UNIVERSITY OF OSNABRÜCK

DOCTORAL DISSERTATION

---

# HOMEOSTATIC PLASTICITY IN INPUT-DRIVEN DYNAMICAL SYSTEMS

HAZEM TOUTOUNJI

*A thesis submitted in fulfilment of the requirements for the degree of Dr. rer. nat.*

THE NEUROINFORMATICS DEPARTMENT
INSTITUTE OF COGNITIVE SCIENCE (IKW)

JUNE 2014

Hazem Toutounji: Homeostatic Plasticity in
Input-Driven Dynamical Systems, June 2014

*Dedicated to those who fear nothing*
*to those who hope for nothing*
*to those who are free, in Syria and elsewhere*

— (with gratitude to Kazantzakis)

# ABSTRACT

The degree by which a species can adapt to the demands of its changing environment defines how well it can exploit the resources of new ecological niches. Since the nervous system is the seat of an organism's behavior, studying adaptation starts from there. The nervous system adapts through neuronal plasticity, which may be considered as the brain's reaction to environmental perturbations. In a natural setting, these perturbations are always changing. As such, a full understanding of how the brain functions requires studying neuronal plasticity under temporally varying stimulation conditions, i. e., studying the role of plasticity in carrying out spatiotemporal computations. It is only then that we can fully benefit from the full potential of neural information processing to build powerful brain-inspired adaptive technologies.

Here, we focus on homeostatic plasticity, where certain properties of the neural machinery are regulated so that they remain within a functionally and metabolically desirable range. Our main goal is to illustrate how homeostatic plasticity interacting with associative mechanisms is functionally relevant for spatiotemporal computations.

The thesis consists of three studies that share two features: (1) homeostatic and synaptic plasticity act on a dynamical system such as a recurrent neural network. (2) The dynamical system is nonautonomous, that is, it is subject to temporally varying stimulation.

In the first study, we develop a rigorous theory of spatiotemporal representations and computations, and the role of plasticity. Within the developed theory, we show that homeostatic plasticity increases the capacity of the network to encode spatiotemporal patterns, and that synaptic plasticity associates these patterns to network states.

The second study applies the insights from the first study to the single node delay-coupled reservoir computing architecture, or DCR. The DCR's activity is sampled at several computational units. We derive a homeostatic plasticity rule acting on these units. We analytically show that the rule balances between the two necessary processes for spatiotemporal computations identified in the first study. As a result, we show that the computational power of the DCR significantly increases.

The third study considers minimal neural control of robots. We show that recurrent neural control with homeostatic synaptic dynamics endows the robots with memory. We show through demonstrations that this memory is necessary for generating behaviors like obstacle-avoidance of a wheel-driven robot and stable hexapod locomotion.

*When the God, whichever one of the gods, had divided the substance of Chaos and ordered it thus in its different constituent members . . .*

— Ovid's "*Metamorphoses*" (8 CE) [88]

## ACKNOWLEDGEMENTS

A pantheon of friends and colleagues contributed to the metamorphosis of this thesis, delivering it from the womb of chaos to its current form.

Gordon Pipa, your guidance through the last few years and your enthusiasm for discovery wrote my scientific coming-of-age story. Thank you for the freedom to explore that you bestowed on me, and for all the research discussions that brought me insight and widened my horizon. On a more personal note, I dearly appreciate your friendship, and the many times you stood by and supported me through personal crises. In that, my thanks also extend to your lovely family.

Frank Pasemann, you taught me the great value of searching for basic principles. Though I only had the pleasure of working with you for a rather short time, your methodology of conducting research will accompany me for the rest of my scientific career.

During the last ten years, I had the fortune to be mentored by inspiring minds. Chadi Aljundi, Eberhard Engel, Peter Kloeden, Constantin Rothkopf, and Jochen Triesch, you have earned my lasting gratitude.

Regards are also due to the members of the Institute of Cognitive Science at the University of Osnabrück. Particularly, to those in the Neuroinformatics and Neurocybernetics Groups for their friendly and stimulating company, Anna Rushing-Jungeilges for her administrative support to the Neuroinformatics Group and for greatly improving the quality of my writing, Beate Eibisch for dealing with all the official matters, and Thorsten Kundoch and Martin Möhrmann for administrating the computing infrastructure.

Johannes Schumacher, my eternal research buddy, companion, and officemate. Thank you for the wonderful partnership. Marta Castellano, you are the witness to my growth, as a person and as a researcher. Thank you for the laughs and the cries.

Thanks are due to those ever persisting friendships. Martin Aher for all the banter, chess, Frisbee, and wine, and for not liking chocolate. Sascha Alexeyenko for your constant concern, and for the art, films, and books. Mehiar Alkhashroum for sharing a dream. Khalil

Alriz for being my compass towards beautiful things. Jonas Eberlein for overseeing every big step I took in Germany. Sascha Fink for starting his difficult questions with a jolly statement. Felipe Gerhard for the fireworks I saw in your eyes, telling me of your first infatuation with the brain. Your passion is mine, Felipe. Kristina Gorodetski for introducing me to the art of round robin vodka shots toast making. Simon Harst for the enjoyable whisky tasting nights. Maria Hedblom for the tight hugs. Lisa Holle for heart-to-heart profound discussions till the first hours of the morning. Frank Jäkel for teaching me never to compromise in science. Ian Kavanagh for always stretching my Arabic speaking muscles. Clemens Korndörfer for the music and existential dialogues. Gabriella Lapesa for the liquorice and your leaning against loose furniture. Johannes Leugering for never stopping talking. It would have been my loss if you ever did. Mikko Määttä for showing me the Finnish warmth of heart. Robert Muil for the many small things we share. Lina Razzouk for being always challenging. Constantin Rothkopf for the science advices and for finding humor in my reading Freud. Cristina Savin for being my rising scientist role model.

My parents, Kheirat Salem and Mutaz Toutounji, you have been my closest friends as far as my memory can take me. Mom, thank you for tolerating my constant absentmindedness, and for pretending not to worry when I sleep where snakes might live. Dad, thank you for nursing me with mathematics, and for all the Lego.

Anna-Birga, in the words of Lawrence Durrell: "*There are only three things to be done with a woman. You can love her, suffer for her, or turn her into literature*". I would suffer for you, but your kind demeanor, at times refreshingly childish, leaves no place for suffering. I do love you, Anna, and as through your lighthearted spirit I find the inspiration to write the following words, I succeed in turning you into literature. Thank you.

This work was supported by the State of Lower Saxony, Germany via the University of Osnabrück, by the European project PHOCUS in the Framework 'Information and Communication Technologies' (FP7-ICT-2009-C/Proposal Nr. 240763), and by the German Research Foundation (DFG) priority program 1527.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACRONYMS

BCM **B**ienenstock-**C**ooper-**M**unro

DCR single node **d**elay-**c**oupled **r**eservoir

IP **i**ntrinsic **p**lasticity

kWTA **k**-**w**inner-**t**ake-**a**ll

NARMA **n**onlinear **a**uto**r**egressive **m**oving **a**verage

nrmse **n**ormalized **r**oot-**m**ean-**s**quare error

RC **r**eservoir **c**omputing

RNN **r**ecurrent-**n**eural **n**etwork

SR **s**elf-**r**egulating

STDP **s**pike-**t**iming-**d**ependent synaptic **p**lasticity

v- **v**irtual

Part I

# INTRODUCTION

We outline the history of the concept of homeostasis and the different models of homeostatic plasticity, we specify the goals of this thesis, and we summarize the different papers comprising its main body.

# 1 INTRODUCTION AND PAPERS' SUMMARY

*She set to work very carefully, nibbling first at one and then at the other, and growing sometimes taller and sometimes shorter, until she had succeeded in bringing herself down to her usual height.*

— Lewis Carroll's "*Alice's Adventures in Wonderland*" (1865) [21]

It seems that natural selection either repeatedly rediscovers useful processes, or is capable of adapting already found successful solutions to new situations, *homeostasis* being a case in point. Homeostasis is the process of bringing a system towards the safety of stability. Through a brief account of this concept's history, we show that homeostatic processes were beneficially adapted through eons of evolution to stabilize inherited vital conditions, to regulate learning behavior, and to modulate the activity of single neurons. The latter process is known as *homeostatic plasticity*. As we later review, besides modulating a neuron's activity for metabolic gain, homeostatic plasticity is also *computationally relevant*. This means that homeostatic plasticity has a functional role in the *information processing engine* that is the neuron. This also entails that homeostatic plasticity is a useful tool for brain-inspired adaptive machines.

*We follow Day [29] in referring to* ***homeostasis*** *as both the* ***property*** *of being stable and the* ***process*** *of achieving stability. The latter is also called* ***allostasis*** *[124].*

This thesis builds upon these ideas. In regard to understanding neural information processing, we extend the function of homeostatic plasticity to a more general form of neural computations. In reality, the input to the brain is never static. Therefore, the neural responses should be context dependent, i. e., neurons and neural circuitry should carry out *spatiotemporal computations*. After demonstrating the importance of homeostatic plasticity to these neural computations, we turn to brain-inspired adaptive machines. We argue that homeostatic plasticity as a solution to neural computational demands is sufficiently general. We show that the principal understanding of its role in neural spatiotemporal computations extends to robotic neural control and a novel computing architecture that consists of a single nonlinear node with delayed feedback.

## 1.1 HISTORICAL BACKGROUND

Walter B. Cannon coined the term homeostasis in 1926[1]. He defined it in the book "*the Wisdom of the Body*" (1932) as the equilibria that

1 Later essays by Cannon and others, including his book "*the Wisdom of the Body*" [20], cited this date as the origin of the term. The title of the 1926 essay is identified in many resources, but we have not succeeded so far in locating the text.

an organism aims to achieve in response to disturbances both within and without the organism's body. Homeostasis, Cannon states, is "*a condition which may vary, but which is relatively constant*" [20]. Cannon also references earlier accounts of these self-regulatory mechanisms, rightly assigning the credit for their first conception to Claude Bernard [11]. Remarkably suitable to the purpose of the current thesis, is Charles Richet's 1900 declaration which we requote in length:

> The living being is stable. It maintains its stability only if it is *excitable* and capable of *modifying itself* according to *external stimuli* and adjusting its response to the stimulation. In a sense it is *stable because it is modifiable*—the slight instability is the necessary condition for the true stability of the organism [105] [my italics].

Richet maintains that environmental perturbations are the primary driving force behind adaptation, which is a central theme of this thesis. Cannon also envisions how these "*general principles for the regulation of steady states*", here acquired from the study of biology, "*would be suggestive for other kinds of organization—even industrial—which suffer from perturbations*" [20]. As this thesis is also concerned with applying principles of biological adaptability to brain-inspired systems, Cannon's vision cannot be overemphasized.

Despite their farsightedness, Cannon's studies are only concerned with *innate* self-regulatory processes, such as classical conditioned responses, thermoregulation[2], and the homeostatic regulation of blood glucose and salts. He explicitly states that homeostasis is solely the responsibility of the *autonomic nervous system*, and that the "*freedom of the activity of the higher levels of the nervous system*" is contingent on the proper function of these automatic regulations. While he declares that these processes are *learned*, he means it in the sense that "*homeostasis is the product of an evolutionary process*" [20]. Consequently, intelligence becomes a property of natural selection that distinguishes between species, rather than being a property of a developing, experiencing individual.

It is only by the advent of *cybernetics* in the 1950s that the study of homeostasis shifts away from inherited processes. In his seminal book "*Design for a Brain*", William Ross Ashby explicitly states that "*with the reflex type of behavior we shall not be concerned*", but "*with behavior that is not inborn but learned*", a behavior that "*is a product of the cerebral cortex, and is modified by the organism's individual experiences*"[3] [8]. While Cannon's case studies considers *classical conditioning*, Ashby investigates the mechanisms of *operant conditioning*, where homeostasis is maintained through adaptive behavior. Richet's declaration that "*the living*

2 the homeostatic regulation of a stable body temperature

3 for consistency with the rest of the text, when quoting Ashby, we exchanged his British spelling "*behaviour*" by the American spelling.

*being is stable*" remains intact. Ashby insists that "*adaptive behavior is equivalent to the behavior of a stable system*", as its goal is to maintain the system's "*essential variables within physiological limits*" [8].

Ashby establishes a rigorous framework for understanding adaptive behavior. He considers a two-part dynamical system that consists of an environment and an organism with two modes of interaction. The first is through a sensorimotor feedback loop, the dynamics of which expresses already established behaviors. The second interaction is affected by the variables essential for the organism's survival. It happens on a *slower time scale* and is active when the essential variables pass outside their functional limits, due to *external stimuli*. This, in turn, triggers changes of value in what Ashby terms *step mechanisms*, which "*determine how* [the organism] *shall react to the environment*". According to Ashby, learning progresses by the reconfiguration of the step mechanisms' values until homeostasis is reached. In that sense, the property of homeostasis is the goal of learning, while the process of homeostasis is the means for learning [8].

The first detailed account of homeostatic regulation at the neural level comes in the 1980s, with the development of the *BCM theory* [12] (see Section 1.2.1). The BCM theory offers a solution to the problem of destabilization of synaptic efficacies, resulting from constant potentiation by the pure form of Hebb's postulate [47]. By 2000, ample empirical evidence has accumulated in support of homeostatic mechanisms that regulate the spiking activity of neurons. These mechanisms are shown to be common across species, and are categorized under two broad classes[4]: (1) *Intrinsic homeostatic plasticity*, which is the family of mechanisms that regulate a neuron's excitability by adapting properties at the soma (reviewed by Zhang and Linden [145]). (2) *Synaptic homeostatic plasticity*, which encompasses mechanisms that regulate neuronal excitability by modulating the overall drive arriving at the neuron from its afferents (reviewed by Turrigiano and Nelson [133]).

Intrinsic homeostatic plasticity requires a molecular marker for the neuron to measure its own level of activity. According to the level of neural activity, the cellular machinery reorganizes the distribution of relevant somatic ion channels. This molecular marker is primarily thought to be the intracellular concentration of $Ca^{2+}$ ions, which correlates with the neuron's level of activity. A possible reason for this correlation is that membrane depolarization leading to the generation of spikes, which results in $Ca^{2+}$ influx [145].

Regarding the underlining cellular mechanisms of synaptic homeostatic plasticity, much empirical evidence points to changes in the distribution of postsynaptic ion channels. Other candidates are changes

4 The homeostatic plasticity model in Paper I falls under the first class, while that of Paper III falls under the second. The model in Paper II is hard to assign to a particular class, given the unconventional computing architecture it acts on.

in the amount of presynaptic neurotransmitter release or reuptake, or the number of synapses [133].

These empirical discoveries were also accompanied by a plethora of models and theoretical studies, discussing the functional implications of homeostatic plasticity on neuronal stability and computations, which we review next.

## 1.2 MODELS OF HOMEOSTATIC PLASTICITY

In this thesis, we build models of homeostatic plasticity for spatiotemporal computations. In order to do so, we build upon the body of knowledge from previous models of homeostatic plasticity. We classify these models in three broad and not mutually exclusive categories.

### 1.2.1 *The BCM Theory*

Bienenstock, Cooper, and Munro introduced homeostasis to synaptic plasticity as a way to assure afferents' stability [12]. Given the activities of the presynaptic and a postsynaptic neurons, $x_j$ and $x_i$ respectively, pure Hebbian potentiation [47] of the form

$$\frac{dw_{ij}}{dt} = x_j \cdot x_i \tag{1}$$

leads to unbounded increase of the synaptic efficacy $w_{ij}$ . To solve this instability, Bienenstock et al. included a superlinear dependence of the postsynaptic neuron's activity [12]:

$$\frac{dw_{ij}}{dt} = x_j \cdot x_i(x_i - \bar{x}_i^p) - \epsilon w_{ij} \tag{2}$$

where $\bar{x}_i$ is the average activity of the postsynaptic neuron, $p > 1$ is an integer that defines the superlinear dependence, and $\epsilon$ is a decay parameter. Other plasticity mechanisms of similar forms also belong to the BCM theory[5] and extend its application [25]. In these mechanisms, the term $\bar{x}_i^p = \theta_i$ corresponds to a sliding threshold separating regimes of potentiation and depression, thereby stabilizing $w_{ij}$. An intuitive interpretation of the stability resulting from the sliding threshold goes as follows. When afferents are potentiated, the postsynaptic neuron's average activity rises, as does the threshold $\theta_i$, which makes potentiation harder. At some point, the threshold $\theta_i$ is crossed, resulting in synaptic depression. By adjusting the threshold

5 Models such as von der Malsburg's synaptic normalization rule [136] and Oja's principal component analysis rule [87] also assure afferents' stability, but do not belong to the BCM theory.

that separates potentiation and depression, the BCM mechanism also homeostatically regulates the *excitability* of the post synaptic neuron.

In a recent study, Clopath et al. proposed a model of spike-timing-dependent synaptic plasticity (STDP) with a homeostatically modulated sliding threshold [24]. While homeostasis in this model regulates the balance between potentiation and depression, it was still necessary to enforce hard boundaries on synaptic efficacies, in order to prevent weight's runaway or the undesired change in its sign.

### 1.2.2 *Maximum Entropy Models*

In the BCM theory, homeostasis in the form of a sliding threshold is necessary for stability. It does not make a statement on the role of the sliding threshold in neural encoding. On the other hand, Stemmler and Koch argued that as a communication channel, a neuron attempts to optimize its encoding and transmission of the incoming sensory streams [123]. In this spirit, they derived an intrinsic plasticity (IP) mechanism that homeostatically acts on the ion channel conductances of Hodgkin-Huxley-like neuronal model [49]. The plasticity rule maximizes the mutual information between the neuron's input and output, therefore resulting in an exponential distribution of the neuron's output firing rate.

The exponential distribution is the *maximum entropy* probability distribution for a given mean firing rate. This observation led Triesch to derive an intrinsic plasticity rule that directly minimizes the *Kullback-Leibler divergence* between the output of a rate model sigmoidal neuron and a target exponential firing rate distribution with a given mean [132] (the autonomous dynamics of a RNN were analyzed by Marković and Gros [74]). Interestingly, when interacting with a BCM-like synaptic plasticity rule with a *fixed* threshold, the afferents to the model neuron were stabilized, which suggests that the homeostatic nature of intrinsic plasticity compensated for the role of the sliding threshold in the BCM theory. Similarly, maximum entropy homeostatic plasticity models were derived for stochastically firing spiking neurons, either as a BCM-like STDP rule [131], or in the form of intrinsic plasticity [110].

### 1.2.3 *Normalization Models*

These models regulate a neuron's excitability or normalize its afferents by a direct estimate of the difference between a certain quantity the neuron expresses and the target value this quantity aims to achieve[6]. For instance, in a RNN model of spatial working memory

6 The homeostatic plasticity models used in Paper I and Paper III fall under the same category.

[104], synaptic conductances $g(\kappa)$ of a cell that is selective to a spatial cue $\kappa$ are normalized by

$$\tau_j \frac{dg(\kappa)}{dt} = -g(\kappa)(r(\kappa) - r_{tg}(\kappa)), \tag{3}$$

where $r$ is the instantaneous firing rate and $r_{tg}$ is the target firing rate.

Several studies considered stability criteria of synaptic efficacies that may emerge from the interaction between similar models and synaptic plasticity in RNNs. For instance, Tetzlaff et al. derived a general form that describes a family of synaptic adaptation mechanisms, and identified a subset of this family where weights are stable [126]. On the other hand, Remme and Wadman discussed the necessity of balancing excitability modulation between inhibitory and excitatory populations in RNNs that are subject to other adaption mechanisms [100].

Normalization models of intrinsic plasticity on spiking RNNs were adapted in [67, 68, 146]. The rules governing these mechanisms have the form

$$\theta_i(t+1) = \theta_i(t) + \eta_{ip}(x_i(t+1) - r_{tg}) \tag{4}$$

where $\theta_i$ is the firing threshold of neuron $i$, $\eta_{ip}$ is a learning rate, and $x_i$ is the binary activity of neuron $i$. Two of these models were the first to study spatiotemporal computations in plastic RNNs within the framework of reservoir computing [67, 68]. Here, we extend the analysis of these models by developing a dynamic theory of the effects of plasticity *within* the RNN. We arrive at general principles that allow us to apply the predictions of this theory, originally concerned with biological neural dynamics, to biologically-inspired computing architectures. The specific objectives of this thesis are presented in the next section.

## 1.3 THESIS' OBJECTIVES

The main aim of this thesis is to demonstrate that, besides its regulatory role, homeostatic plasticity is a necessary information processing device for neural and neuro-inspired spatiotemporal computations. More specifically, the thesis has the following objectives:

1. Developing a rigorous representation theory of spatiotemporal computations. This theory allows us to identify the computational roles of different plasticity mechanisms, including homeostatic plasticity, acting in recurrent cortical networks.

2. Demonstrating the generalizability of the identified computational roles by applying them to biologically-inspired computational architectures.

3. Arguing for controlled bifurcations as a useful computational tool, which is shaped by the different plasticity mechanisms.

4. Suggesting a role for homeostatic plasticity in generating behavior in embodied agents, acting in the sensorimotor loop.

We achieve these objectives by conducting three studies, which are organized in Paper I, Paper II, and Paper III. More specifically, Paper I deals with points 1 and 3, Paper II with point 2, and Paper III with points 3 and 4. The following section summarizes the three studies.

## 1.4 PAPERS' SUMMARY

The thesis is organized in three papers, two of which underwent peer-review and are published in online open access journals. The third paper (Paper II) is currently under review.

### 1.4.1 *Computations in an Excitable and Plastic Brain*

Paper I studies how recurrent neural networks learn to perform spatiotemporal computations through the interaction of STDP and homeostatic IP. The paper introduces nonautonomous dynamical systems for the study of neuronal plasticity under varying stimulation (see Chapter 2).

In this paper, we first show by training linear readouts that RNNs trained by the two forms of plasticity outperform those that are random or trained by a single plasticity mechanism. This test is intended as a sanity check that the interaction of STDP and IP does enhance reconstructing past inputs, predicting future ones, and carrying out nonlinear computations. These are the kind of operations we term *spatiotemporal computations*. We then test the effects of plasticity on the formation of the neural code by measuring network state entropy and its mutual information with input sequences. This information-theoretical analysis shows that STDP learns the associations of network states to input sequences, while IP increases the network's entropy, and thus its encoding potential.

The second part of the paper is dedicated to the reconstruction of a qualitative geometric theory that explains the quantitative assessments of spatiotemporal computations. In its current form, this theory is restricted to discrete-time RNNs. We start by proving that the model RNN is in fact a nonautonomous dynamical system, particularly, an *input-driven* dynamical system. We then proceed to identify

its dynamic regimes, i. e., the *basins of attraction* at which the system dynamics may unfold, depending on the initial conditions.

Next, we introduce a notion of representation within the mathematical framework. *Volumes of representation* are encoding entities within a basin of attraction of input sequences, or functions over input sequences. These entities may be sampled, approximated, and visualized since they correspond to the transient dynamics of a RNN. This allows us to disentangle the role of different plasticity mechanisms according to their respective effects on the sizes of these volumes, and the geometric relations between them. By exploiting the *volumes' inclusion property*, we also construct a hierarchy of representations that allows the study of multiple temporal scales of the input.

We proceed to identify the attractors within the phase space, which requires defining *nonautonomous attractors* as *nonautonomous sets*, i. e., sets whose elements are tuples of a point in time and a network state. In other words, a nonautonomous attractor is a *moving target* which the network's dynamics tends to approach. We also introduce a particular nonautonomous set that may be considered the *moving source* of the dynamics in which the transient activity of the network unfolds, as it follows the moving attractor. We call this special set a *perturbation set*.

Given these concepts, we prove an important result: According to the definition of a nonautonomous attractor, dynamics may converge to the attractor only asymptotically, which imposes constraints on a full identification of a RNN's dynamics. We rigorously prove, however, that within some basin of attraction, the perturbation set (which can be approximately defined) certainly contains the nonautonomous attractor of this basin. Ergo, studying the geometry of a perturbation set indicates the geometry of its nonautonomous attractor.

With the aid of this framework we show that IP alone leads to representations that are *redundant* but highly overlapping. On the other hand, STDP alone leads to *input-insensitive* dynamics that captures the basic temporal structure of the input. Together, these plasticity mechanisms lead to two dynamic regimes: an input-insensitive regime, and an input-sensitive regime. representations are both redundant and highly *separate* in the input-sensitive regime, which are the two properties necessary for spatiotemporal computations.

The final part of the paper elaborates on how plastic RNNs react to network noise. We demonstrate that both redundancy and separability resulting from STDP and IP lead to noise-robustness. Separability allows for a margin of noise between different volumes of representation, while redundancy provides alternative representations for similar input sequences. We also show noise as a mechanism for pushing the dynamics outside the input-insensitive regime, playing in this case a beneficial computational role.

*Reference*

Parts of this paper were presented in the following events: *Osnabrück Computational Cognition Alliance Meeting* on *"The Brain as an Information Processing System"* (June 4–6, 2012, Osnabrück, Germany), and on *"The Brain as a Probabilistic Inference Engine"* (May 7–9, 2014, Osnabrück, Germany), *Nonlinear Dynamics Summer School* (August 24–30, 2012). Another presentation is within the *NeuroBridges Workshop* (July 27–29, 2014, Göttingen, Germany).

This paper was published after a peer-review process in *PLoS Computational Biology*, and is found under the following reference:

[130] **Toutounji, H.**, and Pipa, G. (2014). Spatiotemporal computations of an excitable and plastic brain: neuronal plasticity leads to noise-robust and noise-constructive computations. *PLoS Comput. Biol.* 10:e1003512. doi: 10.1371/journal.pcbi.1003512

### 1.4.2 *Homeostatic Plasticity for Delay-Coupled Reservoir*

Paper II demonstrates the generality of the computational principles of plasticity found in Paper I, by showing their applicability to a *biologically-inspired* computational architecture (see Chapter 3).

The paper centers around enhancing spatiotemporal computations in a novel *reservoir computing* architecture that consists mainly of two elements: a single nonlinear node and a delay line that couples the nonlinear node to itself. Given that this architecture is modeled by a delay differential equation, its phase space is a function space that is *infinite dimensional*. Input is injected into the reservoir by multiplexing it in time after linearly transforming it with a binary mask that spans the full delay. This transformation prevents the reservoir's activity from converging to its fixed point, and losing all history dependence necessary for spatiotemporal computations. Computations are carried out by linear readouts at certain sampling points of the reservoir's activity. The positioning of the these sampling points on the delay line is crucial, regarding the computational power of the delay-coupled reservoir, or DCR.

Similar to the neural model of Paper I, the DCR is an input-driven dynamical system. In Paper I, we showed that redundancy and separability are the result of the interaction of STDP and IP, and that these two properties are computationally necessary ingredients in input-driven RNNs. Similarly, we analytically derive a single plasticity mechanism that homeostatically balances these two properties in the DCR's sampling points. The plasticity mechanism consists of two opposing terms. The first term increases separability of a sampling point's activity by adjusting the point's position towards having a higher activity slope. The second term has the inverse effect of the first. We analytically show that the second term increases the

sampling point's entropy, thus, its redundancy. We demonstrate the significant increase of the computational power of the DCR after applying plasticity. We also relate the two opposing effects of the plasticity mechanism to synaptic facilitation and depression, which further relates the DCR to biologically-inspired phenomena.

*Reference*

Parts of this paper were presented in the *International Symposium on Nonlinear Theory and Its Applications* (October 22–26, 2012, Palma de Mallorca, Spain).

This Paper is currently under peer-review. For the time being, it may be referenced by:

**Toutounji, H.**, Schumacher, J., and Pipa, G. (submitted). *Homeostatic plasticity for single node delay-coupled reservoir computing*.

### 1.4.3 *Behavior Control by Self-Regulating Neurons*

Paper III utilizes recurrent networks of neurons with homeostatic properties for controlling the behavior of embodied agents (see Chapter 4).

This paper introduces a form of synaptic homeostatic plasticity. A hyperbolic tangent neuron is augmented by synaptic dynamics that homeostatically adjusts the neuron's activity towards maximum nonlinearity. Given the antisymmetry of the sigmoidal nonlinearity, two target activity states exist. Under certain connectivity conditions, this *self-regulating neuron* (SR-neuron) becomes *bistable*. Under varying stimulation, bistability endows the SR-neuron with *short-term memory*. This memory makes the SR-neuron suitable for controlling embodied agents.

The SR-neuron model is an input-driven 3-dimensional dynamical system. One dimension corresponds to the neuron's activity, while the other two dimensions correspond to the neuron's receptor and transmitter strength, which together control the synaptic dynamics. The receptor regulates the incoming drive to the neuron, while the transmitter communicates the neuron's activity to its targets.

The first part of the paper concerns the dynamics of a single SR-neuron with and without self-connection. The dynamics is identified by bifurcation analysis while varying *bias* and *input* to the neuron. Stability conditions are found using linear stability analysis and comparing the SR-neuron to the familiar dynamics of a hyperbolic tangent neuron without synaptic dynamics. These analyses reveal the bistability of SR-neurons with positive self-connection, which allows it to show *hysteretic* short-term memory and *reflex oscillations*. The neuron is also capable of generating periodic dynamics when the self-connection is negative.

The last part of the paper demonstrates how the reflex oscillations and short-term memory, both resulting from homeostatic synaptic dynamics, can be used to generate useful behavior. SR-neural control is shown to be able to synchronize forced pendula, control a wheel-driven robot in obstacle-ridden environments, and achieve forward locomotion of a hexapod robot with 18 degrees of freedom. More precisely, reflex oscillations are exploited for generating the swing and stance phases of the pendula and the hexapod's joints, as well as short-term memory for avoiding difficult obstacle conditions.

*Reference*

A similar model to the one presented in this paper was proposed by Zahedi and Pasemann [143], while the same model was first introduced by Pasemann [94].

Paper I was published after a peer-review process in *Frontiers in Neurorobotics*, and is found under the following reference:

[129] **Toutounji, H.**, and Pasemann, F. (2014). Behavior control in the sensorimotor loop with short-term synaptic dynamics induced by self-regulating neurons. *Front. Neurorobot.* 8:19. doi: 10.3389/fnbot.2014. 00019

# Part II

## PAPERS

We present three studies on the role of homeostatic plasticity in information processing in input-driven dynamical systems.

# 2

## PAPER I | SPATIOTEMPORAL COMPUTATIONS OF AN EXCITABLE AND PLASTIC BRAIN: NEURONAL PLASTICITY LEADS TO NOISE-ROBUST AND NOISE-CONSTRUCTIVE COMPUTATIONS

ABSTRACT It is a long-established fact that neuronal plasticity occupies the central role in generating neural function and computation. Nevertheless, no unifying account exists of how neurons in a recurrent cortical network learn to compute on temporally and spatially extended stimuli. However, these stimuli constitute the norm, rather than the exception, of the brain's input. Here, we introduce a geometric theory of learning spatiotemporal computations through neuronal plasticity. To that end, we rigorously formulate the problem of neural representations as a relation in space between stimulus-induced neural activity and the asymptotic dynamics of excitable cortical networks. Backed up by computer simulations and numerical analysis, we show that two canonical and widely-spread forms of neuronal plasticity, that is, spike-timing-dependent synaptic plasticity, and intrinsic plasticity, are both necessary for creating neural representations, such that these computations become realizable. Interestingly, the effects of these forms of plasticity on the emerging neural code relate to properties necessary for both combating and utilizing noise. The neural dynamics also exhibits features of the most likely stimulus in the network's spontaneous activity. These properties of the spatiotemporal neural code resulting from plasticity, having their grounding in nature, further consolidate the biological relevance of our findings.

**Keywords:** STDP, intrinsic plasticity, homeostatic plasticity, recurrent, spatiotemporal computations, nonautonomous dynamics, information theory, noise

AUTHOR SUMMARY The world is not perceived as a chain of segmented sensory still lifes. Instead, it appears that the brain is capable of integrating the temporal dependencies of the incoming sensory stream with the spatial aspects of that input. It then transfers the resulting whole in a useful manner, in order to reach a coherent and causally-sound image of our physical surroundings, and to act within it. These spatiotemporal computations are made possible through a cluster of local and coexisting adaptation mechanisms known collectively as neuronal plasticity. While this role is widely known and supported by experimental evidence, no unifying theory of how the brain, through the interaction of plasticity mechanisms, gets to rep-

resent spatiotemporal computations in its spatiotemporal activity. In this paper, we aim at such a theory. We develop a rigorous mathematical formalism of spatiotemporal representations within the input-driven dynamics of cortical networks. We demonstrate that the interaction of two of the most common plasticity mechanisms, intrinsic and synaptic plasticity, leads to representations that allow for spatiotemporal computations. We also show that these representations are structured to tolerate noise and to even benefit from it.

## 2.1 INTRODUCTION

Neuronal plasticity, both homeostatic and synaptic, is the central ingredient for the generation and adaptation of neural function and computation [25]. However, it remains mostly unclear how neurons in recurrent neural networks utilize neuronal plasticity to self-organize and to learn computing on temporally and spatially extended stimuli [15, 19, 84].

A full grasp of the principles of self-organization by plasticity in recurrent neural networks is *jointly* hampered by the diversity of existing neuronal plasticity mechanisms [2, 145, 133] and the limited understanding of their functions and cooperations, by the *emergent* nature of computation in recurrent systems, in the sense that computation is a collective phenomenon of the system as a whole and cannot be fully understood from the contribution of individual neurons [107, 56], and by the fact that neural systems are subject to noise [115, 37, 122, 106]. In this paper, we simultaneously address these issues by studying the basic principles of self-organization in *recurrent* networks that arise from the interaction of synaptic and homeostatic intrinsic *plasticity*, and given that the network is subject to *noise*. To this end, we use numerical methods to explore the dynamics of nonautonomous, i. e., stimulus-driven, and plastic recurrent networks, and we provide a mathematical formalization for attaining a rigorously sound perspective (see Methods).

Incorporating synaptic plasticity with homeostasis goes back to Bienenstock, Cooper, and Munro's groundbreaking work known as the *BCM theory* [12]. Through rigorous mathematical analysis, the BCM theory predicted the necessity of a certain form of a sliding threshold, i. e., a homeostatic adjustment of neuronal excitability, for stabilizing the plastic afferent weights of a single neuron. Empirical findings supported the hypothesis of adjustable excitability and showed that it manifests through changes of neuronal properties at the soma [145, 133]. While the BCM theory suggests homeostasis as a stabilization mechanism of synaptic weights with no direct influence on the neuron's encoding properties, Triesch proposed a homeostatic ***i****ntrinsic* ***p****lasticity* (IP) mechanism that increases the neuron's encod-

ing capacity and cooperates with ***s**ynaptic **p**lasticity* (SP) to discover nonlinear independent features of the neuron's inputs [132].

These investigations, among others [131, 110], are very insightful in pinpointing how synaptic and homeostatic plasticity interact in *single neurons*. In addition, *feedforward* neural networks greatly simplify the analysis and understanding of self-organization and computation based on neuronal plasticity. For such architectures, both single plasticity rules, as well as combinations of different plasticity mechanisms, had been linked to neural computation, such as the formation of receptive fields [12], the related identification of statistically-independent components [132, 24, 110], and predictive coding [18]. However, it is important to note that neurons are embedded within large and highly recurrent networks [33, 32, 34, 140], and that an efficient use of neuronal resources entails distributed encoding schemes [107, 56]. In addition, besides the spatial features of the world, its temporal structure should also be captured by the neural code [30, 23, 76, 84].

Our understanding of neural information processing would greatly improve by extending the principles of self-organization to recurrent neural circuits, since the latter constitute the basic computational units in the cortex [34]. Lazar et al. were the first to study the emergence of computation from the interaction of different forms of plasticity on recurrent neural networks [67, 68]. This study builds on their findings. However, we *do not* restrict the definition of computation to linear classifiers of the ***r**eservoir **c**omputing* (RC) paradigm [57, 70, 19]. In addition to training linear classifiers for measuring the *computational performance*, we identified the necessity of analyzing the response of the recurrent neural network itself as an *input-driven dynamical system* [63, 62], and of concurrently viewing the network as a *communication channel* by taking an *information-theoretical* perspective [26]. Combining these tools enables us to understand how information is encoded in recurrent systems, how such encoding is developing from self-organization, and how noise is effecting both.

Analyzing the dynamics of a *large* and, most importantly, *input-driven* neural system shaped by biologically-relevant *plasticity* is a hard task due to several methodological constraints. First, most analysis tools from *dynamical systems theory* are confined to small dynamical systems with very few degrees of freedom [125]. Exceptions are studies that circumvent this limitation by focusing on the low-dimensional collective dynamics of neural networks, e. g., [16], or studies that probe the high-dimensional phase space of the neural network, such as the classic example of Hopfield Networks [53]. Other instances of high-dimensional dynamical systems include ring networks and their coexisting periodic attractors [93], stable heteroclinic orbits [98, 99], unstable periodic attractors [127], and others [117, 41, 74].

The second and most important methodological constraint is that the use of standard dynamical systems theory is inappropriate, since it deals with autonomous systems only, i. e., systems with no explicit dependence on time. In reality, however, neural networks are subject to a flux of ever changing stimulation that renders them nonautonomous. A theory of *nonautonomous dynamical systems* is only recently taking shape as a branch of applied mathematics [63, 62]. The fields of neural computation and computational biology are constantly contributing to the theory with concepts such as meta-transients and attractor morphing [93, 83], $\gamma$-systems [91], and the nonautonomous dynamics of echo state networks [71].

A simple intuition of the difference between nonautonomous and autonomous systems can be stated as follows. Attractors of an autonomous dynamical system are defined by the system alone, and are therefore fixed. In contrast, attractors of a nonautonomous system are jointly defined by the dynamical system and its input. As the input changes, so does the attractor landscape of the system. This highlights the fact that studying computations in a driven system using the methods of autonomous dynamical systems is insufficient, since the input-induced changes of the system, i. e., changes of its attractor landscape, are ignored in that case.

The third constraint is that the complexity of the dynamics increases due to the neural system's adaptability. The presence of plasticity imposes restrictions on the dynamics a network can exhibit, thus keeping the network dynamics in a regime that can support complex computations. To the best of our knowledge, no attempt prior to this work has been taken to combine high-dimensionality and nonautonomy with the consequences of plasticity on dynamics. We demonstrate that plasticity *sculptures* the stimulus-specific dynamic landscapes, and by that, serves in improving representation of the provided input. Moreover, neuronal plasticity can adapt and learn stimulus-induced sequences of such stimulus-specific landscapes. We thereby show that neuronal plasticity improves spatiotemporal computations.

Given the above, we highlight and explain that spatiotemporal computations require two basic ingredients: a homeostatic mechanism that regulates neuronal activity, and synaptic learning that adapts the network's recurrent connectivity to the stimulus. We show that combining both types leads to a system that: first, learns the temporal structure of the input and carries out nonlinear computations, second, is noise tolerant, and third, even benefits from the presence of noise that sets the system to an input-sensitive dynamic regime.

The paper is structured as follows. We first characterize the effects of self-organized adaptation that is based on synaptic and homeostatic intrinsic plasticity and their combination. For that, we use tasks where both random and temporally-structured inputs are re-

constructed and predicted, as well as a task where nonlinear computations are performed. We estimate the network's self-information capacity (its *entropy*), and its input-information capacity (the *mutual information* between the input and the network). We then interlude to qualitatively analyze the resulting dynamics of plastic changes based on the theory of nonautonomous dynamical systems. We explain the superior computation of conjoining synaptic and intrinsic plasticity based on both the informational and dynamical analyses. Building upon that, we study network noise, and demonstrate how noise is combated and exploited through the interaction of synaptic and intrinsic plasticity.

