Improving noise robustness of time-delay reservoir computing

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2022

Zichen Kang

Publications and Presentations

Papers

- Zichen Kang, Sho Shirasaka, and Hideyuki Suzuki, "Optimizing input mask for maximum memory performance of time-delay reservoir subjected to state noise," Nonlinear Theory and Its Applications, IEICE, Vol. 12, No. 4, pp. 662-673 (2021).

Presentations

- Zichen Kang, Sho Shirasaka, and Hideyuki Suzuki, "Memory Capacity of Reservoir Computers based on Coupled Time-delay Systems," Proceedings of the 2020 International Symposium on Nonlinear Theory and its Applications, p. 263, Nov. 2020.

Acknowledgements

I would like to express my deep gratitude to Prof. Hideyuki Suzuki, my research supervisor, for his invaluable advice, continuous support, and patience during my PhD study. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would also like to thank Dr. Sho Shirasaka for his enthusiastic encouragement and useful critiques of this research work.

I am heartily grateful to the members of my thesis committee, Prof. Jun Tanida, Prof. Hiroshi Morita and Prof. Masayuki Numao for their constructive comments, suggestions for this thesis.

I would like to thank all the members of the Suzuki laboratory. It is their kind help and support that has made my study and life in Japan a wonderful time.

My grateful thanks are also extended to Prof. Atsushi Yagi for his encouragement and support all through my studies.

I would like to thank my wife Tan Cong for her love and constant support.

I would like to thank my friend Yang Jian for a cherished time spent together in the lab, and in social settings.

Contents

Chapter 1

Introduction

Classical computation may be baldly interpreted as the computers embodied in the Turing/von Neumann paradigm. Nowadays it has been incredibly successful for information processing. However, the increased demand due to highly complex computational tasks, such as speech recognition or chaotic time series prediction, has motivated the search for advanced unconventional computation [1].

In the context of nonlinear science, many artificial or natural dynamical systems driven by external input signals can be regarded as performing real-time computations of these signals. A high-performance, low-power computing system is our brain, which outperforms the modern classical computers in tasks of vision, audition, and pattern recognition even when provided poorly-conditioned inputs. The brain's computational units are neurons. The collection of these neurons connected to each other composes a neural network. With regard to the processing and communicating information, the neural network is plastic. The form of information processing in the neural network has inspired to design the artificial neural networks that mimic the brain's information processing capabilities.

A simple case of the artificial neural network is an adaptive nonlinear information processing system that combines numerous processing units without internal loops. It is meaning that the same neuron never propagates the input signal twice. This structure is referred to as feedforward neural network, which is trained by using the backpropagation algorithm [2]. If the processing units or neurons are combined by the recurrent connections, both the current input and output are influenced by the information from prior inputs. Such type of artificial neural network is called recurrent neural network. The recurrent neural network is a dynamical system whose states retain the information of all previous input signals. This design makes the recurrent neural network good at dealing with sequences from time series data. Speech recognition, image captioning, and time series prediction all require that a model produce outputs that are sequences. As the number of neurons increases, the potential expres-

sive power (the ability to approximate functions [3]) of a recurrent neural network grows exponentially [4]. However, the training procedure of the recurrent neural network has long been considered to be challenging. For example, the problem due to vanishing/exploding gradients, occur when backpropagating errors across many time steps [2].

Initially, reservoir computing is introduced as the Echo state network [5] depicted in Fig.1.1, which is a special type of recurrent neural network that can avoid the vanishing and exploding gradient problem. At the same time, a bio-inspired view suggests a computational model of reservoir computing that can process a multimodal input in real time [6]. These studies emphasize that the concept of the reservoir is referred to as a high dimensional dynamical system with a rich space of internal states. Generally, the reservoir consists of a mass of artificial neurons interconnected randomly. However, recent studies have revealed alternative implementations of the reservoirs, such as implementing a single nonlinear physical node with a delay line as a reservoir [7]–[14], using the cellular automaton as a reservoir [15], [16], exploiting the dynamics of road traffic as a reservoir [17], and employing a bucket of water as a reservoir [18], [19]. The term "reservoir" can be interpreted as certain nonlinear dynamical systems applied to performing computation. Indeed, "reservoirs" can be found everywhere in nature. For example, calm water with no ripples disturbing the surface can be seen as a reservoir without external perturbation or input. Information of input signal may be encoded in a series of drops of water. When the series of drops drip into the calm water, the information is transmitted and converted to spatial patterns of water waves. In this case, water undertakes the transformation from the low-dimensional input space into a high-dimensional feature space of waves in parallel.

In this work, we focus on a new computational methodology, called the time-delay reservoir (TDR) [7], realizes the reservoir using a single nonlinear physical node with a delay line, as depicted in Fig.1.2. The delay line is a chain of points placed equidistantly in time, which are denoted as virtual neurons. In the TDRs, the virtual neurons play a role as the reservoir layer, i.e., they are an alternative to the artificial neural network in conventional reservoir computing. The virtual neurons can exhibit high-dimensional transient responses due to the time delay. A major advantage of the TDRs is that they can be easily implemented in hardware. This novel paradigm has attracted several hardware implementations based on analog electronics [7], electromechanical devices [20] and opto-electronic devices [8], [21].

In the TDRs, their surrounding local environment, thermal noise [22], other noise related fluctuations [23] and quantization noise [10] have a non-negligible impact on their performance. It is necessary to reduce both internal and external noise effects by using noise mitigation techniques. Indeed, the noise susceptibility of the TDRs is closely related to their robustness and adaptability.

The purpose of this study is to improve the noise robustness of the TDRs. We

consider a general model of the TDRs based on a nonlinear dynamical system with time-delayed feedback in the presence of noise (see Chapter 2). Short-term memory [24] is a significant capacity of such TDRs and is easily impaired by noise. Recently, a particular method of measuring the memory trace degraded in the presence of noise has been proposed in the context of the Fisher information [25]. For single node delay-based reservoir computing, a method to optimize the input mask in a task-independent manner is developed for improving memory performance in the presence of state noise (see Chapter 3). We theoretically show that the input mask obtained as the maximal principal component of the spatial Fisher memory matrix optimizes the memory performance of such TDRs with small input signal and Gaussian state noise. The single Mackey-Glass oscillator is used to demonstrate the effectiveness of the proposed method via benchmark tasks strongly depending on memory (see Chapter 4). Compared with the existing optimization method, the proposed one is running-time-efficient and significantly improves memory performance in the presence of state noise. The memory-nonlinearity trade-off in view of the input masks is investigated using the chaotic times series prediction tasks. We also show that the echo state property of such TDRs impaired by state noise can be improved by the optimized input masks via the consistency parameter. This method for input mask optimization is extended for the hierarchical deep TDRs coupled unidirectionally (see Chapter 5). The coupled Ikeda systems are used to illustrate the proposed method. Analogously to the case of the single Mackey-Glass oscillator, we confirm the efficacy of the proposed method on the benchmark memory tasks and investigate the memory-nonlinearity trade-off by using chaotic times series prediction tasks. We also show that the consistency of each layer can be improved by the optimized input masks.

1.1 Echo state network (ESN)

The Echo State Network [5], [24], [26] is a discrete-time recurrent neural network composed of a sparse, random collection of analog neurons as illustrated in Figure 1.1. The neurons in the hidden layer communicate via fixed weighted connections. The model of the neurons applies an activation function (generally a saturation function) to process the sum of its weighted inputs. The activation state $\boldsymbol{x}(n)$ of these neurons at time n are referred to as the echo states that hold the "echo" of the input history [26]. A fading memory [27] (see section 1.4), which is strictly an input/output property, ensures that the input history can be successfully propagated and held in the ESN. This property is provided by the intrinsic dynamics of the ESN.

The network dynamics is generally modelled as follows. Assume that the ESN constructs with K input units, N hidden neurons and L output units. The input units at time n are $\boldsymbol{u}(n) = (u_1(n), \ldots, u_K(n))^\top \in \mathbb{R}^K$. A $N \times K$ weight matrix

$W_{\text{in}} \in \mathbb{R}^{N\times K}$ is applied to the input stream. At every step n, the hidden layer with an $N \times N$ interconnection matrix $W \in \mathbb{R}^{N\times N}$, generates an internal state $\boldsymbol{x}(n) = (x_1(n), \ldots, x_N(n))^\top \in \mathbb{R}^N$, which constitutes its current response to preceding inputs $\boldsymbol{u}(t)$ for $t \leq n$. Assembling these components gives the neuron state update equation at every time step n as

$$\boldsymbol{x}(n) = (1-\alpha)\boldsymbol{x}(n-1) + \alpha \boldsymbol{f}(W_{\text{in}}\boldsymbol{u}(n) + W\boldsymbol{x}(n-1)). \tag{1.1}$$

Here $\boldsymbol{f} = (f_1, \ldots, f_N)$ are the activation function of the internal neurons.

The above expression defines a reservoir map Φ as

$$\boldsymbol{x}(n) = \Phi(\boldsymbol{x}(n-1), \boldsymbol{u}(n)), \tag{1.2}$$

which transform the input signal into a high-dimensional feature space. This is achieved through a large collection of interconnected neurons. Then the preferred features can be extracted in a linear combination of internal states to create the outputs $\boldsymbol{y}(n) = (y_1(n), \ldots, y_L(n))^\top \in \mathbb{R}^L$ as

$$\boldsymbol{y}(n) = \boldsymbol{f}^{\mathbf{out}}(W_{\text{out}}\boldsymbol{x}(n)), \tag{1.3}$$

where $\boldsymbol{f}^{\mathbf{out}} = (f_1^{\text{out}}, \ldots, f_L^{\text{out}})$ are the output functions of output units, and $W_{\text{out}} \in \mathbb{R}^{L\times N}$ is the connections to the output units. The readout function should be memoryless map to generate a desired output. It should only access the history information of input signals through the internal state $\boldsymbol{x}$.

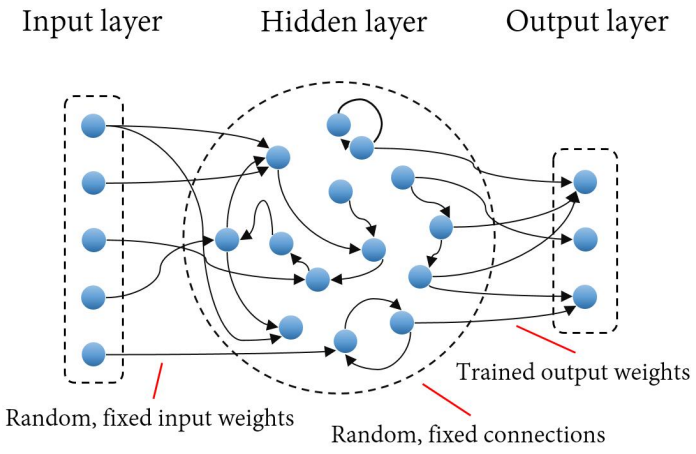


Figure 1.1: Schematic of echo state networks.

1.2 Time-delay reservoir (TDR)

Previous research has shown the huge computational processing power in even the simplest time-delay dynamical system within the context of machine learning, such as attractor reconstruction [28], speech recognition [20], [21]. In general, the dynamics of the time-delay system is governed by the solution of a time-delay differential equation of the form

$$\dot{x}(t) = F(x(t), x(t-\tau)) \tag{1.4}$$

with any given nonlinear function F and delay time τ. Because the dynamical variable $x(t)$ depends on the information of the system during the continuous time interval $[t-\tau, t]$, its phase space is infinite dimensional. This is the key to realizing a high-dimensional mapping from the input space into a high-dimensional feature space.

Time-delay reservoir [7] is an implementation of reservoir computing with a new computational methodology that the network is replaced by a single nonlinear dynamical node subjected to delayed feedback. The time-delay system is applied to emulate a large interconnected network by creating virtual neurons within the delay line using time-multiplexing at the input. The time-continuous input stream $u(t)$ that is sampled and held for the system delay τ defines a step input function $I(t)$ as

$$I(t) = u(k) \quad for \quad \tau k \leq t < \tau(k+1). \tag{1.5}$$

In the next step, a matrix $M \in \mathbb{R}^{N \times K}$ (called input mask), which corresponds to the input-reservoir connections of the ESNs, defines the coupling weights from $I(t)$ to the virtual neurons as

$$J(t) = M \times I(t). \tag{1.6}$$

The input mask M is often chosen randomly to break the symmetry between the virtual neurons. To feed the input sequence $J(t)$ into the nonlinear node, the dynamical variable $x(t)$ based on the interaction of $J(t)$ is modeled as

$$\dot{x}(t) = F(x(t), x(t-\tau) + \gamma J(t)), \tag{1.7}$$

where γ is the input gain.

The number of virtual neurons in the system is defined by N equidistant points separated in time $\theta = \tau/N$ along the delay interval τ. Thus, in TDRs, the states of virtual neurons is always discretized in time. When $(n-1)\tau < t \leq n\tau$, the i^{th} component of the virtual neurons $\boldsymbol{x}(n) = (x_1(n), x_2(n), \ldots, x_N(n))^\top \in \mathbb{R}^N$ is given by

$$x_i(n) = x(n\tau - (N-i)\theta) \quad for \quad i = 1, 2, \ldots, N. \tag{1.8}$$

as illustrated in Figure 1.2. Generally, the output function of TDRs is considered to be an identity function, so that the output is computed according to

$$\boldsymbol{y}(n) = W_{\text{out}} \boldsymbol{x}(n). \tag{1.9}$$

It should be noted that the output $\boldsymbol{y}(n)$ is also discrete due to the virtual neurons. Therefore, the speed of information processing depends on the state update given by the length τ of the delay line.

One attractive feature of TDRs is the simplicity of architecture. They may be implemented in a wide range of nonlinear system with delayed coupling. Examples of TDRs so far include analog circuits [7], electromechanical devices [20] and opto-electronic devices [8], [11], [12], [21].

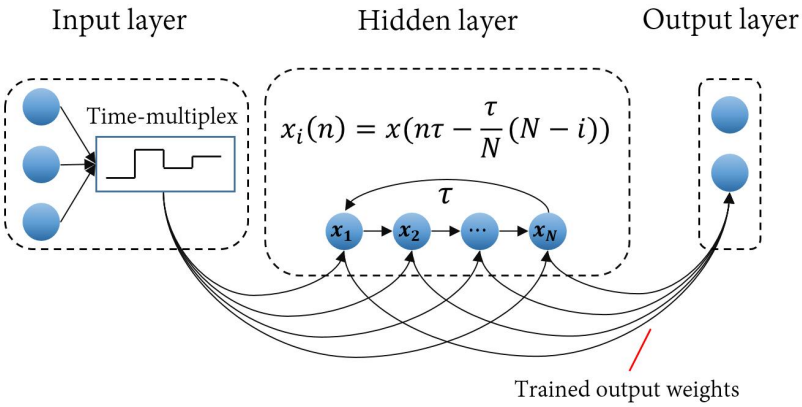


Figure 1.2: Schematic of time-delay reservoirs.

1.3 Training Reservoirs

It is well-known that although recurrent neural networks are potentially powerful approximators of dynamics [29], in practice, it is not easy to train properly. The main reasons are the local optima and vanishing/exploding gradients [2], [30]. In the training of reservoir computing, only the connections between the reservoir neurons and the output units are adapted with a simple method such as linear regression so as to generate desired output [5]. There are many available training methods in this diverse field. Here we introduce two commonly used methods, namely ridge regression learning algorithm and Recursive Least Squares algorithm. In the following, the output function of the output units is always considered to be an identity function, unless noted otherwise.

1.3.1 Off-line training

Traditionally, the process of training a reservoir system involves providing the ridge regression learning algorithm with target data $\boldsymbol{y}^{\text{target}}$ to learn from [31]. Given an es-

timate of the desired response $\boldsymbol{y}$ obtained by Eq.1.9, the reservoir system is trained by minimizing the training error between the desired response $\boldsymbol{y}^{\text{target}}$ and the estimation $\boldsymbol{y}$, which is defined by the difference

$$\boldsymbol{e} = \boldsymbol{y}^{\text{target}} - \boldsymbol{y}. \tag{1.10}$$

Thus the cost function is defined as the root mean square error (RMSE)

$$W_{\text{out}} = \arg\min_{W_{\text{out}}} \mathbf{E}\left[tr(\boldsymbol{e}\boldsymbol{e}^{\top})\right] + \lambda tr(W_{\text{out}} W_{\text{out}}^{\top}), \tag{1.11}$$

where $\mathbf{E}$ denotes the expectation operator, and λ is a regularization coefficient. There are standard well-known ways to solve Eq.1.11 [32]. The solution takes the form

$$W_{\text{out}} = \boldsymbol{y}^{\text{target}} \boldsymbol{x}^{\top} \left(\boldsymbol{x}\boldsymbol{x}^{\top} + \lambda \boldsymbol{I}\right)^{-1}, \tag{1.12}$$

where $\boldsymbol{I}$ is the identity matrix. Setting $\lambda = 0$ gives the same method for solving linear regression.

1.3.2 On-line training

The on-line training method is implemented by the recursive Least Squares (RLS) algorithm [33]. In contrast to the ordinary method of least squares, the training error between the desired response $\boldsymbol{y}^{\text{target}}(i)$ and the estimation $\boldsymbol{y}(i)$ at time n is defined as

$$\begin{aligned} \hat{\boldsymbol{e}}(i) =& \boldsymbol{y}^{\text{target}}(i) - \boldsymbol{y}(i) \\ =& \boldsymbol{y}^{\text{target}}(i) - W_{\text{out}}(n)\boldsymbol{x}(i). \end{aligned} \tag{1.13}$$

Then the cost function is composed of the sum of two components

$$W_{\text{out}} = \arg\min_{W_{\text{out}}} \sum_{i=1}^{n} \lambda^{n-i} tr\left[\hat{\boldsymbol{e}}(i)\hat{\boldsymbol{e}}^{\top}(i)\right] + \delta\lambda^{n} tr\left[W_{\text{out}}(n) W_{\text{out}}^{\top}(n)\right], \tag{1.14}$$

where δ is a positive real number called the regularization parameter, and λ^{n-i} is a exponential weighting factor, or forgetting factor. This weighting factor has the property that $0 < \lambda^{n-i} \leq 1$. If $\lambda = 1$, it is the ordinary method of least squares. The inverse of $1 - \lambda$ is a measure of the "memory" of the algorithm. The special case $\lambda = 1$ corresponds to the "infinite memory".

The RLS algorithm is described as follows:
Initialize the algorithm by setting

$$\begin{aligned} W_{\text{out}}(0) &= \mathbf{0}, \\ \boldsymbol{P}(0) &= \delta^{-1}\boldsymbol{I}, \end{aligned} \tag{1.15}$$

and

$$\delta = \begin{cases} \text{small positive constant for high SNR} \\ \text{large positive constant for low SNR} \end{cases}. \tag{1.16}$$

For each instant of time, $n = 1, 2, \ldots$, compute

$$\begin{aligned} k(n) &= \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{x}(n)}{1+\lambda^{-1}\boldsymbol{x}(n)^{\top}\boldsymbol{P}(n-1)\boldsymbol{x}(n)}, \\ \hat{\boldsymbol{e}}(n) &= \boldsymbol{y}^{\text{target}}(n) - W_{\text{out}}(n-1)\boldsymbol{x}(n), \\ W_{\text{out}}(n) &= W_{\text{out}}(n-1) + k(n)\hat{\boldsymbol{e}}(n) \end{aligned} \tag{1.17}$$

and

$$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}k(n)\boldsymbol{x}^{\top}(n)\boldsymbol{P}(n-1). \tag{1.18}$$

Here $\boldsymbol{P}(n)$ is referred to as the inverse correlation matrix, which can be viewed as the covariance matrix of the RLS estimate W_{out}. For more information on the RLS algorithm see [34].

1.4 Desirable property

1.4.1 Echo state property

In the context of neuroscience, a crucial aspect of the nervous system is generally characterized as the relationship between external stimuli and the corresponding response of a network of excitable elements. In reservoir computing, the recurrent topology creates memory by retaining the previous response of reservoir neurons. The present response of reservoir neurons can be characterized as an "echo" of the recent past input, but the influence of the remote past input gradually fades out. In other words, the reservoir system requires the echo state property (ESP) [5], [24], [35], [36].

As the present response of reservoir neurons is independent of the initial conditions, the ESP guarantees that real-time computing is possible. The original definition of the ESP is as follows [5], [24].

Definition 1 *(Echo state property) Assume that the reservoir network has no output feedback connections. The symbol* $\boldsymbol{u}^{\infty} = (\ldots, \boldsymbol{u}(n-2), \boldsymbol{u}(n-1), \boldsymbol{u}(n), \ldots)$, $\boldsymbol{u}(n) \in \mathbb{R}^K$, $n \in \mathbb{Z}$, *denotes a infinite input sequence. With any input sequence* $\boldsymbol{u}^{\infty}$ *driving the reservoir network, any two reservoir state sequences* $\boldsymbol{x}^{\infty} = (\ldots, \boldsymbol{x}(n-2), \boldsymbol{x}(n-1), \boldsymbol{x}(n), \ldots), \boldsymbol{x}(n) \in \mathbb{R}^N$, $n \in \mathbb{Z}$ *and* $\hat{\boldsymbol{x}}^{\infty} = (\ldots, \hat{\boldsymbol{x}}(n-2), \hat{\boldsymbol{x}}(n-1), \hat{\boldsymbol{x}}(n), \ldots)$, $\hat{\boldsymbol{x}}(n) \in \mathbb{R}^N$, $n \in \mathbb{Z}$ *are derived from a reservoir map* Φ *as*

$$\begin{aligned} \boldsymbol{x}(n) &= \Phi\left(\boldsymbol{x}(n-1), \boldsymbol{u}(n)\right), \\ \hat{\boldsymbol{x}}(n) &= \Phi\left(\hat{\boldsymbol{x}}(n-1), \boldsymbol{u}(n)\right). \end{aligned} \tag{1.19}$$

It must hold that $\boldsymbol{x}(n) = \hat{\boldsymbol{x}}(n)$.

If the reservoir system satisfies the ESP for any input sequence $\boldsymbol{u}^{\infty}$, there exits an input echo function ε so that the current reservoir state is

$$\boldsymbol{x}(n) = \varepsilon(\ldots, \boldsymbol{u}(n-2), \boldsymbol{u}(n-1), \boldsymbol{u}(n)). \tag{1.20}$$

Eq.1.20 implies that the reservoir state only depends on the input sequence.

The input echo function ε has a desirable continuity property, called "fading memory" [27].

Definition 2 *(Fading memory) A input echo function* ε *has fading memory if there is a decreasing function* $w : \mathbb{Z}_+ \to (0, 1]$, $\lim_{n\to\infty} w(n) = 0$, *such that for any left infinite input sequence* $\boldsymbol{u}^{-\infty} = (\ldots, \boldsymbol{u}(-2), \boldsymbol{u}(-1), \boldsymbol{u}(0))$, $\boldsymbol{u}(n) \in \mathbb{R}^K$, $n \in \mathbb{Z}^-$ *and* $\epsilon > 0$, *there is a* $\delta > 0$ *such that for all* $\hat{\boldsymbol{u}}^{-\infty}$ *which satisfies*

$$\sup_{n\leq 0} \|\boldsymbol{u}(n) - \hat{\boldsymbol{u}}(n)\|_2 w(-n) < \delta, \tag{1.21}$$

then

$$\|\varepsilon(\boldsymbol{u}^{-\infty}) - \varepsilon(\hat{\boldsymbol{u}}^{-\infty})\|_2 < \epsilon. \tag{1.22}$$

This continuity property had been proofed for the reservoir computing satisfying the ESP [5].

1.4.2 Consistency as a way of characterizing the ESP

In practice, the conditions of Def.1 are hard to check when the reservoir system is subjected to state noise. Recent studies have revealed that the consistency test of the system response to a complex drive signal can be applied to measure the level of the ESP for the reservoir system [37], [38].

Consistency is essential for information transmission in dynamical systems. It is based on the replica test, which measures the reproducibility of the response signals in a nonlinear dynamical system driven by a repeated signal, starting from different initial conditions. Some phenomena corresponding to the consistency have been studied, such as generalized synchronization in chaotic systems [39]–[41].

In order to investigate the characteristics of consistency, the **consistency parameter** C is defined as the cross correlation of two temporal response signals x and y normalized by the product of their standard deviations as

$$C = \frac{\mathbf{E}[(x - \mu_x)(y - \mu_y)]}{\sigma_x \sigma_y}. \tag{1.23}$$

Here μ_x and μ_y are the mean values of two response signals; σ_x and σ_y are the standard deviations of the two response signals. The value of the consistency parameter C is lying in the range $[-1, 1]$, which $C = 1$ is indicating perfect correlation, $C = 0$ is indicating no correlation, and $C = -1$ is indicating perfect anti-correlation.

