Equalization and Decoding
A Continuous-Time Dynamical Approach

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS
(DR.-ING.)

der Fakultät für Ingenieurwissenschaften
und Informatik der Universität Ulm

von

MOHAMAD MOSTAFA
AUS ALEPPO

Betreuer:	Prof. Dr.-Ing. Jürgen Lindner
Gutachter:	Prof. Dr. Ljupco Kocarev
Amtierende Dekanin:	Prof. Dr. Tina Seufert

Ulm, 04. April 2014

Acknowledgments

This thesis is the outcome of my research activity during 2008-2013 at the university of Ulm. I would like here to thank numerous persons for their substantial contribution.

First of all, I would like to express my gratitude to my supervisor Prof. Dr.-Ing. Jürgen Lindner for giving me the opportunity to join his research group and for his guidance during the whole time. I am thankful for his sustained support to pursue my research in this field. I am also appreciative of Prof. Dr. Ljupco Kocarev for agreeing to act as a co-supervisor and for his hospitality at the Macedonian Academy of Sciences and Arts.

I feel deeply grateful to Dr. Werner Teich. Without his critical input, continual feedback, significant contribution and constructive discussion this thesis would not be what it is.

My special thank is dedicated to Giuseppe Oliveri for his cooperative collaboration and effort on the circuit design side.

I am also thankful for all colleagues, who accompanied the formation of this thesis in its different phases. I would like to mention especially my office-mate Alexander Linduska, Zoran Utkovski and Matthias Wetz. I owe Eva Peiker and Thanawat Thiasiriphet a great thank for their proof-reading.

Ji bo dêk û bavê min

Ji bo Edhem

Contents

Introduction

ALTHOUGH in biological neural systems events take place with a much longer time scale compared to digital computers, the *energetic efficiency* (joule per operation per second) of the brain is much better than the one of the best computers nowadays. This is because the *computing concept* of the brain is entirely different from that of conventional digital computers [20]. The key concept is the *"massive parallel nonlinear collective processing of large number of signals that are continuous in time and amplitude"* [87]. So, neither sampling nor quantization take place.

The brain has the capability to organize the neurons (experience-adapted connections) to perform specific tasks faster than the fastest digital computers in existence today [20]. Pattern recognition and image processing are well suited examples. *"The visual system of a single human being does more image processing than the entire world's supply of supercomputers"* [54].

Artificial neural networks (in the following only neural networks) imitate the computing concept of the brain in order to solve different tasks faced in many scientific disciplines as efficient as possible. A general definition of neural networks as formulated by Haykin [20] is:

"A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

- *Knowledge is acquired by the network through a learning process.*
- *Interneuron connections "synaptic weights" are used for storing the knowledge."*

It is worth mentioning that it is the organization and interconnection of neurons, in both biological and artificial neural networks, which enable powerful collective computation , rather than the functionality of a single neuron. For instance, feedback is observed in the nervous system of almost every animal [20]. Neural networks with feedback represent a special class known as *"recurrent neural networks"*. Properties and capabilities of neural networks are changed essentially by the feedback. Recurrent neural networks stand out due to their rich dynamics.

Very-large-scale integration (VLSI) technology has been shown to fit well as implementation medium for neural networks [54]. Whether the analog VLSI technology or the digital one is favorable has a rich history of research. In [87] it has been shown that a wide range of functions can be realized by a single transistor. This includes, e.g.,

- Generation of square, square root, exponential and logarithmic functions.
- Voltage-controlled current source.
- Voltage-controlled conductance, linear in a limited range.
- Analog multiplication of voltages.
- Short and long term analog storage.
- Additional functions are offered by basic combinations of few transistors.

The conclusion of [87] is that any desired local operation can be realized by a very limited number of transistors, which enables a very dense implementation of continuous-time and continuous-amplitude processing cells for a collective analog processing.

Compared with digital technology, certain computations are less area and/or power consuming when performed in analog. Multiplication is a well known example. However, analog computation is less precise because it is susceptible to noise, temperature, power supply and hardware fluctuations [54]. In addition, analog circuit design and test are more difficult and less flexible than digital ones. In conclusion, analog VLSI signal processing is especially applicable for those tasks, where the massive parallel nonlinear collective processing compensates the less precise local computation [87].

The advantages of the digital implementation over the analog one are widely known. This includes flexibility, ease of design and test and the arbitrary high achievable precision.

At the end, the definite choice of analog or digital technology can not be decided unless the particular algorithm to be implemented is known [20].

However, increasing energy prices and the fast growth of power-hungry high data rate battery-operated mobile services make the energy consumption for nowadays technology an economical as well as environmental challenge. This is especially remarkable in the IT-field. *"Analog signal processing systems can be built that share the robustness of digital systems but outperform digital systems by several orders of magnitude in terms of speed and/or power consumption"* [49].

In the light of this discussion, we focus in this thesis on the investigation and comparison of continuous-time and discrete-time recurrent neural networks for equalization and channel decoding tasks. In this case, the recurrent neural networks represent suboptimum solutions with good performance-complexity trade-off. The application of discrete-time recurrent neural networks in this field is well investigated, in contrast to the continuous-time ones, far away from "proof-of-concept". We take in this thesis a signal processing point of view rather than breaking the problem down to the transistor level.

Equalization is needed at the receiver because of the multipath propagation of the physical channel between transmitter and receiver. The receive signal in this case is expected to suffer from intersymbol interference. In case of multiuser, multisubchannel, multiantenna transmission systems or combination thereof, the receive signal suffers from additional interuser/intersubchannel interference. To cope with this interference, vector equalization has to be applied at the receiver. To improve the bandwidth/power efficiency further channel coding is applied.

Due to the high complexity of the optimum equalization and channel decoding, suboptimum schemes are applied. In many cases, these are soft-valued iterative schemes because of their good complexity-performance characteristic. As mentioned before, recurrent neural networks are such a suboptimum scheme.

Neural networks, including the recurrent ones, can be trained to act as a vector equalizer and to perform channel decoding. However, training neural networks is always associated with computational complexity and is time consuming as well. Large training sets are required, which represent a challenge in practical applications. In addition, the capability of the trained neural network to generalize its performance beyond the "trained set" needs also to be considered.

In our case, *there is no need for a training phase (learning algorithms)*. Instead of that, we make use of the dynamics of these networks (dynamical solver). This idea has been first exploited by Hopfield in his pioneering work [29], where information has been stored in a dynamically stable recurrent neural network. Therefore, no training algorithms are considered in this thesis. Instead of that an extensive stability analysis is done for both, discrete-time and continuous-time recurrent neural networks.

We do believe that continuous-time equalization and decoding is a promising approach for high data rate transmissions, as an example for optical communications, among others, where data is transmitted at rates up to 1000 Gbyte per second. Analog-to-digital converters at such high rates have a high power consumption, the digital signal processing as well. The continuous-time approach does not require analog-to-digital converters and it is widely accepted that it leads to many orders of magnitude performance improvement in terms of speed and/or power consumption.

This thesis is structured as follows.

Chapter 1 introduces the transmission chain and the receiver structure. Especially, the definition of the discrete-time channel matrix on symbol basis is essential for the following chapters. Optimum and some suboptimum solutions for the equalization and channel decoding are also presented.

We begin Chapter 2 with a short introduction to dynamical systems and their important analysis methods. This is followed by the definition of *twin dynamical systems*, which is essential to relate discrete-time recurrent neural networks to continuous-time ones. Different classes of interest of recurrent neural networks are presented as well and an extensive stability analysis is performed. In this chapter, we generalize many known stability results from the real-valued case to the complex-valued one. In addition, new stability results are derived. One of the most important contributions of this chapter is the stability proof with time-varying activation functions. To a lesser extent, high order recurrent neural networks are presented and investigated.

Chapter 3 is dedicated to the application of recurrent neural networks for vector equalization. Many stability conditions from Chapter 2 are interpreted in the light of this application. One of the major contributions of Chapter 3 is an approximation of the optimum estimation function, which enables a numerically stable evaluation.

We begin Chapter 4 with the dynamical representation of two iterative decoding algorithms, namely belief propagation and iterative threshold decoding, in both cases, discrete-time and continuous-time. These dynamical models are related to the high-order recurrent neural networks and are investigated from the stability point of view. For the linear case, closed-form expressions could be found.

In Chapter 5, the connection between recurrent neural networks and high order ones is considered to perform continuous-time joint equalization and decoding. The importance of this approach lies in the fact that joint equalization and decoding leads to an additional performance improvement but at the same time to higher complexity and/or power consumption. This could be compensated by the continuous-time approach. The intention in Chapter 5 is to show, based on the available knowledge in the discrete-time case, how such a continuous-time joint equalization and decoding could look like, rather than introducing its theoretical basis.

Finally, a summary and conclusions are given. Simulation results for equalization, decoding and joint equalization and decoding are presented in Chapter 3, Chapter 4 and Chapter 5, respectively.

Appendix A contains further supporting mathematical derivations. A list of symbols, functions and abbreviations can be found in Appendix B.

The main novelties presented in this thesis are contained in Chapter 2, Chapter 3, Chapter 4 and Chapter 5. Parts of this thesis were published in [57]–[66]. This work has been essentially influenced by [10],[20] and the references therein.

Notation

Hereafter, bold small letters and bold capital letters designate vectors and matrices, respectively. The only exception is the vector of L-values $\boldsymbol{L}$: It is a bold capital letter but it represents a vector. All nonbold symbols are scalars. In addition, $(\cdot)^T$, $(\cdot)^*$ and $(\cdot)^H$ represent the transpose, conjugate and conjugate transpose. A matrix $\boldsymbol{A} > 0$ ($\boldsymbol{A} \geq 0$) means that $\boldsymbol{A}$ is positive definite (positive semidefinite). We assume column vectors. $\mathrm{diag}\{\cdot\}$ with a vector as argument returns a square diagonal matrix with the elements of the vector on the main diagonal. $\mathrm{diag}\{\cdot\}$ with a square matrix as argument returns a column vector of the main diagonal elements of the matrix.

Chapter 1

Vector-Valued Transmission Model

In this chapter we introduce the block diagram of the vector-valued transmission model and we briefly explain the functionality of its individual parts with special focus on the detection part. The detector (DET) consists of two parts: equalization and channel decoding (for simplicity referred to a decoding in the following).

One of the most important items in this chapter is the introduction of a discrete-time vector-valued transmission model on symbol basis and the discrete-time channel matrix on symbol basis $\boldsymbol{R}$ (for simplicity channel matrix). This model suits multiantenna, multicarrier, multiusers systems and combinations thereof.

The complexity of the equalization depends only on the channel matrix $\boldsymbol{R}$. So, different systems which lead to the same channel matrix $\boldsymbol{R}$ have the same equalization complexity and performance. If a receiver with low complexity is desired, the system must be designed in a way that the channel matrix approaches the identity matrix, i.e. $\boldsymbol{R} = \boldsymbol{I}$.

We start this chapter with the derivation of the discrete-time vector-valued transmission model on symbol basis for the time-invariant case (for the time-variant case see [10]). We show also how the channel matrix $\boldsymbol{R}$ looks like for multiantenna, orthogonal frequency-division multiplexing (OFDM), and MIMO-OFDM (multiple-input multiple-output OFDM). An important step in this chapter is also the reformulation of the discrete-time vector valued transmission model for block transmission.

After that, the maximum likelihood (ML) detection, equalization and decoding rules are presented. We consider also suboptimum detection methods. This includes separate and joint equalization and decoding (further discussed in Chapter 5). In addition, few suboptimum equalization schemes (further investigated in Chapter 3), an introduction to iterative decoding (further analyzed in Chapter 4) are considered.

1.1 Transmission Model

This section is based mainly on [45],[46]. We limit ourself to linear modulation schemes. For the ease of notation, we do not distinguish between band pass and low pass signals and systems. Fig. 1.1 shows the continuous-time vector-valued transmission model in the time-invariant case.

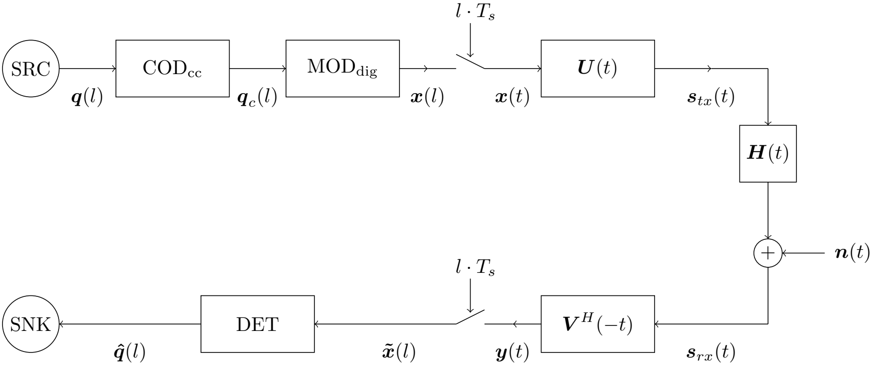


Figure 1.1: Continuous-time vector-valued transmission model. Time-invariant case.

The digital source (SRC) generates a sequence of vectors

$$\cdots,\underbrace{\Big[q_1(l),q_2(l),\cdots,q_k(l)\Big]^T}_{\boldsymbol{q}(l)},\underbrace{\Big[q_1(l+1),q_2(l+1),\cdots,q_k(l+1)\Big]^T}_{\boldsymbol{q}(l+1)},\cdots$$

This can be understood as if the digital source generates a matrix of bits, where the horizontal direction represents the discrete-time and the vertical direction represents the components. The components of the vectors can belong to different users, antennas, subcarriers or combinations thereof depending on the transmission scheme. We show later examples of these cases.

COD$_{\text{cc}}$ performs channel coding by adding redundancy to the bits generated by the SRC. This takes place in time or component direction or in both of them. At the output of COD$_{\text{cc}}$ we obtain a sequence of vectors

$$\cdots,\underbrace{\Big[q_{c,1}(l),q_{c,2}(l),\cdots,q_{c,n}(l)\Big]^T}_{\boldsymbol{q}_c(l)},\underbrace{\Big[q_{c,1}(l+1),q_{c,2}(l+1),\cdots,q_{c,n}(l+1)\Big]^T}_{\boldsymbol{q}_c(l+1)},\cdots$$

where $n > k$ and $r = k/n$ is the code rate.

MOD$_{\text{dig}}$ maps the encoded bits to symbols, which belong to the symbol alphabet $\Psi = \{\psi_1,\psi_2,\cdots,\psi_M\} : \log_2 M \in \mathbb{N}$. This includes Gray coding as well. At the output of MOD$_{\text{dig}}$ we obtain a sequence of vectors of transmit symbols $\boldsymbol{x}(l)$ of length M_{tx}.

$$\cdots,\underbrace{\Big[x_1(l),x_2(l),\cdots,x_{M_{tx}}(l)\Big]^T}_{\boldsymbol{x}(l)},\underbrace{\Big[x_1(l+1),x_2(l+1),\cdots,x_{M_{tx}}(l+1)\Big]^T}_{\boldsymbol{x}(l+1)},\cdots$$

The elements of the vectors of transmit symbols are converted to signals by Dirac sampling. Interleaving can be done between COD$_{\text{cc}}$ and MOD$_{\text{dig}}$ which is not considered here.

$\boldsymbol{U}(t) = \operatorname{diag}\{u_1(t),u_2(t),\cdots,u_{M_{tx}}(t)\}$ is a diagonal matrix of basic waveforms. The diagonal elements are Nyquist impulses or spreading functions. They define the transmission/multiplexing scheme.

$\boldsymbol{s}_{tx}(t)$ is the vector of transmit signals of length M_{tx} given by:

$$\boldsymbol{s}_{tx}(t) = \sum_l \boldsymbol{U}(t - l\cdot T_s)\cdot\boldsymbol{x}(l) \tag{1.1}$$

where T_s is the symbol duration.

$\boldsymbol{H}(t)$ is the physical channel matrix of size $M_{rx} \times M_{tx}$.
$\boldsymbol{s}_{rx}(t)$ is the vector of receive signals of length M_{rx} given by:

$$\boldsymbol{s}_{rx}(t) = \boldsymbol{H}(t) * \boldsymbol{s}_{tx}(t) + \boldsymbol{n}(t) \tag{1.2}$$

where $\boldsymbol{n}(t)$ is a sample function of an additive white Gaussian noise (AWGN) vector process. $\boldsymbol{n}(t)$ is a vector of length M_{rx}.

We define:

$$\begin{aligned} \boldsymbol{V}(t) &= \boldsymbol{H}(t) * \boldsymbol{U}(t) \\ \boldsymbol{R}(l) &= \boldsymbol{V}^H(-t) * \boldsymbol{V}(t)|_{t=l \cdot T_s} \end{aligned} \tag{1.3}$$

where $\boldsymbol{V}^H(-t)$ is the channel matched filter (CMF). $\boldsymbol{R}(l)$ is the channel matrix. In this case it can be shown that:

$$\begin{aligned} \tilde{\boldsymbol{x}}(l) &= \boldsymbol{R}(l) * \boldsymbol{x}(l) + \tilde{\boldsymbol{n}}(l) \\ \boldsymbol{\Phi}_{\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}}(l) &= \frac{N_0}{2} \cdot \boldsymbol{R}(l). \end{aligned} \tag{1.4}$$

$\tilde{\boldsymbol{x}}(l)$ is the vector of receive symbols at the l-th time instant which contains sufficient statistics for a ML-detection [46]. $N_0/2$ is the power spectral density of each component of the double side AWGN vector process $\boldsymbol{n}(t)$ in the band pass domain. $\tilde{\boldsymbol{n}}(l)$ is a sequence of colored noise vectors at the output of the channel matched filter with covariance matrix $\boldsymbol{\Phi}_{\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}}$. The discrete-time vector valued transmission model based on Eq. (1.4) is shown in Fig. 1.2.

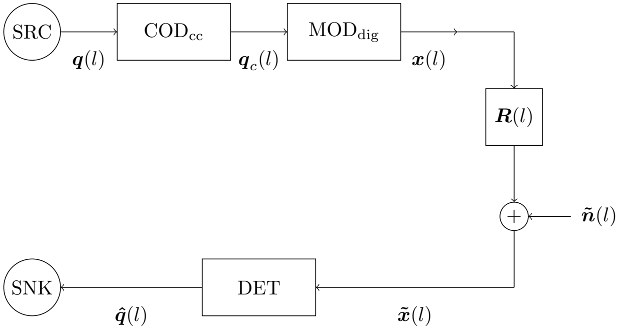


Figure 1.2: Discrete-time vector-valued transmission model on symbol basis.

Remark 1.1 Vector-matrix convolution is performed as vector-matrix multiplication. However, the multiplication sign is replaced by the convolution one. This is valid for the matrix-matrix convolution as well.

DET stands for detector. It performs the equalization and decoding. Equalization is required to cope with the effects of the multipath propagation channel between transmitter and receiver. Decoding serves to correct the errors by exploiting the redundancy which has been added at the transmitter by $\mathrm{COD}_{\mathrm{cc}}$. More about the DET can be found in Sec. 1.2.

If the physical channel is ideal and the diagonal entries of $\boldsymbol{U}(t)$ fulfill the general first Nyquist criterion, it can be easily shown that

$$\boldsymbol{R}(l) = \delta(l) \cdot \boldsymbol{I}. \tag{1.5}$$

For the j-th component, the first relation of Eq. (1.4) can be rewritten as follows:

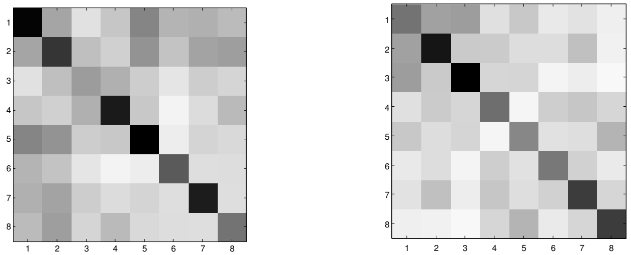
$$\begin{aligned} \tilde{x}_j(l) = r_{jj}(0) \cdot x_j(l) + \sum_{l' \neq l} r_{jj}(l - l') \cdot x_j(l') + \tilde{n}_j(l) \\ + \sum_{j' \neq j} r_{jj'}(0) \cdot x_{j'}(l) + \sum_{l' \neq l} \sum_{j' \neq j} r_{jj'}(l - l') \cdot x_{j'}(l'). \end{aligned} \tag{1.6}$$

We notice that the j-th component of the vector of receive symbols at the l-th time instant can be decomposed into:

- The desired part: $r_{jj}(0) \cdot x_j(l)$.
- Intersymbol interference: $\sum_{l' \neq l} r_{jj}(l - l') \cdot x_j(l')$.
- Interference from other components at the same time instant: $\sum_{j' \neq j} r_{jj'}(0) \cdot x_{j'}(l)$.
- Interference from other components at other time instants: $\sum_{l' \neq l} \sum_{j' \neq j} r_{jj'}(l - l') \cdot x_{j'}(l')$.
- Noise: $\tilde{n}_j(l)$.

$r_{jj'}(l)$ is the element in the j-th row and j'-th column of $\boldsymbol{R}(l)$. To enable a high quality transmission, these interferences must be compensated. This is done by the DET.

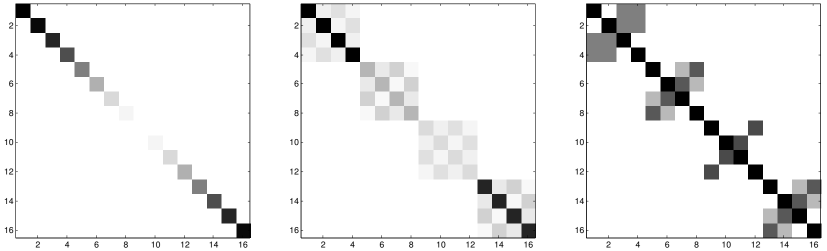
The vector-valued transmission model is able to represent a wide range of transmission schemes. Fig. 1.3 shows the resulting channel matrix for a MIMO system for different number of transmit/receive antennas. Fig. 1.4 shows the channel matrix for OFDM with/without spreading. Fig. 1.5 shows the channel matrix for MIMO-OFDM. In the last two cases, a two path channel with delay T_s and equal amplitudes has been assumed. In Fig. 1.3-1.5 the darker the elements the larger are the absolute values of the entries of the corresponding channel matrix.



(a) $M_{tx} = 8, M_{rx} = 8$.

(b) $M_{tx} = 8, M_{rx} = 16$.