## 2.2 RESULTS

In this section, we guide the reader through the following topics. We start by elucidating the computational power gained through the combination of synaptic and homeostatic plasticity mechanisms on recurrent neural networks of the *k-Winner-Take-All* (kWTA) type. We investigate the role of these plasticity forms in shaping the neural code through their effects on the informational and dynamical landscapes of the network. We conclude by illustrating how synaptically and homeostatically organized recurrent networks both benefit from noise and tolerate its presence. Figure 1 schematically illustrates the network model, the plasticity rules, and the formal probes we used to evaluate and describe the resulting computational properties. More details are available in the Methods section.

### 2.2.1 *Computational Power*

The interaction of different forms of plasticity produces a rather complex emergent behavior that cannot be explained trivially by the individual operation of each. We therefore start with exploring the effects induced by the combination of ***s**pike-**t**iming-**d**ependent synaptic **p**lasticity* (STDP) and ***i**ntrinsic **p**lasticity* (IP). We compare the computational performance of recurrent networks trained either with both synaptic and intrinsic plasticity (SIP-RNs), with synaptic plasticity alone (SP-RNs), or with intrinsic plasticity alone (IP-RNs), in addition to nonplastic recurrent networks, where the synaptic efficacies and firing thresholds are random.

Following the *plasticity phase*, a network is *reset* to random initial conditions and the *training phase* starts. Output weights from the recurrent network to linear readouts are computed with linear regression so that the readouts activity is the optimal linear classifier of a target signal. The target signal depends on the computational task. That is followed by the *testing phase*, at which performance is

computed. Performance is measured by the percentage of correctly matched readout activity to the target signal.

Naturally, during simulation, the recurrent network is excited by a task-dependent external drive. The battery of tasks we deployed was designed to *abstract* a certain aspect of the spatiotemporal computations faced by biological brains, i. e., recalling past stimuli, predicting future ones, and nonlinearly transforming them. The memory task `RAND×4`, the prediction task `Markov-85`, and the nonlinear task `Parity-3`, as well as the plasticity models and simulation conditions, are detailed in the Methods section.

Figure 2 shows that `SIP-RN`s significantly outperform both `IP-RN`s and `SP-RN`s in all tasks. Inputs from 3 time steps in the past are successfully retained far beyond chance level in the memory task `RAND×4` (Figure 2A). Understandably, performance drops to chance level for future stimuli (positive time-lags), since input symbols are equiprobable and their temporal succession carries no structure. Such is the case for the nonlinear task (Figure 2C). It is worth noting that solving the nonlinear task `Parity-3` requires recalling three successive stimuli, which adds to the computational load. The recurrent network, through learning the temporally-structured input of the task `Markov-85`, boosts the readouts' ability to reconstruct past symbols in comparison to the structureless memory task `RAND×4`. It also allows for the prediction of future stimuli far beyond chance (Figure 2B).

`STDP` alone fails to provide the recurrent network with means to encode necessary information. This leads to `SP-RN`s performing at almost chance level in all tasks. Intrinsic plasticity, on the other hand, endues recurrent networks with an intermediate ability to sustain past inputs (Figure 2A). `IP-RN`s also seem to learn the temporal structure of the input, as optimal linear classifiers are capable of predicting future stimuli (Figure 2B). Intrinsic plasticity is, however, insufficient for nonlinear computations, as `IP-RN`s barely perform above chance in the nonlinear parity task.

We also compare the performance of nonplastic `kWTA` networks with similar weight and threshold distributions as `SP-RN`s (shown in gray in Figure 2). They perform better than `IP-RN`s on the memory and nonlinear tasks, and worse on the prediction task. In all tasks, these nonplastic networks perform worse than `SIP-RN`s. We also show in Section 2.5 that nonplastic networks with comparable weight and threshold distributions as `SIP-RN`s also perform significantly lower than plastic networks. These results supply the evidence that the presence of plasticity enhances the computational power of recurrent neural networks (see Section 2.5 for a discussion on heuristics for finding comparable random networks). No further analysis is carried out on these nonplastic networks, since the aim of this paper is to discern the effects of synaptic and intrinsic plasticity on spatiotemporal computations.

### 2.2.2 *Neural Code*

Explaining the superiority of networks modified by deploying both STDP and IP starts from isolating the individual role of each plasticity mechanism in defining the spatiotemporal neural code. In that regard, a well-informed intuition is that STDP learns the basic structure of the input as the connectivity resulting from STDP reflects the input sequence transitions. IP, on the other hand, increases the neural bandwidth by introducing redundancy to the code, as IP leads to the longest periodic cycles in the spontaneous activity of kWTA networks (See Figure 8 and Figure 4A in [67]).

The spatiotemporal neural code, or the *neural code* for short, can be characterized by both the *absolute capacity* of the network activity to store information and by how network activity *encodes* the spatially and temporally extended network input. *Entropy* of the network activity measures its absolute capacity, i. e., the repertoire of network states that the network can actually visit and potentially assign to some input sequence. The assignment of a network state to an input sequence means that this particular network state *encodes* or *represents* that input sequence. *Mutual information* between network input sequences and network states quantifies the extent of how successful this assignment is. Not every visited network state needs be assigned an input sequence. A redundant code is reflected by input sequences being represented by multiple network states. Also, a network state might fail to encode an input, thus reflecting uninformative noise states.

We investigate the neural code characteristics of kWTA networks by estimating both the entropy of the network state and the mutual information between network input sequences and network states. We drive the network by RAND$\times$4 input, and for computational tractability, we limit the estimation of mutual information to three-step inputs. An optimal encoder of this input sequence will then be a network with 6 bits of mutual information. The information-theoretical quantities are computed at intervals of the plasticity phase under the three plasticity conditions. At these intervals, the plastic variables are fixed and the driven network is reinitialized and run for a sufficient number of steps, and passed along with the input to the entropy and mutual information estimators. More details on how these measurements are carried out are found in the Methods section.

Figure 3 shows how these measures develop through the plasticity phase (For a discussion on the effects of longer plasticity exposure, see Section 2.6). SP-RNs' entropy remains constant at 2 bits. This means that SP-RNs visit only 4 network states (green in Figure 3A). However, these network states encode no information of the input sequence, as mutual information remains practically zero (green in Figure 3B). We call this 2 bits input-insensitive code the *minimal code*,

as it captures no more than a single possible succession of the 4 inputs. This effect is the result of the interaction between the machination of STDP and the initial firing thresholds and weights configuration. Transitions, such as $A \rightarrow C$ in the input space, are to be stored in some of the synapses that connect neurons in the receptive field of $A$ ($RF_A$) with those in the receptive field of $C$ ($RF_C$). At each time step, one transition, such as $A \rightarrow C$, could be easier to reinforce with the causal (potentiating) side of STDP for $RF_C$ neurons having little higher excitability (internal drive plus their own firing threshold). Without IP to tune down this excitability and with further contribution from the recurrency of the network, a positive feedback loop is generated, and this transition becomes more and more potentiated at the expense of others. This transition then becomes independent of the actual drive the network is receiving: the network becomes input-insensitive.

On the other side of the entropy spectrum, we find IP-RNs. Through IP's constant adjustment of the neuronal excitability, many neurons contribute to the neural code and IP-RNs visit a large number of states. Entropy and the network state bandwidth are the highest (blue in Figure 3A). One may view IP's effect as an introduction of *intrinsic deterministic noise* to the network activity. The increase in bandwidth of the network activity raises the odds for the random weights of an IP-RN to store an input sequence. In fact, many network states encode the same input sequence, resulting in a redundant code. However, without a synaptic reinforcement of representations, many states are visited due to the internal dynamics of the network, and not due to the external drive. These states remain uninformative and input sequences not successfully encoded: the mutual information (blue in Figure 3B), and hence the classification performance, are low.

The development of the neural code for SIP-RNs follows, however, a more interesting path. At the beginning, STDP has the upper hand and a 2 bits minimal code is generated. Through providing intrinsic deterministic noise, IP enriches the neural code by increasing redundancy and entropy (orange in Figure 3A). At the same time, STDP incrementally associates different network states to different input sequences by adjusting the synaptic weights as seen from the increase of mutual information (orange in Figure 3B). Then together, synaptic and homeostatic plasticity cooperate to create a code that is both *redundant* and *input-specific*. These properties are crucial for noise-robustness, as will be shown later in this text.

### 2.2.3 *Post-Plasticity Perturbation*

A dynamical system's behavior depends on its past activity. Therefore, testing a system requires assuming plausible initial conditions. The recurrent neural network at hand, even though it is small in comparison to a real neural circuit, has a number of possible initial condi-

tions too large for all its initial conditions to be tested. So far, we have chosen random initial conditions for the network activity following the plasticity phase. From now on, we choose the initial conditions systematically by reinitializing the network activity depending on a *perturbation* $\pi$. This perturbation is applied to the end state of the plasticity phase, such that the end state of the plasticity phase and the initial state of the training phase are at a distance $2\pi$ from one another. For details of how the initial conditions are selected depending on the parameter $\pi$, we refer the reader to the Methods section.

To discern the effect of this perturbation, we compute the performance of the trained system with the three combinations of synaptic and intrinsic plasticity. We do this both for a system that is perturbed and for a system that starts from the last state that the dynamics reaches at the end of the preceding plasticity phase. We find no difference between the two cases of initial conditions for either `IP-RN`s or `SP-RN`s. However, when the neural network is trained by both synaptic and intrinsic plasticity (`SIP-RN`s), we find that the perturbed networks have better performance, as is illustrated in Figure 4A–C. The high performance of `SIP-RN`s that results from random initial conditions, as is shown in Figure 2, is easily explainable. It stems from the fact that random initialization is merely a large perturbation, since the probability of choosing a random state from such a large set of possibilities that is at a small distance from a particular region of the state space is insignificant, compared to a state that is at a large distance. Moreover, we find that regardless of the task, larger perturbations result in higher average performance. This is also reflected in the neural code, where network state entropy and the mutual information with input correlate with higher perturbation (see Figure 4D–E).

This suggests that within the phase space of `SIP-RN`s there exist at least two dynamic regimes. Post-plasticity perturbation also provides the first sign of how `SIP-RN`s can benefit from noise, as it might put the system in the regime more suitable for computation.

### 2.2.4 *Dynamic Regimes*

Optimal linear classifiers show that `kWTA` networks equipped with both homeostatic and synaptic plasticity are capable of creating spatiotemporal codes and performing nonlinear computation. Measuring entropy and mutual information allows for a quantification of the emerging neural code. But what are the geometric features of the neural code that allow for such computations? How do network states *represent* the spatiotemporal input in a useful way? A major part of the Methods section is devoted to developing the mathematical formalization of discrete-time nonautonomous dynamical systems. References to definitions, a proposition, and a theorem from that section are featured in the following results, as we apply these concepts to

our model neural network. We view this treatment not merely as an exercise in mathematics. It allows for a rigorous description of the computational properties emerging from plasticity that are beyond the scrutiny of quantitative measures, such as linear classification performance and carried information. A consequence of these properties is also the two noise-related features we examine later.

For a formal treatment of spatiotemporal computations which result from plasticity, we need to extend the theory of nonautonomous dynamical systems to provide a notion for representations, to specify how these representations allow for computations, and to discern the effect of plasticity in enhancing these representations for the sake of computation. But first, we start by identifying the modes of operation, i. e., the dynamic regimes, the model plastic neural network has, since not all regimes might be suitable for computation.

According to Proposition 3 and Definition 6, when subject to stimulation, `kWTA` networks are *input-driven discrete-time dynamical systems*. For such systems, two extremes exist regarding the degree of sensitivity the system exhibits in response to its input. At one extreme, the system shows no change of response for different inputs, so that it follows its own dynamics, as if no input exists. In such a mode of operation, the system is *input-insensitive*. The other extreme is when the system's response is different for each input and initial condition. A single system can show, in principle, multiple modes of operation, depending on the initial conditions. The set of initial conditions that show a single mode of operation defines a dynamic regime and a basin of attraction.

In a first step, we visualize the high-dimensional response of the system to its input. To that end, we down-project the network activity to the first three principal components, and we study the effects of `STDP` and `IP` on the network's dynamics and input representations in this reduced 3-dimensional space (Figure 5). This analysis is performed on networks with `Markov-85` input which fully demonstrate the relevant properties. It is important to note that while our analysis concerns the dynamics following the plasticity phase, we are still able to infer how it unfolds during this phase from the development of the neural code (Figure 3), as we make clear later.

As suggested by the performance of `SP-RNs` (Figure 2) and their neural code entropy and mutual information (Figure 3), their state space is dominated by an *input-insensitive* basin of attraction and these networks behave like autonomous semi-dynamical systems (prefixing with "semi" refers to the fact that the dynamics needs not be invertible). This is confirmed by the asymptotic dynamics of `SP-RNs`, which is independent of the input (Figure 5B). The dynamics within this dynamic regime follows the minimal code. The minimal code manifests itself through a period-4 periodic attractor which corresponds, in the case of `Markov-85` input, to the most probable tran-

sition in the input space $A \to B \to C \to D \to \cdots$. This observation confirms the fact that STDP allows the system to learn the basic structure of its input.

SIP-RNs exhibits similar dynamics at the end of the plasticity phase (Figure 5C). However, as is evident from varying the perturbation parameter for SIP-RNs (Figure 4), the set of initial conditions that constitutes this input-insensitive basin is confined by a distance relation to the neighborhood of the periodic attractor: the probability of being in this basin diminishes the further away the initial conditions are from the input-insensitive periodic attractor.

The increase of performance and the neural bandwidth of SIP-RNs for higher $\pi$ (Figure 4) shows that outside of the *input-insensitive* dynamic regime there exists a different basin of attraction. Within this basin, the network is sensitive to input, and computations are possible. The observation that $\pi$ has no effect on IP-RNs and that they show intermediate performance and mutual information suggests that they are dominated by a dynamic regime with intermediate input-sensitivity. It also confirms that intrinsic plasticity is responsible for the emergence of the *input-sensitive* dynamic regime in SIP-RNs.

### 2.2.5 *Volumes of Representation*

Now that the dynamic regimes of trained networks with the three combinations of synaptic and intrinsic plasticity are identified, we next move to formulating the notion of representations inside the input-sensitive dynamic regime. Developing such a notion allows linking the theory of nonautonomous dynamical systems to a theory of spatiotemporal computations. To this purpose, we coin the term *volumes of representation*, which is a concept that describes the response of a nonautonomous dynamical system in respect to its drive. The volume of representation of some input sequence within some dynamic regime is the set of network states that are *accessible* through exciting the network with the corresponding input sequence, starting from all network states in this dynamic regime as initial conditions (Definition 10). The *order* of a volume is defined by the length of the input sequence it represents. We also introduce the *volumes' inclusion property* which hierarchically links the system's response to spatiotemporal input sequences to their sub-sequences.

To visualize a network's volumes of representation, we sample the network's response. We do this because the size of the state space and the input-sensitive dynamic regime is too large, making a complete coverage impossible. Also, since volumes of representation can have complicated shapes in both the full and reduced state space, we approximate these volumes with ellipsoids.

Figure 5D provides such an approximation to the volumes of representation of order-1. The sample is a single 10000 symbols `Markov-85` input sequence to a `SIP-RN`. Each volume is replaced by an ellipsoid. The center of this ellipsoid is the coordinates' average of the visited network states in the principal components space. Each of its semi-axes has a length that is the standard deviation from the mean of the corresponding coordinate. Also, according to the *volumes' inclusion property*, stated formally in the Methods section, a volume of representation of order-1 of some input p includes all volumes of order-2 for sequences whose most recent input is p. As such, Figure 5E, that depicts a similar approximation to all volumes of order-2, is also a better approximation to volumes of order-1. In Figure 5E, each order-1 volume consists of four order-2 volumes that are color-coded to match the rougher approximation in Figure 5D. In a supporting figure, we further show that this way of presentation is sufficient, compared to using percentiles of bootstrapped network states (see Figure 6).

The volumes of representation provide a geometric view of spatiotemporal computations as the ability of the recurrent neural network to *represent* in its activity, in other words to *encode*, useful functions of the network's input sequences, and for these representations to be distinguishable and reliable. In the case of the tasks `RAND`×4 and `Markov-85`, the functions that the network activity represents are the identity, delayed or forecast. As shown in Figure 5D–E, the volumes of representation of `SIP-RN`s under `Markov-85` input exhibit higher *separability*, which explains both their high classification performance and high mutual information. One also notices that the volumes of representation of order-2 that belong to the most probable transitions in the `Markov-85` input, e. g., B → C, are also the most distant from one another (Figure 5E). This results in the most probable transitions to be more easily distinguishable by optimal linear classifiers.

In order to isolate the roles of synaptic and intrinsic plasticity in generating useful representations, we show in Figure 5A the order-1 volumes of representation of an `IP-RN` in response to `Markov-85` input. Compared to the `SIP-RN`, these volumes are highly overlapping, which explains the lower classification performance. Also, the low mutual information between the network state and the input (Figure 3) can now be explained by various network states belonging to multiple volumes of representation, at once. Also, many network states represent the same single input which is a signature of *redundancy* resulting from `IP`. These observations point towards `STDP` being the source of separability of representations in `SIP-RN`s, in addition to learning the structure of the input through situating the representations of the input's most probable transitions at further distances from one another.

In the case of the task `Parity-3`, the function that the network activity needs to represent is the sequential *exclusive or* operation over

three successive binary inputs. As such, within the input-sensitive dynamic regime, two volumes of representation exists, each encodes one outcome of the nonlinear task `Parity-3`. According to Definition 10, these volumes are formed from an appropriate union of order-3 volumes of representation of the binary input. We provide an illustration of these two volumes of representation in Figure 7. Here also, `STDP` provides the separability that allows these representations to be distinguishable, while `IP` gives the possibility of an input-sensitive and redundant regime to emerge, and, aided by `STDP`, for the volumes of representation to expand.

### 2.2.6 *Attractor Landscape*

The presence of dynamic regimes entails the existence of *attractors*, i. e., limit sets of the dynamics, that apply a pulling force on the dynamical system's activity and dictate its course of flow. In an input-driven dynamical system, attractors are not easily defined as sets of states. Instead, *nonautonomous attractors* are input-dependent moving targets of the dynamics, which adds a temporal aspect to their definition (see Definition 8). As follows, for our nonautonomous dynamical systems theory of spatiotemporal computations to be complete, we link the geometry of the computational entities, i. e., the volumes of representation, to the geometry of the nonautonomous attractors. This allows us to connect the features of the volumes of representation emerging from plasticity, namely, separability and redundancy, to the effects of plasticity on the nonautonomous attractor. To that end, starting from the volumes of representations, we define the *perturbation set* (Definition 10) as a moving source of the neural activity towards its moving target, the nonautonomous attractor. Since the perturbation set changes with time, it is called a *nonautonomous set* (Definition 7). This also applies to nonautonomous attractors. The set of states constituting a nonautonomous set at a fixed time t is called the set's *t-fiber*. We later show how the t-fibers of these nonautonomous sets relate to each other.

In the input-insensitive dynamic regime, the dynamical system behaves as an autonomous dynamical system, and so does its attractor, which is the period-4 attractor in Figure 5B–C. In addition, the existence of a *nonautonomous basin of attraction* (Definition 9), that constitutes the input-sensitive dynamic regime in `SIP-RNs`, necessitates the existence of a *nonautonomous attractor*.

It is not possible to fully identify the nonautonomous attractor by looking into the nonautonomous dynamics. This is because the attractor is not fixed in space and because the dynamics almost never converges to it. However, we prove in Theorem 11.1 that in an input-driven discrete-time dynamical system, and within a basin of attraction, the nonautonomous attractor is a subset of the basin's perturba-

tion set, and that the t-fibers of a nonautonomous attractor are subsets of the t-fibers of the perturbation set. Given this result, the location of the nonautonomous attractor within the state space of the network can be approximated by the perturbation set. The perturbation set summarizes how the network activity passes from one volume of representation to another, at every time step, according to the input's transition statistics. We replace the time dimension in Figure 5D by arrows that correspond to the transitions in `Markov-85` input. The volume of representation visited at time $t$ is the volume corresponding to the input at that time, and it forms the t-fiber of the perturbation set.

Instead of defining the asymptotic dynamics of the model neural network within the input-sensitive basin of attraction by a single nonautonomous attractor with different t-fibers, we can define it by multiple autonomous attractors, each belonging to a particular input. According to Theorem 11.2, within the input-sensitive basin of attraction, there exists for each input $p$, an autonomous attractor (Definition 4) of the autonomous semi-dynamical system defined by $p$. The theorem also shows that this attractor is a subset of the volume of representation of $p$. Theorem 11.3 further shows that the basin of attraction of the autonomous attractor is also the input-sensitive basin. Accordingly, the network dynamics undergoes a *bifurcation* at each time step the input changes its identity. A bifurcation is a change in the topological properties of invariant sets, such as attractors. We observe bifurcations in the input-sensitive regime of `kWTA` networks. The topological property undergoing the change is the loss of stability of the periodic attractor associated with an input $p(t-1)$, and the appearance of an attractor with a different period and location that is associated with the input $p(t)$.

Figure 5F shows the autonomous periodic attractors associated with each `Markov-85` input within the input-sensitive basin of attraction of a `SIP-RN`. Each of these attractors is also a t-fiber of the nonautonomous input-sensitive attractor. While these autonomous attractors are depicted in one state space, overlaying them in a single plot serves only in illustrating the geometric relations between them. In reality, these attractors do not *coexist*. Each autonomous attractor appears in the phase space of the network when its associated input drives the network, and the attractor from the previous time step disappears.

The geometry of the nonautonomous attractor within an input-sensitive dynamic regime is very important regarding spatiotemporal computations. In fact, computations are completely defined according to the relative positions of the nonautonomous attractor's t-fibers to one another, and to the volumes of representation. An attractor consists of limit points of a basin of attraction. Thus, it exerts a pulling force on the network states that define the volumes of representa-

tion. So, if the t-fibers of a nonautonomous attractor are close to one another in the state space of the network, different volumes will be overlapping and computations will be difficult to carry through. Such is the case in IP-RNs. On the other hand, distant t-fibers of the nonautonomous attractor result in separate volumes of representation and better spatiotemporal computations, which is the case in SIP-RNs (Figure 5D–F). Also, the number of states comprising the t-fibers of the nonautonomous attractor effects the redundancy of representations. As intrinsic plasticity increases the number of states of these t-fibers, the perturbation set becomes more redundant. Given the above, while the perturbation set contains the nonautonomous attractor, it is the attractor that defines how the perturbation set, and as a consequence the volumes of representation, extends in space.

For a correct characterization of spatiotemporal computations according to the geometry of the nonautonomous attractor and function representations, we borrow the concept of *meta-transients* [83]. A transient activity of an autonomous (semi-)dynamical system is the trajectory its dynamics follows as it approaches a fixed attractor. Alternatively, an attractor of an input-driven dynamical system changes constantly. This leads the trajectory pursued by the dynamics to switch its course, so as to keep track of its moving target. Such an input-dependent trajectory is termed a meta-transient. When the input changes, the meta-transient passes from one volume of representation to another, i. e., the dynamics bifurcates and the meta-transient approaches the vertexes of the current attractor, while being repelled from the others that are now unstable. It is in this geometric relation to the different attractors (or t-fibers) that computation resides. In fact, as a proof of principle, the autonomous attractors of SIP-RNs were allocated. This was done by clamping each input for a sufficient time until the dynamics converges to that input's periodic attractor. Then, optimal linear classifiers were fitted to perform the three computational tasks. As training data, the Hamming distances between the meta-transient and the vertexes of these periodic attractors were used. Figure 9 shows the performance resulting from this computational procedure, which outperforms both SP-RNs and IP-RNs. While the performance is far from what is achieved directly from the activity of SIP-RNs, especially in the nonlinear task Parity-3, it is important to note that distance is a very rough compression of the geometric relations between the meta-transient and the autonomous attractors. For instance, distance does not allow the distinction between network states that are symmetrical in relation to the autonomous attractors.

### 2.2.7 *Emergence of Computation*

We now outline how the interaction of homeostatic and synaptic plasticity gives rise to spatiotemporal computations through devel-

oping useful representations. To this end, we combine the analysis of dynamic regimes, volumes of representation, and autonomous and nonautonomous attractors (Figure 5) with the informational-theoretic intuitions regarding the evolution of the neural code (Figure 3).

At the beginning of the plasticity phase, STDP has the upper hand and it generates a minimal code of the input. This is evident from the 2 bits network state entropy (Figure 3A) and the close to zero mutual information with input (Figure 3B) at the beginning of the plasticity phase of SIP-RNs. The minimal code captures, through an input-insensitive periodic attractor, the most probable transitions in the input (Figure 5B). Another feature of the input-insensitive periodic attractor is the high separability of its vertexes in the state space of the SIP-RN.

at the same time, IP succeeds in reducing the excitability thresholds of some neurons, such that more network states become accessible at the vicinity of the vertexes of the input-insensitive attractor: entropy increases alongside the potential for redundancy. STDP concurrently assigns these network states to the inputs that induce them: mutual information and redundancy increase. This incremental process manifests dynamically in the appearance of the input-sensitive basin of attraction, and the associated appearance and expansion of volumes of representation (Figure 5D–E). Due to the highly separate vertexes of the input-insensitive attractor and the neighborhood relations of the volumes with these vertexes, the volumes of representation are highly separate. This shows that the input-insensitive dynamics is a necessary prerequisite for the emergence of spatiotemporal computations, as it sets the stage for the appearance of separate representations that also carry the structure of the input.

The emerging dynamics can also be viewed through formulating the SIP-RN during the plasticity phase, as an input-driven dynamical system parametrized by the weights and the excitability thresholds. Through varying the parameters of the system with STDP and IP, the dynamics at some point in the parameters space bifurcates from one stable dynamics, the input-insensitive dynamics, to two stable dynamics with the appearance of the input-sensitive attractor in whose basin computations are realizable. This also applies to each member of the family of semi-dynamical systems with the appearance of new dynamics and the associated new periodic attractor (Figure 5F).

### 2.2.8 *Noise-Robustness*

Equipped with different vantage points to describe the information processing properties of plastic recurrent neural networks, we now turn to ask a central question: what does an information processing system like the brain require in order to be noise-robust? We state the following hypothesis. Noise-robustness is an effect of the interplay

between 1) a *redundant code* that provides multiple possible encodings of an input, and 2) *separability* of representations which allows for a *margin of noise* without obscuring the identity of the input.

The analysis of the neural code (Figure 3) shows how IP increases the potential for redundancy by increasing the neuronal bandwidth. STDP could exploit this potential redundancy by assigning multiple neurons to the same input. Viewing the network dynamics in the principal components space, on the other hand, made clear that STDP ensures separability in the volumes of representation (Figure 5D–E). This also suggests that the recurrent network should be more robust to noise, the more recent the decoded input is, as the margin of noise becomes smaller for older inputs. The expansion of volumes of representation in IP-RNs also points towards a higher potential redundancy.

We test the hypothesis and the role of STDP and IP interaction in noise-robustness by injecting nondeterministic noise into the recurrent network. Following the plasticity phase, we deploy a certain rate of random bit flips on the network state that reserves the kWTA dynamics, i. e., if some neuron is silenced due to noise, another neuron is selected at random and it fires instead. Different networks with different input statistics will amplify the same amount of noise to a varying extent. The shaded area in Figure 8 marks the ratio-of-noisy-spikes range within the network states of 100 recurrent networks. For all tasks and networks, we measured performance of optimal linear classifiers on both the noise-free and noisy network states, and computed the relative change in performance.

We compare the change in performance for each time-lag with the ratio of noisy spikes. To understand how this comparison aids in characterizing noise-robustness, we rely on an example. If 10% of a network's spiking activity has been replaced by noise, spikes being the carriers of information, 10% of the information in the network would be lost. However, if the activity of other neurons within the network is a replica of half the lost spikes, only 5% of the information would be lost, and the performance of the linear classifiers would decrease just as much. Having the change of performance below noise level is evidence of noise-robustness due to redundancy and intrinsic plasticity.

Information carried by the network cannot deteriorate beyond the amount of noise; the ability to perform computations, on the other hand, is another story, since distinguishing between representations is a necessary condition for computation. Noise can lead to an overlap in the volumes of representation, which hinders the information processing capability of the recurrent neural network, since overlapping representations are indistinguishable and prone to over-fitting by decoders, linear or otherwise. However, when volumes of representation are well separated due to STDP, and redundancy is at play,

performance will not exceed the amount of noise in the network: noise-robustness is still achieved.

Figure 8 shows that redundancy and separability are assuring noise-robustness in the three tasks. The effects are the strongest for the task RAND×4. The change of performance never exceeds the range of noise for all time-lags. The change of performance on the task Markov-85 remains below the range of noise for few time-lags in the past and it remains within the bounds of the noise range for older stimuli. The networks then are still capable of tolerating noise, while the volumes of representation are becoming more overlapping. The decrease of noise-robustness for larger time-lags in the past confirms our suggestion that volumes of representation become less separate for older inputs. The analysis of order-2 volumes of representation (Figure 5E) also suggests that less probable transitions of the input are more prone to noise. This, however, was not tested. The task Parity-3 is noise-robust for 0-time-lag only and with the change in performance being within the noise range. This is understandable, since for each time-lag, order-3 volumes of representation and the associated volumes of the Parity-3 function should be separate and redundant.

These observations confirm our hypothesis that redundancy and separability are the appropriate ingredients for a noise-robust information processing system, such as our model neural network. These properties being the outcome of STDP's and IP's collaboration, suggest the pivotal role of the interaction between homeostatic and synaptic plasticity for combating noise.

### 2.2.9 *Constructive Role of Noise*

Now that we have demonstrated the contributions of STDP and IP in combating noise, we turn to investigating noise's beneficial role. We have seen that perturbation at the end of the plasticity phase provides a solution to the network being trapped in an input-insensitive regime. Besides viewing perturbation as a form of one-shot strong noise, which is, biologically speaking, an unnatural phenomenon, what effect would a perpetual small amount of noise have on the dynamics of the recurrent neural network?

We again deploy a certain rate of random bit flips on the network state that reserves the kWTA dynamics. Unlike the previous section, we do not restrict noise to the training and testing phase, but apply it also during the plasticity phase. We also do not reset the network activity after the plasticity phase, i. e., the perturbation parameter $\pi$ is set to 0.

Figure 10A–C compares the performance of optimal linear classifiers on the three tasks for different levels of noise. For all tasks, some levels of noise resulted in a significantly higher average performance than the noiseless case. The task Markov-85 had the highest average

performance at the largest level of noise, while the tasks RAND×4 and Parity-3, where the input was uniformly random, had the highest performance at the third and fourth levels of noise, and the average performance dropped substantially at the fifth level of noise. In all tasks, performance was far off the levels it reached in the noiseless case (Figure 2)

Information-theoretical quantities are again measured on networks with RAND×4 input. As expected, the network state entropy increases monotonically with noise (Figure 10D). Mutual information, on the other hand, starts dropping for noise larger than the third level (Figure 10E). This is also expected from the change of performance (Figure 10A). Noise then appears to provide, in some of the SIP-RNs, the necessary means to escape the input-insensitive dynamics. At some levels, however, the network activity becomes dominated by noise beyond the compensatory effects of redundancy and separability achieved through plasticity. In addition, more unstructured noise during the plasticity phase delays the creation and expansion of useful volumes of representation, thereby hindering computations further.

## 2.3 DISCUSSION

We demonstrated how the interaction of synaptic learning and homeostatic regulation boosts memory capacity of recurrent neural networks, allows them to discover regularities in the input stream, and enhances nonlinear computations. We provided a geometric interpretation of the emergence of these spatiotemporal computations through analyzing the driven dynamic response of the recurrent neural network. We view computations as a geometric relationship between *representations* of functions over stimuli, representations that consist of network states, and the asymptotic dynamics of the network, i. e., attractors. Accordingly, Figure 11A shows a possible driven-dynamics viewpoint on computation, which is the following. As the stimulus changes, a bifurcation occurs where the current attractor of the network becomes unstable, while another stabilizes according to the current stimulus. That leads the network dynamics to change its course towards the new stable region, or attractor, of the state space, and away from the previous attractors that are all unstable. As such, this path of the network activity, i. e., the meta-transient [83], is defined by both the stimulus sequence and the locations of the network's attractors. Together, they lead the meta-transient to pass through particular representations which encode computations. An equivalent alternative to the *chain of bifurcations* between autonomous attractors is that of a single *nonautonomous attractor* that behaves as a stimulus-dependent moving target of the dynamics.

We showed that a successful implementation of these spatiotemporal computations requires the interaction of synaptic and homeo-

static intrinsic plasticity which generates *useful representations* in the dynamics of excitable cortical networks. Figure 11 schematically illustrates the stimulus-driven dynamical viewpoint of spatiotemporal computations and the effects of plasticity. Synaptic plasticity produces stimulus-insensitive dynamics that captures the temporal structure of the input. Intrinsic plasticity increases the neuronal bandwidth by increasing sensitivity to stimuli, which reduces the dominance of the stimulus-insensitive dynamics. This, in combination with synaptic plasticity, generates stimulus-sensitive attractors and *redundant* representations around them. These stimulus-sensitive components are pulled apart by the stimulus-insensitive dynamics, so that the structure of the input is preserved, and the *separability* of representations is higher and computations are realizable.

We pointed out throughout the text that computation is an *emergent* property of the recurrent network, and that it cannot be fully understood from the individual contribution of the parts, be it neurons or plasticity mechanisms. It might appear contradictory to that statement that the analysis was often concerned with the isolated role of each single plasticity mechanism. However, the quantitative assessments of computations point back to the emergent and collective aspect of computation. Namely, measured on `SIP-RN`s, neither performance of linear classifiers nor mutual information with input can be accounted for by a linear relationship between the respective quantities measured on `SP-RN`s and `IP-RN`s. In fact, the performance of networks where the recurrent weights and firing thresholds are adapted separately, and then combined following the plasticity phase, is far less than the performance of `SIP-RN`s, where intrinsic and synaptic plasticity are mutually active (see Figure 12). This further consolidates the claim that computations in `SIP-RN`s *emerge* from the *interaction* of `STDP` and `IP`, and *not* from their isolated contributions. It also points back to the formation of separate and redundant representations from the continuous interplay of these two mechanisms.

We also illustrated the combined role of synaptic and homeostatic intrinsic plasticity in creating noise-robust encoding through the generation of a redundant neural code. Many studies have investigated the redundant nature of neural information transmission in many cortical regions, and have justified this expensive allocation of neural resources by redundancy serving as an error-correction strategy that provides neural assemblies with the capacity to average out noise [115, 89, 81, 97, 22]. Tkačik et al. have shown that in the presence of noise, a maximum entropy model of the retina increases redundancy for higher noise levels. A side effect of their model is that stimulus representations become highly separate, which increases the tolerance margin of noise and enhances information transmission [128]. Our model was able, through *local* plasticity mechanisms, to capture both of these properties, achieved in [128] through optimality prin-

ciples, and to lead to a noise-robust population code (Figure 11B). Namely, synaptic plasticity enhances the separability of representations through the pulling force of the input-insensitive attractor, while intrinsic plasticity perturbs the network states and increases redundancy when interacting with synaptic plasticity, which allows for alternative representations of similar input sequences. Another point of similarity with the model of Tkačik et al. [128] and with empirical findings [61, 36] is the remnant fingerprint of the most common stimulus in the network's spontaneous activity, which manifests in our model neural network in the stimulus-insensitive dynamics (Figure 5B–C).

In addition to combating noise, our model explores a potential benefit from its presence. We pointed out the necessity of the stimulus-insensitive dynamics for the emergence of computation in the model neural network. The stimulus-insensitive attractor provides the baseline dynamics for the appearance of highly separate representations, and thus, the excitable dynamics necessary for computations. Getting from the input-insensitive regime to the excitable one depended, however, on the *ad hoc* reinitialization of the network activity at the end of the plasticity phase. Noise provides an alternative. During the plasticity phase, noise shallows the boundaries between the two basins of attraction, which reduces the dominance of the stimulus-insensitive attractor. After the plasticity phase, noise supplies the small perturbations needed to get the network activity to the sensitive dynamics where computations are possible. This solution, in comparison to reinitializing the network activity, is more inferior, specifically because noise also delays the learning of representations. We postulate that another homeostatic plasticity mechanism, *synaptic scaling*, might contribute to the shallowing of the attractor boundary by constraining the strength of synapse bundles between neural subpopulations (e. g., between $\mathrm{RF}_A$ and $\mathrm{RF}_C$). For instance, synaptic scaling was necessary for implementing spatiotemporal computations in ***s**elf-**o**rganizing **r**ecurrent **n**etworks* (SORN) [68], but no analysis of the dynamics of these networks was done. Testing this hypothesis is, however, beyond the scope of this work.

It is also tempting to connect the topology of the attractor landscape of SIP-RNs to neuropathology and to a model by Pfister and Tass [95]. They suggest that two stable regimes of recurrent network activity, a synchronous pathological regime and an asynchronous healthy regime, coexist, and that their coexistence is a necessary condition for the functioning of a model of deep brain stimulation. In their model, the stimulation of the recurrent network destabilizes the synchronous dynamics through inducing STDP. The destabilization drives the network activity towards the healthy asynchronous basin of attraction. By eliminating the stimulation, the energy hill between the two dynamic regimes rises again and the network remains in the

healthy dynamics. Our study has shown how these two coexisting dynamic regimes and their associated forms of activity might come into being through neuronal plasticity. We also suggested noise as a possible mechanism for avoiding the unhealthy dynamics. Further analysis is necessary to investigate how the interaction between noise and different plasticity mechanisms might contribute to our understanding of neurological disorders.

Our analysis of spatiotemporal computations was restricted to Markovian dependencies in the temporal structure of the stimulus or to no dependencies at all. This is often not the case in natural stimuli faced by animals and humans, where the Markov property does not always hold. Lazar et al. have shown that SIP-RNs are capable, to a certain degree, of performing predictions on second-order Markov chains [67]. However, optimal encoding of non-Markovian stimuli and performing computations over them require forms of spike-timing-dependent plasticity that are less myopic to the temporal dependencies than what we considered in this work (Figure 1B). For instance, Brea and colleagues have shown that storing and reproducing a non-Markovian sequence in a recurrent neural network require a nonlocal form of STDP with more complex temporal dependencies between pre- and post-synaptic spikes [14]. While their model was not concerned with carrying through spatiotemporal computations of the kind we presented here, it successfully reproduced the stored non-Markovian input in the spontaneous activity of the neural network. This refers to a point of similarity to the simpler case we presented here, where Markovian input was stored and recalled in the spontaneous activity of the input-insensitive dynamics. In any case, while spatiotemporal computations over non-Markovian stimuli and the necessarily more complex plasticity mechanisms that lead to their emergence, are not considered here, we view the concepts and methodology developed above as a general framework for future studies.

In this article, we provided a first analysis of the combined role of synaptic and intrinsic plasticity on the emergent dynamics of recurrent neural networks subject to input. We redefined computations in relation to these emergent dynamics and related that to properties of the neural code. We also considered how the neural dynamics interact with noise, both as a nuisance to combat, and as a driving force towards healthy neural activity. The model we used is simplified, however, both in network architecture and plasticity mechanisms. While this simplification is necessary for mathematical convenience, biology never cares for formal abstractions, for the brain is a complex information processing system that is rich with a variety of neuronal morphologies and functions. The plastic changes the brain undergoes are neither confined to the two mechanisms we dealt with here, nor are they uniform across different regions. On the other hand, math-

ematical formalization of computation and adaptability allows the identification of unifying principles in computational biology, in general, and neural computations, in particular. We intended the current article as a step in that direction.