In this study, we use the consistency parameter to measure the level of the ESP for the reservoir system in the presence of state noise. Considering consistency in the terminology of reservoir computing, the complete consistency $C = 1$ is equivalent to the ESP, and $C \leq 0$ is the lack of the ESP [37].

1.4.3 Optimal reservoir working regime

To design an optimal reservoir one should find a well-working regime such that it can produce sufficiently complex signals, and have the ESP to ensure reproducibility and reliability of response signal with the driving system.

The motion of the dynamical system in the phase space is described as trajectories. Under certain conditions, a trajectory can converge towards a steady state called attractor, such as fixed point, and chaotic [42]. Both theoretical and empirical studies suggest that if the reservoir working regime is around a stable fixed point, then for inputs small enough the reservoir system will satisfy the ESP [5], [7], [38]. The asymptotic stability of the fixed point implies that the transient state of the dynamical system is independent of the initial condition, and in [27] it has been demonstrated that there is a connection between the fading memory and unique steady state (global stability). Recent studies have shown that the consistency of chaotic systems is dependent on the amplitude of the drive signal, such as generalized synchronization in chaotic systems [39]–[41]. However, it is easy to make the reservoir system operate well in a chaotic regime. In this case, the reservoir system is highly sensitive to initial conditions and noise. Namely, the fading memory is no longer available. Legenstein and Maass [43] have shown that the ability of some reservoir networks to achieve the desired computational outcome is maximized at a critical point near the edge of chaos. It should be noted that the edge of chaos as an optimal working regime for a reservoir system is not, in general, true, and a counterexample has been introduced [44].

1.5 Applications

Reservoir systems are generally very suited for highly complex computational tasks. In most cases, very good performance can be attained without having to care too much about specifically setting any of the reservoir parameters. Some successful applications of ESNs have been reported in literature. In robotics, ESNs have been

used to perform event detection [45], [46] and several application in the Robocup competitions [47], [48]. In the field of digital signal processing, it also has been applied in speech recognition [49] and noise modeling [26]. The use of ESNs for chaotic attractor reconstruction and short-term prediction has been reported in [24], [28], [50], [51]. On the other hand, TDRs have also successfully attracted a wide range of real world engineering applications. The spoken digit recognition task have been implemented on various TDRs based on different types of hardware, such as analog electronics [7], [10], electromechanical devices [20], optoelectronic devices [8], [11], [12], [21]. In [10], a TDR with an analog electronic circuit has been used in a classification problem and a chaotic time-series prediction task. Novel applications of deep TDRs in cyber-security and wireless communication also have been introduced [52].

1.6 Overview of this thesis

The main goal of this thesis is to improve the noise robustness of the TDRs. we develop a method to optimize reservoir performance of the TDRs subjected to state noise with respect to input masks in the context of Fisher information.

- In Chapter 2, the general model of TDRs subjected to state noise is introduced. We elaborate on the architecture of TDRs and show the noise-induced performance degradation by way of the consistency parameter.
- In Chapter 3, we develop a method to optimize reservoir performance of the TDRs subjected to state noise with respect to input masks in the context of Fisher information. We theoretically show that the input mask obtained as the maximal principal component of the spatial Fisher memory matrix optimizes the memory performance of such TDRs with Gaussian state noise.
- In Chapter 4, for single node delay-based reservoir computing, a method to optimize the input mask in a task-independent manner is developed for improving memory performance in the presence of state noise. The single Mackey-Glass oscillator is used to demonstrate the availability of the proposed method.
- In Chapter 5, for the hierarchical deep TDRs coupled unidirectionally, a method for input mask optimization is developed for the entire reservoir space. The coupled Ikeda systems are used to illustrate the proposed method.
- Chapter 6 is conclusions.

Chapter 2

TDRs subjected to state noise

The physical implementation of TDRs has attracted considerable attention in the context of machine learning lately. It reveals the huge information processing power of delayed dynamical systems. In practice, the impacts of noise propagation in recurrent reservoir layers cannot be ignored. We consider some randomness in TDRs such that we can obtain a more realistic mathematical model Eq.2.1 of the situation, as follows:

$$\dot{x}(t) = -x(t) + f(x(t-\tau), J(t), z(t)), \tag{2.1}$$

where $x(t) \in \mathbb{R}$ is a dynamical variable, f is a smooth nonlinear function with a delay feedback term $x(t-\tau)$, $J(t)$ is time-multiplexed input signal over the delay period τ, and $z(t) \in \mathbb{R}$ is state noise.

2.1 State noise

We start by considering the source and the impact of noise upon the time-delay node. Assume that without external injected information $J(t) = 0$, the dynamical variable x is perturbed by the state noise,

$$\begin{aligned} \dot{x}(t) &= -x(t) + f(x(t-\tau) + \gamma J(t) + z(t)) \\ &= -x(t) + f(x(t-\tau) + z(t)) \end{aligned} \tag{2.2}$$

where γ is the input gain.

In general, the state noise is according to three arguments. First, it accounts for the loss of information in the input preprocessing stage, such as input referred noise, or in the process of extracting virtual neurons, such as the quantization noise that results from the analog-digital conversion [10]. Second, the time-delay system constructing TDRs can exhibit complex dynamical behaviors, such as chaos, meaning

that the response trajectory of the time-delay system to the drive signals is unstable. In the case of multistability, the response trajectory near a stable equilibrium can be driven closer to another stable equilibrium when the level of input signals is too high or uncommonly large. These complex dynamical behaviors can amplify different initial conditions and small noise, resulting in different responses despite receiving the same drive. Third, the intrinsic noise originating from hardware components, such as thermal noise in electronic circuits [10] or spontaneous emission in semiconductor lasers [23], contributes to $z(t)$.

In this work, the state noise $z(t)$ is defined as the total noise which is contributed by a large number of noise sources from different physical components and the surrounding local environment [9], [10], [22], [23]. Assume that $z(t)$ is additive and uncorrelated. Then the state noise $z(t)$ can be approximated by a Gaussian distribution. We consider that $z(t)$ refers to a white Gaussian noise process with mean $\mu(z(t)) = 0$ and covariance $\langle z(t_1)z(t_2)\rangle = \epsilon\delta_{t_1,t_2}$ (δ is the Dirac delta function).

2.2 Time-discretized model

By discretizing consecutive delay intervals such that

$$\begin{aligned} \boldsymbol{x}(n) &= (x_1(n), x_2(n), \ldots, x_N(n))^\top, \quad x_i(n) = x(n\tau - (N-i)\theta), \\ \boldsymbol{J}(n) &= (J_1(n), J_2(n), \ldots, J_N(n))^\top, \quad J_i(n) = J(n\tau - (N-i)\theta), \end{aligned} \tag{2.3}$$

are parameterized by a discrete time $n \in \mathbb{Z}$ and a sampling length θ related to the number of virtual neurons N by $\theta = \tau/N$. Then the TDR construction can be visualized as a neural network approach. $x_i(n)$ denotes the i^{th} virtual neuron state at the n^{th} time step.

To consider a discrete version [53] of Eq.2.1:

$$x_i(n) - x_{i-1}(n) = -x_i(n) + f(x_i(n-1) + \gamma J_i(n) + \Delta z_i(n)) \tag{2.4}$$

where the increment $\Delta z_i(n) = z(n\tau - (N-i)\theta) - z(n\tau - (N-i+1)\theta)$ follows the normal distribution with zero mean and covariance $\langle \Delta z_i(k)\Delta z_j(l)\rangle = \epsilon\theta\delta_{k,l}\delta_{i,j}$. Note that, $\Delta z_i(n)$ in Eq.2.4 can include some noise sources which do not appear as white in Eq.2.1 by redefining its strength $\epsilon\theta$. For example, quantization noise, which crucially affects TDR's performance [10], can be considered as the discrete-time white Gaussian noise under some conditions [54].

The solutions of Eq.2.4 are described by the following recursive relation:

$$\begin{cases} x_i(n) = e^{-\xi} x_{i-1}(n) + \left(1 - e^{-\xi}\right) f\left(x_i(n-1) + \gamma J_i(n) + \Delta z_i(n)\right) \\ x_0(n) = x_N(n-1) \\ \xi = \log(1+\theta) \end{cases} . \tag{2.5}$$

The Eq.2.5 shows that the neuron state is updated by a linear combination of the previous neuron state in the same layer and a nonlinear function parameterized by the neuron state in the same location of the previous layer, the input, and the noise increment. The weight $e^{-\xi}$ is determined by the separation length θ between virtual neurons. If the distance θ is too small, the adjacent virtual neurons are too close to be independent. And in the other hand, a suitable small θ can increase the effective degrees of freedom of the virtual neurons. The recursions Eq.2.5 uniquely determine a smooth map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ in which the states of the virtual neurons are updated recursively:

$$\boldsymbol{x}(n) = F(\boldsymbol{x}(n-1), \boldsymbol{J}(n), \Delta\boldsymbol{z}(n)). \tag{2.6}$$

F is referred to as the **reservoir map**.

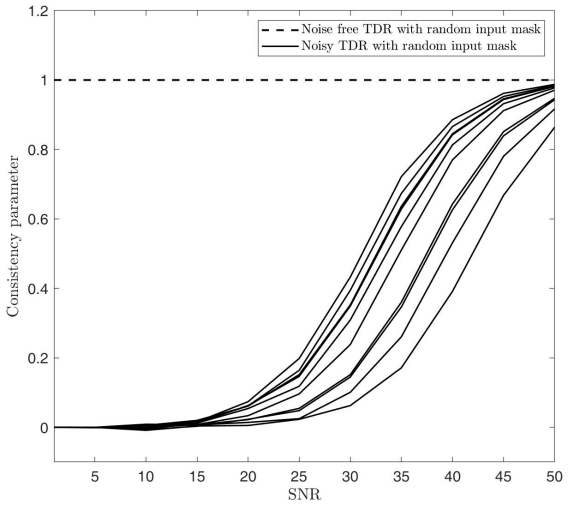
2.3 Noise-induced performance degradation

Being an inevitable consequence of hardware implementation of TDRs, noise can propagate along the reservoir network and potentially accumulate. Consequently, an important concern is that such TDRs might ultimately succumb to the detrimental impact of noise. A recent report has shown that the output of feed-forward neural networks (FNNs), as well as recurrent neural networks (RNNs) in analog hardware, is susceptible to noisy neurons considering a variety of network topologies and parameters [22].

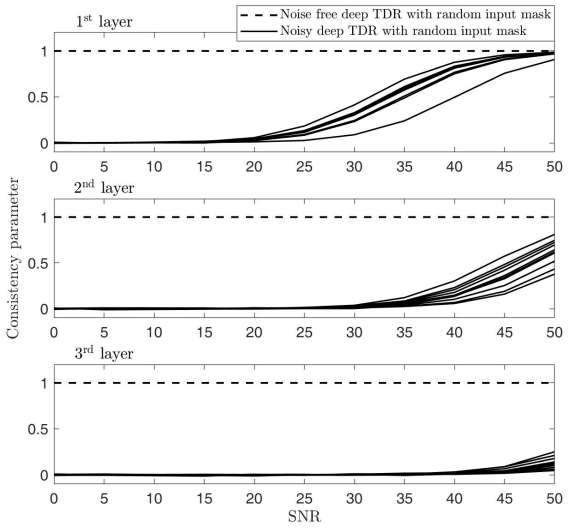
A fundamental question is that whether the reservoir systems subjected to state noise have the ESP. It should be noted that Lymburn et al. have shown that the consistency can be used to measure the level of the ESP for the reservoir system in the presence of state noise [37]. This method gives us an alternative way to check the effect of state noise on the given reservoir systems. In this study, we focus on two types of implementations of TDRs, which are the TDRs based on a Mackey–Glass type nonlinear node and the deep TDRs based on coupled Ikeda-type nonlinear nodes. In Figure 2.1 we plot how the consistency evolves for the two types of TDRs as a function of signal to noise ratio (SNR) and the input masks. The black dashed lines denote the consistency parameter of noise-free reservoirs with random input masks. As the values of these consistency parameter equal one, a well-defined reservoir (such as the working regime is around a stable fixed point) without the impact of noise always have the ESP. The case of the noisy TDRs based on the Mackey-Glass system is described by the black solid lines in Figure 2.1a. Unsurprisingly the consistency is sensitive to the negative effects of noise. Next, the impact of noise on the consistency is considered for a three-layer deep TDR based on coupled Ikeda delay systems. This is illustrated in Figure 2.1b. In this case, we calculate the consistency parameter for

each layer respectively. We find that The final layer is most susceptible to noise. The reason is that the noise originating from the first layer and the second layer potentially accumulates in the final layer.

In Figure 2.1, the input mask related to the pre-processing of the input signal (time-multiplexing) is the other chosen parameter. Generally, a random mask is mostly applied to the input signal in order to generate complex transient responses. Through the simulation results in Figure 2.1, we find that the input mask has a significant impact on the ESP of the noisy reservoir systems in the context of the consistency. Studies on the design of the input mask have been reported, such as a binary mask that maximizes the diversity of the reservoir states for a small set of the virtual neurons [55] and a chaotic mask that improves the performance of an optical TDR for a time-series prediction task [11]. Recent studies have investigated the short-term memory capacity of the echo state network in the context of consistency. It has been shown that an increase in consistency improves the short-term memory performance which may be degraded by state noise [37]. In this thesis, we optimize the input mask for the TDRs with a white Gaussian state noise in a computational time efficient manner within the context of the Fisher memory.



(a)



(b)

Figure 2.1: Consistency parameter as a function of signal to noise ratio (SNR) and input masks. (a) A TDR based on the Mackey-Glass system (see Chapter 4), for $\tau = 40$, $\eta = 0.7$, $\gamma = 0.1$ and $p = 1$, is used to calculate consistency parameter for 10 different input masks. (b) A three-layer deep TDR based on coupled Ikeda delay systems (see Chapter 5), with parameter setting as Table 5.1, is used to calculate consistency parameter for 10 different input masks. The black dashed lines and the black solid lines denote the consistency parameter of noise free reservoir systems with random input masks and noisy reservoir systems with random input masks.

Chapter 3

Optimized input mask for TDRs subjected to state noise

In this section, we develop a method to optimize the reservoir performance of the TDRs subjected to state noise with respect to input masks in the context of Fisher information.

3.1 Short-term memory capacity

In [24] Jaeger introduced the measurement of the short-term memory capacity of a reservoir. It quantifies the capacity that how many delayed versions of the input $u(n-k)$ can be reproduced by the reservoir. The variance of the delayed input signal can be reproduced is measured by correlating the delayed input signal with the trained output signal, summed over all delays. In terms of the squared correlation coefficient between the desired output $u(n-k)$ and the observed network output $y(n)$, the short-term memory capacity can be expressed as

$$\begin{aligned} \mathrm{MC} &= \sum_{k=1}^{\infty} \mathrm{MC}_k \\ &= \sum_{k=1}^{\infty} \frac{cov^2(u(n-k), y(n))}{\sigma^2(u(n))\sigma^2(y(n))}, \end{aligned} \tag{3.1}$$

where *cov* denotes the covariance operator, $\sigma^2(u(n))$ and $\sigma^2(y(n))$ is the variance of $u(n)$ and $y(n)$ respectively. Jaeger showed theoretically that the short-term memory capacity MC cannot exceed N for any recurrent neural network with i.i.d. input signal, where N is the number of neurons [24].

3.2 Fisher information as a measure of short-term memory

A different quantification of short-term memory for linear reservoirs subjected to a white Gaussian noise has been proposed in the context of Fisher information [25]. Recalling that the discrete noise increment $\Delta z(n)$ is derived from a memoryless process of a white Gaussian noise $z(t)$, with mean $\mu = 0$ and covariance $\langle \Delta z_i(k) \Delta z_j(l) \rangle = \epsilon\theta\delta_{k,l}\delta_{i,j}$. Then a given input sequence $\boldsymbol{u} := (\dots, u((n-2)\tau), u((n-1)\tau), u(n\tau), \dots)$ is considered as the parameter on which the probability of $\boldsymbol{x}$ depends. This means that the conditional distribution $p(\boldsymbol{x}(n)|\boldsymbol{u})$ varies with the input sequence $\boldsymbol{u}$. The average rate of change of the log-likelihood function $\log(p(\boldsymbol{x}(n)|\boldsymbol{u}))$ with respect to small perturbation in the input history is measured by the Fisher information matrix whose elements are defined as

$$\mathcal{I}_{l,k}(\boldsymbol{u}) = -E_{p(\boldsymbol{x}(n)|\boldsymbol{u})}\left[\frac{\partial^2}{\partial u((n-l)\tau)\partial u((n-k)\tau)} \log(p(\boldsymbol{x}(n)|\boldsymbol{u}))\right]. \tag{3.2}$$

This matrix captures the changes of the conditional distribution $p(\boldsymbol{x}(n)|\boldsymbol{u})$ with the signal history $\boldsymbol{u}$ changing.

In [56] Jeffreys introduced the Kullback–Leibler divergence as a measure of the difference between two probability distributions, and found the relationship with Fisher information. Assume that a slight perturbed input signal $\boldsymbol{u} + \delta\boldsymbol{u}$ is derived from the input signal $\boldsymbol{u}$ in the presence of noise. Then the Kullback–Leibler divergence between the two induced distributions $p(\boldsymbol{x}(n)|\boldsymbol{u})$ and $p(\boldsymbol{x}(n)|\boldsymbol{u} + \delta\boldsymbol{u})$ can generally be approximated by the Fisher information matrix Eq.3.2 as

$$D_{KL}(p(\boldsymbol{x}(n)|\boldsymbol{u} \parallel p(\boldsymbol{x}(n)|\boldsymbol{u} + \delta\boldsymbol{u})) = \frac{1}{2}\delta\boldsymbol{u}^\top \mathcal{I}(\boldsymbol{u})\delta\boldsymbol{u} + o(||\delta\boldsymbol{u}||^2). \tag{3.3}$$

Moreover, for the special case of the Gaussian family distributions $p(\boldsymbol{x}(n)|\boldsymbol{u})$ with the same dimension, the Kullback–Leibler divergence between the distributions is as follows [57]:

$$D_{KL}(p(\boldsymbol{x}(n)|\boldsymbol{u} \parallel p(\boldsymbol{x}(n)|\boldsymbol{u}')) = \frac{1}{2}(\mu - \mu')^\top \boldsymbol{C}^{-1}(\mu - \mu') \tag{3.4}$$

where μ is the mean of $p(\boldsymbol{x}(n)|\boldsymbol{u})$, μ' is the mean of $p(\boldsymbol{x}(n)|\boldsymbol{u}')$, and $\boldsymbol{C}$ is the covariance matrices of $p(\boldsymbol{x}(n)|\boldsymbol{u})$.

Through Eq.3.3 and Eq.3.4, we can obtain an alternative expression [25] of the Fisher information matrix Eq.3.2 as

$$\mathcal{I}_{l,k}(\boldsymbol{u}) = \frac{\partial\mu(\boldsymbol{u})^\top}{\partial u((n-l)\tau)}\boldsymbol{C}^{-1}\frac{\partial\mu(\boldsymbol{u})}{\partial u((n-k)\tau)}. \tag{3.5}$$

The matrix $\mathcal{I}_{l,k}(\boldsymbol{u})$ is generally called as the **Fisher memory matrix** [25], which measures short-term memory through the capacity of the past signal to perturb the present state of the reservoir neurons.

3.3 Fisher memory curve in TDRs

In this study, we are interested in the diagonal elements of the Fisher memory matrix $I_k = \mathcal{I}_{k,k}$, which quantify how much information about an input signal entering the network k time steps ago can be held by the reservoir neurons $\boldsymbol{x}(n)$. The collection of $\{I_k\}_{k=0}^{\infty}$ is referred to as the **Fisher memory curve**.

In section 2.2, we show that for a general TDR with state noise, such as Eq.2.1, the state of reservoir neurons is updated according to the reservoir map as Eq.2.6. Consider the general case when the TDRs operate in the neighborhood of an asymptotically stable fixed point, we can approximate it by the Jacobian linearization at that point. This approximation allows us to visualize the TDR as a general neural network with N nodes. Then we can measure the short-term memory capacity through the use of the Fisher information.

Assume that $\bar{x} \in \mathbb{R}$ is a stable equilibrium of the continuous time model according to Eq.2.1 with $u(t) = 0$ and $z(t) = 0$. It is equivalent to a stable fixed point of the reservoir map Eq.2.6 of the form $\bar{\boldsymbol{x}} := (\bar{x}, \bar{x}, \ldots, \bar{x})^\top \in \mathbb{R}^N$. If we initialize the TDR at the stable fixed point and apply $\boldsymbol{J}(n) = \boldsymbol{0}$, $\Delta\boldsymbol{z}(n) = \boldsymbol{0}$, its reservoir states will remain fixed at $\boldsymbol{x}(n) = \bar{\boldsymbol{x}}$ for all n. We consider the Jacobian linearization of the reservoir map Eq.2.6 about the equilibrium point $(\bar{\boldsymbol{x}}, \boldsymbol{0}, \boldsymbol{0})$. By performing the Taylor expansion of the right hand side and neglecting all higher (higher than the 1st) order terms, we obtain an expression of the form:

$$\begin{aligned}\boldsymbol{x}(n) =& F(\bar{\boldsymbol{x}}, \boldsymbol{0}, \boldsymbol{0}) + \left.\frac{\partial F}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x},\boldsymbol{J},\Delta\boldsymbol{z})=(\bar{\boldsymbol{x}},\boldsymbol{0},\boldsymbol{0})} (\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}) + \left.\frac{\partial F}{\partial \boldsymbol{J}}\right|_{(\boldsymbol{x},\boldsymbol{J},\Delta\boldsymbol{z})=(\bar{\boldsymbol{x}},\boldsymbol{0},\boldsymbol{0})} \boldsymbol{J}(n) \\ &+ \left.\frac{\partial F}{\partial \Delta\boldsymbol{z}}\right|_{(\boldsymbol{x},\boldsymbol{J},\Delta\boldsymbol{z})=(\bar{\boldsymbol{x}},\boldsymbol{0},\boldsymbol{0})} \Delta\boldsymbol{z}(n).\end{aligned} \tag{3.6}$$

For convenience, we define these three $N \times N$ Jacobian matrices as

$$A := \left.\frac{\partial F}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x},\boldsymbol{J},\Delta\boldsymbol{z})=(\bar{\boldsymbol{x}},\boldsymbol{0},\boldsymbol{0})}, \quad B := \left.\frac{\partial F}{\partial \boldsymbol{J}}\right|_{(\boldsymbol{x},\boldsymbol{J},\Delta\boldsymbol{z})=(\bar{\boldsymbol{x}},\boldsymbol{0},\boldsymbol{0})}, \quad C := \left.\frac{\partial F}{\partial \Delta\boldsymbol{z}}\right|_{(\boldsymbol{x},\boldsymbol{J},\Delta\boldsymbol{z})=(\bar{\boldsymbol{x}},\boldsymbol{0},\boldsymbol{0})}.$$

Recalling that the time-multiplexed signal $\boldsymbol{J}(n)$ is linearly dependent on the sampled input signal $u(n\tau)$ with the mask vector M. Then the reservoir state update equation

can be expressed as follows

$$\boldsymbol{x}(n) = \bar{\boldsymbol{x}} + A(\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}) + BMu(n\tau) + C\Delta\boldsymbol{z}(n). \tag{3.7}$$

Here A is an $N \times N$ constant matrix that is referred to as the connectivity matrix. The feedforward connections from the input signal into the reservoir layer are represented as the product of the matrix B and the mask M. The asymptotic stability of this fixed point requires that the spectral radius $\rho(A)$ is smaller than the one. But it is impossible to calculate the spectral radius $\rho(A)$ for any number of reservoir neurons. In [58] Grigoryeva et al. showed theoretically that for an arbitrary number of virtual neurons N, $\rho(A) \leq \|A\|_\infty < 1$ if and only if $|\frac{\partial f}{\partial x}(\bar{x}, 0, 0)| < 1$. This result relates the stability conditions for the discrete time system with the stability conditions for the continuous time system.

The linearized system Eq.4.6 allow us to measure the short-term memory capacity by using the Fisher information. Suppose that the initial condition of the reservoir map is prepared at sufficiently far past. Then its effect disappears and using Eq.4.6 the state of the virtual neurons at time n can be expressed as follows

$$\boldsymbol{x}(n) = \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} [A^k BMu((n-k)\tau) + A^k C\Delta\boldsymbol{z}(n-k)]. \tag{3.8}$$

Recalling that the increment $\Delta\boldsymbol{z}(t)$ follows the Gaussian distribution, the conditional distribution $p(\boldsymbol{x}(n)|\boldsymbol{u})$ is also Gaussian with mean $\mu = \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} A^k BMu((n-k)\tau)$ and covariance matrix $\boldsymbol{C}_n = \epsilon\theta \sum_{k=0}^{\infty} A^k CC^\top A^{k\top}$. The mean is only dependent on the input signal history linearly, and the covariance matrix is independent of the input signal. By substituting $\bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} A^k BMu((n-k)\tau)$ for μ in Eq.3.5, the Fisher memory matrix of the TDR at the equilibrium $(\bar{\boldsymbol{x}}, \boldsymbol{0}, \boldsymbol{0})$ takes the form

$$\mathcal{I}_{l,k} = M^\top B^\top A^{l\top} \boldsymbol{C}_n^{-1} A^k BM \tag{3.9}$$

and is independent of the input signal. The Fisher memory curve $\{I_k\}_{k=0}^{\infty}$ in TDRs can be characterized as the collection of the diagonal elements of Eq.3.9.