Figure 1.3: Visualization of the channel matrix for a MIMO transmission scheme with different number of transmit/receive antennas. M_{tx}/M_{rx} is the number of transmit/receive antennas.



(a) $\boldsymbol{R}_{OFDM}$.

(b) $\boldsymbol{R}_{MC-CDM}$ with noninterleaved spreading.

(c) $\boldsymbol{R}_{MC-CDM}$ with interleaved spreading.

Figure 1.4: Visualization of the channel matrix for OFDM for 16 subcarriers and spreading over four subcarriers with/without interleaving for a two path channel with delay T_s and equal amplitudes.

1.1.1 Block Transmission

In many practical cases, the source symbols are divided into many shorter blocks. Every two successive blocks are separated by a guard time longer than the duration of the channel impulse response. In this case, interblock interference can be avoided, i.e. no interference takes place between transmitted blocks. Thus, equalization of each block can be treated independently of other blocks. This is valid for the decoding as well if the channel coding does not take place over blocks. Assuming that each block

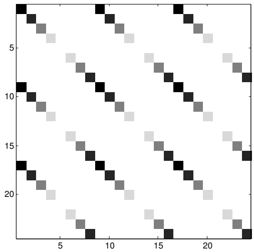


Figure 1.5: Visualization of the channel matrix for MIMO-OFDM with 8 subcarriers and three transmit antennas and two receive antennas. The channel between any pair of transmit/receive antennas is a two path channel with delay T_s and equal amplitudes.

consists of N_b vectors of transmit symbols $\boldsymbol{x}(l), \boldsymbol{x}(l+1), \cdots, \boldsymbol{x}(l+N_b-1)$, we can combine these vectors to create a longer transmit vector $\boldsymbol{x}_b$ of length $N = M_{tx} \cdot N_b$. In this case, Eq. (1.4) can be rewritten as:

$$\begin{aligned} \tilde{\boldsymbol{x}}_b &= \boldsymbol{R}_b \cdot \boldsymbol{x}_b + \tilde{\boldsymbol{n}}_b \\ \boldsymbol{\Phi}_{\tilde{\boldsymbol{n}}_b \tilde{\boldsymbol{n}}_b} &= \frac{N_0}{2} \cdot \boldsymbol{R}_b \end{aligned} \tag{1.7}$$

where:

$$\begin{aligned} \boldsymbol{x}_b &= \left[\boldsymbol{x}^T(l), \boldsymbol{x}^T(l+1), \cdots, \boldsymbol{x}^T(l+N_b-1)\right]^T \\ \tilde{\boldsymbol{x}}_b &= \left[\tilde{\boldsymbol{x}}^T(l), \tilde{\boldsymbol{x}}^T(l+1), \cdots, \tilde{\boldsymbol{x}}^T(l+N_b-1)\right]^T \\ \tilde{\boldsymbol{n}}_b &= \left[\tilde{\boldsymbol{n}}^T(l), \tilde{\boldsymbol{n}}^T(l+1), \cdots, \tilde{\boldsymbol{n}}^T(l+N_b-1)\right]^T \end{aligned} \tag{1.8}$$

and

$$\boldsymbol{R}_b = \begin{bmatrix} \boldsymbol{R}(0) & \boldsymbol{R}(-1) & \boldsymbol{R}(-2) & \cdots & \cdots & \cdots & \boldsymbol{R}(-N_b+1) \\ \boldsymbol{R}(1) & \boldsymbol{R}(0) & \boldsymbol{R}(-1) & \cdots & \cdots & \cdots & \boldsymbol{R}(-N_b+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{R}(N_b-1) & \cdots & \cdots & \cdots & \boldsymbol{R}(2) & \boldsymbol{R}(1) & \boldsymbol{R}(0) \end{bmatrix}. \tag{1.9}$$

If the channel is time-variant during one block, the submatrices in Eq. (1.9) change from the above left corner to the below right one [46]. Usually, the matrix $\boldsymbol{R}_b$ is not fully occupied because the block length N_b is much longer than the duration of the channel impulse response. Comparing Eq. (1.4),(1.7), we recognize:

- The convolution has been replaced by a multiplication.
- The time index has been omitted because no interference is taking place between successive transmitted blocks.

We assume in this work a block transmission where the channel coding is restricted to single blocks. Therefore, the transmission model is described after omitting the block index , cf. Fig. 1.6, as:

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \boldsymbol{R} \cdot \boldsymbol{x} + \tilde{\boldsymbol{n}} \\ \boldsymbol{\Phi}_{\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}} &= \frac{N_0}{2} \cdot \boldsymbol{R}.\end{aligned} \tag{1.10}$$

In addition, it is assumed that the channel impulse response is time-invariant. The digital source generates statistically independent (memoryless digital source) and equally probable bits. Thus, the elements of the symbol alphabet Ψ occur with equal probability. Important to mention is also:

$$\begin{aligned}\boldsymbol{R} &= \boldsymbol{R}^H \\ \boldsymbol{R} &\geq 0.\end{aligned} \tag{1.11}$$

$\boldsymbol{R} \geq 0$ means that $\boldsymbol{R}$ is positive semidefinite.

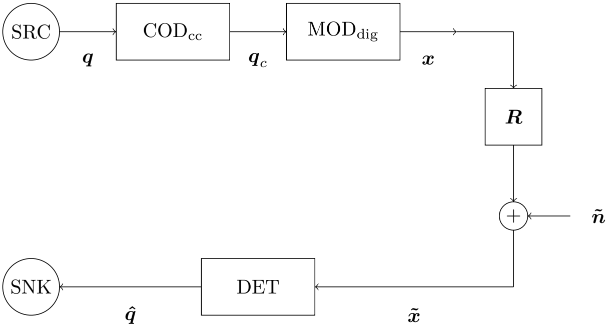


Figure 1.6: Discrete-time vector-valued block transmission model.

1.1.2 Uncoded Transmission

The uncoded discrete-time vector-valued block transmission model is given in Fig. 1.7. Compared with Fig. 1.6, we notice that the detector DET becomes an equalizer EQ.

The decision device DECI depends on the symbol alphabet Ψ. This model is used in this work for investigating different equalization methods. Eq. (1.10) is still valid.

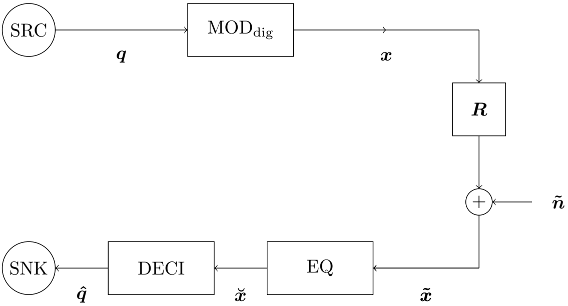


Figure 1.7: Uncoded discrete-time vector-valued block transmission model.

1.1.3 Coded Transmission Over AWGN Channel

Fig. 1.8 shows a coded transmission over a discrete-time AWGN channel. We limit ourself to a binary phase shift keying (BPSK) symbol alphabet and to binary linear block codes. This model is used for investigating iterative decoding methods over the AWGN channel. The decision device for BPSK is a sgn function. In this case

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \boldsymbol{x} + \boldsymbol{n} \\ \boldsymbol{\Phi}_{\boldsymbol{nn}} &= \frac{N_0}{2} \cdot \boldsymbol{I}.\end{aligned} \tag{1.12}$$

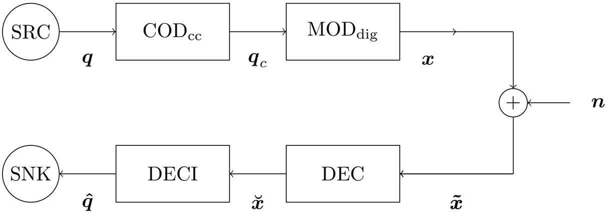


Figure 1.8: Coded transmission over a discrete-time AWGN channel.

1.2 Vector Detection

The vector detector DET in Fig. 1.6 has the vector of receive symbols $\tilde{\boldsymbol{x}}$ in Eq. (1.10) as input and possesses the knowledge about the channel matrix $\boldsymbol{R}$, the symbol alphabet Ψ, and the used forward error correction code. Based on this knowledge, the vector detector DET delivers, in case of ML-detection, the vector of decided bits $\hat{\boldsymbol{q}}$ which has been transmitted with the largest probability.

$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}\in\{0,1\}^{r\cdot N\cdot\log_2 M}}{\mathbf{argmax}} \; p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}) \tag{1.13}$$

$p(\dot{\boldsymbol{x}}|\boldsymbol{q})$ is the conditional probability density function of the vector of receive symbols $\tilde{\boldsymbol{x}}$ given that $\boldsymbol{q}$, a realization to be understood as an event, has been occurred (transmitted). $\dot{\boldsymbol{x}}$ represents the variable of the conditional probability density function. $\underset{\boldsymbol{q}\in\{0,1\}^{r\cdot N\cdot\log_2 M}}{\mathbf{argmax}} \; p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q})$ returns the vector $\boldsymbol{q}$ which has the largest $p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q})$.

Due to complexity issues, the vector detection is spit in general into two parts: vector equalization and vector decoding. However, if a ML-detection is desired, the vector detector can not be separated apriori. In this case, vector equalization and vector decoding must be considered jointly. In other words, splitting the vector detection can not be done in general without performance lost.

1.2.1 Separate Equalization and Decoding

The detector DET in this case has a feed forward structure as illustrated in Fig. 1.9. The vector detector performs at first a vector equalization process and the vector of soft/hard decided symbols $\check{\boldsymbol{x}}/\hat{\boldsymbol{x}}$ is fed to the vector decoder, which in turn delivers, after DECI, the vector of decided bits $\hat{\boldsymbol{q}}$. It is obvious that no "knowledge" exchange takes place between the vector equalizer EQ and the vector decoder DEC. This kind of detection is widely used in practice because it is less complex than the joint equalization and decoding. However, even if both parts of the detector are maximum likelihood, this does not mean that the whole detection is also maximum likelihood because the EQ does not exploit the knowledge that not all $\boldsymbol{x}$ are possible because of the channel coding.

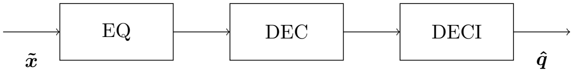


Figure 1.9: Separate equalization and decoding.

1.2.2 Joint Equalization and Decoding

In the joint equalization and decoding case, cf. Fig. 1.10, the vector equalizer EQ and the vector decoder DEC deliver ”soft” decisions. In addition to the vector of receive symbols $\tilde{\boldsymbol{x}}$, the vector equalizer EQ does use the knowledge (constraints) from the vector decoder DEC to improve its performance, which in turn improves the performance of the vector decoder DEC. The ”knowledge” exchange takes place a few times.

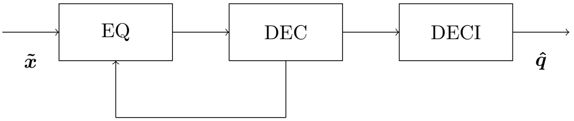


Figure 1.10: Joint equalization and decoding.

1.3 Vector Equalization

In this section, we focus on the vector equalization in the uncoded case, cf. Fig. 1.7 and Eq. (1.10). We start with ML-vector equalization. After that, we present various suboptimum vector equalization schemes. The input of the vector equalizer EQ is the vector of receive symbols $\tilde{\boldsymbol{x}} \in \mathbb{C}^N$ and the output, including the decision device DECI, is the vector of decided symbols $\hat{\boldsymbol{x}}$. We notice that for uncoded transmission we obtain $\hat{\boldsymbol{q}}$ directly from $\hat{\boldsymbol{x}}$.

1.3.1 Blockwise Maximum Likelihood Vector Equalization

The optimum vector equalizer has to deliver the vector of decided symbols $\hat{\boldsymbol{x}}$ which has been sent with the largest probability :

$$\hat{\boldsymbol{x}}_{\mathrm{APP}} = \underset{\boldsymbol{\xi} \in \mathcal{X}}{\mathbf{argmax}} \, \mathrm{Prob}(\boldsymbol{x} = \boldsymbol{\xi} | \tilde{\boldsymbol{x}}). \tag{1.14}$$

$\mathcal{X}$ is the set of all possible vector of transmit symbols with cardinality M^N. $\boldsymbol{\xi}$ is one possible vector of transmit symbols. Prob(|) is a conditional probability. The arguments in Prob(|) must be understood as events, where the event on the right side of | has occurred. In this sense, $\mathrm{Prob}(\boldsymbol{x} = \boldsymbol{\xi} | \tilde{\boldsymbol{x}})$ in Eq. (1.14) delivers the probability that $\boldsymbol{x} = \boldsymbol{\xi}$ had been transmitted, given that $\tilde{\boldsymbol{x}}$ has been received. We notice that Eq. (1.14) maximizes the aposteriori probability $\mathrm{Prob}(\boldsymbol{x} = \boldsymbol{\xi}|\tilde{\boldsymbol{x}})$, therefore Eq. (1.14) is called blockwise maximum aposteriori probability equalization rule.

Applying the Bayesian rule, Eq. (1.14) can be rewritten as:

$$\hat{\boldsymbol{x}}_{\mathrm{APP}} = \underset{\boldsymbol{\xi}\in\mathcal{X}}{\mathbf{argmax}}\; p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{x} = \boldsymbol{\xi}) \cdot \mathrm{Prob}(\boldsymbol{\xi}). \tag{1.15}$$

$p(\dot{\boldsymbol{x}}|\boldsymbol{x} = \boldsymbol{\xi})$ represents the transition probability density function of the channel. $\mathrm{Prob}(\boldsymbol{\xi})$ is the probability of sending $\boldsymbol{\xi}$. It is assumed that the elements of the symbol alphabet are equally probable. Thus, the last relation can be further simplified as follows:

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \underset{\boldsymbol{\xi}\in\mathcal{X}}{\mathbf{argmax}}\; p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{x} = \boldsymbol{\xi}). \tag{1.16}$$

Eq. (1.16) is known as the blockwise maximum likelihood vector equalization rule.

The noise process $\tilde{\boldsymbol{n}}$ in Eq. (1.10) follows a circular symmetric complex normal distribution [77] with zero mean and covariance matrix $\boldsymbol{\Phi}_{\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}}$. $p(\dot{\boldsymbol{x}}|\boldsymbol{x} = \boldsymbol{\xi})$ has the same distribution but a mean of $\boldsymbol{R} \cdot \boldsymbol{\xi}$. This leads to

$$p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{x} = \boldsymbol{\xi}) \propto \exp\left\{-\frac{1}{2} \cdot (\tilde{\boldsymbol{x}} - \boldsymbol{R} \cdot \boldsymbol{\xi})^{H} \cdot \boldsymbol{\Phi}_{\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}}^{-1} \cdot (\tilde{\boldsymbol{x}} - \boldsymbol{R} \cdot \boldsymbol{\xi})\right\}. \tag{1.17}$$

Based on Eq. (1.10),(1.16),(1.17) the decision rule can be rewritten as follows

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \underset{\boldsymbol{\xi}\in\mathcal{X}}{\mathbf{argmin}}\; \left\{(\tilde{\boldsymbol{x}} - \boldsymbol{R} \cdot \boldsymbol{\xi})^{H} \cdot \boldsymbol{R}^{-1} \cdot (\tilde{\boldsymbol{x}} - \boldsymbol{R} \cdot \boldsymbol{\xi})\right\}. \tag{1.18}$$

Considering that $\boldsymbol{R} = \boldsymbol{R}^{H}$ this can be simplified to:

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \underset{\boldsymbol{\xi}\in\mathcal{X}}{\mathbf{argmin}}\; \left\{\frac{1}{2} \cdot \boldsymbol{\xi}^{H} \cdot \boldsymbol{R} \cdot \boldsymbol{\xi} - \Re\left\{\boldsymbol{\xi}^{H} \cdot \tilde{\boldsymbol{x}}\right\}\right\}. \tag{1.19}$$

The argument in Eq. (1.19) is also known as the Mahalanobis distance [7].

We notice from the last relation that the ML-vector equalizer needs to minimize the last relation with respect to the set of all possible vectors of transmit symbols $\boldsymbol{\xi}$, which form a discrete set of cardinality M^{N}. Thus, exhaustive search is required in general. This can be realized for small M^{N}. However, this is too complex in general to be implemented in practice. Thus, suboptimum vector equalizers have been an important research topic. They aim to achieve near optimum performance with much lower complexity.

1.3.2 Structure of Suboptimum Vector Equalizer

Fig. 1.11 shows the generic structure of many suboptimum vector equalizers. If the switch is connected, we consider an iterative scheme. In this case, the suboptimum vector equalizer consists of two parts. The first part is the feed forward part, where the vector of receive symbols $\tilde{\boldsymbol{x}}$ is multiplied with the feed forward matrix $\boldsymbol{W}_f$. The second part is the feedback part. It consists of the estimation function ESTI and the feedback matrix $\boldsymbol{W}_b$. The second part attempts to generate an estimate of the interference in order to subtract it. In this case vector equalization is often called interference cancellation. This procedure can be repeated several times.

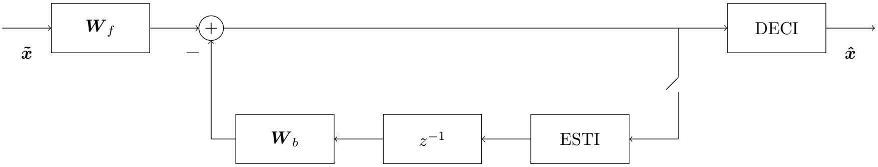


Figure 1.11: Generic structure of interference cancellation.

Depending on the estimation function ESTI, we distinguish between:

- *Hard* interference cancellation: In this case the estimation function ESTI provides hard estimates which belong to the symbol alphabet Ψ.

- *Soft* interference cancellation: In this case the estimation function ESTI provides soft estimates which can take any value.

The superiority of the soft iterative interference cancellation over the hard one has been already shown and is widely accepted.

At the end of the interference cancellation process the decision device DECI generates the vector of decided symbols $\hat{\boldsymbol{x}}$. Both feed forward and feedback parts can work on symbol basis (serial update) or on vector basis (parallel update).

If the switch in Fig. 1.11 is off, we are dealing with a noniterative vector equalization technique. In this case the vector of receive symbols $\tilde{\boldsymbol{x}}$ is multiplied with the feed forward matrix $\boldsymbol{W}_f$. After that a decision is taken by means of the decision device DECI.

There are many vector equalizers which use the principle depicted in Fig. 1.11. Multistage detector MD, zero forcing-block linear equalizer ZF-BLE, minimum mean square error-block linear equalizer MMSE-BLE, successive interference cancellation SIC, parallel interference cancellation PIC [84], and recurrent neural networks RNNs [10] are some examples. They differ from each other by serial/parallel update, soft/hard estimation, one step/iterative cancellation. In the following, we make this comparison for ZF-BLE and MMSE-BLE. Chapters 2 and 3 are dedicated to RNNs.

Zero Forcing-Block Linear Equalizer ZF-BLE

The ZF-BLE can be applied only if the channel matrix $\boldsymbol{R}$ is invertible. In this case it eliminates the interference totally by multiplying the vector of the receive symbols $\tilde{\boldsymbol{x}}$ with the inverse of the channel matrix $\boldsymbol{W}_{zf} = \boldsymbol{R}^{-1}$, cf. Fig. 1.12 and Eq. (1.20). However, noise amplification takes place.

$$\begin{aligned} \breve{\boldsymbol{x}} &= \boldsymbol{W}_{zf} \cdot \tilde{\boldsymbol{x}} = \boldsymbol{x} + \underbrace{\boldsymbol{R}^{-1} \cdot \tilde{\boldsymbol{n}}}_{\boldsymbol{n}_{zf}} \\ \hat{\boldsymbol{x}} &= \mathrm{DECI}(\breve{\boldsymbol{x}}) \\ \boldsymbol{\Phi}_{\boldsymbol{n}_{zf}\boldsymbol{n}_{zf}} &= \frac{N_0}{2} \cdot \boldsymbol{R}^{-1} \end{aligned} \tag{1.20}$$

Comparing Fig. 1.12 with Fig. 1.11, we notice that $\boldsymbol{W}_f = \boldsymbol{W}_{zf}$ and the switch is off (noniterative approach).

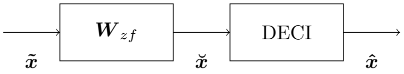


Figure 1.12: Zero forcing-block linear equalizer ZF-BLE.

Minimum Mean Square Error-Block Linear Equalizer MMSE-BLE

The MMSE-BLE aims to reduce the noise enhancement drawback of the ZF-BLE by multiplying the vector of receive symbols $\tilde{\boldsymbol{x}}$ with the matrix $\boldsymbol{W}_{mmse}$, which has been derived according to the MMSE criterion as follows:

$$\boldsymbol{W}_{mmse} = \underset{\boldsymbol{W}_{mmse}}{\mathbf{argmin}}\; E_{xp}\left\{ |\boldsymbol{x} - \boldsymbol{W}_{mmse} \cdot \tilde{\boldsymbol{x}}|^2 \right\} \tag{1.21}$$

which leads to:

$$\boldsymbol{W}_{mmse} = \left[\boldsymbol{R} + \frac{1}{\mathrm{SNR}} \cdot \boldsymbol{I} \right]^{-1} \tag{1.22}$$

SNR represents the signal-to-noise ratio and E_{xp} is the mathematical expectation.

In this case:

$$\begin{aligned} \breve{\boldsymbol{x}} &= \boldsymbol{W}_{mmse} \cdot \tilde{\boldsymbol{x}} \\ \hat{\boldsymbol{x}} &= \mathrm{DECI}(\breve{\boldsymbol{x}}) \\ \boldsymbol{\Phi}_{\boldsymbol{n}_{mmse}\boldsymbol{n}_{mmse}} &= \frac{N_0}{2} \cdot \boldsymbol{W}_{mmse} \cdot \boldsymbol{R} \cdot \boldsymbol{W}_{mmse} \end{aligned} \tag{1.23}$$

Comparing Fig. 1.13 with Fig. 1.11, we notice that $\boldsymbol{W}_f = \boldsymbol{W}_{mmse}$ and the switch is off (noniterative approach). ZF-BLE and MMSE-BLE coincide for large SNR.

ZF-BLE and MMSE-BLE are originally noniterative approaches which require a matrix inversion. Matrix inversion can be done iteratively by applying the Jacobi or Gauss-Seidel method [5]. In this case, the switch is on and the estimation function ESTI is a linear function. In contrast to iterative vector equalization based on RNNs presented in Chapters 2 and 3, the iterative matrix inversion does not change the linear character of these vector equalizers.