## 2.4 METHODS

The setup on which we assessed spatiotemporal computations in recurrent neural networks is partially inspired by the theory of *reservoir computing* (RC) [57, 70, 19]. However, as shown in the Results section, our analysis is independent of the RC paradigm, as it is concerned with the effects of plasticity on the recurrent network, and optimal linear classifiers are only used as one possible probe to quantify these effects. We present in this section the *recurrent network* (RN) architecture and the plasticity mechanisms active in shaping the neural response. We follow by introducing the computational tasks and justifying their selection. We then specify the simulation conditions and the training of optimal linear classifiers, followed by demonstrating how information-theoretical quantities are estimated. We finally lay down the mathematical formalization of the autonomous, input-driven, and input-insensitive dynamics of the recurrent network: We adapt Definitions 2, 4, 6–8 from [63] to the special case of discrete-time dynamics [62], which is the case that concerns the current article. We contribute the new concepts of volumes of representation and perturbation sets (Definition 10), input-insensitive dynamics (Definition 12), and the volumes' inclusion property. We also state and prove Theorem 11, which is necessary for the characterization of the attraction properties of the autonomous and input-driven dynamics. Definitions 5 and 9 formulate autonomous and nonautonomous basins of attraction according to our notation and purposes.

### 2.4.1 *Network Architecture*

In this paper, the model recurrent network is of the *k-Winner-Take-All* (`kWTA`) type [67] that consists of $n$ memoryless binary neurons from which only $k$ neurons are active. The discrete-time dynamics of the recurrent network at each time step $t \in \mathbb{Z}^+$ is given by

$$x(t+1) = f\big(w \cdot x(t) - \theta + \delta(t)\big), \tag{5}$$

where $x \in \mathbb{R}^n$ is the network state. The nonlinear function $f$ sets the $k$ units with the highest activities to 1 (spiking), and the rest to 0 (silent). As such, the population firing rate is held constant at $k$, and there is no need to introduce inhibitory neurons to balance excitation and inhibition. Recurrent synaptic efficacy is defined by the weight matrix $w \in [0,1]^{n\times n}$ with $w_{ij}$ being the efficacy of the synapse connecting

neuron $j$ to neuron $i$. Self-coupling is avoided by setting diagonal elements $w_{ii}$ to 0. $\theta \in \mathbb{R}^n$ defines neuronal firing thresholds that modulate the neurons' resistance to firing, and hence, their excitability. $\delta \in \mathbb{R}^n$ is the external drive whose dynamics depends on the task performed.

More formally, the set of possible network states is a metric space:

**Definition 1.** Given the set $Y = B^n = \{0,1\}^n$ of all binary vectors of size $n$, we define the *Hamming metric* by the function:

$$d : Y \times Y \to \mathbb{Z}^+ \cap [0,n] : d(y_1, y_2) = \sum_{i=1}^{n} |y_1^{(i)} - y_2^{(i)}|$$

According to this metric, the distance between two vectors of $Y$ is the number of bits at which these two vectors differ. The Hamming metric is a proper metric on strings of fixed length which is the case for $Y$. The pair $(Y, d)$ then forms a *metric space*. It is also equivalent to the $L_1$ norm on the set $Y$, which allows us to define the *Hamming length* of a vector $y \in Y$ as the Hamming distance between $y$ and the 0-vector, i. e., $|y|_d = d(y, 0)$.

Given the kWTA dynamics (see equation 5), the network activity is restricted to the set:

$$X = B_k^n = \{x \in Y : |x|_d = k\} \tag{6}$$

Since $X \subset Y$, the pair $(X, d)$ forms a metric space as well. Distances between subsets of $X$ can be measured using the Hausdorff metric, which we also denote $d$.

### 2.4.2 *Plasticity Mechanisms*

We are concerned with the interplay of two forms of plasticity in enhancing the computational capability of the model recurrent network.

***S**pike-**t**iming-**d**ependent synaptic **p**lasticity* (STDP) is a set of Hebbian and anti-Hebbian learning rules, where synaptic efficacy is modified according to the relative firing time between pre- and post-synaptic neurons [75]. We adapted a simple causal STDP learning rule by which a synapse is potentiated whenever the pre-synaptic neuron fires one time step before the post-synaptic neuron, and is depressed when a post-synaptic spike precedes a pre-synaptic spike by one time step:

$$\Delta w_{ij}(t+1) = \eta_{sp}\big(x_j(t) \cdot x_i(t+1) - x_i(t) \cdot x_j(t+1)\big), \tag{7}$$

where $\eta_{sp}$ is the synaptic plasticity learning rate set to 0.001. To prevent weights from switching signs or growing uncontrollably, we enforce hard bounds such that the weights remain within the interval $[0, 1]$.

Competition between synapses due to STDP leads to neurons with synapses that won the competition to fire consistently and those who lost the competition to be constantly silent [120]. To counteract this pathological state, the time-averaged firing rate for a neuron is modulated through homeostatic modification of its excitability threshold using ***i**ntrinsic **p**lasticity* (IP) [145, 133]:

$$\Delta\theta_i(t+1) = \eta_{ip}\big(x_i(t+1) - k/n\big), \tag{8}$$

where $\eta_{ip}$ is the intrinsic plasticity learning rate set to 0.001. This rule uses subtractive normalization to pull the time-averaged firing rate of each neuron closer to the population firing rate $k$.

### 2.4.3 *Computational Tasks*

Neural circuits in different brain regions adapt to best serve the region's functional purpose. To that end, we constructed three tasks, each of which resembles in spirit the demands of one such canonical function. We then, under the stimulation conditions of each task, compared the performance, information content, and dynamical response of networks optimized by combining both STDP and IP with networks that are optimized by STDP alone or IP alone.

In all tasks, the network is subject to perturbation by a set of inputs $P$. The receptive fields of non-overlapping subsets of neurons $x^{(p)}$ are tuned exclusively to each input $p \in P$. As such, each input $p$ has its corresponding receptive field $x^{(p)} = \mathrm{RF}_p$ in the recurrent neural network. When an input $p$ drives the network, all neurons $x^{(p)}$ receive a positive drive $d = 0.25$, while the rest $x \setminus x^{(p)}$ receive none. Readouts are trained on the current network state $x(t)$ to compute a function over input sequences $u_{\tau_1\tau_2}(t) = \big\langle p(t+\tau_1), \ldots, p(t+\tau_2)\big\rangle$, $\tau_1$ and $\tau_2$ being time-lags at which target inputs are applied where positive lags corresponds to future inputs and negative lags to past ones. We restrict time-lags $\tau$ to the range $\mathbb{Z} \cap [-8, 8]$.

In a first task, RAND×4, we assessed the capacity of the network to retain memory of past stimuli within its activity. The recurrent network is driven by four randomly drawn inputs $P = \{A, B, C, D\}$. The receptive field of each input consists of 15 neurons, and one optimal linear classifier $O^{(p,\tau)}$ is trained for each input/time-lag pair, i. e., $O^{(p,\tau)}$ fires when $p(t+\tau) = p$ and is silent otherwise.

The second task, Markov-85, explores the ability of the recurrent network to discover temporal regularities in its input. The recurrent network receives one of four possible inputs $P = \{A, B, C, D\}$ generated from a Markov chain with 85% probability for A to be followed by B, B followed by C, C followed by D, and D followed by A. All other transitions occur with a 5% probability. Again, the receptive

field of each input consists of 15 neurons, and one optimal linear classifier $\mathcal{O}^{(p,\tau)}$ is trained for each input/time-lag pair.

With the third task, `Parity-3`, we exploit the nonlinear expansion provided by the recurrent neural network. Here, the network is subject to binary input $P = \{0, 1\}$, where each symbol has a receptive field of 40 neurons. The task is to identify the parity of a sequence of three successive inputs. This means that given an input sequence $u_{(\tau-2)\tau}(t) = \big\langle p(t+\tau-2), p(t+\tau-1), p(t+\tau)\big\rangle$, an optimal linear classifier $\mathcal{O}^{(1,\tau)}$ fires when $g\big(u_{(\tau-2)\tau}(t)\big) = p(t+\tau-2) \oplus p(t+\tau-1) \oplus p(t+\tau) = 1$, and is silent otherwise. $\oplus$ is the nonlinear *exclusive or* binary operation.

Even though every task used here is very much simplified compared to stimuli usually processed by neural systems, we would still like to link the basic properties of every task presented here to a realistic case processed by a human or an animal. The property of the memory task `RAND×4` that we want to emphasize is that a neural system must be able to process rapidly changing stimuli that are only shortly presented. That property is partly reminiscent of retinal input, which is rather stationary during moments of fixation, and rapidly changing due to saccadic eye movements. However, it needs to be noted that saccadic eye movements might be difficult to predict and may appear rather random, but are very likely structured and stimulus-dependent. This motivated the prediction task `Markov-85` that models temporally structured and rapidly changing sensory input that is shortly presented. Such input could either be generated by retinal input and saccadic eye movements, or by the whisking behavior and the produced neural activity in the barrel cortex of a mouse. In addition, nonlinearities are prevailing in natural stimuli, and to highlight the necessity of processing these stimuli, we used the nonlinear task `Parity-3`. Such computational demands can be easily motivated by occlusion in vision, where pixel intensities do not sum up linearly at points where one object occludes another in the visual field. Again, we stress that none of these tasks is a good model of real processing in neural systems in nature. However, each is sharing individual aspects that are motivated by real life examples.

### 2.4.4 *Simulation Conditions*

In order to isolate the role of `STDP` and `IP` in shaping the computational and information processing properties of the recurrent network, we compared networks trained by both `STDP` and `IP`, with networks that are trained by `STDP` alone or `IP` alone.

Throughout all experiments, we trained networks of $n = 100$ neurons on either the `STDP+IP` condition, the `STDP` condition, or the `IP` condition for a *plasticity phase* of $t_{pl}$ time steps. For convenience, we call a *recurrent network* trained with both *synaptic* and *intrinsic* plas-

ticity SIP-RN. In contrast, we name a recurrent network that learned with a single plasticity mechanism either SP-RN or IP-RN. $k$ is set to 12, the initial weights are chosen uniformly on the interval $[0, 0.1]$ with 10% connectivity probability, and thresholds are drown from a Gaussian distribution with 0 mean and 0.1 standard deviation. Under the IP condition, to assure that weights' distribution is not different from conditions where STDP modifies the synaptic efficacies $w$, a *pre-plasticity phase* of similar length to the plasticity phase precedes the latter, where both STDP and IP are active. Afterwards, the weights structure is destroyed by random shuffling and the plasticity phase starts where IP is turned on.

In all experiments where the performance of optimal linear classifiers is estimated, the plasticity phase was $t_{pl} = 20000$ time steps long. Afterwards, weights and thresholds are held fixed, the network state is *reset* to a random initial state, and the *training phase* starts where linear classifiers are trained using linear regression on $t_{tr} = 5000$ time steps, followed by a *testing phase* of performance for another $t_{ts} = 5000$ time steps.

### 2.4.5 *Post-Plasticity Perturbation*

At the beginning of the training phase, the network state is reset to a random initial state. If the network dynamics is multistable, this resetting could bring it to a different regime than where the network was at the end of the plasticity phase. To test this possibility systematically, we perform the following post-plasticity perturbation.

Given some perturbation parameter $\pi \in \mathbb{Z} \cap [0, k]$. We assume the network state at the end of the plasticity phase is $x(t_{ps})$. Instead of randomly choosing the initial network state for the training phase, we choose a network state $x_\pi$ such that the condition $d(x_\pi, x(t_{ps})) = 2\pi$ holds. To satisfy this condition, $x_\pi$ is chosen as follows. In the network state $x(t_{ps})$, $\pi$ firing neurons and $\pi$ silent neurons are randomly selected. The $\pi$ firing neurons are then silenced and the $\pi$ silent neurons are set to firing.

### 2.4.6 *Output Weights and Performance*

According to the RC paradigm, an input signal undergoes a nonlinear feature expansion by projecting into a recurrent neural network of nonlinear units. The network recurrency also provides a sustained but damped trace of past inputs (echo state [57] or fading memory [70]) to propagate through the network. The network state is then read out by simple linear units through linear regression.

Following the plasticity phase, the network activity during the training phase

$$X_{tr} = [x^\top(t)]_{t_{pl}<t\leqslant t_{pl}+t_{tr}} \in \{0,1\}^{t_{tr}\times n} \tag{9}$$

provides the training data for all optimal linear classifiers, where $(\cdot)^\top$ denotes matrix transpose. The target signal of output neurons for a particular time-lag $\tau$ is clamped in a supervised fashion to

$$\mathcal{O}_{tr}^{(\tau)} = [\mathcal{O}^{(p,\tau)}(t+\tau)]_{t_{pl}<t\leqslant t_{pl}+t_{tr}} \in \{0,1\}^{t_{tr}\times|g|}, \tag{10}$$

where $|g|$ depends on the task and is the cardinality of the set of possible values which the target signal can take. $|g|$ equals $|P|$ for the tasks `RAND×4` and `Markov-85`. Output weights $w^o$ for each time-lag are then computed using linear regression through ordinary least squares

$$w^{(\tau)} = X_{tr}^\dagger \cdot \mathcal{O}_{tr}^{(\tau)} = (X_{tr}^\top X_{tr})^{-1} X_{tr}^\top \cdot \mathcal{O}_{tr}^{(\tau)} \in \mathbb{R}^{n\times|g|}, \tag{11}$$

where $(\cdot)^\dagger$ is the *Moore-Penrose pseudoinverse* of a matrix, and $(\cdot)^{-1}$ is the regular inverse of square matrices.

These optimal linear classifiers are then validated on the network activity

$$X_{ts} = [x^\top(t)]_{t_{pl}+t_{tr}<t\leqslant t_{pl}+t_{tr}+t_{ts}} \in \{0,1\}^{t_{ts}\times n} \tag{12}$$

during the testing phase. First, a pre-estimate of the target signal is computed for each time-lag:

$$\tilde{\mathcal{O}}^{(\tau)} = X_{ts} \cdot w^{(\tau)} \in \mathbb{R}^{t_{ts}\times|g|} \tag{13}$$

Only one output neuron fires each time step for each time-lag, and this is specified through winner-take-all on the rows of $\tilde{\mathcal{O}}^{(\tau)}$. This leads to the final estimate $\hat{\mathcal{O}}_{ts}^{(\tau)} \in \{0,1\}^{t_{ts}\times|g|}$. The *classification performance* for each time-lag is finally computed as the percentage of correct classifications:

$$\Pi^{(\tau)} = \frac{100}{t_{ts}} \cdot \sum_{t=1}^{t_{ts}} \sum_{i=1}^{|g|} \left(1 - \frac{\left|\mathcal{O}_{ts}^{(\tau)}(t,i) - \hat{\mathcal{O}}_{ts}^{(\tau)}(t,i)\right|}{2}\right) \tag{14}$$

### 2.4.7 *Computing Entropy and Mutual Information*

On multiple occasions, both the self-information capacity of the network state and its dependence on input was measured. Entropy mea-

sures self-information capacity which is the expected value of information carried by the network activity X and is given by

$$\mathcal{H}(X) = -\sum_{x \in X} p(x) \cdot \log p(x), \tag{15}$$

where $\log(\cdot)$ is the base-2 logarithm, so that entropy (and mutual information) are measured in bits. Mutual information measures the dependence of the network activity X on a corresponding input sequence $U$ and is given by

$$\mathcal{I}(U, X) = \sum_{u \in U} \sum_{x \in X} p(u, x) \cdot \log \frac{p(u, x)}{p(u) \cdot p(x)} \tag{16}$$

In computing entropy and mutual information, we used the algorithm and code developed in [64] that computes entropy from an adaptive k-nearest-neighbor estimate of probability density functions. This allows for reliable estimates of these quantities with far fewer samples in comparison to other algorithms. Nevertheless, due to the high number of channels we have (100 neurons), and to truncate unnecessary computation time, samples from the network activity are first transfered to the principal components space, and only components that carry 95% of the information are passed to the mutual information estimator.

We always considered inputs from the task `RAND`$\times$`4` and we computed the mutual information between samples of the network state $x(t)$ and the three most recent inputs $u(t) = \langle p(t-2), p(t-1), p(t) \rangle$. We encoded each of the four input symbols $P = \{A, B, C, D\}$ by a 3-bits code $\tilde{P} = \{000, 011, 101, 110\}$ to ensure equal pairwise Hamming distances between symbols. For all cases but one, as few as 5000 samples of the network state $x(t)$ and input sequence $u$ were enough to reliably estimate entropy and mutual information. The exception was computing mutual information between input and `IP-RN` activity, which demanded a higher number of samples (500000 time steps) and very long computation time, as covering 95% of the information required no less than 60 principal components.

### 2.4.8 *Autonomous Dynamics*

For a full understanding of the emerging information processing properties of the interaction of synaptic and intrinsic plasticity, it was necessary to rely on and develop concepts from the newly emerging mathematical theory of *nonautonomous dynamical systems* [63, 62]. Throughout what follows, the correspondence of the introduced concepts to our model is clarified. First, autonomous dynamics are defined, since they form a special instance of the nonautonomous case.

**Definition 2.** Let $(X, d)$ be a metric space with a metric $d$. A *discrete-time semi-dynamical system* is a function $\phi : \mathbb{Z}^+ \times X \to X$ that satisfies

1. $\phi(0, x) = x \quad \forall x \in X$.
2. $\phi(t_1 + t_2, x) = \phi\big(t_2, \phi(t_1, x)\big) \quad \forall x \in X$ and $\forall t_1, t_2 \in \mathbb{Z}^+$.
3. $\phi$ is continuous.

Equation 5 defines the driven or nonautonomous kWTA dynamics. The autonomous alternative is given by the discrete-time difference equation

$$x(t+1) = f^{(0)}(x(t)) = f\big(w \cdot x(t) - \theta\big), \tag{17}$$

where $f$ is the kWTA nonlinearity defined as above. To relate equation 17 to Definition 2, the function $\phi$ (the *solution mapping*) is chosen such that

$$\phi(t, x) = \underbrace{f \circ f \circ \cdots \circ f}_{t \text{ times}}(w \cdot x - \theta), \tag{18}$$

where $\circ$ is function composition. For $\phi$ to be an autonomous semi-dynamical system, it has to satisfy the three conditions of Definition 2. The first two conditions are trivial, as they result from the definition of function composition. We turn to prove the third condition, namely, the continuity of $\phi$. We first observe that $\phi$ is merely the $t$-fold composition of the function $f$, and since the composition of continuous functions is continuous, it is sufficient to prove the continuity of $f$.

**Proposition 3.** The kWTA function $f^{(0)}$ from equation 17 defined on the metric space $(X, d)$ is continuous, i. e.,

$$\begin{aligned}&\forall x_1 \in X \text{ and } \forall \epsilon > 0\ ; \exists \delta > 0 \text{ such that}\\ &\forall x_2 \in X \text{ with } d(x_1, x_2) < \delta,\ d\big(f^{(0)}(x_1), f^{(0)}(x_2)\big) < \epsilon \text{ holds.}\end{aligned}$$

**Proof.** For all $x_1 \in X$ and all $\epsilon > 0$, we choose $\delta(\epsilon) = 1$. For all $x_2 \in X$, if the Hamming distance $d(x_1, x_2) < 1$, $x_1$ and $x_2$ have to be equal, since the kWTA dynamics restricts the distances between any two states to the set $[0, n] \cap \{0, 2, \cdots, 2k\}$. As such, since $d$ is a metric, $d(f(x_1), f(x_2)) = 0$, which is always smaller than $\epsilon > 0$. Ergo, $f$ is continuous. □

We note that the proof to Proposition 3 becomes trivial if we consider a result from topology which states that any function from a *discrete topological space* to another is continuous. However, the proof is interesting in that it shows that $f^{(0)}$ has a stronger form of continuity, that is, $f^{(0)}$ is *uniformly continuous*, since the proof shows

that there exists a *packing radius* $r = \delta = 1 > 0$ such that either $x_1 = x_2$ or $d(x_1, x_2) > r$ for all $x_1$ and $x_2 \in X$.

With the proof of Proposition 3, we conclude that the kWTA autonomous dynamics in equation 17 generates a discrete-time semi-dynamical system. A dynamical system is a semi-dynamical system with invertible dynamics, which is not the case for kWTA networks. However, for all intents and purposes, being a semi-dynamical system is sufficient for formalizing the nonautonomous dynamics of the model network.

### 2.4.9 *Autonomous Attractors*

Characterizing the computational properties of the model neural network requires defining invariant sets and attractors.

**Definition 4.** Let $\phi : \mathbb{Z}^+ \times X \to X$ be a discrete-time semi-dynamical system generated by an autonomous difference equation $f$ on a metric space $(X, d)$. A subset $\mathcal{A} \subseteq X$ is *invariant* under $\phi$ and $f$ if $f(\mathcal{A}) = \mathcal{A}$, and is *positively invariant* if $f(\mathcal{A}) \subset \mathcal{A}$. $\mathcal{A}$ is an *attractor* of $\phi$ if the following conditions hold:

1. $\mathcal{A}$ is invariant under $\phi$ and $f$.
2. $\mathcal{A}$ is compact.
3. There exists a neighborhood $\mathcal{N}_r$ of radius $r > 0$ of $\mathcal{A}$ such that $\lim_{t\to\infty} d\big(\phi(t, \mathcal{N}_r(\mathcal{A})), \mathcal{A}\big) = 0$

For the kWTA dynamics, the second condition is assured, since $X$ is discrete and finite, which makes all subsets compact. The third condition assures that no subset of $\mathcal{A}$ satisfies the invariance and compactness conditions. Another important concept is that of a *basin of attraction* which associates each attractor with the region of the state space that converges to that attractor:

**Definition 5.** Let $\phi : \mathbb{Z}^+ \times X \to X$ be a discrete-time semi-dynamical system generated by an autonomous difference equation $f$ on a metric space $(X, d)$. The *basin of attraction* of an attractor $\mathcal{A}$ of $\phi$ is defined by

$$\mathcal{B}_{\mathcal{A}} = \{x \in X : \lim_{t\to\infty} d\big(\phi(t, x), \mathcal{A}\big) = 0\}$$

### 2.4.10 *Nonautonomous Dynamics*

Unlike autonomous (semi-)dynamical systems, the elapsed time is not sufficient to find the solution for nonautonomous dynamics: both the start and end times must be specified. Accordingly, we now define a *discrete-time nonautonomous dynamical system* as a *process*. In what follows, we will make use of the set $\mathbb{Z}^2_{\geqslant} = \{(t, t_0) \in \mathbb{Z}^2 : t \geqslant t_0\}$.

**Definition 6.** Let $(X, d)$ be a metric space with a metric $d$. A *discrete-time process* is a function $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ that satisfies

1. $\phi(t_0, t_0, x) = x \quad \forall t_0 \in \mathbb{Z}$ and $\forall x \in X$.
2. $\phi(t_2, t_0, x) = \phi\big(t_2, t_1, \phi(t_1, t_0, x)\big)$
   $\forall t_0 \leqslant t_1 \leqslant t_2 \in \mathbb{Z}$ and $\forall x \in X$.
3. $\phi$ is continuous.

We now turn to formulating the driven kWTA difference equation (see equation 5) as a discrete-time process. We first note that for a particular task, a set of possible inputs $P \cup \{0\}$ is defined. For completeness, this set covers the autonomous case by including the 0-vector. For each member of this set, we define a separate map $f^{(p)} : X \to X$ such that $f^{(p)}(x) = f\big(w \cdot x - \theta + \delta^{(p)}\big)$. The set of maps $\{f^{(p)}\}$ with cardinality $|P| + 1$ defines a family of discrete-time autonomous semi-dynamical systems. These maps are chosen either randomly for the tasks RAND×4 and Parity-3, or in a more structured fashion for the task Markov-85. In either case, the kWTA discrete-time nonautonomous dynamics in equation 5 can be rewritten in the form

$$x(t+1) = f^{(p_t)}\big(x(t)\big) = f\big(w \cdot x(t) - \theta + \delta^{(p_t)}(t)\big), \tag{19}$$

which generates a solution mapping

$$\phi\big(t_2, t_1, x(t_1)\big) = f^{(p_{t_2-1})} \circ \cdots \circ f^{(p_{t_1})}\big(x(t_1)\big) \tag{20}$$

The solution mapping $\phi$ satisfies the three properties of a process. The first two properties are a product of the definition of function composition, and the continuity condition is proven exactly as in Proposition 3. Given the above, the family of discrete-time autonomous difference equations $\{f^{(p)}\}$ on the metric space $(X, d)$ generates a process $\phi$, and thus, it defines a particular kind of nonautonomous dynamical systems termed an *input-driven dynamical system*.

It is important to point out that an input-driven dynamical system is not defined for a particular input sequence, but for all input sequences drawn from its input set. This becomes more explicit if one considers the alternative *skew product* definition of a nonautonomous dynamical system, where the input is treated as a driving autonomous dynamical system [63, 62]. We compare the two definitions of nonautonomous dynamical systems in Section 2.7. We now cover a few important concepts that will aid in defining the dynamic behavior of the model neural network.

### 2.4.11 *Nonautonomous Attractors*

Attractors in nonautonomous dynamical systems are defined on *nonautonomous sets*, relating strongly to the concepts of *invariance* and *entire solutions*.

**Definition 7.** Let $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ be a discrete-time input-driven dynamical system generated by the family of autonomous difference equations $\{f^{(p)}(x)\}$ on a metric space $(X, d)$. A subset $\mathcal{C} \subset \mathbb{Z} \times X$ is called a *nonautonomous set*, and for all $t \in \mathbb{Z}$, the set

$$\mathcal{C}_t = \{x \in X : (t, x) \in \mathcal{C}\}$$

is called the *t-fiber* of $\mathcal{C}$. $\mathcal{C}$ is said to be *invariant* under $\phi$ if $\phi(t, t_0, \mathcal{C}_{t_0}) = \mathcal{C}_t$ for all $(t, t_0) \in \mathbb{Z}^2_{\geqslant}$. An *entire solution* of $\phi$ is an invariant set under $\phi$ whose t-fibers are the singleton sets $\{\xi(t)\}$ that are the images of the function $\xi : \mathbb{Z} \to X$ such that

$$\xi(t_2) = \phi\big(t_2, t_1, \xi(t_1)\big) = f^{(p_{t_2-1})} \circ \cdots \circ f^{(p_{t_1})}\big(\xi(t_1)\big) \quad \forall (t_2, t_1) \in \mathbb{Z}^2_{\geqslant}$$

An important property of invariant nonautonomous sets is that they consist exclusively of entire solutions (for a proof, see Lemma 2.15 in [63]). Nonautonomous attractors are nonautonomous sets. As such, they consist of entire solutions as well. There are several types of attractors of nonautonomous dynamical systems. Only of interest to our model neural network are forward attractors, so we drop the qualifier 'forward' and substitute it with 'nonautonomous'.

**Definition 8.** Let $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ be a discrete-time input-driven dynamical system generated by the family of autonomous difference equations $\{f^{(p)}(x)\}$ on a metric space $(X, d)$. A nonautonomous set $\mathcal{A} \subset \mathbb{Z} \times X$ is a *nonautonomous attractor* of $\phi$ if the following conditions hold:

1. $\mathcal{A}$ is invariant under $\phi$.
2. $\mathcal{A}$ is compact.
3. There exists a neighborhood $\mathcal{N}_r$ of radius $r > 0$ such that $\lim_{t\to\infty} d\big(\phi(t, t_0, \mathcal{N}_r(\mathcal{A}_{t_0})), \mathcal{A}_t\big) = 0$ for all $t_0 \in \mathbb{Z}$

As in the autonomous dynamics of `kWTA` networks, all subsets of $X$ are compact. The third condition assures that no subset of $\mathcal{A}$ satisfies the invariance and compactness conditions. One may generalize the concept of a basin of attraction in an autonomous dynamical system to the nonautonomous case. This concept associates each nonautonomous attractor with the region of the state space that converges to that attractor:

**Definition 9.** Let $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ be a discrete-time input-driven dynamical system generated by the family of autonomous difference equations $\{f^{(p)}(x)\}$ on a metric space $(X, d)$. The *nonautonomous basin of attraction* of a nonautonomous attractor $\mathcal{A}$ of $\phi$ is defined by

$$\mathcal{B}_{\mathcal{A}} = \{x \in X : \lim_{t\to\infty} d\big(\phi(t, t_0, x), A_t\big) = 0 \text{ for all } t_0 \in \mathbb{Z}\}$$

### 2.4.12 *Volumes of Representation*

Spatiotemporal computations requires encoding different input sequences in the states of the neural network. The set of network states accessible from some initial conditions within a basin of attraction through perturbing the network with a particular input sequence $\langle p_1, \ldots, p_s\rangle$ defines this sequence's *volume of representation*.

**Definition 10.** Let $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ be a discrete-time input-driven dynamical system generated by the family of autonomous difference equations $\{f^{(p)}(x)\}$ on a metric space $(X, d)$. Given an input sequence $u_s = \langle p_1, \ldots, p_s\rangle \in \big(P \cup \{0\}\big)^s$ and a basin of attraction $\mathcal{B}_{\mathcal{A}}$, a subset

$$\begin{aligned}\mathcal{V}_{\mathcal{A}}^{u_s} = \{&x \in \mathcal{B}_{\mathcal{A}} : \exists x_0 \in \mathcal{B}_{\mathcal{A}} \text{ such that}\\ &x = \phi(s, 0, x_0) = f^{(p_s)} \circ \cdots \circ f^{(p_1)}(x_0)\}\end{aligned}$$

is called the *volume of representation* of the input sequence $u_s$ within the basin $\mathcal{B}_{\mathcal{A}}$. The sequence length $s \in \mathbb{Z}^+ \setminus \{0\}$ defines the *order* of this volume. The nonautonomous set $\mathcal{V}_{\mathcal{A}} \subset \mathbb{Z} \times \mathcal{B}_{\mathcal{A}}$ whose t-fibers are order-1 volumes of representation $\mathcal{V}_{\mathcal{A},t} = \mathcal{V}_{\mathcal{A}}^{p(t-1)}$ is called the *perturbation set* within $\mathcal{B}_{\mathcal{A}}$. Also, given a function $g : P^s \to \Omega$ on input sequences such that $g(u_s) = \omega \in \Omega$, the set

$$\mathcal{V}_{\mathcal{A}}^{\omega:g} = \bigcup_{g(u_s)=\omega} \mathcal{V}_{\mathcal{A}}^{u_s}$$

is the volume of representation of $\omega$ given $g$.

It is straightforward to show that, within a basin of attraction, the volume of representation of some sequence $u_s = \langle p_1, \ldots, p_s\rangle$ is a superset of the volume of a sequence $u_{s'+s} = \langle q_1, \ldots, q_{s'}, p_1, \ldots, p_s\rangle$ for all $u_{s'} = \langle q_1, \ldots, q_{s'}\rangle \in \big(P \cup \{0\}\big)^{s'}$, and that the volume of $u_s$ is equivalent to the union of the volumes of $u_{s'+s}$ for all $u_{s'} \in \big(P \cup \{0\}\big)^{s'}$. We term this property the *volumes' inclusion property*.

The concept of 'volumes of representation' allows us to state the following theorem on the nature of attractors in discrete-time input-driven dynamical systems:

**Theorem 11.** Let $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ be a discrete-time input-driven dynamical system generated by the family of autonomous difference

equations $\{f^{(p)}(x)\}$ on a metric space $(X, d)$, and let $\mathcal{B}_{\mathcal{A}}$ be a *compact* nonautonomous basin of attraction. The following holds:

1. The perturbation set $\mathcal{V}_{\mathcal{A}}$ is a superset of $\mathcal{A}$.
2. Within $\mathcal{B}_{\mathcal{A}}$, and for all $p \in P \cup \{0\}$, there exists one attractor $\mathcal{A}^{(p)} \subset \mathcal{V}_{\mathcal{A}}^{p}$ of the discrete-time autonomous semi-dynamical system generated by $f^{(p)}$.
3. $\mathcal{B}_{\mathcal{A}}$ is the basin of attraction of $\mathcal{A}^{(p)}$ for all $p \in P \cup \{0\}$.

**Proof.**

1. Since every attractor, whether autonomous or nonautonomous, is an invariant set, it is sufficient to prove that all invariant sets within a basin $\mathcal{B}_{\mathcal{A}}$ are a subset of its perturbation set $\mathcal{V}_{\mathcal{A}}$. Let's consider an entire solution $\xi : \mathbb{Z} \to \mathcal{B}_{\mathcal{A}}$. For all $t_1 \in \mathbb{Z}$, it holds that $\xi(t_1 + 1) = \phi\big(t_1 + 1, t_1, \xi(t_1)\big) = f^{(p_{t_1})}\big(x(t_1)\big) \in \mathcal{V}^{p(t_1)}$. It follows by induction that $\xi(t_2 + 1) \in \mathcal{V}^{p(t_2)}$ for all $t_2 \geqslant t_1$. This translates to t-fibers of entire solutions being always a member of order-1 volumes of representation and that all entire solutions within a basin of attraction $\mathcal{B}_{\mathcal{A}}$ are subsets of the perturbation set $\mathcal{V}_{\mathcal{A}}$. Since invariant sets consist exclusively of entire solutions, it follows that all invariant sets are subsets of the perturbation set $\mathcal{V}_{\mathcal{A}}$, including the nonautonomous attractor $\mathcal{A}$.
2. Given some input $p$, we consider the discrete-time semi-dynamical system generated by $f^{(p)}$ on $(X, d)$ with the solution mapping

$$\varphi(t, x) = \underbrace{f^{(p)} \circ f^{(p)} \circ \cdots \circ f^{(p)}}_{t \text{ times}}(x)$$

From Definition 10 of volumes of representation, the order-$s$ volume generated by $p$ is the set $\mathcal{V}_{\mathcal{A}}^{p^s} = \varphi(s, \mathcal{B}_{\mathcal{A}}) \subseteq \mathcal{B}_{\mathcal{A}}$. Due to the compactness of $\mathcal{B}_{\mathcal{A}}$ and the continuity of $f^{(p)}$, $\mathcal{V}_{\mathcal{A}}^{p^s}$ is compact for all $s \in \mathbb{Z}^+$. Moreover, due to the volumes' inclusion property, the family of compact volumes $(\mathcal{V}_{\mathcal{A}}^{p^s})_{s \in \mathbb{Z}^+}$ is nested with $\mathcal{V}_{\mathcal{A}}^{p^0} = \mathcal{B}_{\mathcal{A}}$. As such, and according to Theorem 1.28 in [63], there exists a nonempty set

$$\mathcal{A}^{(p)} = \bigcap_{s \geqslant 0} \mathcal{V}_{\mathcal{A}}^{p^s}$$

that is both *compact* and *invariant* under $f^{(p)}$ and $\varphi$. It also follows from the compactness of $\mathcal{B}_{\mathcal{A}}$ that $\mathcal{A}^{(p)}$ attracts $\mathcal{B}_{\mathcal{A}}$, i. e., there exists $r > 0$ such that $\lim_{t \to \infty} d\big(\varphi(t, \mathcal{N}_r(\mathcal{B}_{\mathcal{A}})), \mathcal{A}^{(p)}\big) = 0$,

and since $\mathcal{A}^{(p)}$ is a subset of $\mathcal{B}_{\mathcal{A}}$, the neighborhood $\mathcal{N}_r(\mathcal{B}_{\mathcal{A}})$ is also a neighborhood of $\mathcal{A}^{(p)}$. Hence, the compact and invariant set $\mathcal{A}^{(p)}$ is an attractor of the discrete-time semi-dynamical system generated by $f^{(p)}$ and is a subset of $\mathcal{V}^p_{\mathcal{A}}$.

3. Since $\mathcal{A}^{(p)}$ attracts $\mathcal{B}_{\mathcal{A}}$, it follows that the basin of attraction of $\mathcal{A}^{(p)}$ satisfies $\mathcal{B}_{\mathcal{A}^{(p)}} \supseteq \mathcal{B}_{\mathcal{A}}$. Given a point $x \in \mathcal{B}_{\mathcal{A}^{(p)}} \setminus \mathcal{B}_{\mathcal{A}}$, and since $\mathcal{A}^{(p)} \subset \mathcal{V}^p_{\mathcal{A}}$, there exists $t \in \mathbb{Z}^+$ such that $\varphi(t, x) \in \mathcal{V}^p_{\mathcal{A}} \subseteq \mathcal{B}_{\mathcal{A}}$, which is a contradiction, since $x \notin \mathcal{B}_{\mathcal{A}}$. Ergo, $\mathcal{B}_{\mathcal{A}^{(p)}} \setminus \mathcal{B}_{\mathcal{A}}$ is an empty set, and $\mathcal{B}_{\mathcal{A}}$ is the basin of attraction of $\mathcal{A}^{(p)}$ □

This theorem allows us to characterize the properties and relations between autonomous and nonautonomous attractors of kWTA networks, where all subsets of $X$ are compact due to $X$'s finiteness and discreteness. Namely, it allows us, within some compact basin, to allocate the nonautonomous attractor's t-fibers as subsets of the t-fibers of the perturbation set, and it shows that the autonomous attractor of the input at time $t$ is the t-fiber of the nonautonomous attractor.

### 2.4.13 *Input-Insensitive Dynamics*

It is possible for a process to behave locally or globally as an autonomous (semi-)dynamical system. That is equivalent, in the case of input-driven dynamical systems, to being input-insensitive.

**Definition 12.** Let $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ be a discrete-time input-driven dynamical system generated by the family of autonomous difference equations $\{f^{(p)}(x)\}$ on a metric space $(X, d)$. A state $x \in X$ is said to be *input-insensitive* if $f^{(p)}(x) = f^{(0)}(x)$ for all $p \in P$. An *input-insensitive basin* is a basin of attraction that consists entirely of input-insensitive states.

This definition implies that the volumes of representation of a particular order and the t-fibers of each nonautonomous set within this basin are equivalent, including the perturbation set and the nonautonomous attractor: they reduce to autonomous sets. The *input-insensitive attractor* becomes the autonomous attractor of each discrete-time semi-dynamical system generated by a difference equation $f^{(p)}$.

## 2.5 APPENDIX A: COMPARING NONPLASTIC NETWORKS

Comparing SIP-RNs and nonplastic networks requires generating random networks with reasonable weight and threshold distributions. Usually, the spectral radius of the weight matrix is scaled "appropriately" in the reservoir computing community, such that the recurrent network achieves the echo state property, which assures a decaying memory of the network's input. Unfortunately, no heuristics from the reservoir computing literature exist for kWTA networks. Using the

spectral radius scaling as a rule of thumb is no guarantee to make the network better or worse, since it has no theoretical grounding of its effect without taking the model neural network into account, along with its input statistics. Sufficient conditions for the existence of the echo-state property, and that relates its existence with certain input statistics and spectral properties of the weight matrix, is only formally stated for hyperbolic tangent sigmoid neurons, as is proven in [71]. Given the above, we see no existing heuristics in the literature for selecting weights and thresholds of our kWTA networks and our given input distributions, and using any would jeopardize the formality of our treatment.

For a fair comparison between SIP-RNs and nonplastic networks, we devised a procedure for generating random networks that have the same weight and firing threshold distributions as plastic networks. In a first set of networks, weights were trained with STDP and then shuffled to assure the destruction of structure, while keeping the weight distribution comparable to SP-RNs. These networks showed significantly lower average performance than SIP-RNs in all tasks, as is demonstrated in Figure 2.

In order to show that the high performance of SIP-RNs is a result of the structure imposed by synaptic and intrinsic plasticity on the network' connectivity and the neurons' firing thresholds, we trained a second set of networks on both plasticity mechanisms. We then randomly shuffled both their weights and thresholds so that they have the same weight and threshold distributions as SIP-RNs. The performance of these networks, compared to networks where only the weights were trained and shuffled, depends on the task. They perform better on the prediction task, but worse on the memory and nonlinear tasks (see Figure 13). Nevertheless, they still show significantly lower average performance than SIP-RNs.

