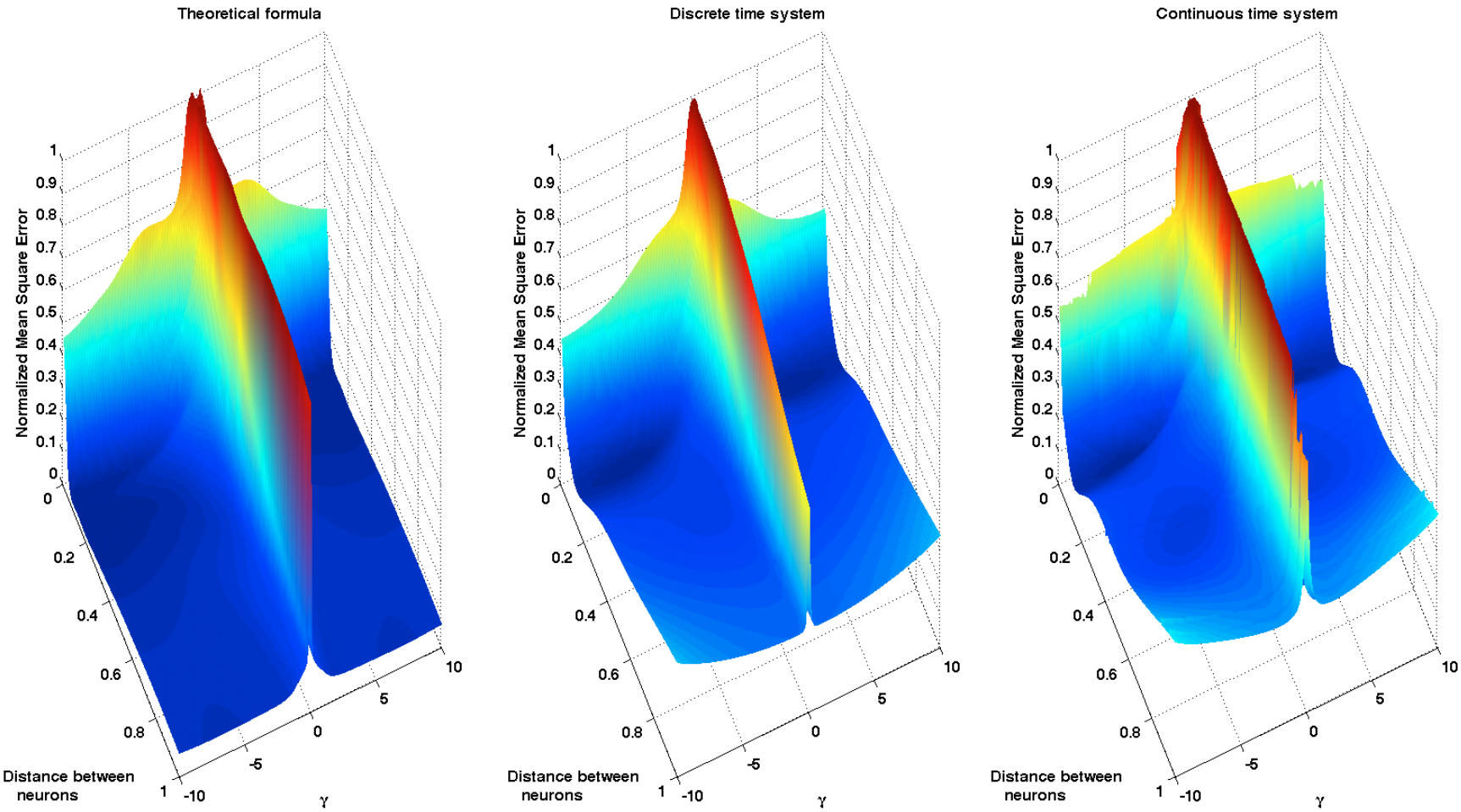
Optimal nonlinear capacity

The h-lag quadratic memory task. Take a quadratic task function of the form $H(\mathbf{z}^h(t)) := \mathbf{z}^h(t)^\top Q\mathbf{z}^h(t)$, for some symmetric $h+1$-dimensional matrix Q. In this case $\mathrm{var}(y(t)) = (\mu_z^4 - \sigma_z^4)\sum_{i=1}^{h+1} Q_{ii}^2 + 4\sigma_z^4 \sum_{i=1}^{h+1}\sum_{j>i}^{h+1} Q_{ij}^2$, and

$$\mathrm{Cov}(y(t), x_i(t)) = (1 - e^{-\xi}) \sum_{j=1}^{h+1}\sum_{r=1}^{N} Q_{jj}(A^{j-1})_{ir} \times (s_R(\mu_z, c_1, \ldots, c_r) - \sigma_z^2 q_R(\mu_z, c_1, \ldots, c_r)),$$