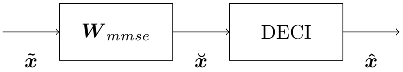


Figure 1.13: Minimum mean square error-block linear equalizer MMSE-BLE.

1.4 Vector Decoding

This section is based mainly on [34]. Forward error correction techniques aim to achieve error free transmission over noisy channels with the maximum possible transmission rate. They emerged from the famous statement of Shannon [74]: Arbitrarily high reliable transmission over noise corrupted channels can be achieved if the transmission rate is less than the channel capacity.

This is done by adding redundancy (extra bits) to the message $\boldsymbol{q}$, such that the codewords $\boldsymbol{q}_c$ are sufficiently distinguishable and the transmitted messages can be gained at the receiver, even if the noisy channel corrupts some bits of the codewords during the transmission.

The vector coding mentioned in Sec. 1.1 is nothing else than a scalar coding, which can be done in time and/or component direction, depending on the physical meaning of the components in the vector-valued transmission model. For multiuser scenarios vector coding is a parallel scalar coding, so coding takes place in time direction. For OFDM transmission schemes coding can take place in the subcarrier and/or time direction. Therefore, we focus in this section on scalar channel coding and we use the transmission model depicted in Fig. 1.8 and Eq. (1.12). We restrict ourself to binary linear block codes with BPSK symbol alphabet.

Binary linear block codes are characterized by the following parameters: n, k, $d_{\min}$, $\boldsymbol{G}$, $\boldsymbol{H}_p$. n is the length of the codeword. k is the length of the information word. The minimum distance $d_{\min}$ describes the smallest number of bit positions, which is required to change from one codeword to another codeword when bits are flipped. The generator matrix $\boldsymbol{G}$ is a binary matrix of size $k \times n$. The parity check matrix $\boldsymbol{H}_p$ is a binary matrix of size $(n-k) \times n$. Furthermore it holds:

$$\boldsymbol{H}_p \cdot \boldsymbol{G}^T = \boldsymbol{0}_{(n-k)\times k}. \tag{1.24}$$

Addition and multiplication are modulo 2.

In Fig. 1.8, the digital source generates a sequence of bits, which are grouped into blocks of length k, each of them is called information word. The encoder maps the information words to codewords of length $n > k$, where the ratio $r = k/n$ is the code rate. Because the encoding is a bijective map, it can be understood as a table, which maps unambiguously binary vectors of length k to binary vectors of length n. This table contains 2^k vectors. It is obvious that this table becomes unrealizable if k increases. Therefore mapping the information words to the codewords takes place by another way, namely by the binary generator matrix $\boldsymbol{G}$.

$$\boldsymbol{q}_c = \boldsymbol{G}^T \cdot \boldsymbol{q} \tag{1.25}$$

From Eq. (1.24), a codeword $\boldsymbol{q}_c$ belongs to a code if the following relation is fulfilled

$$\boldsymbol{H}_p \cdot \boldsymbol{q}_c = \boldsymbol{0}_{n-k}. \tag{1.26}$$

By means of the last relation, it can be decided, whether a received message is a codeword of the used code. However, to correct an error caused by the noisy channel, a decoding process is needed at the receiver. We distinguish between blockwise and symbolwise decoding approaches. The set of all codewords of a code is called code book $\mathcal{C}$.

1.4.1 Blockwise Maximum Aposteriori Probability

The decoding process in this case aims to maximize the probability of a correct decoding of codewords.

$$\hat{\boldsymbol{c}}_{\text{APP}} = \underset{\boldsymbol{c}\in\mathcal{C}}{\textbf{argmax}} \, \text{Prob}(\boldsymbol{q}_c = \boldsymbol{c}|\tilde{\boldsymbol{x}}) \tag{1.27}$$

By means of the Bayesian rule, the last relation can be rewritten as:

$$\hat{\boldsymbol{c}}_{\mathrm{APP}} = \underset{\boldsymbol{c}\in\mathcal{C}}{\mathbf{argmax}}\; p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c}) \cdot \mathrm{Prob}(\boldsymbol{c}) \tag{1.28}$$

Assuming that all information words are equally probable, we obtain the block maximum likelihood decoding rule:

$$\hat{\boldsymbol{c}}_{\mathrm{ML}} = \underset{\boldsymbol{c}\in\mathcal{C}}{\mathbf{argmax}}\; p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c}) \tag{1.29}$$

Following the same approach as in Sec. 1.3.1 and taking into account Eq. (1.12), the block maximum likelihood decoding approach leads to the minimum Euclidean distance. Thus, according to this approach, the Euclidean distance between the received message and all 2^k codewords has to be calculated and a decision is taken in the favor of the codeword with the minimum Euclidean distance to the received message. The Viterbi algorithm is a block maximum likelihood decoding approach for convolutional codes [86].

The block maximum likelihood decoding approach leads for binary symmetric channels to the minimum Hamming distance.

1.4.2 Symbolwise Maximum Aposteriori Probability

The decoding process in this case aims to maximize the probability of a correct decoding of code symbols.

$$\hat{c}_{j,\mathrm{s/s\text{-}APP}} = \underset{s\in\{0,1\}}{\mathrm{argmax}}\; \mathrm{Prob}(q_{c,j} = s|\tilde{\boldsymbol{x}}) \tag{1.30}$$

The importance of this approach lies in the ability to approximate it iteratively as introduced later in Chapter 4. $q_{c,j} \in \{0,1\}$ is the j-th element of the transmitted codeword. Thus, $\boldsymbol{q}_c$ is a valid codeword, i.e. $\boldsymbol{q}_c = \boldsymbol{c} \in \mathcal{C}$, cf. Fig. 1.8.

Eq. (1.30) comprises the calculation of two probabilities, the first one for $s = 0$ and the second one for $s = 1$. Therefore, it is more familiar to consider the ratio of them. The log-likelihood ratio (LLR) of the j-th code symbol is defined therefore as:

$$L(c_j) = \ln \frac{\mathrm{Prob}(q_{c,j} = 0|\tilde{\boldsymbol{x}})}{\mathrm{Prob}(q_{c,j} = 1|\tilde{\boldsymbol{x}})} \tag{1.31}$$

where the sign of $L(c_j)$ delivers the decision of $\hat{c}_j$ being 0 or 1

$$\hat{c}_j = \begin{cases} 0 & : L(c_j) \geq 0 \\ 1 & : L(c_j) < 0 \end{cases} \tag{1.32}$$

and the magnitude $|L(c_j)|$ the reliability of the decision.
The advantage of the log-likelihood ratios arises from the fact that multiplying probabilities leads to the summation of the log-likelihood ratios.

We split the code book $\mathcal{C}$ into two subcode books $\mathcal{C}_0$ and $\mathcal{C}_1$. $\mathcal{C}_0$ contains all codewords from the original code book $\mathcal{C}$, which contain a 0 at the j-th position. $\mathcal{C}_1$ contains all codewords from the original code book $\mathcal{C}$, which contain a 1 at the j-th position. In this case, Eq. (1.31) can be rewritten as:

$$L(c_j) = \ln \frac{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_0} \text{Prob}(\boldsymbol{q}_c = \boldsymbol{c}|\tilde{\boldsymbol{x}})}{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_1} \text{Prob}(\boldsymbol{q}_c = \boldsymbol{c}|\tilde{\boldsymbol{x}})}. \tag{1.33}$$

According to the Bayesian rule:

$$L(c_j) = \ln \frac{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_0} p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c}) \cdot \text{Prob}(\boldsymbol{c})}{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_1} p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c}) \cdot \text{Prob}(\boldsymbol{c})}. \tag{1.34}$$

Eq. (1.34) represents the symbol-by-symbol maximum aposteriori probability decoding approach (s/s APP).

If all codewords are equally probable, we obtain from the last relation the symbol-by-symbol maximum likelihood (s/s ML) decoding approach

$$L(c_j) = \ln \frac{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_0} p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c})}{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_1} p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c})}. \tag{1.35}$$

For memoryless channels:

$$p(\dot{\boldsymbol{x}} = \tilde{\boldsymbol{x}}|\boldsymbol{q}_c = \boldsymbol{c}) = \prod_{j'} p(\dot{x}_{j'} = \tilde{x}_{j'}|q_{c,j'} = c_{j'}). \tag{1.36}$$

Inserting Eq. (1.36) in Eq. (1.35) leads to:

$$L(c_j) = \ln \frac{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_0} \prod\limits_{j'} p(\dot{x}_{j'} = \tilde{x}_{j'}|q_{c,j'} = c_{j'})}{\sum\limits_{\boldsymbol{c}\in\mathcal{C}_1} \prod\limits_{j'} p(\dot{x}_{j'} = \tilde{x}_{j'}|q_{c,j'} = c_{j'})}. \tag{1.37}$$

We define now the intrinsic L-value (channel L-value):

$$L_{\text{ch}}(\tilde{x}_j) = \ln \frac{p(\dot{x}_j = \tilde{x}_j|q_{c,j} = 0)}{p(\dot{x}_j = \tilde{x}_j|q_{c,j} = 1)}. \tag{1.38}$$

For an AWGN channel $\mathcal{N}(0, \sigma_n^2)$ and BPSK modulation, $x_j = 1 - 2 \cdot q_{c,j}$:

$$\begin{aligned} p(\dot{x}_j = \tilde{x}_j | q_{c,j} = 0) &= \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_n^2}} \cdot \exp\left\{ -\frac{(\tilde{x}_j - 1)^2}{2 \cdot \sigma_n^2} \right\} \\ p(\dot{x}_j = \tilde{x}_j | q_{c,j} = 1) &= \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_n^2}} \cdot \exp\left\{ -\frac{(\tilde{x}_j + 1)^2}{2 \cdot \sigma_n^2} \right\}. \end{aligned} \tag{1.39}$$

Inserting Eq. (1.39) in Eq. (1.38) leads to:

$$L_{\text{ch}}(\tilde{x}_j) = \frac{2 \cdot \tilde{x}_j}{\sigma_n^2}. \tag{1.40}$$

Depending on

$$p(\dot{x}_j = \tilde{x}_j) = \frac{1}{2} \cdot p(\dot{x}_j = \tilde{x}_j | q_{c,j} = 0) + \frac{1}{2} \cdot p(\dot{x}_j = \tilde{x}_j | q_{c,j} = 1) \tag{1.41}$$

for $\text{Prob}(q_{c,j} = 0) = \text{Prob}(q_{c,j} = 1) = 0.5$ and using Eq. (1.39) it can be found that:

$$p(\dot{x}_j = \tilde{x}_j | q_{c,j}) = \frac{p(\dot{x}_j = \tilde{x}_j)}{1 + \exp\left\{ -L_{\text{ch}}(\tilde{x}_j) \right\}} \cdot \exp\left\{ -L_{\text{ch}}(\tilde{x}_j) \cdot q_{c,j} \right\} \tag{1.42}$$

As mentioned before, $q_{c,j} \in \{0, 1\}$. Inserting the last relation in Eq. (1.37) leads to:

$$L(c_j) = \underbrace{L_{\text{ch}}(\tilde{x}_j)}_{L_{\text{int}}(c_j)} + \underbrace{\ln \frac{\sum\limits_{c \in \mathcal{C}_0} \prod\limits_{j' \neq j} \exp\left\{ -L_{\text{ch}}(\tilde{x}_{j'}) \cdot c_{j'} \right\}}{\sum\limits_{c \in \mathcal{C}_1} \prod\limits_{j' \neq j} \exp\left\{ -L_{\text{ch}}(\tilde{x}_{j'}) \cdot c_{j'} \right\}}}_{L_{\text{ext}}(c_j)}. \tag{1.43}$$

We conclude from the last relation that the LLR of the j-th code symbol consists of the intrinsic L-value $L_{\text{int}}(c_j) = L_{\text{ch}}(\tilde{x}_j)$, which contains information about the code symbol from its received version and the extrinsic L-value $L_{\text{ext}}(c_j)$, which uses the constraints of the code to obtain information about the j-th coded symbol from all other code symbols.

The famous BCJR [3] decoding algorithm is a decoding algorithm based on Eq. (1.43), where the numerator and denominator are calculated in an efficient way based on the trellis description of the code.

For a single parity check code where the parity check matrix becomes a row vector it can be shown that

$$L_{\text{ext}}(c_j) = \ln \frac{1 + \prod\limits_{j' \neq j} \tanh\left[\frac{L(\tilde{x}_{j'})}{2}\right]}{1 - \prod\limits_{j' \neq j} \tanh\left[\frac{L(\tilde{x}_{j'})}{2}\right]}. \tag{1.44}$$

Binary linear block codes can be understood as a set of single parity check codes. Thus, iterative decoding algorithms exploit this fact by exchanging iteratively the L-values of the code symbols. A detailed description of iterative decoding is given in Chapter 4. $j,\ j' \in \{1, 2, \cdots n\}$ for Eq.(1.30)-(1.44).

1.5 Chapter Summary

In this chapter we have given an overview of the continuous-time vector-valued transmission model and its individual elements. The discrete-time counterpart has been introduced too, where the importance of the channel matrix $\boldsymbol{R}$ for equalization has been explicitly represented. Furthermore, we have graphically shown the channel matrix for few widely used transmission schemes as multiantanna, OFDM and MIMO-OFDM.

Two special cases have been mentioned specifically, namely uncoded transmission and coded transmission over AWGN channel. The first case is relevant to compare the performance of different suboptimum equalization schemes, whereas the second one is relevant for comparing different codes and decoding algorithms.

The maximum likelihood rule for detection, equalization and decoding has been derived and the importance of suboptimum schemes has been emphasized. The generic structure of an iterative interference cancellation has been given and two special cases have been mentioned.

A short introduction to the mathematical background needed for iterative decoding has been given as well.

Chapter 2

Dynamical Neural Networks

THIS chapter provides a detailed investigation of recurrent neural networks and high order recurrent neural networks as dynamical systems (DSs). We focus on their qualitative and long term dynamical behavior. Training (learning) algorithms are not considered. This investigation includes an extensive study of the local and global asymptotical stability, where new stability conditions are derived and known ones are extended. The stability investigation using time-variant activation functions is considered as well. This represents an important aspect, when using recurrent neural networks as vector equalizer.

We start with a short introduction of DSs followed by essential definitions of stability and the Lyapunov method of stability. After that, a novel method is introduced to define continuous-time dynamical systems (CTDSs), which share specific properties with given discrete-time dynamical systems (DTDSs). Thereafter, different recurrent neural networks structures are introduced. Before presenting the details of the stability investigation, some definitions are introduced and various lemmas are proven. The same procedure is repeated for high order recurrent neural networks.

This chapter represents the theoretical basis of this work. Stability results in this chapter are going to be interpreted in the light of the vector equalization and the vector decoding tasks in Chapter 3 and Chapter 4, respectively.

2.1 Dynamical Systems

A DS is a system, whose state varies with time [20]. Virtually, anything that evolves over time can be thought of as a DS [71]. We distinguish between DTDSs and CTDSs. We consider complex-valued DSs and limit the investigation to autonomous ones.

The general form of a DTDS is given by [47],[71]:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \boldsymbol{s}_d[\boldsymbol{u}(l)] \quad : \quad l = 0, 1, 2, \cdots \\ \boldsymbol{u}(0) &= \boldsymbol{u}_{ini} \\ \boldsymbol{u} &= \boldsymbol{u}_r + \imath \boldsymbol{u}_i \quad : \quad \boldsymbol{u}_r, \boldsymbol{u}_i \in \mathbb{R}^N \\ \boldsymbol{s}_d[\boldsymbol{u}_r, \boldsymbol{u}_i] &= \boldsymbol{s}_{d,r}[\boldsymbol{u}_r, \boldsymbol{u}_i] + \imath \boldsymbol{s}_{d,i}[\boldsymbol{u}_r, \boldsymbol{u}_i] \quad : \quad \boldsymbol{s}_{d,r}, \boldsymbol{s}_{d,i} : \mathbb{R}^{2N} \to \mathbb{R}^N. \end{aligned} \tag{2.1}$$

For a CTDS we define [47],[71]:

$$\begin{aligned} \mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} &= \boldsymbol{s}_c[\boldsymbol{u}(t)] \quad : \quad t \geq 0 \\ \boldsymbol{u}(0) &= \boldsymbol{u}_{ini} \\ \mathbf{Y} &= \mathrm{diag}\{\tau_1, \tau_2, \cdots, \tau_N\} \\ \forall j \in \{1, 2, \cdots, N\} &: \tau_j > 0 \\ \boldsymbol{u} &= \boldsymbol{u}_r + \imath \boldsymbol{u}_i \quad : \quad \boldsymbol{u}_r, \boldsymbol{u}_i \in \mathbb{R}^N \\ \boldsymbol{s}_c[\boldsymbol{u}_r, \boldsymbol{u}_i] &= \boldsymbol{s}_{c,r}[\boldsymbol{u}_r, \boldsymbol{u}_i] + \imath \boldsymbol{s}_{c,i}[\boldsymbol{u}_r, \boldsymbol{u}_i] \quad : \quad \boldsymbol{s}_{c,r}, \boldsymbol{s}_{c,i} : \mathbb{R}^{2N} \to \mathbb{R}^N. \end{aligned} \tag{2.2}$$

$\boldsymbol{u}$ is called the state of the DS and N is its dimension. In addition, we assume that $\boldsymbol{s}_{d,r}$, $\boldsymbol{s}_{d,i}$, $\boldsymbol{s}_{c,r}$ and $\boldsymbol{s}_{c,i}$ are continuously differentiable with respect to $\boldsymbol{u}_r$, $\boldsymbol{u}_i$.

Definition 2.1 The *fixed points* $\boldsymbol{u}_{fp}$ of a DTDS, cf. Eq. (2.1), fulfill [47]:

$$\boldsymbol{u}_{fp} = \boldsymbol{s}_d[\boldsymbol{u}_{fp}]. \tag{2.3}$$

Definition 2.2 The *equilibrium points* $\boldsymbol{u}_{ep}$ of a CTDS, cf. Eq. (2.2), fulfill [47]:

$$\boldsymbol{s}_c[\boldsymbol{u}_{ep}] = \boldsymbol{0}_N. \tag{2.4}$$

Definition 2.3 A DTDS reaches a *limit cycle of length* $l_c \in \mathbb{N}$ and $l_c > 1$ if it repeats the trajectory between l and $l + l_c$ infinitely often:

$$\begin{aligned} &\boldsymbol{u}(l) = \boldsymbol{u}(l + l_c) \\ &\boldsymbol{u}(l) \neq \boldsymbol{u}(l + j) \\ &\quad \forall j \in \{1, 2, \cdots, l_c - 1\}. \end{aligned} \tag{2.5}$$

Remark 2.1 The original definitions of DSs in [47],[71] are made for the real-valued case. It can be easily shown that complex-valued DSs of dimension N as described in Eq. (2.1), (2.2) can be represented by real-valued DSs of dimension $2 \cdot N$.

Remark 2.2 The terms fixed/equilibrium point are interchangeable in the literature, cf. [47], [71]. Other terms for fixed/equilibrium point include stationary point, singular point, critical point and rest point [47]. In this thesis, we dedicate the term fixed point for DTDSs and the term equilibrium point for CTDSs.

Remark 2.3 In this thesis, CTDSs are simulated using the first Euler method [50], where it is assumed that $\tau = \tau_1 = \tau_2 = \cdots = \tau_N$ and $\tau/\Delta t = 10$. Simulation based on other numerical schemes, as Runge-Kutta method, reveals similar results. Δt is the sampling step.

2.1.1 Qualitative Behavior of Dynamical Systems

The theory of DSs focuses, among others, on their long term temporal behavior. Instead of giving explicit solutions of Eq. (2.1), (2.2), which is possible in very few cases, it describes their qualitative behavior if the discrete/continuous time tends to infinity (long term evolution). Qualitative behavior includes the existence (or nonexistence) of fixed/equilibrium points, attractors, chaos etc.

A very important method to investigate the qualitative behavior of DSs is the Lyapunov method. Before introducing it, we give some definitions based essentially on [20].

Definition 2.4 The equilibrium point $\boldsymbol{u}_{ep}$ is said to be *uniformly stable* if the trajectory $\boldsymbol{u}(t)$ of the CTDS can be made to stay within a small neighborhood of the equilibrium point $\boldsymbol{u}_{ep}$ if the initial state $\boldsymbol{u}_{ini}$ is close to the equilibrium point $\boldsymbol{u}_{ep}$:

$$\forall \epsilon_1 > 0 : \exists\, \epsilon_2 > 0 : \|\boldsymbol{u}_{ini} - \boldsymbol{u}_{ep}\| < \epsilon_2 \Rightarrow \|\boldsymbol{u}(t) - \boldsymbol{u}_{ep}\| < \epsilon_1 \quad : \quad \forall t > 0.$$

Definition 2.5 The equilibrium point $\boldsymbol{u}_{ep}$ is said to be *convergent* if the trajectory $\boldsymbol{u}(t)$ will approach the equilibrium point $\boldsymbol{u}_{ep}$ as the time t approaches infinity in case that the initial state $\boldsymbol{u}_{ini}$ of the trajectory $\boldsymbol{u}(t)$ is close enough to the equilibrium point $\boldsymbol{u}_{ep}$. Thus, there exists a positive ϵ such that:

$$\|\boldsymbol{u}_{ini} - \boldsymbol{u}_{ep}\| < \epsilon \Rightarrow \lim_{t\to\infty} \|\boldsymbol{u}(t) - \boldsymbol{u}_{ep}\| \to \mathbf{0}_N.$$

Definition 2.6 The equilibrium point $\boldsymbol{u}_{ep}$ is said to be *(locally) asymptotically stable* if it is uniformly stable and convergent.

Definition 2.7 The equilibrium point $\boldsymbol{u}_{ep}$ is said to be *globally asymptotically stable* if it is stable and all trajectories of the CTDS, regardless their initial states $\boldsymbol{u}_{ini}$, converge to $\boldsymbol{u}_{ep}$ as t tends to infinity. This implies that the CTDS does not have any other equilibrium points.

2.1.2 Stability Analysis Based on Lyapunov Functions

The equilibrium point $\boldsymbol{u}_{ep}$ of an autonomous nonlinear CTDS is asymptotically stable if in a small neighborhood $\mathcal{A}$ of $\boldsymbol{u}_{ep}$ there exists a positive definite function $E[\boldsymbol{u}]$. The derivative of $E[\boldsymbol{u}]$ with respect to the time along the dynamical evolution is negative definite in that region. A scalar function $E[\boldsymbol{u}]$ that satisfies these requirements is called a strict Lyapunov function for the equilibrium point $\boldsymbol{u}_{ep}$. In other words, $\boldsymbol{u}_{ep}$ is asymptotically stable if $E[\boldsymbol{u}]$ fulfill the following conditions [20]:

1. $E[\boldsymbol{u}]$ has continuous partial derivatives with respect to the elements of the state vector $\boldsymbol{u}$.