3.4 Optimized input mask for noisy TDRs

In this section, we optimize the Fisher-information based short-term memory capacity in terms of the input mask. The total memory capacity of a TDR is defined as

$$I_{\text{tot}} = \sum_{k=0}^{\infty} I_k. \tag{3.10}$$

By substituting Eq.3.9 for I_k in Eq.3.10

$$I_{\mathrm{tot}} = \sum_{k=0}^{\infty} I_k = M^{\top} I^s M, \tag{3.11}$$

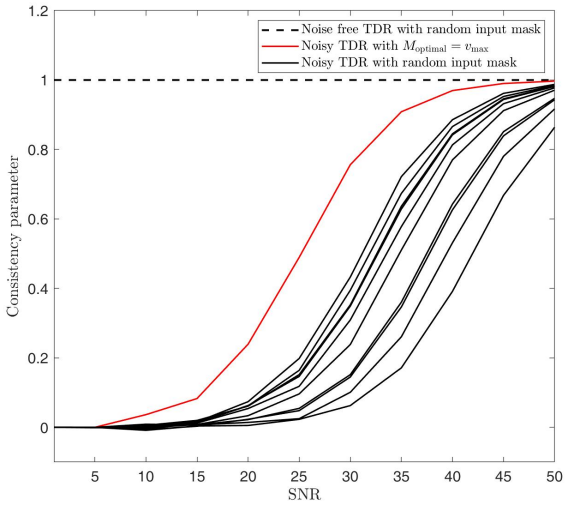
where

$$I^s = \sum_{k=0}^{\infty} B^{\top} A^{k\top} \boldsymbol{C}_n^{-1} A^k B \tag{3.12}$$

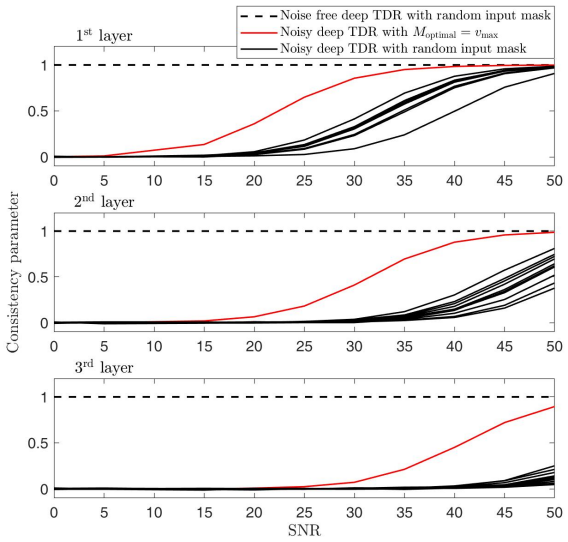
is called the spatial Fisher memory matrix. Since I^s is a real symmetric matrix, its eigenvalues are real and eigenvectors associated with the different eigenvalues are orthogonal. Assume that $\lambda_{\max} \geq \lambda_2 \geq \ldots \geq \lambda_N$ are the eigenvalues of I^s, and $v_{\max}, v_2, \ldots, v_N$ are the corresponding eigenvectors. The maximized total memory I_{tot} can be characterized by the largest eigenvalue $\lambda_{\max}$ of I^s, when the associated eigenvector is chosen to be the input mask as $M = v_{\max}$. The optimized input mask implies the preferred direction in the reservoir state space corresponding to the large principal component of I^s. In this case, the reservoir neurons can retain the most information of the past input signal.

In Figure 3.1, for two types of TDRs, we plot the consistency parameter of the optimized input masks compared to the case of random input masks, which has been illustrated in Figure 2.1. The red solid lines denote the consistency parameter of the input masks optimized by using the proposed method.

Figure 3.1a is according to the TDR based on the Mackey-Glass system. When $5 \leq \mathrm{SNR} \leq 50$, the consistency parameter obtained by the optimized input mask is much larger than by other random masks. This means that the ESP of the TDR is improved by the optimized input mask. Similar results are obtained for the three-layer deep TDR based on the coupled Ikeda delay systems, which are illustrated in Figure 3.1b. In this case, the consistency parameter of each layer is calculated respectively by using the optimized input mask. All of these results demonstrate that the input mask optimized by the proposed method significantly outperforms other random masks.



(a)



(b)

Figure 3.1: Consistency parameter as a function of signal to noise ratio (SNR) and input masks. (*a*) A TDR based on the Mackey-Glass system (see Chapter 4), for $\tau = 40$, $\eta = 0.7$, $\gamma = 0.1$ and $p = 1$, is used to calculate consistency parameter for 10 different input masks. (*b*) A three-layer deep TDR based on coupled Ikeda delay systems (see Chapter 5), with parameter setting as Table 5.1, is used to calculate consistency parameter for 10 different input masks. The black dashed lines and the black solid lines denote the consistency parameter of noise free reservoir systems with random input masks and noisy reservoir systems with random input masks. The red solid lines denote the consistency parameter of noisy reservoir systems with the optimized input masks.

3.5 Discussion

The main contribution of this work introduces how to measure the short-term memory capacity of the TDRs with Gaussian state noise via the Fisher memory curve. The form of the Fisher memory matrix for such TDRs is shown in Eq.3.9. This result can be used to optimize the input mask for improving the short-term memory performance in the presence of state noise. By defining the total memory as shown in Eq.3.11, the input mask obtained as the maximal principal component of the spatial Fisher memory matrix I^s maximizes the total memory of the TDRs. Indeed, the discussion in this chapter develops a method to optimize the input mask in a task-independent manner for improving the short-term memory performance in the presence of state noise.

In the following, the total memory always refers to the largest eigenvalue of I^s. The above description does not mean that the short-term memory capacity is changed by the input mask. Actually, the short-term memory capacity is immutable when the reservoir layer of TDRs is designed. What input masks affect is the short-term memory performance of the TDRs.

Chapter 4

Improving noise robustness of TDRs based on a Mackey-Glass oscillator

In this chapter, We apply the proposed method to the TDRs implemented by a single Mackey-Glass nonlinear oscillator [59].

4.1 Mackey-Glass oscillator

We consider a particular oscillator, introduced by Mackey and Glass [59] as a model of blood production. It is

$$\dot{x}(t) = -x(t) + \frac{\eta x(t-\tau)}{1 + x(t-\tau)^p}. \tag{4.1}$$

Here τ is the time delay, $\eta > 0$ is referred to as the feedback gain, and $p > 0$ controls the 'strength' of the nonlinearity.

The Eq.4.1 displays a wide variety of dynamical behavior including equilibrium, limit cycle, and chaos. It has been studied in great detail in [60]. In order to ensure reproducibility and reliability of the response signal, the used Mackey-Glass oscillator must operate in a certain regime such that the TDR has the ESP. Previous studies have shown that the optimal working regime of the TDRs is around a stable equilibrium for inputs small enough [7], [27], [58].

How the number of equilibria of Eq.4.1 depends on the parameters is summarized as follows [61]. In the case of $p = 1$, Eq.4.1 has two equilibria, which are $\bar{x} = 0$ and $\bar{x} = \eta - 1$. When $\eta \in (0, 1)$, the equilibrium $\bar{x} = 0$ is stable and the other $\bar{x} = \eta - 1$ is unstable. The transcritical bifurcation occurs if η crosses 1. In the case of $p \geq 2$, Eq.4.1 has more than two equilibria. Recalling that the TDR well for a broad class of input only if Eq.4.1 operates in the dynamical regime around a

unique stable equilibrium. When the number of the stable equilibria exceeds one, the transient response of the TDR near a stable equilibrium can be driven closer to the other due to strong input or loud noise. Thus the multistability makes the TDR's output depend not only on input history but on its initial condition, that is, it violates the ESP [5], which is commonly assumed in order to guarantee a TDR well-functioning.

The model of the TDR based on a single Mackey-Glass oscillator is

$$\dot{x}(t) = -x(t) + \frac{\eta[x(t-\tau) + \gamma J(t) + z(t)]}{1 + [x(t-\tau) + \gamma J(t) + z(t)]^p}. \tag{4.2}$$

Here

$$J(t) = M \times I(t), \quad I(t) = u(k) \quad for \quad \tau k \leq t < \tau(k+1) \tag{4.3}$$

is time-multiplexed input signal by the input mask M over the delay period τ, $z(t) \in \mathbb{R}$ is a white Gaussian state noise, and $\gamma > 0$ is the input gain. The randomness in the state noise of Eq.4.2 is involved in a stochastic TDR. In this case, the ESP affected by noise can be measured by the consistency parameter, which is introduced in section 1.4. In Figure 4.1, we plot how the consistency parameter evolves for the TDRs based on Maceky-Glass oscillator with random input masks as a function of the feedback gain η and the SNR level. This figure illustrates how the ESP is degraded by the state noise.

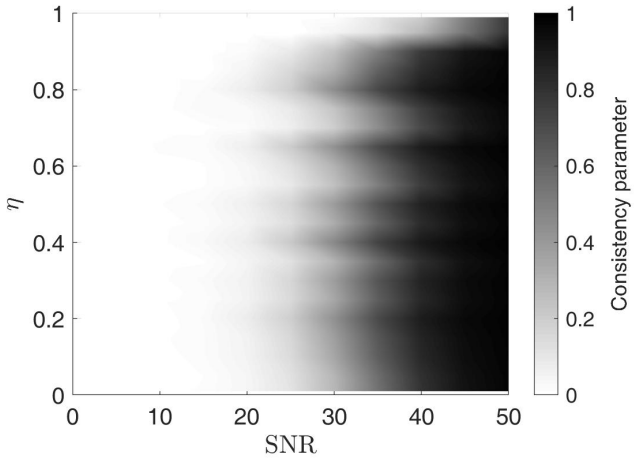


Figure 4.1: A heat map of the consistency parameter for the Mackey-Glass oscillator by TDRs with various feedback gains (vertical axis) and SNR levels (horizontal axis). The parameters are summarized as $\tau = 40$, $p = 1$, and $\theta = 0.2$, and a random input mask is used.

In this study we set $p = 1$, $\eta \in (0, 1)$. Moreover, even when the TDR in its autonomous form ($u(t) = z(t) = 0$) is monostable, the large input gain can break the ESP. Hence we set the input gain low enough to guarantee the ESP. In this numerical simulation, we set $\gamma = 0.1$.

4.2 Jacobian linearization and the connectivity matrix

The reservoir map $F : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$

$$\boldsymbol{x}(n) = F(\boldsymbol{x}(n-1), \boldsymbol{J}(n), \Delta\boldsymbol{z}(n)). \tag{4.4}$$

introduced in section 2.2 can be derived from the recursions Eq.2.5 as follows

$$\begin{cases} x_i(n) = e^{-\xi} x_{i-1}(n) + (1 - e^{-\xi}) \frac{\eta[x_i(n-1) + \gamma J_i(n) + \Delta z_i(n)]}{1 + [x_i(n-1) + \gamma J_i(n) + \Delta z_i(n)]^p} \\ x_0(n) = x_N(n-1) \\ \xi = \log(1 + \theta) \end{cases} . \tag{4.5}$$

Let $\bar{\boldsymbol{x}} := (\bar{x}, \bar{x}, \ldots, \bar{x})^\top \in \mathbb{R}^N$ be the stable fixed point around which F works. Then by performing the Taylor expansion of the right hand side and neglecting all higher (higher than the 1st) order terms, we obtain an expression of the Jacobian linearization:

$$\boldsymbol{x}(n) = \bar{\boldsymbol{x}} + A(\bar{\boldsymbol{x}})(\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}) + B(\bar{\boldsymbol{x}})\boldsymbol{J}(n) + C(\bar{\boldsymbol{x}})\Delta\boldsymbol{z}(n). \tag{4.6}$$

The connectivity matrix $A(\bar{\boldsymbol{x}})$ of the reservoir map at the fixed point $\bar{\boldsymbol{x}}$ has the following form

$$\begin{aligned} A(\bar{\boldsymbol{x}}) =& D_{\boldsymbol{x}} F(\bar{\boldsymbol{x}}, \boldsymbol{0}, \boldsymbol{0}) \\ =& \begin{bmatrix} \Phi_x & 0 & \cdots & 0 & e^{-\xi} \\ e^{-\xi}\Phi_x & \Phi_x & \cdots & 0 & e^{-2\xi} \\ e^{-2\xi}\Phi_x & e^{-\xi}\Phi_x & \cdots & 0 & e^{-3\xi} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ e^{-(N-1)\xi}\Phi_x & e^{-(N-2)\xi}\Phi_x & \cdots & e^{-\xi}\Phi_x & \Phi_x + e^{-N\xi} \end{bmatrix}, \end{aligned} \tag{4.7}$$

where $\Phi_x = (1 - e^{-\xi})\partial_x f(\bar{x}, 0, 0) = \eta(1 - e^{-\xi}) \frac{1 + (1-p)\bar{x}^p}{(1 + \bar{x}^p)^2}$. The matrix $B(\bar{\boldsymbol{x}})$ is given by

$$
\begin{aligned}
B(\bar{\boldsymbol{x}}) =& D_{\boldsymbol{J}} F(\bar{\boldsymbol{x}}, \mathbf{0}, \mathbf{0}) \\
=& \begin{bmatrix}
\Phi_J & 0 & \cdots & 0 & 0 \\
e^{-\xi}\Phi_J & \Phi_J & \cdots & 0 & 0 \\
e^{-2\xi}\Phi_J & e^{-\xi}\Phi_J & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
e^{-(N-1)\xi}\Phi_J & e^{-(N-2)\xi}\Phi_J & \cdots & e^{-\xi}\Phi_J & \Phi_J
\end{bmatrix},
\end{aligned} \tag{4.8}
$$

where $\Phi_J = (1 - e^{-\xi})\partial_J f(\bar{x}, 0, 0) = \eta\gamma(1 - e^{-\xi})\frac{1+(1-p)\bar{x}^p}{(1+\bar{x}^p)^2}$. And the matrix $C(\bar{\boldsymbol{x}})$ has the form as

$$
\begin{aligned}
C(\bar{\boldsymbol{x}}) =& D_{\Delta \boldsymbol{z}} F(\bar{\boldsymbol{x}}, \mathbf{0}, \mathbf{0}) \\
=& \begin{bmatrix}
\Phi_{\Delta z} & 0 & \cdots & 0 & 0 \\
e^{-\xi}\Phi_{\Delta z} & \Phi_{\Delta z} & \cdots & 0 & 0 \\
e^{-2\xi}\Phi_{\Delta z} & e^{-\xi}\Phi_{\Delta z} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
e^{-(N-1)\xi}\Phi_{\Delta z} & e^{-(N-2)\xi}\Phi_{\Delta z} & \cdots & e^{-\xi}\Phi_{\Delta z} & \Phi_{\Delta z}
\end{bmatrix}
\end{aligned} \tag{4.9}
$$

where $\Phi_{\Delta z} = (1 - e^{-\xi})\partial_{\Delta z} f(x_0, 0, 0) = \eta(1 - e^{-\xi})\frac{1+(1-p)\bar{x}^p}{(1+\bar{x}^p)^2}$.

4.3 Fisher memory curve at the equilibrium

Because the long-term behavior near a sable equilibrium does not depend significantly on the initial conditions, by using Eq.3.8, the reservoir state at time n is

$$
\boldsymbol{x}(n) = \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} [A(\bar{\boldsymbol{x}})^k B(\bar{\boldsymbol{x}}) M u((n-k)\tau) + A(\bar{\boldsymbol{x}})^k C(\bar{\boldsymbol{x}}) \Delta \boldsymbol{z}(n-k)]. \tag{4.10}
$$

Recalling that the increment $\Delta \boldsymbol{z}(n)$ follows the Gaussian distribution, the conditional distribution $p(\boldsymbol{x}(n)|\boldsymbol{u})$ is also Gaussian with mean $\mu = \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} A(\bar{\boldsymbol{x}})^k B(\bar{\boldsymbol{x}}) M u((n-k)\tau)$ and covariance matrix $\boldsymbol{C}_n = \epsilon\theta \sum_{k=0}^{\infty} A(\bar{\boldsymbol{x}})^k C(\bar{\boldsymbol{x}}) C(\bar{\boldsymbol{x}})^\top A(\bar{\boldsymbol{x}})^{k\top}$. By using Eq.3.5, the Fisher memory curve $\{I_k\}_{k=0}^{\infty}$ at the equilibrium $(\bar{\boldsymbol{x}}, \mathbf{0}, \mathbf{0})$ can be described as

$$
I_k = M^\top B(\bar{\boldsymbol{x}})^\top A(\bar{\boldsymbol{x}})^{k\top} \boldsymbol{C}_n^{-1} A(\bar{\boldsymbol{x}})^k B(\bar{\boldsymbol{x}}) M. \tag{4.11}
$$

Under the condition of $p = 1$, $\eta \in (0, 1)$, the TDR operates in the dynamical regime around the equilibrium $\bar{x} = 0$. In Figure 3.9, we plot the Fisher memory curve for the TDR as a function of the chosen feedback gains η. The input mask

is always taken as the eigenvector corresponding to the largest eigenvalue of I^s such that the short-term memory capacity is maximized in the presence of state noise. The figure shows that the process of memory decay or forgetting in the presence of state noise becomes slow when the feedback gain is close to one. The results in Figure 3.9 support the notion that the short-term memory capacity of a reservoir system can be maximized at the edge of chaos [43], [62]. But if the feedback gain crosses one, the equilibrium $\bar{x} = 0$ is unstable. In this case, the TDR can not guarantee the ESP in the working regime around the unstable equilibrium $\bar{x} = 0$. In conclusion, while high-level feedback is available for improving the strength of the memory trace in the presence of state noise, it can break the ESP under specific scenarios.

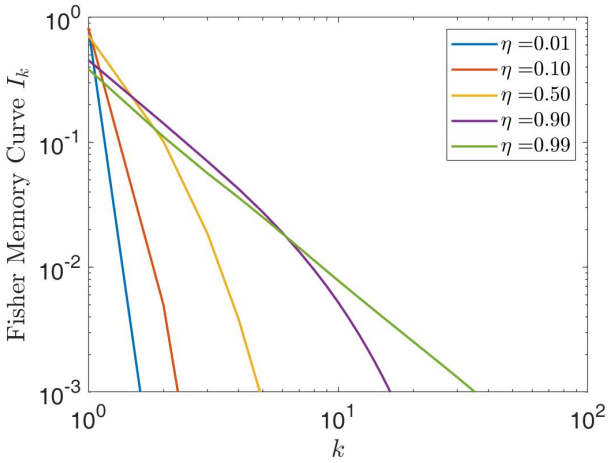


Figure 4.2: Fisher memory curve exhibited by the TDRs based on the Mackey-Glass system with parameter settings: $\tau = 80$, $\gamma = 0.1$ **and** $p = 1$, **for different values of the feedback gain** η.

4.4 Optimized Input mask in TDRs with state noise

As discussed in section 3.4, the total capacity of the short-term memory is measured by summing the Fisher memory curve over all delay as

$$I_{\text{tot}} = \sum_{k=0}^{\infty} I_k = M^\top I^s M, \tag{4.12}$$

where

$$I^s = \sum_{k=0}^{\infty} B(\bar{\boldsymbol{x}})^\top A(\bar{\boldsymbol{x}})^{k\top} \boldsymbol{C}_n^{-1} A(\bar{\boldsymbol{x}})^k B(\bar{\boldsymbol{x}}). \tag{4.13}$$

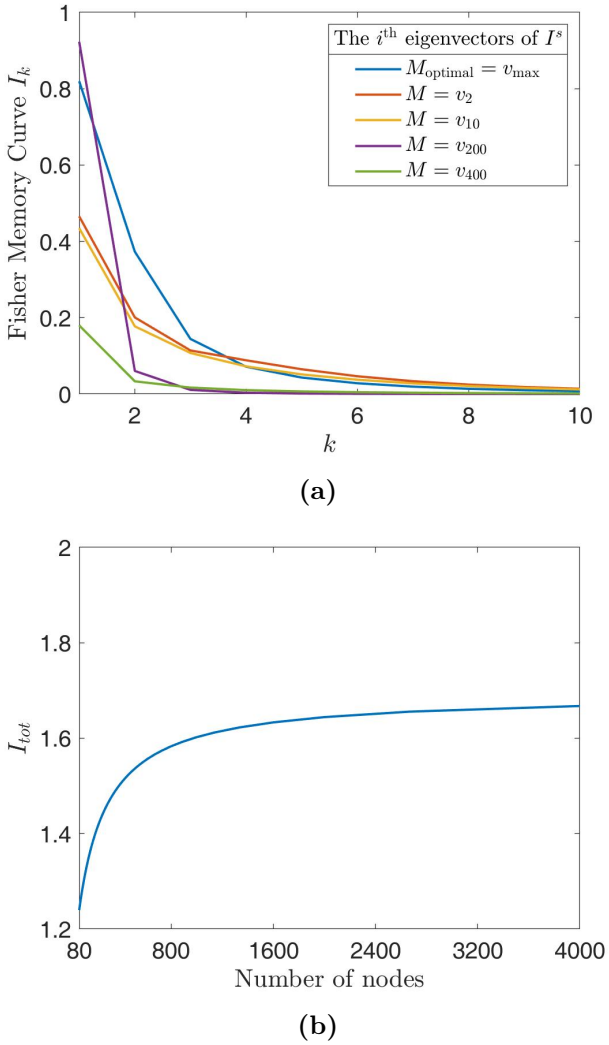


Figure 4.3: Short-term memory capacity exhibited by the TDRs based on the Mackey-Glass system with parameter settings: $\tau = 80$, $\eta = 0.9$, $\gamma = 0.1$, **and** $p = 1$. (a) The input masks are corresponding to the eigenvectors associated with the eigenvalues of I^s. (b) The total memory for the TDRs based on the Mackey-Glass System as a function of the number of neurons, with the sampling length $\theta \in [0.02, 1]$.

We define the eigenvalues of I^s as $\lambda_{\max} \geq \lambda_2 \geq \ldots \geq \lambda_N$, which are arranged in descending order, and the corresponding eigenvectors as $v_{\max}, v_2, \ldots, v_N$. I_{tot} can be maximized by the largest eigenvalue $\lambda_{\max}$ of I^s, when the associated eigenvector is chosen to be the input mask as $M = v_{\max}$.

In Figure 4.3a, we plot the Fisher memory curves obtained by using the chosen eigenvectors as the input masks. The blue solid line represents the case of optimized

input mask $M = v_{\max}$. The results in Figure 4.3a imply that the Fisher memory curve can be seen as a quantitative measurement to observe the ESP (fading memory).

In Figure 4.3b, we plot how the total memory I_{tot} evolves for the TDR as a function of the number of reservoir neurons. The figure illustrates that increasing the number of reservoir neurons can prevent the memory from decaying in the presence of state noise. But there is a limit to the total memory. Because in TDRs the virtual neurons are corresponding to the points on the delay line placed equidistantly. As the number of the virtual neurons increases, the adjacent virtual neurons are too close to be independent; namely, the increase in the effective degrees of freedom is limited.

4.5 Comparison with the conventional method

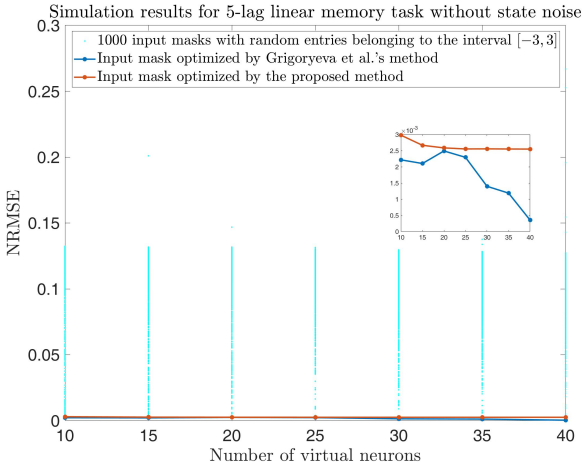
Table 4.1: CPU-time performance of the Grigoryeva et al.'s method and the proposed method.