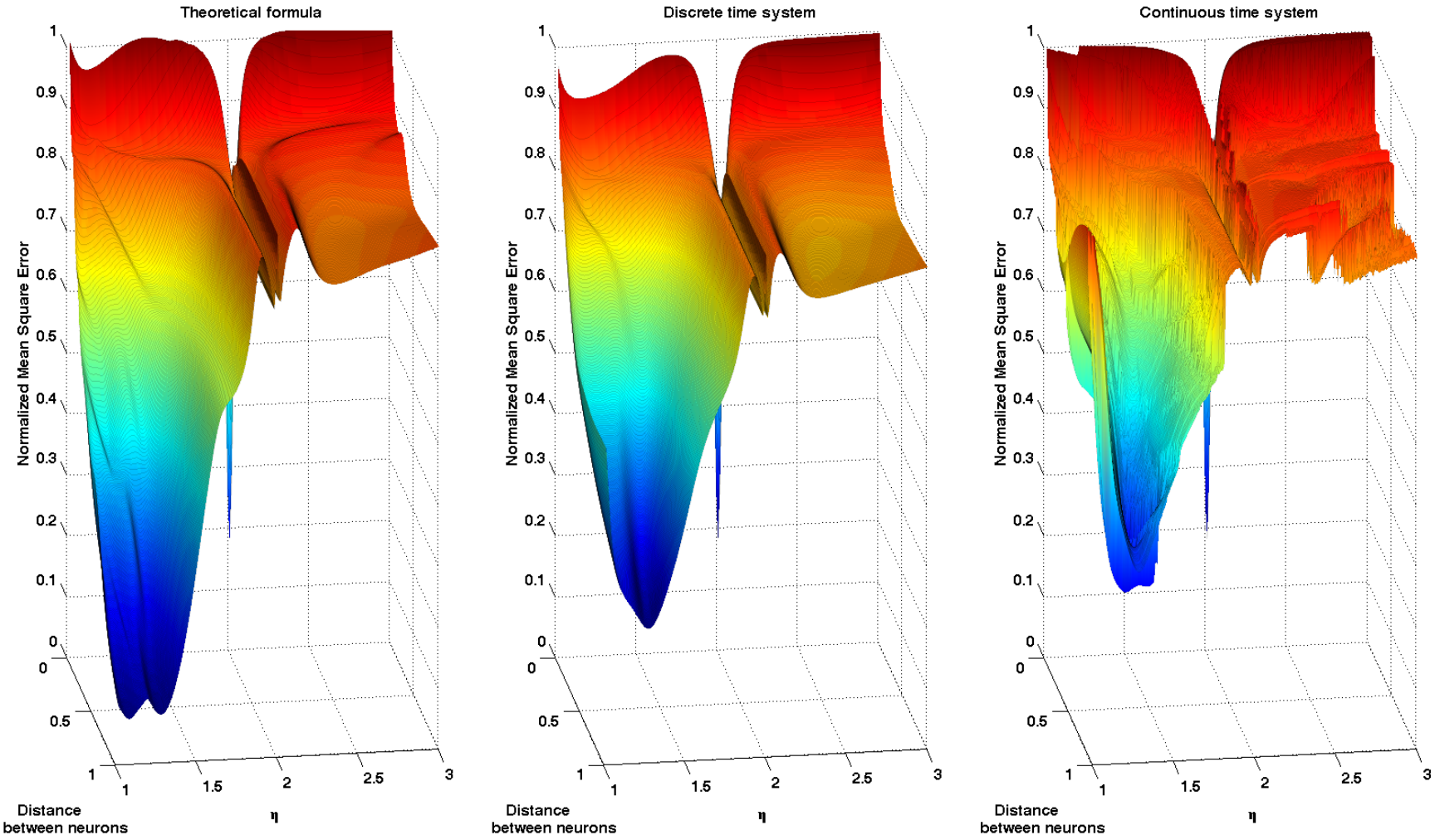
where the polynomial s_R on the variable x is defined as $s_R(x, c_1, \ldots, c_r) := x^2 \cdot q_R(x, c_1, \ldots, c_r)$.

University of St.Gallen



Error exhibited by a TDR computer with a Mackey-Glass kernel in a 3-lag quadratic memory task as a function of the separation between neurons d and the parameter γ, respectively. The points in the surfaces of the middle and right panels are the result of Monte Carlo evaluations of the NMSE exhibited by the discrete and continuous time TDRs, respectively. The left panel was constructed modeling the reservoir with an approximating VAR(1) model.