2. $E[\boldsymbol{u}_{ep}] = 0$

3. $E[\boldsymbol{u}] > 0$ if $\boldsymbol{u} \in \mathcal{A} \setminus \{\boldsymbol{u}_{ep}\}$

4. $\dfrac{\mathrm{d}E[\boldsymbol{u}(t)]}{\mathrm{d}t} < 0$ if $\boldsymbol{u} \in \mathcal{A} \setminus \{\boldsymbol{u}_{ep}\}$

5. $\dfrac{\mathrm{d}E[\boldsymbol{u}(t)]}{\mathrm{d}t} = 0$ if $\boldsymbol{u} = \boldsymbol{u}_{ep}$

If $\mathcal{A}$ is unlimited, the CTDS possesses only one equilibrium point and it is globally asymptotically stable.

For local stability, we focus on the equilibrium $\boldsymbol{u}_{ep} = \mathbf{0}_N$. For global stability, we shift Eq. (2.2) to the single equilibrium point.

For a DTDS, we replace the equilibrium point $\boldsymbol{u}_{ep}$ with the fixed point $\boldsymbol{u}_{fp}$ and we replace the differentiation $\dfrac{\mathrm{d}E[\boldsymbol{u}(t)]}{\mathrm{d}t}$ with $\Delta E = E[\boldsymbol{u}(l+1)] - E[\boldsymbol{u}(l)]$.

We notice that there is no systematic approach to find Lyapunov functions. The existence of a Lyapunov function is sufficient but not necessary for the stability.

Remark 2.4 In case of DSs with a unique globally asymptotically stable equilibrium point $\boldsymbol{u}_{ep}$ (fixed point $\boldsymbol{u}_{fp}$), the initial conditions of the dynamical system $\boldsymbol{u}_{ini}$ do not affect the long term behavior of the system. This is not true in the case of DSs with multiple locally asymptotically stable equilibrium points (fixed points). In the later case, the initial conditions influence the equilibrium point $\boldsymbol{u}_{ep}$ (the fixed point $\boldsymbol{u}_{fp}$) to be reached after long term evolution (iteration).

2.1.3 Stability Analysis Based on the Linearization Method

Nonlinear DSs may have more than one fixed/equilibrium point. Their local asymptotical stability can be investigated by applying the linearization method. This is done by linearizing Eq. (2.1), (2.2) in the vicinity of a fixed/equilibrium point using a Taylor series and ignoring high order derivatives. This leads to [16], [71]:

- A fixed point $\boldsymbol{u}_{fp}$ is locally asymptotically stable if the absolute values of all eigenvalues of the Jacobian matrix of $\boldsymbol{s}_d$ at $\boldsymbol{u}_{fp}$ are smaller than one.

- An equilibrium point $\boldsymbol{u}_{ep}$ is locally asymptotically stable if the real parts of all eigenvalues of the Jacobian matrix of $\boldsymbol{s}_c$ at $\boldsymbol{u}_{ep}$ are negative.

2.1.4 Twin Dynamical Systems

In this subsection, we are interested in the definition of $\boldsymbol{s}_c$ in Eq. (2.2) by means of $\boldsymbol{s}_d$ in Eq. (2.1) such that both DSs share specific characteristics. To do that, we need to make further restrictions, namely [63]:

- The fixed points of the DTDS Eq. (2.1), (2.3) are the same as the equilibrium points of the CTDS Eq. (2.2), (2.4), i.e. $\boldsymbol{u}_{fp} = \boldsymbol{u}_{ep}$.

- If $\boldsymbol{s}_d$ is linear and the corresponding DTDS is globally asymptotically stable, $\boldsymbol{s}_c$ must also be linear and the corresponding CTDS must be globally asymptotically stable too.

Let us start with the linear case [16], [71]. The DTDS in the linear case is given by:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \boldsymbol{D}_{ls} \cdot \boldsymbol{u}(l) + \boldsymbol{d}_{ls} \quad : \quad l = 0, 1, 2, \cdots \\ \boldsymbol{u}(0) &= \boldsymbol{u}_{ini}. \end{aligned} \tag{2.6}$$

The DTDS in Eq. (2.6) possesses only one fixed point if $\boldsymbol{I} - \boldsymbol{D}_{ls}$ is nonsingular. In this case:

$$\boldsymbol{u}_{fp} = -[\boldsymbol{D}_{ls} - \boldsymbol{I}]^{-1} \cdot \boldsymbol{d}_{ls} \tag{2.7}$$

and $\boldsymbol{u}_{fp}$ is globally asymptotically stable if:

$$\forall j \in \{1, 2, \cdots, N\} : \left| \lambda^{(j)}_{\boldsymbol{D}_{ls}} \right| < 1. \tag{2.8}$$

The CTDS in the linear case is given by:

$$\begin{aligned} \mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} &= \boldsymbol{C}_{ls} \cdot \boldsymbol{u}(t) + \boldsymbol{c}_{ls} \quad : \quad t \geq 0 \\ \boldsymbol{u}(0) &= \boldsymbol{u}_{ini}. \end{aligned} \tag{2.9}$$

$\mathbf{Y}$ is given in Eq. (2.2). The CTDS in Eq. (2.9) possesses only one equilibrium point if $\boldsymbol{C}$ is nonsingular. In this case:

$$\boldsymbol{u}_{ep} = -\boldsymbol{C}_{ls}^{-1} \cdot \boldsymbol{c}_{ls} \tag{2.10}$$

and $\boldsymbol{u}_{ep}$ is globally asymptotically stable if:

$$\forall j \in \{1, 2, \cdots, N\} : \Re \left\{ \lambda^{(j)}_{\boldsymbol{C}_{ls}} \right\} < 0 \tag{2.11}$$

where $\lambda^{(j)}_{\boldsymbol{C}_{ls}}$ ($\lambda^{(j)}_{\boldsymbol{D}_{ls}}$) are the j-th eigenvalue of $\boldsymbol{C}_{ls}$ ($\boldsymbol{D}_{ls}$).

In case $\boldsymbol{c}_{ls} = \boldsymbol{d}_{ls}$ (both DSs have the same stimulator) the above mentioned conditions for the definition of twin dynamical systems lead in the linear case to:

$$\begin{aligned} \boldsymbol{u}_{ep} &= \boldsymbol{u}_{fp} \\ \boldsymbol{C}_{ls} &= -\boldsymbol{I} + \boldsymbol{D}_{ls}. \end{aligned} \tag{2.12}$$

From Eq. (2.12) we conclude $\forall j \in \{1, 2, \cdots, N\}$:

$$\begin{aligned} \Re \left\{ \lambda^{(j)}_{\boldsymbol{C}_{ls}} \right\} &= -1 + \Re \left\{ \lambda^{(j)}_{\boldsymbol{D}_{ls}} \right\} \\ \Im \left\{ \lambda^{(j)}_{\boldsymbol{C}_{ls}} \right\} &= \Im \left\{ \lambda^{(j)}_{\boldsymbol{D}_{ls}} \right\}. \end{aligned} \tag{2.13}$$

Fig. 2.1 shows that if the fixed point of the linear DTDS in Eq. (2.6), (2.7) is globally asymptotically stable, i.e. Eq. (2.8) is fulfilled, the equilibrium point of the linear

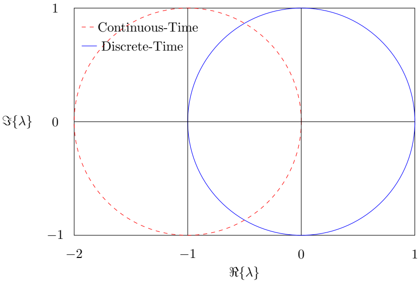


Figure 2.1: Eigenvalues of twin linear dynamical systems.

CTDS in Eq. (2.9), (2.10) through Eq. (2.12), (2.13) is also globally asymptotically stable.

For the nonlinear case, assuming that the linearization method is applicable, this leads to:

$$\boldsymbol{s}_c[\boldsymbol{u}] = -\boldsymbol{u} + \boldsymbol{s}_d[\boldsymbol{u}]. \tag{2.14}$$

This means for the Jacobian matrices:

$$\begin{aligned} \boldsymbol{J}_{\boldsymbol{s}_c}[\boldsymbol{u}] &= -\boldsymbol{I} + \boldsymbol{J}_{\boldsymbol{s}_d}[\boldsymbol{u}] \\ \operatorname{eig}\left\{\boldsymbol{J}_{\boldsymbol{s}_c}[\boldsymbol{u}]\right\} &= -\boldsymbol{I} + \operatorname{eig}\left\{\boldsymbol{J}_{\boldsymbol{s}_d}[\boldsymbol{u}]\right\}. \end{aligned} \tag{2.15}$$

The last equation is the vector-valued notation of Eq. (2.13). eig$\{\ \}$ stands for the eigenvalues of the matrix in the argument. $\boldsymbol{J}_{\boldsymbol{s}_c}[\boldsymbol{u}]$, $\boldsymbol{J}_{\boldsymbol{s}_d}[\boldsymbol{u}]$ represent the Jacobian matrices of $\boldsymbol{s}_c$, $\boldsymbol{s}_d$ at $\boldsymbol{u}$, respectively.

Definition 2.8 The DTDS in Eq. (2.1) and the CTDS in Eq. (2.2) are twin if Eq. (2.14) is fulfilled.

Remark 2.5 Even if the DTDS and the CTDS are twin, the trajectory of the CTDS is generally not an interpolation of the trajectory of the DTDS.

Remark 2.6 The definition of twin dynamical systems is based on coinciding the fixed and equilibrium points, because they represent the solutions of our equalization and decoding tasks. Other behaviors, which might appear in nonlinear DSs as chaos, is not considered in this work, since we know that it does not lead to a (satisfactory) solution.

2.2 Single Layer Recurrent Neural Networks Without Hidden Neurons

The fundamental element of neural networks is the neuron. In the general case, the neuron represents a nonlinear information processing unit. It possesses many inputs and a single output. The functionality of the neuron can be divided into a linear part and a nonlinear part:

- The linear part: performs a weighted summation of the inputs. The result of this summation is called "inner state".

- The nonlinear part: the inner state is subject to a nonlinear mapping by means of a nonlinear element "activation function". The result of this nonlinear mapping is the output of the neuron.

In the discrete-time case, cf. Fig. 2.2 (a):

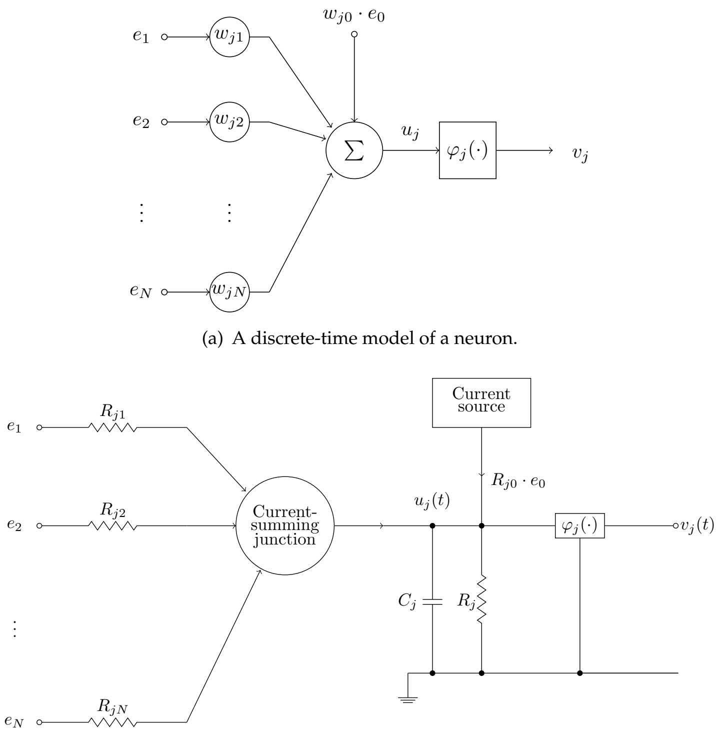
$$\begin{aligned} u_j &= \sum_{j'=0}^{N} w_{jj'} \cdot e_{j'} \\ v_j &= \varphi_j(u_j). \end{aligned} \tag{2.16}$$

In the continuous-time case, cf. Fig. 2.2 (b) which is referred as "additive model" or "resistance-capacitance model" [20]:

$$\begin{aligned} \tau_j \cdot \frac{\mathrm{d}u_j(t)}{\mathrm{d}t} &= -u_j(t) + \sum_{j'=0}^{N} w_{jj'} \cdot e_{j'} \\ v_j &= \varphi_j(u_j) \\ \tau_j &= R_j \cdot C_j > 0 \\ w_{jj'} &= \frac{R_j}{R_{jj'}}. \end{aligned} \tag{2.17}$$

In Eq. (2.16), (2.17) u_j, v_j, φ_j represent the inner state, the output and the activation function of the j-th neuron in a neural network, respectively. $w_{jj'}$ is the weight factor of the j-th neuron for its j'-th input. We restrict ourself in this work to bounded activation functions.

In Eq. (2.17) the inputs, the inner state and the output are assumed to be voltage signals. The summation node represents an ideal current summation junction. The weight factors are scaled resistances (conductances).



(a) A discrete-time model of a neuron.

(b) A continuous-time model of a neuron. The ”resistance-capacitance” model.

Figure 2.2: Discrete and continuous-time models of a neuron.

The single layer recurrent neural network without hidden neurons, for simplicity recurrent neural network (RNN), consists of one layer of N neurons. Each neuron has one external input. The outputs of the neurons $v_1, v_2, \cdots, v_N$ are fed back to the input, i.e. each output is fed back in general to all neurons by weight coefficients.

This class of networks has been attracting a lot of interest because of their widespread applications. They can be either trained to approximate a multiple-input multiple-output system [17] (system identification, usually with hidden neurons), or they can be considered as DSs. In the later case, one of the most important properties of these networks is their ability to solve optimization problems without the need for a training (learning) phase. In this case, the training process, always associated with computational complexity, time and free parameter optimization can be avoided. This relies on the ability of these networks (under specific conditions) to be Lyapunov-stable. This property is desirable in many engineering fields like signal processing, communications, automatic control etc.

One of the first and most well investigated applications of the RNNs is the content addressable memory (CAM). A detailed description of RNN-CAM (two-state neurons) with storage capacity can be found in [20]. Other applications of RNNs are A/D converter [78] and the traveling salesman problem [31].

We focus in this work on the application of RNNs as vector equalizer and we focus on the comparison between discrete- and continuous-time models. The dynamical behavior of RNNs is usually given by the state-space representation.

2.2.1 Continuous-Time Recurrent Neural Networks

Fig. 2.3 shows a continuous-time RNN, to be abbreviated in the following by RNN-1. The dynamical behavior is given by:

$$\begin{aligned}
\mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} &= -\boldsymbol{u}(t) + \boldsymbol{W} \cdot \boldsymbol{v}(t) + \boldsymbol{W}_0 \cdot \boldsymbol{e} \\
\boldsymbol{v}(t) = \boldsymbol{\varphi}[\boldsymbol{u}(t)] &= \Big[\varphi_1[u_1(t)], \varphi_2[u_2(t)], \cdots, \varphi_N[u_N(t)]\Big]^T \\
\mathbf{Y} &= \operatorname{diag}\{\tau_1, \tau_2, \cdots, \tau_N\} \\
\boldsymbol{W}_0 &= \operatorname{diag}\{w_{10}, w_{20}, \cdots, w_{N0}\}\,.
\end{aligned} \tag{2.18}$$

In Eq. (2.18) $\boldsymbol{v}$ is the output, $\boldsymbol{u}$ the inner state, $\boldsymbol{e}$ the external input, $\varphi_j(\cdot)$ the bounded activation function of the j-th neuron, $w_{jj'} = \frac{R_j}{R_{jj'}}$ the weight coefficient between the output of the j'-th neuron and the input of the j-th neuron, $w_{j0} = \frac{R_j}{R_{j0}}$ the weight coefficient of the j-th external input and N the number of neurons in the network. $\boldsymbol{v}, \boldsymbol{u}, \boldsymbol{e} \in \mathbb{C}^N$. $\boldsymbol{W}, \boldsymbol{W}_0 \in \mathbb{C}^{N\times N}$. $\mathbf{Y} \in \mathbb{R}^{N\times N}$. $\boldsymbol{W}_0$ and $\mathbf{Y}$ are diagonal matrices.

By comparing Eq. (2.18) with Eq. (2.2) we notice that:

$$\boldsymbol{s}_c(\boldsymbol{u}) = -\boldsymbol{u} + \boldsymbol{W} \cdot \boldsymbol{\varphi}(\boldsymbol{u}) + \boldsymbol{W}_0 \cdot \boldsymbol{e}.$$

Since we restrict ourself to bounded activation functions, the RNN-1 possesses at least one equilibrium point [91].

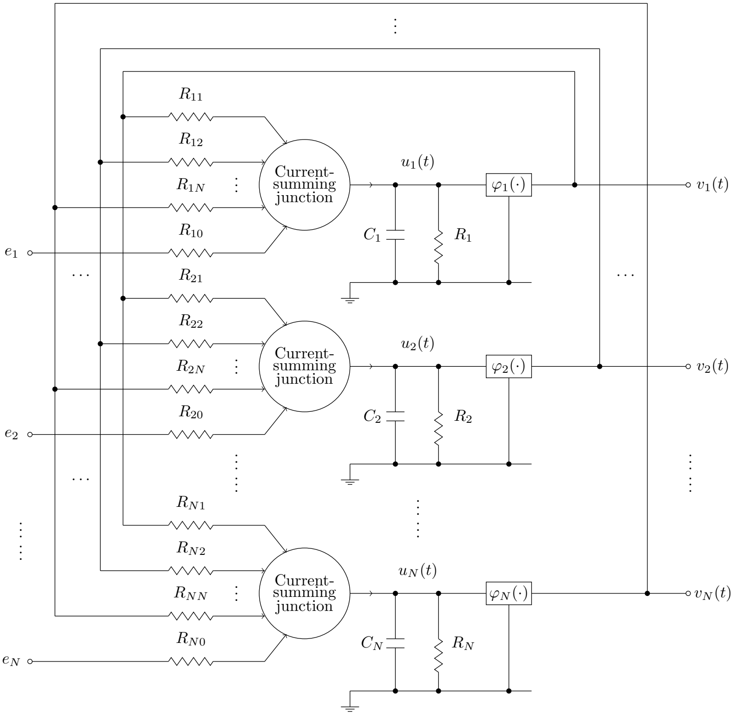


Figure 2.3: Continuous-time recurrent neural network. $\boldsymbol{v}(t)$ is the output, $\boldsymbol{u}(t)$ the inner state, $\boldsymbol{e}$ the external input and $\varphi(\cdot)$ the activation function.

2.2.2 Discrete-Time Recurrent Neural Networks With Serial Update

The dynamical behavior of the discrete-time RNN with serial update, to be abbreviated in the following by RNN-2, assuming the j-th neuron is being updated, is given by:

$$\begin{aligned} u_j(\rho+1) &= \sum_{j'=1}^{N} w_{jj'} \cdot v_{j'}(\rho) + w_{j0} \cdot e_j \\ v_j(\rho) &= \varphi_j[u_j(\rho)]. \end{aligned} \tag{2.19}$$

The parameters in Eq. (2.19) are the same as in Eq. (2.18). Depending on Eq. (2.19), we define the output vector before and after updating the j-th neuron, i.e. the transition from the serial discrete-time index (ρ) to $(\rho+1)$, as follows:

$$\begin{aligned} \boldsymbol{v}(\rho) &= \Big[v_1(\rho), \cdots, v_{j-1}(\rho), v_j(\rho), v_{j+1}(\rho), \cdots, v_N(\rho)\Big]^T \\ \boldsymbol{v}(\rho+1) &= \Big[v_1(\rho), \cdots, v_{j-1}(\rho), v_j(\rho+1), v_{j+1}(\rho), \cdots, v_N(\rho)\Big]^T \end{aligned} \tag{2.20}$$

This means, at each time step (ρ) the output of one neuron is updated, whereas the others stay unchanged:

$$v_{j'}(\rho+1) = v_{j'}(\rho) \quad : \quad j' \neq j$$

It is obvious that the order of the update can play a role when applying the RNN-2 to solve optimization tasks. In addition, N steps are needed to update all neurons one time.

2.2.3 Discrete-Time Recurrent Neural Networks With Parallel Update and Without Inner State Feedback

Fig. 2.4 shows the discrete-time RNN with parallel update and without inner state feedback, to be abbreviated in the following by RNN-3. The dynamical behavior is given by:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \boldsymbol{W} \cdot \boldsymbol{v}(l) + \boldsymbol{W}_0 \cdot \boldsymbol{e} \\ \boldsymbol{v}(l) &= \boldsymbol{\varphi}[\boldsymbol{u}(l)]. \end{aligned} \tag{2.21}$$

The parameters in Eq. (2.21) are the same as in Eq. (2.18).

At this step we notice the difference between serial (asynchronous) and parallel (synchronous) update. In the parallel update mode all neurons are updated at the same time, whereas in the serial update mode only one neuron is updated every time step. Combinations of both update modes are possible but they are out of the scope of this work.

Remark 2.7 Depending on Definition 2.8, RNN-1 and RNN-3 represent twin dynamical systems.