Number of virtual neurons	10	15	20	25	30	35	40
Grigoryeva et al.'s method	11.75	29.58	47.66	82.30	117.16	233.16	823.55
Proposed method	0.22	0.25	0.31	0.48	0.64	4.41	5.23

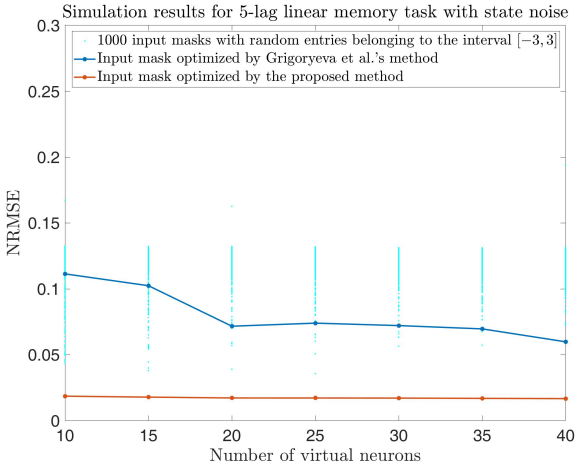
In [58], the input mask for the memory tasks is determined by solving structured optimization problems, which have been successfully used in linear and quadratic memory tasks. In these tasks, a linear or quadratic function is used to generate a one-dimensional target signal from the time-lagged input signal. However, the reservoir performance can be degraded by state noise originating from analog components of the reservoir layer [9], [10], [23]. In the input layer, time-multiplexing is used to inject the input signal into temporally separated virtual neurons, then the dimension of the input mask is determined by the size of the reservoir layer. For a large set of the virtual neurons used to improve the reservoir performance, Grigoryeva et al.'s method gives rise to a high-dimensional nonlinear optimization problem of high computational cost. On the other hand, our method only involves a linear problem, i.e., finding the maximal principal component of the spatial Fisher memory matrix.

Considering the 5-lag linear memory task, we compare the time performance of Grigoryeva et al.'s method with the proposed method in Table 4.1. Here, we solve the optimization problem obtained from Grigoryeva et al.'s method by using the particle swarm optimization (PSO) algorithm [63]. The computational time is measured so that the performance of the two methods is aligned. That is, the PSO algorithm is stopped when the resulting mask provides the same performance as that of ours. The

CPU times are measured in seconds on an Intel Core i7, 3.40 GHz PC (8 GB RAM). The results in Table 4.1 show that the proposed method is time efficient.



(a)



(b)

Figure 4.4: Reservoir performance for the 5-lag linear memory task with parameter settings: $p = 1$, $\eta = 0.5$, $\gamma = 0.1$, $\theta = 0.2$ **and** SNR $= 40$. (a) NRMSEs for the 5-lag linear memory task without state noise as a function of the number of virtual neurons. (b) NRMSEs for the 5-lag linear memory task with state noise as a function of the number of virtual neurons. The cyan points indicate the NRMSEs obtained from the 1000 different randomly picked masks. The blue lines link the points that indicate the NRMSEs simulated by using the input mask optimized by [58]. The red lines link the points that indicate the NRMSEs simulated by using the input mask optimized by the proposed method.

In Figure 4.4, the reservoir performances optimized by using the Grigoryeva et al.'s method and the proposed method are compared in the 5-lag linear memory task. The reservoir performance is expressed by the normalized root mean square error (NRMSE) (see Appendix A). In this comparison, the termination condition for the PSO algorithm is set as if the relative change of the objective function over the last 20 iterations is less than 10^{-6}. The simulation results of the TDR in the absence of state noise are shown in Figure 4.4a. In this case, Grigoryeva et al.'s method outperforms the proposed method, as in the partial enlarged view of Figure 4.4a. This is somewhat natural since Grigoryeva et al.'s method is a task-specific optimization while our approach is task-independent. Figure 4.4b illustrates the impact of the state noise on the memory tasks. In these simulation experiments, the signal-to-noise ratio of SNR = 40 is obtained using the given state noise strength ϵ. These results imply that the proposed method is helpful for improving the noise robustness of TDRs.

4.6 Simulation experiments

In this section, the proposed method is validated by solving two types of benchmark tasks. Each type of benchmark task requires different crucial properties to make a good approximation of the target function. Some tasks require the short-term memory capacity for good performance, and others are strongly dependent on the nonlinear transformation. Previous studies have shown that these two properties are often in conflict with each other [64], [65]. It is called memory-nonlinearity trade-off. As a by-product of our proposed method, input masks of various memory capacity, which is quantitatively characterized by eigenvalues of the spatial Fisher memory matrix, are obtained. Using these masks, the memory-nonlinearity trade-off of the TDRs in terms of the masks can be studied in detail.

In the following, we use the eigenvectors $v_{\max}, v_2, \ldots, v_N$ of the spatial Fisher memory matrix I^s as the input masks in the simulation experiments. These eigenvectors are arranged in descending order according to the eigenvalues of I^s as $\lambda_{\max} \geq \lambda_2 \geq \ldots \geq \lambda_N$. The short-term memory capacity can be maximized in the presence of state noise, when the input mask $M = v_{\max}$.

4.6.1 Weakly nonlinear function approximation task

We use the eigenvectors of I^s as the input mask and compare their test errors for a memory task. We consider an i.i.d. random signal $u(t)$ from a uniform distribution over $[-1, 1]$. The TDRs are trained to produce an output $w(u) = \sin(u(t - \tau'))$ driven by the input signal of τ' step before as closely as possible. For the task of $\sin(u(t - \tau'))$, the memory performance is necessary but little performance of the

nonlinear transformation is required. It allows us to focus on the studying of the short-term memory capacity with the case of the different input masks. The task parameter τ' controls the requirement of memory. In our experiments, the TDRs are implemented by the Mackey-Glass node with the parameter setting: $\tau = 80$. The separation distance of the virtual neurons is set at $\theta = 0.2$ that offers a suitable short-term memory capacity as in Figure 4.3b. Then the number of virtual neurons is $N = 400$. The chosen state noise strength ϵ results in a signal-to-noise ratio of SNR = 40. The readout weights were trained by an online algorithm (see section 1.3.2 above).

Concretely, we let each TDR run for 5000 steps, starting from a random initial state, and discard the initial 100 steps during which the effect of the initial state dies out.

In Figure 4.5 and Figure 4.6, we show the reservoir performance according to the normalized root mean square error (NRMSE) for the memory task $\sin(u(t-\tau'))$. In these simulation experiments, the best performance according to the minimal NRMSE is always obtained by using the input mask $M = v_{\max}$. In Figure 4.5a and Figure 4.6a, we plot the test error surface in the memory task $\sin(u(t-\tau'))$, as a function of the feedback gain and the input mask. Unsurprisingly a decrease in the NRMSE corresponds to an increase in the feedback gain η for every input mask. These results support the notion that the high-level feedback can enhance the strength of memory trace in the presence of state noise.

Specially, we plot how the NRMSEs evolve for $\eta = 0.1, 0.5, 0.9$ as a function of the chosen input masks in Figure 4.5b and Figure 4.6b from top to bottom. We also confirmed that similar results are obtained by employing other tasks, such as $f(u) = u(t-\tau')$ or $f(u) = u^2(t-\tau')$. We note that the masks with indices not less than 27 seem not approximate eigenvectors of the exact spatial Fisher memory matrix but numerical artifacts (see Appendix B). Thus the “local minima" of the NRMSEs around the 200^{th} indices for $\eta = 0.9$ seen in Figure 4.5b and Figure 4.6b might be numerical phenomena which have no relevance to the dynamical properties of the TDR.

These results confirm that the input mask in direction of the maximal principal component of the spatial Fisher memory matrix optimizes the short-term memory capacity in the presence of the white Gaussian state noise (see section 3.4 above).

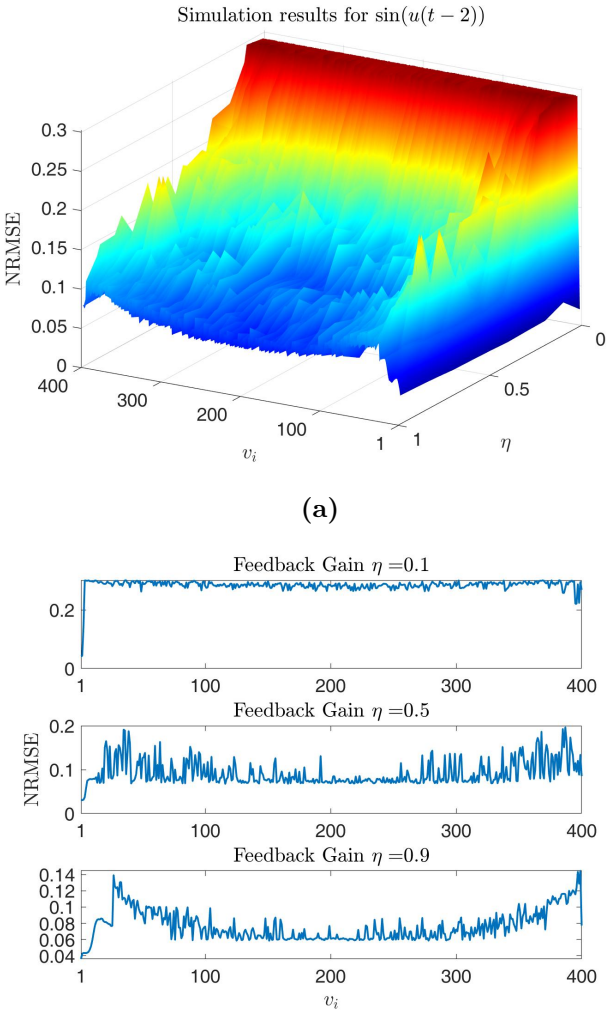


Figure 4.5: Simulation results for the learning task of $\sin(u(t-2))$. (a) Error surface for $\sin(u(t-2))$, as a function of the input mask and the feedback gain. (b) The NRMSE versus the input masks for $\sin(u(t-2))$ with parameter setting: $\eta = 0.1, 0.5, 0.9$ from top to bottom. The indices of the input masks are arranged in descending order according to the eigenvalues of I^s.

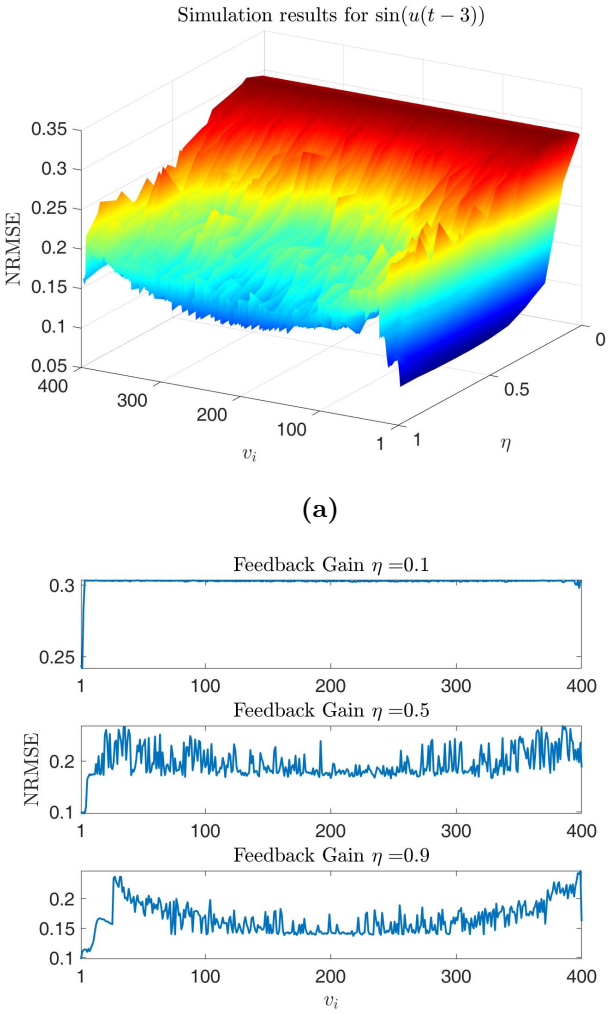


Figure 4.6: Simulation results for the learning task of $\sin(u(t-3))$. (a) Error surface for $\sin(u(t-3))$, as a function of the input masks and the feedback gain. (b) The NRMSE versus the input masks for $\sin(u(t-3))$ with parameter setting: $\eta = 0.1, 0.5, 0.9$ from top to bottom. The indices of the input masks are arranged in descending order according to the eigenvalues of I^s.

4.6.2 Chaotic attractor learning task

The chaotic attractor learning task is one of the most widely used benchmarks in reservoir computing. It is often used to verify the performance of nonlinear transformation, which is also an essential property for the TDRs. The input signal is transformed in various nonlinear ways via the high-dimensional nonlinear dynamics of the reservoirs. Here we test the TDRs on the chaotic attractor learning tasks of two dynamical systems: the Hénon map and the logistic map. For these learning tasks, the performance of high-level nonlinear transformation is necessary.

Hénon map

The training data set of the Hénon map is generated by

$$\begin{aligned} x(t+1) &= 1 - ax(t)^2 + by(t), \\ y(t+1) &= x(t-1), \end{aligned} \tag{4.14}$$

with parameter setting: $a = 1.4$ and $b = 0.3$. For this parameter value the Hénon map is chaotic.

Logistic map

The logistic map is described by the equation

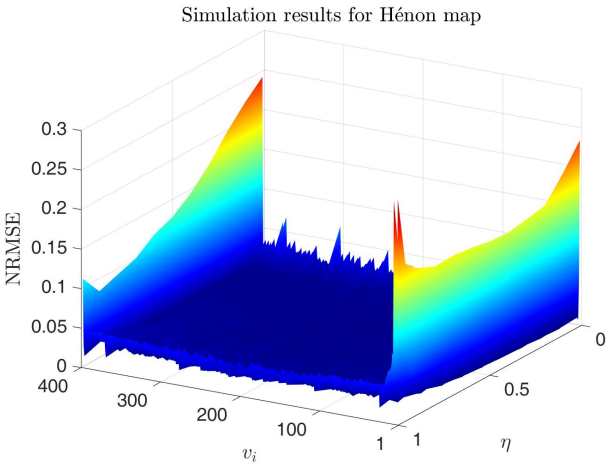
$$x(t+1) = rx(t)(1 - x(t)). \tag{4.15}$$

The time series $x(t)$ is chaotic with parameter setting: $r = 4$.

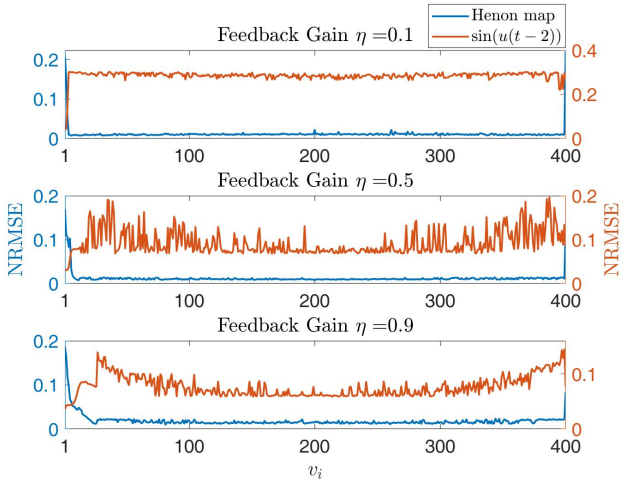
We choose $x(t)$ as the input signal, and the TDRs are trained to produce an output of $x(t+1)$. The TDRs are trained from a run of 5000 steps and the first 100 steps were discarded. In these simulation experiments, the signal-to-noise ratio of SNR = 70 is obtained using the given state noise strength ϵ.

In Figure 4.7a and Figure 4.8a, we plot the test error surface in the learning tasks of the Hénon map and the logistic map, respectively, as a function of the feedback gain and the input mask. These figures show that a broad range of the input mask can provide a good performance of the nonlinear transformation. It is interesting to note that the performance is degraded when the input mask takes the eigenvector corresponding to the largest eigenvalue of I^s.

Specially, we compare the test errors for the tasks of the Hénon map and the logistic map with the test errors for the tasks $w(u) = \sin(u(t - \tau'))$ in different cases of $\eta = 0.1, 0.5, 0.9$, as in Figure 4.7b and Figure 4.8b. The test errors show opposite trends with respect to the input mask. This phenomenon can be interpreted as the memory-nonlinearity trade-off [64], [65] in the TDRs.

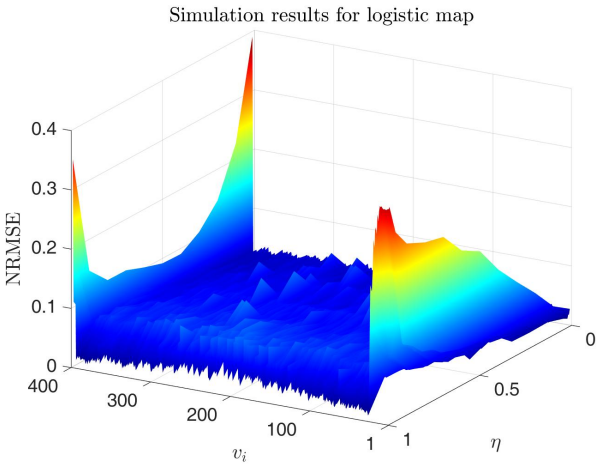


(a)

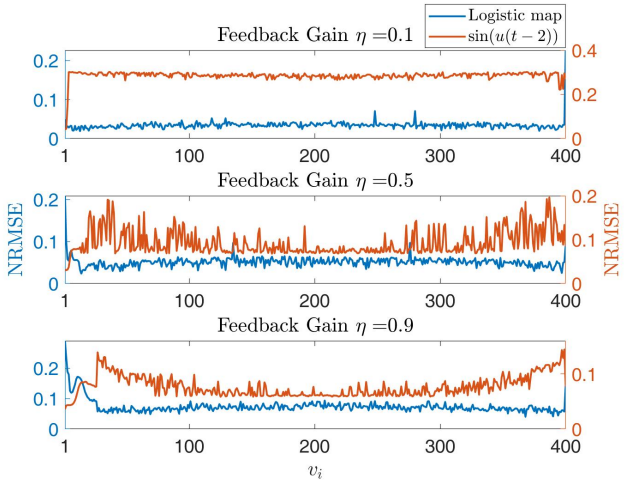


(b)

Figure 4.7: Simulation results for Hénon map. (a) Error surface for Hénon map, as a function of the input masks and the feedback gain. (b) The NRMSE versus the input masks with parameter setting: $\eta = 0.1, 0.5, 0.9$ from top to bottom. The blue lines denote the NRMSE for the learning task of Hénon map. The red lines denote the NRMSE for the learning task of $\sin(u(t-2))$. The indices of the input masks are arranged in descending order according to the eigenvalues of I^s.



(a)



(b)

Figure 4.8: Simulation results for logistic map. (a) Error surface for logistic map, as a function of the input masks and the feedback gain. (b) The NRMSE versus the input masks with parameter setting: $\eta = 0.1, 0.5, 0.9$ from top to bottom. The blue lines denote the NRMSE for the learning task of logistic map. The red lines denote the NRMSE for the learning task of $\sin(u(t-2))$. The indices of the input masks are arranged in descending order according to the eigenvalues of I^s.

4.7 Discussion

Recalling that the consistency parameter is a practical tool for characterizing the ESP of the stochastic TDRs (see section 1.4 above). At the end of this chapter, the consistency parameter obtained by the input masks optimized by the proposed method is compared to the consistency parameter obtained by random input masks. (The corresponding heat map of the consistency parameter has been plotted in Figure 4.1) when scanning the feedback gain η and SNR level. These results are illustrated in Figure 4.9. The black dashed lines denote the consistency parameter obtained by random input masks. The black solid lines denote the consistency parameter obtained by optimized input masks. These results show that our proposed method improves the echo state property in the presence of state noise while also facilitating the design of the input masks for the TDRs based on a single Mackey-Glass oscillator with desired computational properties.

In conclusion, we applied the proposed method (see chapter 3 above) to the TDRs based on a single Mackey-Glass oscillator in the presence of a white Gaussian state noise. In particular, the availability of the Fisher memory curve of Eq.4.11 is helpful to understand how an input signal propagates and fades in the Mackey-Glass oscillator. More importantly, it provides a task-independent and computationally-efficient method to optimize the input mask such that the memory performance of the TDRs impaired by state noise can be improved. In comparison with the existing optimization method, the proposed one is running-time-efficient since it only involves a linear problem in the optimization procedure and improves memory performance in a great extent in the presence of state noise. Through numerical experiments, we confirmed that the input mask in direction of the maximal principal component of the spatial Fisher memory matrix certainly provides the best performance on benchmark memory tasks requiring only weak nonlinear transformation capacity. Also, we illustrated a memory-nonlinearity trade-off in terms of the input masks, whose associated memory performance can be quantitatively characterized in detail through spectral properties of the spatial Fisher memory matrix, using chaotic attractor learning tasks. In the next chapter, we focus on another type of TDRs based on coupled Ikeda time-delay systems. The proposed method is applied to improve the noise robustness of such TDRs.

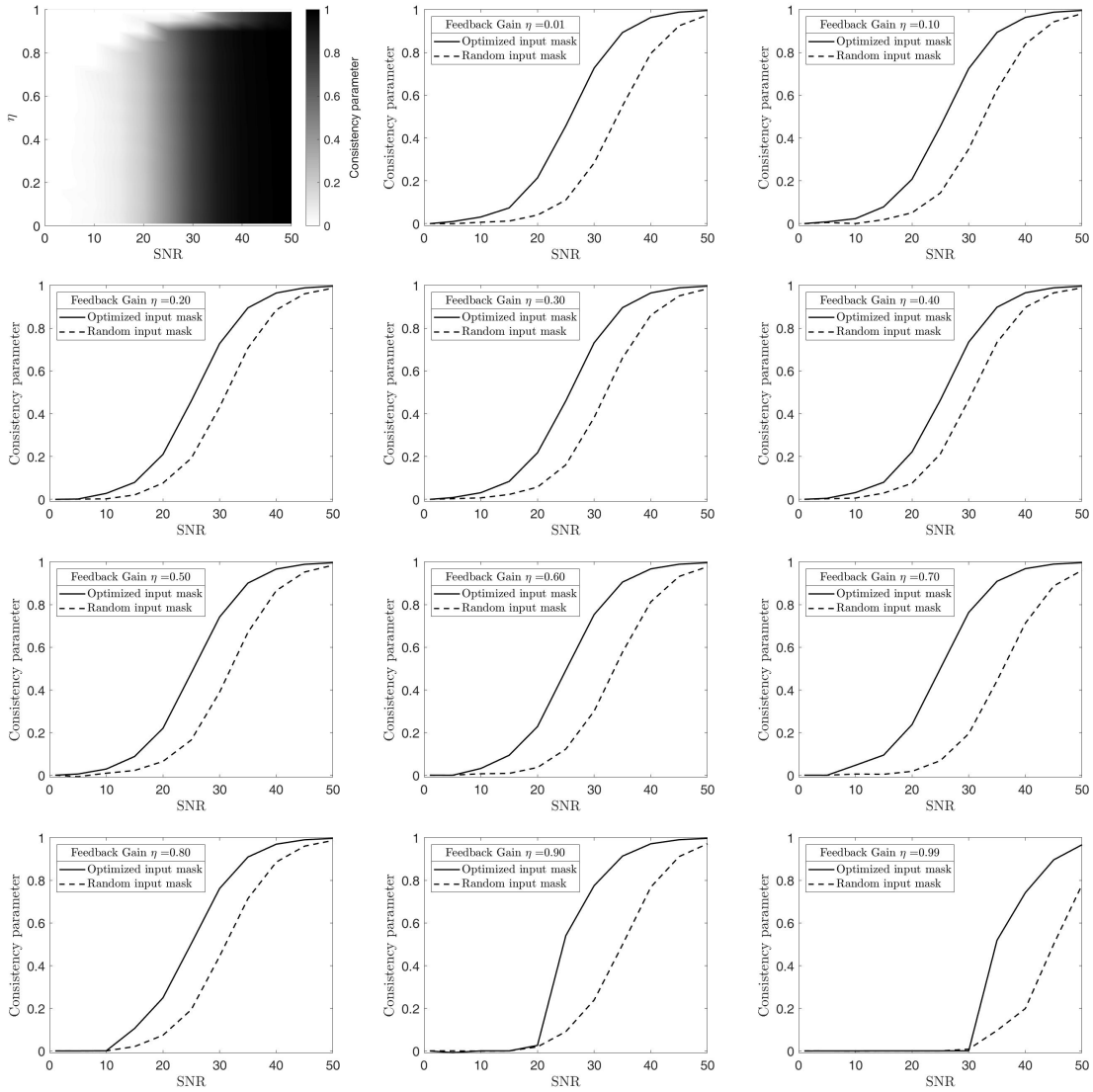


Figure 4.9: Consistency parameter for the Mackey-Glass oscillator by TDRs with various feedback gains η and SNR levels. The black dashed lines denote the consistency parameter obtained by random input masks. The black solid lines denote the consistency parameter obtained by optimized input masks.