It might still be possible to find, in random networks, ones that are comparable or ones that outperform our best SIP-RNs, but we suppose that it is expensive for the genetic code to determine these networks, and therefore brain plasticity was the solution for this limitation. These hypothetical networks might also not have properties of the input statistics embedded in their structure and spontaneous behavior, as our plastic networks show. They also might not be generically better for all stimuli, while our networks adapt to their stimuli.

## 2.6 APPENDIX B: LONG-TERM BEHAVIOR OF LEARNING

A limitation of the current model is the nonconvergence of learning. As Figure 3 shows, the SIP-RNs mutual information with their input degrades for longer plasticity exposure, which results in a decrease in their computational performance. This degradation is due to the unoptimized time scales at which synaptic and intrinsic plasticity are

operating. At some point, intrinsic plasticity becomes unable to keep up with synaptic plasticity that reinforces the input-insensitive dynamics. A possible solution would be to implement some form of *metaplasticity* [3] that operates on the learning rates of the plasticity mechanisms, increasing or decreasing their effects, as necessary, or to introduce synaptic scaling as an additional mechanism that prevents the input-insensitive regime from dominating. However, either solution would make the model more complicated than intended for this work, where we aimed at providing, for the first time, a nonautonomous dynamical systems treatment on the roles and interaction of `STDP` and `IP` for spatiotemporal computations in recurrent neural networks.

## 2.7 APPENDIX C: DEFINITIONS OF NONAUTONOMOUS DYNAMICAL SYSTEMS

The following definitions are adapted from [63, 62].

The *process* definition of a discrete-time nonautonomous dynamical system can be stated as follows:

**Definition C1.** Let $(X, d)$ be a metric space with a metric $d$. A *discrete-time process* is a function $\phi : \mathbb{Z}^2_{\geqslant} \times X \to X$ that satisfies

1. $\phi(t_0, t_0, x) = x \quad \forall t_0 \in \mathbb{Z}$ and $\forall x \in X$.
2. $\phi(t_2, t_0, x) = \phi\big(t_2, t_1, \phi(t_1, t_0, x)\big)$
   $\forall t_0 \leqslant t_1 \leqslant t_2 \in \mathbb{Z}$ and $\forall x \in X$.
3. $\phi$ is continuous.

One may reformulate a discrete-time process as a discrete-time autonomous semi-dynamical system on the extended state space $\mathcal{X} := \mathbb{Z} \times X$ with the mapping

$$\zeta\big(t, (t_0, x_0)\big) := \big(t + t_0, \phi(t + t_0, t_0, x_0)\big) \text{ for all } \big(t, (t_0, x_0)\big) \in \mathbb{Z}^+_0 \times \mathcal{X}$$

However, no compact set in $\mathcal{X}$ is invariant under $\zeta$, which necessitates extending the concepts of invariance and attractivity in $\phi$, as we show in the main text.

An alternative formulation of nonautonomous dynamical systems is given by a *skew product flow*

**Definition C2.** Let $(X, d_X)$ and $(P, d_P)$ be metric spaces with metrics $d_X$ and $d_P$. A *discrete-time nonautonomous dynamical system* $(\psi, \varphi)$ is defined as follows:

- a *discrete-time dynamical system* $\psi$ on a parameter space $P$ is a group of homeomorphisms $(\psi_t)_{t\in\mathbb{Z}}$ under composition on $P$ such that

1. $\psi_0(p) = p$ for all $p \in P$.
2. $\psi_{t_1+t_2} = \psi_{t_2}\big(\psi_{t_1}(p)\big)$ for all $t_1, t_2 \in \mathbb{Z}$.
3. the mapping $(t, p) \mapsto \psi_t(p)$ is continuous.

- and a *cocycle mapping* $\varphi : \mathbb{Z}_0^+ \times P \times X \to X$ that satisfies

1. $\varphi(0, p, x) = x$ for all $(p, x) \in P \times X$.
2. $\varphi(t_1 + t_2, p, x) = \varphi\big(t_2, \psi_{t_1}(p), \varphi(t_1, p, x)\big)$ for all $t_1, t_2 \in \mathbb{Z}_0^+, (p, x) \in P \times X$.
3. the mapping $(t, p, x) \mapsto \varphi(t, p, x)$ is continuous.

As with the process formulation, one may define a discrete-time autonomous semi-dynamical system $\zeta : \mathbb{Z}_0^+ \times \mathcal{X} \to \mathcal{X}$ on the extended state space $\mathcal{X} := P \times X$. $\zeta$ is called the *skew product flow* associated with the nonautonomous semi-dynamical system $(\psi, \varphi)$ on $\mathcal{X}$, and is given by

$$\zeta\big(t, (p, x)\big) := \zeta\big(\theta_t(p), \varphi(t, p, x)\big)$$

The skew product flow of the systems we considered here is more specific, since it falls under the class of *random dynamical systems* [7], the treatment of which at this point would only serve in complicating the analysis. However, with its general form presented above, the skew product formulation is more intuitive than the process formulation regarding input-driven dynamical systems, since the input dynamics in the former is explicitly stated by the driving dynamical system $\psi$, and is only implicitly considered in the case of a process. On the other hand, the process formulation is simpler and is sufficient for drawing the results and conclusions of the main text.

ACKNOWLEDGMENTS

Discussions with Johannes Schumacher are gratefully acknowledged. The authors acknowledge the financial support of the State of Lower Saxony, Germany via the University of Osnabrück, and the European project PHOCUS in the Framework 'Information and Communication Technologies' (FP7-ICT-2009-C / Proposal Nr. 240763).

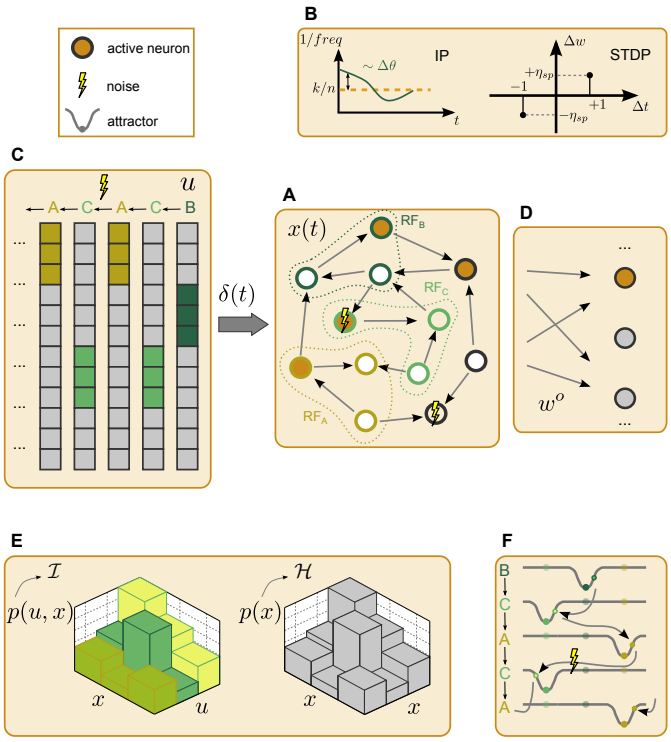


Figure 1: **Overview of the recurrent network model and the methods for analyzing its computational capabilities.** (A) An exemplary recurrent neural network of 12 neurons. The network state $x$ has a 4-Winner-Take-All (`4WTA`) nonlinear dynamics, where the 4 neurons with the highest membrane potential fire and the rest are silent. The membrane potential is the sum of the recurrent afferents and the external drive $\delta$. It is also depolarized (hyperpolarized) with decreasing (increasing) excitability threshold $\theta$. The recurrent network can also be subject to *noise*, while reserving the `4WTA` dynamics: when a neuron fails to spike due to noise, another fires instead. (B) The recurrent network is adapted by two plasticity mechanisms. The excitability threshold $\theta$ is modulated by ***i**ntrinsic **p**lasticity* (`IP`), the recurrent afferents $w$ by ***s**pike-**t**iming-**d**ependent synaptic **p**lasticity* (`STDP`). (C) The external drive $\delta$ consists of discrete symbols that follow a certain stochastic dynamics, and each projects to a corresponding ***r**eceptive **f**ield* (RF). The exemplary drive is a 3-symbols Markov chain A $\to$ B $\to$ C that allows a probability for *noisy* transitions, i. e., A $\to$ C. (D) Linear functions of the network state $x$ parametrized by output weights $w^o$ fitted to (possibly nonlinear) target functions of sequences of the external drive. (E) Nonlinear information-theoretic quantities are measured: network state entropy $\mathcal{H}$ and the mutual information $\mathcal{I}$ of the network state $x$ and input sequence $u$. (F) Analysis of the appearance and disappearance of attractors due to the external drive within the network as an input-driven dynamical system.

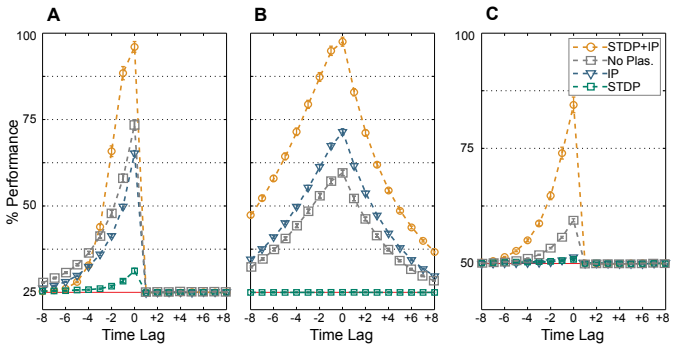


Figure 2: **Average classification performance.** 100 networks are trained by `STDP` and `IP` simultaneously (orange), `IP` alone (blue), `STDP` alone (green), or are nonplastic (gray). Optimal linear classifiers are then trained to perform (A) the memory task `RAND`$\times 4$, (B) the prediction task `Markov-85`, and (C) the nonlinear task `Parity-3`. Nonplastic networks have their weights trained by `STDP` and then randomly shuffled, so that they have the same weight and threshold distributions as `SP-RNs`. However, due to the shuffling, their weight matrices carry no structure. Error bars indicate standard error of the mean. The red line marks chance level. The x-axis shows the input time-lag. Negative time-lags indicate the past, and positive ones, the future.

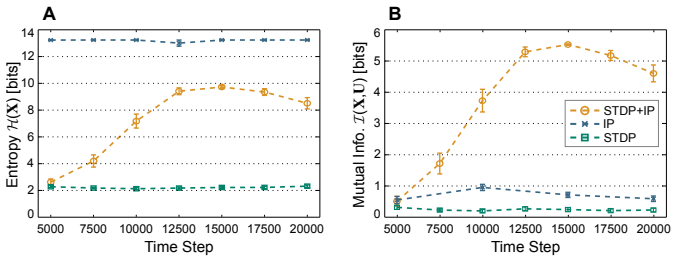


Figure 3: **Network state entropy and the mutual information with input.** (A) Network state entropy $\mathcal{H}(X)$ and (B) the mutual information with the three most recent `RAND`$\times 4$ inputs $\mathcal{I}(U, X)$ as they develop through the plasticity phase for `SP-RNs` (green), `IP-RNs` (blue), and `SIP-RNs` (orange). Mutual information for `IP-RNs` is estimated from 500000 time steps, and is averaged over 5 networks only. Other values are averaged over 50 networks and estimated from 100000 samples for each network. Error bars indicate standard error of the mean.

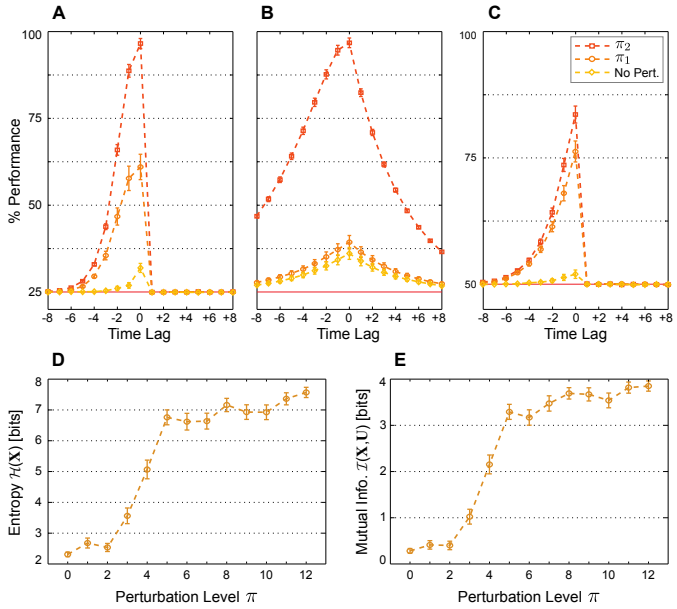


Figure 4: **Post-plasticity perturbation.** 100 networks are trained by `STDP` and `IP` simultaneously on (A) the memory task `RAND`$\times$`4`, (B) the prediction task `Markov-85`, and (C) the nonlinear task `Parity-3` with increasing perturbation level: $\pi = 0$ (yellow), $\pi = 4$ (orange), and $\pi = 12$ (red). Error bars indicate standard error of the mean. The red line marks chance level. The x-axis shows the input time-lag. Negative time-lags indicate the past, and positive ones, the future. (D) Network state entropy $\mathcal{H}(X)$ and (E) the mutual information with the three most recent `RAND`$\times$`4` inputs $\mathcal{I}(U, X)$ at the end of the plasticity phase for different perturbation levels. Values are averaged over 50 networks and estimated from 5000 samples for each network. Error bars indicate standard error of the mean.

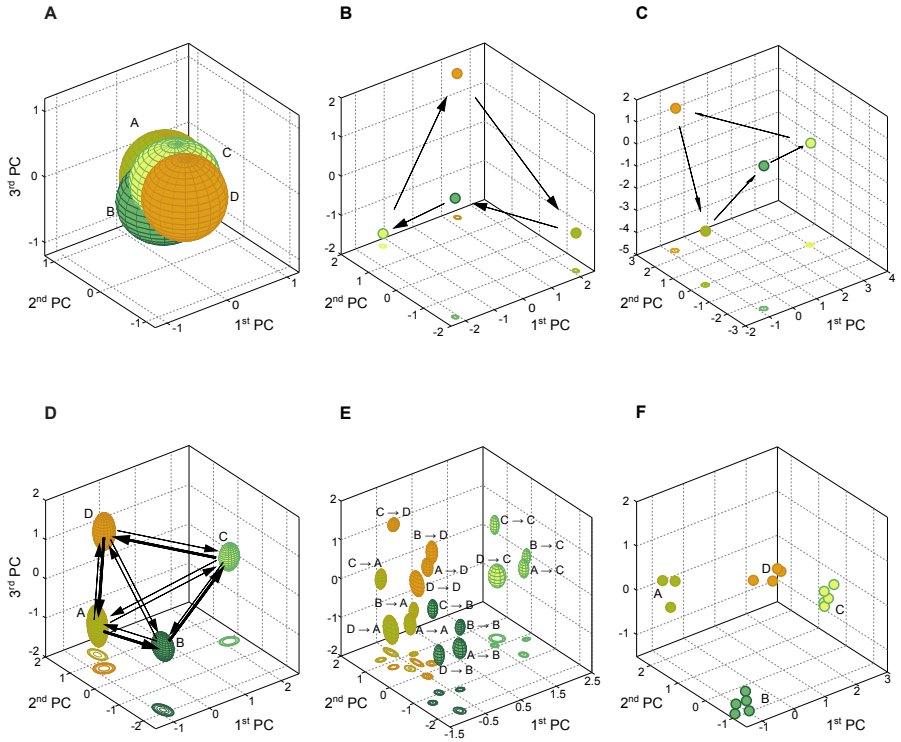


Figure 5: **Plasticity effects on networks dynamics and input representations under the prediction task input.** The three dimensions correspond to the first three principal components (PCs) of the network activity. (A) Highly-overlapping order-1 volumes of representation of an `IP-RN`. (B) Input-insensitive global attractor of a `SP-RN` that corresponds to a minimal code. (C) With no perturbed ($\pi = 0$), a `SIP-RN` dynamics also converges to an input-insensitive attractor and exhibits a minimal code. (D) Approximate visualization of order-1 volumes of representation of a `SIP-RN`. The approximation uses the means and the standard deviations of the corresponding coordinates of the network activity in the principal components space as the center and semi-axes lengths of ellipsoids. Arrows correspond to the transitions from one input symbol to the other. Their thickness symbolizes the probability of a transition, which reflects the `Markov-85` transition probability. The collection of volumes of representation and the arrows show the perturbation set within which the nonautonomous attractor resides. (E) Order-2 volumes of representation of a `SIP-RN` also approximated using the mean and standard deviations of coordinates. Order-2 volumes are more exact approximations to the order-1 representations according to the volumes' inclusion property. The correspondence is clarified by using similar color coding. (F) Autonomous periodic attractors of a `SIP-RN`, each belonging to one of the autonomous semi-dynamical systems associated with one `Markov-85` input. For clarity, no arrows are drawn between the vertexes of an attractor.

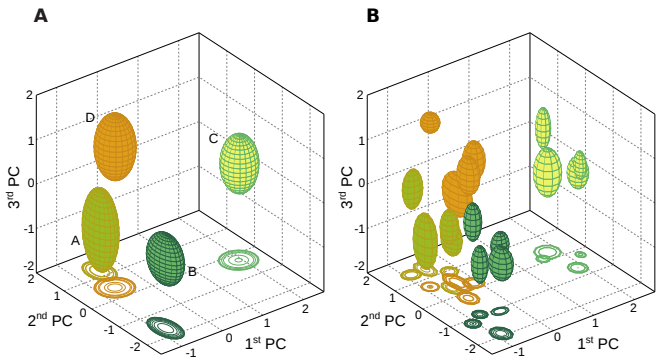


Figure 6: **Approximating volumes of representation using percentiles.** (A) Percentile approximation of the order-1 volumes of representation of a `SIP-RN`. (B) Percentile approximation of the order-2 volumes of representation of a `SIP-RN`. Order-2 volumes are more exact approximations to the order-1 volumes according to the volumes' inclusion property. The correspondence is clarified by using similar color coding. (A,B) This approximation is done as follows. After transforming the network states to the principal components space, the coordinates of the first three principal components belonging to each volume of representation are first bootstrapped to 10000 samples, and the $5^{th}$ and $95^{th}$ percentiles are computed. Each volume is then approximated by an ellipsoid whose semi-axes extend to these percentiles and is centered at their average. This alternative approximation is less liberal than the one that uses means and standard deviations in that it extends the ellipsoids to assure including more true positives, but at the expense of including more false positives. One still sees, however, that the observations from the other approximation still hold, namely, that volumes of representation are both redundant and separate from one another.

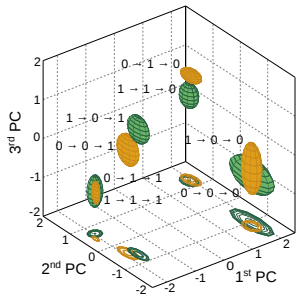


Figure 7: **Volumes of representation of a nonlinear function over input sequences.** Approximation of order-3 volumes of representation of the task `Parity-3` binary input to a `SIP-RN`. By an appropriate union of these volumes, the volumes of representation of the outcome 0 (green) and 1 (orange) are identified. The approximation uses the mean and standard deviation of the coordinates. While the first three principal components are sufficient for showing distinct order-3 volumes of representation, more dimensions are necessary to illustrate separate volumes of the outcome of the nonlinear function. The separability of the function's outcomes explains the ability of optimal linear classifiers to successfully perform the nonlinear task.

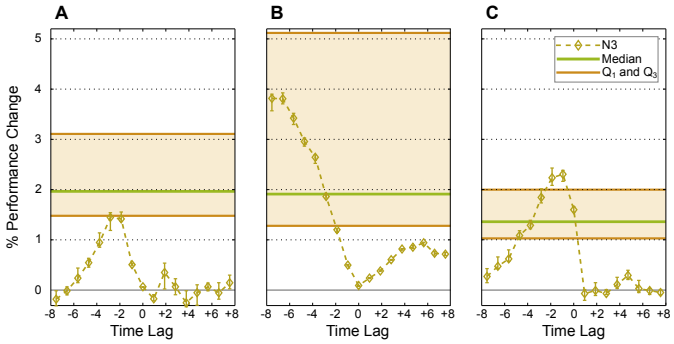


Figure 8: **Noise-robustness is achieved through the interaction of synaptic and intrinsic plasticity.** Bootstrapped median relative change from the noiseless performance of 100 networks trained with both `STDP` and `IP` on (A) the memory task `RAND`$\times 4$, (B) the prediction task `Markov-85`, and (C) the nonlinear task `Parity-3`. High perturbation of $\pi = 12$ is applied at the end of the plasticity phase. Error bars correspond to the $25^{\text{th}}$ and the $75^{\text{th}}$ percentiles. Noise level $N3 = 3\%$ is the probability of a bit flip in the network state, that is, the probability of one of the $k$ spiking neurons at time step $t$ to become silent, while a silent neuron fires instead. The shaded area indicates the ratio of noisy spikes which is measured in comparison to the noiseless `SIP-RN`s. The green line indicates the median and the orange lines the $25^{\text{th}}$ and the $75^{\text{th}}$ percentiles of the noisy spikes ratio.

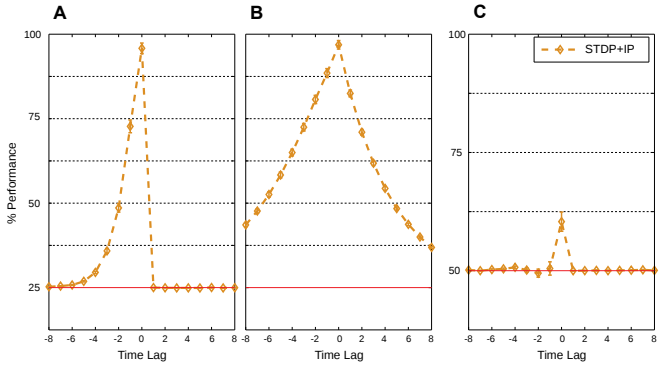


Figure 9: **Average classification performance using the Hamming distance of the network states from the vertexes of autonomous attractors.** 100 networks are trained by `STDP` and `IP` simultaneously on (A) the memory task `RAND`$\times$`4`, (B) the prediction task `Markov-85`, and (C) the nonlinear task `Parity-3`. Given the input set $P$, and the family of discrete-time autonomous semi-dynamical systems generating these networks $\{f^{(p)}(x)\}$, the network states comprising the autonomous attractor (the attractor's vertexes) are identified as follows. First, initial conditions are selected within the input-sensitive basin of attraction. Second, the input is clamped to one member of $P$. Third, the solution of $f^{(p)}(x)$ is generated for a sufficient number of time steps, so that the dynamics, following a transient period, converges to the attractor. Training and testing optimal linear classifiers is carried through as before. The training and testing data is, however, the Hamming distance between the network states and the vertexes of the attractors. Error bars indicate standard error of the mean. The red line marks chance level. The $x$-axis shows the input time-lag. Negative time-lags indicate the past, and positive ones, the future.

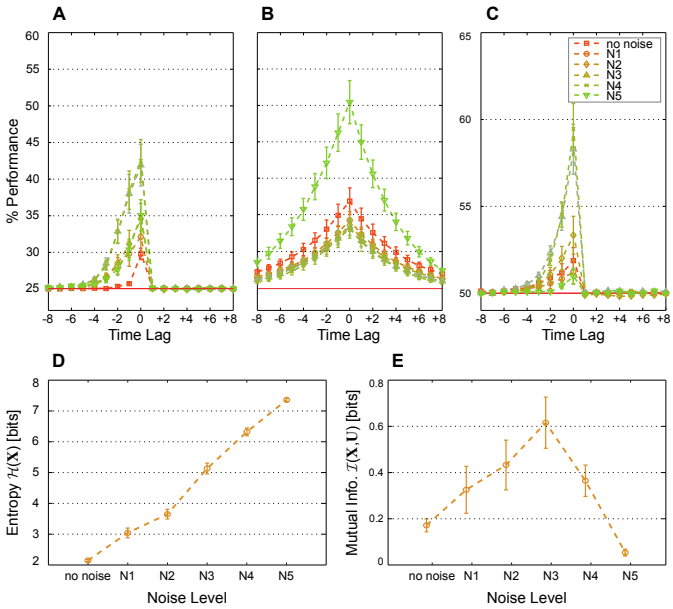


Figure 10: **Noise at certain levels is rendered constructive when synaptic and intrinsic plasticity interact.** Average classification performance of 100 networks trained with both `STDP` and `IP` on (A) the memory task `RAND`×4, (B) the prediction task `Markov-85`, and (C) the nonlinear task `Parity-3` for increasing levels of noise and no perturbation at the end of the plasticity phase ($\pi = 0$). (D) Network state entropy $\mathcal{H}(X)$ and (E) the mutual information with the three most recent `RAND`×4 inputs $\mathcal{I}(U, X)$ at the end of the plasticity phase for different levels of noise. Values are averaged over 50 networks and estimated from 5000 samples for each network. (A–E) Noise levels are applied during the plasticity, training, and testing phases. They indicate the probability of a bit flip in the network state, that is, the probability of one of the $k$ spiking neurons at time step $t$ to become silent, while silent neuron to fire instead. $N1 = 0.6\%$, $N2 = 1.2\%$, $N3 = 3\%$, $N4 = 6\%$, and $N5 = 12\%$. Error bars indicate standard error of the mean.

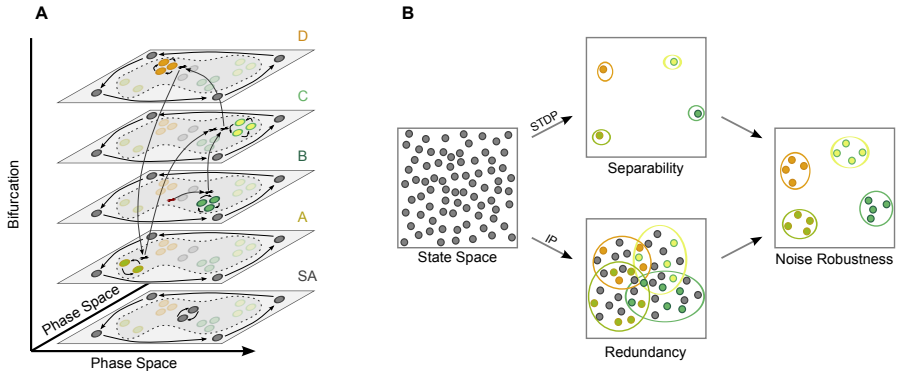


Figure 11: **Schematics of the driven dynamics of networks endowed by synaptic and homeostatic plasticity, and the emergence of noise-robust spatiotemporal computations.** (A) The dynamics of a recurrent network that is trained by homeostatic and synaptic plasticity and driven by a Markovian input. Each layer corresponds to one input. The layer illustrates a two-dimensional projection of the phase space of the autonomous (semi-)dynamical system associated with that input. A layer that corresponds to the spontaneous activity (SA) is added for completeness. Due to the interaction of synaptic and homeostatic plasticity, each of these (semi-)dynamical systems has two dynamic regimes: an input-insensitive dynamic regime that is shared by all the layers and that captures the temporal structure of the input, and an input-sensitive dynamic regime that contains a single periodic attractor. The input-sensitive attractor depends on the layer and is close to one of the vertexes of the input-insensitive attractor. The network is excited by the exemplary input sequence $B \to C \to D \to A \to C$. The red cross refers to the initial conditions that are chosen within the input-sensitive dynamics. Given the input sequence, the network dynamics follows the meta-transient that is illustrated by the arrows between the different layers. For instance, when the network is excited by the input B, the network activity approaches the B-attractor within the corresponding layer. When C follows, a bifurcation occurs, where the B-attractor becomes unstable and the C-attractor becomes stable. The meta-transient approaches the C-attractor from the direction of the unstable B-attractor. When C is preceded by the less common input A, the C-attractor is approached differently, such that the distance to it is bigger than in the case of the most common transition $B \to C$. (B) Noise-robust computations are a result of the interaction between synaptic and homeostatic intrinsic plasticity. Synaptic plasticity leads to high separability and intrinsic plasticity to redundancy. These effects lead to a neural code that allows a higher margin of noise and alternative representations of computations, thus facilitating noise-robustness.

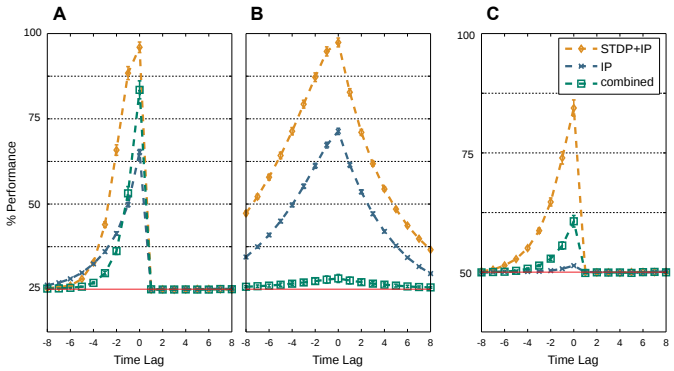


Figure 12: **Average classification performance of networks combining the weights of SP-RNs and thresholds of IP-RNs.** 100 networks are trained by STDP and IP simultaneously (orange), IP alone (blue), or trained by STDP alone followed by injecting the thresholds resulting from IP at the end of the plasticity phase (green) on (A) the memory task RAND×4, (B) the prediction task Markov-85, and (C) the nonlinear task Parity-3. The combined networks (green) lack the contribution of the interaction between synaptic and intrinsic plasticity during the plasticity phase. This results in their performance being inferior to the networks where synaptic and intrinsic plasticity interact. Error bars indicate standard error of the mean. The red line marks chance level. The x-axis shows the input time-lag. Negative time-lags indicate the past, and positive ones, the future.

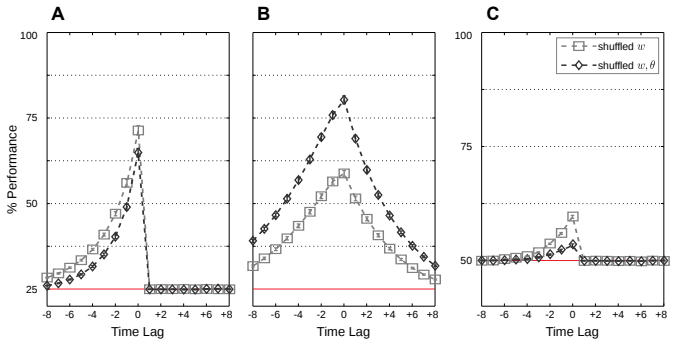


Figure 13: **Performance of random networks.** 100 networks are either trained by STDP alone or by both STDP and IP.The weights $w$ in the first group were shuffled (gray), while both the weights $w$ and the thresholds $\theta$ were shuffled in the second group (black). Optimal linear classifiers are then trained to perform (A) the memory task RAND×4, (B) the prediction task Markov-85, and (C) the nonlinear task Parity-3. Error bars indicate standard error of the mean. The red line marks chance level. The x-axis shows the input time-lag. Negative time-lags indicate the past, and positive ones, the future.

# 3

## PAPER II | HOMEOSTATIC PLASTICITY FOR SINGLE NODE DELAY-COUPLED RESERVOIR COMPUTING

ABSTRACT Supplementing a differential equation with delays results in an infinite dimensional dynamical system. This property provides the basis for a reservoir computing architecture, where the recurrent neural network is replaced by a single nonlinear node, delay-coupled to itself. Instead of the spatial topology of a network, subunits in the delay-coupled reservoir are multiplexed in time along one delay span of the system. The computational power of the reservoir is contingent on this temporal multiplexing. Here, we learn optimal temporal multiplexing by means of a biologically-inspired homeostatic plasticity mechanism. Plasticity acts locally and changes the distances between the subunits along the delay, depending on how responsive these subunits are to the input. After analytically deriving the learning mechanism, we illustrate its role in improving the reservoir's computational power. To this end, we investigate, firstly, the increase of the reservoir's memory capacity. Secondly, we predict a NARMA-10 time series, showing that plasticity reduces the normalized root-mean-square error by more than 20%. Thirdly, we discuss plasticity's influence on the reservoir's input-information capacity, the coupling strength between subunits, and the distribution of the readout coefficients.

**Keywords:** Reservoir computing, delay, self-coupling, homeostatic plasticity, sensitivity, entropy, spatiotemporal computations

### 3.1 INTRODUCTION

***R**eservoir **c**omputing*, or RC for short [57, 70, 19, 69], is a computational paradigm that provides both a model for neural information processing [46, 60, 142, 84], and powerful tools to carry out a variety of spatiotemporal computations. This includes time series forecasting [58], signal generation [59], pattern recognition [135], and information storage [91]. RC also affords a framework for advancing and refining our understanding of neuronal plasticity and self-organization in recurrent neural networks [67, 68, 130].

This article presents a biologically inspired neuronal plasticity rule to boost the computational power of a novel RC architecture that is called a *single node **d**elay-**c**oupled **r**eservoir*, or DCR for short. The DCR realizes the same RC concepts using a single nonlinear node with *delayed feedback* [5]. This simplicity makes the DCR particularly appeal-

ing for physical implementations, which has already been demonstrated on electronic [5], optoelectronic [66, 90], and all-optical hardware [17]. The optoelectronic and all-optical implementations utilizes a semiconductor laser diode as the nonlinear node, and an optical fiber as a delay line, allowing them to maintain high sampling rates. They are also shown to compare in performance to standard RC architectures in benchmark computational tasks.

The DCR operates as follows. Different nonlinear transformations and mixing of stimuli from the past and the present are achieved by sampling the DCR's activity at *virtual nodes*, or *v-nodes*, along the delay line. While neurons of a recurrent network are mixing stimuli via their synaptic coupling, which forms a network topology, the v-nodes of a DCR are mixing signals via their (nonlinear) temporal interdependence. Therefore, the v-nodes' temporal distances from one-another, henceforth termed *v-delays*, are made shorter than the characteristic time scale of the nonlinear node. Thus, v-nodes become analogous to the connections of a recurrent network, providing the DCR with a certain network-like topology. In analogy to the *spatial multiplexing* in a classical reservoir, stimuli in a DCR are *temporally multiplexed* (see Figure 14). To process information, the external stimuli are applied to the dynamical system, thereby perturbing the reservoir dynamics. Here, we operate the nonlinear node in an asymptotically stable fixed point regime. To render the response of the DCR transient, i. e., reflecting nonlinear combinations of past and present inputs, the reservoir dynamics must not converge to the fixed point, where it becomes dominated by the current stimulus. To ensure this, a random piecewise constant masking sequence is applied to the stimulus before injecting the latter to the reservoir [5]. The positions where this *mask* may switch value match the positions of the v-nodes, which are initially chosen *equidistant*. However, given the fact that the v-delays directly influence the interdependence of the corresponding v-nodes states, and therefore the nonlinear mixing of the stimuli, it is immediately evident that v-delays are important parameters that may significantly influence the performance of the DCR.

To optimize the computational properties of the DCR, we employ neuroscientific principles using biologically-inspired *homeostatic plasticity* [28, 145, 133] for adjusting the v-delays. Biologically speaking, homeostatic plasticity does not refer to a single particular process. It is rather a generic term for a family of adaptation mechanisms that regulate different components of the neural machinery, bringing these components to a functionally desirable operating regime. The choice of the operating regime depends on the functionality a model of homeostatic plasticity aims to achieve. This resulted in many flavors of homeostatic plasticity for regulating recurrent neural networks in computational neuroscience [119, 121, 104, 67, 68, 74, 100, 82, 146, 130], neurorobotics [141, 134, 52, 27, 129], and reservoir com-

puting [113, 27]. Here, we use a homeostatic plasticity mechanism to regulate the v-delays so as to balance responsiveness to the input and its history on the one hand, against optimal expansion of its informational features into the high dimensional phase space of the system on the other hand. Furthermore, we show that this process can be understood as a competition between the v-nodes' *sensitivity* and their *entropy*, resulting in a functional specialization of the v-nodes. This leads to a high increase in the DCR's memory capacity, and to a significant improvement in its ability to carry out nonlinear spatiotemporal computations. We discuss the implications of the plasticity mechanism with respect to the DCR's entropy, as well as the virtual network topology, and the resulting regression coefficients.

## 3.2 MODEL

In this section, we describe the RC architecture that is based on a single nonlinear node with delayed feedback. We then formulate this architecture using concepts from neural networks.

### 3.2.1 *Single Node Delay-Coupled Reservoir*

Generally speaking, RC comprises a set of models where a large dynamical system called a reservoir, a recurrent neural network for example, nonlinearly maps a set of varying stimuli to a high-dimensional space [57, 70]. The recurrency allows a damped trace of the stimuli to travel within the reservoir for a certain period of time. This phenomenon is termed *fading memory* [13]. Then, random nonlinear motifs within the reservoir nonlinearly mix past and present inputs, allowing a desired output to be *linearly* combined from the activity of the reservoir using a linear regression operation. As the desired output is usually a particular transformation of the temporal and spatial aspects of the stimuli, the operations that a RC architecture are trained to carry out are termed *spatiotemporal computations*.

In a classical RC architecture, past and present inputs $\delta \in \mathbb{R}^m$ undergo nonlinear mixing via injection into a ***r**ecurrent **n**eural **n**etwork* (RNN) of $n$ nonlinear units. This spatial multiplexing of the input is a mapping $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$. Formally, the dynamics are modeled by a difference equation for discrete time

$$x(t+1) = f\big(x(t), \delta(t)\big), \tag{21}$$

or an ***o**rdenary **d**ifferential **e**quation* (ODE) for continuous time

$$\dot{x}(t) = f\big(x(t), \delta(t)\big), \tag{22}$$

where $x(t) \in \mathbb{R}^n$ is the network activity, and $\dot{x}(t)$ the activity's time derivative.

In a *single node **d**elay-**c**oupled **r**eservoir* (DCR), the recurrent neural network is replaced by a single nonlinear node with delayed feedback. Formally, the dynamics can be modeled by a forced (or driven) ***d**elay **d**ifferential **e**quation* (DDE) of the form

$$\dot{x}(t) = -x(t) + f\big(x(t-\tau), \delta(t)\big) \tag{23}$$

where $\tau$ is the delay time, and $x(t), x(t-\tau) \in \mathbb{R}$ are the current and delayed DCR activities. Figure 14 illustrates the DCR architecture and compares it to the standard RNN approach to reservoir computing.

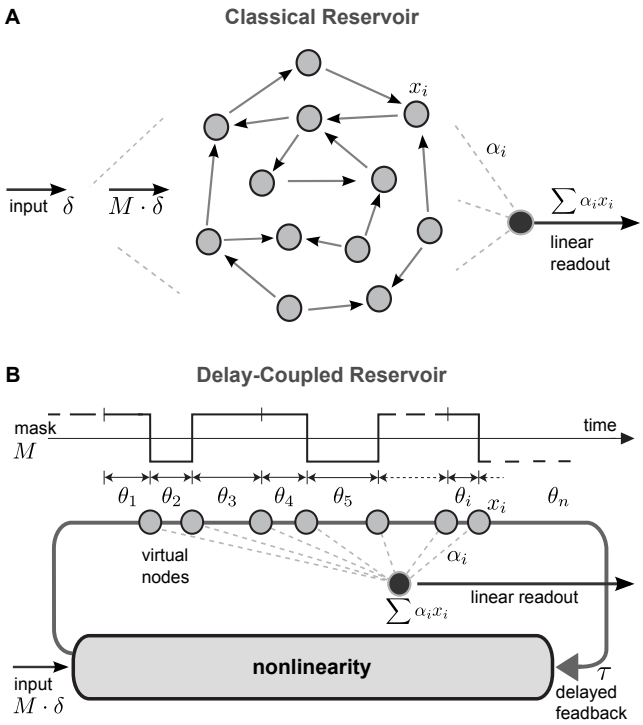


Figure 14: **Comparing classical and single node delay-coupled RC architectures.** (A) A classical RC architecture. The input $\delta$ is *spatially multiplexed* by input weights $M$ to a RNN of size $n$. The activity of the RNN is then linearly readout. (B) A single node delay-coupled reservoir. The input $\delta$ is *temporally multiplexed* across a delay line of length $\tau$ by using a random binary mask $M$ of $n$ bits. Each mask bit $M_i$ is held constant for a short v-delay $\theta_i$ such that the sum of these delays is the length of the delay line $\tau$. The masked input is then nonlinearly transformed and mixed with past input by a nonlinear node with delayed feedback. At the end of each v-delay $\theta_i$ resides a v-node from which linear readouts learn to extract information and perform spatiotemporal computations through linear regression.