2.2.4 Discrete-Time Recurrent Neural Networks With Parallel Update and Inner State Feedback

In this thesis we focus also on discrete-time RNNs with inner state feedback. Fig. 2.5 shows a RNN with inner state feedback of memory one. This network is abbreviated

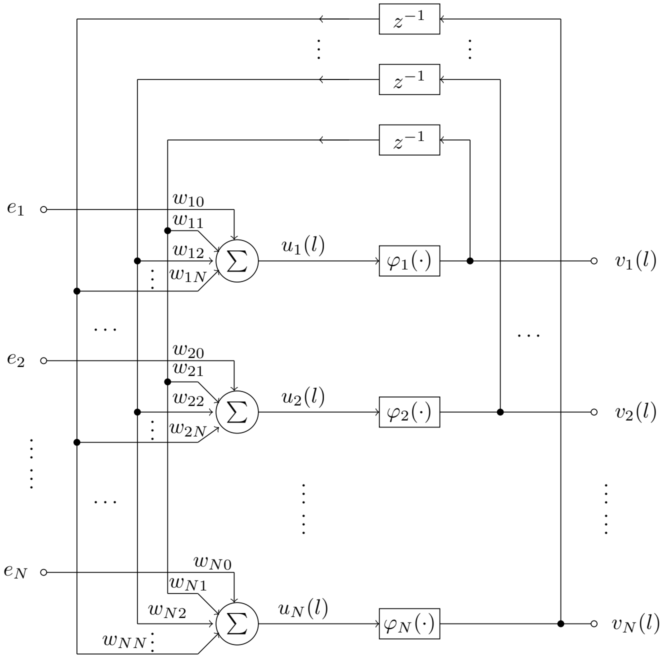


Figure 2.4: Discrete-time recurrent neural network without inner state feedback. $\boldsymbol{v}(l)$ is the output, $\boldsymbol{u}(l)$ the inner state, $\boldsymbol{e}$ the external input and $\varphi(\cdot)$ the activation function.

in the following by RNN-4. The dynamical behavior of RNN-4 is given by:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \boldsymbol{A}\cdot\boldsymbol{u}(l) + \boldsymbol{W}\cdot\boldsymbol{v}(l) + \boldsymbol{W}_0\cdot\boldsymbol{e} \\ \boldsymbol{v}(l) &= \boldsymbol{\varphi}[\boldsymbol{u}(l)]. \end{aligned} \tag{2.22}$$

The parameters in Eq. (2.22) are the same as in Eq. (2.18). In addition, $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{A} = \text{diag}\{a_1, a_2, \cdots, a_N\}$, where $\forall j \in \{1, 2, \cdots, N\} : a_j \in [0, 1)$. a_j is the inner state feedback of the j-th neuron.

If the memory of the inner state feedback is two, we obtain the RNN depicted in Fig. 2.6. This RNN is abbreviated in this work by RNN-5. The dynamical behavior is given by:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \boldsymbol{A}\cdot\boldsymbol{u}(l-1) + \boldsymbol{W}\cdot\boldsymbol{v}(l) + \boldsymbol{W}_0\cdot\boldsymbol{e} \\ \boldsymbol{v}(l) &= \boldsymbol{\varphi}[\boldsymbol{u}(l)]. \end{aligned} \tag{2.23}$$

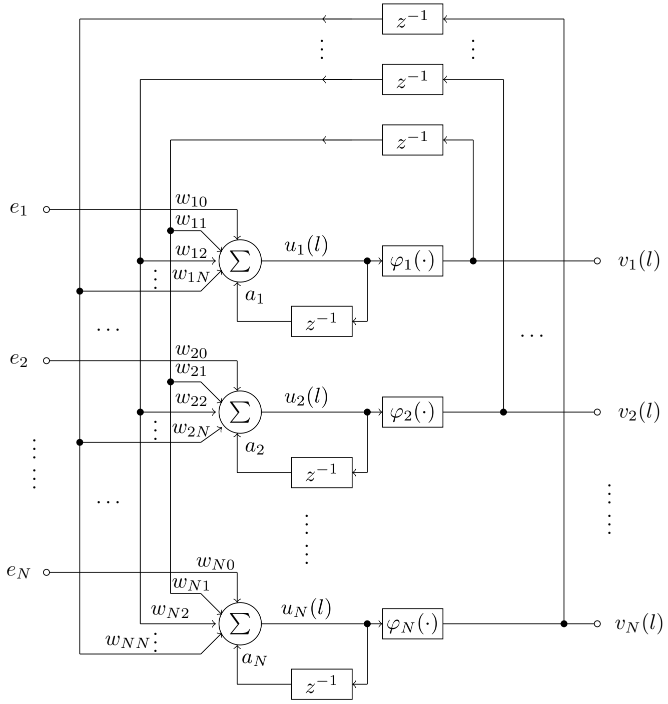


Figure 2.5: Discrete-time recurrent neural network with inner state feedback of memory one. $\boldsymbol{v}(l)$ is the output, $\boldsymbol{u}(l)$ the inner state, $\boldsymbol{e}$ the external input, a the inner state feedback and $\varphi(\cdot)$ the activation function.

We notice that the difference between RNN-4 and RNN-5 is an additional delay in the inner state feedback. The parameters in Eq. (2.23) are the same as in Eq. (2.22).

Remark 2.8 If $A = \mathbf{0}_{N\times N}$, RNN-3, RNN-4 and RNN-5 are identical.

Remark 2.9 As for the RNN-1, it can be proven that the RNN-2, RNN-3, RNN-4 and RNN-5 possess also at least one fixed point [91].

Remark 2.10 It is often assumed that $\varphi_1 = \varphi_2 = \cdots = \varphi_N = \varphi$.

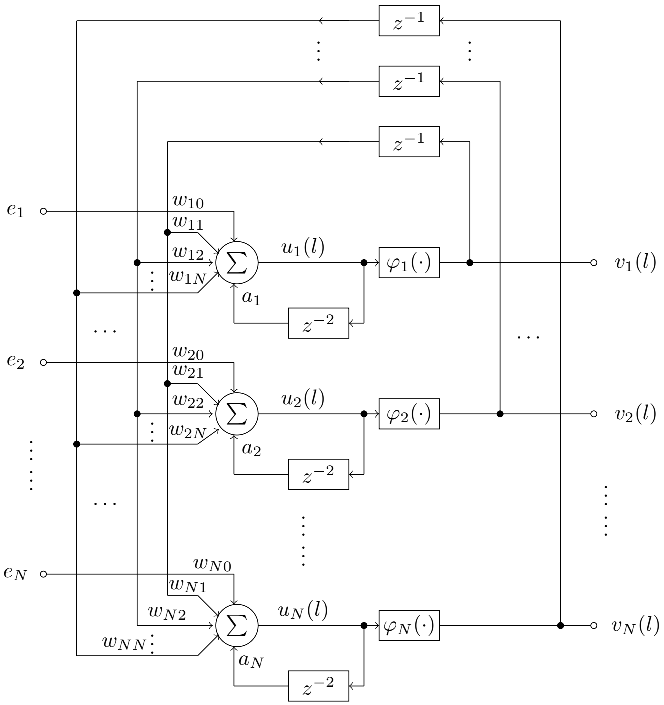


Figure 2.6: Discrete-time recurrent neural network with inner state feedback of memory two. $\boldsymbol{v}(l)$ is the output, $\boldsymbol{u}(l)$ the inner state, $\boldsymbol{e}$ the external input, a the inner state feedback and $\varphi(\cdot)$ the activation function.

2.2.5 Mathematical Preliminaries

In this subsection, Definitions are introduced and many lemmas are proven. This is indispensable for the rest of this chapter.

Definition 2.9 A set of complex-valued functions g is said to be class $g^{(1)}$ if they fulfill the following conditions:

- $g : \mathbb{C} \to \mathcal{B} : \mathcal{B} \subset \mathbb{C}$, $g(u) = g_r(u_r, u_i) + \imath g_i(u_r, u_i)$, $g(0) = 0$, $u = u_r + \imath u_i$
- g are bounded, i.e. there exists a μ such that:

$$|g(u)| \leq \mu < \infty$$

This implies that $g_r(u_r, u_i)$ and $g_i(u_r, u_i)$ are bounded too.

- $g_r(u_r, u_i)$ and $g_i(u_r, u_i)$ are continuously differentiable with respect to u_r and u_i.
- The real and imaginary parts of the functions g are strictly increasing with respect to u_r and u_i, respectively. This means:

$$0 < \frac{\partial g_r(u_r, u_i)}{\partial u_r} \leq \gamma_r \tag{2.24}$$

$$0 < \frac{\partial g_i(u_r, u_i)}{\partial u_i} \leq \gamma_i \tag{2.25}$$

- The determinant of the Jacobian matrix of the functions $g \in g^{(1)}$ is always positive, i.e.

$$\delta_{J_g} = \frac{\partial g_r(u_r, u_i)}{\partial u_r} \cdot \frac{\partial g_i(u_r, u_i)}{\partial u_i} - \frac{\partial g_r(u_r, u_i)}{\partial u_i} \cdot \frac{\partial g_i(u_r, u_i)}{\partial u_r} > 0 \tag{2.26}$$

- In addition:

$$\frac{\partial g_r(u_r, u_i)}{\partial u_i} = \frac{\partial g_i(u_r, u_i)}{\partial u_r} \tag{2.27}$$

Based on the condition given above, the following properties of the inverse function g^{-1} can be derived [76]:

- $g^{-1} : \mathcal{B} \to \mathbb{C}\,, g^{-1}(v) = g_r^{-1}(v_r, v_i) + \imath g_i^{-1}(v_r, v_i), g^{-1}(0) = 0, v = v_r + \imath v_i$
- $g_r^{-1}(v_r, v_i)$ and $g_i^{-1}(v_r, v_i)$ are continuously differentiable with respect to v_r and v_i.
- The real and imaginary parts of the inverse functions g^{-1} are strictly increasing with respect to v_r and v_i, respectively, i.e.

$$0 < \frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_r} \tag{2.28}$$

$$0 < \frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_i} \tag{2.29}$$

- The determinant of the Jacobian matrix of the inverse functions g^{-1} is always positive, i.e.

$$\delta_{J_{g^{-1}}} = \frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_r} \cdot \frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_i} - \frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_i} \cdot \frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_r} > 0 \quad (2.30)$$

- In addition:

$$\frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_i} = \frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_r} \quad (2.31)$$

- The derivatives of $g_r^{-1}(v_r, v_i)$ and $g_i^{-1}(v_r, v_i)$ with respect to v_r and v_i are under bounded, i.e.

$$\min\left\{\frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_r}\right\} = \left[\gamma_r - \frac{\min\left\{\frac{\partial g_r(u_r, u_i)}{\partial u_i} \cdot \frac{\partial g_i(u_r, u_i)}{\partial u_r}\right\}}{\gamma_i}\right]^{-1} \geq \frac{1}{\gamma_r} \quad (2.32)$$

$$\min\left\{\frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_i}\right\} = \left[\gamma_i - \frac{\min\left\{\frac{\partial g_r(u_r, u_i)}{\partial u_i} \cdot \frac{\partial g_i(u_r, u_i)}{\partial u_r}\right\}}{\gamma_r}\right]^{-1} \geq \frac{1}{\gamma_i} \quad (2.33)$$

The proofs of Eq. (2.28)-(2.33) are considered in Lemma 2.1.

Definition 2.10 A set of complex-valued functions g is said to be class $g^{(2)}$ if they fulfill the conditions mentioned in Definition 2.9 and in addition:

$$\frac{\partial g_r(u_r, u_i)}{\partial u_i} = \frac{\partial g_i(u_r, u_i)}{\partial u_r} = 0 \quad \Leftrightarrow \quad \frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_i} = \frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_r} = 0$$

This means $g \in g^{(2)} \Rightarrow g(u) = g_r(u_r) + \imath g_i(u_i)$. We call this property *Separability of real and imaginary parts*.

Definition 2.11 A set of complex-valued functions g is said to be class $g^{(3)}$ if they fulfill the conditions mentioned in Definitions 2.9, 2.10. In addition, if g_1, $g_2 \in g^{(3)}$ and

$$\gamma_{1,r} = \max\left\{\frac{\mathrm{d}g_{1,r}(u_r)}{\mathrm{d}u_r}\right\} \quad , \quad \gamma_{1,i} = \max\left\{\frac{\mathrm{d}g_{1,i}(u_i)}{\mathrm{d}u_i}\right\}$$

$$\gamma_{2,r} = \max\left\{\frac{\mathrm{d}g_{2,r}(u_r)}{\mathrm{d}u_r}\right\} \quad , \quad \gamma_{2,i} = \max\left\{\frac{\mathrm{d}g_{2,i}(u_i)}{\mathrm{d}u_i}\right\}$$

then $\forall v = v_r + \imath v_i \in \mathcal{B}$:

$$\gamma_{2,r} > \gamma_{1,r} \Rightarrow \int_0^{v_r} g_{1,r}^{-1}(\vartheta)\mathrm{d}\vartheta \geq \int_0^{v_r} g_{2,r}^{-1}(\vartheta)\mathrm{d}\vartheta$$
$$\gamma_{2,i} > \gamma_{1,i} \Rightarrow \int_0^{v_i} g_{1,i}^{-1}(\vartheta)\mathrm{d}\vartheta \geq \int_0^{v_i} g_{2,i}^{-1}(\vartheta)\mathrm{d}\vartheta$$

This means: if $\gamma_{2,r} > \gamma_{1,r}$ and $\gamma_{2,i} > \gamma_{1,i}$, it holds:

$$\Re\left\{\int_0^{v} g_2^{-1}(\vartheta)\mathrm{d}\vartheta^* - \int_0^{v} g_1^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \leq 0 \tag{2.34}$$

Remark 2.11 $g^{(3)} \subset g^{(2)} \subset g^{(1)}$.

Definition 2.12 A set of real-valued functions $g : \mathbb{R} \to \mathcal{B}_r : \mathcal{B}_r \subset \mathbb{R}$, $g(0) = 0$ is said to be class $g^{(4)}$ if they fulfill the conditions mentioned in Definition 2.9, 2.10. In addition, it holds $g_i(u_i) = 0$.

Lemma 2.1 If a function $g \in g^{(1)}$, it holds:

$$\min\left\{\frac{\partial g_r^{-1}(v_r, v_i)}{\partial v_r}\right\} = \left[\gamma_r - \frac{\min\left\{\dfrac{\partial g_r(u_r, u_i)}{\partial u_i} \cdot \dfrac{\partial g_i(u_r, u_i)}{\partial u_r}\right\}}{\gamma_i}\right]^{-1} \geq \frac{1}{\gamma_r}$$

$$\min\left\{\frac{\partial g_i^{-1}(v_r, v_i)}{\partial v_i}\right\} = \left[\gamma_i - \frac{\min\left\{\dfrac{\partial g_r(u_r, u_i)}{\partial u_i} \cdot \dfrac{\partial g_i(u_r, u_i)}{\partial u_r}\right\}}{\gamma_r}\right]^{-1} \geq \frac{1}{\gamma_i}$$

Proof
See Appendix A.1.

Lemma 2.2 If a function $g \in g^{(1)}$, there exists a function $\phi(v_r, v_i)$ such that:

$$\frac{\partial \phi(v_r, v_i)}{\partial v_r} = g_r^{-1}(v_r, v_i) \quad \& \quad \frac{\partial \phi(v_r, v_i)}{\partial v_i} = g_i^{-1}(v_r, v_i)$$

where $v \in \mathcal{B}$.

Proof
See Appendix A.2.

Remark 2.12 This Lemma has been mentioned without detailed proof in [41].

Lemma 2.3 If a function $g \in g^{(1)}$, it holds:

$$\Re\left\{\int_{b_1}^{b_2} g^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} = \Re\left\{(b_2 - b_1)^* \cdot g^{-1}(b_0)\right\}$$

where b_1, $b_2 \in \mathcal{B}$. b_0 is unique and lies on the line segment joining the points $(b_{1,r}, b_{1,i})$ and $(b_{2,r}, b_{2,i})$ (intermediate point).

Proof
See Appendix A.3.

Lemma 2.4 If a function $g \in g^{(2)}$ and $\forall v_1, v_2 \in \mathcal{B}$ it holds:

$$\Re\left\{(v_1 - v_2)^* \cdot (u_1 - u_0)\right\} \geq 0$$

given that $v_1 = g(u_1)$ and $v_2 = g(u_2)$. In addition, $u_{0,r}$ is between $u_{1,r}$ and $u_{2,r}$. $u_{0,i}$ is between $u_{1,i}$ and $u_{2,i}$.

Proof
See Appendix A.4.

Lemma 2.5 If a function $g \in g^{(2)}$, it holds:

$$\begin{aligned}\Re\left\{\int_{b_1}^{b_2} g^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \leq \Re\left\{(b_2 - b_1)^* \cdot g^{-1}(b_2)\right\} &- \frac{1}{2}\cdot\gamma_r^{-1}\cdot(b_{2,r} - b_{1,r})^2 \\ &- \frac{1}{2}\cdot\gamma_i^{-1}\cdot(b_{2,i} - b_{1,i})^2\end{aligned}$$

where b_1, $b_2 \in \mathcal{B}$.

Proof
See Appendix A.5.

Lemma 2.6 If a function $g \in g^{(2)}$, it holds:

$$\forall u_1, u_2 \in \mathbb{C} : \Re\Big\{(u_1 - u_2)^* \cdot \Big[g(u_1) - g(u_2)\Big]\Big\} \geq 0 \tag{2.35}$$

Proof
See Appendix A.6.

Lemma 2.7 If $g \in g^{(2)}$, $g_r(\cdot)$ & $g_i(\cdot)$ are Lipschitz functions.

Proof
See Appendix A.7.

Lemma 2.8 If $g \in g^{(2)}$, there exists $\eta > 0$ such that:

$$\forall u_1, u_2 \in \mathbb{C} : \Big|g(u_1) - g(u_2)\Big| \leq \eta \cdot |u_1 - u_2| \tag{2.36}$$

Proof
See Appendix A.8.

Lemma 2.9 $\forall M_r', M_i' \in \mathbb{N}_0$: $u = u_r + \imath u_i$, $\varphi(u) = \varphi_r(u_r) + \imath \varphi_i(u_i)$ such that:

$$\varphi_r(u_r) = \sum_{j=-M_r'}^{M_r'} \tanh\Big[\gamma_r \cdot (u_r - 2 \cdot j)\Big] \tag{2.37}$$

$$\varphi_i(u_i) = \sum_{j=-M_i'}^{M_i'} \tanh\Big[\gamma_i \cdot (u_i - 2 \cdot j)\Big] \tag{2.38}$$

then $\varphi(\cdot) \in g^{(3)}$.

Proof
See Appendix A.9.

Example 2.1 Fig. 2.7–2.11 illustrate the properties of the functions in Eq. (2.37),(2.38) for $M_r' = 1$ and $M_i' = 0$ and different values of γ_r and γ_i. In this case:

$$\varphi_r(u_r) = \tanh\Big[\gamma_r \cdot (u_r - 2)\Big] + \tanh\Big[\gamma_r \cdot u_r\Big] + \tanh\Big[\gamma_r \cdot (u_r + 2)\Big] \tag{2.39}$$

$$\varphi_i(u_i) = \tanh\Big[\gamma_i \cdot u_i\Big] \tag{2.40}$$

We notice also that the shape of these functions does not change substantially if γ_r and γ_i are large enough. In this case Eq. (A.17) tends to zero.

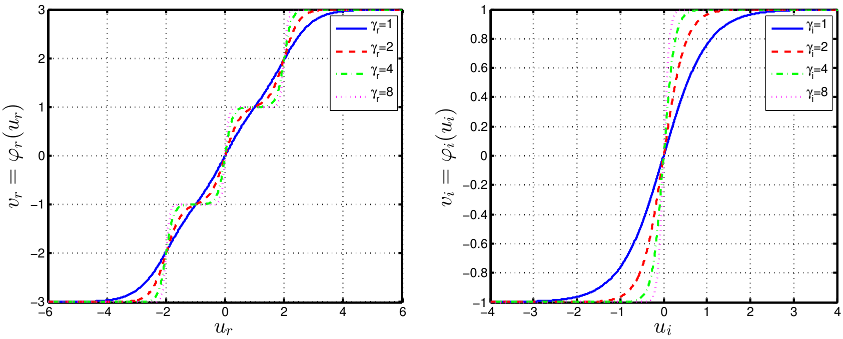
Lemma 2.10 $\forall \boldsymbol{C} \in \mathbb{C}^{N \times N}$ & $\forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{C}^N$ it holds:

$$\Re\Big\{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\Big\} = \begin{bmatrix} \boldsymbol{v}_{1,r}^T & \boldsymbol{v}_{1,i}^T \end{bmatrix} \cdot \left[\begin{array}{c|c} \boldsymbol{C}_r & -\boldsymbol{C}_i \\ \hline \boldsymbol{C}_i & \boldsymbol{C}_r \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{v}_{2,r} \\ \boldsymbol{v}_{2,i} \end{bmatrix}$$

Proof
See Appendix A.10.

Lemma 2.11 $\forall\ \boldsymbol{C} = \boldsymbol{C}^H \in \mathbb{C}^{N \times N}$ & $\forall \boldsymbol{v} \in \mathbb{C}^N\ :\ \Im\Big\{\boldsymbol{v}^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}\Big\} = 0$

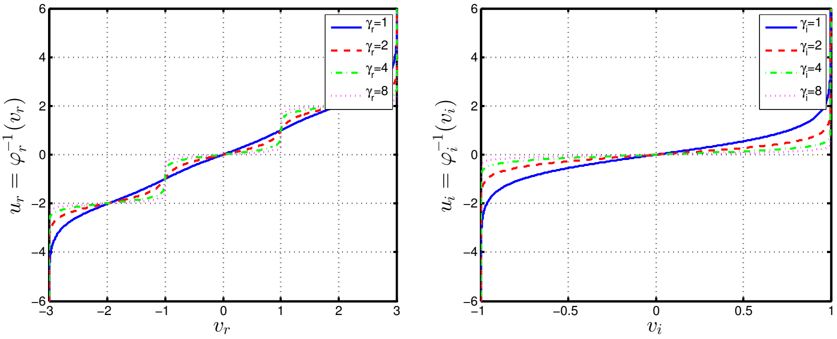
Proof
See Appendix A.11. the absolute value here is elementwise.



(a) The function in Eq. (2.39).

(b) The function in Eq. (2.40).

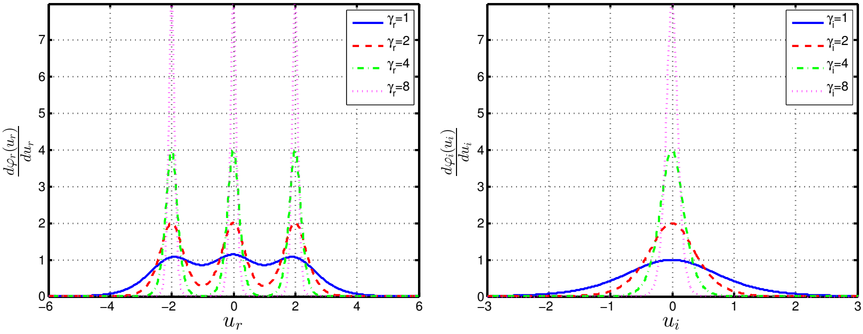
Figure 2.7: The functions in Eq. (2.39), (2.40).



(a) The inverse function of Eq. (2.39).

(b) The inverse function of Eq. (2.40).