University of St.Gallen



Error exhibited by a TDR computer with a Mackey-Glass kernel in a 6-lag quadratic memory task as a function of the separation between neurons d and the parameter η. The points in the surfaces of the middle and right panels are the result of Monte Carlo evaluations of the NMSE exhibited by the discrete and continuous time TDRs, respectively. The left panel was constructed modeling the reservoir with an approximating VAR(1) model.

University of St.Gallen

www.nature.com/scientificreports

OPEN

Optimal nonlinear information processing capacity in delay-based reservoir computers

Received: 04 March 2015
Accepted: 03 July 2015
Published: 11 September 2015

Lyudmila Grigoryeva[1], Julie Henriques[1,2], Laurent Larger[3] & Juan-Pablo Ortega[4]

University of St.Gallen

Other applications of the reservoir model

- **Evaluation of the finite sample training and testing errors:** Given a reservoir output X of size T, the total mean square reservoir training error conditional on X and for any teaching signal Y, is given by

$$\begin{aligned}\mathrm{MSE}_{\mathrm{total},\lambda} \mid X &= \mathrm{trace}\,(\Sigma) \\ &+ \frac{1}{T}\mathrm{trace}[\mathrm{trace}\,(\Sigma)\left(R_\lambda \mathcal{X} A \mathcal{X}^\top \left(R_\lambda \mathcal{X}\mathcal{X}^\top - 2\mathbb{I}_{N+1}\right)\right) \\ &\qquad + \lambda^2 T^2 R_\lambda \mathcal{W}\mathcal{W}^\top R_\lambda \mathcal{X}\mathcal{X}^\top],\end{aligned}$$

where N is the number of neurons of the reservoir, $\mathcal{X} = \left(\mathbf{i}_T || X^\top\right)^\top$, and $\mathcal{W} := \left(\boldsymbol{a} || W^\top\right)^\top$ with $W := \Gamma(0)^{-1}\mathrm{Cov}\,(\mathbf{x}(t), \mathbf{y}(t))$ and $\boldsymbol{a} := \boldsymbol{\mu}_y - W^\top \boldsymbol{\mu}_x$. Finally, $R_\lambda := (\mathcal{X} A \mathcal{X}^\top + \lambda T \mathbb{I}_{N+1})^{-1}$ and $\Sigma := \mathrm{Cov}\,(\mathbf{y}(t), \mathbf{y}(t)) - W^\top \Gamma(0) W$.

- **The RC defining features:** Consider the reservoir model driven by the real valued and non-necessarily stationary input signal $\{z(t)\}_{t\in\mathbb{Z}}$.

  (i) Let $\mathbf{c} \in \mathbb{R}^N$ be an input mask and $\mathbf{I}(t) := \mathbf{c}z(t)$ the corresponding input forcing. Let

$$F_I^R(\mathbf{I}(t), \mathbf{x}_0, \boldsymbol{\theta}) := \sum_{i=1}^{R} \frac{1}{i!} D_{\mathbf{I}}^{(i)} F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta}) \overbrace{\mathbf{I}(t) \otimes \cdots \otimes \mathbf{I}(t)}^{i \text{ factors}}$$

  Assume that one of the following conditions holds:

  (a) The map $F_I^R(\cdot, \mathbf{x}_0, \boldsymbol{\theta}) : \mathbb{R}^N \to \mathbb{R}^N$ is injective.

  (b) The input signal is bounded.

  If $A(\mathbf{x}_0, \boldsymbol{\theta}) := D_{\mathbf{x}} F(\mathbf{x}_0, \mathbf{0}_N, \boldsymbol{\theta})$ has no zero eigenvalues, then the reservoir model satisfies the separation property.

  (ii) The input signal $\{z(t)\}_{t\in\mathbb{Z}}$ is strictly stationary with finite automoments up to order $2R$ and that it is bounded and the linear map $A(\mathbf{x}_0, \boldsymbol{\theta})$ is such that $\|A(\mathbf{x}_0, \boldsymbol{\theta})\| < 1$, then the reservoir model satisfies the uniform fading memory property.

Application examples: usual benchmarks and early applications

- RC has outperformed well-established methods of nonlinear system identification, prediction, and classification (see [LJ09] for a review):
  - Prediction of chaotic dynamics (three orders of magnitude accuracy improvement [JH04])
  - Nonlinear wireless channel equalization (two orders of magnitude improvement [JH04])
  - Japanese Vowel benchmark (zero test error rate, previous best: 1.8% [JLPS07])
  - Financial forecasting (winner of the international forecasting competition NN32[1])
  - Isolated spoken digits recognition (improvement of word error rate on benchmark from 0.6% of previous best system to 0.2% and further to 0% test error in more recent works [JLPS07, ASV+11, LSB+12, PDS+12, BSMF13])
  - NARMA model identification task [AP00, RT11].