Solving the system 23 for $t \geqslant 0$ requires specifying an appropriate initial value function $\phi_0 : [-\tau, 0] \to \mathbb{R}$. As is already suggested by the initial conditions, the phase space of system 23 is a *Banach space* $C_{1,\tau} = C([-\tau, 0], \mathbb{R}) \ni \phi_0$ which is *infinite dimensional* [43]. Using a DDE as a reservoir, this phase space thus provides a high-dimensional feature expansion for the input signal, which is usually achieved by using a RNN with more neurons than input channels.

To inject a signal into the reservoir, it is multiplexed in time: The DCR receives a single constant input $u(\bar{t}) \in \mathbb{R}^m$ in each reservoir time step $\bar{t} = \lceil \frac{t}{\tau} \rceil$, corresponding to one $\tau$-cycle of the system. During each $\tau$-cycle, the input is again linearly transformed by a mask $M \in [0, \tau]^m$ that is piecewise constant for short periods $\theta_i$, representing the temporal spacing, or *v-delays*, between sampling points of $i = 1, \ldots, n$ virtual nodes, or *v-nodes*, along the delay line. Accordingly, the v-delays satisfy $\sum_{i=1}^{n} \theta_i = \tau$, where $n$ is the effective dimensionality of the DCR. Here, the mask $M$ is chosen to be binary with random mask bits $M_i \in \{-\mu, +\mu\}^m$, so that the v-node $i$ receives a weighted input $M_i u(\bar{t})$. In order to assure that the DCR posses fading memory of the input, the system 23 is set to operate in a regime governed by a single fixed point in case the input is constant. Thus, the masking procedure effectively prevents the driven dynamics of the underlying system from saturating to the fixed point.

A sample is read out at the end of each $\theta_i$, yielding $n$ predictor variables (v-nodes) $x_i(\bar{t})$ per time step $\bar{t}$. Computations are performed on the predictors using a linear regression model for some scalar target time series $y$, given by $\hat{y}(\bar{t}) = \sum_{i=1}^{n} \alpha_i x_i(\bar{t})$ where $x_i$ with $i = 1, \ldots, n$ denote the DCR's v-nodes (see equation 26, and $\alpha_i$ are the coefficients determined by regression, e.g. using the *least squares solution* minimizing the sum of squared errors $\sum_{\bar{t}} \big(y(\bar{t}) - \hat{y}(\bar{t})\big)^2$.

In what follows, our model of choice for the DCR nonlinearity is an input-driven *Mackey-Glass* system [38] that is operating, when not driven by input, at a fixed point regime:

$$\dot{x}(t) = -x(t) + \frac{\eta\big(x(t-\tau) + \gamma M \delta(t)\big)}{1 + \big(x(t-\tau) + \gamma M \delta(t)\big)} \tag{24}$$

where $\gamma$ and $\eta$ are model parameters. Figure 15A shows the response of the DCR governed by equation 24 to a single-channel input.

### 3.2.2 *The DCR as a Virtual Network*

The goal is to optimize the computational properties of the DCR as a network, given a vector of v-delays $\Theta = (\theta_1, \ldots, \theta_i, \ldots, \theta_n)$ of its $n$ v-nodes. Approximate v-node equations were already derived by [5] for the case of equidistant v-delays. We extend this result to account for arbitrary v-node spacings on which our plasticity rule can operate.

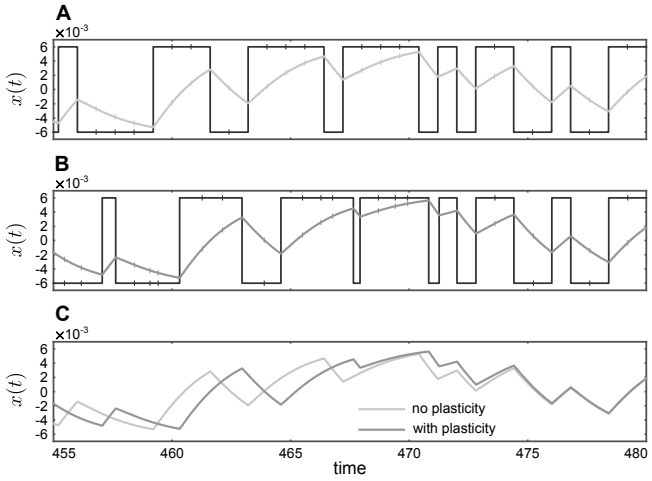


Figure 15: **DCR activity superimposed on the corresponding mask** (A) before and (B) after plasticity. (C) Comparison between the DCR's activity before and after plasticity.

To that end, we first need to define the activity $x(t)$ of the DCR given $\theta_i$ for $i = 1, \ldots, n$.

First, we solve the DDE 23 by applying the *method of steps* (see Section 3.7 for details on solving and simulating the DCR). If the system 23 is evaluated at $(\nu-1)\tau \leqslant t \leqslant \nu\tau$, where a continuous function $\phi_\nu \in C_{[(\nu-2)\tau,(\nu-1)\tau]}$ is the solution for $x(t)$ on the previous $\tau$-interval, we can replace $x(t-\tau)$ by $\phi_\nu(t-\tau)$. Consequently, the solution to 23 subject to $x\big((\nu-1)\tau\big) = \phi_\nu\big((\nu-1)\tau\big)$ is given by

$$\begin{aligned} x(t) =& e^{(\nu-1)\tau-t}\phi_\nu\big((\nu-1)\tau\big)+ \\ & e^{(\nu-1)\tau-t}\int_{(\nu-2)\tau}^{t-\tau} f\big(\phi_\nu(s),\delta(s)\big)e^{s-(\nu-2)\tau}ds. \end{aligned} \tag{25}$$

Let the DCR activity at a particular v-node $x_i(\bar{t}) = x\big((\nu-1)\tau + \sum_{j=1}^{i}\theta_j\big)$, its nonlinearity $f_i(\bar{t}) = f\big(x_i(\bar{t}-1), M_i \cdot u(\bar{t})\big)$, and the DCR time step $\bar{t} = \lceil \frac{t}{\tau} \rceil = \nu$. As shown in Section 3.7, the solution mapping 25 to the DCR can be approximated by assuming $f(\cdot)$ to be piecewise constant at each $\theta_i$. This is a valid approximation since $\theta_i \ll \tau$, and it yields the following expression of the DCR activity at a v-node $i$ as a function of $\{\theta_1, \ldots, \theta_i\}$:

$$x_i(\bar{t}) = e^{-\sum_{j=1}^{i}\theta_j}x_n(\bar{t}-1) + \sum_{j=1}^{i}(1-e^{-\theta_j})e^{-\sum_{k=j+1}^{i}\theta_k}\cdot f_j(\bar{t}) \tag{26}$$

Equation 26 suggests that the activity of v-node $i$ is a weighted sum of the nonlinear component of the preceding v-nodes' activity, down to the last v-node $n$ in the cyclic network, the activity of which is carried over from the previous reservoir time step. The resulting

directed network topology is shown as a virtual weight matrix for equidistant v-nodes ($\theta_i = \tau/n$) in Figure 16A.

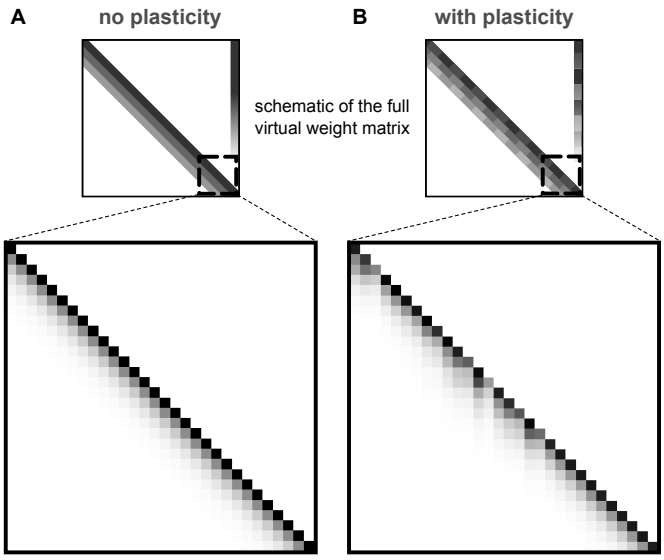


Figure 16: **Virtual weight matrix of a DCR** (A) before and (B) after plasticity. The magnified section corresponds roughly to connectivity within part of the delay span $\tau$ shown in Figure 15.

## 3.3 PLASTICITY

An important role of the randomly alternating mask $M$ is to prevent the DCR dynamics from saturating and thus losing history dependence and sensitivity to new input. However, the random choice of the mask values and the equal v-delays do not guarantee an optimal choice of masking. A simple example which already illustrates this point is given by the occurrence of sequences of equal valued mask bits, as shown in Figure 15A, which leads to unwanted saturation. In general, many more factors exist that determine optimal computation in the reservoir and that need balancing.

Our goal in this section therefore is to develop a plasticity mechanism that optimizes the resulting v-delays with respect to *sensitivity*, while retaining a suitable nonlinear feature expansion into the DCR's phase space. As will be shown in section 3.5.1, this results in a trade-off between sensitivity and *entropy* of the v-nodes. Entropy and sensitivity counteract each other, thus forcing v-nodes to specialize. In a first step (section 3.3.1), we develop a partial plasticity mechanism that maximizes solely the *sensitivity* of individual v-nodes. In a second step (section 3.3.2), the mechanism will be augmented by a counteracting regulatory term which tries to retain diverse feature expansion of the input. The delay $\tau$ together with the number $n$ of v-nodes are treated as fixed and given hyperparameters which de-

termine the particular DCR that is the subject of the optimization process.

### 3.3.1 *Sensitivity Maximization*

We measure a v-node's sensitivity by the slope of its activity at the readout point, i. e., the end point of the $\theta_i$ interval, where bigger slope corresponds to less saturation. The objective is to maximize the overall sensitivity of the DCR for all v-nodes simultaneously. First, we use the approximate solution mapping of a v-node's dynamics from equation 26 to derive a formula of a v-node's activity as a function of the v-delay $\theta_i$ from the previous v-node alone:

$$x_i(\bar{t}) = e^{-\theta_i} x_{i-1}(\bar{t}) + (1 - e^{-\theta_i}) f_i(\bar{t}), \qquad i = 2, \ldots, n \tag{27}$$

$$x_1(\bar{t}) = e^{-\theta_1} x_n(\bar{t} - 1) + (1 - e^{-\theta_i}) f_n(\bar{t} - 1). \tag{28}$$

In addition, the dynamics of the DCR at a particular v-node $i$ in units of reservoir time steps $\bar{t}$ is given by

$$\dot{x}_i(\bar{t}) = -x_i(\bar{t}) + f_i(\bar{t}). \tag{29}$$

Substituting equation 27 into 29 yields the following expression for the sensitivity of a v-node $i$ as a function of $\theta_i$:

$$S_i(\bar{t}) = \dot{x}_i(\bar{t}) = \big(-x_{i-1}(\bar{t}) + f_i(\bar{t})\big) e^{-\theta_i} \tag{30}$$

From equation 30, we define a sensitivity vector $\mathbf{S} \in \mathbb{R}^n$. To optimize the overall sensitivity of the DCR, we maximize an objective function under the constraint that the sum of the v-delays stays equal to the overall delay $\tau$:

$$\arg\max_{\Theta \geqslant 0} \{\|\mathbf{S}\|_2^2\} \quad \text{subject to} \quad \sum_{i=1}^{n} \theta_i = \tau, \tag{31}$$

where $\|\cdot\|_2$ is the Euclidean norm.

To find the vector $\Theta$ that solves the constrained optimization problem 31, we follow the direction of the steepest ascent which is the gradient of the objective function, and we project the outcome to the simplex $\sum_{i=1}^{n} \theta_i = \tau$. The element-wise gradient is given by:

$$\nabla_i \|\mathbf{S}\|_2^2 = \frac{\partial \|\mathbf{S}\|_2^2}{\partial \theta_i} \tag{32}$$

By iteratively inserting expression 27 into the sensitivity formula 30, and eliminating the iteration with 28, we can show that the sensitivity

of a v-node $i$ depends on the v-delays $\theta_j$ of all the preceding v-nodes $j \leqslant i$:

$$S_i(\bar{t}) = e^{-\sum_{k=j+1}^{i} \theta_k} \cdot S_j(\bar{t}) + \Gamma(\theta_{j+1}, \cdots, \theta_i), \tag{33}$$

where $\Gamma(\cdot)$ is a term independent of $\theta_j$. However, since the term $e^{-\sum_{k=j+1}^{i} \theta_k}$ decays exponentially the further the v-node $i$ is from the v-node $j$, one can ignore the contribution of $\theta_j$ to the sensitivity of the v-node $i$ for $i > j$. This simplifies the element-wise gradient to

$$\begin{aligned} \nabla_i \|\mathbf{S}\|_2^2 &= \frac{\partial S_i^2}{\partial \theta_i} \\ &= -2\big(-x_{i-1}(\bar{t}) + f_i(\bar{t})\big)^2 e^{-2\theta_i}. \end{aligned} \tag{34}$$

### 3.3.2 *Homeostatic Plasticity*

The optimization problem 31 maximizes the sensitivity of a v-node $i$ by decreasing $\theta_i$, its temporal distance from the previous v-node, as is suggested by the element-wise gradient 34. As a result, the objective prefers small $\theta_i$, many of which may even go to $0$ despite the constraint $\sum_{i=1}^{n} \theta_i = \tau$. This leads to a reduction of the reservoir's effective dimensionality and, accordingly, to a loss of diversity in expanding informational features of the input. To measure the latter, we consider entropy as an indicator of the v-nodes' informational diversity and decorrelation. As we will show in section 3.5.1, a plasticity mechanism that solely maximizes entropy of the v-nodes would lead to an unbounded increase of $\theta_i$ and therefore presents a natural counteract to sensitivity. We hypothesize that good computational performance is achieved when these two forces are balanced. To this end, we introduce a regulatory term into the sensitivity measure that punishes small v-delays. This leads to a homeostatic plasticity mechanism for v-delays that regulates between sensitivity and entropy and, in combination with the constraint $\sum_{i=1}^{n} \theta_i = \tau$ given by the DCR state space, enforces competition and specialization of v-nodes.

The resulting homeostatic sensitivity measure has the form

$$S_{i\rho}(\bar{t}) = \theta_i^{\rho} \cdot \dot{x}_i(\bar{t}) = \theta_i^{\rho}\big(-x_{i-1}(\bar{t}) + f_i(\bar{t})\big)e^{-\theta_i} \tag{35}$$

where $\rho > 0$ is a *regulating parameter* that modulates the penalty afflicted on the decrease in $\theta_i$. Lower $\rho$ leads the objective to favor smaller v-delays and vice versa.

From equation 35, we define a homeostatic sensitivity vector $\mathbf{S}_\rho \in \mathbb{R}^n$, and an optimization problem

$$\arg\max_{\Theta \geqslant 0} \big\{ \mathcal{O}(\Theta) = \|\mathbf{S}_j\|_2^2 \big\} \quad \text{subject to} \quad \sum_{i=1}^{n} \theta_i = \tau, \tag{36}$$

and we maximize $\mathcal{O}$ by following the direction of the steepest ascent. Since the contribution of $\theta_j$ to the sensitivity of a v-node $i$ for $i > j$ is ignorable, following the argumentation from equation 33, the element-wise gradient is simplified to

$$\begin{aligned}\nabla_i \mathcal{O}(\Theta) &= \frac{\partial S_{i\rho}^2}{\partial \theta_i} \\ &= -2\theta_i^{2\rho-1}(\theta_i - \rho)\big(-x_{i-1}(\bar{t}) + f_i(\bar{t})\big)^2 e^{-2\theta_i}. \end{aligned} \tag{37}$$

Given the above, the update rule of the vector $\Theta = (\theta_1, \cdots, \theta_n)$ reads

$$\Theta \leftarrow \pi_V\big(\Theta + \alpha \cdot \mathcal{J}_{\mathcal{O}}(\Theta)\big), \tag{38}$$

where $\alpha$ is a scalar learning rate, $\mathcal{J}_{\mathcal{O}} = \nabla \mathcal{O}(\Theta)$ is the Jacobian matrix of $\mathcal{O}$ with respect to $\Theta$, and $\pi_V$ is an orthogonal projection which assures that $\Theta$ remains on the constraint simplex $V$ defined by $\sum_1^n \theta_i = \tau$ (see Section 3.8 for details of the constraint satisfaction).

Given a v-node's positive scaling factor $\sigma_i(\bar{t}) = \big(-x_{i-1}(\bar{t}) + f_i(\bar{t})\big)^2$, the homeostatic plasticity learning rule for a single v-node $i$ then reads

$$\theta_i(\bar{t}+1) = \theta_i(\bar{t}) - 2\alpha\sigma_i(\bar{t})\big(\theta_i(\bar{t}) - \rho\big)\theta_i^{2\rho-1}(\bar{t})e^{-2\theta_i(\bar{t})}. \tag{39}$$

with the term $\big(\theta_i(\bar{t}) - \rho\big)$ homeostatically balancing between the v-node's increase and decrease in the v-delay to its predecessor, depending on the choice of the regulating term $\rho$.

## 3.4 COMPUTATIONAL PERFORMANCE

In the following, we test the effect of the homeostatic plasticity mechanism 39 on the performance of the DCR. Simulations are carried on 100 DCRs, the activity of each is sampled at 600 v-nodes that are initially equidistant with $\theta = 0.8$. Each DCR is completely distinguishable from the other by its binary mask $M$, and the 600 mask values are randomly chosen from the set $\{-0.1, +0.1\}$. Simulation starts with a short initial period for stabilizing the dynamics, followed by a plasticity phase of $n_p = 500$ time steps, each corresponding to one $\tau$. The learning rate $\alpha$ is set to 0.01 and the regulating parameter $\rho$ to 1.0. Afterwards, readouts are trained on $n_t = 5000$ samples for both the original and modified v-delays $\theta_i$, and validated on another $n_v = 1000$ samples. The model parameters of the Mackey-Glass nonlinearity (see equation 24) are set to $\gamma = 0.05$ and $\eta = 0.4$.

Given a task-dependent target time series $y$ and a linear regression estimate $\hat{y}(\bar{t}) = \sum_{i=1}^n \alpha_i x_i(\bar{t})$ ($x_i$ being the DCR's v-nodes response

to the input $u$), we measure the performance using the *normalized root-mean-square error*

$$\text{nrmse}(y, \hat{y}) = \sqrt{\frac{\sum_{n_v}(y - \hat{y})^2}{n_v \text{var}(y)}}. \tag{40}$$

### 3.4.1 *Memory Capacity*

The memory capacity of a reservoir is a measure of its ability to retain in its activity a trace of its input history. Optimal linear classifiers are trained for reconstructing a uniformly distributed scalar input $u(\bar{t}) \sim \mathcal{U}_{[0,0.5]}$ at different time lags $\ell$. Figure 17 compares the memory capacity of DCRs before and after plasticity. For time lags $|\ell| > 5$, where the ability to reconstruct the input history starts to diverge from optimal (see Figure 17A), the increase of the DCR's memory capacity can reach up to 70%. The improvement is measured as the relative change in nrmse at each time lag, due to plasticity. Only one of the 100 DCRs showed $\sim$ 20% deterioration in memory capacity after plasticity for the largest time lag (see inset in Figure 17B).

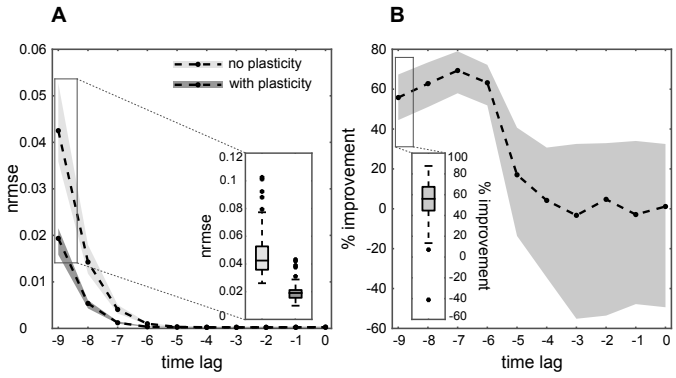


Figure 17: **Memory capacity before and after plasticity.** (A) Performance on memory construction before and after plasticity for different time lags. The inset shows performance on memory construction ten time steps in the past ($\ell = -9$) before and after plasticity. (B) Relative improvement, measured by the decrease in nrmse, after applying homeostatic plasticity. The inset shows the improvement on memory construction ten time steps in the past ($\ell = -9$). (A,B) The dotted lines are the medians of the corresponding plots, while the shaded areas mark the first and third quartiles. In addition to marking the quartiles, the insets show whiskers that extend to 1.5 times the midspread. The crosses specify outliers.

### 3.4.2 *Nonlinear Spatiotemporal Computations*

A widely used benchmark in reservoir computing is the capacity to model a tenth order ***n**onlinear **a**utoregressive **m**oving **a**verage* system $y$

in response to a uniformly distributed scalar input $u(\bar{t}) \sim \mathcal{U}_{[0,0.5]}$. The NARMA-10 task requires the DCR to compute at each time step $\bar{t}$ a response

$$y(\bar{t}) = 0.3y(\bar{t}-1) + 0.05y(\bar{t}-1)\sum_{\bar{s}=1}^{10} y(\bar{t}-\bar{s}) + 1.5u(\bar{t}-1)u(\bar{t}-10) + 0.1. \tag{41}$$

Thus, NARMA-10 requires modeling quadratic nonlinearities and shows a strong history dependence that challenges the DCR's memory capacity. Figure 18 compares the performance in nrmse of DCRs before and after plasticity for different time lags. Even with no time lag $|\ell| = 0$, the task still requires the DCR to retain fading memory. This is in order to account for the dependence on inputs and outputs 10 time steps in the past. The plasticity mechanism achieves $\sim$ 22.8% improvement in performance on average, surpassing state-of-the-art values in both classical [135] and delay-coupled reservoirs [5] with an average nrmse of $0.138 \pm 0.02$std. Only in five trials did the performance deteriorate (see inset in Figure 18B). The improvement decreases for larger time lags due to the deterioration in the DCR's memory capacity observed in Figure 17, but remains significant for $|\ell| < 5$.

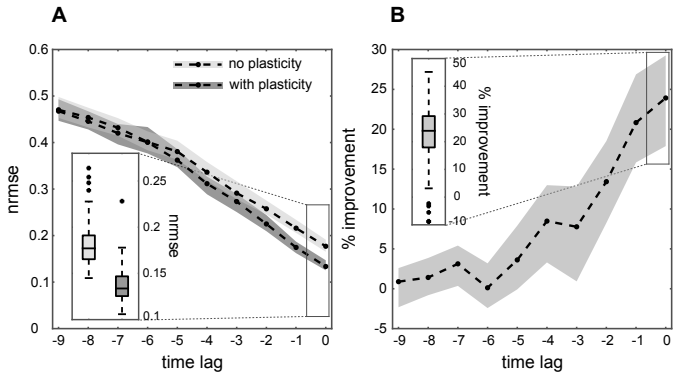


Figure 18: **Spatiotemporal computational power before and after plasticity.** (A) Performance on the NARMA-10 task before and after plasticity for different time lags. The inset shows performance at zero time lag before and after plasticity. (B) Relative improvement, measured by the decrease in nrmse, after applying homeostatic plasticity. The inset shows the improvement at zero time lag. (A,B) The dotted lines are the medians of the corresponding plots, while the shaded areas mark the first and third quartiles. In addition to marking the quartiles, the insets show whiskers that extend to 1.5 times the midspread. The crosses specify outliers.

## 3.5 DISCUSSION: EFFECTS OF PLASTICITY

In order to explain the observed results, we analyze and discuss the effects of the homeostatic plasticity mechanism 39 on the system's entropy $\mathcal{H}(x)$, virtual network topology, and the readout weights distribution $p(\alpha)$. We also discuss the role of the regulating parameter $\rho$.

### 3.5.1 *Entropy*

In section 3.3.2, we stated that expanding the informational features of the input requires a mechanism that counteracts the reduction of a v-delay due to the maximization of the v-node's sensitivity. To prove this hypothesis, we derive a learning mechanism that explicitly maximizes the *mutual information* between the DCR's response and its input. Again, we assume the v-nodes are independent, and for a particular v-node $i$, we maximize the quantity

$$\mathcal{I}(x_i; u) = \mathcal{H}(x_i) - \mathcal{H}(x_i|u), \tag{42}$$

where $\mathcal{H}(x_i)$ is the entropy of the v-node's response, while $\mathcal{H}(x_i|u)$ is the entropy of the v-node's response *conditioned* on the input. In other words, $\mathcal{H}(x_i|u)$ is the entropy of the response that does not result from the input. Bell and Sejnowski argued that maximizing 42 with respect to some parameter $\theta$ is equivalent to maximizing $\mathcal{H}(x_i)$, since the conditional entropy $\mathcal{H}(x_i|u)$ does not depend on $\theta$, i. e., maximizing a v-node's input-information capacity is equivalent to maximizing its self-information capacity or entropy [10].

The entropy of $x_i$ is given by $\mathcal{H}(x_i) = -E\left[\ln p_x(x_i)\right]$, where $p_x(x_i)$ is the probability density function (pdf) of the v-node's response. Since $x_i$ is an invertible function of the nonlinearity $f_i$ (see equation 27), that is itself an invertible function of the input $u$ (if the nonlinearity is chosen appropriately such as in equation 24, the pdf of $x_i$ can be written as a function of the pdf of $f_i$:

$$p_x(x_i) = \frac{p_f(f_i)}{\left|\frac{\partial x_i}{\partial f_i}\right|} \tag{43}$$

The entropy of the v-node's response is then given by

$$\mathcal{H}(x_i) = E\left[\ln\left|\frac{\partial x_i}{\partial f_i}\right|\right] - E\left[\ln p_f(f_i)\right] \tag{44}$$

The term $-E\left[\ln p_f(f_i)\right]$ measures the entropy of the nonlinearity $f_i$ and is independent of $\theta_i$. From equation 44, and taking into account

equation 27, we can derive a learning rule that maximizes the entropy of the response by applying stochastic gradient ascent:

$$\begin{aligned}\Delta\theta_i \propto \frac{\partial \mathcal{H}(x_i)}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i}\left(\ln\left|\frac{\partial x_i}{\partial f_i}\right|\right) \\ &= \left(\frac{\partial x_i}{\partial f_i}\right)^{-1}\frac{\partial}{\partial \theta_i}\left(\frac{\partial x_i}{\partial f_i}\right) \\ &= (1-e^{-\theta_i})^{-1}\frac{\partial}{\partial \theta_i}(1-e^{-\theta_i}). \end{aligned} \tag{45}$$

This leads to the following learning rule:

$$\Delta\theta_i = \alpha\frac{e^{-\theta_i}}{1-e^{-\theta_i}}, \tag{46}$$

where $0 < \alpha \ll 1$ is a learning rate.

The update term 46 is a strictly positive monotonic function of the v-delay $\theta_i$. This entails that, when unconstrained, maximizing a v-node's informational feature expansion results in an unbounded increase in its v-delay, i. e., $\theta_i \to +\infty$. On the other hand, the plasticity rule 39 can be rewritten as

$$\Delta\theta_i = \alpha\sigma_i\rho - \alpha\sigma_i\theta_i, \tag{47}$$

where $\sigma_i = 2\sigma_i\theta_i^{2\rho-1}e^{-2\theta_i} > 0$. The term $\alpha\sigma_i\rho$ in the plasticity mechanism 47 is also positive. This entails that it results, similar to 46, in an unbounded increase in the v-delay, and as a corollary, in an increase in the v-node's informational feature expansion.

Given the above, the homeostatic plasticity mechanism 39, for a particular DCR with delay $\tau$, improves spatiotemporal computations by leading v-nodes to *specialize* in function. This is mediated by a competition between the v-nodes' sensitivity and their entropy. Some v-nodes become more sensitive to small fluctuations in input history, while others are brought closer to saturation where their entropy is higher, and as such, their ability for expanding informational features.

### 3.5.2 *Virtual Network Topology*

The effects of the homeostatic plasticity mechanism 39 on the DCR's network topology can be deduced from equation 26, according to which self-wights are given by $w_{ii} = (1-e^{-\theta_i})$, and the weights the v-node $i$ receives from the preceding v-node $j = i-1$ is $w_{ij} = (1-e^{-\theta_j})e^{-\theta_i}$.

When $\theta_i$ decreases, so does the self-amplification $w_{ii}$, which is consistent with less saturation of the v-node's activity. On the other hand,

the v-node's afferent $w_{ij}$ increases, which in turn increases the v-node j's influence on the activity of the v-node i, resulting in higher correlation between the two (or anti-correlation, depending on the signs of the corresponding mask values $M_j$ and $M_i$). The increase of correlation is in agreement with simulation results and in accord with the decrease of the v-node's entropy as its v-delay decreases. This is the case since the influence of the current input is overshadowed by information from the input history that is delivered from the preceding v-node j, which now drives the v-node i. Figure 16B shows an exemplary virtual weight matrix following plasticity, which illustrates these changes in network topology due to the repositioning of v-nodes on the delay line.

### 3.5.3 *Homeostatic Regulation Level*

Introducing the parameter $\rho$ is necessary to regulate the trade-off between sensitivity and entropy, i. e., increasing and decreasing $\theta_i$, as discussed analytically in section 3.5.1. It is also the defining factor in the v-node's tendency to collapse, as is evident from the form of the plasticity function 47. The collapse of v-nodes is tantamount to a reduction in the DCR's dimensionality, which may be unfavorable with regard to the DCR's computational performance.

We test the latter hypothesis, and the choice of the regulating parameter, by running 1000 NARMA-10 trials for different $\rho$ values that ranged between 0 and 2. Each trial shares the same mask M and the same NARMA-10 time series. As shown in Table 1, the average improvement in performance in comparison to the reference equidistant case $\rho = 0.8$ increases for smaller $\rho$ values, but drops again for $\rho = 0$. An increase in $\rho > 0.8$ also increases the improvement of performance but this increase saturates at $\rho = 1.6$. This is the case since the increase in v-delays favored by high $\rho$ values, makes the collapse of other v-delays inevitable, in order to preserve the DCR's constant delay $\tau$.

In a more detailed analysis, for each of the 1000 trials, we ranked different $\rho$ values according to the resulting improvement of performance in reference to the equidistant case $\rho = 0.8$. We then calculated the percentage of trials that achieved the highest improvement in performance ($1^{st}$ rank) for some $\rho$ value, compared to all other $\rho$ values. We carried the same procedure for the $2^{nd}$ and $3^{rd}$ ranks as well. Figure 19 confirms the previous results as it shows that for $\rho = 0$, it is still possible to achieve the best improvement in performance, but it is less likely than other values. Figure 19 also illustrates a striking result. For none of the trials was the equidistant case, where no plasticity took place, the best choice regarding the computational power of the DCR. Only in 0.3% of the trials, did the nonplastic case rank $3^{rd}$. As a result, for a given DCR setup there always exists a choice of $\rho$ that results

| regulating parameter ρ | 0.0 | 0.2 | 0.4 | 0.6 | 1.0 |
|---|---|---|---|---|---|
| % improvement | 17.76 | 20.31 | 19.67 | 16.56 | 21.0 |
| % reduction $\|\bar{\alpha}\|$ | 17.27 | 19.59 | 18.32 | 17.04 | 22.98 |
| regulating parameter ρ | 1.2 | 1.4 | 1.6 | 1.8 | 2.0 |
| % improvement | 22.58 | 23.07 | 23.76 | 23.67 | 23.76 |
| % reduction $\|\bar{\alpha}\|$ | 24.22 | 24.74 | 25.06 | 25.61 | 26.02 |

Table 1: **Effects of the homeostatic regulation level on performance and readout coefficients.** Average improvement in performance and reduction in average absolute values of the readout coefficients, in comparison to the equidistant v-nodes case $\rho = 0.8$.

in nonequidistant v-nodes where spatiotemporal computations are enhanced. This is also summarized in the Figure 19B, which shows the improvement in performance given the best choice in the regulating parameter ρ for each trial. The nrmse is reduced by $\sim 33.7\%$, with an average performance that reaches an unprecedented value of nrmse $\sim 0.117 \pm 0.01$std. We point out that the homeostatic plasticity mechanism 39 also reduces the average absolute values of the readout coefficients $|\bar{\alpha}|$ (see Table 1), which is similar in effect to an $L_2$-regularized model fit. This is not only advantageous with respect to numerical stability but $L_2$-regularization also allows for lower mean-squared errors as compared to an unregularized fit [50].

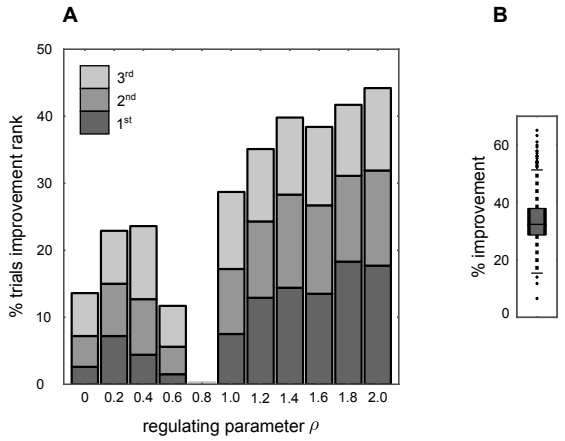


Figure 19: Performance of 1000 NARMA-10 trials for regulating parameter ρ values between 0 and 2. (A) Percentage of trials that achieved the 1st, 2nd, and 3rd highest improvement in performance for each ρ value. (B) Relative improvement, measured by the decrease in nrmse, after applying homeostatic plasticity, given for each trial the best choice in ρ. The box plot marks the median and the first and third quartiles. Whiskers extend to 1.5 times the midspread. The crosses specify outliers.

## 3.6 CONCLUSION

We have introduced a plasticity mechanism for improving the computational capabilities of a DCR, a novel RC architecture where a single nonlinear node is delay-coupled to itself. The homeostatic nature of the plasticity mechanism 39 relates directly to the information processing properties of the DCR in that it balances between sensitivity and informational expansion of input (section 3.5.1). While the role of homeostasis in information processing and computation has only been discussed more recently, its function as a stabilization process of neural dynamics has acquired earlier attention [136, 12]. From the perspective of the nervous system, pure Hebbian potentiation and anti-Hebbian depression would lead to destabilization of synaptic efficacies by generating amplifying feedback loops [78, 120], necessitating a homeostatic mechanism for stabilization [28, 145, 133]. Similarly, as suggested by the effects of the plasticity mechanism 47 on the virtual network topology (section 3.5.2), the facilitating sensitivity term $-\alpha\sigma\theta$ is counteracted by the depressive entropy term $+\alpha\sigma\rho$, which prevents synaptic efficacies from overpotentiating or collapsing.

The optimal choice of the regulating parameter $\rho$ is found at the moment by brute force (section 3.5.3). It is reasonable to assume that heuristics may exist for the optimal choice of $\rho$, given a particular mask structure, since the alterations in the mask values influence a v-node's sensitivity and entropy. A possible heuristic may relate the value of $\rho$ to properties of *maximum length sequences*, by which Appeltant et al. constructed mask sequences with equidistant v-nodes [6]. Establishing this connection is a subject of current research.

Enhancing the temporal multiplexing of input to the nonlinear node was the main goal of this article. We speculate that similar multiplexing may suggest a further important functionality of the extensive dendritic trees in some neuron types. On the one hand, Izhikevich discussed the infinite dimensionality dendritic propagation delays offer to recurrent neural networks [56]. On the other hand, several studies investigated the computational role of the spatial distribution of active dendrites [108, 39, 40]. In this article, we advocate a unified computational account that may integrate both the temporal and spatial aspects of dendritic computations. In particular, the spatial location of dendritic arbors may be optimized to achieve computationally favorable temporal multiplexing of the soma's input, in the fashion suggested by the DCR architecture. Consolidating this speculation would be the subject of future studies.

## 3.7 APPENDIX A: SOLVING AND SIMULATING THE DCR

In this section, we derive equations 25 and 26. We would like to solve system 23 for $x(t)$, with $(\nu-1)\tau \leqslant t \leqslant \nu\tau$. Due to the recurrent depen-

dency $x(t-\tau)$, this is not possible right away. However, if we assume a continuous function $\phi_\nu \in C_{[(\nu-2)\tau,(\nu-1)\tau]}$ is the solution for $x(t)$ on the previous $\tau$-interval, we can replace $x(t-\tau)$ by $\phi_\nu(t-\tau)$. After the substitution, system 23 becomes solvable by the elementary method of *variation of constants* [48]. The latter provides a solution to an equation of type $\dot{x}(t) = a(t)x(t) + b(t)$ with initial condition $x(t_0) = c$. The general solution on the interval I to the inhomogeneous equation is then given by

$$x(t) = x_h(t)\left(c + \int_{t_0}^{t} x_h(t)^{-1} b(t)\,dt\right), \quad t \in I,$$

where

$$x_h(t) = \exp\left(\int_{t_0}^{t} a(t)\,dt\right)$$

denotes a solution to the associated homogeneous differential equation. Consequently, for $a(t) = -1$ and $b(t) = f\big(\phi_\nu(t-\tau), \delta(t)\big)$, the solution to

$$\dot{x}(t) = -x(t) + f\big(\phi_\nu(t-\tau), \delta(t)\big),$$

subject to $x\big((\nu-1)\tau\big) = \phi_\nu\big((\nu-1)\tau\big)$, is given by

$$x(t) = e^{(\nu-1)\tau - t}\left(\phi_\nu\big((\nu-1)\tau\big) + \int_{(\nu-2)\tau}^{t-\tau} f\big(\phi_\nu(s), \delta(s)\big) e^{s-(\nu-2)\tau}\,ds\right). \tag{48}$$

This expression can be used right away in a numerical solution scheme, where the integral is solved using the *cumulative trapezoidal rule*. The resulting simulation of the DCR has been shown to be comparable in its accuracy and computational capabilities to adaptive numerical solutions, while considerably saving computation time [114].