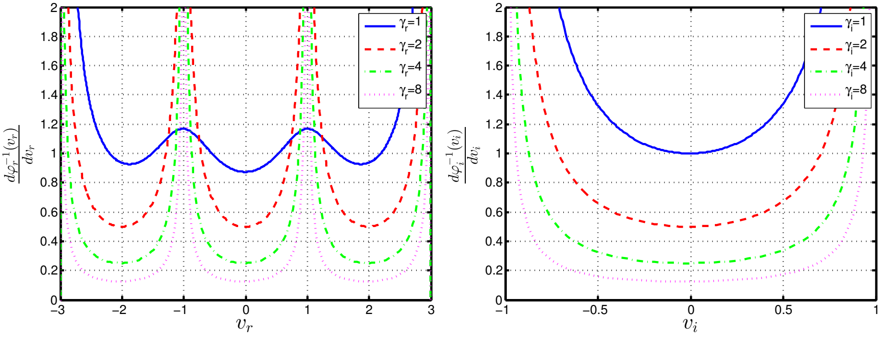
Figure 2.8: The inverse functions of Eq. (2.39), (2.40).



(a) The derivative of the function in Eq. (2.39).

(b) The derivative of the function in Eq. (2.40).

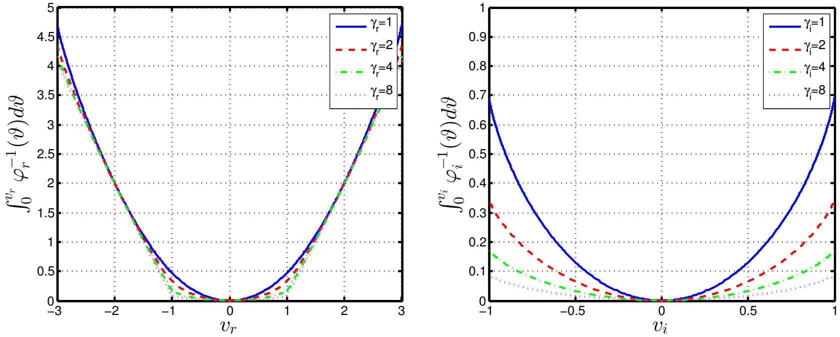
Figure 2.9: The derivatives of the functions in Eq. (2.39), (2.40).



(a) The derivative of the inverse function of Eq. (2.39).

(b) The derivative of the inverse function of Eq. (2.40).

Figure 2.10: The derivatives of the inverse functions of Eq. (2.39), (2.40).



(a) The integration of the inverse function of Eq. (2.39).

(b) The integration of the inverse function of Eq. (2.40).

Figure 2.11: The integration of the inverse function of Eq. (2.39), (2.40).

Lemma 2.12 If a matrix $\boldsymbol{C} = \boldsymbol{C}^H \in \mathbb{C}^{N \times N}$, then $\forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{C}^N$ it holds:

$$-\frac{1}{2} \cdot \{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\} = -\frac{1}{2} \cdot (\boldsymbol{v}_1 - \boldsymbol{v}_2)^H \cdot \boldsymbol{C} \cdot (\boldsymbol{v}_1 - \boldsymbol{v}_2) \\ - \Re\left\{(\boldsymbol{v}_1 - \boldsymbol{v}_2)^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\right\}$$

Proof
See Appendix A.12.

Lemma 2.13

$$\forall \boldsymbol{v}, \boldsymbol{u} \in \mathbb{C}^N : \Re\left\{\boldsymbol{v}^H \cdot \boldsymbol{u}\right\} \leq |\boldsymbol{v}|^T \cdot |\boldsymbol{u}|$$

Proof
See Appendix A.13.

Remark 2.13 Because $g^{(3)} \subset g^{(2)} \subset g^{(1)}$:

- All Lemmas proven for functions belonging to $g^{(1)}$ are applicable for functions belonging to $g^{(2)}$ and $g^{(3)}$ but not vice versa.
- All Lemmas proven for functions belonging to $g^{(2)}$ are applicable for functions belonging to $g^{(3)}$ but not vice versa.

Remark 2.14 We notice that Lemmas 2.3-2.8 are applicable to functions belonging to $g^{(4)}$ too.

2.2.6 Local Asymptotical Stability of Recurrent Neural Networks

In this section, we investigate the local asymptotical stability (LAS) of the different RNNs presented before. We always follow the same approach: For each RNN, we introduce a function and derive specific conditions that guarantee that the function we have given is a Lyapunov function, i.e. it is strictly (monotonically) decreasing with time along the dynamics of the discussed RNN. Later on, we introduce time-variant activation functions and discuss the LAS of RNNs with this kind of activation function.

Locally asymptotically stable RNNs are desirable for some applications. Associative memory design based on RNNs is a natural and well investigated example. Another application is the vector equalization based on RNNs, which is considered in details in the next chapter.

In case of variables (functions) with double subscripts separated by a comma, the right one refers to the real/imaginary part of the variable (function) and the left one refers to the neuron index. For the ease of notation, we drop the dependency of variables and functions on time, if possible. Parts of this section have been published in [66].

Theorem 2.1 The RNN-1, cf. Fig. 2.3 and Eq. (2.18), reaches an equilibrium point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(1)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \operatorname{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.

Proof
Under theses assumptions, RNN-1 possesses the following Lyapunov function:

$$\begin{aligned} E[\boldsymbol{v}(t)] = &- \frac{1}{2} \cdot \boldsymbol{v}^H(t) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(t) - \Re\left\{\boldsymbol{v}^H(t) \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} \\ &+ \sum_{j=1}^{N} d_j \cdot \phi_j\left[v_{j,r}(t), v_{j,i}(t)\right] \end{aligned} \tag{2.41}$$

$$\phi_j(v_{j,r}, v_{j,i}) = \int_0^{v_{j,r}} \varphi_{j,r}^{-1}(\vartheta, 0)\mathrm{d}\vartheta + \int_0^{v_{j,i}} \varphi_{j,i}^{-1}(v_{j,r}, \vartheta)\mathrm{d}\vartheta$$

Using Lemma 2.2

$$\frac{\mathrm{d}\phi_j(v_{j,r}, v_{j,i})}{\mathrm{d}t} = u_{j,r} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} + u_{j,i} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} = \Re\left\{\frac{\mathrm{d}v_j^*}{\mathrm{d}t} \cdot u_j\right\}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\sum_{j=1}^{N} d_j \cdot \phi_j(v_{j,r}, v_{j,i})\right] = \Re\left\{\frac{\mathrm{d}\boldsymbol{v}^H}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \boldsymbol{u}\right\}$$

$$\begin{aligned}
\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} &= \overbrace{-\frac{1}{2} \cdot \frac{\mathrm{d}\boldsymbol{v}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(t) - \frac{1}{2} \cdot \boldsymbol{v}^H(t) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \frac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t}}^{-\Re\left\{\frac{\mathrm{d}\boldsymbol{v}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(t)\right\}} \\
&\quad - \Re\left\{\frac{\mathrm{d}\boldsymbol{v}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} + \Re\left\{\frac{\mathrm{d}\boldsymbol{v}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \boldsymbol{u}(t)\right\} \\
&= -\Re\left\{\frac{\mathrm{d}\boldsymbol{v}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \Big[-\boldsymbol{u}(t) + \boldsymbol{W} \cdot \boldsymbol{v}(t) + \boldsymbol{W}_0 \cdot \boldsymbol{e}\Big]\right\}
\end{aligned}$$

From Eq. (2.18):

$$\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} = -\Re\left\{\frac{\mathrm{d}\boldsymbol{v}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{D} \cdot \mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t}\right\} = -\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \left\{\frac{\mathrm{d}u_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} + \frac{\mathrm{d}u_{j,i}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t}\right\}$$

But $\forall j \in \{1, 2, \ldots, N\}$:

$$\begin{aligned}
\frac{\mathrm{d}u_{j,r}}{\mathrm{d}t} &= \frac{\partial \varphi_{j,r}^{-1}(v_{j,r}, v_{j,i})}{\partial v_{j,r}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} + \frac{\partial \varphi_{j,r}^{-1}(v_{j,r}, v_{j,i})}{\partial v_{j,i}} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} \\
\frac{\mathrm{d}u_{j,i}}{\mathrm{d}t} &= \frac{\partial \varphi_{j,i}^{-1}(v_{j,r}, v_{j,i})}{\partial v_{j,r}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} + \frac{\partial \varphi_{j,i}^{-1}(v_{j,r}, v_{j,i})}{\partial v_{j,i}} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t}
\end{aligned}$$

For clarity of presentation, we suppress the dependency of $\varphi_{j,r}^{-1}(v_{j,r}, v_{j,i})$ and $\varphi_{j,i}^{-1}(v_{j,r}, v_{j,i})$ on $v_{j,r}$,$v_{j,i}$:

$$\begin{aligned}
\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} = -\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \Bigg\{ & \frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,r}} \cdot \left(\frac{\mathrm{d}v_{j,r}}{\mathrm{d}t}\right)^2 + \frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,i}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} \\
& + \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} + \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \left(\frac{\mathrm{d}v_{j,i}}{\mathrm{d}t}\right)^2 \Bigg\}
\end{aligned} \tag{2.42}$$

Now we define:

$$\begin{aligned}\tilde{E}[\boldsymbol{v}(t)] = &-\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \left\{ \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} + \frac{\dfrac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}}}{\dfrac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \right\}^2 \\ &-\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \frac{\dfrac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,r}} \cdot \dfrac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} - \left(\dfrac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}} \right)^2}{\dfrac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}}} \cdot \left(\frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \right)^2\end{aligned} \tag{2.43}$$

Using Definition 2.9 and Eq. (2.28)-(2.31): $\tilde{E}[\boldsymbol{v}(t)] \leq 0$ and the equality holds if and only if $\dfrac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t} = \mathbf{0}_N$. Depending on Appendix A.14:

$$\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} = \tilde{E}[\boldsymbol{v}(t)] \leq 0 \tag{2.44}$$

The equality holds if and only if $\dfrac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t} = \mathbf{0}_N$. This means an equilibrium point has been reached. Otherwise, the function in Eq. (2.41) is strictly decreasing with the time along the dynamics in Eq. (2.18). ■

Remark 2.15 The original proof in [41] assumes $\boldsymbol{D} = \boldsymbol{I}$ and $\boldsymbol{W} = \boldsymbol{W}^H$, which is a special case of the proof presented above. We will see later, cf. Eq. (3.26) that the assumption of $\boldsymbol{D} \neq \boldsymbol{I}$ is very useful when using RNN-1 as vector equalizer.

Remark 2.16 If the imaginary parts of all variables and activation functions in Eq. (2.18) are zeros and $\varphi \in g^{(4)}$ (sigmoid input-output relation), we obtain the well known continuous-time Hopfield network. The LAS in this case, where $\boldsymbol{D} = \boldsymbol{I}$ & $\boldsymbol{W} = \boldsymbol{W}^T$ has been proven in [30].

Theorem 2.2 The RNN-2, cf. Eq. (2.19),(2.20), reaches a fixed point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(2)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.

iii) The diagonal elements of $\boldsymbol{W}$ are nonnegative.

Proof
Under these assumptions, RNN-2 possesses the following Lyapunov function:

$$\begin{aligned} E[\boldsymbol{v}(\rho)] = & -\frac{1}{2} \cdot \boldsymbol{v}^H(\rho) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(\rho) - \Re\left\{\boldsymbol{v}^H(\rho) \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} \\ & + \Re\left\{\sum_{j=1}^{N} d_j \cdot \int_0^{v_j(\rho)} \varphi_j^{-1}(\vartheta) \mathrm{d}\vartheta^*\right\} \end{aligned} \tag{2.45}$$

Assuming that the j-th neuron has been updated, i.e.

$$\forall j' \in \{1, 2, \cdots, N\} \quad : \quad v_{j'}(\rho+1) = v_{j'}(\rho) \quad : \quad j' \neq j$$

we define $\Delta E_j = E[\boldsymbol{v}(\rho+1)] - E[\boldsymbol{v}(\rho)]$.
From Appendix A.15:

$$\begin{aligned} \Delta E_j = & - d_j \cdot \Re\left\{\left[v_j^*(\rho+1) - v_j^*(\rho)\right] \cdot u_j(\rho+1) - \int_{v_j(\rho)}^{v_j(\rho+1)} \varphi_j^{-1}(\vartheta) \mathrm{d}\vartheta^*\right\} \\ & - \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left|v_j(\rho+1) - v_j(\rho)\right|^2 \end{aligned} \tag{2.46}$$

Using Lemma 2.3:

$$\begin{aligned} & \Delta E_j = -d_j \cdot \Re\left\{\left[v_j(\rho+1) - v_j(\rho)\right]^* \cdot \left[u_j(\rho+1) - u_0\right]\right\} - \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left|v_j(\rho+1) - v_j(\rho)\right|^2 \\ & u_{0,r} \in \Big(\min\{u_{j,r}(\rho), u_{j,r}(\rho+1)\}, \max\{u_{j,r}(\rho), u_{j,r}(\rho+1)\}\Big) \\ & u_{0,i} \in \Big(\min\{u_{j,i}(\rho), u_{j,i}(\rho+1)\}, \max\{u_{j,i}(\rho), u_{j,i}(\rho+1)\}\Big) \end{aligned}$$

Depending on Lemma 2.4, we conclude that $\Delta E_j \leq 0$.

The equality holds if and only if $v_j(\rho+1) = v_j(\rho)$. The network will reach a fixed point if $\forall j \in \{1, 2, \ldots, N\} : \Delta E_j = 0$. ■

Remark 2.17 Because we are aiming to generalize the stability results from the real-valued case to the complex-valued one, the LAS of the RNN-2 can be proven in another way based on Lemma 2.5 as in the following theorem.

Theorem 2.3 The RNN-2, cf. Eq. (2.19),(2.20), reaches a fixed point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(2)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \operatorname{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.

iii) The diagonal elements of $\boldsymbol{W}$ are nonnegative.

Proof
We assume that the RNN-2 possesses the same Lyapunov function as in Theorem 2.2. From Eq. (2.46):

$$\begin{aligned}\Delta E_j = &- d_j \cdot \Re\left\{\left[v_j^*(\rho+1) - v_j^*(\rho)\right] \cdot u_j(\rho+1) - \int_{v_j(\rho)}^{v_j(\rho+1)} \varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \\ &- \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left|v_j(\rho+1) - v_j(\rho)\right|^2\end{aligned}$$

Using Lemma 2.5:

$$\begin{aligned}\Delta E_j \leq &- \frac{1}{2} \cdot d_j \cdot \gamma_{j,r}^{-1} \cdot [v_{j,r}(\rho+1) - v_{j,r}(\rho)]^2 - \frac{1}{2} \cdot d_j \cdot \gamma_{j,i}^{-1} \cdot [v_{j,i}(\rho+1) - v_{j,i}(\rho)]^2 \\ &- \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left|v_j(\rho+1) - v_j(\rho)\right|^2 \\ \Delta E_j \leq &\ 0\end{aligned}$$

We have shown in Theorem 2.2 that the equality holds if and only if $v_j(\rho+1) = v_j(\rho)$. The network will reach a fixed point if $\forall j \in \{1, 2, \ldots, N\} : \Delta E_j = 0$. ∎

Remark 2.18 The LAS of RNN-2 with different activation functions than $g^{(2)}$ has already been investigated. We mention in the list below some of them. However, we notice that the stability conditions remains the same, regardless of the activation function.

- If the imaginary parts of all variables and activation functions in Eq. (2.19) are zeros and the network is composed of two-state neurons, we obtain the discrete-time Hopfield network (serial update), its LAS (without external inputs $\boldsymbol{e} = \boldsymbol{0}_N$) has been proven in [29].

- If the imaginary parts of all variables and activation functions in Eq. (2.19) are zeros and $\varphi(u)$ is a multithreshold function, the RNN-2 in this case can be transformed into a discrete-time Hopfield network with larger number of neurons. Thus, the LAS results for the discrete-time Hopfield networks are applicable to the multithreshold RNNs too, cf. [10] and the references therein.
- If the imaginary parts of all variables and activation functions in Eq. (2.19) are zeros and $\varphi(u)$ is strictly increasing within an interval and constant outside it, the LAS has been proven in [14].
- If the imaginary parts of all variables and activation functions in Eq. (2.19) are zeros and $\varphi(u)$ is strictly increasing, the LAS has been proven in [10].
- If $\varphi(u)$ is a complex-valued, multistate function, the LAS (without external inputs $\boldsymbol{e} = \boldsymbol{0}_N$) in this case has been proven in [33],[43]. The stability conditions have been relaxed in [42],[43].
- In the phasor model RNNs, where the outputs of the network lie on the unite circle in the complex plane, the LAS (without external inputs $\boldsymbol{e} = \boldsymbol{0}_N$) has been proven in [56].

For complex-valued RNNs (the last two cases mentioned above) we notice that the activation function is noninvertible.

Theorem 2.4 The RNN-4, cf. Fig. 2.5 and Eq. (2.22), reaches a fixed point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(2)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \operatorname{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$ and $\boldsymbol{D} \cdot \boldsymbol{W} \geq 0$.

iii) $\boldsymbol{A} = \operatorname{diag}\{a_1, a_2, \ldots, a_N\} \in \mathbb{R}^{N \times N}$. $\forall j \in \{1, 2, \ldots, N\} : a_j \in [0, 1)$, i.e. $\boldsymbol{A}$ is a diagonal and positive semidefinite matrix.

Proof

Under theses assumptions, RNN-4 possesses the following Lyapunov function:

$$\begin{aligned} E[\boldsymbol{v}(l)] = &-\frac{1}{2} \cdot \boldsymbol{v}^H(l) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(l) - \Re\left\{\boldsymbol{v}^H(l) \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} \\ &+ \Re\left\{\sum_{j=1}^{N} d_j \cdot (1 - a_j) \cdot \int_0^{v_j(l)} \varphi_j^{-1}(\vartheta) \mathrm{d}\vartheta^*\right\} \end{aligned} \tag{2.47}$$

We define in this case:

$$\Delta\boldsymbol{v} = \boldsymbol{v}(l+1) - \boldsymbol{v}(l) \quad , \quad \Delta E = E[\boldsymbol{v}(l+1)] - E[\boldsymbol{v}(l)]$$

$$\begin{aligned}\Delta E = &-\frac{1}{2}\cdot\left\{\boldsymbol{v}^H(l+1)\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\boldsymbol{v}(l+1) - \boldsymbol{v}^H(l)\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\boldsymbol{v}(l)\right\}\\ &-\Re\left\{\left[\boldsymbol{v}^H(l+1) - \boldsymbol{v}^H(l)\right]\cdot\boldsymbol{D}\cdot\boldsymbol{W}_0\cdot\boldsymbol{e}\right\}\\ &+\Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_{v_j(l)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\end{aligned}$$

Using Lemma 2.12:

$$\begin{aligned}\Delta E = &-\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v} - \Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\boldsymbol{v}(l)\right\} - \Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}_0\cdot\boldsymbol{e}\right\}\\ &+\Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_{v_j(l)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\\ = &-\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v} - \Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{W}\cdot\boldsymbol{v}(l) + \boldsymbol{W}_0\cdot\boldsymbol{e}]\right\}\\ &+\Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_{v_j(l)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\end{aligned}$$

Using Eq. (2.22):

$$\begin{aligned}\Delta E = &-\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v} - \Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{u}(l+1) - \boldsymbol{A}\cdot\boldsymbol{u}(l)]\right\}\\ &+\Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_{v_j(l)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\end{aligned} \tag{2.48}$$

Using Lemma 2.3:

$$\begin{aligned}\Delta E = &-\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v} - \Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{u}(l+1) - \boldsymbol{A}\cdot\boldsymbol{u}(l)]\right\}\\ &+\Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]\cdot\boldsymbol{u}_0\right\}\end{aligned}$$

$\boldsymbol{u}_0$ is between $\boldsymbol{u}(l)$ and $\boldsymbol{u}(l+1)$, i.e. $\boldsymbol{u}_{0,r}$ is between $\boldsymbol{u}_r(l)$ and $\boldsymbol{u}_r(l+1)$. $\boldsymbol{u}_{0,i}$ is between $\boldsymbol{u}_i(l)$ and $\boldsymbol{u}_i(l+1)$.

$$\begin{aligned}\Delta E = &-\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v}\\ &-\Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{u}(l+1)-\boldsymbol{u}_0]\right\}-\Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{A}\cdot[\boldsymbol{u}_0-\boldsymbol{u}(l)]\right\}\end{aligned}$$

Based on Lemma 2.4, $\boldsymbol{D} > 0$ and $\boldsymbol{D}\cdot\boldsymbol{A} \geq 0$, we conclude that all three terms in the last relation are nonpositive and added constructively, i.e. $\Delta E \leq 0$. The equality holds if and only if $\Delta\boldsymbol{v} = \boldsymbol{0}_N$ i.e. $\boldsymbol{v}(l+1) = \boldsymbol{v}(l)$. This means a fixed point has been reached.■

Remark 2.19 To relax the positive semidefinite condition in the last theorem, the LAS of the RNN-4 can be proved in another way based on Lemma 2.5 as it is illustrated in the following theorem. In this case, there is no necessity that $\boldsymbol{D}\cdot\boldsymbol{W} \geq 0$. However, a similar condition must be fulfilled. This can be done by controlling the inner state feedback $\boldsymbol{A}$ and/or the activation function.

Theorem 2.5 The RNN-4, cf. Fig. 2.5 and Eq. (2.22), reaches a fixed point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(2)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D}\cdot\boldsymbol{W} = \{\boldsymbol{D}\cdot\boldsymbol{W}\}^H$ and

$$\tilde{\boldsymbol{W}} = \left[\begin{array}{c|c}\boldsymbol{D}\cdot\boldsymbol{W}_r+\boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_r^{-1} & -\boldsymbol{D}\cdot\boldsymbol{W}_i\\ \hline \boldsymbol{D}\cdot\boldsymbol{W}_i & \boldsymbol{D}\cdot\boldsymbol{W}_r+\boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_i^{-1}\end{array}\right] > 0$$

where

$$\boldsymbol{\Gamma}_r = \text{diag}\{\gamma_{1,r}, \gamma_{2,r}, \ldots, \gamma_{N,r}\} \quad , \quad \boldsymbol{\Gamma}_i = \text{diag}\{\gamma_{1,i}, \gamma_{2,i}, \ldots, \gamma_{N,i}\}$$

iii) $\boldsymbol{A} = \text{diag}\{a_1, a_2, \ldots, a_N\} \in \mathbb{R}^{N\times N}$. $\forall j \in \{1, 2, \ldots, N\} : a_j \in [0, 1)$, i.e. $\boldsymbol{A}$ is a diagonal and positive semidefinite matrix.