[1] http://www.neural-forecasting-competition.com/NN3/index.htm

University of St.Gallen

Application examples: classification

- Deep RC networks outperform all state-of-the-art techniques in the written digit classification using the MNIST corpus [JDD+15]. RC halves the error exhibited by deep neural network committees and the results are robust with respect to the presence of various noises.
- A similar architecture [TJSM10] has shown performances comparable to state-of-the-art technology in the phoneme recognition problem based on the TIMIT corpus with a competitive training effort.
- Hi-Res EEG signals: monitoring of epileptic seizures in animals [BSVS09, BVvM+11, BVN+13, NDK11] and in the discrimination of the emotion valence in humans [KHBG15].
- Electrocardiogram signals (ECGs) [LS13].
- Fuel cell diagnostics [Hugo]

University of St.Gallen

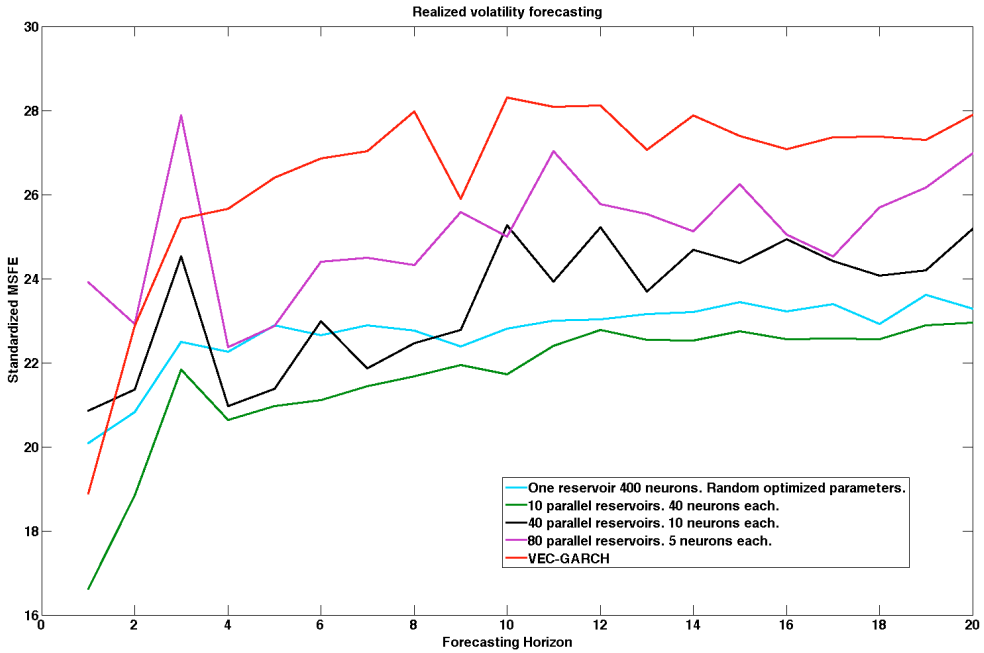
Application examples: forecasting

- Industrial production time series [WSS08, WS10]
- Great Lakes water level in [Cou10]
- Short-term forecasting of wind speed [FLdA+08]
- Water inflow forecasting [SOP+07]
- Short-term electric consumption [DS12] and temperature [DOS13]
- Telephone calls load [BSU+15]
- Short-term stock price prediction [LYS09] with applications to intelligent stock trading systems [LYS11]

University of St.Gallen

Application examples: volatility forecasting

- TDRs have been shown in [GHLO14] to outperform standard multivariate parametric models in the modeling of realized financial volatility and correlations.



Average realized volatility forecasting performance using RC and VEC(1,1) models estimated via maximum likelihood (MLE). The sMSFE reported is obtained with the estimated parametric models. All the TDRs considered have been generated using the nonlinear Mackey-Glass kernel with $p = 2$.

Neural Networks 55 (2014) 59–71

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Stochastic nonlinear time series forecasting using time-delay reservoir computers: Performance and universality

Lyudmila Grigoryeva [a], Julie Henriques [b], Laurent Larger [c], Juan-Pablo Ortega [d,*]

[a] *Laboratoire de Mathématiques de Besançon, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon Cedex, France*

[b] *Cegos Deployment. 11 rue Denis Papin. F-25000 Besançon, France*

[c] *UMR CNRS FEMTO-ST 6174/Optics Department, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon Cedex, France*

[d] *Centre National de la Recherche Scientifique, Laboratoire de Mathématiques de Besançon, Université de Franche-Comté, UFR des Sciences et Techniques. 16, route de Gray. F-25030 Besançon Cedex, France*