Recall that $t_i = (\nu-1)\tau + \sum_{j=1}^{i} \theta_j$, with $\theta_j$ the temporal distances between consecutive virtual nodes. To arrive at a manageable analytical expression of the above solution for the sampling point $t_i$ of virtual node $i$ during the $\nu^{th}$ $\tau$-cycle, we make the following approximation:

Let the DCR activity at a particular v-node $x_i(\bar{t}) = x(t_i)$, its nonlinearity $f_i(\bar{t}) = f\big(x_i(\bar{t}-1), M_i \cdot u(\bar{t})\big)$, and the DCR time step $\bar{t} = \lceil \frac{t}{\tau} \rceil = \nu$. If we assume that $f(\cdot)$ is piecewise constant at each $\theta_i$, which is a valid approximation since $\theta_i \ll \tau$, expression 48 simplifies further to

$$
\begin{aligned}
x_i(\bar{t}) &= e^{-\sum_{j=1}^{i}\theta_j}\left(x_n(\bar{t}-1)+\sum_{j=1}^{i} f_i(\bar{t})\int_0^{\theta_j} e^s\,ds\right)\\
&= e^{-\sum_{j=1}^{i}\theta_j}\left(x_n(\bar{t}-1)+\sum_{j=1}^{i} f_i(\bar{t})\left(e^{\theta_j}-1\right)\right)\\
&= e^{-\sum_{j=1}^{i}\theta_j}x_n(\bar{t}-1)+\sum_{j=1}^{i}(1-e^{-\theta_j})e^{-\sum_{k=j+1}^{i}\theta_k}\cdot f_j(\bar{t}).
\end{aligned}
$$

## 3.8 APPENDIX B: CONSTRAINT SATISFACTION

The sensitivity update rule of the virtual node distances $\theta_j$ has to satisfy the constraint $\sum_j \theta_j = \tau$. This describes a constraint manifold for valid virtual node distance vectors $\Theta \in \mathbb{R}^n$ during learning. The manifold has the structure of a simplex

$$
V := \{x | x = (x_1, \ldots, x_n)^T, \sum_{i=1}^{n} x_i = \tau\}
$$

with $\dim V = n-1$ and simplex corners given by $\tau e_i$ $(i = 1, \ldots, n)$, where $(e_i)_{i=1}^n$ is the standard orthonormal basis of $\mathbb{R}^n$. We implemented the constraint optimization problem by first computing an unconstrained update for $\Theta$, followed by an orthogonal projection onto $V$. Due to the simple linear structure of $V$, this strategy will converge onto the constrained optimum for $\Theta$.

Denote by $n_V = \frac{\tau}{n}\sum_i e_i$ the central point of the constraint simplex, and let $(\nu_i)_{i=1}^{n-1}$, $\nu_i \in \mathbb{R}^n$, be an orthonormal basis for $V$. The latter is computed from an orthogonal basis, which can be constructed by simple geometrical considerations from the simplex corner point vectors as

$$
\begin{pmatrix} -\tau & \cdots & -\tau \\ & \tau I_{n-1} & \end{pmatrix} \in \mathbb{R}^{n\times n-1}, \tag{49}
$$

where $I_{n-1}$ denotes the $(n-1)$-dimensional unit matrix. It is easily verified that this basis spans $V$ and is indeed orthogonal. In conjunction with the inhomogeneity $n_V$, a normal vector with respect to $V$, any point on $V$ can be expressed via the $\nu_i$. For some $x \in \mathbb{R}^n$ being the result of an unconstrained sensitivity update step, the constraint can be met by projecting $x$ orthogonally onto $V$ via the mapping

$$\begin{aligned}\pi_V(x) &= n_V + \sum_{i=1}^{n-1} \big((x-n_V)^T v_i\big) v_i \\ &= n_V + \sum_{i=1}^{n-1} \big(v_i^T (x-n_V)\big) v_i \\ &= n_V + \underbrace{\left(\sum_{i=1}^{n-1} v_i v_i^T\right)}_{:= M \in \mathbb{R}^{n\times n}} (x-n_V) \\ &= n_V + M(x-n_V). \end{aligned} \tag{50}$$

The addition and subtraction of $n_V$ take care of the fact that $V$ as a hyperplane is translated out of the origin by the inhomogeneity $n_V$. If the $V$-plane was centered in the origin, $\underbrace{(v_i^T x)}_{\in\mathbb{R}} v_i$ would denote the orthogonal projection of $x$ onto the $i^{th}$ orthonormal basis vector. Accordingly, the linear combination of these projections yields the representation of $\pi_V(x)$ with respect to the basis $(v_i)_{i=1}^{n-1}$.

ACKNOWLEDGMENTS

The contributions of Marcel Nonnenmacher and Anna-Birga Ostendorf to early stages of this work are gratefully acknowledged, so are the fruitful discussions with the members of the PHOCUS consortium. The authors acknowledge the financial support of the State of Lower Saxony, Germany via the University of Osnabrück, and the European project PHOCUS in the Framework 'Information and Communication Technologies' (FP7-ICT-2009-C / Proposal Nr. 240763).

# 4

## PAPER III | BEHAVIOR CONTROL IN THE SENSORIMOTOR LOOP WITH SHORT-TERM SYNAPTIC DYNAMICS INDUCED BY SELF-REGULATING NEURONS

ABSTRACT The behavior and skills of living systems depend on the distributed control provided by specialized and highly recurrent neural networks. Learning and memory in these systems is mediated by a set of adaptation mechanisms, known collectively as neuronal plasticity. Translating principles of recurrent neural control and plasticity to artificial agents has seen major strides, but is usually hampered by the complex interactions between the agent's body and its environment. One of the important standing issues is for the agent to support multiple stable states of behavior, so that its behavioral repertoire matches the requirements imposed by these interactions. The agent also must have the capacity to switch between these states in time scales that are comparable to those by which sensory stimulation varies. Achieving this requires a mechanism of short-term memory that allows the neurocontroller to keep track of the recent history of its input, which finds its biological counterpart in short-term synaptic plasticity. This issue is approached here by deriving synaptic dynamics in recurrent neural networks. Neurons are introduced as self-regulating units with a rich repertoire of dynamics. They exhibit homeostatic properties for certain parameter domains, which result in a set of stable states and the required short-term memory. They can also operate as oscillators, which allow them to surpass the level of activity imposed by their homeostatic operation conditions. Neural systems endowed with the derived synaptic dynamics can be utilized for the neural behavior control of autonomous mobile agents. The resulting behavior depends also on the underlying network structure, which is either engineered, or developed by evolutionary techniques. The effectiveness of these self-regulating units is demonstrated by controlling locomotion of a hexapod with eighteen degrees of freedom, and obstacle-avoidance of a wheel-driven robot.

**Keywords:** Sensorimotor loop, autonomous agent, synaptic plasticity, short-term plasticity, homeostasis, self-regulation, hysteresis, oscillation

### 4.1 INTRODUCTION

Living systems, which have to survive in a complex, permanently changing environment have to exhibit a life-sustaining behavior. For

autonomous agents, like *animats*, this is one of the desired capacities. For achieving this objective, autonomous agents are equipped with different types of sensors, with proprioceptors monitoring their internal states, and with motors to articulate their body movements. In addition, since every movement of the body will change the inputs to the sensors and proprioceptors, these agents always operate in a sensorimotor loop.

Even when the overall task is apparently simple, autonomous agents are still expected to express diverse behavior in order to accomplish the task, and the rich dynamics provided by artificial recurrent neural networks is usually invoked for the control of this behavior. Examples include tropisms of wheel-driven robots [54, 118], biped walking [72, 65], active tracking [83], quadruped locomotion, [73, 55, 116], hexapod locomotion [9], and swimming robots [55, 116].

The ability of recurrent neurocontrollers to generate successful behavior depends highly on its connectivity structure as well as on the synaptic efficacies of its connections. Suitable neurocontrollers are usually found by evolutionary techniques [86]. However, synaptic plasticity and regulatory mechanisms of neural activity constitute the biological basis for learning and memory [25], and were taken up by (evolutionary) robotics as a tool for adding learning abilities to autonomous agents [85, 31, 118, 141, 134, 109, 52]. Incorporating plasticity in the neural control of robots takes the load off evolution for finding right synaptic weights and/or operating range of the neurons within the network, and limits the role of the evolutionary process to the allocation of right connectivity structure, which considerably reduces the search space.

We follow on the lead of these studies, where we assume that the connectivity structure is given as a result of an evolutionary process, and we concentrate on deriving synaptic dynamics for the neural control of artificial agents acting in the sensorimotor loop. Our model is referred to as the *self-regulating neuron*, or the SR-neuron for short. A similar model was first proposed for a slightly different synaptic dynamics and another neuron type [143]. The SR-neuron differs from the previous approaches in that its synaptic dynamics act on a faster time scale. Here, synaptic efficacies do not change due to a slow adaptation process, based on repetition of pre- and postsynaptic activity patterns. Instead, they adapt to sensory stimulations at the rate by which these stimulations change. This feature makes the SR-neurons suitable for the requirement of real-time diverse dynamic behavior and for a quick reaction to varying stimuli. As such, and unlike previous studies, the SR-neuron dynamics does not aim at augmenting the neurocontroller with learning, in the sense of a gradual change of behavior to a better one. In other words, there is no training phase that ends with higher fitness and a steady state of the synaptic weights.

Instead, synaptic weights are constantly adapting in response to the changes of external stimuli.

The synaptic dynamics of the self-regulating neuron does not replicate a particular plasticity mechanism that is empirically observed in biological systems. Nevertheless, it is *biologically-inspired* in three different ways, by which it exploits the functional properties of biological plasticity for the benefit of a stable and successful behavior of an artificial agent.

First, self-regulating neurons act as *homeostatic* elements, which try to maintain one of two desired activity states, one referring to low, and the other to high activity. Homeostatic regulation is only necessary to operate when the system is confronted with some external perturbations. Since recurrent neurocontrollers of artificial agents have to work in the sensorimotor loop, they are permanently driven by continuously changing sensory inputs. A neural mechanism for homeostatic plasticity should therefore lead to a stabilization of behavior, by providing the controller with the means necessary to cope with these fast varying sensory inputs.

Biological findings strongly support the existence of such mechanisms, where the incoming signals to a neuron [28], or the neuron's own excitability [133] is homeostatically adjusted to match a functionally desirable neural activation, such as maximizing the entropy of the neural output [132, 74]. Many models from theoretical neuroscience incorporates homeostatic plasticity mechanisms in recurrent neural networks, either in the form of *synaptic scaling* of afferents [100, 144], *intrinsic plasticity* of neural excitability [67, 74, 82, 130], or both [68, 146]. These mechanisms also found their application in improving time series prediction in echo state networks [113]. Homeostasis has also been discussed in the context of adaptation and learning in cybernetics [8], and there are many examples of its successful contribution to learning in recurrent neural control of robots [31, 51, 141, 134, 109, 52].

Second, the synaptic dynamics of the self-regulating neuron partially adheres to Hebb's postulate [47], where the synapses between mutually active neurons are potentiated. Homeostasis, however, prevents the overgrowth of synaptic weights due to the constant potentiation in a fashion similar to the *BCM theory* [12], and its spiking neurons variants [131, 24]. In robotics, learning with a variant of Hebbian plasticity is demonstrated for example by Harter and Kozma [44], Santos et al. [109], and Hoinville et al. [52].

While these studies favor steady-state synaptic weights, controlled bifurcations of neural dynamics might be very desirable in the context of the sensorimotor loop [8]. During the autonomous agent's lifespan, it is important that changes in its stimulation elicit history-dependent responses, which entails a form of *working memory* for the agent [83]. The importance of this functionality comes from the fact

that environmental cues are themselves temporally extended [19, 130]. As such, an autonomous agent's behavior must come as a response to these temporally extended stimuli, rather than to instantaneous states of its environment. This directly connects to the third point of relatedness to biological plasticity, that is, *short-term plasticity* [147, 1]. Due to short-term plasticity, synaptic efficacy changes on faster time scales in ways that reflect the history of the presynaptic activity. This history-dependence may mediate working memory in recurrent neural networks [79]. The self-regulating neuron exhibits this history-dependence, where changes in temporally extended stimuli are captured by the fast synaptic dynamics. This synaptic dynamics then controls the neuron's bifurcation between the two desired activity states which leads to history-dependent adjustment of behavior.

Here, it is shown that self-regulating neurons are suitable for the control of an autonomous agent's behavior under the sensory perturbations of the sensorimotor loop. The activity of neurons, together with the synaptic efficacies, change over time, but usually fluctuate around some average values, as has been demonstrated for simple examples in [94]. A self-regulating neuron is able to attain and maintain a desirable level of activity even if it is confronted with unpredictable, and more or less severe perturbations, induced by changing sensory inputs. Furthermore, it has different internal states at its disposal, leading to different stable behaviors, which may be appropriate for one or another external situation.

The following section introduces self-regulating neurons, together with the properties of the induced synaptic plasticity rule. Because these self-regulating neurons have to operate as elements of neurocontrollers in the sensorimotor loop, the synaptic weights of these neurocontrollers change dynamically according to sensory stimuli or internal feedback loops. With this in mind, the dynamics of simple neural modules is analyzed next under varying stimulation, so as to reach a basic understanding of the stability properties of these modules. This is followed by discussing examples of successful control of behavior for synchronizing coupled reflex loops, for locomotion of a hexapod walking machine, and for obstacle-avoidance of a wheel-driven robot.

## 4.2 SELF-REGULATING NEURONS

Given a neural network $N$ with $n$ neurons, and a structure matrix $c$ defined by $c_{ij} = +1(-1)$ for an excitatory (inhibitory) connection from neuron $j$ to neuron $i$ and $c_{ij} = 0$ otherwise. A single self-regulating neuron $i$ is described as a parametrized discrete-time 3-dimensional dynamical system with state variables $(a_i, \xi_i, \eta_i) \in \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}^+$ for $i = 1, \ldots, n$, where $a_i$ denotes its activation, and $\xi_i$ and $\eta_i$ its *receptor* and *transmitter* strength respectively. Furthermore, it

may have a bias value $\bar{\theta}_i$ that is the sum of a constant bias $\theta_i$ and an external drive I. The output $o_i = \tau(a_i)$ of a neuron $i$ is given by the sigmoidal hyperbolic tangent transfer function $\tau := \tanh$. The weight $w_{ij}$ of the connection from neuron $j$ to neuron $i$ is then defined by

$$w_{ij} := c_{ij}\, \xi_i\, \eta_j\,. \tag{51}$$

We assume that there exists a desirable state $a_i^*$ for the activation of a neuron, and that the 3-dimensional dynamics is to be defined so as to stabilize this state for a certain range of input signals. Such a state defines a preferred operational range of the neurons' dynamics. There are two canonical choices for such a desirable state. One is for the neuron to operate around the *linear* domain of the transfer function, i. e., $a_i^* = 0$ for the hyperbolic tangent nonlinearity. However, recurrent neural networks are expected to capture and respond to environmental stimuli that are riddled by *nonlinear* dependencies. As such, it is reasonable to enforce the nonlinear properties of recurrent neural networks, in order for them to reflect, in their activity, these nonlinear environmental conditions. Therefore, the desired state in the following corresponds to an activation $a_i^*$ for which the nonlinearity of the transfer function $\tau$ is "maximal", i. e., its third derivative satisfies $\tau'''(a^*) = 0$. Since $\tau$ is an antisymmetric function, its third derivative $\tau'''$ is symmetric, and there are two such *operating points* satisfying this condition and they take values

$$a^* := a_\pm^* \approx \pm 0.658479 \quad \text{and} \quad \tau(a^*) = \pm\sqrt{\frac{1}{3}} \approx \pm 0.5773503\,.$$

This means that a neuron prefers a high or low state of activity, or, in terms of rate models, a high or low firing rate.

The basic equations for the dynamics are then set up as follows. The standard additive discrete-time dynamics for the activation $a_i$ of a neuron is given by

$$a_i(t+1) = \bar{\theta}_i + \xi_i(t) \sum_{j=1}^{n} c_{ij}\, \eta_j(t) \tau\big(a_j(t)\big) \quad \text{where } i = 1, \ldots, n\,. \tag{52}$$

Furthermore, it is assumed that the receptor strength $\xi_i$ and the transmitter strength $\eta_i$ for $i = 1, \ldots, n$ are both *positive* for all times. The dynamics of the receptor strength $\xi_i$ modulates the incoming signals to the neuron such that its response becomes maximally nonlinear. In other words, the receptor strength is responsible for pushing the activation $a_i$ of the neuron towards one of the operating points $a_\pm^*$, and is given by

$$\xi_i(t+1) = \xi_i(t) \left[1 + \beta \cdot (\tau^2(a^*) - \tau^2(a_i))\right] \quad \text{where} \quad 0 < \beta < 1\,. \tag{53}$$

The transmitter strength $\eta_i$ communicates the neuron's activity to its targets, i. e., it increases with the activation $a_i$ of the neuron. It also has a decay rate $(1-\gamma)$, which is necessary for the convergence of the dynamics as we show later. Thus, the transmitter dynamics is defined by

$$\eta_i(t+1) = (1-\gamma)\,\eta_i(t) + \delta\big[1+\tau(a_i)\big] \quad \text{where} \quad 0 < \gamma, \delta < 1\,. \tag{54}$$

The discrete-time dynamics $f : \mathbb{R}\times\mathbb{R}^+\times\mathbb{R}^+ \to \mathbb{R}\times\mathbb{R}^+\times\mathbb{R}^+$ given by equations 52-54 is called the dynamics of self-regulating neurons or *SRN-dynamics* for short.

The weight change per time step is then given by

$$\begin{aligned}\Delta w_{ij}(t) &= w_{ij}(t+1) - w_{ij}(t)\\ &= c_{ij}\big(\xi_i(t+1)\eta_j(t+1) - \xi_i(t)\eta_j(t)\big).\end{aligned} \tag{55}$$

Replacing $\xi_i(t+1)$ and $\eta_j(t+1)$ by their dynamics from equations 53,54 leads to

$$\Delta w_{ij}(t) = c_{ij} w_{ij}(t)\cdot\Big[F\big(a_i(t)\big) + G\big(a_j(t)\big) + H\big(a_i(t), a_j(t)\big)\Big]\,, \tag{56}$$

where

$$\begin{aligned}F(a_i) &= -\gamma + \beta(1-\gamma)\big(\tau^2(a^*) - \tau^2(a_i)\big)\,,\\ G(a_j) &= \frac{\delta}{\eta_j}\big(1+\tau(a_j)\big)\,,\\ H(a_i, a_j) &= \frac{\beta\delta}{\eta_j}\big(1+\tau(a_j)\big)\big(\tau^2(a^*) - \tau^2(a_i)\big)\,.\end{aligned} \tag{57}$$

This demonstrates two of the biologically-inspired features of the synaptic dynamics. The weight change depends on the product of the presynaptic and postsynaptic activations through the (anti-)Hebbian element $H(a_i, a_j)$, which includes the term $\tau(a_j)\tau^2(a_i)$. In addition, the term $H(a_i, a_j)$ is not always positive, since its sign depends on the postsynaptic activity $a_i$. When $|a_i| < |a^*|$, the term is positive which leads to Hebbian-like synaptic potentiation. Otherwise, the term is negative and the synaptic efficacy is depressed in an anti-Hebbian fashion. In other words, the term $\big(\tau^2(a^*) - \tau^2(a_i)\big)$ reflects the postsynaptic-dependent homeostatic nature of the synaptic dynamics, where a regime of potentiation is separated from a regime of depression at the threshold $a^*$.

## 4.3 RESULTS

In what follows, we rigorously analyze the dynamics of simple self-regulating neural modules. Namely, we study the stable dynamics of a SR-neuron without self connection. We then prove that a SR-neuron with an excitatory self-connection is bistable under certain conditions, which confirms observations that were made in [94]. We show in addition that a SR-neuron with inhibitory self-connection oscillates with period-2. We finally demonstrate the operation of networks of these modules for the control of behavior in the sensorimotor loop.

### 4.3.1 *Dynamics of Self-Regulating Neurons*

To get a first impression of the SRN-dynamics we study the dynamics of a single neuron with and without self-connection. Suppressing the neuron's index $i$, the 3-dimensional dynamics reads

$$\begin{aligned} a(t+1) &= \theta + c\,\xi(t)\,\eta(t)\,\tau\big(a(t)\big) + \xi I(t)\,, \\ \xi(t+1) &= \xi(t)\left[1+\beta\cdot(\tau^2(a^*)-\tau^2(a))\right], \\ \eta(t+1) &= (1-\gamma)\,\eta(t) + \delta\big[1+\tau(a)\big]\,, \end{aligned} \tag{58}$$

where $I$ represents the inputs coming from other neurons, i. e.,

$$I(t) := \sum_{j\neq i} c_j \eta_j \tau\big(a_j(t)\big)\,. \tag{59}$$

For the moment, we assume that $I$ is constant over time, and that there exists a stable fixed point $(a^*, \xi^*, \eta^*)$ of the 3-dimensional SRN-dynamics, in order to derive conditions for its existence. Throughout this section, the parameters $\beta, \delta$, and $\gamma$ are set to 0.1. To determine the stability of the dynamical system 58 at a fixed point $(a^*, \xi^*, \eta^*)$, we study its linearization at a state $(a, \xi, \eta) \in \mathbb{R}\times\mathbb{R}^+\times\mathbb{R}^+$, which is given by the Jacobian matrix

$$(Df)(a,\xi,\eta) = \begin{pmatrix} c\xi\eta(1-\tau^2) & c\eta\tau + I & c\xi\tau \\ -2\beta\xi\tau(1-\tau^2) & 1+\beta(\tau^2(a^*)-\tau^2(a)) & 0 \\ \delta(1-\tau^2) & 0 & 1-\gamma \end{pmatrix}. \tag{60}$$

There are three possible fixed points for the dynamical system 58. These are the two *desirable* fixed points $x_\pm = (a^*_\pm, \xi^*_\pm, \eta^*_\pm)$ with transmitter strength $\eta^*_\pm = \frac{\delta}{\gamma}(1+\tau(a^*_\pm))$, and the *trivial* fixed point $x_0 = (\theta, 0, \eta_0)$ with a vanishing receptor strength, and a transmitter strength $\eta_0 = \frac{\delta}{\gamma}\big(1+\tau(\theta)\big)$. We refer to the last situation as a "dead

neuron" because it is not able to process incoming signals. Whether one of these fixed points is asymptotically stable or not depends on the eigenvalues of $(Df)(a^*, \xi^*, \eta^*)$, as we show next.

#### 4.3.1.1 *Dynamics without Self-Connection*

For a first analysis, we study a single neuron without self-connection, i. e.,, $c = 0$, and with a fixed bias value $\theta$. It is driven by the input signal $I$. The linearization of SRN-dynamics then reads

$$(Df)(a, \xi, \eta) = \begin{pmatrix} 0 & I & 0 \\ -2\beta\xi\tau(1-\tau^2) & 1+\beta(\tau^2(a^*)-\tau^2(a)) & 0 \\ \delta(1-\tau^2) & 0 & 1-\gamma \end{pmatrix}. \tag{61}$$

A fixed point $x^*$ is asymptotically stable if all the eigenvalues $\lambda_k$ of $(Df)(x^*)$ satisfy $|\lambda_k| < 1$. The two desirable fixed points $x_\pm = (a^*_\pm, \xi^*_\pm, \eta^*_\pm)$ for this neuron also satisfy the equation

$$a^*_\pm - \theta = \xi^*_\pm I\,. \tag{62}$$

First, one observes from condition 62)that the receptor strength $\xi^*_\pm$ diverges for inputs $I \to 0$, and thus, $x_\pm$ are both unstable when $I = 0$. Otherwise, replacing the input $I > 0$ in the linearization 61 with its value from condition 62, leads to the following eigenvalues around the fixed points $x_+$:

$$\lambda_{1,2}(a^*_+) = \lambda_\pm(a^*_+) = \frac{1 \pm \sqrt{1 - 8\beta\left(1-\tau^2(a^*_+)\right)(a^*_+ - \theta)\tau(a^*_+)}}{2}, \qquad \lambda_3 = 1-\gamma\,, \tag{63}$$

and similarly for $I < 0$ and the fixed point $x_-$, but with $\lambda_\pm$ being a function of $a^*_-$ instead. For both fixed points, the stability condition $|\lambda_k| < 1$ always holds for $\lambda_-$ and $\lambda_3$. This also stresses the necessity of introducing the decay term parametrized by $\gamma$ of the transmitter dynamics $\eta$ for the stability of the SR-neuron, without which $\lambda_3 = 1$. On the other hand, the stability condition only holds for $\lambda_+$ when $(a^*_\pm - \theta)\tau(a^*_\pm) < 0$. It follows that for $\theta \in (a^*_-, a^*_+)$, the SR-neuron is homeostatic, i. e., one of the fixed points $x_\pm$ is stable, for all inputs $I \in \mathbb{R} \setminus \{0\}$. We thus call a bias $\theta$ that is within the range $(a^*_-, a^*_+)$ a *homeostatic bias*. Asymptotically, it acts like a binary neuron switching from low activity $a^*_-$ to high activity $a^*_+$ around $I = 0$. This is also confirmed by Figure 20 showing bifurcation diagrams for the output $\tau(a)$ and the receptor strength $\xi$ under these conditions.

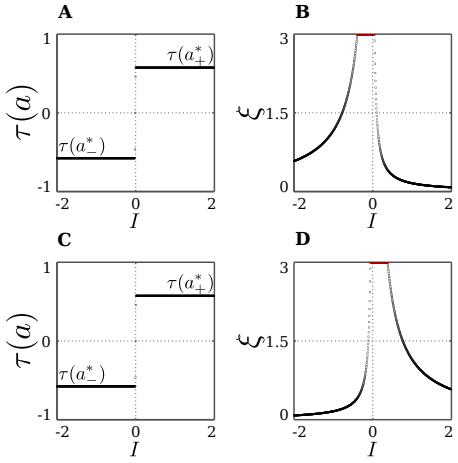


Figure 20: **Stable dynamics of a SR-neuron without self-connection for a homeostatic bias and varying input.** (A,B) Bifurcation diagrams of the output $\tau(a)$ and the receptor strength $\xi$, for varying input $I$ and a *positive* homeostatic bias $\theta = +0.5$. (C,D) Bifurcation diagrams of the output $\tau(a)$ and the receptor strength $\xi$, for varying input $I$ and a *negative* homeostatic bias $\theta = -0.5$.

In addition, keeping in mind that $a^*_- = -a^*_+$, the fixed point $x_-$ satisfies condition 62 when $I < 0$, if $\theta > a^*_+$, which leads to $(a^*_- - \theta)\tau(a^*_-) < 0$, and as such, $|\lambda_+(a^*_-)| < 1$ holds. This entails that $x_-$ is asymptotically stable when $I < 0$ and $\theta > a^*_+$. Correspondingly, $x_+$ is asymptotically stable when $I > 0$ and $\theta < a^*_-$. In other words, the SR-neuron without self-connection is homeostatic only over half of the input domain when $\theta \notin [a^*_-, a^*_+]$. On the other hand, the trivial fixed point $x_0$, corresponding to a dead neuron, becomes stable for all $I$, since the eigenvalues of $(Df)(x_0)$ are

$$\lambda_1 = 0 \quad , \quad \lambda_2 = 1 + \beta(\tau^2(a^*_\pm) - \tau^2(\theta)) \quad , \quad \lambda_3 = 1 - \gamma \,, \tag{64}$$

which satisfy $|\lambda_k| < 1$ when $\theta \notin [a^*_-, a^*_+]$.

To summarize, the SR-neuron without self connection and a bias $\theta \notin [a^*_-, a^*_+]$ is bistable over half of the input domain, where one stable fixed point corresponds to the homeostatic state, and the other to the trivial state. The SR-neuron would then converge to one of the two fixed points depending on the initial conditions. On the other half of the input domain, the neuron is globally stable at the trivial fixed point. These observations are confirmed by Figure 21, showing bifurcation diagrams for the output $\tau(a)$ and the receptor strength $\xi$, under these conditions.

#### 4.3.1.2 *Trivial Dynamics with Self-Connection*

Adding a self-connection $w := c\,\xi\,\eta$ to the SR-neuron provides an additional input, so that the new input signal becomes $I(t) + c\,\eta\,\tau\big(a(t)\big)$,

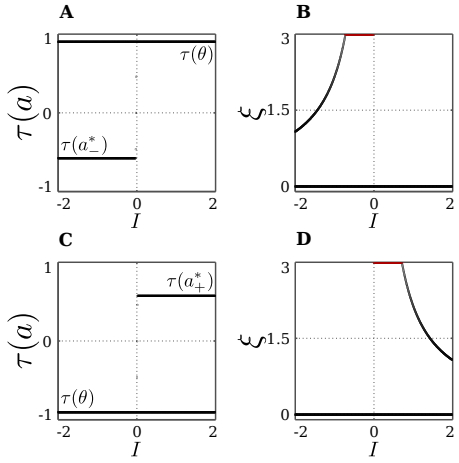


Figure 21: **Stable dynamics of a SR-neuron without self-connection for a non-homeostatic bias and varying input.** (A,B) Bifurcation diagrams of the output $\tau(a)$ and the receptor strength $\xi$ for varying input I and a *positive* non-homeostatic bias $\theta = +1.5$. (C,D) Bifurcation diagrams of the output $\tau(a)$ and the receptor strength $\xi$ for varying input I and a *negative* non-homeostatic bias $\theta = -1.5$.

where I again corresponds to the input from other neurons as in equation 59.

The linearization 60 around the trivial fixed point $x_0 = (\theta, 0, \eta_0)$ leads to the same eigenvalues 64, regardless whether the self connection is excitatory or inhibitory. This entails that the SR-neuron with self-connection is stable at the trivial fixed point for all I, when its bias is non-homeostatic, i. e., $\theta \notin [a^*_-, a^*_+]$.

On the other hand, the linearization 60 around the desirable fixed points $x_\pm = (a^*_\pm, \xi^*_\pm, \eta^*_\pm)$ leads to complex closed-form formulas for the eigenvalues that are of no help regarding the stability of these fixed points. However, we may rely on the 1-dimensional standard hyperbolic tangent neuron with self-connection:

$$a(t+1) = \theta + w\tau(a(t))\,. \tag{65}$$

This neuron is parametrized by its bias $\theta$ and self-weight $w$, and, for each parametrization, its asymptotic dynamics is easy to derive. Since both neuron models, the SR-neuron and the standard neuron, share the same transfer function tanh, it is possible to infer the stability of the former from the more familiar properties of the latter, given certain bias and self-weight values, as we show next.

#### 4.3.1.3 *Dynamics with Excitatory Self-Connection*

Suppose that the fixed points $x_\pm = (a^*_\pm, \xi^*_\pm, \eta^*_\pm)$ for a SR-neuron with self-connection are asymptotically stable. These fixed points then satisfy

$$a^* = \theta + \xi^*\big(I + c\,\eta^*\tau(a^*)\big)\,. \tag{66}$$

We start by setting $\theta = I = 0$. Then, the following holds

$$c\,\xi^*_\pm\eta^*_\pm = \frac{a^*_\pm}{\tau(a^*_\pm)} \approx 1.14 > 0\,, \tag{67}$$

which is only true for the case of an *excitatory self-connection*, i. e., $c = +1$. For an increasing excitatory self-connection and a zero bias, the standard additive hyperbolic tangent neuron 65 undergoes a *cusp catastrophe* [42] at the critical point $(\theta_c = 0, w_c = 1)$, and the neuron corresponds to a *bistable* system [92, 54]. Because the asymptotic self-weight $w^*_\pm = \xi^*_\pm\eta^*_\pm$ of the SR-neuron (equation 67) is larger than the critical value $w_c = 1$, the SR-neuron becomes bistable as well, which allows for *hysteresis* phenomena.

The critical point $(\theta_c = 0, w_c = +1)$ belongs to the *bifurcation set* $\mathcal{B}^+$, at which the standard hyperbolic tangent neuron 65 changes from being monostable to being bistable. The bifurcation set is parametrized by the bias and self-weight, and is derived in [92] for a standard neuron with logistic nonlinearity $\sigma(a) = (1 + e^{-a})^{-1}$. For a hyperbolic tangent nonlinearity, $\mathcal{B}^+$ is given by

$$\theta^2 = \frac{4(w-1)^3}{9w}\,, \tag{68}$$

while, at the fixed point $x_\pm$, the positive self-coupling $w^*_\pm$ of the SR-neuron changes *linearly* with the bias according to

$$w^*_\pm = \frac{a^*_\pm - \theta}{\tau(a^*_\pm)}\,. \tag{69}$$

The SR-neuron is bistable when $w^*_+$ or $w^*_-$ or both are above the bifurcation set $\mathcal{B}^+$. As such, the intersection of the bifurcation set $\mathcal{B}^+$ defined by equation 68 and the self-coupling of a SR-neuron as a function of the bias in equation 69, leads to the bias range $\theta \in [-0.11, +0.11]$, within which the SR-neuron is bistable. Outside of this range, both $w^*_+$ and $w^*_-$ are bellow $\mathcal{B}^+$, resulting in the SR-neuron becoming monostable. These findings can be verified by keeping $I = 0$ and varying the bias term $\theta$ as shown in Figure 22.

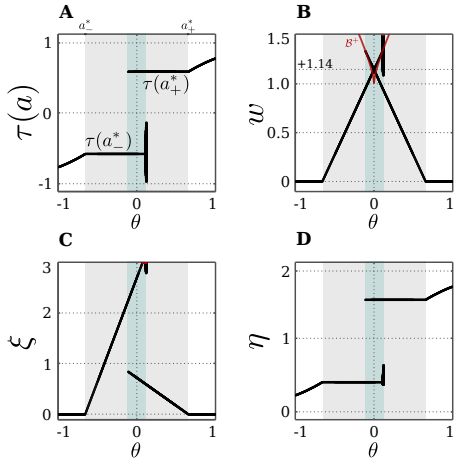


Figure 22: **Stable dynamics of a SR-neuron with excitatory self-connection for varying bias.** Bifurcation diagrams of (A) the output $\tau(a)$, (B) the *positive* self-weight $w = +\xi\eta$, (C) the receptor strength $\xi$, and (D) the transmitter strength $\eta$ for varying bias $\theta$. The gray-shaded area corresponds to the bias domain $\theta \in (a^*_-, a^*_+)$ at which the SR-neuron is *homeostatic*. The cyan-shaded area marks the hysteresis domain $\theta \in [-0.11, 0.11]$ at which the SR-neuron is *bistable*. The neuron shows a narrow range of quasi-periodic behavior when passing from $a^*_-$ to $a^*_+$. (B) The red curve denotes the bifurcation set $\mathcal{B}^+$ that marks the parameters domain where a standard additive hyperbolic tangent neuron is bistable. The SR-neuron seizes from exhibiting bistability when the positive self-coupling weight becomes lower than the bifurcation set.

We now assume that there exists a stationary input $I$, and that the bias $\theta \in (a^*_-, a^*_+)$. Under these conditions, the SR-neuron is homeostatic over the whole input domain, and it exhibits hysteresis phenomena over some input range, as is shown for $\theta = +0.5$ in Figure 23. For a narrow input range, one observes that the SR-neuron may show quasi-periodic oscillations when passing from one operating point to the other. These oscillations depend on the bias value and the parameters $\beta$, $\delta$, and $\gamma$.

#### 4.3.1.4 *Dynamics with Inhibitory Self-Connection*

For an *inhibitory self-connection*, i. e., $c = -1$, and no input, equation 66 can be solved when $\theta \notin [a^*_-, a^*_+]$. However, the trivial fixed point $x_0$ is stable at this bias domain, as shown in Section 4.3.1.2, and an inhibitory self connection can never satisfy the bistability condition bounded from below by the bifurcation set $\mathcal{B}^+$ (equation 68). This rules out the possibility for $a^*_\pm$ being stable, which entails that the SR-neuron is never homeostatic under these conditions. However, with no bias and a self-weight $w \approx -1.14$, the state $a^*_+$ is mapped to $a^*_-$ and vice versa, as suggested by equation 67. Thus, we expect a period-2 oscillation between the two states. Regarding the stability of

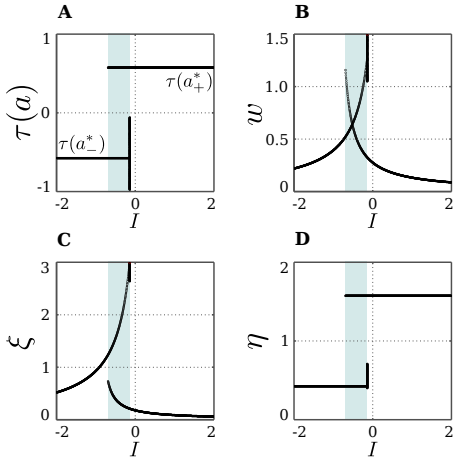


Figure 23: **Stable dynamics of a SR-neuron with excitatory self-connection for a homeostatic bias and varying input.** Bifurcation diagrams of (A) the output $\tau(a)$, (B) the *positive* self-weight $w = +\xi\eta$, (C) the receptor strength $\xi$, and (D) the transmitter strength $\eta$ for varying input $I$ and a *positive* homeostatic bias $\theta = +0.5$. The cyan-shaded area marks the hysteresis domain at which the SR-neuron is *bistable*. The neuron shows a narrow range of quasi-periodic behavior when passing from $a^*_-$ to $a^*_+$.

this oscillation, we return to he standard additive hyperbolic tangent neuron 65. For an increasing inhibitory self-connection, neuron 65 undergoes a *supercritical period doubling bifurcation* at the critical point $(\theta_c = 0, w_c = -1)$, and the neuron corresponds to a *period-2 oscillator*. This supports the existence of a stable period-2 oscillation for the SR-neuron when $(\theta = 0, w \approx -1.14)$, since this point lies within the period-2 parameter range of a standard hyperbolic tangent neuron.

Figure 24 demonstrates that the SR-neuron *do* oscillate with period-2 on the bias domain $(-0.95, 1.5)$ when $I = 0$. For zero bias, the self-weight oscillates due to the SR-dynamics with an average of $w \approx -1.14 < w_c = -1$, as is suggested by equation 67. Interestingly, the oscillatory dynamics for non-zero bias allow the SR-neuron's output to reach average values that are different from the canonical $\tau(a^*_\pm)$, or the trivial $\tau(\theta)$.

For a stationary input $I$, and a bias $\theta \in (a^*_-, a^*_+)$, a solution of equation 66 may exist, and the SR-neuron acts as a homeostatic unit for a certain input domain. Also, since the bias is within the oscillation domain for no input, the SR-neuron should oscillate with period-2 for some input range around 0. In fact, as shown in Figure 25, the homeostatic domain overlaps with the oscillatory domain for a narrow input range.

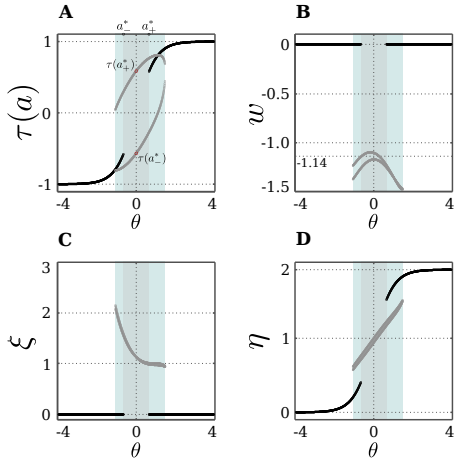


Figure 24: **Stable dynamics of a SR-neuron with inhibitory self-connection for a homeostatic bias and varying input.** Bifurcation diagrams of (A) the output $\tau(a)$, (B) the *negative* self-weight $w = -\xi\eta$, (C) the receptor strength $\xi$, and (D) the transmitter strength $\eta$ for varying bias $\theta$. The cyan-shaded area marks the domain at which the SR-neuron may *oscillate* with a period-2 between the two gray branches. The gray-shaded area marks the homeostatic bias domain $\theta \in (a^*_-, a^*_+)$ where the SR-neuron is globally oscillating. Outside of this domain, and depending on the initial conditions, the neuron may converge to the trivial fixed point $(\theta, 0, \eta_0)$, corresponding to the black branches. (A) The red dots mark the oscillation in activity between $a^*_+$ and $a^*_-$ when $\theta = 0$.