Proof

We assume that the RNN-4 possesses in this case the same Lyapunov function as in Theorem 2.4. We define in this case:

$$\Delta\boldsymbol{v} = \boldsymbol{v}(l+1)-\boldsymbol{v}(l) \quad , \quad \Delta\boldsymbol{u} = \boldsymbol{u}(l+1)-\boldsymbol{u}(l) \quad , \quad \Delta E = E[\boldsymbol{v}(l+1)]-E[\boldsymbol{v}(l)]$$

From Eq. (2.48):

$$\begin{aligned}\Delta E = & -\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v}-\Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{u}(l+1)-\boldsymbol{A}\cdot\boldsymbol{u}(l)]\right\}\\ & +\Re\left\{\sum_{j=1}^{N}d_j\cdot(1-a_j)\cdot\int_{v_j(l)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\end{aligned}$$

Using Lemma 2.5 considering that $\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]$ is diagonal and positive definite:

$$\begin{aligned}\Re\left\{\sum_{j=1}^{N}d_j\cdot(1-a_j)\cdot\int_{v_j(l)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\leq & \Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]\cdot\boldsymbol{u}(l+1)\right\}\\ & -\frac{1}{2}\cdot\Delta\boldsymbol{v}_r^T\cdot\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_r^{-1}\cdot\Delta\boldsymbol{v}_r\\ & -\frac{1}{2}\cdot\Delta\boldsymbol{v}_i^T\cdot\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_i^{-1}\cdot\Delta\boldsymbol{v}_i\end{aligned}$$

In this case:

$$\begin{aligned}\Delta E\leq & -\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v}-\Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{A}\cdot\Delta\boldsymbol{u}\right\}\\ & -\frac{1}{2}\cdot\Delta\boldsymbol{v}_r^T\cdot\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_r^{-1}\cdot\Delta v_r-\frac{1}{2}\cdot\Delta\boldsymbol{v}_i^T\cdot\boldsymbol{D}\cdot[\boldsymbol{I}-\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_i^{-1}\cdot\Delta\boldsymbol{v}_i\end{aligned}$$

According to Lemmas 2.7,2.10 taking into account that $\boldsymbol{D}\cdot\boldsymbol{A}\geq 0$:

$$\begin{aligned}\Delta E\leq & -\frac{1}{2}\cdot\Delta\boldsymbol{v}_r^T\cdot\boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_r^{-1}\cdot\Delta v_r-\frac{1}{2}\cdot\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v}\\ & -\frac{1}{2}\cdot\Delta\boldsymbol{v}_i^T\cdot\boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_i^{-1}\cdot\Delta\boldsymbol{v}_i\\ \leq & -\frac{1}{2}\cdot\left[\begin{array}{cc}\Delta\boldsymbol{v}_r^T & \Delta\boldsymbol{v}_i^T\end{array}\right]\cdot\tilde{\boldsymbol{W}}\cdot\left[\begin{array}{c}\Delta\boldsymbol{v}_r\\ \Delta\boldsymbol{v}_i\end{array}\right]\\ \Delta E\leq & \ \ 0\end{aligned}\tag{2.49}$$

From Theorem 2.4, the equality holds if and only if $\Delta\boldsymbol{v}=\boldsymbol{0}_N$, i.e. $\boldsymbol{v}(l+1)=\boldsymbol{v}(l)$. This implies reaching a fixed point. ∎

Discussion

- For complex-valued RNN-4 with complex-valued weight matrix: $\tilde{\boldsymbol{W}}>0$.

- For complex-valued RNN-4 with real-valued weight matrix, i.e. $\boldsymbol{W} = \boldsymbol{W}_r$, $\boldsymbol{W}_i = \boldsymbol{0}_{N\times N}$, and $\boldsymbol{D}\cdot\boldsymbol{W} = \{\boldsymbol{D}\cdot\boldsymbol{W}\}^T$:

$$\boldsymbol{D}\cdot\boldsymbol{W} + \boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_r^{-1} > 0$$
$$\boldsymbol{D}\cdot\boldsymbol{W} + \boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}_i^{-1} > 0$$

- For real-valued RNN-4, $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_r$:

$$\boldsymbol{D}\cdot\boldsymbol{W} + \boldsymbol{D}\cdot[\boldsymbol{I}+\boldsymbol{A}]\cdot\boldsymbol{\Gamma}^{-1} > 0$$

This generalizes the results in [52],[92].

Theorem 2.6 The RNN-5, cf. Fig. 2.6 and Eq. (2.23), reaches a fixed point or a limit cycle of length two if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(2)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D}\cdot\boldsymbol{W} = \{\boldsymbol{D}\cdot\boldsymbol{W}\}^H$.

iii) $\boldsymbol{A} = \mathrm{diag}\{a_1, a_2, \ldots, a_N\} \in \mathbb{R}^{N\times N}$. $\forall j \in \{1, 2, \ldots, N\} : a_j \in [0, 1)$, i.e. $\boldsymbol{A}$ is a diagonal and positive semidefinite matrix.

Proof
Under these assumptions, RNN-5 possesses the following Lyapunov function:

$$\begin{aligned} E[\boldsymbol{v}(l)] = &- \Re\left\{\boldsymbol{v}^H(l)\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\boldsymbol{v}(l-1)\right\} - \Re\left\{\left[\boldsymbol{v}^H(l)+\boldsymbol{v}^H(l-1)\right]\cdot\boldsymbol{D}\cdot\boldsymbol{W}_0\cdot\boldsymbol{e}\right\} \\ &+ \Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_0^{v_j(l)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \\ &+ \Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_0^{v_j(l-1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \end{aligned} \tag{2.50}$$

We define in this case:

$$\Delta\boldsymbol{v} = \boldsymbol{v}(l+1) - \boldsymbol{v}(l-1) \quad , \quad \Delta E = E[\boldsymbol{v}(l+1)] - E[\boldsymbol{v}(l)]$$

$$\Delta E = -\Re\left\{\Delta\boldsymbol{v}^H\cdot\boldsymbol{D}\cdot[\boldsymbol{W}\cdot\boldsymbol{v}(l)+\boldsymbol{W}_0\cdot\boldsymbol{e}]\right\} + \Re\left\{\sum_{j=1}^{N} d_j\cdot(1-a_j)\cdot\int_{v_j(l-1)}^{v_j(l+1)}\varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}$$

Using Eq. (2.23):

$$\begin{aligned}\Delta E = &- \Re\left\{\Delta \boldsymbol{v}^H \cdot \boldsymbol{D} \cdot [\boldsymbol{u}(l+1) - \boldsymbol{A} \cdot \boldsymbol{u}(l-1)]\right\} \\ &+ \Re\left\{\sum_{j=1}^{N} d_j \cdot (1-a_j) \cdot \int_{v_j(l-1)}^{v_j(l+1)} \varphi_j^{-1}(\vartheta) \mathrm{d}\vartheta^*\right\}\end{aligned} \tag{2.51}$$

Using Lemma 2.3:

$$\Delta E = -\Re\left\{\Delta \boldsymbol{v}^H \cdot \boldsymbol{D} \cdot [\boldsymbol{u}(l+1) - \boldsymbol{u}_0]\right\} - \Re\left\{\Delta \boldsymbol{v}^H \cdot \boldsymbol{D} \cdot \boldsymbol{A} \cdot [\boldsymbol{u}_0 - \boldsymbol{u}(l-1)]\right\}$$

$\boldsymbol{u}_0$ is between $\boldsymbol{u}(l-1)$ and $\boldsymbol{u}(l+1)$. Based on Lemma 2.4, we conclude that $\Delta E \leq 0$. Equality holds if and only if $\Delta \boldsymbol{v} = \boldsymbol{0}_N$. This occurs in two cases:

- Limit cycle of length two: $\boldsymbol{v}(l+1) = \boldsymbol{v}(l-1) \neq \boldsymbol{v}(l)$.
- Fixed point: $\boldsymbol{v}(l+1) = \boldsymbol{v}(l) = \boldsymbol{v}(l-1)$.

■

The LAS of the RNN-5 can be proved in another way based on Lemma 2.5 as in the following theorem.

Theorem 2.7 The RNN-5, cf. Fig. 2.6 and Eq. (2.23), reaches a fixed point or a limit cycle of length two if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(2)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.

iii) $\boldsymbol{A} = \mathrm{diag}\{a_1, a_2, \ldots, a_N\} \in \mathbb{R}^{N \times N}$. $\forall j \in \{1, 2, \ldots, N\} : a_j \in [0, 1)$, i.e. $\boldsymbol{A}$ is a diagonal and positive semidefinite matrix.

Proof
We assume that the RNN-5 possesses in this case the same Lyapunov function as in Theorem 2.6 and we define:

$$\begin{aligned} \Delta \boldsymbol{v} &= \boldsymbol{v}(l+1) - \boldsymbol{v}(l-1) & \boldsymbol{\Gamma}_r &= \mathrm{diag}\{\gamma_{1,r}, \gamma_{2,r}, \ldots, \gamma_{N,r}\} \\ \Delta \boldsymbol{u} &= \boldsymbol{u}(l+1) - \boldsymbol{u}(l-1) & \boldsymbol{\Gamma}_i &= \mathrm{diag}\{\gamma_{1,i}, \gamma_{2,i}, \ldots, \gamma_{N,i}\} \\ \Delta E &= E[\boldsymbol{v}(l+1)] - E[\boldsymbol{v}(l)] \end{aligned}$$

From Eq. (2.51):

$$\begin{aligned}\Delta E = &- \Re\left\{\Delta\boldsymbol{v}^H \cdot \boldsymbol{D} \cdot [\boldsymbol{u}(l+1) - \boldsymbol{A} \cdot \boldsymbol{u}(l-1)]\right\} \\ &+ \Re\left\{\sum_{j=1}^{N} d_j \cdot (1-a_j) \cdot \int_{v_j(l-1)}^{v_j(l+1)} \varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\}\end{aligned}$$

Using Lemma 2.5:

$$\begin{aligned}\Delta E \leq &- \Re\left\{\Delta\boldsymbol{v}^H \cdot \boldsymbol{D} \cdot [\boldsymbol{u}(l+1) - \boldsymbol{A} \cdot \boldsymbol{u}(l-1)]\right\} + \Re\left\{\Delta\boldsymbol{v}^H \cdot \boldsymbol{D} \cdot [\boldsymbol{I} - \boldsymbol{A}] \cdot \boldsymbol{u}(l+1)\right\} \\ &- \frac{1}{2} \cdot \Delta\boldsymbol{v}_r^T \cdot \boldsymbol{D} \cdot [\boldsymbol{I} - \boldsymbol{A}] \cdot \boldsymbol{\Gamma}_r^{-1} \cdot \Delta\boldsymbol{v}_r - \frac{1}{2} \cdot \Delta\boldsymbol{v}_i^T \cdot \boldsymbol{D} \cdot [\boldsymbol{I} - \boldsymbol{A}] \cdot \boldsymbol{\Gamma}_i^{-1} \cdot \Delta\boldsymbol{v}_i \\ \Delta E \leq &- \Re\left\{\Delta\boldsymbol{v}^H \cdot \boldsymbol{D} \cdot \boldsymbol{A} \cdot \Delta\boldsymbol{u}\right\} \\ &- \frac{1}{2} \cdot \Delta\boldsymbol{v}_r^T \cdot \boldsymbol{D} \cdot [\boldsymbol{I} - \boldsymbol{A}] \cdot \boldsymbol{\Gamma}_r^{-1} \cdot \Delta\boldsymbol{v}_r - \frac{1}{2} \cdot \Delta\boldsymbol{v}_i^T \cdot \boldsymbol{D} \cdot [\boldsymbol{I} - \boldsymbol{A}] \cdot \boldsymbol{\Gamma}_i^{-1} \cdot \Delta\boldsymbol{v}_i\end{aligned}$$

Using Lemma 2.7:

$$\begin{aligned}\Delta E \leq &- \frac{1}{2} \cdot \Delta\boldsymbol{v}_r^T \cdot \boldsymbol{D} \cdot [\boldsymbol{I} + \boldsymbol{A}] \cdot \boldsymbol{\Gamma}_r^{-1} \cdot \Delta\boldsymbol{v}_r - \frac{1}{2} \cdot \Delta\boldsymbol{v}_i^T \cdot \boldsymbol{D} \cdot [\boldsymbol{I} + \boldsymbol{A}] \cdot \boldsymbol{\Gamma}_i^{-1} \cdot \Delta\boldsymbol{v}_i \\ \Delta E \leq &0\end{aligned}$$

Based on Theorem 2.6, the equality holds if and only if $\Delta\boldsymbol{v} = \boldsymbol{0}_N$. This occurs in two cases:

- Limit cycle of length two: $\boldsymbol{v}(l+1) = \boldsymbol{v}(l-1) \neq \boldsymbol{v}(l)$.
- Fixed point: $\boldsymbol{v}(l+1) = \boldsymbol{v}(l) = \boldsymbol{v}(l-1)$.

■

Remark 2.20 In contrast to RNN-1, RNN-2 and RNN-4, the RNN-5 can reach a limit cycle of length two.

Remark 2.21 It is easy to prove that an RNN composed of RNN-4 and RNN-5 is also locally asymptotically stable.

Remark 2.22 We notice that in all previous cases the inner state feedback $\boldsymbol{A}$ appears in the third term (and fourth if given) of the Lyapunov functions. By controlling the values on the diagonal elements of $\boldsymbol{A}$, the relative importance between the different terms of the Lyapunov functions is changed (shaping of the Lyapunov function).

Particularly if $\boldsymbol{A} \to \boldsymbol{I}$ the third term (and the fourth if given) in the Lyapunov functions vanish.

Remark 2.23 Proving the LAS for discrete-time RNNs for activation functions with *dependent* real and imaginary parts is more interesting than doing that for activation functions belonging to $g^{(2)}$ "independent real and imaginary parts". However, it can be proven that this is not possible if Lyapunov functions of the form as in Eq. (2.45),(2.47),(2.50) are used. One exception is the phasor model, where the output of each neuron lies on the unit circle in the complex plane [56]. The activation function in this case is noninvertible.

We focus on Lyapunov functions of the form as in Eq. (2.45),(2.47),(2.50) because the maximum likelihood function of the vector equalization has a similar form.

In this context, activation functions with independent real and imaginary parts are important for square M-ary quadrature amplitude modulation (M-QAM) symbol alphabets, which are more relevant in practice. This is further investigated in Chapter 3. The phasor model on the other hand is relevant for M-ary phase shift keying (M-PSK) symbol alphabets. This is not considered further.

Recurrent Neural Network with Parallel Update and without Inner State Feedback

In case $\boldsymbol{A} = \mathbf{0}_{N\times N}$, RNN-4 and RNN-5 coincide and both are reduced to the RNN-3, cf. Fig. 2.4 and Eq. (2.21).

When discussing the LAS of the RNN-3 in the complex-valued case depending on Theorems 2.4-2.7 we distinguish for $\varphi(\cdot) \in g^{(2)}$:

- The RNN-3 reaches a fixed point. In this case $\boldsymbol{D}\cdot\boldsymbol{W} = \{\boldsymbol{D}\cdot\boldsymbol{W}\}^H$ and $\boldsymbol{D}\cdot\boldsymbol{W} \geq 0$ (or $\tilde{\boldsymbol{W}} > 0$)

- The RNN-3 reaches a fixed point or a limit cycle of length two. In this case $\boldsymbol{D}\cdot\boldsymbol{W} = \{\boldsymbol{D}\cdot\boldsymbol{W}\}^H$.

Remark 2.24 The LAS of RNN-3 as in Theorems 2.4-2.7 coincides with already known stability results with different activation functions than $g^{(2)}$. Here are some examples:

- If the imaginary parts of all variables and activation functions in Eq. (2.21) are zeros and the network is composed of two-state neurons, we obtain the discrete-time Hopfield network (parallel update), cf. [10] and the references therein.

- If the imaginary parts of all variables and activation functions in Eq. (2.21) are zeros and $\varphi(u)$ is a multithreshold function, then the RNN-3 in this case can be transformed into a discrete-time Hopfield network, cf. [10] and the references therein. Thus, the LAS results for the discrete-time Hopfield networks are applicable to the multithreshold RNNs too [18].

- If the imaginary parts of all variables and activation functions in Eq. (2.19) are zeros and $\varphi(u)$ is strictly increasing within an interval and constant outside it, the LAS has been proven in [14].
- If the imaginary parts of all variables and activation functions in Eq. (2.19) are zeros and $\varphi(u)$ is strictly increasing, the LAS has been proven in [10].
- If $\varphi(u)$ is a complex-valued, multistate function, the LAS (without external inputs $\boldsymbol{e} = \mathbf{0}_N$) has been proven in [43],[44].
- If $\varphi(u)$ is a linear threshold function, the LAS has been proven in [94].

Theorem 2.8 RNN-2 and RNN-3 share the same fixed points.

Proof
This has been proven for the real-valued case in [10]. The generalization to the complex-valued case is analogue. We introduce here the main idea.

- Imagine that RNN-2 has been updated (in serial) and it reached a fixed point. If you continue updating it in parallel (RNN-3), no changes on the output do occur.
- Imagine that RNN-3 has been updated (in parallel) and it reached a fixed point. If you continue updating it in serial (RNN-2), no changes on the output do occur.

We conclude that the RNN-2 and RNN-3 share the same fixed points, which are also the equilibrium points of the RNN-1 according to the definition of twin dynamical systems. ■

Remark 2.25 We notice that the advantage of the RNN-2 is the ability to avoid the limit cycle of length two in case of the RNN-3 without requiring that $\boldsymbol{D} \cdot \boldsymbol{W} \geq 0$. This takes place at the cost of a new condition, namely the diagonal elements of $\boldsymbol{W}$ are nonnegative.

Remark 2.26 The RNN-1 combines the update behavior of RNN-3 and the stability behavior of RNN-2.

Theorem 2.9 RNN-4 and RNN-5 share the same fixed points.

Proof
In both cases the fixed points fulfill:

$$\boldsymbol{u}_{fp} = [\boldsymbol{I} - \boldsymbol{A}]^{-1} \cdot [\boldsymbol{W} \cdot \boldsymbol{\varphi}(\boldsymbol{u}_{fp}) + \boldsymbol{W}_0 \cdot \boldsymbol{e}].$$

■

Remark 2.27 RNN-4 & RNN-5 share the same set of fixed points. This means, an optimization problem solved by one of them can be solved by the second too. The choice of using RNN-4 or RNN-5 depends whether the condition $\boldsymbol{D} \cdot \boldsymbol{W} \geq 0$ is fulfilled (RNN-4) or not (RNN-5). However, for RNN-5 there exists the possibility to get stuck in a limit cycle of length 2. This can be understood as the price of not fulfilling $\boldsymbol{D} \cdot \boldsymbol{W} \geq 0$.

2.2.7 Stability Analysis of RNNs With Time-Variant Activation Functions

Time-variant activation functions in this work, cf. Definition 2.11 and Lemma 2.9, are activation functions $\varphi(\cdot) \in g^{(3)}$, where the factors γ_r and γ_i are time-variant during the evolution (iteration) process. Parts of this section have been published in [66].

Remark 2.28 In the following, we limit the $g^{(3)}$ function class to functions as given in Lemma 2.9.

Theorem 2.10 The RNN-1, cf. Fig. 2.3 and Eq. (2.18), reaches an equilibrium point if the following conditions are fulfilled:

- The activation function $\varphi(\cdot) \in g^{(3)}$ and γ_r and γ_i are time-variant and nondecreasing for all neurons during the evolution process.
- There exists a diagonal positive definite matrix $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.

Proof
Under these assumptions, RNN-1 possesses the following Lyapunov function, cf. Theorem 2.1:

$$\begin{aligned} E[\boldsymbol{v}(t)] = &-\frac{1}{2} \cdot \boldsymbol{v}^H(t) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(t) - \Re\left\{\boldsymbol{v}^H(t) \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} \\ &+ \sum_{j=1}^{N} d_j \cdot \phi_j \left[v_{j,r}(t), v_{j,i}(t), \gamma_{j,r}(t), \gamma_{j,i}(t)\right] \end{aligned} \tag{2.52}$$

$$\phi_j(v_{j,r}, v_{j,i}, \gamma_{j,r}, \gamma_{j,i}) = \int_0^{v_{j,r}} \varphi^{-1}_{\gamma_{j,r}}(\vartheta) \mathrm{d}\vartheta + \int_0^{v_{j,i}} \varphi^{-1}_{\gamma_{j,i}}(\vartheta) \mathrm{d}\vartheta$$

$\varphi^{-1}_{\gamma_{j,r}}$ Represents the real part of the inverse activation function of the j-th neuron with $\gamma_{j,r} = \max\left\{\dfrac{\mathrm{d}\varphi_{j,r}(u_{j,r})}{\mathrm{d}u_{j,r}}\right\}$.