University of St.Gallen

A final example: volatility filtering

The **standard ARSV** model is given by the prescription

$$\begin{cases} y_t &= \mu + \sigma_t \epsilon_t, & \{\epsilon_t\} \sim \mathrm{IID}(0,1) \\ b_t &= \gamma + \phi b_{t-1} + w_t, & \{w_t\} \sim \mathrm{IID}(0, \sigma_w^2) \end{cases} \tag{24}$$

where $b_t := \log(\sigma_t^2)$, $\gamma \in \mathbb{R}$, $\phi \in (-1,1)$. Assume that $\{\epsilon_t\}$ and $\{w_t\}$ are uncorrelated (can be relaxed to account for the leverage effects and assymetric behaviour of stock prices).
Observations:

- the process $\{\sigma_t\}$ is a non-traded stochastic latent variable that, unlike in GARCH-like models [Eng82, Bol86] is not a predictable process that can be written as a function of previous returns and volatilities;
- the unique stationary returns process induced by (24) provided that $\phi \in (-1,1)$ is a WN (no autocorrelation) with finite moments of arbitrary order.

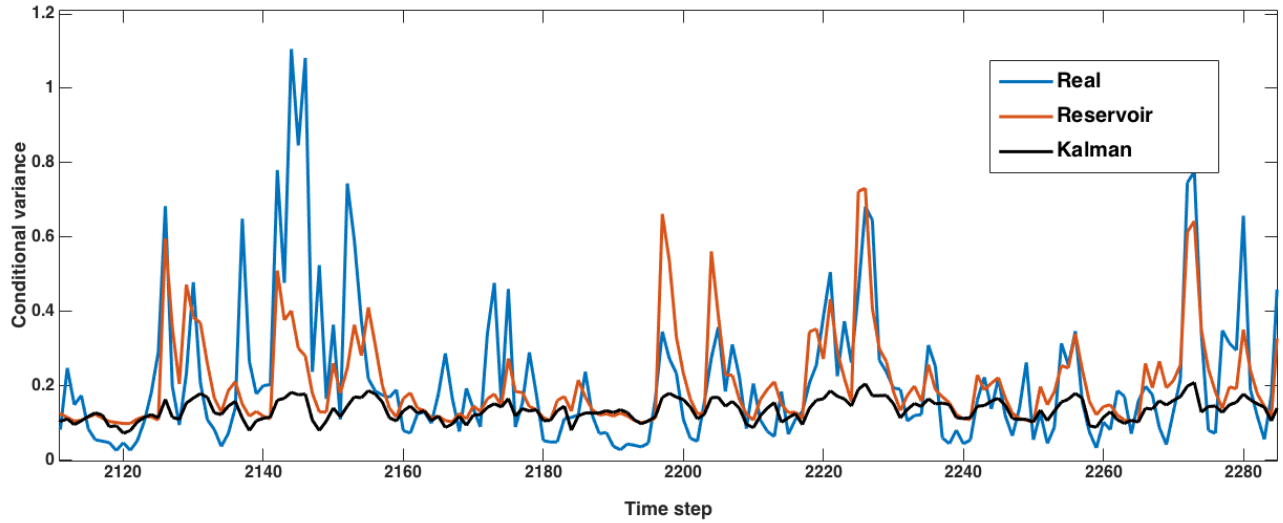
ARSV: estimation and filtering techniques

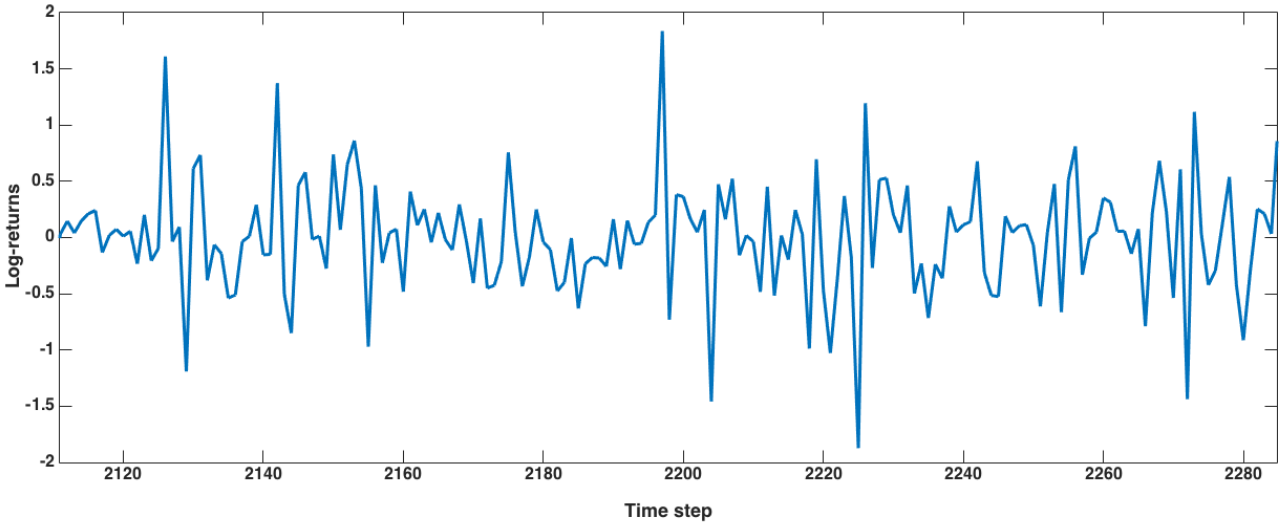
References on the model: Taylor [Tay86, Tay05]