Chapter 5

Improving noise robustness of deep TDRs based on coupled Ikeda delay systems

In this chapter, we focus on the recently introduced opto-electronic reservoir based on coupled Ikeda time-delay systems [12]–[14]. The proposed method is applied to optimize the performance of the deep TDRs, and validated by carrying out some simulation experiments.

5.1 Coupled Ikeda delay systems

We focus our attention towards the simple and well characterized Ikeda time-delay system, which is first inroduced by Kensuke Ikeda as a model of a passive optical resonator system. L Ikeda time-delay systems are coupled unidirectionally as follows

$$\begin{cases} \dot{x}_l(t) = -x_l(t) - \delta_l y_l(t) + \beta_l \sin^2[x_l(t-\tau_l) + \kappa_l x_{l-1}(t) + b_l] \\ \dot{y}_l(t) = x_l(t) \end{cases}, l = 1, 2, \ldots, L \quad (5.1)$$

where τ_l is the time delay, δ_l is the damping constant satisfying $\delta_l \leq \delta_{l+1}$, β_l is the feedback gain, and b_l is a scalar phase shift of the nonlinearity. The time evolution of the dynamical variable $\boldsymbol{v}_l = (x_l, y_l)^\top$ is characterized as the nonlinear feedback $\sin^2$ which concludes the past state, which leads to the occurrence of time delay, and the dynamical state of the previous layer with a scalar phase shift. Then $\kappa_1 = 0$; and when $l \geq 2$, κ_l is the coupling strength between the layer $l-1$ and l.

The complex dynamics in Eq.5.1 depend upon the feedback gain β_l. It has been expounded in great detail in [13], [14], [66], [67]. As is well known, as the feedback

gain β_l increases the stationary solution of Eq.5.1 becomes unstable and occurs period doubling bifurcation. When $\beta_l \gg 1$, the trajectory $\boldsymbol{v_l}$ shows chaotic behavior. Recalling that for TDRs based on a single nonlinear node with delayed feedback, both theoretical and empirical studies have been shown that its ESP can be guaranteed by the asymptotic stability of the equilibrium where the TDRs operates. In this chapter, we focus on the deep TDRs comprising a hierarchy of multiple reservoir layers with delayed feedback. The fundamental conditions for the ESP of such systems have been established in [36]. It shows that the asymptotic stability of Eq.5.1 is relevant to the ESP.

Considering L coupled Ikeda time-delay systems with parameter setting: if $l = 1$, $\beta_1 \in (0, 1]$, $\delta_1 = 0$ and $\kappa_1 = 0$; if $l \geq 2$, $\beta_l \in (0, 1]$, $\delta_l > 0$ and $\kappa_l > 0$. Then the equilibrium of each layer of Eq.5.1 is given as the solution of the following equations: if $l = 1$,

$$\bar{x}_1 = \beta_1 \sin^2(\bar{x}_1 + b_1); \tag{5.2}$$

and if $l \geq 2$,

$$\begin{aligned} \bar{x}_l =& 0, \\ \bar{y}_l =& \frac{\beta_l}{\delta_l} \sin^2(\kappa_l \bar{x}_{l-1} + b_l). \end{aligned} \tag{5.3}$$

The stability of the equilibrium $\boldsymbol{v}_l = (\bar{x}_l, \bar{y}_l)^\top$ can be studied using the characteristic equation derived from the linear approximation at this point. This equilibrium is asymptotically stable under the condition that the real part of every solution of the characteristic equation is negative [14].

5.2 Model of deep TDRs with state noise

By applying the TDR scheme to coupled Ikeda time-delay systems, we establish the model of deep time-delay reservoir (Deep TDR) computer as illustrated in Figure 5.1. The time evolution of dynamical variable of the layer l is given by

$$\begin{cases} \dot{x}_l(t) = -x_l(t) - \delta_l y_l(t) + \beta_l \sin^2[x_l(t - \tau_l) + \kappa_l J_l(t) + z_l(t) + b_l] \\ \dot{y}_l(t) = x_l(t) \end{cases} \tag{5.4}$$

where

$$J_l(t) = \begin{cases} M \times I(t) & \text{for} \quad l = 1, \\ x_{l-1}(t) & \text{for} \quad l \geq 2. \end{cases} \tag{5.5}$$

is the input signal for each layer. Only the first layer is fed by the task-specific input sequence $I(t)$, which is obtained by the sample and hold operation mixed with an additional input mask as Eq.1.5. In this case, κ_1 denotes the input gain. When $l \geq 2$,

each consecutive layer receives only the first component $x_{l-1}(t)$ of the dynamical state $\boldsymbol{v}_{l-1}(t)$ of the previous layer with coupling strength κ_l. $z_l(t)$ is a white Gaussian state noise.

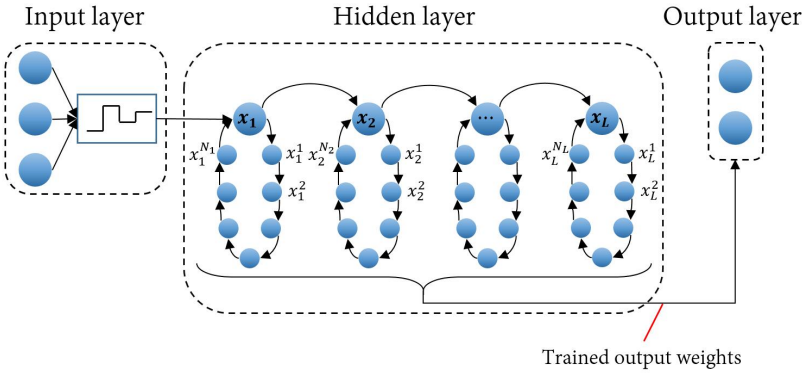


Figure 5.1: Schematic of deep time-delay reservoirs.

Generally, these coupled nonlinear oscillators communicate with each other under instantaneous interaction, namely, without time delay [13]. In the case of different time delay in each layer, a clock cycle T given by the input speed is applied to each layer such that inter-layer coupling is instantaneous; see more details in [14], [68]. It is a generalization of the frequently used TDRs in contrast to the case of $T = \tau$. In this chapter, we consider that the time delay τ_l in each layer has the same length as $\tau_l = \tau$, $l = 1, 2, \ldots, L$, and the clock cycle T is set to τ. In such case, each layer has the same number of virtual neurons $N_l = N$, $l = 1, 2, \ldots, L$ which is parameterized by a sampling length θ by $N = \tau/\theta$. As the interconnection between these coupled layer is realized via the first component $x_l(t)$ of the dynamical state $\boldsymbol{v}_l(t)$, the virtual neurons

$$\boldsymbol{x}_l(n) = (x_l^1(n), x_l^2(n), \ldots, x_l^{N_l}(n))^\top \in \mathbb{R}^{N_l} \tag{5.6}$$

in layer l is defined as

$$\begin{aligned} x_l^i(n) =& x_l(nT - (N_l - i)\theta) \\ =& x_l(n\tau - (N - i)\theta) \quad for \quad i = 1, 2, \ldots, N. \end{aligned} \tag{5.7}$$

Here N is the number of virtual neurons in layer l, and θ is subjected to the time-multiplexing method in Eq.1.5.

In the output layer of the deep TDR, all virtual nodes $x_l(n\tau - (N - i)\theta)$ of each layer $l = 1, 2, \ldots, L$ are readout by combining a trained output matrix W_{out} linearly.

Assuming that the collection of the virtual neurons is given by

$$\boldsymbol{x}(n) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_{\mathrm{tot}}} \end{bmatrix} = \begin{bmatrix} x_1(n\tau) \\ x_1(n\tau+\theta) \\ \vdots \\ x_L(n\tau-(N-2)\theta) \\ x_L(n\tau-(N-1)\theta) \end{bmatrix} \tag{5.8}$$

where $N_{\mathrm{tot}} = LN$ is the total reservoir size. Then the output is computed according to

$$\boldsymbol{y}(n) = W_{\mathrm{out}}\boldsymbol{x}(n). \tag{5.9}$$

In Figure 5.2, we show how the consistency parameter of the two-layer deep TDRs is impaired by noise. In Figure 5.2a, the feedback gain β_1 of the 1st layer is chosen as the dependent variable belongs on the vertical axis and $\beta_2 = 0.9$ is fixed. The opposite situation that the feedback gain β_2 of the 2nd layer is chosen as the dependent variable belongs on the vertical axis and $\beta_1 = 0.6$ is illustrated in Figure 5.2b. These figures illustrate how the ESP of the deep TDR is degraded by noise. We note that for each parameter combination the normalized random input masks are used in these simulation experiments. For the simulation results of the consistency parameter for different values of the coupling strength κ_l, we refer to Appendix C.

5.3 Reservoir map of deep TDRs

In order to study the memory trace in the multi-layered reservoir network, we consider the collection of the dynamical state $(x_l, y_l)^\top$ as a total system, and these infinite time series obtained by Eq.5.4 for layer l are split into the delay interval τ. We define

$$\begin{aligned} \boldsymbol{x}(n) &= (x_1^1(n), x_1^2(n), \ldots, x_l^i(n), \ldots, x_L^{N-1}(n), x_L^N(n))^\top, \\ \boldsymbol{y}(n) &= (y_1^1(n), y_1^2(n), \ldots, y_l^i(n), \ldots, y_L^{N-1}(n), y_L^N(n))^\top, \end{aligned} \tag{5.10}$$

with $n \in \mathbb{Z}$. Here the i^{th} virtual neuron state of the l^{th} layer at the n^{th} time step is denoted as $x_l^i(n) = x_l(n\tau - (N-i)\theta)$, and the corresponding dynamical state is $y_l^i(n) = y_l(n\tau - (N-i)\theta)$. With the discrete time index n, the set of segments can be understood as the representation of the deep TDR. Considering the Euler–Maruyama

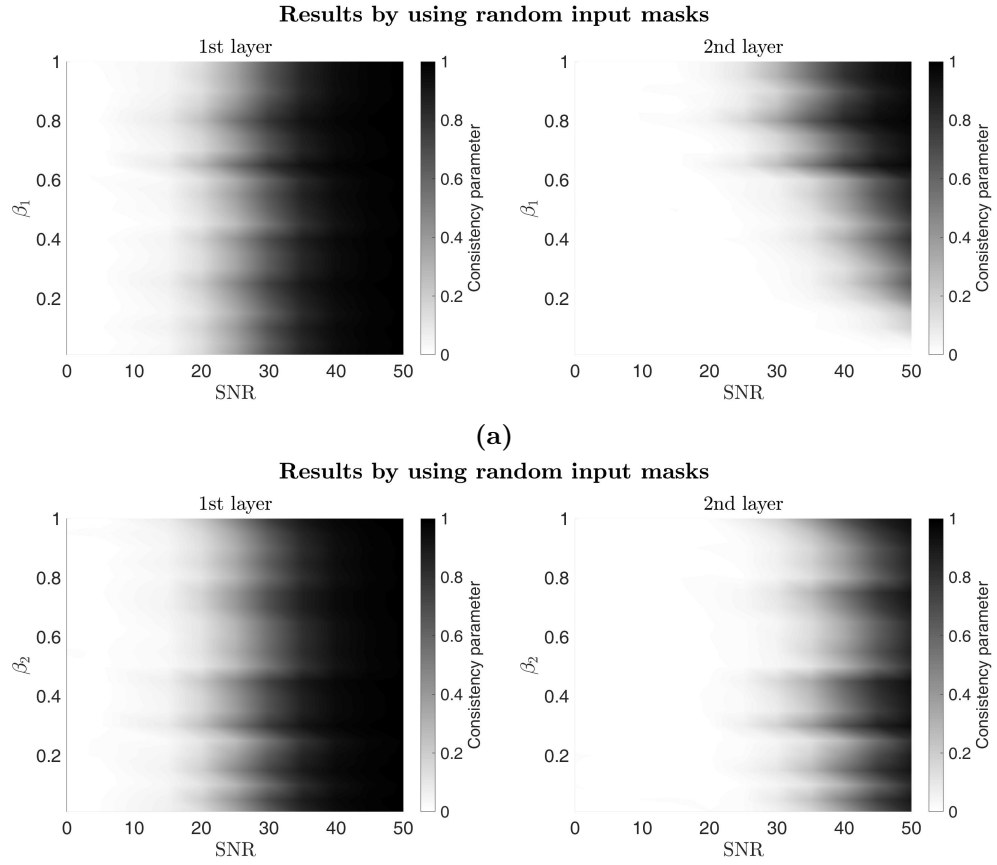


Figure 5.2: Consistency parameter for random input masks, as a function of the feedback gain β_l (vertical axis) and SNR level (horizontal axis). (a) $\beta_2 = 0.9$. (b) $\beta_1 = 0.6$. The type of two-layer deep TDRs based on coupled Ikeda delay system is considered with parameter setting: $\tau_1 = \tau_2 = 20$, $\delta_1 = 0$, $\delta_2 = 0.01$, $\kappa_1 = 0.4$, $\kappa_2 = 1$, $b_1 = b_2 = 0.2$ and $\theta = 0.2$.

time-discretization with integration step θ, Eq.5.10 has the specific form as follows

$$\begin{cases} x_l^i(n) = & \frac{1}{1+\theta+\delta_l\theta^2}x_l^{i-1}(n) - \frac{\delta_l\theta}{1+\theta+\delta_l\theta^2}y_l^{i-1}(n) \\ & + \frac{\beta_l\theta}{1+\theta+\delta_l\theta^2}f(x_l^i(n-1), J_l^i(n), \Delta z_l^i(n)), \\ y_l^i(n) = & \frac{\theta}{1+\theta+\delta_l\theta^2}x_l^{i-1}(n) + \frac{1+\theta}{1+\theta+\delta_l\theta^2}y_l^{i-1}(n) \\ & + \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}f(x_l^i(n-1), J_l^i(n), \Delta z_l^i(n)), \\ x_l^0(n) = & x_l^N(n-1), \\ y_l^0(n) = & y_l^N(n-1), \end{cases} \tag{5.11}$$

where $f(x_l^i(n-1), J_l^i(n), \Delta z_l^i(n)) = \sin^2(x_l^i(n-1) + \kappa_l J_l^i(n) + b_l + \Delta z_l^i(n))$. The layer dependent input is given by

$$J_l^i(n) = \begin{cases} J_1(n\tau - (N-i)\theta) & \text{for} \quad l = 1, \\ x_{l-1}^i(n) & \text{for} \quad l \geq 2. \end{cases} \tag{5.12}$$

And the increment $\Delta z_l^i(n) = z_l(n\tau - (N-i)\theta) - z_l(n\tau - (N-i+1)\theta)$ follows the normal distribution with zero mean and covariance $\langle \Delta z_l^i(k) \Delta z_l^j(l) \rangle = \epsilon\theta\delta_{k,l}\delta_{i,j}$. Recalling that only the first reservoir layer receives the external input signal which is denoted as $\boldsymbol{J}_1(n)$.

The recursions Eq.5.11 determine the mapping of one τ-segment to the next, which has an expression of the form

$$\begin{cases} \boldsymbol{x}(n) = & \boldsymbol{F}(\boldsymbol{x}(n-1), \boldsymbol{y}(n-1), \boldsymbol{J}_{\text{in}}(n), \Delta\boldsymbol{z}(n)) \\ \boldsymbol{y}(n) = & \boldsymbol{G}(\boldsymbol{x}(n-1), \boldsymbol{y}(n-1), \boldsymbol{J}_{\text{in}}(n), \Delta\boldsymbol{z}(n)) \end{cases}, \tag{5.13}$$

where $\boldsymbol{F} = (F_1^1, F_1^2, \ldots, F_l^i, \ldots, F_L^{N-1}, F_L^N)$ and $\boldsymbol{G} = (G_1^1, G_1^2, \ldots, G_l^i, \ldots, G_L^{N-1}, G_L^N)$ are constructed out of the nonlinear map F_l^i and G_l^i that depend on the i^{th} dynamical states (x_l^i, y_l^i) of the l^{th} layer and the external input signal with state noise; $\boldsymbol{J}_{\text{in}}(n) = \boldsymbol{J}_1(n)$ is the external input signal, which is only received by the first reservoir layer. Indeed, we are interested in the transient response of the total reservoir system to the external input signal at a given time as illustrated in Eq.5.13.

5.4 Jacobian linearization and the connectivity matrix

The evolution of dynamical states $(\boldsymbol{x}(n), \boldsymbol{y}(n))$ of τ-segments has been introduced in Eq.5.13. Consider the stable equilibrium $(\bar{x}_l, \bar{y}_l)$ of the autonomous system Eq.5.4 for layer l or, equivalently, the stable fixed point of Eq.5.11 of the form $\{(x_l^i, y_l^i) \,|\, x_l^i =$

$\bar{x}_l$, $y_l^i = \bar{y}_l, i = 1, 2, \ldots, N\}$ with $\boldsymbol{J}_{\text{in}} = \boldsymbol{0}$, $\Delta \boldsymbol{z} = \boldsymbol{0}$ in each layer l. Then the total system Eq.5.13 has the stable fixed point of the form

$$\{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \,|\, \bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_1, \ldots, \bar{x}_l, \ldots, \bar{x}_L),\ \bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_1, \ldots, \bar{y}_l, \ldots, \bar{y}_L),\ l = 1, 2, \ldots, L\}$$

To approximate Eq.5.13 by its Jacobian linearization at $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0})$, we obtain an expression of the form:

$$\begin{cases} \boldsymbol{x}(n) = & \bar{\boldsymbol{x}} + A_F(\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}_l) + B_F(\boldsymbol{y}(n-1) - \bar{\boldsymbol{y}}) + C_F \boldsymbol{J}_{\text{in}}(n) \\ & + D_F \Delta \boldsymbol{z}(n) \\ \boldsymbol{y}(n) = & \bar{\boldsymbol{y}} + A_G(\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}) + B_G(\boldsymbol{y}(n-1) - \bar{\boldsymbol{y}}) + C_G \boldsymbol{J}_{\text{in}}(n) \\ & + D_G \Delta \boldsymbol{z}(n) \end{cases}, \tag{5.14}$$

where

$$\begin{aligned} &A_F = D_{\boldsymbol{x}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}),\ B_F = D_{\boldsymbol{y}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}), \\ &C_F = D_{\boldsymbol{J}_{\text{in}}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}),\ D_F = D_{\Delta \boldsymbol{z}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}), \\ &A_G = D_{\boldsymbol{x}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}),\ B_G = D_{\boldsymbol{y}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}), \\ &C_G = D_{\boldsymbol{J}_{\text{in}}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}),\ D_G = D_{\Delta \boldsymbol{z}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}), \end{aligned}$$

are the corresponding Jacobian matrices. For the detail of the calculation process for these Jacobian matrices, we refer to Appendix D. The matrix $A_F \in \mathbb{R}^{N_{\text{tot}} \times N_{\text{tot}}}$ is referred to as the connectivity matrix. The feedforward connections from the external input signal $\boldsymbol{J}_{\text{in}}$ into the reservoir layer are represented as the product of the matrix $C_F \in \mathbb{R}^{N_{\text{tot}} \times N}$ and the input mask M.

Using Eq.5.14, the states of the collection of the virtual neurons at time n has the solution

$$\begin{cases} \boldsymbol{x}(n) = & \bar{\boldsymbol{x}} + Q_F(n)(\boldsymbol{x}(0) - \bar{\boldsymbol{x}}) + P_F(n)(\boldsymbol{y}(0) - \bar{\boldsymbol{y}}) \\ & + \sum_{k=0}^{\infty} W_F(k) \boldsymbol{J}_{\text{in}}(n-k) + \sum_{k=0}^{\infty} H_F(k) \Delta \boldsymbol{z}(n-k) \\ \boldsymbol{y}(n) = & \bar{\boldsymbol{x}} + Q_G(n)(\boldsymbol{x}(0) - \bar{\boldsymbol{x}}) + P_G(n)(\boldsymbol{y}(0) - \bar{\boldsymbol{y}}) \\ & + \sum_{k=0}^{\infty} W_G(k) \boldsymbol{J}_{\text{in}}(n-k) + \sum_{k=0}^{\infty} H_G(k) \Delta \boldsymbol{z}(n-k) \end{cases}. \tag{5.15}$$

Here the coefficient matrices can be calculated recursively as follows

$$\begin{aligned}
Q_F(n) =& A_F Q_F(n-1) + B_F Q_G(n-1),\\
P_F(n) =& A_F P_F(n-1) + B_F P_G(n-1),\\
W_F(n) =& A_F W_F(n-1) + B_F W_G(n-1),\\
H_F(n) =& A_F H_F(n-1) + B_F H_G(n-1),\\
Q_G(n) =& A_G Q_F(n-1) + B_G Q_G(n-1),\\
P_G(n) =& A_G P_F(n-1) + B_G P_G(n-1),\\
W_G(n) =& A_G W_F(n-1) + B_G W_G(n-1),\\
H_G(n) =& A_G H_F(n-1) + B_G H_G(n-1).
\end{aligned}$$

Note that the time-multiplexed signal $\boldsymbol{J}_{\text{in}}(n)$ is linearly dependent on the sampled input signal $u(n\tau)$ with the mask vector M. Because the long-term behavior near a sable equilibrium does not depend significantly on the initial conditions, Eq.5.15 can be expressed as follows

$$\begin{cases} \boldsymbol{x}(n) = & \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} W_F(k) M u((n-k)\tau) + \sum_{k=0}^{\infty} H_F(k) \Delta \boldsymbol{z}(n-k) \\ \boldsymbol{y}(n) = & \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} W_G(k) M u((n-k)\tau) + \sum_{k=0}^{\infty} H_G(k) \Delta \boldsymbol{z}(n-k) \end{cases} . \tag{5.16}$$

5.5 Fisher memory curve at the equilibrium

In this section, we measure the ability of the deep TDR to hold information in short-term memory. Recalling that the states of virtual neurons is derived from the first component x_l of the dynamical state $\boldsymbol{v}_l$ in each layer l. Then the fisher information is applied on $\boldsymbol{x}(n)$ at time n. Assuming that the increment $\Delta \boldsymbol{z}(n)$ follows the Gaussian distribution with zero mean and covariance $\langle \Delta z_l^i(k) \Delta z_l^j(l) \rangle = \epsilon\theta \delta_{k,l} \delta_{i,j}$. This implies that the conditional distribution $p(\boldsymbol{x}(n)|\boldsymbol{u})$ is also Gaussian with mean

$$\mu = \bar{\boldsymbol{x}} + \sum_{k=0}^{\infty} W_F(k) M u((n-k)\tau), \tag{5.17}$$

and covariance matrix

$$\boldsymbol{C}_n = \epsilon\theta \sum_{k=0}^{\infty} H_F(k) H_F(k)^{\top}. \tag{5.18}$$

By using Eq.3.5, the Fisher memory curve $\{I_k\}_{k=0}^{\infty}$ at the equilibrium $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0})$ can be described as

$$I_k = M^{\top} W_F^{\top}(k) \boldsymbol{C}_n^{-1} W_F(k) M. \tag{5.19}$$

We must note that for deep TDRs the Fisher memory curve measures the information of the past input signal held by the all reservoir layers.

Under the condition of $\beta_l \in (0, 1]$, $l = 1, 2, \ldots, L$, the deep TDRs based on couple Ikeda systems operate in the dynamical regime around the fixed point $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \mathbf{0}, \mathbf{0})$. In Figure 5.3, we plot the Fisher memory curves for the two-layer deep TDRs. The input masks are always taken as the eigenvector corresponding to the largest eigenvalue of I^s such that the short-term memory capacity is optimized. These figures show how the memory trace fades away with the time step k in the presence of state noise. Firstly, we choose the input gain κ_1 as the dependent variable belongs on the vertical axis, and the coupling strength κ_2 between the layers 1 and 2 is fixed at 0.9, as illustrated in the left panel of Figure 5.3a. These results show that there is no significant relationship between the input gain κ_1 and the Fisher memory curve. In the right panel of Figure 5.3a, the Fisher memory curve for different levels of coupling strength κ_2 with $\kappa_1 = 0.5$ is presented. The coupling strength κ_2 and the Fisher memory curve have no significant relationship, according to these findings. The relationship between the feedback gain β_l and the fisher memory curve is then discussed. The left panel of Figure 5.3b shows that the high-level feedback gain is helpful for improving the short-term memory in the presence of state noise. We confirmed that similar results in the left panel of Figure 5.2a. In the right panel of Figure 5.3b, we plot the Fisher memory curve for various feedback gain β_2 of the 2nd layer with $\beta_1 = 0.6$. These results indicate that there is no significant relationship between the input gain β_2 and the Fisher memory curve. This is obvious that in our model the architecture of the deep TDRs is the unidirectional coupling between the recurrent layers, such as feed-forward connections. The interaction between the two layers, in other words, is one-way. The feedback gain β_1 has an effect not only on the first layer but also on the second layer. The feedback increase effect of β_2 cannot be used on the first layer.