### 4.3.2 *Synaptic Dynamics in the Sensorimotor Loop*

In this section, we demonstrate for three examples how SR-neurons are able to operate successfully within the sensorimotor loop. On specific network structures, SR-neurons generate a desired behavior for coupled pendula, a hexapod walking machine, and a wheel-driven robot.

#### 4.3.2.1 *Coupled Reflex Loops*

Self-excitatory SR-neurons are good candidates for building oscillatory reflex loops. This was already shown in [94], where a single SR-neuron with excitatory self-connection was used to drive a pendulum with damping to oscillate with a constant amplitude. An angular position sensor is coupled to the reflex loop which drives the angle-controlled servomotor of the pendulum. Reflex loops generate smooth oscillatory movements which can be used for the control of limbs [139]. There are two important mechanisms involved in the generation of these oscillations. First, the integration of properties of the body – the body's inertia in the case of pendula or limbs – and the environment by means of the sensorimotor loop. Second, the nonlin-

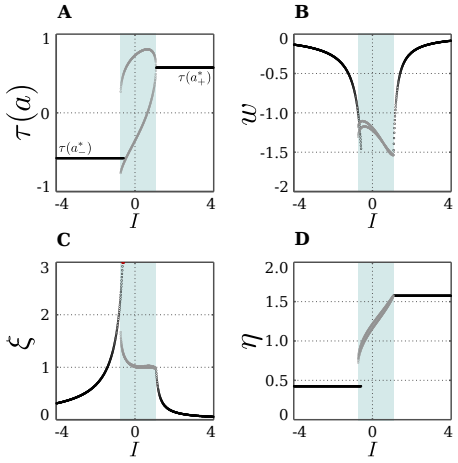


Figure 25: **Stable dynamics of a SR-neuron with inhibitory self-connection for a homeostatic bias and varying input.** Bifurcation diagrams of (A) the output $\tau(a)$, (B) the *negative* self-weight $w = -\xi\eta$, (C) the receptor strength $\xi$, and (D) the transmitter strength $\eta$ for varying input $I$ and a *positive* homeostatic bias $\theta = +0.5$. The cyan-shaded area marks the domain at which the SR-neuron *oscillates* with a period-2 between the two gray branches. The neuron shows a narrow range of input at which the oscillatory and homeostatic activity are overlapping.

earity of the neural elements, leading to a hysteresis effect. Stated differently, oscillations do appear if the system can "jump" from one fixed point to another by following the slow transients generated by the inertia of the body. If there is no hysteresis but the sigmoid is steep as in Figure 20A,C, oscillations may appear but with much smaller amplitudes, since there is no bistability interval to make the transients longer, and these oscillations will not be sufficient to provide the full swing of a limb for successful locomotion. In the case of a SR-neuron, hysteresis is provided by an excitatory self-connection (see Figure 23), which leads to bistable motor outputs. The time delay in the sensorimotor loop due to the physical characteristics of the body, its inertia namely, then causes the slow oscillations, referred to as *reflex oscillations*.

Before utilizing the SRN-dynamics and reflex loops for the locomotion of a Hexapod walking machine, we demonstrate that the coupling of two such reflex loops leads to synchronization or anti-synchronization, depending on whether the coupling is excitatory or inhibitory. Coupling the hysteresis elements of two reflex loops by symmetric excitatory connections (or a unilateral connection for that matter) will enforce the synchronization of the resulting oscillations. Correspondingly, inhibitory coupling will result in anti-synchronization. Two pendula are driven by servomotors placed at each pendulum's pivot and are angle-controlled (see Figure 26A). Each servomotor is driven by a motor neuron whose output range $(-1, +1)$ is

mapped to the desired angle range $(-180°, +180°)$. The desired angle is achieved through the servomechanism of position feedback, which applies a force of up to 0.5 N, until the error between the actual and desired angle is minimized. The parameters for the pendula are fixed to 0.2 kg for the bob mass and 0.5 m for the rod length. The angular position sensors are linear buffers, while the self-excitatory and the motor neurons are SR-neurons, as shown in Figure 26B,C.

Figure 26D,E demonstrate the dependence of the oscillation amplitude, and consequently its frequency, on the SR-neurons parameters. $\beta$ and $\gamma$ are fixed to 0.1 and 0.01 respectively. $\delta$ is either 0.001 (Figure 26D) or 0.005 (Figure 26E). The behavior of each pendulum is captured by its respective angular position sensor. By comparing the sensory signals, coming from the angular position sensors (top panel in Figure 26D,E), to that of the output of the motor neurons (bottom panel in Figure 26D,E), one notices that, despite the presence of damping, the pendula are oscillating harmoniously (sinusoidal motion with constant amplitude), although the outputs of the motor neurons show a different behavior. This can be confirmed by performing a Fourier analysis on the signals, which shows a single dominant frequency in the signal produced by the sensor, indicating that the pendulum generates a sinusoidal motion, i. e., a simple harmonic oscillation, while the motor produces multiple harmonies. This is illustrated in Figure 26F for $\delta = 0.001$. One also observes that the amplitude of oscillation depends on the quotient $\delta/\gamma$. For growing quotient $\delta/\gamma \leqslant 1$ the amplitude increases, and correspondingly, the frequency decreases (Figure 26G). For $\delta/\gamma > 1$, the hysteresis domain widens to the point where the changing input is not enough for the dynamics to cross the bistable region, so it converges to one of the stable fixed points, and oscillations stop. These results, illustrated on the anti-synchronous case with lateral inhibition, also apply to the synchronous case with lateral excitation. The two cases are demonstrated in **Movie S1**.

Interestingly, for pendula with nonidentical bob masses and rod lengths, one observes the emergence of phase-locking phenomena, but with differing oscillation amplitudes of the two pendula, as shown in Figure 26H,I and also in **Movie S1**. A mathematical analysis of this result is currently under development.

#### 4.3.2.2 *Controlling a Hexapod Walking Machine*

It was demonstrated in [94] that reflex loops of SR-neurons can drive the three joints of a single leg to induce locomotion of the modular hexapod walking machine Octavio [138], shown in Figure 27A. Having observed that excitatory (inhibitory) coupling of SR-neurons in reflex loops leads to their synchronization (anti-synchronization), it follows that this method may be used to couple the neurocontrollers of single legs to get a walking behavior from the 18 degrees of free-

dom of the hexapod walking machine. For setting up a promising coupling structure, we assume that the protractor/retractor joint, named the ThCx-joint, of the left and right frontal legs, `L1` and `R1` respectively, gives the leading signals for the middle and hind legs (`L2`,`R2` and `L3`,`R3`), and that the movement of these joints (of `L1` and `R1`) needs to anti-synchronize. Thus, reflex loops of ThCx-joints of `L1` and `R1` are laterally coupled by inhibitory connections. The reflex loop of the ThCx-joint of the middle leg `L2` (`R2`) receives an inhibitory synapse from the reflex loop of the ThCx-joint of `L1` (`R1`), while the corresponding reflex loop of `L3` (`R3`) receives an excitatory synapse from its counterpart in `L1` (`R1`). This coupling scheme, shown in Figure 27B, should then lead to a typical *tripod gate*. The rationale behind this is as follows. The controller of each leg consists of three reflex loops. This entails that a leg could be considered as a high-dimensional reflex oscillator. According to the results from the previous section, coupling two reflex oscillators with an inhibitory connection would lead to their anti-synchronization, and with excitatory connection would lead to their synchronization. In other words, `L1` and `L3` would synchronize, due to the excitatory coupling between the two. `L1` would also synchronize with `R2`, since the former is coupled to the latter by a chain of two inhibitory connections, which is equivalent to an excitatory coupling. The synaptic delay between `L1`, `R2`, and `L3` is maximally two time steps, which has no effect and can be ignored, given the period of the reflex oscillations. This entails that the triplet (`L1`,`R2`,`L3`) would go through the *stance phase* simultaneously, while the anti-synchronous triplet (`R1`,`L2`,`R3`) would be in the *swing phase*, which results in a tripod gate.

Starting with a single leg reflex loop controller, and demanding the same controller structure for all the six legs, the described coupling scheme did not immediately lead to successful walking. To circumvent this, the evolution environment of the NERD Toolkit [101] was utilized for evolving the structure further, and optimizing the bias values. The fitness function was given as "the distance walked in forward direction in a given number of time steps". Regarding bias terms, a symmetry constraint was set to have identical left and right leg modules. As for the network structure, a constraint is set such that all legs are identical, and they follow the structure of `L1`. Other constraints, such as distance of the central body from ground or allowed joint angles, were not used in this case (also compare von Twickel et al. [137] and von Twickel et al. [138]). Section 4.5 outlines the details of the evolution process. Following evolution, the bias values of frontal, middle, and hind legs turned out to be different for achieving better forward walking. This is due to the fact that their task is different: frontal legs pull the body, while hind legs push the body. Figure 27B displays the complete modular neurocontroller. The resulting modules have identical structures for all legs due to the im-

posed constraints, and one of these modules is detailed in Figure 27C. SRN-parameters for this controller are set to $\beta = 0.1$, $\gamma = 0.1$, and $\delta = 0.2$, which matches their values in the reflex loop controller of a single leg [94].

In addition to the simple reflex loops of the three joints from which evolution started, we find here an additional neuron $SRN_4$ with inhibitory self-connection, which forms an odd 2-ring with the neuron $SRN_1$ (the reflex loop of the ThCx-joint). This self-inhibitory neuron $SRN_4$ and its connections were added by structure evolution. This additional structure induces period-2 oscillations, which at the first sight, might appear as superficial or destructive. However, all controllers that succeeded in achieving the forward motion of the body included this oscillatory neuron, and analysis shows that inhibiting these oscillations will result in a brake down of walking. Figure 27F depicts the oscillatory odd 2-ring network, and Figure 27D,E show the bifurcation diagrams of its SR-neurons' output for a changing input signal. One observes that the module behaves as a period-2 oscillator over most of the input domain. It oscillates around positive amplitudes for negative inputs and around negative amplitudes for positive inputs. The asymptotic dynamics also shows a narrow regime of quasi-periodicity in the middle, which has no effect on behavior, since the dynamics passes over this domain for a short transitory period. This becomes clearer from Figure 27G, which illustrates the effect of a sensory signal sweeping over the interval $[-1, 1]$ on the oscillatory module. The sensory signal sweeps over the interval with a frequency comparable to that of the ThCx-joint oscillations. This further highlights the dependence of the oscillation amplitudes on the sensory signal. We postulate that these oscillations are necessary for behavior because they increase the range of admissible outputs. By having a changing mean value, which depends on the input strength, $SRN_4$ allows for motor signals that are not restricted to the $\tau(a^*_\pm)$ values provided by reflex loops. Furthermore, the oscillatory effect of $SRN_4$ is not seen anymore on the sensory signals coming from the joint angle sensors, as illustrated in Figure 27H. The oscillatory signal also has no direct effect on behavior in the sensorimotor loop. As demonstrated in Figure 27I, it only results in small amplitudes at the motors, and the effective motor signal corresponds to the mean value of these oscillations.

Walking starts with feet having ground contact. That the walking pattern is not a perfect tripod gate, but still represents a reasonably good walking behavior, can be read from Figure 28. As the walking pattern of OCTAVIO in Figure 28 indicates, the stance phase of the middle legs are considerably shorter than those of the frontal and hind legs. Nevertheless, there is a uniform timing of the phases so that walking on a flat surface is stable. That is, one notices that the stance phases of the triples (L1,R2,L3) and (R1,L2,R3) proceed almost

periodically from one step to the next, which is a signature of stable tripod forward locomotion (see **Movie S2** for demonstration).

In summary, although the suggested basic neural structure, the simple reflex loops, does not produce the desired behavior, an additional structure, even when adding oscillations, will generate this behavior. Here it appears that walking is driven by mean values of fast oscillations. The amplitudes are small at the motors and integrated away by the body as can be seen from the (noisy) sensory signals. One can also observe that inputs crossing bifurcation points, as is shown for instance in Figure 27G, do not derogate the desired behavior.

#### 4.3.2.3 *Obstacle-Avoidance with a Wheel-Driven Robot*

The SRN-dynamics is not restricted to the control of coupled reflex loops. We now show how a network of SR-neurons can be used by a wheel-driven robot (Figure 29A) to navigate its environment and avoid obstacles (Figure 29B). The wheel-driven robot is called ALICE (see Figure 29A). ALICE is endowed with five long-range distance sensors in the frontal part of the body and are used for detecting obstacles. Each of ALICE's two wheels is controlled independently by a motor neuron that drives a velocity-controlled servomotor. Each motor neuron's output range $(-1, +1)$ is mapped to the corresponding servomotor's desired velocity range $(-20°, +20°)$ per time step. The desired velocity is achieved through the servomechanism of position feedback, which applies a torque of up to $2\,\mathrm{N{\cdot}m}$, until the error between the actual and desired velocity is minimized. A preliminary example for a Khepera robot was also presented in [143], where the neurons had a different SRN-dynamics and a logistic sigmoidal nonlinearity, and a simpler neurocontroller was used. In what follows, we elaborate on the role of the current SRN-dynamics in achieving a successful obstacle-avoidance behavior, and we compare the behavior to the previous approach.

Figure 29C shows the control network using SR-neurons for obstacle-avoidance. It consists of three layers. The sensory layer assembles the five distance sensors into three groups corresponding to left, center, and right distance sensors, i. e., $S_{\text{left}}$, $S_{\text{center}}$, and $S_{\text{right}}$ respectively. The input layer projects into a layer of hidden neurons of the self-regulating type. The hidden layer in its turn projects to the motor layer. The three sensor neurons and the motor neurons $M_{\text{left}}$ and $M_{\text{right}}$ are standard additive neurons with a hyperbolic tangent transfer function.

In order to understand the functioning of this network in controlling obstacle-avoidance, and the role of the self-regulating dynamics in achieving this, we look in more detail into the hidden layer. It consists of two SR-neurons: $\text{SRN}_{\text{left}}$ and $\text{SRN}_{\text{right}}$. Both are receiving input from $S_{\text{center}}$. $\text{SRN}_{\text{left}}$ is connected to the left-side distance sensors and in turn projects to the right motor. The reverse is true for

$SRN_{right}$. The SR-neurons are self-coupled with excitatory synapses. As shown in Figure 29D, an obstacle approached from the left side inhibits $SRN_{left}$ and the sign of its output changes into negative. This in turn leads the velocity of the right wheel to become negative which corresponds to a backward rotation of the wheel. Due to the lateral inhibition of $SRN_{right}$ by $SRN_{left}$, the left motor neuron $M_{left}$ is excited, and the left wheel rotates faster in the forward direction. The combination of the backward rotation of the right wheel and the forward rotation of the left, leads ALICE to turn to the right, and away from the left-side obstacle.

The switch of the sign of a self-regulating neuron in the hidden layer is particularly important when approaching a narrow corner. It is simply not sufficient for the output of the neuron to decrease due to the inhibition from the distance sensors. If this switch did not occur, ALICE would turn right but it would keep going forwards with less velocity, and it would not be able to avoid the sharp corner. In addition, the hysteresis effect resulting from the self-excitation allows the SR-neuron to memorize the history of its input, which is necessary for the avoidance behavior to continue in the same direction, preventing the robot from getting stuck (see **Movie S3**). Figure 29E,F show how the dynamics of $SRN_{left}$ changes when a narrow corner is approached from the left (Figure 29G). The bifurcation diagram shows a hysteresis phenomenon where the neuron's output is bistable for a narrow range of input (recall the analysis of self-excitation above). The sign of the output $SRN_{left}$ only changes when the input is strong enough to cross the hysteresis domain. Bistability, and the resulting hysteresis, which are necessary for behavior, cannot be explained from a particular component of the 6-dimensional dynamical system that is the hidden layer. The same network structure with no self-regulating connectivity can achieve the same effect if the weights were fine-tuned by hand or through evolution. For instance, it was shown in [54] that with a similar controller but with standard hyperbolic tangent neurons, the self-connections should be set above the critical value of $w_c = 1$ for the hysteresis phenomenon to occur. Figure 29E,F shows, however, that with SR-neurons, the phenomenon occur without the self-connection crossing the critical value. These observations are also confirmed in the plots in Figure 29D. In other words, these properties *emerges* from the SRN-dynamics. In the previous study by Zahedi and Pasemann [143], the different SRN-dynamics and neurocontroller, where also capable of memorizing the history of the stimulus, allowing the Khepera robot to avoid narrow corners. However, due to the logistic sigmoidal nonlinearity being strictly positive, the robot was only capable of slowing down when turning away from narrow corner. On the other hand, the ability of current model to generate a negative motor output allows ALICE to turn in place, and as shown

in Figure 30, to avoid more challenging obstacle scenarios, where the robot is at a close proximity to the walls and corners.

## 4.4 DISCUSSION

We demonstrated that SR-neurons have a wide range of functions, depending on their bias terms and inputs coming from sensors or other neurons in the network. Without self-connection, they can serve as self-regulating units that are able to stabilize their activation around two desired outputs, which, in a way, correspond to low ($a^*_-$) and high ($a^*_+$) activity. For bias terms outside the interval $(a^*_-, a^*_+)$, SR-neurons may get dysfunctional, i. e., their receptor strength converges to zero. Adding self-excitation to a neuron preserves the neuron's homeostatic properties, and introduces bistability, which allows the neuron to exhibit a hysteresis effect over a certain input range. A second operational mode of SR-neurons, due to self-inhibition, is that of a period-2 oscillator with varying and shifted amplitudes, depending on the bias and input.

Afterwards, we studied the properties of SR-neurons when operating in the sensorimotor loop. That is, SR-neurons are driven by changing sensory inputs, and they generate motor signals accordingly, which in their turn drive the actuators of an animat. From experiments with pendula, single legs [94], and hexapod walking machines, one concludes that SR-neurons are suitable for coupling reflex loops, because desired sensory inputs do change frequently or are oscillating. As a result, and due to SRN-dynamics, appropriate mean values of synaptic efficacy adjust themselves properly. However, examples from networks controlling wheel-driven robots demonstrate that the function of SR-neurons is not restricted to reflex loops. Even if sensory inputs are not often changing, as is the case when no obstacles are present, mean values of the synaptic efficacies self-adjust, depending on the connectivity and the bias values, so that a desired behavior is achieved. It is worth noting that in the example of the wheel-driven robot, motor neurons are not self-regulating. However, choosing them to be self-regulating leads qualitatively to the same behavior.

The SR-neuron with excitatory self-connection is of particular importance for the control of an animat in the sensorimotor loop. The hysteresis effect such a module exhibits provides the neuron with a working memory of the stimulus history, which allows it to produce oscillatory output. The period of these reflex oscillations depends on the width of the hysteresis domain, which is a function of the SR-neuron's parameters. This was the basis for generating the locomotion behavior of the hexapod walking machine. This dependence on input history also allowed the wheel-driven robot to turn in place and away from sharp corners by "remembering" the direction of the

obstacle long enough to swing away from it. A SR-neuron with excitatory self-connection is a particular instance of a class of systems that exhibit bistability, and as a corollary, hysteresis. Namely, every ring of standard sigmoidal neurons undergoes a bifurcation for some values of the weights and biases, if and only if the number of inhibitory synapses is even, which leads to the existence of two fixed point attractors (bistability), in addition to coexisting periodic attractors [93]. The bistability phenomenon is also relevant to genetic networks, and is shown to exist in these systems under similar conditions [4]. The significance of SRN-dyanamics is that it pushes the neuron's parameters *autonomously* towards the bistable regime, allowing it to implement a form of short-term plasticity, and the resulting working memory of input history [147, 1, 79].

The design of the SR-neurons with two operating points provides a natural implementation of the principles of *step mechanisms* and *ultrastability* suggested by Ashby as main ingredients of adaptive behavior [8]. These concepts are better explained through the example of obstacle-avoidance by the wheel-driven robot. The essential variables of this system are the readings of the distance sensors, which should remain close to their minimum for the survival of the robot. When the stability of the moving-forward behavior is broken due to the approach of an obstacle from the left, it triggers the change of value of a step mechanism implemented in the left SR-neuron by the SRN-dynamics, while no change occurs at the right SR-neuron, i. e., a new behavior, turning-right, becomes stable. In other words, while the actions of the robot are continuous, only four stable modes of behavior are identified by the two step mechanisms provided by the two SR-neurons. These allow the robot to keep its essential variables within the desired range: the robot's behavior is ultrastable.

Synaptic plasticity with homeostatic regulation has been applied several times in the context of *evolutionary robotics* [31, 45, 109], and has been related to Ashby's theory [8] as well. In these studies, neurocontrollers for autonomous robots are evolved such that each synapse is assigned a synaptic plasticity rule from a set of possible variants of Hebbian plasticity. Synaptic dynamics get activated only when neural output diverges from a selected homeostatic domain. Others investigated comparable mechanisms where homeostasis was also discussed in the context of walking behavior [51, 52]. Our approach differs from those in that homeostatic stability is achieved using a single plasticity mechanism, and in that it is written completely in dynamic terms. The SRN-dynamics is also related in part to the BCM theory [12, 25]. Both the BCM rule and SRN-dynamics achieve stability of synaptic weights through a quadratic dependence on postsynaptic activity, and on a threshold that separates the regimes of synaptic depression and potentiation. However, unlike BCM learning, it is not necessary for the threshold $a^*_\pm$ of SRN-dynamics to be sliding. This

is due to the fact that homeostatic stability, as is the case in Triesch [132], is explicitly implemented in the receptor dynamics. However, the SRN-dynamics differs functionally from the BCM rule in that the latter is a learning rule while the former is not.

Obstacle-avoidance with wheel-driven robots is a benchmark task in neural control, and successful controllers were found either through synaptic plasticity of the weights that connect sensors to built-in reflexes [44], the homeostatic regulation of a GasNet control networks with artificial chemicals during evolution [134], or maintaining homeostasis by modulating the random reconfiguration of the conroller's parameters by artificial hormones [96]. The SRN-dynamics control of the wheel-driven robot does not incorporate learning as in [44], and unlike [134, 96] where the robot has to carry multiple tasks concurrently, ALICE's behavior is restricted to obstacle-avoidance. However, in these studies, the neurocontrollers are derived and tested in spacious maze-like environments [44], or in a featureless rectangular arenas [134, 96], and would not avoid narrow impasses or sharp corners. On the other hand, neural control with SR-neurons exploits the full potential of the recurrent neural network and the bistability resulting from the synaptic dynamics to succeed where other controllers would fail.

A hallmark of the current study is the derivation of a stable forward walking behavior of a hexapod with 18 degrees of freedom, corresponding to the 18 joints of the insect-like robot OCTAVIO. Beer and Gallagher used an evolutionary process to derive a neurocontroller to achieve stable walking of a hexapod [9]. While that hexapod also contained 18 degrees of freedom, it only had 6 joints. Achieving stable behavior of a quadruped or a hexapod with multiple joints per leg is far from trivial. For instance, Shim and Husbands used intrinsic chaos of weakly-coupled central pattern generators to search for a neurocontroller of a quadruped with 8 degrees of freedom, and later stored the successful controllers in the connections between the oscillators using a form of synaptic plasticity [116]. While the same strategy lead to a stable forward locomotion of a swimming robot, Shim and Husbands reported that the behavior of the quadruped broke after some time [116]. However, a stable 18-joints hexapod forward locomotion is achieved using Walknet [111]. Walknet allows for a verity of behaviors and extensions to match the behavioral repertoire of a stick insect [112]. This flexibility comes with the price of a highly complex and heavily engineered controller with many non-neuronal elements. On the other hand, the SRN-controller of OCTAVIO provides from simple design intuitions, and a small contribution from evolution (a single hidden neuron for each leg), a minimal architecture with dynamic synapses that is, to this point, unprecedented.

From the experiments described here, it is obvious that an effective control also depends on convenient SR-parameters, which were

currently picked by hand. However, these parameters can in principle be optimized using evolutionary techniques provided, for instance, by the NERD Toolkit [101]. The same applies to bias terms. An alternative is to find suitable bias dynamics, which is a subject of current research. Often, there are reasonable constraints on the structure of more complex neural controllers. The NERD evolution environment allows the use of functional substructures, symmetry constraints, modularization, specific synaptic communication lines or nerve bundles, and a variety of different neuron types like sensor neurons, bias neurons, standard neurons, and SR-neurons [101]. These capabilities were used, for instance, for the control of forward/backward locomotion of a single leg [94], and the current control of locomotion of the hexapod walking machine.

In addition, the connectivity of the network is equally essential to the synaptic dynamics for deriving an effective control. Instead of finding solutions in a high-dimensional real-valued parameter space, evolution can be utilized to find *only* those $(-1, 0, +1)$ connectivity structures on which the SRN-dynamics leads to a satisfactory behavior. However, finding the real-valued bias terms remains a bottleneck, due to the lack of an appropriate bias dynamics. An alternative approach to evolution in refining an agent's behavior is the introduction of proprioceptive units that dissipate artificial neuromodulatory signals. These units are placed within preconfigured networks that are separate from the robot's neurocontroller, and are responsible for monitoring the robot's behavior. For instance, a monitoring network may be responsive to the robot's failing to avoid an obstacle, or approach food sources. When either undesired behavior occurs, the monitoring network would stimulate its corresponding proprioceptive unit. The latter would then release a signal that initiate the learning of SR-parameters, bias terms, or connectivity structure *during* the lifespan of the robot. The neuromodulatory signals stop when the robot's behavior is appropriate and the monitoring networks are deactivated [103].

In the context of connectivity, an interesting property of a SR-neuron is that it can turn off its input by reducing its receptor strength down to zero, thereby becoming a "dead neuron". This fact may be used to facilitate the evolution of effective connectivity structures. For example, starting with a fully connected network, the bias term of a neuron may enter the dead neuron domain, either through evolution or by accommodating *bias dynamics*. Taking such a SR-neuron, which can not contribute to a behavior-relevant synaptic dynamics anymore, out of the network will correspond to a mechanism like programmed death of a cell and will prune the network structure.

| PARAMETER | INITIAL STAGE | FINAL STAGES |
|---|---|---|
| Population size | 200 | 100 |
| Initial bias values | $\pm 0.1$ | |
| Evaluation time | 1000 | 5000 |
| Number of tries | 1 | 5 |
| Elite individuals | 10 | 3 |
| Tournament size | 5 | 13, down to 5 later |
| Bias mutation probability | 0.1 | 0.1 |
| Bias range | $[-0.3, +0.3]$ | $[-0.3, +0.3]$ |
| Distribution of bias change | $\mathcal{N}(0, 0.01)$ | $\mathcal{N}(0, 0.01)$ |
| Maximum number of hidden neurons | 5 | |
| Probability of inserting or deleting a neuron | 0.01 | 0 |
| Maximum number of new synapses | 10 | |
| Probability of inserting or deleting a synapse | 0.1 | 0 |

Table 2: **Parameters of the evolution process.** Parameters are changed by the experimenter at different points of the evolution process. The initial values of the parameters are shown along with the values at the end of evolution.

## 4.5 APPENDIX: EVOLVING A NEUROCONTROLLER FOR HEXAPOD LOCOMOTION

The neurocontroller for forward locomotion of the Hexapod walking machine OCTAVIO was evolved using the *NERD Toolkit* [101], an open source software for research in evolutionary robotics and neurorobotics. The NERD Toolkit implements the *Interactively Constrained Neuro-Evolution* method (ICONE) [102], which allows the experimenter to adjust parameters during the evolution process, and to reduce the search space by imposing constraints on the admissible network structures.

The evolution process proceeds as follows. First, an initial population of neurocontrollers is created. At each generation, individuals are evaluated according to a given fitness function. Then, an elite of the population is passed directly to the next generation. The population reaches its full capacity through a selection strategy among the individuals of the last generation. This is followed by adaptation operations on the resulting population that involves modifying the

bias terms and introducing new neurons and synapses, depending on the imposed constraints. The experimenter may then decide to change certain parameters of the evolution process. Table 2 summarizes the relevant parameters and their values during the initial and final stages of the evolution process. The following sections elaborate on the different elements of the evolution process by which the successful neurocontroller of OCTAVIO was found.

### 4.5.1 *Initial Population and Evolution Constraints*

Each individual has the single leg reflex loop controller structure of its frontal left leg `L1` cloned to all six legs. The single leg controller consists of 3 positively self-coupled hidden SR-neurons that are fixed during evolution. Evolution then starts with 200 individuals. Regarding bias terms, a symmetry constraint was set to have identical left and right leg modules. As for the network structure, a constraint is set such that all legs are identical, and they follow the structure of `L1`. These constraints were enforced during the whole evolution process. In the initial generation, bias values of the neural modules of the legs `L1`, `L2`, and `L3` were randomized to $\pm 0.1$, and then copied to those of `R1`, `R2`, and `R3`, in order to preserve the lateral symmetry constraint.

### 4.5.2 *Evaluation and Selection*

At each generation, the performance of all individuals is evaluated for several *tries* by the fitness function given by "the distance walked in forward direction in a given number of time steps". The final fitness of an individual is the average fitness across tries. Introducing several tries aims at evaluating an individual for changing neural initial conditions and different noise in the sensors and motors. The evolution process is split into several stages regarding evaluation time. It starts at 1000 steps and a single try for an initial phase of several generations, after which, evaluation time is increased by the experimenter, and it reaches 5000 steps, 5 tries each, at the final stages of evolution. The decision to increase the evaluation time and the number of tries is mainly to ensure smoother and regular locomotion. It may also be due to the plateauing of the population overall fitness, following the eradication of individuals that fail to produce substantial forward locomotion in the shorter allotted evaluation time. In addition to increasing the evaluation time, the experimenter reduces the population size down to 100 individuals at the final stages of evolution.

After evaluation, specifying the next generation involves selecting a small *elite* of the fittest 3 individuals (the fittest 10 at the first stage). Then, the population is completed to its capacity using a *tournament selection strategy* [77], which goes as follows. A subpopulation of 5 in-

dividuals at the beginning and up to 13 individuals in intermediate stages is randomly selected with replacement. The fittest individual of this subpopulation wins the tournament and is passed to the next generation. This controls the distribution of fitness values of a population during evolution.

### 4.5.3 *Adaptation*

Before proceeding to the next generation, all individuals undergo three adaptation processes: inserting and deleting neurons, inserting and deleting synapses, and bias mutation. Probabilities for these operations are controlled by the experimenter.

The maximum number of allowed hidden neurons per leg module is 5 (maximally, 2 new neurons). Neurons are added to the hidden layer of the neural module of `L1`. Each of the potential 2 neurons has a 0.01 initial probability of creation. The maximum number of new synapses is 10, and each has 0.1 initial probability of creation. Both sensory and hidden neurons are potential sources of newly added synapses, while only hidden neurons are potential targets, all with uniform probability. The weight of the added synapse is then randomized to $\pm 0.1$, and is now subject to the SRN-dynamics. After the first initialization, a synapse never changes its sign. The deletion of neurons and synapses use the same insertion parameters. The original elements (the three hidden neurons and their efferents and afferents) are protected from deletion.

The bias terms of the hidden and motor neurons of the three left legs are all potential candidates for mutation by a Gaussian random walk. The bias of each neuron is selected for mutation with 0.1 probability. If a neuron is selected, its bias is adjusted by a random value drawn from a normal distribution with 0 mean and 0.01 variance. All bias terms are truncated to remain in the range $[-0.3, +0.3]$. After mutation, the new network structure is cloned to the other legs, and bias terms of the neurons from the left legs are copied to their contralateral counterparts to preserve the constraint.

After a successful structure with high fitness is found, the probabilities of inserting new synapses and neurons is set to 0 by the experimenter, so that evolution is focused on refining the bias terms.

#### ACKNOWLEDGEMENT

The authors thank Josef Behr, Andrea Suckro, and Florian Ziegler for testing and refining the simulation models and for their contributions to the NERD Toolkit. Andrea Suckro and Florian Ziegler also compiled the supplemental movies that accompany the current manuscript. This research was partially funded by the German Research Foundation (DFG) priority program 1527.

## SUPPLEMENTAL DATA

Experiments are performed using the NERD Toolkit [101], an open-source software that can be used under an extension of the GNU General Public Licence (GPL). The Software is available at

`http://www.ultopia.de/drupal/nerddoc/`

The following are provided as supplementary material:

**Movie S1** corresponding to Figure 26, coupled reflex loops.

**Movie S2** corresponding to Figure 27, hexapod forward walking, tripod gate.

**Movie S3** corresponding to Figure 29, obstacle-avoidance with wheel-driven robot.

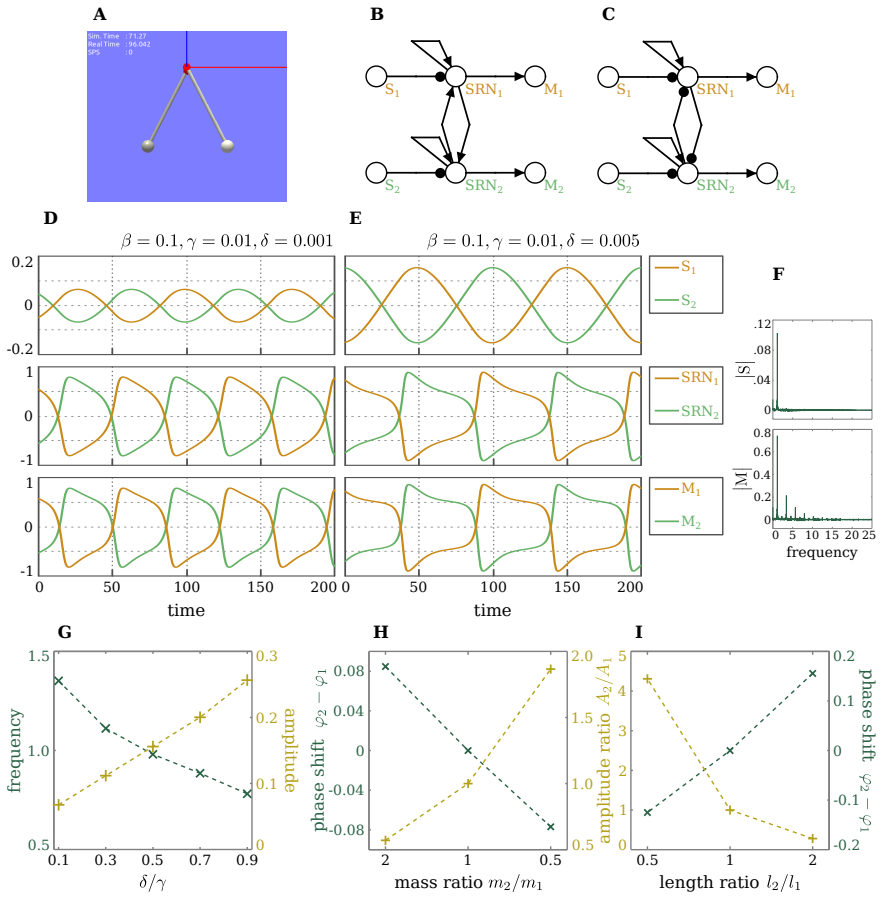


Figure 26: **Coupled reflex loops.** (A) Simulator of two identical pendula. The bob mass $m_{1,2} = 0.2\,\text{kg}$ and the rod length $l_{1,2} = 0.5\,\text{m}$. (B,C) Coupled reflex loops for controlling the two identical pendula. Angular velocity sensors are linear buffers. The self-excitatory and the motor neurons are SR-neurons. (B) Lateral excitation leads to synchronization. (C) Lateral inhibitory leads to anti-synchronization. (D,E) Outputs of the angular velocity sensors, the SR-neurons, and the motor neurons of the two identical pendula, oscillating *anti-synchronously* due to *inhibitory* coupling. The parameters of the SRN-dynamics are set such that (D) $\delta/\gamma = 0.1$, or (E) $\delta/\gamma = 0.5$. (F) Fourier analysis of the signal coming from the sensor (top) and the motor (bottom) for $\delta = 0.001$. (G) The effect of the quotient $\delta/\gamma$ on the amplitude and frequency of the oscillations. **(H-I)** Nonidentical pendula. The effect of changing (H) the bob mass ratio and (I) the rod length ratio on the phase shift between the two pendula and on the relative amplitude of their oscillation.

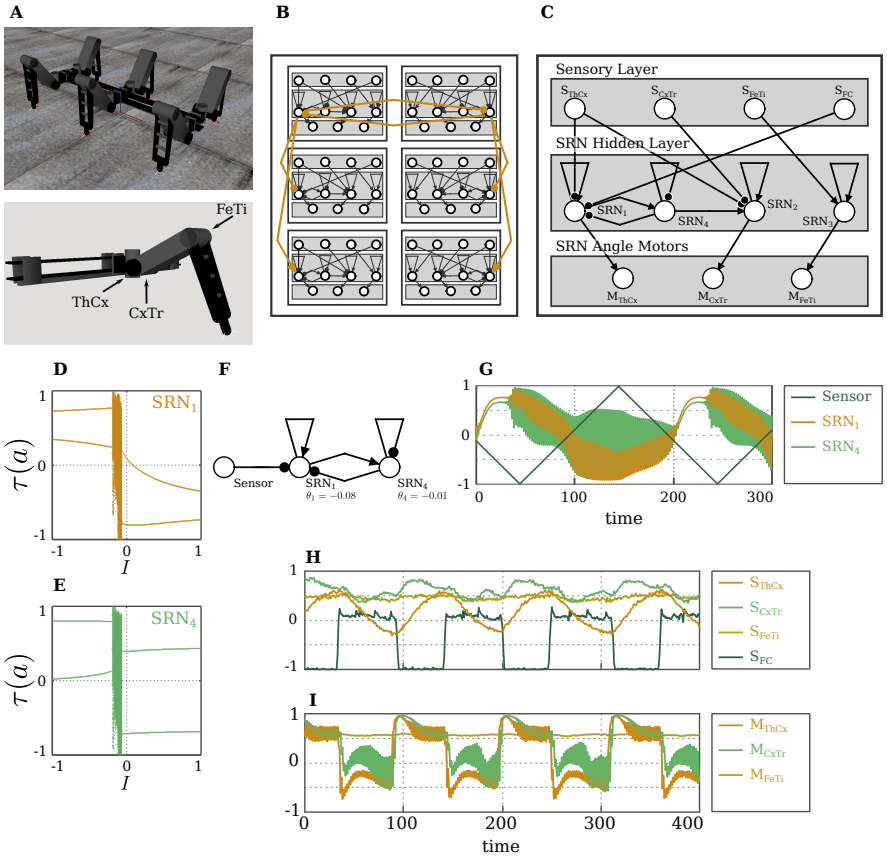


Figure 27: **Hexapod walking.** (A) The physical simulation of the hexapod walking machine Octavio (top), and of a single leg with the three joints marked (bottom). (B) The SRN-controller for the hexapod machine, highlighting the coupling scheme between the legs. (C) The SRN-controller of a single leg. (D,E) Bifurcation diagrams for varying sensory input I coming to the neuron $SRN_1$. **(D)** the output of $SRN_1$, and (E) the output of $SRN_4$ in the leg module L1. (F) The oscillatory odd 2-ring network in the leg module L1. (G) The outputs of $SRN_1$ and $SRN_4$ for a sweeping sensory signal with a frequency comparable to that of the ThCx-joint oscillations. (H) Sensory and (I) motor signals of the left frontal leg module L1 during the hexapod walking.

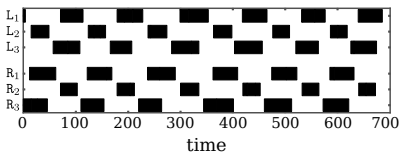


Figure 28: **Tripod gate of Octavio.** The walking pattern resulting from the neural control network of Octavio. Black regions mark the stance phase of the corresponding leg, which is the time the foot is at contact with the ground.