1. Bayesian approach: Jacquier et al. [JPR94], many others.
2. Non-Bayesian approaches:
   - Harvey et al. [HRS94], Ruiz [Rui94] suggested a QML estimator based on the Kalman filter
   - Meyer et al. [MFB03] and Shimada and Tsukuda [ST05] use approximated linear filtering methods based on Laplace approximation to produce a MLE
   - *h*-likelihood estimation approach of Castillo and Lee [dCL08], [LWLdC11] based on treating the ARSV models as a GLM with varying random effects

University of St.Gallen

Performance comparison





University of St.Gallen

- Kalman testing error: 100.63%
- *h*-likelihood testing error: 82.50%
- Reservoir testing error (5 nodes, Ikeda kernel, optimized parameters): 73.88%
- No restrictions on the model prescription or on the innovations character

University of St.Gallen

References I

A. F. Atiya and A. G. Parlos.
New results on recurrent network training: unifying the algorithms and accelerating convergence.
IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, 11(3):697–709, jan 2000.

V. I. Arnold.
On functions of three variables.
Proceedings of the USSR Academy of Sciences, 114:679–681, 1957.

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, 2:468, jan 2011.

S. Boyd and L. Chua.
Fading memory and the problem of approximating nonlinear operators with Volterra series.
IEEE Transactions on Circuits and Systems, 32(11):1150–1161, nov 1985.

Tim Bollerslev.
Generalized autoregressive conditional heteroskedasticity.
Journal of Econometrics, 31(3):307–327, 1986.

Michael S. Branicky.
Universal computation and other capabilities of hybrid and continuous dynamical systems.
Theoretical Computer Science, 138(1):67–100, 1995.

D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer.
Parallel photonic information processing at gigabyte per second data rates using transient states.
Nature Communications, 4(1364), 2013.

University of St.Gallen

References II

Filippo Maria Bianchi, Simone Scardapane, Aurelio Uncini, Antonello Rizzi, and Alireza Sadeghian.
Prediction of telephone calls load using Echo State Network with exogenous variables.
Neural Networks, 71(November):204–213, 2015.

Pieter Buteneers, Benjamin Schrauwen, David Verstraeten, and Dirk Stroobandt.
Real-Time Epileptic Seizure Detection on Intra-cranial Rat Data Using Reservoir Computing.
In *Advances in Neuro-Information Processing*, pages 56–63. Springer Berlin Heidelberg, 2009.

Pieter Buteneers, David Verstraeten, Bregt Van Nieuwenhuyse, Dirk Stroobandt, Robrecht Raedt, Kristl Vonck, Paul Boon, and Benjamin Schrauwen.
Real-time detection of epileptic seizures in animal models using reservoir computing.
Epilepsy Research, 103(2):124–134, 2013.

Pieter Buteneers, David Verstraeten, Pieter van Mierlo, Tine Wyckhuys, Dirk Stroobandt, Robrecht Raedt, Hans Hallez, and Benjamin Schrauwen.
Automatic detection of epileptic seizures on the intra-cranial electroencephalogram of rats using reservoir computing.
Artificial Intelligence in Medicine, 53(3):215–223, 2011.

Neil E. Cotter and Thierry J. Guillerm.
The CMAC and a theorem of Kolmogorov.
Neural Networks, 5(2):221–228, 1992.

Paulin Coulibaly.
Reservoir Computing approach to Great Lakes water level forecasting.
Journal of Hydrology, 381(1-2):76–88, 2010.

University of St.Gallen

References III

G. Cybenko.

Approximation by superpositions of a sigmoidal function.
Mathematics of Control, Signals, and Systems, 2(4):303–314, dec 1989.

Joan del Castillo and Youngjo Lee.

GLM-methods for volatility models.
Stat. Model., 8(3):263–283, 2008.

Ali Deihimi, Omid Orang, and Hemen Showkati.

Short-term electric load and temperature forecasting using wavelet echo state networks with neural reconstruction.
Energy, 57:382–401, 2013.

Ali Deihimi and Hemen Showkati.

Application of echo state networks in short-term electric load forecasting.
Energy, 39(1):327–340, 2012.