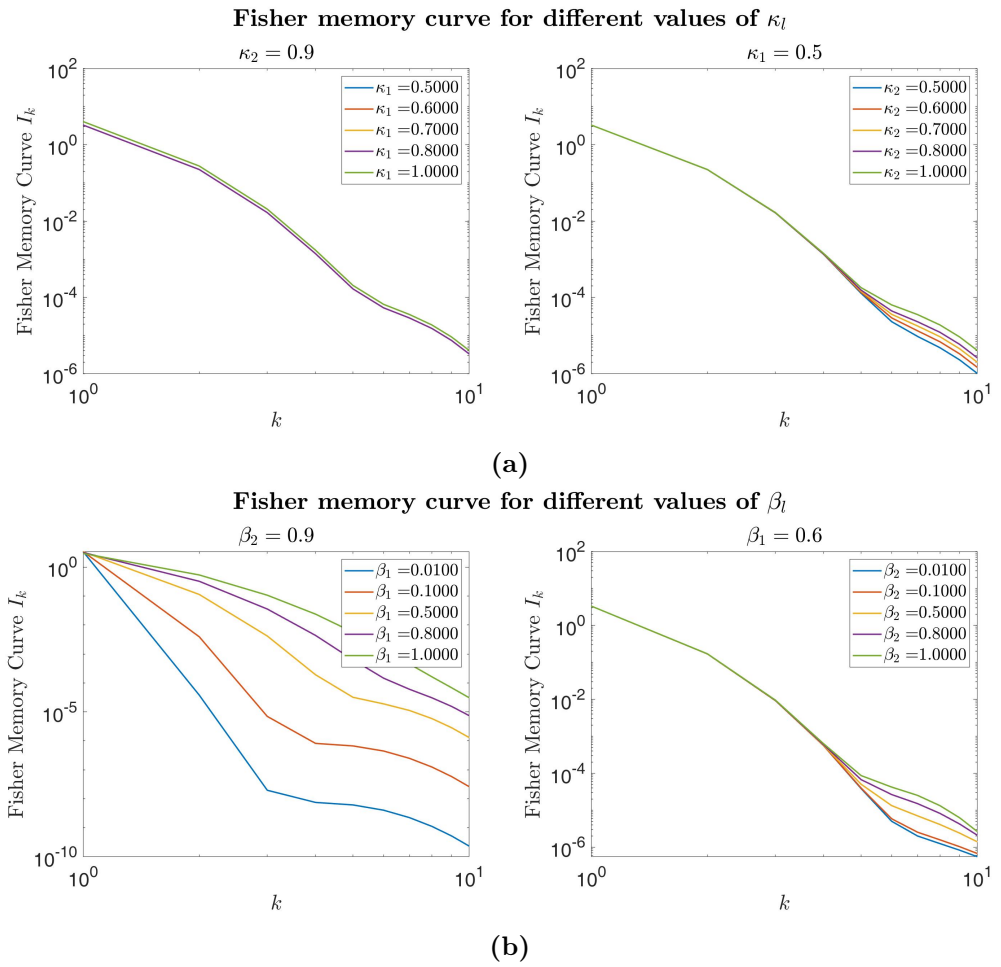


Figure 5.3: Fisher memory curve exhibited by the deep TDRs based on two coupled Ikeda delay systems. (a) In the left panel, $\beta_2 = 0.9$ is fixed; in the right panel, $\beta_1 = 0.6$ is fixed. (b) In the left panel, $\beta_2 = 0.9$ is fixed; in the right panel, $\beta_1 = 0.6$ is fixed. We used the same parameters setting: $\tau_1 = \tau_2 = 20$, $\delta_1 = 0$, $\delta_2 = 0.01$, $b_1 = b_2 = 0.2$ and $\theta = 0.2$.

5.6 Optimized Input mask in deep TDRs with state noise

Table 5.1: Parameters of the three-layer deep TDR used for the calculation of the Fisher memory curve in Figure 5.4 and for the simulation experiments in section 5.7.

	Delay τ	Damping δ	Phase b	feedback β	coupling κ
1st layer	20	0	0.2	0.68	0.4
2nd layer	20	0.01	0.2	0.8	1
3rd layer	20	0.01	0.2	0.97	1

The total memory is

$$I_{\text{tot}} = \sum_{k=0}^{\infty} I_k = M^\top I^s M, \tag{5.20}$$

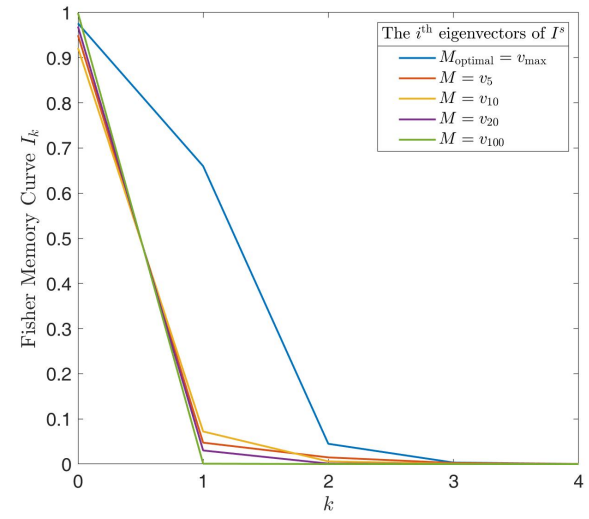
where

$$I^s = \sum_{k=0}^{\infty} W_F^\top(k) \boldsymbol{C}_n^{-1} W_F(k). \tag{5.21}$$

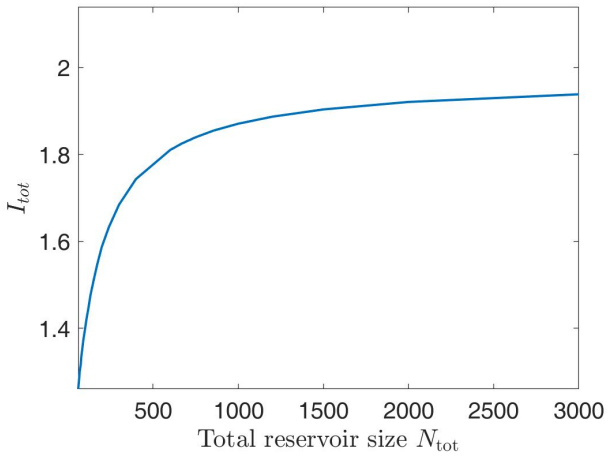
We define the eigenvalues of I^s as $\lambda_{\max} \geq \lambda_2 \geq \ldots \geq \lambda_N$, which are arranged in descending order, and the corresponding eigenvectors as $v_{\max}, v_2, \ldots, v_N$. I_{tot} can be maximized by the largest eigenvalue $\lambda_{\max}$ of I^s, when the associated eigenvector is chosen to be the input mask as $M = v_{\max}$. The heatmap used for visualizing the eigenvectors of I^s is shown in Appendix E.

The Fisher memory curves generated by employing the chosen eigenvectors as input masks are presented in Figure 5.4. The optimal input mask $M = v_{\max}$ is represented by the blue solid line. The used parameters are given in Table 5.1.

We show how the total memory I_{tot} evolves for the TDR as a function of the number of reservoir neurons in Figure 5.4b. The figure illustrates that increasing the number of reservoir neurons can prevent the memory from decaying in the presence of state noise.



(a)



(b)

Figure 5.4: Short-term memory capacity for three coupled Ikeda delay systems with the chosen input masks. The used parameters are given in Table 5.1.

5.7 Simulation experiments

A variety of benchmark tasks are available to provide information on the processing power of the deep TDRs. Each of these tasks has different essential properties to accurately estimate the target function. In this section, we provide the simulation experiments of three types of learning tasks: h-lag memory tasks, weakly nonlinear

function approximation task and chaotic attractor learning task.

5.7.1 h-lag memory task

We compare the optimized input mask with two hundred random input masks for different noise levels. The elements of these input masks are drawn from a uniform distribution in the interval $(-1, 1)$. In order to avoid changing the strength of the input signal, these random input masks are normalized.

h-lag linear memory task

The ability to reconstruct the past input signal from the current reservoir state is the focus of the linear memory tasks. In the context of reservoir computing, it has been utilized in many other publications, such as in [14], [58]. The linear memory tasks are associated with the linear function $F_h : \mathbb{R}^{h+1} \to \mathbb{R}$. Let the independent variable $\boldsymbol{u}_h(t) = (u(t), u(t-1), \ldots, u(t-h))^\top$ and the linear operator $L_h \in \mathbb{R}^{h+1}$, then

$$F_h(\boldsymbol{u}_h(t)) = L_h^\top \boldsymbol{u}_h(t). \tag{5.22}$$

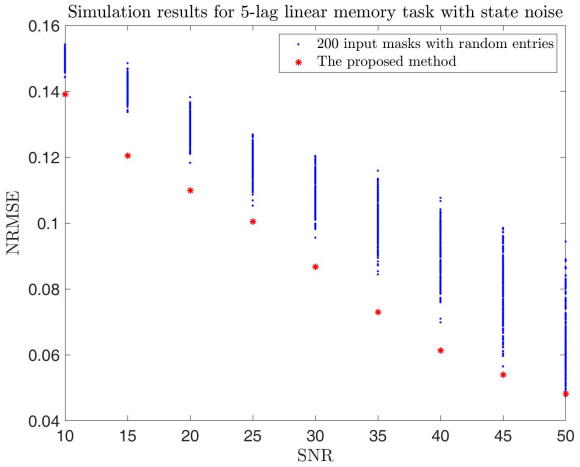
h-lag quadratic memory task

The nonlinear memory capacity is introduced in [64] as a generalization of the linear memory capacity. It can be investigated by the quadratic task memory task, which links to a quadratic function $H_h : \mathbb{R}^{h+1} \to \mathbb{R}$. The form of H_h is given by

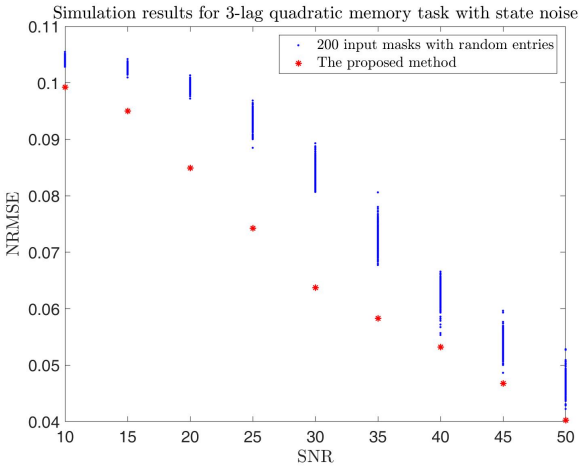
$$\begin{aligned} H_h(\boldsymbol{u}_h(t)) &= \boldsymbol{u}_h^\top(t) Q_h \boldsymbol{u}_h(t) \\ &= \sum_{i=1}^{h+1} \sum_{j=1}^{h+1} Q_{ij} u(t-i+1) u(t-j+1), \end{aligned} \tag{5.23}$$

where $Q_h \in \mathbb{R}^{(h+1)\times(h+1)}$ is a real symmetric matrix such that $Q_{ij} = Q_{ji}$ for $i \in \mathbb{Z}^+$ and $j \in \mathbb{Z}^+$.

In our experiments, the input signal $u(t)$ consists of scalar random numbers drawn from a normal distribution with mean zero and standard deviation one. The parameter setting of the deep TDRs is given in Table 5.1. The separation distance of the virtual neurons is set at $\theta = 0.2$ that offers a suitable short-term memory capacity as in Fig.5.4b.



(a)



(b)

Figure 5.5: Reservoir performance for the 5-lag linear memory task and the 3-lag quadratic memory task. (*a*) NRMSEs for the 5-lag linear memory task as a function of the input mask and SNR. (*b*) NRMSEs for the 3-lag quadratic memory task as a function of the input mask and SNR. The blue points indicate the NRMSEs obtained from the 200 different randomly picked masks. The red asterisk points indicate the NRMSEs simulated by using the input mask optimized by the proposed method. The used parameters are given in Table 5.1.

In practice, we let the deep TDRs run for 7000 steps, starting from random initial states. We use the first 5000 steps for training, but discard the first 100 steps, during which the initial state's effect fades away. The last 2000 steps are used for determining the test errors. In Figure 5.5a, we plot the simulation results for the 5-lag

linear memory task with different noise levels. The blue points indicate the NRMSEs obtained from the 200 different randomly picked masks. The red asterisk points indicate the NRMSEs simulated by using the input mask optimized by the proposed method. The lowest value is NRMSE = 0.0482 obtained by the optimized input mask for SNR = 50. Even for a high noise level, such as SNR = 10, the optimized input masks outperform the 200 normalized random input masks. On the other hand, as the noise level increases, the performance of the deep TDRs is significantly degraded. These results are consistent with the previous simulation results of the consistency parameter illustrated in Figure 5.2. Analogously to the 3-lag linear memory task case, we plot the simulation results for the 3-lag quadratic memory task with different noise levels in Figure 5.5b. We confirmed that similar results are obtained.

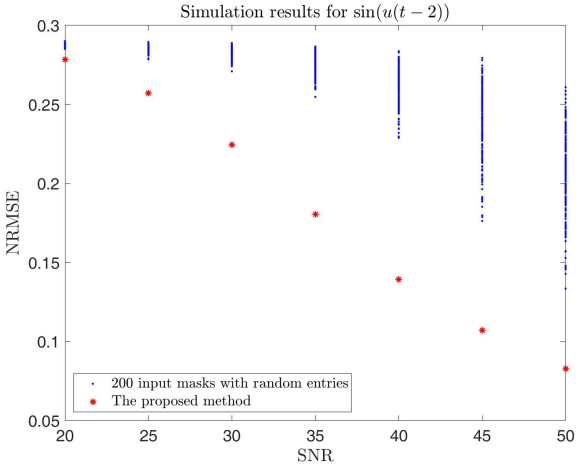
5.7.2 Weakly nonlinear function approximation task

The description of the weakly nonlinear function approximation task has been introduced in section 4.6.1 above. The input signal $u(t)$ is drawn from a uniform distribution in the interval $(-0.1, 0.1)$. In a similar way to the case of the h-lag memory task, the deep TDRs are trained from a run of 5000 steps, of which the first 100 steps are discarded. The trained deep TDRs are tested by 2000 steps.

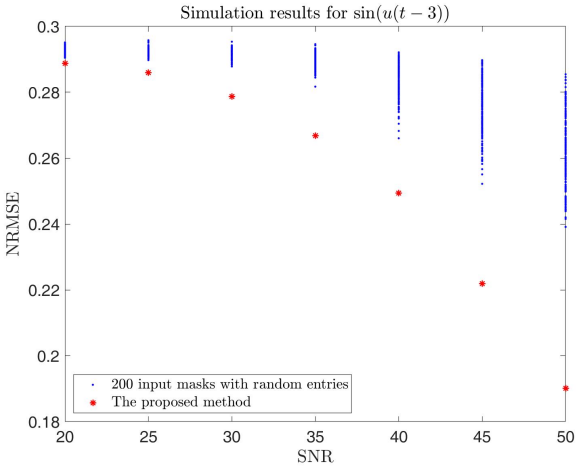
We compare the input mask optimized by the proposed method with 200 random input masks at the first step in Figure 5.6. When scanning the reservoir performance against the SNR, we obtain the simulation results of the learning task of $\sin(u(t-2))$ in Figure 5.6a. The used parameters are kept constant in Table 5.1. It is obvious that the higher SNR is, the greater the performance gap caused by different input masks is. The best performance can be found for the SNR = 50 with optimized input mask, where the minimum reached NRMSE is 0.0828. We show the simulation results for the learning task of $\sin(u(t-3))$ with different noise levels in Figure 5.6b. For SNR = 50, we experimentally obtain the lowest test error with NRMSE = 0.1902 using the optimized input mask.

Next, we use the eigenvectors of I^s as the input mask and compare their test errors when scanning the feedback gain β_1 of the 1st layer. Assuming that the eigenvectors of the spatial Fisher memory matrix I^s is denoted by $v_{\max}, v_2, \ldots, v_N$, which are arranged in descending order according to the eigenvalues of I^s as $\lambda_{\max} \geq \lambda_2 \geq \ldots \geq \lambda_N$. Figure 5.7a depicts the test error of the task $\sin(u(t-2))$ as a function of the feedback gain β_1 and the input mask. A narrow blue region with NRMSE < 0.15 has been obtained by the optimized input mask. The minimum is NRMSE = 0.0782 when $\beta_1 = 0.95$. For the optimized input masks, a decrease in the NRMSE corresponds to an increase in the feedback gain β_1. The green region with $0.15 <$ NRMSE < 0.2 implies the region where performance is reasonable. The red region denotes the point at which the reconstruction of the target signal fails altogether. A similar result is

obtained by employing the task of $\sin(u(t-3))$ illustrated in Figure 5.8. In this task, the proposed method is failed for the feedback gain at low level $\beta_1 \leq 0.2$. The minimum is NRMSE $= 0.1706$ when $\beta_1 = 0.95$.



(a)



(b)

Figure 5.6: Reservoir performance for the learning task of $\sin(u(t-\tau'))$. (a) NRMSEs for the task of $\sin(u(t-2))$ as a function of the input mask and SNR. (b) NRMSEs for the task of $\sin(u(t-3))$ as a function of the input mask and SNR. The blue points indicate the NRMSEs obtained from the 200 different randomly picked masks. The red asterisk points indicate the NRMSEs simulated by using the input mask optimized by the proposed method. The used parameters are given in Table 5.1.

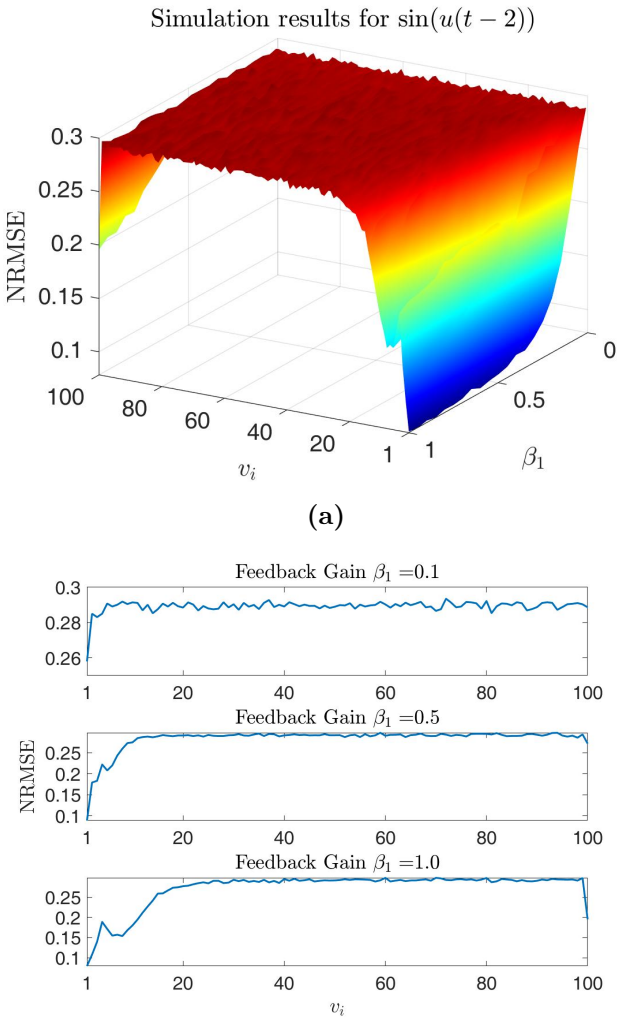


Figure 5.7: Reservoir performance for the learning task of $\sin(u(t-\tau'))$. (a) Error surface for $\sin(u(t-2))$, as a function of the input mask and the feedback gain β_1 of the 1st layer. (b) The NRMSE versus the input masks for $\sin(u(t-2))$ with parameter setting: $\beta_1 = 0.1, 0.5, 1.0$ from top to bottom. The used parameters are given in Table 5.1.

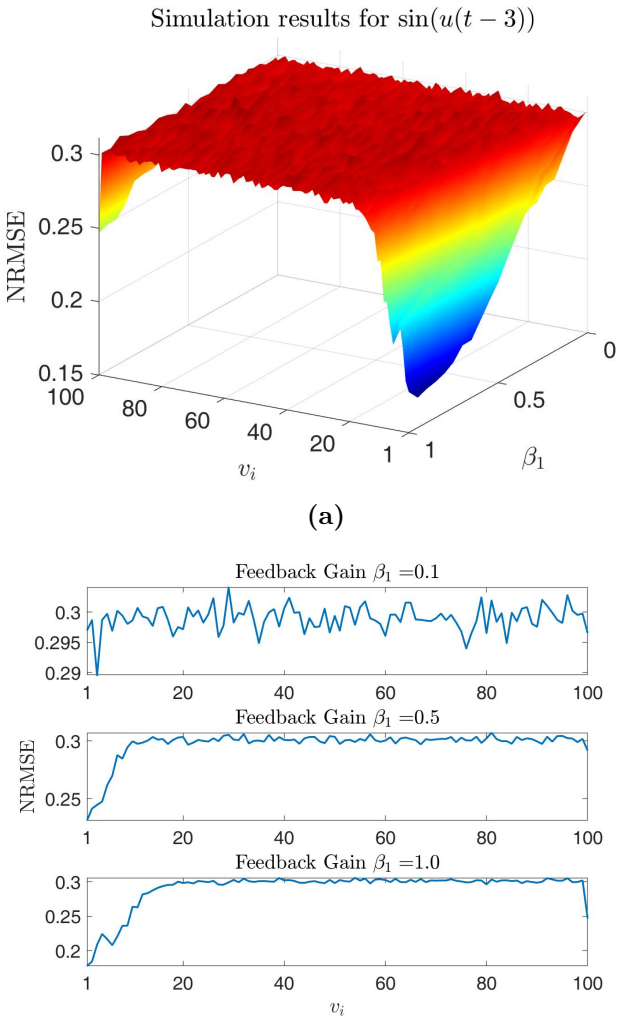
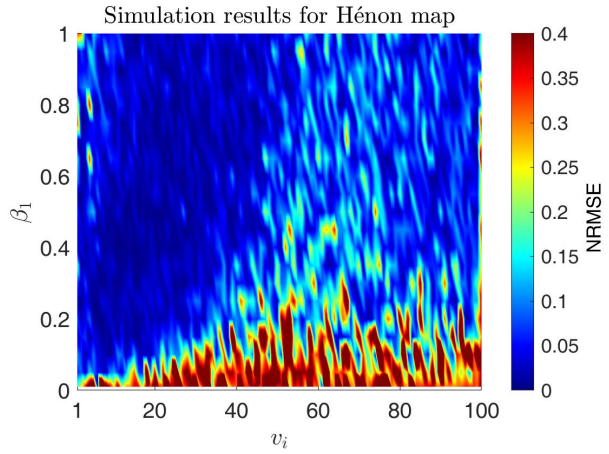


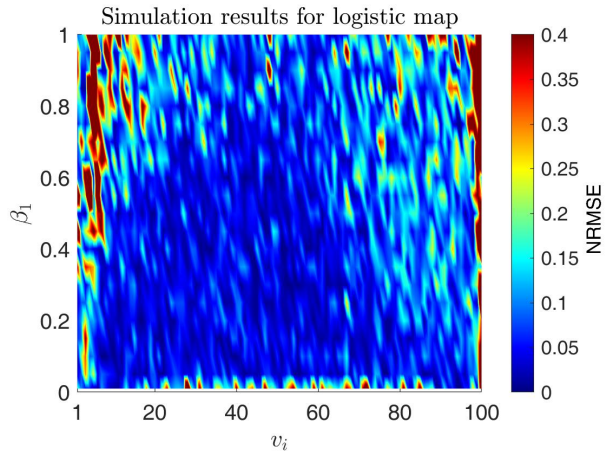
Figure 5.8: Reservoir performance for the learning task of $\sin(u(t-\tau'))$. (a) Error surface for $\sin(u(t-3))$, as a function of the input mask and the feedback gain β_1 of the 1st layer. (b) The NRMSE versus the input masks for $\sin(u(t-3))$ with parameter setting: $\beta_1 = 0.1, 0.5, 1.0$ from top to bottom. The used parameters are given in Table 5.1.

5.7.3 Chaotic attractor learning task

Here, we show numerical results for the chaotic attractor learning task, which has been introduced in section 4.6.2. In these simulation experiments, the signal-to-noise ratio of SNR = 60 for the Hénon map and SNR = 70 for the logistic map is obtained using the given state noise strength ϵ. When scanning the feedback gain β_1 of the first layer, we use the eigenvectors of I^s as the input mask and compare their test errors.



(a)



(b)

Figure 5.9: Simulation results for the learning tasks: Hénon map and logistic map. (a) Error surface for Hénon map, as a function of the input masks and the feedback gain β_1 of the 1st layer. (b) Error surface for logistic map, as a function of the input masks and the feedback gain β_1 of the 1st layer. The used parameters are given in Table 5.1 except for $\kappa_1 = 1$.