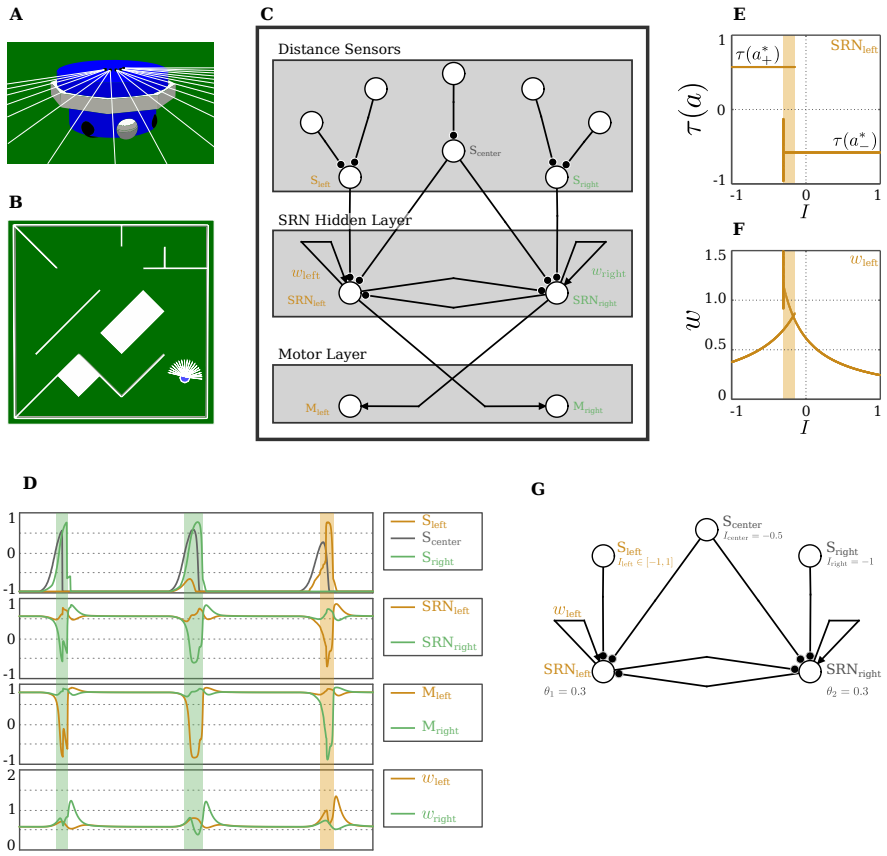


Figure 29: **Obstacle-avoidance with a wheel-driven robot.** The two-wheeled robot ALICE with distance sensors and wheels shown. (B) A typical navigation environment with obstacles. (C) A SRN-controller for obstacle-avoidance. Both sensory and motor layers have standard neurons. (D) Plots illustrating the dynamics of obstacle-avoidance behavior. From top to bottom: Output of the distance sensors; Output of the self-regulating neurons; Output of the motor neurons; Strength of the self-coupling of the SR-neurons. The shaded areas mark the time when $S_{left}$ or $S_{right}$ are sufficiently stimulated, and is color-coded to match the side from which the obstacle is approached. (E,F) Bifurcation diagrams for varying input from the sensor $S_{left}$ of (E) the output $\tau(a)$ and (F) the self-weight $w = +\xi\eta$ of $SRN_{left}$. The shaded area marks the bistable domain. (G) The hidden layer of the obstacle-avoidance control network at which the bifurcations (E,F) are observed. A narrow corner approached from the left is emulated by stimulating the sensor $S_{center}$ and varying the input from the sensor $S_{left}$.

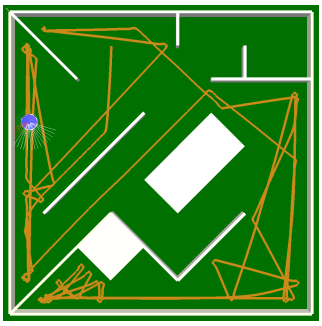

Figure 30: **Robot trajectory during obstacle-avoidance behavior.** The robot is capable of avoiding sharp corners, while being at a close proximity to the walls, due to hysteresis effects of the neurocontroller and the hyperbolic tangent nonlinearity. The latter allows the robot to stop and turn in place.

# Part III

# CONCLUSION

We summarize the achievements of this thesis, and we outline future directions.

# 5 CONCLUSION AND FUTURE PERSPECTIVE

## 5.1 THESIS' ACHIEVEMENTS

Homeostasis is a universal biological process. By the virtue of stability being a favorable state, natural selection perfected homeostatic mechanisms that act on multiple levels of biological organization. This includes regulatory genetic networks [80], single neurons and their afferents [145, 133], recurrent neural networks [126, 100], and even the whole organism's innate [20] and learned behavior [8].

In this thesis, we focused on the recurrent neural network level of organization. The reason behind this choice relates to the questions we aim to answer: Is homeostasis a regulatory process only or is it computationally relevant? If it is computationally relevant, what is the most general class of neural computations we expect homeostasis to contribute to solving? What are suitable models for this class of computations? Finally, are the answers to these questions specific to neural information processing, or are they transferable pieces of knowledge? We review the contributions of this thesis in answering these questions in the following sections.

### 5.1.1 *Is Homeostasis Computationally Relevant?*

This question is answered in the affirmative by maximum entropy models of homeostatic plasticity [123, 131, 132, 110] (see Section 1.2.2). These studies show that homeostatic plasticity enhances neural information transmission, and when interacting with associative synaptic plasticity, discovers nonlinear features of the input stream.

The three studies in this thesis also confirm the computational role of homeostasis. In Paper I, we showed that homeostatic intrinsic plasticity increases recurrent neural networks' entropy and, by doing so, it boosts the neuronal bandwidth. We showed that this is a necessary ingredient for carrying out nonlinear spatiotemporal computations, and for network noise-robustness.

We introduced in Paper II a model of homeostatic plasticity acting on the temporal distances between the computational units, or v-nodes, of a single node delay-coupled reservoir (DCR). Consistent with our findings in the previous paper, we showed that homeostatic plasticity creates a fine balance between the DCR's entropy and its sensitivity to input history, which suggests that principles of neural information processing generalize to very different systems than the

brain. This balance led to a substantial increase in the DCR's performance in nonlinear spatiotemporal computations.

We designed in Paper III a homeostatic synaptic plasticity model that assures maximum nonlinear transformation of sensory signals arriving to the neurocontroller of an autonomous agent. By the virtue of having two maximally-nonlinear homeostatic states at the disposal of each neuron, the autonomous agents were able to generate multiple stable behaviors, such as the stance and sway phases for a hexapod locomotion, and forward, turn-right, and turn-left behaviors of a wheel-driven robot.

### 5.1.2 *Which Class of Computations Is Homeostasis Relevant for?*

We argued in this thesis that appreciating the full potential of neural information processing requires studying the more general class of *spatiotemporal computations*. Spatiotemporal computations are the ability of a system to integrate incoming information in space and time, and to transfer the resulting whole in a functionally relevant manner. This class of computations was originally studied by the reservoir computing community [19], and models by Lazar et al. were the first to study homeostatic and spike-timing-dependent synaptic plasticity for spatiotemporal computations within the reservoir computing paradigm [67, 68]. However, principal understanding of the function of homeostatic and other forms of plasticity in regard to spatiotemporal computations remained lacking.

Recall Objective 1 as outlined in Section 1.3:

> Developing a rigorous representation theory of spatiotemporal computations.

A major part of Paper I was dedicated to developing such a theory. We identified the representation of a spatiotemporal pattern (or a function over a spatiotemporal pattern) as the set of responses this pattern elicits in a neural network. This formalism was the means for a geometric interpretation for homeostatic plasticity's computational role: We showed that the increase in entropy due to homeostatic plasticity results in an expansion-in-space of representations as defined above. This expansion is also the signature of *redundancy* as a mechanism for noise-robustness.

As discussed in Section 5.1.1 above, the homeostatic plasticity mechanism in Paper II improves the spatiotemporal computational capabilities of the DCR.

A different approach to spatiotemporal computations was taken in Paper III. We argued that an autonomous agent's behavior comes in response to changes in its sensory input. We showed that the tendency to approach the two homeostatic states of the model neuron within the agent's neurocontroller results, under some conditions, in

the emergence of bistability. Bistability, in turn, allows under varying stimulation for short-term sensory memory. All in all, homeostasis and bistability make it possible for the agent to generate behavior as a reaction to these sensory memories.

### 5.1.3 *What Are Suitable Models for Spatiotemporal Computations?*

Since spatiotemporal computations require integrating information over time, a dynamical system becomes the natural model choice, since temporality becomes native to the model. In addition, since the incoming information to the dynamical system is always changing according to the environment dynamics, the dynamical system becomes *nonautonomous*.

It is reasonable to assume that the brain exploits the high recurrency in its networks [33, 32, 34, 140] to achieve its spatiotemporal computational demands within its changing surrounding. Paper I was the first, to the best of our knowledge, to introduce the theory of nonautonomous dynamical systems to the study of spatiotemporal computations and plasticity. Recall Objective 3 in Section 1.3:

> Arguing for controlled bifurcations as a useful computational tool, which is shaped by the different plasticity mechanisms.

We identified in Paper I the nonautonomous attractor of a model recurrent neural network under varying stimulation. We considered this attractor as a *moving target* of the dynamics which has the following properties: (1) its geometry is shaped by plasticity. (2) At each point in time, this attractor's spatial location is identified by the current stimulus, and thus by the stimulus' representation. We observed that when moving, the nonautonomous attractor changes its topological properties, which corresponds to a *bifurcation* in the network dynamics that is parametrized by the stimulus. The role of plasticity in this picture of spatiotemporal computations is to specify the relevant positions of the attractor's non-coexistent constituents as they change over time.

The DCR model in Paper II is a nonautonomous delay differential equation that is governed by fixed-point asymptotic dynamics for each input value. As such, with the change of input, the attractor of the DCR does not bifurcate, but *morph*, in the sense that it preserves its fixed-point topology, but shift to a different location of the phase space. Unlike the network model in Paper I, the DCR is a continuous-time dynamical system. This entails that our representation theory and the nonautonomous dynamical systems perspective of spatiotemporal computations do not apply. However, given the similar observations in regard to the effects of plasticity in both models, we speculate

that a unifying theory of both discrete and continuous-time systems is in reach.

Similarly in Paper III, a recurrent neural network is used for controlling the autonomous agent as it reacts to the changing stimulation, coming from its sensors. This renders the neurocontroller a nonautonomous dynamical system. We showed that due to the influence of the varying stimulation, and under certain conditions, a neuron within the neurocontroller undergoes a bifurcation from one homeostatic state to the other. These bifurcations result in hysteretic effects which generate the short-term memory necessary for behavior.

## 5.2 NONAUTONOMOUS GRAPHS

In this section, we propose an extension to the theory of nonautonomous dynamical systems that would include the original theory as a special case.

The three nonautonomous dynamical systems presented in this thesis all assume a *unidirectional* coupling between a driven and a driving dynamics. That is, these models are expressed as a triangular system of difference or differential equations [63]. As an example, the following triangular system of difference equations is autonomous:

$$x_{t+1} = f(x_t), \tag{70}$$
$$y_{t+1} = g(x_t, y_t), \tag{71}$$

and its dynamics is that of a *skew product flow* (see Definition C2 in Section 2.7). The interaction of this dynamical system's components and the unidirectional nature of their coupling can be expressed graphically in Figure 31A. The subsystem $x$ is the driving dynamical system, and due to its forcing, the subsystem $y$ has a nonautonomous flow. The subsystem $y$ corresponds to the `kWTA` spiking neural network in Paper I, the delay-coupled reservoir in Paper II, and the neurocontroller in Paper III. These nonautonomous dynamical systems are driven by a stochastic dynamical system in Paper I and II [7], but by the deterministic sensory signals provided by the environment in Paper III.

Consider the following scenario, similar to the model in Paper I. A recurrent neural network with state $x$ undergoes *activity-dependent* plastic changes which act on the synaptic weights $w$ and the firing thresholds $\theta$. During the plasticity phase, the network is also subject to Markovian drive $\delta$. The full dynamical system, consisting of the network state $x$, the plastic elements $w$ and $\theta$, and the drive $\delta$ is expressed by the following system of difference equations:

$$\delta_{t+1} = f_1(\delta_t), \tag{72}$$
$$w_{t+1} = f_2(w_t, x_t), \tag{73}$$
$$\theta_{t+1} = f_3(\theta_t, x_t), \tag{74}$$
$$x_{t+1} = f_4(\delta_t, w_t, \theta_t, x_t). \tag{75}$$

A graphical representation of the interactions between the elements of this dynamical system clearly shows that the coupling is not unidirectional anymore (see Figure 31B).

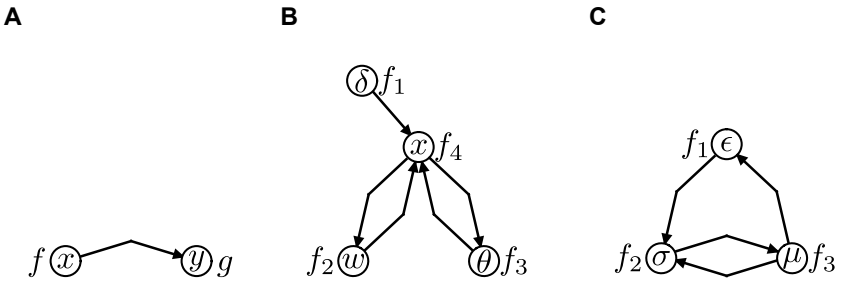


Figure 31: **Graphical representations of nonautonomous graphs.** (A) Skew product flow. (B) Input-driven plastic recurrent neural network. (C) autonomous agent acting in the sensorimotor loop and affecting its environment.

A second situation with bidirectional coupling is the scenario considered in Paper III. Even with no plastic dynamics of its neurocontroller, an autonomous agent's sensory readings $\sigma$ are at the agent's disposal for generating motor behavior $\mu$. On the other hand, the motor behavior may bring the agent to a location where a food source exists, thus changing the sensory input of the agent. Moreover, the environment $\epsilon$ within which the agent lives need not be fixed, either due to its own dynamics, or because the agent's motor actions may influence it directly (by consuming food sources or pushing around objects, for example). This form of interaction may be expressed mathematically by the system of difference equations:

$$\epsilon_{t+1} = f_1(\epsilon_t, \mu_t), \tag{76}$$
$$\sigma_{t+1} = f_2(\epsilon_t, \sigma_t, \mu_t), \tag{77}$$
$$\mu_{t+1} = f_3(\sigma_t, \mu_t). \tag{78}$$

as graphically represented in Figure 31C.

In the above cases, the full system is an autonomous dynamical system with nonautonomous parts. However, the unidirectional skew product formulation cannot account for the parts' dynamics. Therefore, we suggest *nonautonomous graphs* as a more general formalism of nonautonomous dynamical systems that can account for systems

such as those in Figure 31B and 31C. We might be interested, for instance, in describing the stable configurations an environment converges to under the changes afflicted on it by an autonomous agent. Or it may be necessary to describe the changes in the attractor landscape of a network as a result of plasticity. We observed in Paper I, for instance, that during the plasticity phase a `SIP-RN`'s dynamics is first dominated by the input-insensitive attractor. At some point, a bifurcation-like transition occurs where the input-sensitive attractor appears, and the system becomes bistable.

Similar phenomena, as our examples suggest, are not uncommon in biological systems. As such, a future mathematical research program dedicated to the study of nonautonomous graphs becomes crucial. At this point, we can only speculate about the value such a new mathematical concept could provide us for the study of complex biological dynamics. The introduction of this concept as outlined here is still informal. A rigorous theory first requires a formal definition of nonautonomous graphs, similar in structure to that of processes or skew product flows. This needs to be followed by extending notions of nonautonomous stability, attractivity, and bifurcations to this class of systems. Also, as we demonstrated in Paper I, a theory of spatiotemporal representation and computation is necessary. Therefore, this theory must be reformulated within the new framework.

# BIBLIOGRAPHY


[1] Abbott, L. and Regehr, W. G. (2004). Synaptic computation. *Nature*, 431(7010):796–803.

[2] Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity: taming the beast. *Nat. Neurosci.*, 3:1178–1183.

[3] Abraham, W. C. and Philpot, B. (2009). Metaplasticity. *Scholarpedia*, 4(5):4894.

[4] Angeli, D., Ferrell, J. E., and Sontag, E. D. (2004). Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc. Natl. Acad. Sci. U.S.A.*, 101(7):1822–1827.

[5] Appeltant, L., Soriano, M. C., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C. R., and Fischer, I. (2011). Information processing using a single dynamical node as complex system. *Nat. Commun.*, 2:468.

[6] Appeltant, L., Van der Sande, G., Danckaert, J., and Fischer, I. (2014). Constructing optimized binary masks for reservoir computing with delay systems. *Sci. Rep.*, 4.

[7] Arnold, L. (1998). *Random dynamical systems*. Springer Berlin.

[8] Ashby, W. R. (1960). *Design for a Brain: The Origin of Adaptive Behavior (2nd edition)*. Chapman and Hall, London UK.

[9] Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behavior. *Adapt. behav.*, 1(1):91–122.

[10] Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural comput.*, 7(6):1129–1159.

[11] Bernard, C. (1878). *Leçons sur les phénomènes de la vie communs aux animaux et aux végétaux (two volumes)*. Paris, J.-B. Baillère et Fils.

[12] Bienenstock, E. L., Cooper, L. N., and Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.*, 2(1):32–48.

[13] Boyd, S. and Chua, L. O. (1985). Fading memory and the problem of approximating nonlinear operators with volterra series. *IEEE Trans. Circuits Syst.*, 32(11):1150–1161.

[14] Brea, J., Senn, W., and Pfister, J.-P. (2013). Matching recall and storage in sequence learning with spiking neural networks. *J. Neurosci.*, 33(23):9565–9575.

[15] Broome, B. M., Jayaraman, V., and Laurent, G. (2006). Encoding and decoding of overlapping odor sequences. *Neuron*, 51(4):467–482.

[16] Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci.*, 8(3):183–208.

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

[18] Buesing, L. and Maass, W. (2010). A spiking neuron as information bottleneck. *Neural Comput.*, 22(8):1961–1992.

[19] Buonomano, D. V. and Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nat. Rev. Neurosci.*, 10(2):113–125.

[20] Cannon, W. B. (1939). *The wisdom of the body: How the human body reacts to disturbance and danger and maintains the stability essential to life (revised and enlarged edition)*. W. W. Norton & Company.

[21] Carroll, L., Tenniel, S. J., Gardner, M., and Norton, W. (2000). *The annotated Alice: Alice's adventures in Wonderland & Through the looking glass*. W. W. Norton & Company.

[22] Chechik, G., Anderson, M. J., Bar-Yosef, O., Young, E. D., Tishby, N., and Nelken, I. (2006). Reduction of information redundancy in the ascending auditory pathway. *Neuron*, 51(3):359–368.

[23] Christopher deCharms, R., Blake, D. T., and Merzenich, M. M. (1998). Optimizing sound features for cortical neurons. *Science*, 280(5368):1439–1444.

[24] Clopath, C., Büsing, L., Vasilaki, E., and Gerstner, W. (2010). Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nat. Neurosci.*, 13(3):344–352.

[25] Cooper, L. N., Intrator, N., Blais, B. S., and Shouval, H. Z. (2004). *Theory of Cortical Plasticity*. World Scientific Pub. Co. Inc., Singapore.

[26] Cover, T. M. and Thomas, J. A. (2006). *Elements of information theory*. John Wiley & Sons, New York, second edition.

[27] Dasgupta, S., Wörgötter, F., and Manoonpong, P. (2013). Information dynamics based self-adaptive reservoir for delay temporal memory tasks. *Evolving Systems*, 4(4):235–249.

[28] Davis, G. W. and Goodman, C. S. (1998). Synapse-specific control of synaptic efficacy at the terminals of a single neuron. *Nature*, 392(6671):82–86.

[29] Day, T. A. (2005). Defining stress as a prelude to mapping its neurocircuitry: no help from allostasis. *Prog. Neuropsychopharmacol. Biol. Psychiatry*, 29(8):1195–1200.

[30] DeAngelis, G. C., Ohzawa, I., and Freeman, R. (1993). Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. ii. linearity of temporal and spatial summation. *J. Neurophysiol.*, 69(4):1118–1135.

[31] Di Paolo, E. A. (2000). Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. In Meyer, J.-A., Berthoz, A., Floreano, D., Roitblat, H., and Wilson, S., editors, *From Animals to Animats 6. Proceedings of the VI International Conference on Simulation of Adaptive Behavior*, volume 6, pages 440–449. MIT Press.

[32] Douglas, R. J., Koch, C., Mahowald, M., Martin, K., and Suarez, H. H. (1995). Recurrent excitation in neocortical circuits. *Science*, 269(5226):981–985.

[33] Douglas, R. J. and Martin, K. (1991). A functional microcircuit for cat visual cortex. *J. Physiol.*, 440(1):735–769.

[34] Douglas, R. J. and Martin, K. A. (2004). Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*, 27:419–451.

[35] Durrell, L. (2012). *The Alexandria Quartet*. Faber & Faber, London.

[36] Fiser, J., Chiu, C., and Weliky, M. (2004). Small modulation of ongoing cortical dynamics by sensory input during natural vision. *Nature*, 431(7008):573–578.

[37] Gerstner, W. and Kistler, W. M. (2002). *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.

[38] Glass, L. and Mackey, M. (2010). Mackey-Glass equation. *Scholarpedia*, 5(3):6908.

[39] Gollo, L. L., Kinouchi, O., and Copelli, M. (2009). Active dendrites enhance neuronal dynamic range. *PLoS Comput. Biol.*, 5(6):e1000402.

[40] Graupner, M. and Brunel, N. (2012). Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *Proc. Natl. Acad. Sci. U.S.A.*, 109(10):3991–3996.

[41] Gros, C. (2009). Cognitive computation with autonomously active neural networks: an emerging field. *Cognit. Comput.*, 1(1):77–90.

[42] Guckenheimer, J. and Kuznetsov, Y. A. (2007). Cusp bifurcation. *Scholarpedia*, 2(4):1852.

[43] Guo, S. and Wu, J. (2013). *Bifurcation theory of functional differential equations*. Springer New York.

[44] Harter, D. and Kozma, R. (2005). Chaotic neurodynamics for autonomous agents. *IEEE Trans. Neural Netw.*, 16(3):565–579.

[45] Harvey, I., Di Paolo, E., Wood, R., Quinn, M., Tuci, E., and Iridia, E. T. (2005). Evolutionary robotics: a new scientific tool for studying cognition. *Artif. Life*, 11(1-2):79–98.

[46] Häusler, S. and Maass, W. (2007). A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. *Cereb. Cortex*, 17(1):149–162.

[47] Hebb, D. O. (1949). *The organization of behavior*. John Wiley & Sons, New York.

[48] Heuser, H. (2003). *Lehrbuch der analysis*. Number pt. 1 in Lehrbuch der analysis. Teubner.

[49] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117(4):500.

[50] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

[51] Hoinville, T. and Hénaff, P. (2004). Comparative study of two homeostatic mechanisms in evolved neural controllers for legged locomotion. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, pages 2624–2629.

[52] Hoinville, T., Siles, C. T., and Hénaff, P. (2011). Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers. *Adapt. Behav.*, 19(3):187–207.

[53] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.*, 79(8):2554–2558.

[54] Hülse, M. and Pasemann, F. (2002). Dynamical neural schmitt trigger for robot control. In Dorronsoro, J. R., editor, *Artificial Neural Networks–ICANN 2002*, pages 783–788. Springer.

[55] Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420.

[56] Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Comput.*, 18(2):245–282.

[57] Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. Techn. rep. GMD 148, Bremen: German National Research Center for Information Technology.

[58] Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80.

[59] Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Netw.*, 20(3):335–352.

[60] Karmarkar, U. R. and Buonomano, D. V. (2007). Timing in the absence of clocks: encoding time in neural network states. *Neuron*, 53(3):427–438.

[61] Kenet, T., Bibitchkov, D., Tsodyks, M., Grinvald, A., and Arieli, A. (2003). Spontaneously emerging cortical representations of visual attributes. *Nature*, 425(6961):954–956.

[62] Kloeden, P. E., Pötzsche, C., and Rasmussen, M. (2013). Discrete-time nonautonomous dynamical systems. In Johnson, R. and Pera, M., editors, *Stability and Bifurcation Theory for Non-Autonomous Differential Equations*, pages 35–102. Springer.

[63] Kloeden, P. E. and Rasmussen, M. (2011). *Nonautonomous dynamical systems*. AMS Bookstore.

[64] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 69(6):066138.

[65] Kubisch, M., Werner, B., and Hild, M. (2011). Using co-existing attractors of a sensorimotor loop for the motion control of a humanoid robot. In *Proceedings of the International Conference on Neural Computation Theory and Applications*, pages 385–390.

[66] Larger, L., Soriano, M., Brunner, D., Appeltant, L., Gutiérrez, J. M., Pesquera, L., Mirasso, C. R., and Fischer, I. (2012). Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express*, 20(3):3241–3249.

[67] Lazar, A., Pipa, G., and Triesch, J. (2007). Fading memory and time series prediction in recurrent networks with different forms of plasticity. *Neural Netw.*, 20(3):312–322.

[68] Lazar, A., Pipa, G., and Triesch, J. (2009). SORN: a self-organizing recurrent neural network. *Front. Comput. Neurosci.*, 3(23).

[69] Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

[70] Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–2560.

[71] Manjunath, G. and Jaeger, H. (2013). Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural Comput.*, 25(3):671–696.

[72] Manoonpong, P., Geng, T., Kulvicius, T., Porr, B., and Wörgötter, F. (2007). Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput. Biol.*, 3(7):e134.

[73] Manoonpong, P., Pasemann, F., and Roth, H. (2006). A modular neurocontroller for a sensor-driven reactive behavior of biologically inspired walking machines. *Int. J. Computing*, 5(3):75–86.

[74] Marković, D. and Gros, C. (2012). Intrinsic adaptation in autonomous recurrent neural networks. *Neural Comput.*, 24(2):523–540.

[75] Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215.

[76] Mauk, M. D. and Buonomano, D. V. (2004). The neural basis of temporal processing. *Annu. Rev. Neurosci.*, 27:307–340.

[77] Miller, B. L. and Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193–212.

[78] Miller, K. D. (1996). Synaptic economics: competition and cooperation in synaptic plasticity. *Neuron*, 17(3):371–374.

[79] Mongillo, G., Barak, O., and Tsodyks, M. (2008). Synaptic theory of working memory. *Science*, 319(5869):1543–1546.

[80] Nadeau, J. H., Burrage, L. C., Restivo, J., Pao, Y.-H., Churchill, G., and Hoit, B. D. (2003). Pleiotropy, homeostasis, and functional networks based on assays of cardiovascular traits in genetically randomized populations. *Genome Res.*, 13(9):2082–2091.

[81] Narayanan, N. S., Kimchi, E. Y., and Laubach, M. (2005). Redundancy and synergy of neuronal ensembles in motor cortex. *J. neuroscie.*, 25(17):4207–4216.

[82] Naudé, J., Cessac, B., Berry, H., and Delord, B. (2013). Effects of cellular homeostatic intrinsic plasticity on dynamical and computational properties of biological recurrent neural networks. *J. Neurosci.*, 33(38):15032–15043.

[83] Negrello, M. and Pasemann, F. (2008). Attractor landscapes and active tracking: The neurodynamics of embodied action. *Adapt. Behav.*, 16(2-3):196–216.

[84] Nikolić, D., Häusler, S., Singer, W., and Maass, W. (2009). Distributed fading memory for stimulus properties in the primary visual cortex. *PLoS Biol.*, 7(12):e1000260.

[85] Nolfi, S. and Floreano, D. (1999). Learning and evolution. *Auton. Robots*, 7(1):89–113.

[86] Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press Cambridge.

[87] Oja, E. (1982). Simplified neuron model as a principal component analyzer. *J. Math. Biol.*, 15(3):267–273.

[88] Ovid and Raeburn, D. (2004). *Metamorphoses*. Penguin Classics, Harmondsworth.

[89] Panzeri, S., Schultz, S. R., Treves, A., and Rolls, E. T. (1999). Correlations and the encoding of information in the nervous system. *Proc. R. Soc. Lond. B. Biol. Sci.*, 266(1423):1001–1012.

[90] Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., and Massar, S. (2012). Optoelectronic reservoir computing. *Sci. Rep.*, 2.

[91] Pascanu, R. and Jaeger, H. (2011). A neurodynamical model for working memory. *Neural Netw.*, 24(2):199–207.

[92] Pasemann, F. (1993). Dynamics of a single model neuron. *Int. J. Bifurcat. Chaos*, 3(02):271–278.

[93] Pasemann, F. (1995). Characterization of periodic attractors in neural ring networks. *Neural Netw.*, 8(3):421–429.

[94] Pasemann, F. (2013). Self-regulating neurons in the sensorimotor loop. In Rojas, I., Joya, G., and Gabestany, J., editors, *Advances in Computational Intelligence*, volume 7902 of *Lecture Notes in Computer Science*, pages 481–491. Springer Berlin Heidelberg.

[95] Pfister, J.-P. and Tass, P. A. (2010). STDP in oscillatory recurrent networks: theoretical conditions for desynchronization and applications to deep brain stimulation. *Front. Comput. Neurosci.*, 4.

[96] Pitonakova, L. (2013). Ultrastable neuroendocrine robot controller. *Adapt. Behav.*, 21(1):47–63.

[97] Puchalla, J. L., Schneidman, E., Harris, R. A., and Berry, M. J. (2005). Redundancy in the population code of the retina. *Neuron*, 46(3):493–504.

[98] Rabinovich, M., Volkovskii, A., Lecanda, P., Huerta, R., Abarbanel, H., and Laurent, G. (2001). Dynamical encoding by networks of competing neuron groups: winnerless competition. *Phys. Rev. Lett.*, 87(6):068102.

[99] Rabinovich, M. I., Huerta, R., Varona, P., and Afraimovich, V. S. (2008). Transient cognitive dynamics, metastability, and decision making. *PLoS Comput. Biol.*, 4(5):e1000072.

[100] Remme, M. W. and Wadman, W. J. (2012). Homeostatic scaling of excitability in recurrent neural networks. *PLoS Comput. Biol.*, 8(5):e1002494.

[101] Rempis, C., Thomas, V., Bachmann, F., and Pasemann, F. (2010). NERD - Neurodynamics and Evolutionary Robotics Development Kit. In Ando, N., Balakirsky, S., Hemker, T., Reggiani, M., and Stryk, O., editors, *Simulation, Modeling, and Programming for Autonomous Robots*, Lecture Notes in Computer Science, pages 121–132. Springer Berlin Heidelberg.

[102] Rempis, C. W. (2012). *Evolving Complex Neuro-Controllers with Interactively Constrained Neuro-Evolution*. PhD thesis, Osnabrück University.

[103] Rempis, C. W., Toutounji, H., and Pasemann, F. (2013). Evaluating neuromodulator-controlled stochastic plasticity for learning recurrent neural control networks. In *Proceedings of the 5th International Joint Conference on Computational Intelligence*, pages 489–496.

[104] Renart, A., Song, P., and Wang, X.-J. (2003). Robust spatial working memory through homeostatic synaptic scaling in heterogeneous cortical networks. *Neuron*, 38(3):473–485.

[105] Richet, C. (1900). *Dictionnaire de physiologie*. Paris, F. Alcan.

[106] Rolls, E. T. and Deco, G. (2010). *The noisy brain: stochastic dynamics as a principle of brain function*, volume 28. Oxford university press New York.

[107] Rumelhart, D. E. and McClelland, J. L. (1986). *Parallel distributed processing: explorations in the microstructure of cognition, 2 volumes*. MIT Press, Cambridge, Ma.

[108] Rumsey, C. C., Abbott, L. F., et al. (2006). Synaptic democracy in active dendrites. *J. Neurophysiol.*, 96(5):2307–2318.

[109] Santos, B. A., Husbands, P., and Froese, T. (2010). Accommodating homeostatically stable dynamical regimes to cope with different environmental conditions. In Fellermann, H., Dörr, M., Hanczy, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., and Støy, K., editors, *Proc. Artificial Life XII*, pages 395–402. MIT Press.

[110] Savin, C., Joshi, P., and Triesch, J. (2010). Independent component analysis in spiking neurons. *PLoS Comput. Biol.*, 6(4):e1000757.

[111] Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H. (2013a). Walknet, a bio-inspired controller for hexapod walking. *Biol. Cybern.*, 107(4):397–419.

[112] Schilling, M., Paskarbeit, J., Hoinville, T., Hüffmeier, A., Schneider, A., Schmitz, J., and Cruse, H. (2013b). A hexapod walker using a heterarchical architecture for action selection. *Front. Comput. Neurosci.*, 7.

[113] Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J. J., and Stroobandt, D. (2008). Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7):1159–1171.

[114] Schumacher, J., Toutounji, H., and Pipa, G. (2013). An analytical approach to single node delay-coupled reservoir computing. In Mladenov, P., Koprinkova-Hristova, V., Palm, G., Villa, A. E. P., Appollini, B., and Kasabov, N., editors, *Artificial Neural Networks and Machine Learning–ICANN 2013*, volume 8131 of *Lecture Notes in Computer Science*, pages 26–33. Springer Berlin Heidelberg.

[115] Shadlen, M. N. and Newsome, W. T. (1994). Noise, neural codes and cortical organization. *Curr. Opin. Neurobiol.*, 4(4):569–579.

[116] Shim, Y. and Husbands, P. (2012). Chaotic exploration and learning of locomotion behaviors. *Neural Comput.*, 24(8):2185–2222.

[117] Skarda, C. A. and Freeman, W. J. (1987). How brains make chaos in order to make sense of the world. *Behav. Brain Sci.*, 10(2):161–195.

[118] Smith, T., Husbands, P., Philippides, A., and O'Shea, M. (2002). Neuronal plasticity and temporal adaptivity: GasNet robot control networks. *Adapt. Behav.*, 10(3-4):161–183.

[119] Somers, D. C., Nelson, S. B., and Sur, M. (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *J. Neurosci.*, 15(8):5448–5465.

[120] Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nat Neurosci.*, 3(9):919–926.

[121] Soto-Treviño, C., Thoroughman, K. A., Marder, E., and Abbott, L. (2001). Activity-dependent modification of inhibitory synapses in models of rhythmic neural networks. *Nat. Neurosci.*, 4(3):297–303.

[122] Stein, R. B., Gossen, E. R., and Jones, K. E. (2005). Neuronal variability: noise or part of the signal? *Nat. Rev. Neurosci.*, 6(5):389–397.

[123] Stemmler, M. and Koch, C. (1999). How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nat. Neurosci.*, 2(6):521–527.

[124] Sterling, P. and Eyer, J. (1988). Allostasis: a new paradigm to explain arousal pathology. In Fisher, S. and Reason, J., editors, *Handbook of life stress, cognition and health*, pages 629–649. John Wiley & Sons, New York.

[125] Strogatz, S. H. (2001). *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering (studies in nonlinearity)*. Westview Press.

[126] Tetzlaff, C., Kolodziejski, C., Timme, M., and Wörgötter, F. (2011). Synaptic scaling in combination with many generic plasticity mechanisms stabilizes circuit connectivity. *Front. Comput. Neurosci.*, 5.

[127] Timme, M., Wolf, F., and Geisel, T. (2002). Prevalence of unstable attractors in networks of pulse-coupled oscillators. *Phys. Rev. Lett.*, 89(15):154105.

[128] Tkačik, G., Prentice, J. S., Balasubramanian, V., and Schneidman, E. (2010). Optimal population coding by noisy spiking neurons. *Proc. Natl. Acad. Sci. U.S.A.*, 107(32):14419–14424.

[129] Toutounji, H. and Pasemann, F. (2014). Behavior control in the sensorimotor loop with short-term synaptic dynamics induced by self-regulating neurons. *Front. Neurorobot.*, 8:19.

[130] Toutounji, H. and Pipa, G. (2014). Spatiotemporal computations of an excitable and plastic brain: neuronal plasticity leads to noise-robust and noise-constructive computations. *PLoS Comput. Biol.*, 10(3):e1003512.

[131] Toyoizumi, T., Pfister, J.-P., Aihara, K., and Gerstner, W. (2005). Generalized Bienenstock–Cooper–Munro rule for spiking neurons that maximizes information transmission. *Proc. Natl. Acad. Sci. U.S.A.*, 102(14):5239–5244.

[132] Triesch, J. (2007). Synergies between intrinsic and synaptic plasticity mechanisms. *Neural Comput.*, 19(4):885–909.

[133] Turrigiano, G. G. and Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.*, 5(2):97–107.

[134] Vargas, P. A., Moioli, R. C., Von Zuben, F. J., and Husbands, P. (2009). Homeostasis and evolution together dealing with novelties and managing disruptions. *Int. J. Intelligent Computing and Cybernetics*, 2(3):435–454.

[135] Verstraeten, D., Schrauwen, B., and Stroobandt, D. (2006). Reservoir-based techniques for speech recognition. In *International Joint Conference on Neural Networks*, pages 1050–1053. IEEE.

[136] von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14(2):85–100.

[137] von Twickel, A., Büschges, A., and Pasemann, F. (2011). Deriving neural network controllers from neuro-biological data: implementation of a single-leg stick insect controller. *Biol. Cybern.*, 104(1-2):95–119.

[138] von Twickel, A., Hild, M., Siedel, T., Patel, V., and Pasemann, F. (2012). Neural control of a modular multi-legged walking machine: Simulation and hardware. *Rob. Auton. Syst.*, 60(2):227–241.

[139] von Twickel, A. and Pasemann, F. (2007). Reflex-oscillations in evolved single leg neurocontrollers for walking machines. *Nat. Comput.*, 6(3):311–337.

[140] Weiler, N., Wood, L., Yu, J., Solla, S. A., and Shepherd, G. M. (2008). Top-down laminar organization of the excitatory network in motor cortex. *Nat. Neurosci.*, 11(3):360–366.

[141] Williams, H. and Noble, J. (2007). Homeostatic plasticity improves signal propagation in continuous-time recurrent neural networks. *Biosystems*, 87(2):252–259.

[142] Yamazaki, T. and Tanaka, S. (2007). The cerebellum as a liquid state machine. *Neural Netw.*, 20(3):290–297.

[143] Zahedi, K. and Pasemann, F. (2007). Adaptive behavior control with self-regulating neurons. In Lungarella, M., Iida, F., Bongard, J., and Pfeifer, R., editors, *50 Years of Artificial Intelligence*, volume 4850 of *Lecture Notes in Computer Science*, pages 196–205. Springer Berlin Heidelberg.

[144] Zenke, F., Hennequin, G., and Gerstner, W. (2013). Synaptic plasticity in neural networks needs homeostasis with a fast rate detector. *PLoS Comput. Biol.*, 9(11):e1003330.

[145] Zhang, W. and Linden, D. J. (2003). The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nat. Rev. Neurosci.*, 4(11):885–900.

[146] Zheng, P., Dimitrakakis, C., and Triesch, J. (2013). Network self-organization explains the statistics and dynamics of synaptic connection strengths in cortex. *PLoS Comput. Biol.*, 9(1):e1002848.

[147] Zucker, R. S. and Regehr, W. G. (2002). Short-term synaptic plasticity. *Annu. Rev. Physiol.*, 64(1):355–405.

# DECLARATION

I hereby confirm that I wrote this thesis independently and that I have not made use of resources other than those indicated. I guarantee that I significantly contributed to all materials used in this thesis. Furthermore, this thesis was neither published in Germany nor abroad, except the parts indicated above, and has not been used to fulfill any other examination requirements.

*Osnabrück, June 2014*

Hazem Toutounji, 01.07.2014