Robert F. Engle.

Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation.
Econometrica, 50(4):987–1007, 1982.

Aida A. Ferreira, Teresa B. Ludermir, Ronaldo R. B. de Aquino, Milde M. S. Lira, and Otoni N. Neto.

Investigating the use of Reservoir Computing for forecasting the hourly wind speed in short -term.
In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1649–1656. IEEE, jun 2008.

Jürgen Franke and Michael H. Neumann.

Bootstrapping neural networks.
Neural Computation, 12(8):1929–1949, aug 2000.

University of St.Gallen

References IV

Lyudmila Grigoryeva, Julie Henriques, Laurent Larger, and Juan-Pablo Ortega.

Stochastic time series forecasting using time-delay reservoir computers: performance and universality.
Neural Networks, 55:59–71, 2014.

A. Ronald Gallant and Halbert White.

On learning the derivatives of an unknown mapping with multilayer feedforward networks.
Neural Networks, 5(1):129–138, 1992.

D. Hilbert.

Über die Gleichung neunten Grades.
Mathematische Annalen, 97(1):243–250, dec 1927.

Hecht-Nielsen.

Theory of the backpropagation neural network.
In *International Joint Conference on Neural Networks*, pages 593–605 vol.1. IEEE, 1989.

A. C. Harvey, E Ruiz, and Neil Shephard.

Multivariate stochastic variance models.
Review of Economic Studies, 61:247–264, 1994.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White.

Multilayer feedforward networks are universal approximators.
Neural Networks, 2(5):359–366, 1989.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White.

Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks.
Neural Networks, 3(5):551–560, 1990.

University of St.Gallen

References V

Kensuke Ikeda.

Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system.
Optics Communications, 30(2):257–261, aug 1979.

Herbert Jaeger.

The 'echo state' approach to analysing and training recurrent neural networks.
Technical report, German National Research Center for Information Technology, 2001.

Azarakhsh Jalalvand, Kris Demuynck, Wesley De Neve, Rik Van de Walle, and Jean-Pierre Martens.

Design of reservoir computing systems for noise-robust speech and handwriting recognition.
*28th Conference on Graphics, Patterns and Images (accepted in the Workshop of Theses and Dissertations (WTD))
Proceedingsaccepted in the Workshop of Theses and Dissertations (WTD)) Proceedings*, (2), 2015.

Herbert Jaeger and Harald Haas.

Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication.
Science, 304(5667):78–80, 2004.

Herbert Jaeger, Mantas Lukoševičius, Dan Popovici, and Udo Siewert.

Optimization and applications of echo state networks with leaky-integrator neurons.
Neural Networks, 20(3):335–352, 2007.

E Jacquier, N G Polson, and P E Rossi.

Bayesian analysis of stochastic volatility models.
Journal of Business and Economic Statistics, 12:371–417, 1994.

University of St.Gallen

References VI

P. Koprinkova-Hristova, L. Bozhkov, and P. Georgieva.

Echo state networks for feature selection in affective computing.
In *Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection*, pages 131–141. Springer Verlag, 2015.

A. N. Kolmogorov.

On the representation of continuous functions of several variables as superpositions of functions of smaller number of variables.
Soviet Math. Dokl, 108:179–182, 1956.

J. F. Kaashoek and H. K. van Dijk.

Neural networks: an econometric tool.
In *Computer-Aided Econometrics*, chapter 12. CRC Press, 2003.

M. Lukoševičius and H. Jaeger.

Reservoir computing approaches to recurrent neural network training.
Computer Science Review, 3(3):127–149, 2009.

Claudia Lainscsek and Terrence J Sejnowski.

Electrocardiogram classification using delay differential equations.
Chaos (Woodbury, N.Y.), 23(2):023132, jun 2013.

L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer.

Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing.
Optics Express, 20(3):3241, jan 2012.

References VII

Helmut Lütkepohl.

New Introduction to Multiple Time Series Analysis.
Springer-Verlag, Berlin, 2005.

Johan Lim, Lee Woojoo, Youngjo Lee, and Joan del Castillo.

The hierarchical-likelihood approach to autoregressive stochastic volatility models.
Computational Statistics and Data Analysis, 55(55):248–260, 2011.

Xiaowei Lin, Zehong Yang, and Yixu Song.

Short-term stock price prediction based on echo state networks.
Expert Systems with Applications, 36(3):7313–7317, 2009.

Xiaowei Lin, Zehong Yang, and Yixu Song.

Intelligent stock trading system based on improved technical analysis and Echo State Network.
Expert Systems with Applications, 38(9):11347–11354, 2011.

Wolfgang Maass.