Concretely, the TDRs are trained from a run of 5000 steps and the first 100 steps were discarded. In Figure 5.9a, the NRMSE values are shown in color coding, while scanning the feedback gain of the first layer β_1 and the eigenvector v_i. This figure shows that the region of good performance is for high values of β_1 and the performance gap between the used input masks is not significant. On the other hand, the simulation results for the logistic map are shown in Figure 5.9b. In this case, a large region of good performance with NRMSE < 0.15 is obtained when $\beta_1 \in [0.1, 0.7]$.

5.8 Discussion

Here, we show the consistency parameter obtained by the random input mask and the input masks optimized by the proposed method which is introduced in Chapter 3. In Figure 5.10, the values of the consistency parameter are depicted by shades of mono-color, while scanning the feedback gain of the first layer β_1 and SNR. In the plot of Figure 5.10a, the simulation results of the random input masks are shown. As a contrast, the simulation results of the optimized input masks are illustrated in Figure 5.10b. Specifically, we compare the simulation results for random input masks and optimized input masks in different cases of $\beta_1 = 0.1, 0.3, 0.5, 0.7, 1$ in Figure 5.11. This figure shows that the proposed method significantly improves the echo state property in the presence of state noise.

We have successfully applied the proposed method to the deep TDRs based on coupled Ikeda time-delay systems in the presence of noise. The formula of Eq. 5.19 for the Fisher memory curve can be used to estimate the memory trace in such deep TDRs. The deep TDR has a number of parameters that can be adjusted, such as the feedback strength β_l, the input gain/coupling strength κ_l. Due to the unidirectional coupling of the Ikeda time-delay systems, the feedback strength β_l and the input gain/coupling strength κ_l of layer l have a great influence on the next layer $l + 1$, but not vice versa. By using the maximal principal component of the spatial Fisher memory matrix as the input mask, we succeed in improving the noise robustness of such deep TDRs. It is confirmed by implementing the deep TDRs on several benchmark tasks with random input masks and the optimized input masks. We found that the optimized input mask outperforms the random mask in both the h-lag memory task and the weakly nonlinear function approximation task. However, the proposed method is no longer effective for the chaotic attractor learning task which is strongly dependent on the nonlinear transformation capacity for good performance.

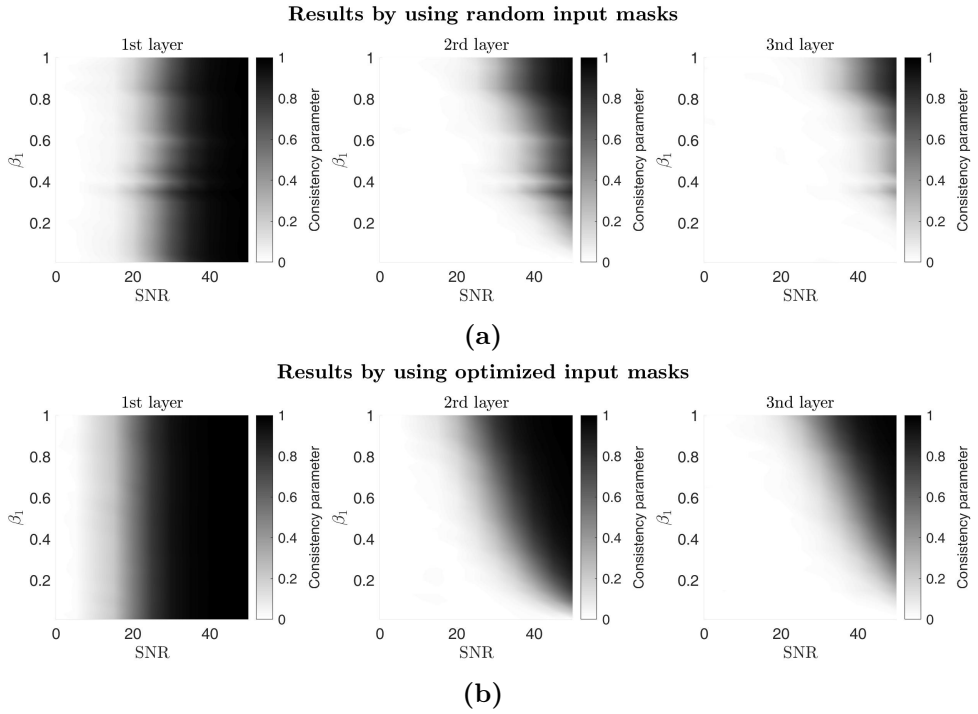


Figure 5.10: Consistency parameter for random input masks (upper) and optimized input masks (lower), as a function of the feedback gain of the first layer β_1 (vertical axis) and SNR level (horizontal axis). (a) Normalized random input masks. (b) Input masks optimized by the proposed method. The used parameters are given in Table 5.1.

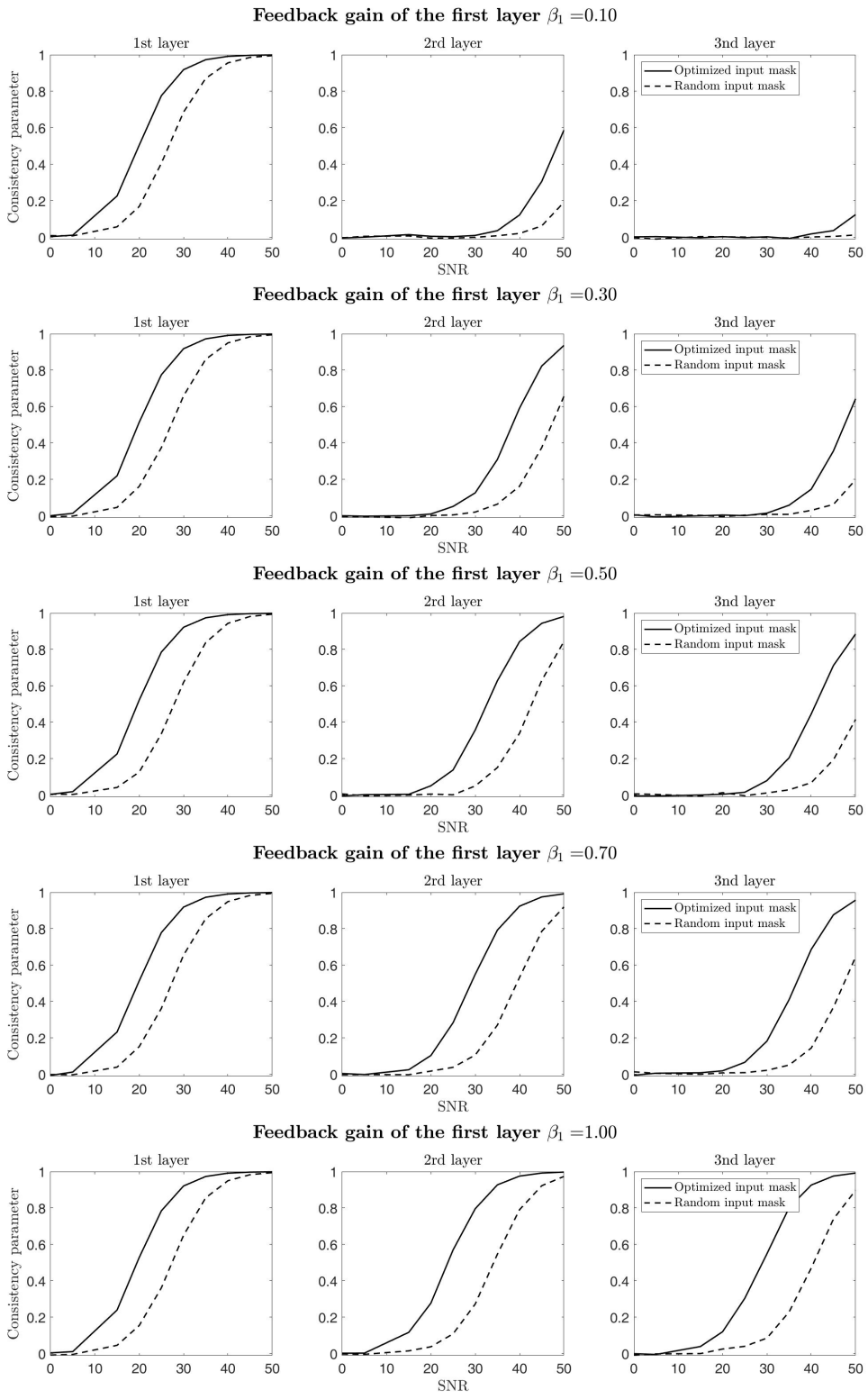


Figure 5.11: Consistency parameter for random input masks and optimized input masks with parameter setting: $\beta_1 = 0.1, 0.3, 0.5, 0.7, 1$ **from top to bottom**. The black dashed lines denote the consistency parameter obtained by random input masks. The black solid lines denote the consistency parameter obtained by optimized input masks. The used parameters are given in Table 5.1.

Chapter 6

Conclusions

Reservoir computing is a brain-inspired machine-learning framework that has been successfully used in information processing, such as attractor reconstruction or speech recognition. The basic structure of reservoir computing is composed of an input layer, a reservoir layer, and an output layer. The input layer connects the real-world signal to the reservoir layer via fixed weighted connections. Besides the artificial neural network, the reservoir layer can be consist of a variety of nonlinear dynamical systems with a rich space of internal states. We focus on the type of reservoir layer that is realized by using a single nonlinear physical node with a delay line. Such reservoirs are susceptible to both internal and external noise. In Figure 2.1, we show that how the echo state property of the TDRs is impaired by the state noise via the consistency parameter.

We have developed an optimization method for improving the noise robustness of the TDRs in the context of Fisher information. Firstly, instead of the conventional approach of measuring the short-term memory, we extend the Fisher memory framework to the TDRs in the presence of state noise. This measurement is only acceptable when the working regime of the TDRs remains close to a stable equilibrium. In such a case, the Fisher memory curve can be applied to the linear approximation of the TDRs at the stable equilibrium, and assess the capacity that how much information about the past input signal the current reservoir states can hold. The total memory capacity is indicated by the sum of the Fisher memory curve overall delay. It shows that the maximal total memory capacity can be characterized by the largest eigenvalue of the spatial Fisher memory matrix when the associated eigenvector is chosen to be the input mask. Such input mask implies the preferred direction in the reservoir state space corresponding to the large principal component of the spatial Fisher memory matrix. The short-term memory capacity is closely linked to the echo state property. To propagate and hold the input history, the TDRs requires the echo state property (or the corresponding reservoir map has a fading memory). In Figure

3.1, we plot how the consistency parameter evolves for two types of the TDRs as a function of the optimized input masks. Compared with the random input masks illustrated in Figure 2.1, the optimized input masks significantly improve the noise robustness of the TDRs.

Concretely, we applied the proposed method to two types of TDRs subjected to state noise. The first TDR is implemented by a single Mackey-Glass nonlinear oscillator. By using the consistency parameter, the echo state property of such TDRs with random input masks is illustrated in Figure 4.1. We optimized the input masks by using the proposed method and compared its performance with the existing optimization method for the h-lag memory task. We showed that the proposed method is running-time-efficient since it only involves a linear problem in the optimization procedure and improves memory performance in a great extent in the presence of state noise. Then several benchmark tasks were used to quantify the performance. For the function approximation task of $f(u(t)) = \sin(u(t-2))$, the minimal normalized root mean square error obtained by the optimized input mask is as low as NRMSE = 0.0216. And for another task of $f(u(t)) = \sin(u(t-3))$, the minimal NRMSE obtained by the optimized input mask equals 0.0953. For the chaotic attractor learning tasks, which require a strong nonlinear transformation, the optimized input masks were no longer providing good performance, such as NRMSE = 0.1240 in the case of the Hénon map and NRMSE = 0.1681 in the case of the logistic map. It is interpreted as the memory-nonlinearity trade-off. In Figure 4.9, we compared the consistency parameter obtained by the optimized input masks with the one obtained by the random input masks when scanning the feedback gain.

The second case is the deep TDRs based on coupled Ikeda delay systems. Analogously to the case of the Mackey-Glass oscillator, we tested the reservoir performance for three benchmark tasks. Numerically, for the 5-lag linear memory task the minimum NRMSE = 0.0482 obtained by the optimized input mask was found, and for the 3-lag quadratic memory task the minimum NRMSE = 0.0403 obtained by the optimized input mask was found. In the tasks of $f(u(t)) = \sin(u(t-2))$ and $f(u(t)) = \sin(u(t-3))$, we compared the optimized input mask with 200 random input masks. Both the minimum NRMSE = 0.0828 for $f(u(t)) = \sin(u(t-2))$ and the minimum NRMSE = 0.1902 for $f(u(t)) = \sin(u(t-3))$ were obtained by the optimized input mask at SNR = 50. We also tested the reservoir performance on these tasks by using the principal components of the spatial Fisher memory matrix as the input masks. The minimum NRMSE = 0.0782 for $f(u(t)) = \sin(u(t-2))$ and NRMSE = 0.1706 for $f(u(t)) = \sin(u(t-3))$ were obtained by using the maximal principal component of the spatial Fisher memory matrix as the input mask. However, the proposed method is no longer effective for the chaotic attractor learning task which is strongly dependent on the nonlinear transformation capacity for good performance. In Figure 5.10 and Figure 5.11, we showed the values of consistency

parameter for the random input masks and optimized input masks.

In conclusion, the method we have proposed could optimize the memory performance of the TDRs subjected to state noise with respect to input masks in a task-independent and computationally-efficient manner. By using the consistency parameter, we showed that the proposed method improves the echo state property of TDRs to a great extent in the presence of state noise. The proposed method facilitates the design of the input masks for improving the noise robustness of TDRs on memory tasks.

Appendices

A. Normalized root mean squared error

The normalized root mean squared error (NRMSE) is used to assess performance on a specific task.

$$\text{NRMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{\parallel \boldsymbol{y}(i) - \boldsymbol{y}^{\text{target}}(i) \parallel}{var(\boldsymbol{y}(i))}},$$

where n is the length of testing steps, $\boldsymbol{y}$(i) is the output of the trained reservoir system at time step i, and $\boldsymbol{y}^{\text{target}}(i)$ is the corresponding desired output. Furthermore, $var(\boldsymbol{y}(i))$ is the variance of the observation output, and $\parallel \parallel$ is the Euclidean norm.

B. Eigenvectors of the spatial Fisher memory matrix in TDRs

In Figure 6.1, the heatmap is used to visualize the eigenvectors of the spatial Fisher memory matrix I^s. The eigenvectors of indices ranging from 27^{th} to 399^{th} seem spurious. The reason is that the eigenvalues with indices of 27^{th} or more are located extremely close to unity.

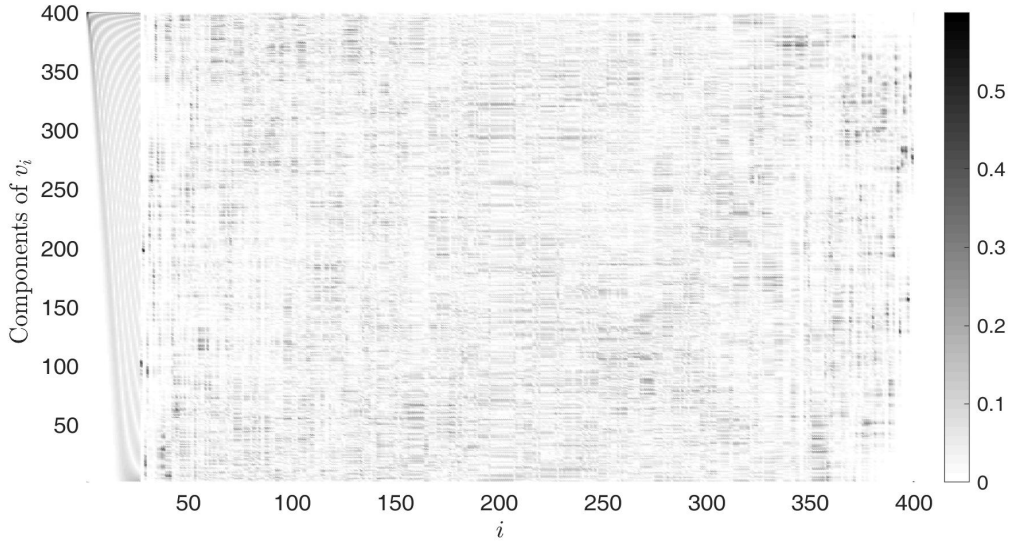


Figure 6.1: Heatmap of the eigenvectors of I^s with parameter setting: $\gamma = 0.1$, $p = 1$, $\eta = 0.9$, $\tau = 80$ **and** $\theta = 0.2$. The color of each pixel is determined by the absolute value of components of the input mask. The indices of the input masks are arranged in descending order according to the eigenvalues of I^s.

C. Coupling strength in deep TDRs with state noise

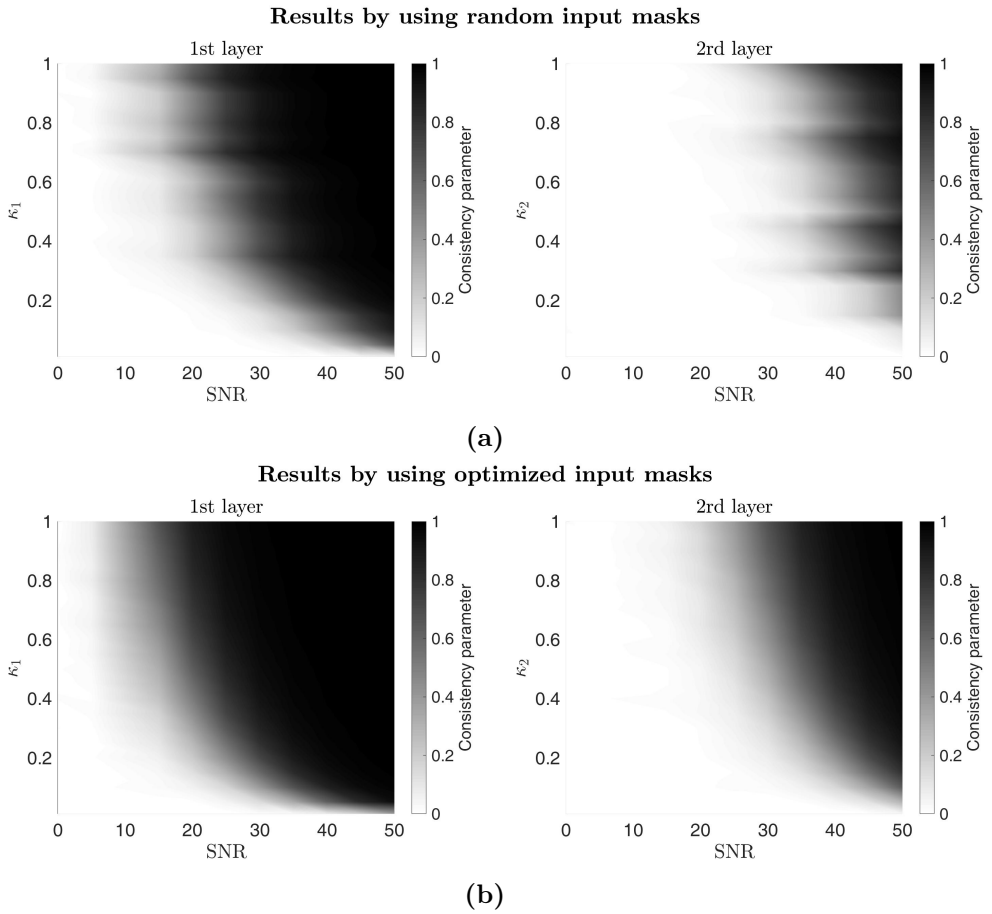


Figure 6.2: Consistency parameter for random input masks (upper) and optimized input masks (lower), as a function of the coupling strength κ_l (vertical axis) and SNR level (horizontal axis). (a) Normalized random input masks. (b) Optimized input masks using the proposed method. The type of two-layer deep TDRs based on coupled Ikeda delay system is considered with parameter setting: $\tau_1 = \tau_2 = 20$, $\delta_1 = 0$, $\delta_2 = 0.01$, $\beta_1 = 0.68$, $\beta_2 = 0.8$, $b_1 = b_2 = 0.2$ and $\theta = 0.2$.

D. The connectivity matrix of deep TDRs

The evolution of τ-segments $\boldsymbol{x}(n)$ and $\boldsymbol{y}(n)$ is governed by

$$\begin{cases} \boldsymbol{x}(n) = & \boldsymbol{F}(\boldsymbol{x}(n-1), \boldsymbol{y}(n-1), \boldsymbol{J}_{\text{in}}(n), \Delta\boldsymbol{z}(n)) \\ \boldsymbol{y}(n) = & \boldsymbol{G}(\boldsymbol{x}(n-1), \boldsymbol{y}(n-1), \boldsymbol{J}_{\text{in}}(n), \Delta\boldsymbol{z}(n)) \end{cases},$$

which has been introduced in Eq.5.11. Consider the stable equilibrium $(\bar{x}_l, \bar{y}_l)$ of the autonomous system Eq.5.4 for layer l or, equivalently, the stable fixed point of Eq.5.11 of the form $\{(x_l^i, y_l^i) \,|\, x_l^i = \bar{x}_l,\ y_l^i = \bar{y}_l, i = 1, 2, \ldots, N\}$ with $\boldsymbol{J}_{\text{in}} = \boldsymbol{0}$, $\Delta \boldsymbol{z} = \boldsymbol{0}$ in each layer l. Then the total system Eq.5.13 has the stable fixed point of the form

$$\{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \,|\, \bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_1, \ldots, \bar{x}_l, \ldots, \bar{x}_L),\ \bar{\boldsymbol{y}} = (\bar{y}_1, \bar{y}_1, \ldots, \bar{y}_l, \ldots, \bar{y}_L),\ l = 1, 2, \ldots, L\}$$

To approximate Eq.5.13 by its Jacobian linearization at $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0})$, we obtain an expression of the form:

$$\begin{cases} \boldsymbol{x}(n) = & \bar{\boldsymbol{x}} + A_F(\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}_l) + B_F(\boldsymbol{y}(n-1) - \bar{\boldsymbol{y}}) + C_F \boldsymbol{J}_{\text{in}}(n) \\ & + D_F \Delta \boldsymbol{z}(n) \\ \boldsymbol{y}(n) = & \bar{\boldsymbol{y}} + A_G(\boldsymbol{x}(n-1) - \bar{\boldsymbol{x}}) + B_G(\boldsymbol{y}(n-1) - \bar{\boldsymbol{y}}) + C_G \boldsymbol{J}_{\text{in}}(n) \\ & + D_G \Delta \boldsymbol{z}(n) \end{cases} . \tag{6.1}$$

The expression of A_F

The Jacobian matrix $A_F \in \mathbb{R}^{N_{\text{tot}} \times N_{\text{tot}}}$, $N_{\text{tot}} = LN$, is given by

$$A_F = D_{\boldsymbol{x}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}).$$

Its element at the intersection of row $i \times l$ and column $j \times k$ is obtained by $\frac{\partial F_l^i}{\partial x_k^j}$. This value is calculated as follows:
If $l = k$,

$$\frac{\partial F_l^1}{\partial x_k^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = 1 \\ 0, \text{ for } i = 1,\ j < N \\ \frac{1}{1+\theta+\delta_l \theta^2}, \text{ for } i = 1,\ j = N \end{cases}$$

$$\frac{\partial F_l^i}{\partial x_k^j} = \begin{cases} \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial x_l^j} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial x_l^j}, \text{ for } 1 < j < i < N \\ \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } 1 < i = j < N \\ 0, (1 < i < j < N) \\ \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial x_l^N} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial x_l^N}, \text{ for } 1 < i < N,\ j = N \end{cases}$$

$$\frac{\partial F_l^N}{\partial x_k^j} = \begin{cases} \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{N-1}}{\partial x_l^j} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{N-1}}{\partial x_l^j}, \text{ for } i = N,\ 1 \le j < N \\ \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{N-1}}{\partial x_l^N} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{N-1}}{\partial x_l^N} + \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = N \end{cases}$$

If $l \neq k$,

$$\frac{\partial F_l^i}{\partial x_k^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l \sin(2(\bar{x}_l + b_l)) \frac{\partial F_{l-1}^i}{\partial x_k^j}, \text{ for } l > k,\ 1 \le i \le N,\ 1 \le j \le N \\ 0, \text{ for } l < k,\ 1 \le i \le N,\ 1 \le j \le N \end{cases}$$

The expression of B_F

The Jacobian matrix $B_F \in \mathbb{R}^{N_{\rm tot} \times N_{\rm tot}}$ is given by

$$B_F = D_{\boldsymbol{y}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0})).$$