Liquid state machines: motivation, theory, and applications.
In S. S. Barry Cooper and Andrea Sorbi, editors, *Computability In Context: Computation and Logic in the Real World*, chapter 8, pages 275–296. 2011.

Renate Meyer, David A. Fournier, and Andreas Berg.

Stochastic volatility: Bayesian computation using automatic differentiation and the extended Kalman filter.
Econometrics Journal, 6(2):408–420, dec 2003.

M. C. Mackey and L. Glass.

Oscillation and chaos in physiological control systems.
Science, 197:287–289, 1977.

University of St.Gallen

References VIII

Wolfgang Maass, Prashant Joshi, and Eduardo D. Sontag.
Computational aspects of feedback in neural circuits.
PLoS Computational Biology, 3(1):e165, 2007.

Wolfgang Maass and Henry Markram.
On the computational power of circuits of spiking neurons.
Journal of Computer and System Sciences, 69(4):593–616, 2004.

W. Maass, T. Natschläger, and H. Markram.
Real-time computing without stable states: a new framework for neural computation based on perturbations.
Neural Computation, 14:2531–2560, 2002.

Wolfgang Maass and Eduardo D. Sontag.
Neural Systems as Nonlinear Filters.
Neural Computation, 12(8):1743–1772, aug 2000.

Nuttapod Nuntalid, Kshitij Dhoble, and Nikola Kasabov.
EEG classification with BSA spike encoding algorithm and evolving probabilistic spiking neural network.
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7062 LNCS(PART 1):451–460, 2011.

P. Orponen.
A survey of continuous-time computation theory.
In *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Springer US, 1997.

Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar.
Optoelectronic reservoir computing.
Scientific reports, 2:287, jan 2012.

University of St.Gallen

References IX

Ali Rodan and Peter Tino.

Minimum complexity echo state network.

IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, 22(1):131–44, jan 2011.

Esther Ruiz.

Quasi-maximum likelihood estimation of stochastic volatility models.

Journal of Econometrics, 63:284–306, 1994.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini.

Group Sparse Regularization for Deep Neural Networks.

jul 2016.

Rodrigo Sacchi, Mustafa C. Ozturk, Jose C. Principe, Adriano A. F. M. Carneiro, and Ivan N. da Silva.

Water inflow forecasting using the echo state network: a Brazilian case study.

In *2007 International Joint Conference on Neural Networks*, pages 2403–2408. IEEE, aug 2007.

David A. Sprecher.

A representation theorem for continuous functions of several variables.

Proceedings of the American Mathematical Society, 16(2):200, apr 1965.

David A. Sprecher.

A numerical implementation of Kolmogorov's superpositions.

Neural Networks, 9(5):765–772, 1996.

David A. Sprecher.

A numerical implementation of Kolmogorov's superpositions II.

Neural Networks, 10(3):447–457, 1997.

University of St.Gallen

References X

Hava T Siegelmann and Eduardo D. Sontag.

On the computational power of neural nets.
In *COLT '92 Proceedings of the fifth annual workshop on Computational learning theory*, pages 440–449, 1992.

Hava T Siegelmann and Eduardo D. Sontag.

Analog computation networks via neural networks.
Theoretical Computer Science, 131:331–360, 1994.

Junji Shimada and Yoshihiko Tsukuda.

Estimation of Stochastic Volatility Models: An Approximation to the Nonlinear State Space Representation.
Communications in Statistics - Simulation and Computation, 34(2):429–450, apr 2005.

Xiang Sun and Xiang.

The Lasso and its implementation for neural networks.
1999.

Stephen J Taylor.

Modelling Financial Time Series.
John Wiley & Sons, Chichester, 1986.

Stephen J Taylor.

Asset Price Dynamics, Volatility, and Prediction.
Princeton University Press, Princeton, 2005.

A. N. Tikhonov.

On the stability of inverse problems.
Dokl. Akad. Nauk SSSR, 39(5):195–198, 1943.

University of St.Gallen

References XI

Fabian Triefenbach, Azarakhsh Jalalvand, Benjamin Schrauwen, and Jean-Pierre Martens.
Phoneme recognition with large hierarchical reservoirs.
Advances in Neural Information Processing Systems 23, 23:1–9, 2010.

Misha Tsodyks, Klaus Pawelzik, and Henry Markram.
Neural networks with dynamic synapses.
Neural Computation, 10(4):821–835, may 1998.

D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt.
An experimental unification of reservoir computing methods.
Neural Networks, 20:391–403, 2007.

F. Wyffels and B. Schrauwen.
A comparative study of Reservoir Computing strategies for monthly time series prediction.
Neurocomputing, 73(10):1958–1964, 2010.

Francis Wyffels, Benjamin Schrauwen, and Dirk Stroobandt.
Using reservoir computing in a decomposition approach for time series prediction, 2008.

University of St.Gallen