Its the $(i \times l, j \times k)$-th element is obtained by $\frac{\partial F_l^i}{\partial y_k^j}$. This value is calculated as follows:
If $l = k$,

$$\frac{\partial F_l^1}{\partial y_k^j} = \begin{cases} 0, \text{ for } i = 1,\ j < N \\ -\frac{\delta_l \theta}{1+\theta+\delta_l \theta^2}, \text{ for } i = 1,\ j = N \end{cases}$$

$$\frac{\partial F_l^i}{\partial y_k^j} = \begin{cases} 0, \text{ for } 1 < i \leq N,\ j < N \\ \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial y_k^N} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial y_k^N}, \text{ for } 1 < i \leq N,\ j = N \end{cases}$$

If $l \neq k$,

$$\frac{\partial F_l^i}{\partial y_k^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l \sin(2(\bar{x}_l + b_l)) \frac{\partial F_{l-1}^i}{\partial y_k^j}, \text{ for } l > k,\ 1 \leq i \leq N,\ 1 \leq j \leq N \\ 0, \text{ for } l < k,\ 1 \leq i \leq N,\ 1 \leq j \leq N \end{cases}$$

The expression of C_F

The Jacobian matrix $C_F \in \mathbb{R}^{N_{\rm tot} \times N}$ is given by

$$C_F = D_{\boldsymbol{J}_{\rm in}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0})).$$

Its the $(i \times l, j)$-th element is obtained by $\frac{\partial F_l^i}{\partial J_{\rm in}^j}$. This value is calculated as follows:
If $l = 1$,

$$\frac{\partial F_l^1}{\partial J_{\rm in}^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l sin(2(\bar{x}_l + b_l)), \text{ for } i = j = 1 \\ 0, \text{ for } i = 1,\ 1 < j \leq N \end{cases}$$

$$\frac{\partial F_l^i}{\partial J_{\rm in}^j} = \begin{cases} \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial J_{\rm in}^j} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial J_{\rm in}^j}, \text{ for } 1 \leq j < i < N \\ \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l \sin(2(\bar{x}_l + b_l)), \text{ for } 1 < i = j < N \\ 0, \text{ for } 1 < i < j \leq N \end{cases}$$

$$\frac{\partial F_l^N}{\partial J_{\rm in}^j} = \begin{cases} \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{N-1}}{\partial J_{\rm in}^j} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{N-1}}{\partial J_{\rm in}^j}, \text{ for } i = N,\ 1 \leq j < N \\ \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = N \end{cases}$$

If $l > 1$,

$$\frac{\partial F_l^i}{\partial J_{\rm in}^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l \sin(2(\bar{x}_l + b_l)) \frac{\partial F_{l-1}^i}{\partial J_{\rm in}^j}, \text{ for } 1 \leq i \leq N,\ 1 \leq j \leq N \end{cases}$$

The expression of D_F

The Jacobian matrix $D_F \in \mathbb{R}^{N_{\mathrm{tot}} \times N_{\mathrm{tot}}}$ is given by

$$D_F = D_{\Delta \boldsymbol{z}} F(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}).$$

Its the $(i \times l, j \times k)$-th element is obtained by $\frac{\partial F_l^i}{\partial \Delta z_k^j}$. This value is calculated as follows:
If $l = k$,

$$\frac{\partial F_l^1}{\partial \Delta z_k^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = 1 \\ 0, \text{ for } i = 1,\ 1 < j \leq N \end{cases}$$

$$\frac{\partial F_l^i}{\partial \Delta z_k^j} = \begin{cases} \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial \Delta z_k^j} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial \Delta z_k^j}, \text{ for } 1 \leq j < i < N \\ \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } 1 < i = j < N \\ 0, \text{ for } 1 < i < j \leq N \end{cases}$$

$$\frac{\partial F_l^N}{\partial \Delta z_k^j} = \begin{cases} \frac{1}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{N-1}}{\partial \Delta z_k^j} - \frac{\delta_l \theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{N-1}}{\partial \Delta z_k^j}, \text{ for } i = N,\ 1 \leq j < N \\ \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = N \end{cases}$$

If $l \neq k$,

$$\frac{\partial F_l^i}{\partial \Delta z_k^j} = \begin{cases} \frac{\beta_l \theta}{1+\theta+\delta_l \theta^2} \kappa_l \sin(2(\bar{x}_l + b_l)) \frac{\partial F_{l-1}^i}{\partial \Delta z_k^j}, \text{ for } l > k,\ 1 \leq i \leq N,\ 1 \leq j \leq N \\ 0, \text{ for } l < k,\ 1 \leq i \leq N,\ 1 \leq j \leq N \end{cases}$$

The expression of A_G

The Jacobian matrix $A_G \in \mathbb{R}^{N_{\mathrm{tot}} \times N_{\mathrm{tot}}}$ is given by

$$A_G = D_{\boldsymbol{x}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}).$$

Its the $(i \times l, j \times k)$-th element is obtained by $\frac{\partial G_l^i}{\partial x_k^j}$. This value is calculated as follows:
If $l = k$,

$$\frac{\partial G_l^1}{\partial x_k^j} = \begin{cases} \frac{\beta_l \theta^2}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = 1 \\ 0, \text{ for } i = 1,\ j < N \\ \frac{\theta}{1+\theta+\delta_l \theta^2}, \text{ for } i = 1,\ j = N \end{cases}$$

$$\frac{\partial G_l^i}{\partial x_k^j} = \begin{cases} \frac{\theta}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial x_k^j} + \frac{1+\theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial x_k^j}, \text{ for } 1 < j < i < N \\ \frac{\beta_l \theta^2}{1+\theta+\delta_l \theta^2} \sin(2(\bar{x}_l + b_l))), \text{ for } 1 < i = j < N \\ 0, (1 < i < j < N) \\ \frac{\theta}{1+\theta+\delta_l \theta^2} \frac{\partial F_l^{i-1}}{\partial x_k^N} + \frac{1+\theta}{1+\theta+\delta_l \theta^2} \frac{\partial G_l^{i-1}}{\partial x_k^N}, \text{ for } 1 < i < N,\ j = N \end{cases}$$

$$\frac{\partial G_l^N}{\partial x_k^j} = \begin{cases} \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{N-1}}{\partial x_k^j} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{N-1}}{\partial x_k^j}, \text{ for } i = N,\ 1 \le j < N \\ \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{N-1}}{\partial x_k^N} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{N-1}}{\partial x_k^N} + \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\sin(2(\bar{x}_l + b_l)), \text{ for } i = j = N \end{cases}$$

If $l \neq k$,

$$\frac{\partial G_l^i}{\partial x_k^j} = \begin{cases} \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l))\frac{\partial F_{l-1}^i}{\partial x_k^j}, \text{ for } l > k,\ 1 \le i \le N,\ 1 \le j \le N \\ 0, \text{ for } l < k,\ 1 \le i \le N,\ 1 \le j \le N \end{cases}$$

The expression of B_G

The Jacobian matrix $B_G \in \mathbb{R}^{N_{\mathrm{tot}} \times N_{\mathrm{tot}}}$ is given by

$$B_G = D_{\boldsymbol{y}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}).$$

Its the $(i \times l, j \times k)$-th element is obtained by $\frac{\partial G_l^i}{\partial y_k^j}$. This value is calculated as follows:
If $l = k$,

$$\frac{\partial G_l^1}{\partial y_k^j} = \begin{cases} 0, \text{ for } i = 1,\ 1 \le j < N \\ \frac{1+\theta}{1+\theta+\delta_l\theta^2}, \text{ for } i = 1,\ j = N \end{cases}$$

$$\frac{\partial G_l^i}{\partial y_k^j} = \begin{cases} 0, \text{ for } 1 < i \le N,\ 1 \le j < N \\ \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{i-1}}{\partial y_k^N} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{i-1}}{\partial y_k^N}, \text{ for } 1 < i \le N,\ j = N \end{cases}$$

If $l \neq k$,

$$\frac{\partial G_l^i}{\partial y_k^j} = \begin{cases} \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l))\frac{\partial F_{l-1}^i}{\partial y_k^j}, \text{ for } l > k,\ 1 \le i \le N,\ 1 \le j \le N \\ 0, \text{ for } l < k,\ 1 \le i \le N,\ 1 \le j \le N \end{cases}$$

The expression of C_G

The Jacobian matrix $C_G \in \mathbb{R}^{N_{\mathrm{tot}} \times N}$ is given by

$$C_G = D_{\boldsymbol{J}_{\mathrm{in}}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}).$$

Its the $(i \times l, j)$-th element is obtained by $\frac{\partial G_l^i}{\partial J_{\mathrm{in}}^j}$. This value is calculated as follows:
If $l = 1$,

$$\frac{\partial G_l^1}{\partial J_{\mathrm{in}}^j} = \begin{cases} \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = 1 \\ 0, \text{ for } i = 1,\ 1 < j \le N \end{cases}$$

$$\frac{\partial G_l^i}{\partial J_{\text{in}}^j} = \begin{cases} \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{i-1}}{\partial J_{\text{in}}^j} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{i-1}}{\partial J_{\text{in}}^j}, \text{ for } 1 \le j < i < N \\ \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l)), \text{ for } 1 < i = j < N \\ 0, \text{ for } 1 < i < j \le N \end{cases}$$

$$\frac{\partial G_l^N}{\partial J_{\text{in}}^j} = \begin{cases} \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{N-1}}{\partial J_{\text{in}}^j} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{N-1}}{\partial J_{\text{in}}^j}, \text{ for } i = N,\ 1 \le j < N \\ \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = N \end{cases}$$

If $l > 1$,

$$\frac{\partial G_l^i}{\partial J_{\text{in}}^j} = \begin{cases} \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l))\frac{\partial F_{l-1}^i}{\partial J_{\text{in}}^j}, \text{ for } 1 \le i \le N,\ 1 \le j \le N \end{cases}$$

The expression of D_G

The Jacobian matrix $D_F \in \mathbb{R}^{N_{\text{tot}} \times N_{\text{tot}}}$ is given by

$$D_G = D_{\Delta \boldsymbol{z}} G(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \boldsymbol{0}, \boldsymbol{0}).$$

Its the $(i \times l, j \times k)$-th element is obtained by $\frac{\partial G_l^i}{\partial \Delta z_k^j}$. This value is calculated as follows:
If $l = k$,

$$\frac{\partial G_l^1}{\partial \Delta z_k^j} = \begin{cases} \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } i = j = 1 \\ 0, \text{ for } i = 1,\ 1 < j \le N \end{cases}$$

$$\frac{\partial G_l^i}{\partial \Delta z_k^j} = \begin{cases} \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{i-1}}{\partial \Delta z_k^j} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{i-1}}{\partial \Delta z_k^j}, \text{ for } 1 \le j < i < N \\ \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2} \sin(2(\bar{x}_l + b_l)), \text{ for } 1 < i = j < N \\ 0, \text{ for } 1 < i < j \le N \end{cases}$$

$$\frac{\partial G_l^N}{\partial \Delta z_k^j} = \begin{cases} \frac{\theta}{1+\theta+\delta_l\theta^2}\frac{\partial F_l^{N-1}}{\partial \Delta z_k^j} + \frac{1+\theta}{1+\theta+\delta_l\theta^2}\frac{\partial G_l^{N-1}}{\partial \Delta z_k^j}, \text{ for } i = N,\ 1 \le j < N \\ \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2} sin(2(\bar{x}_l + b_l)), \text{ for } i = j = N \end{cases}$$

If $l \neq k$,

$$\frac{\partial G_l^i}{\partial \Delta z_k^j} = \begin{cases} \frac{\beta_l\theta^2}{1+\theta+\delta_l\theta^2}\kappa_l \sin(2(\bar{x}_l + b_l))\frac{\partial F_{l-1}^i}{\partial \Delta z_k^j}, \text{ for } l > k,\ 1 \le i \le N,\ 1 \le j \le N \\ 0, \text{ for } l < k,\ 1 \le i \le N,\ 1 \le j \le N \end{cases}$$

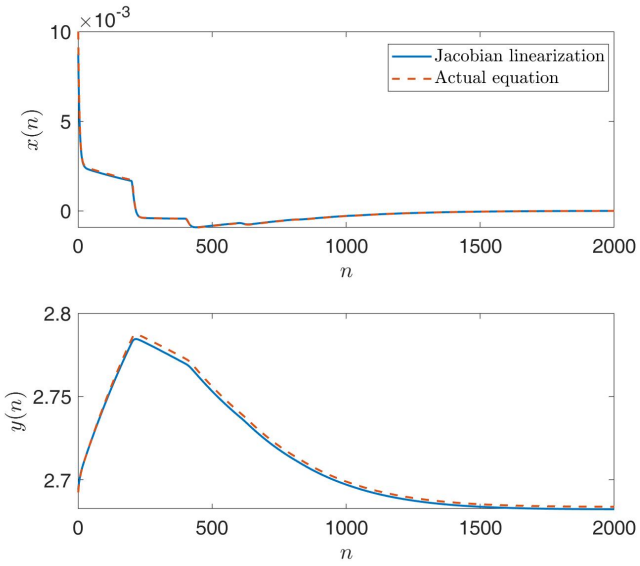


Figure 6.3: Comparison of the Jacobian linearization and Eq.5.4 computed by using Runge-Kutta method. In this case, we set $J(t) = 0$, $z(t) = 0$. The trajectory of a one-layer deep TDR based on coupled Ikeda delay system is govern by Eq.5.4 with parameter setting: $\tau = 40$, $\delta = 0.01$, $\beta = 0.68$, $\kappa = 1$ and $b = 0.2$.

In Figure 6.3, we plot the simulation results of the Jacobian linearization and Eq.5.4 computed by using Runge-Kutta method. The figure show that both the linear approximation and the actual simulation result overlap almost exactly near a stable equilibrium. It illustrates how the Jacobian linearization could be used to approximate Eq.5.4 near a stable equilibrium and can describe its behavior quite accurately.

E. Eigenvectors of the spatial Fisher memory matrix in deep TDRs

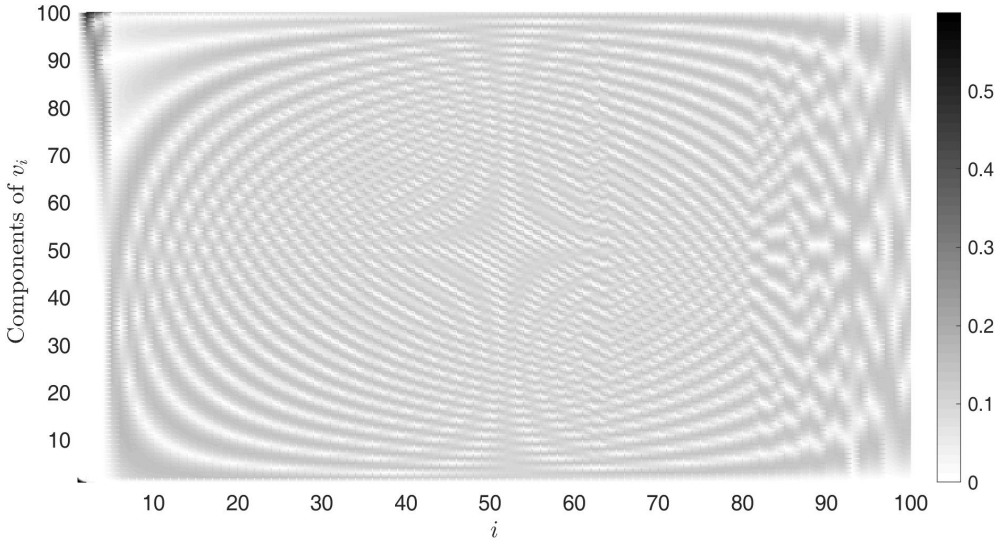


Figure 6.4: Heatmap of the eigenvectors of I^s of the deep TDR. The color of each pixel is determined by the absolute value of components of the input mask. The indices of the input masks are arranged in descending order according to the eigenvalues of I^s. The used parameters are given in Table 5.1.

References

[1] S. Stepney, S. Rasmussen, and M. Amos, *Computational Matter*. Springer, 2018.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[3] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *International Conference on Artificial Neural Networks*, Springer, 2006, pp. 632–640.

[4] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[5] H. Jaeger, "The" echo state" approach to analysing and training recurrent neural networks-with an erratum note'," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, Jan. 2001.

[6] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[7] L. Appeltant, M. Soriano, G. Van der Sande, *et al.*, "Information processing using a single dynamical node as complex system," *Nature communications*, vol. 2, p. 468, Sep. 2011. DOI: 10.1038/ncomms1476.

[8] L. Larger, M. C. Soriano, D. Brunner, *et al.*, "Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing," *Opt. Express*, vol. 20, no. 3, pp. 3241–3249, Jan. 2012. DOI: 10.1364/OE.20.003241.

[9] M. C. Soriano, S. Ortın, D. Brunner, *et al.*, "Optoelectronic reservoir computing: Tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[10] M. C. Soriano, S. Ortın, L. Keuninckx, *et al.*, "Delay-based reservoir computing: Noise effects in a combined analog and digital implementation," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 2, pp. 388–393, 2014.

[11] J. Nakayama, K. Kanno, and A. Uchida, "Laser dynamical reservoir computing with consistency: An approach of a chaos mask signal," *Optics express*, vol. 24, no. 8, pp. 8679–8692, 2016.

[12] D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger, "Tutorial: Photonic neural networks in delay systems," *Journal of Applied Physics*, vol. 124, no. 15, p. 152 004, 2018.

[13] B. Penkovsky, X. Porte, M. Jacquot, L. Larger, and D. Brunner, "Coupled nonlinear delay systems as deep convolutional neural networks," *Physical review letters*, vol. 123, no. 5, p. 054 101, 2019.

[14] M. Goldmann, F. Köster, K. Lüdge, and S. Yanchuk, "Deep time-delay reservoir computing: Dynamics and memory capacity," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 9, p. 093 124, 2020.

[15] O. Yilmaz, "Reservoir computing using cellular automata," *arXiv:1410.0162*, 2014.

[16] S. Nichele and M. S. Gundersen, "Reservoir computing using non-uniform binary cellular automata," *Complex Systems Publications, Inc.*, vol. 26, no. 3, pp. 225–245, 2017.

[17] H. Ando and H. Chang, "Road traffic reservoir computing," *arXiv preprint arXiv:1912.00554*, 2019.

[18] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *European conference on artificial life*, Springer, 2003, pp. 588–597.

[19] Z. Liu, J. Z. Kim, and D. S. Bassett, "Supervised chaotic source separation by a tank of water," *Chaos: An interdisciplinary journal of nonlinear science*, vol. 30, no. 2, p. 3, 2020.

[20] G. Dion, S. Mejaouri, and J. Sylvestre, "Reservoir computing with a single delay-coupled non-linear mechanical oscillator," *Journal of Applied Physics*, vol. 124, no. 15, p. 152 132, 2018. DOI: 10.1063/1.5038038.

[21] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Phys. Rev. X*, vol. 7, p. 011 015, 1 Feb. 2017. DOI: 10.1103/PhysRevX.7.011015.

[22] N. Semenova, X. Porte, L. Andreoli, M. Jacquot, L. Larger, and D. Brunner, "Fundamental aspects of noise in analog-hardware neural networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 10, p. 103 128, 2019.

[23] R. M. Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van der Sande, "Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics," *Optics express*, vol. 22, no. 7, pp. 8672–8686, 2014.

[24] H. Jaeger, "Short term memory in echo state networks. gmd-report 152," in *GMD-German National Research Institute for Computer Science,*, 2002.

[25] S. Ganguli, D. Huh, and H. Sompolinsky, "Memory traces in dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 105, no. 48, pp. 18 970–18 975, 2008.

[26] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004. DOI: 10.1126/science.1091277.

[27] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on Circuits and Systems*, vol. 32, no. 11, pp. 1150–1161, 1985. DOI: 10.1109/TCS.1985.1085649.

[28] Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 6, p. 061 104, 2018. DOI: 10.1063/1.5039508.

[29] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[30] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, PMLR, 2013, pp. 1310–1318.

[31] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 659–686.

[32] J. Zhao, C. Zhao, F. Zhang, G. Wu, and H. Wang, "Water quality prediction in the waste water treatment process based on ridge regression echo state network," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 435, 2018, p. 012 025.

[33] D. Sussillo and L. F. Abbott, "Generating coherent patterns of activity from chaotic neural networks," *Neuron*, vol. 63, no. 4, pp. 544–557, 2009.

[34] S. Haykin, *Adaptive Filter Theory*. Pearson, 2014, ISBN: 9780132671453.

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

[36] C. Gallicchio and A. Micheli, "Echo state property of deep reservoir computing networks," *Cognitive Computation*, vol. 9, no. 3, pp. 337–350, 2017.

[37] T. Lymburn, A. Khor, T. Stemler, D. C. Corrêa, M. Small, and T. Jüngling, "Consistency in echo-state networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 2, p. 023 118, 2019.

[38] T. Lymburn, D. M. Walker, M. Small, and T. Jüngling, "The reservoir's perspective on generalized synchronization," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 9, p. 093 133, 2019.

[39] N. Rulkov, M. Sushchik, L. Tsimring, and H. Abarbanel, "Generalized synchronization of chaos in directionally coupled chaotic systems," *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, vol. 51, pp. 980–994, Mar. 1995. DOI: 10.1103/PhysRevE.51.980.

[40] H. Abarbanel, N. Rulkov, and M. Sushchik, "Generalized synchronization of chaos: The auxiliary system approach," *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, vol. 53, pp. 4528–4535, Jun. 1996. DOI: 10.1103/PhysRevE.53.4528.

[41] K. Pyragas, "Weak and strong synchronization of chaos," *AIP Conference Proceedings*, vol. 411, no. 1, pp. 63–68, 1997. DOI: 10.1063/1.54216.

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

[43] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," *Neural networks*, vol. 20, no. 3, pp. 323–334, 2007.

[44] T. L. Carroll, "Do reservoir computers work best at the edge of chaos?" *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 12, p. 121 109, 2020.

[45] J. Hertzberg, H. Jaeger, and F. Schönherr, "Learning to ground fact symbols in behavior-based robots," in *ECAI*, vol. 2, 2002, pp. 708–712.

[46] H. Jaeger, "Reservoir riddles: Suggestions for echo state network research," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, IEEE, vol. 3, 2005, pp. 1460–1462.

[47] P. G. Plöger, A. Arghir, T. Günther, and R. Hosseiny, "Echo state networks for mobile robot modeling and control," in *Robot Soccer World Cup*, Springer, 2003, pp. 157–168.

[48] M. Salmen and P. G. Ploger, "Echo state networks used for motor control," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, IEEE, 2005, pp. 1953–1958.

[49] M. D. Skowronski and J. G. Harris, "Minimum mean squared error time series classification using an echo state network prediction model," in *2006 IEEE International Symposium on Circuits and Systems*, IEEE, 2006, 4–pp.

[50] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," *Advances in neural information processing systems*, vol. 15, pp. 609–616, 2002.

[51] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 12, p. 121 102, 2017.

[52] K. Hamedani, Z. Zhou, K. Bai, and L. Liu, "The novel applications of deep reservoir computing in cyber-security and wireless communication," in *Intelligent System and Computing*, IntechOpen, 2019.

[53] B. Oksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[54] B. Widrow and I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511754661.

[55] L. Appeltant, G. Van der Sande, J. Danckaert, and I. Fischer, "Constructing optimized binary masks for reservoir computing with delay systems," *Scientific reports*, vol. 4, no. 1, pp. 1–5, 2014.

[56] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[57] J. Duchi, "Derivations for linear algebra and optimization," *Berkeley, California*, vol. 3, no. 1, pp. 2325–5870, 2007.

[58] L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega, "Optimal nonlinear information processing capacity in delay-based reservoir computers," *Scientific reports*, vol. 5, no. 1, pp. 1–11, 2015.

[59] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[60] J. D. Farmer, "Chaotic attractors of an infinite-dimensional dynamical system," *Physica D: Nonlinear Phenomena*, vol. 4, no. 3, pp. 366–393, 1982.

[61] J. Wei and D. Fan, "Hopf bifurcation analysis in a mackey–glass system," *International Journal of Bifurcation and Chaos*, vol. 17, no. 06, pp. 2149–2157, 2007.

[62] J. Boedecker, O. Obst, J. T. Lizier, N. M. Mayer, and M. Asada, "Information processing in echo state networks at the edge of chaos," *Theory in Biosciences*, vol. 131, no. 3, pp. 205–213, 2012.

[63] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, IEEE, vol. 4, 1995, pp. 1942–1948.

[64] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, no. 1, pp. 1–7, 2012.

[65] M. Inubushi and K. Yoshimura, "Reservoir computing beyond memory-nonlinearity trade-off," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[66] K. Ikeda, "Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system," *Optics communications*, vol. 30, no. 2, pp. 257–261, 1979.

[67] K. Ikeda and K. Matsumoto, "High-dimensional chaotic behavior in systems with time-delayed feedback," *Physica D: Nonlinear Phenomena*, vol. 29, no. 1-2, pp. 223–235, 1987.

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