$\varphi^{-1}_{\gamma_{j,i}}$ Represents the imaginary part of the inverse activation function of the j-th neuron

with $\gamma_{j,i} = \max\left\{\dfrac{\mathrm{d}\varphi_{j,i}(u_{j,i})}{\mathrm{d}u_{j,i}}\right\}$.
Depending on Theorem 2.1, Lemma 2.2 and the law of the total derivative:

$$\begin{aligned}\frac{\mathrm{d}\phi_j}{\mathrm{d}t} =& u_{j,r} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} + u_{j,i} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} \\ &\underbrace{+ \frac{\partial}{\partial \gamma_{j,r}} \cdot \left[\int_0^{v_{j,r}} \varphi^{-1}_{\gamma_{j,r}}(\vartheta)\mathrm{d}\vartheta\right] \cdot \frac{\mathrm{d}\gamma_{j,r}}{\mathrm{d}t} + \frac{\partial}{\partial \gamma_{j,i}} \cdot \left[\int_0^{v_{j,i}} \varphi^{-1}_{\gamma_{j,i}}(\vartheta)\mathrm{d}\vartheta\right] \cdot \frac{\mathrm{d}\gamma_{j,i}}{\mathrm{d}t}}_{\check{E}_j}\end{aligned}$$

According to Eq. (2.44)

$$\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} = \tilde{E}[\boldsymbol{v}(t)] + \sum_{j=1}^{N} d_j \cdot \check{E}_j$$

- $\tilde{E}[\boldsymbol{v}(t)] \leq 0$ according to Eq. (2.43).
- $\dfrac{\mathrm{d}\gamma_{j,r}}{\mathrm{d}t} \geq 0$ and $\dfrac{\mathrm{d}\gamma_{j,i}}{\mathrm{d}t} \geq 0$, then according to Definition 2.11 and Lemma 2.9, cf. also the next example:

$$\check{E}_j = \frac{\partial}{\partial \gamma_{j,r}} \cdot \left[\int_0^{v_{j,r}} \varphi^{-1}_{\gamma_{j,r}}(\vartheta)\ d\vartheta\right] \cdot \frac{\mathrm{d}\gamma_{j,r}}{\mathrm{d}t} + \frac{\partial}{\partial \gamma_{j,i}} \cdot \left[\int_0^{v_{j,i}} \varphi^{-1}_{\gamma_{j,i}}(\vartheta)\ d\vartheta\right] \cdot \frac{\mathrm{d}\gamma_{j,i}}{\mathrm{d}t} \leq 0$$

This leads to:

$$\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} \leq 0$$

The equality holds if and only if $\dfrac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t} = \mathbf{0}_N$ and either $\dfrac{\mathrm{d}\gamma_r(t)}{\mathrm{d}t} = \dfrac{\mathrm{d}\gamma_i(t)}{\mathrm{d}t} = 0$ or $\gamma_r(t)$, $\gamma_i(t)$ are so large (for all neurons) such that increasing them makes no changes on the shape of the activation functions, cf. Lemma 2.9, Fig. 2.7-2.11. ■

Example 2.2

$$v_r = \varphi_{\gamma_r}(u_r) = \tanh(\gamma_r \cdot u_r) \Leftrightarrow u_r = \varphi^{-1}_{\gamma_r}(v_r) = \frac{1}{2 \cdot \gamma_r} \cdot \ln \frac{1+v_r}{1-v_r}$$

$$\int_0^{v_r} \varphi^{-1}_{\gamma_r}(\vartheta)\mathrm{d}\vartheta = \frac{1}{2 \cdot \gamma_r} \cdot \left[\ln(1-v_r^2) + v_r \cdot \ln \frac{1+v_r}{1-v_r}\right] \ : \ v_r \in (-1, +1)$$

$$\frac{\partial}{\partial \gamma_r}\left[\int_0^{v_r} \varphi^{-1}_{\gamma_r}(\vartheta)\mathrm{d}\vartheta\right] = -\frac{1}{2 \cdot \gamma_r^2} \cdot \left[\ln(1-v_r^2) + v_r \cdot \ln \frac{1+v_r}{1-v_r}\right] \leq 0$$

The equality holds in two cases: either $v_r = 0$ or $\gamma_r \gg 1$.

Theorem 2.11 The RNN-2, cf. Eq. (2.19),(2.20), reaches a fixed point if the following conditions are fulfilled:

- The activation function $\varphi(\cdot) \in g^{(3)}$ and γ_r and γ_i are time-variant and nondecreasing for all neurons during the iteration process.
- There exists a diagonal positive definite matrix $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.
- The diagonal elements of $\boldsymbol{W}$ are nonnegative.

Proof
Under these assumptions, RNN-2 possesses the following Lyapunov function, cf. Theorem 2.2:

$$\begin{aligned} E[\boldsymbol{v}(\rho)] = &- \frac{1}{2} \cdot \boldsymbol{v}^H(\rho) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(\rho) - \Re\left\{\boldsymbol{v}^H(\rho) \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} \\ &+ \Re\left\{\sum_{j=1}^{N} d_j \cdot \int_0^{v_j(\rho)} \varphi^{-1}_{\gamma_j(\rho)}(\vartheta)\mathrm{d}\vartheta^*\right\} \end{aligned} \tag{2.53}$$

Where $\varphi^{-1}_{\gamma_j(\rho)}[u_j(\rho)] = \varphi^{-1}_{\gamma_{j,r}(\rho)}[u_{j,r}(\rho)] + \imath\varphi^{-1}_{\gamma_{j,i}(\rho)}[u_{j,i}(\rho)]$ refers to the inverse activation function of the j-th neuron at discrete-time step (ρ) such that

$$\gamma_{j,r}(\rho) = \max\left\{\frac{\mathrm{d}\varphi_{\gamma_{j,r}(\rho)}(u_{j,r})}{\mathrm{d}u_{j,r}}\right\} \quad , \quad \gamma_{j,i}(\rho) = \max\left\{\frac{\mathrm{d}\varphi_{\gamma_{j,i}(\rho)}(u_{j,i})}{\mathrm{d}u_{j,i}}\right\}$$

We build $\Delta E_j = E[\boldsymbol{v}(\rho+1)] - E[\boldsymbol{v}(\rho)]$ assuming that the j-th neuron has been updated. In this case, as long as $j' \neq j$ it is fulfilled:

$$v_{j'}(\rho+1) = v_{j'}(\rho) \quad , \quad \gamma_{j',r}(\rho+1) = \gamma_{j',r}(\rho) \quad , \quad \gamma_{j',i}(\rho+1) = \gamma_{j',i}(\rho)$$

Following the same steps as in Theorem 2.3 taking into account that according to Definition 2.11, Lemma 2.9 and Eq. (2.34),(A.17)

$$\breve{E}_j = \Re\left\{\int_0^{v_j(\rho)} \varphi^{-1}_{\gamma_j(\rho+1)}(\vartheta)\mathrm{d}\vartheta^* - \int_0^{v_j(\rho)} \varphi^{-1}_{\gamma_j(\rho)}(\vartheta)\mathrm{d}\vartheta^*\right\} \leq 0$$

yields:

$$\begin{aligned}\Delta E_j \leq\ & d_j \cdot \breve{E}_j - \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left| v_j(\rho+1) - v_j(\rho) \right|^2 \\ & - \frac{1}{2} \cdot d_j \cdot \gamma_{j,r}^{-1}(\rho+1) \cdot \left[v_{j,r}(\rho+1) - v_{j,r}(\rho) \right]^2 \\ & - \frac{1}{2} \cdot d_j \cdot \gamma_{j,i}^{-1}(\rho+1) \cdot \left[v_{j,i}(\rho+1) - v_{j,i}(\rho) \right]^2 \\ & \leq 0\end{aligned}$$

A fixed point is reached if $\forall j \in \{1, 2, \ldots, N\} : \Delta E_j = 0$. i.e. $v_j(\rho+1) = v_j(\rho)$ and either $\gamma_{j,r}$, $\gamma_{j,i}$ become constants or $\gamma_{j,r}$, $\gamma_{j,i}$ are so large such that increasing them makes no changes on the shape of the activation functions $\breve{E}_j \to 0$, cf. Remark 2.29. ■

Theorem 2.12 The RNN-3, cf. Fig. 2.4 and Eq. (2.21), reaches a fixed point or a limit cycle of length two if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(3)}$ and γ_r and γ_i are time-variant and nondecreasing for all neurons during the iteration process.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$.

Proof

Under these assumptions, RNN-3 possesses the following Lyapuov function, cf. Theorem 2.6 taking into account that $\boldsymbol{A} = \boldsymbol{0}_{N\times N}$:

$$\begin{aligned}E[\boldsymbol{v}(l)] = & -\Re\left\{ \boldsymbol{v}^H(l) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(l-1) \right\} - \Re\left\{ \left[\boldsymbol{v}^H(l) + \boldsymbol{v}^H(l-1) \right] \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e} \right\} \\ & + \Re\left\{ \sum_{j=1}^{N} d_j \cdot \int_0^{v_j(l)} \varphi_{\gamma_j(l)}^{-1}(\vartheta) \mathrm{d}\vartheta^* \right\} + \Re\left\{ \sum_{j=1}^{N} d_j \cdot \int_0^{v_j(l-1)} \varphi_{\gamma_j(l-1)}^{-1}(\vartheta) \mathrm{d}\vartheta^* \right\}\end{aligned} \tag{2.54}$$

We define:

$$\begin{aligned}\Delta \boldsymbol{v} &= \boldsymbol{v}(l+1) - \boldsymbol{v}(l-1) & \boldsymbol{\Gamma}_r(l) &= \mathrm{diag}\{\gamma_{1,r}(l), \gamma_{2,r}(l), \cdots, \gamma_{N,r}(l)\} \\ \Delta E &= E[\boldsymbol{v}(l+1)] - E[\boldsymbol{v}(l)] & \boldsymbol{\Gamma}_i(l) &= \mathrm{diag}\{\gamma_{1,i}(l), \gamma_{2,i}(l), \cdots, \gamma_{N,i}(l)\}\end{aligned}$$

Following the same steps as in Theorem 2.7 taking into account according to Definition 2.11, Lemma 2.9 and Eq. (2.34),(A.17) that:

$$\breve{E}_j = \Re\left\{ \int_0^{v_j(l-1)} \varphi_{\gamma_j(l+1)}^{-1}(\vartheta) \mathrm{d}\vartheta^* - \int_0^{v_j(l-1)} \varphi_{\gamma_j(l-1)}^{-1}(\vartheta) \mathrm{d}\vartheta^* \right\} \leq 0$$

yields:

$$\Delta E \leq -\frac{1}{2} \cdot \Delta \boldsymbol{v}_r^T \cdot \boldsymbol{D} \cdot \boldsymbol{\Gamma}_r^{-1}(l+1) \cdot \Delta \boldsymbol{v}_r - \frac{1}{2} \cdot \Delta \boldsymbol{v}_i^T \cdot \boldsymbol{D} \cdot \boldsymbol{\Gamma}_i^{-1}(l+1) \cdot \Delta \boldsymbol{v}_i + \sum_{j=1}^{N} d_j \cdot \breve{E}_j$$

$$\Delta E \leq 0$$

The equality holds in two cases:

- Limit cycle of length two: $\boldsymbol{v}(l+1) = \boldsymbol{v}(l-1) \neq \boldsymbol{v}(l)$
- Fixed point: $\boldsymbol{v}(l+1) = \boldsymbol{v}(l) = \boldsymbol{v}(l-1)$

In addition, either $\boldsymbol{\Gamma}_r$, $\boldsymbol{\Gamma}_i$ become constant or $\boldsymbol{\Gamma}_r$, $\boldsymbol{\Gamma}_i$ become so large ($\breve{E}_j \rightarrow 0$, cf. Eq. (A.17)) such that increasing them makes no changes on the shape of the activation functions, cf. Remark 2.29. ■

Theorem 2.13 The RNN-3, cf. Fig. 2.4 and Eq. (2.21), reaches a fixed point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(3)}$ and γ_r and γ_i are time-variant and nondecreasing for all neurons during the iteration process.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$ such that $\boldsymbol{D} \cdot \boldsymbol{W} = \{\boldsymbol{D} \cdot \boldsymbol{W}\}^H$ and $\boldsymbol{D} \cdot \boldsymbol{W} \geq 0$.

Proof
Under these assumptions, RNN-3 possesses the following Lyapuov function, cf. Theorem 2.4 taking into account that $\boldsymbol{A} = \boldsymbol{0}_{N \times N}$:

$$\begin{aligned} E[\boldsymbol{v}(l)] = & -\frac{1}{2} \cdot \Re\left\{\boldsymbol{v}^H(l) \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v}(l)\right\} - \Re\left\{\boldsymbol{v}^H(l) \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} \\ & + \Re\left\{\sum_{j=1}^{N} d_j \cdot \int_0^{v_j(l)} \varphi_{\gamma_j(l)}^{-1}(\vartheta) \mathrm{d}\vartheta^*\right\} \end{aligned} \tag{2.55}$$

We define in this case:

$$\Delta \boldsymbol{v} = \boldsymbol{v}(l+1) - \boldsymbol{v}(l) \qquad \boldsymbol{\Gamma}_r(l) = \text{diag}\{\gamma_{1,r}(l), \gamma_{2,r}(l), \cdots, \gamma_{N,r}(l)\}$$
$$\Delta E = E[\boldsymbol{v}(l+1)] - E[\boldsymbol{v}(l)] \qquad \boldsymbol{\Gamma}_i(l) = \text{diag}\{\gamma_{1,i}(l), \gamma_{2,i}(l), \cdots, \gamma_{N,i}(l)\}$$

Following the same steps as in Theorem 2.5 taking into account that according to Definition 2.11, Lemma 2.9 and Eq. (2.34),(A.17)

$$\breve{E}_j = \Re\left\{\int_0^{v_j(l)} \varphi_{\gamma_j(l+1)}^{-1}(\vartheta) \mathrm{d}\vartheta^* - \int_0^{v_j(l)} \varphi_{\gamma_j(l)}^{-1}(\vartheta) \mathrm{d}\vartheta^*\right\} \leq 0$$

yields:

$$\begin{aligned}
\Delta E \leq & -\frac{1}{2}\cdot\Delta\boldsymbol{v}^{H}\cdot\boldsymbol{D}\cdot\boldsymbol{W}\cdot\Delta\boldsymbol{v}+\sum_{j=1}^{N}d_j\cdot\breve{E}_j \\
& -\frac{1}{2}\cdot\Delta\boldsymbol{v}_r^{T}\cdot\boldsymbol{D}\cdot\boldsymbol{\Gamma}_r^{-1}(l+1)\cdot\Delta\boldsymbol{v}_r-\frac{1}{2}\cdot\Delta\boldsymbol{v}_i^{T}\cdot\boldsymbol{D}\cdot\boldsymbol{\Gamma}_i^{-1}(l+1)\cdot\Delta\boldsymbol{v}_i \\
\Delta E \leq & \quad 0
\end{aligned}$$

A fixed point is reached if $\Delta\boldsymbol{v} = \boldsymbol{0}_N$, i.e. $\boldsymbol{v}(l+1) = \boldsymbol{v}(l)$ and either $\boldsymbol{\Gamma}_r$, $\boldsymbol{\Gamma}_i$ become constant or $\boldsymbol{\Gamma}_r$, $\boldsymbol{\Gamma}_i$ become so large ($\breve{E}_j \rightarrow 0$, cf. Eq. (A.17)) such that increasing them makes no changes on the shape of the activation functions, cf. Remark 2.29 ■

Remark 2.29 Theorems 2.11-2.13 can also be proven based on Lemma 2.3. In this case we follow the same steps as in Theorems 2.2, 2.4,2.6 taking into account Definition 2.11, Lemma 2.9 and Eq. (2.34),(A.17). Doing that, it can be shown that the equality holds for the situations mentioned at the end of Theorem 2.11-2.13.

2.2.8 Global Asymptotical Stability of Recurrent Neural Networks

When solving optimization problems, the RNN is usually designed to have a unique equilibrium and to be globally asymptotically stable to avoid spurious responses or the problem of local minima [32]. Therefore, in both the continuous and the discrete-time cases, the global asymptotical stability (GAS) of RNNs has attracted a lot of interest compared with the LAS.

Among few GAS results we present the following two theorems which are relevant to the vector equalization based on RNNs. The stability conditions, in the light of this application, will be further discussed in Chapter 3.

Theorem 2.14 The RNN-1, cf. Fig. 2.3 and Eq. (2.18), has a unique and globally asymptotically stable equilibrium point if:

- The activation function $\varphi(\cdot) \in g^{(2)}$.
- There exists a diagonal positive definite matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ such that
$$\boldsymbol{Q} = \boldsymbol{D}\cdot\left(\boldsymbol{\Omega}^{-1} - |\boldsymbol{W}|\right) > 0 \tag{2.56}$$
$\boldsymbol{\Omega} = \mathrm{diag}\{\eta_1, \eta_2, \ldots, \eta_N\} > 0$, cf. Lemma 2.8.

Proof
We represent an equilibrium point by $\boldsymbol{u}_{ep} \in \mathbb{C}^N$ and we shift the state-space equation Eq. (2.18) to $\boldsymbol{u}_{ep}$. For this purpose we define:

$$\boldsymbol{z}(t) = \boldsymbol{u}(t) - \boldsymbol{u}_{ep} \quad \Rightarrow \quad \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t}$$

From Eq. (2.18) and Definition 2.2:

$$\boldsymbol{u}(t) = \boldsymbol{u}_{eq} \quad \Leftrightarrow \quad \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = \boldsymbol{0}_N \qquad \Rightarrow \qquad \boldsymbol{u}_{ep} = \boldsymbol{W} \cdot \boldsymbol{\varphi}(\boldsymbol{u}_{ep}) + \boldsymbol{W}_0 \cdot \boldsymbol{e}$$

Substituting these relations in Eq. (2.18) yields:

$$\begin{aligned} \boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} &= -\boldsymbol{z}(t) + \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}\Big[\boldsymbol{z}(t)\Big] \\ \boldsymbol{\varphi}^{(s)}\Big[\boldsymbol{z}(t)\Big] &= \boldsymbol{\varphi}[\boldsymbol{z}(t) + \boldsymbol{u}_{ep}] - \boldsymbol{\varphi}[\boldsymbol{u}_{ep}] \end{aligned} \tag{2.57}$$

$\boldsymbol{\varphi}^{(s)}(\cdot)$ is the same as $\boldsymbol{\varphi}(\cdot)$ but shifted to $(-\boldsymbol{u}_{ep}, -\boldsymbol{v}_{ep}) : \ \boldsymbol{v}_{ep} = \boldsymbol{\varphi}[\boldsymbol{u}_{ep}]$.

The network in Eq. (2.57) has an equilibrium point $\boldsymbol{z} = \boldsymbol{0}_N$. To prove that $\boldsymbol{z} = \boldsymbol{0}_N$ is unique and globally asymptotically stable, we consider the following Lyapunov function:

$$E[\boldsymbol{z}(t)] = \frac{1}{2} \cdot \boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{\Upsilon} \cdot \boldsymbol{z}(t) \tag{2.58}$$

$\boldsymbol{\Upsilon}$, $\boldsymbol{\Omega}$ and $\boldsymbol{D}$ are diagonal and positive definite matrices.

$$E[\boldsymbol{z}(t)] : \begin{cases} = & 0 : \boldsymbol{z}(t) = \boldsymbol{0}_N \\ > & 0 \ \ \text{otherwise} \end{cases}$$

$E[\boldsymbol{z}(t)]$ is unbounded.
According to Eq. (2.57),(2.58):

$$\begin{aligned} \frac{\mathrm{d}E[\boldsymbol{z}(t)]}{\mathrm{d}t} &= \frac{1}{2} \cdot \frac{\mathrm{d}\boldsymbol{z}^H(t)}{\mathrm{d}t} \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{\Upsilon} \cdot \boldsymbol{z}(t) + \frac{1}{2} \cdot \boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} \\ &= \Re\left\{\boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t}\right\} \\ &= \Re\left\{\boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \Big[-\boldsymbol{z}(t) + \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(t)]\Big]\right\} \\ &= \Re\left\{\boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(t)]\right\} - \Re\left\{\boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{z}(t)\right\} \\ &= \Re\left\{\boldsymbol{z}^H(t) \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(t)]\right\} - |\boldsymbol{z}(t)|^T \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot |\boldsymbol{z}(t)| \end{aligned}$$

According to Lemma 2.13:

$$\frac{\mathrm{d}E[\boldsymbol{z}(t)]}{\mathrm{d}t} \leq -|\boldsymbol{z}(t)|^T \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot |\boldsymbol{z}(t)| + |\boldsymbol{z}(t)|^T \cdot \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot |\boldsymbol{W}| \cdot \left|\boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(t)]\right|$$

According to the properties of $g^{(2)}$ functions and Lemma 2.8:

$$\begin{aligned} \frac{\mathrm{d}E[\boldsymbol{z}(t)]}{\mathrm{d}t} &\leq - \left|\boldsymbol{z}(t)\right|^T \cdot \left\{\boldsymbol{\Omega} \cdot \boldsymbol{D} - \boldsymbol{\Omega} \cdot \boldsymbol{D} \cdot \left|\boldsymbol{W}\right| \cdot \boldsymbol{\Omega}\right\} \cdot \left|\boldsymbol{z}(t)\right| \\ &\leq - \left|\boldsymbol{z}(t)\right|^T \cdot \boldsymbol{\Omega} \cdot \boldsymbol{Q} \cdot \boldsymbol{\Omega} \cdot \left|\boldsymbol{z}(t)\right| \qquad : \qquad \boldsymbol{Q} > 0 \\ &\leq \ 0 \end{aligned}$$

The equality holds if and only if $\boldsymbol{z}(t) = \boldsymbol{0}_N$, i.e. $\boldsymbol{u}(t) = \boldsymbol{u}_{ep}$. ■

Remark 2.30 The original proof [91] is slightly different from the proof in Theorem 2.14. Small modifications have been done in order to fit our RNN vector equalizer application.

Theorem 2.15 The RNN-3, cf. Fig. 2.4 and Eq. (2.21), has a unique and globally asymptotically stable fixed point if:

- The activation function $\varphi(\cdot) \in g^{(2)}$.
- There exists a diagonal positive definite matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ such that
$$\boldsymbol{Q} = \boldsymbol{D}^2 \cdot \boldsymbol{\Omega}^{-2} - |\boldsymbol{W}|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{W}| > 0 \tag{2.59}$$
$\boldsymbol{\Omega} = \mathrm{diag}\{\eta_1, \eta_2, \ldots, \eta_N\}$, cf. Lemma 2.8.

Proof
We represent the fixed point by $\boldsymbol{u}_{fp} \in \mathbb{C}^N$ and we shift Eq. (2.21) to $\boldsymbol{u}_{fp}$. For this purpose we define:

$$\boldsymbol{z}(l) = \boldsymbol{u}(l) - \boldsymbol{u}_{fp}$$

According to Definition 2.1 and Eq. (2.21):

$$\boldsymbol{W}_0 \cdot \boldsymbol{e} = \boldsymbol{u}_{fp} - \boldsymbol{W} \cdot \boldsymbol{\varphi}[\boldsymbol{u}_{fp}]$$

Substituting these relations in Eq. (2.21) yields:

$$\begin{aligned} \boldsymbol{z}(l+1) &= \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}\left[\boldsymbol{z}(l)\right] \\ \boldsymbol{\varphi}^{(s)}\left[\boldsymbol{z}(l)\right] &= \boldsymbol{\varphi}[\boldsymbol{z}(l) + \boldsymbol{u}_{fp}] - \boldsymbol{\varphi}[\boldsymbol{u}_{fp}] \end{aligned} \tag{2.60}$$

$\boldsymbol{\varphi}^{(s)}(\cdot)$ is the same as $\boldsymbol{\varphi}(\cdot)$ but shifted to $(-\boldsymbol{u}_{fp}, -\boldsymbol{v}_{fp}) : \ \boldsymbol{v}_{fp} = \boldsymbol{\varphi}[\boldsymbol{u}_{fp}]$.

The network in Eq. (2.60) has a fixed point $\boldsymbol{z} = \mathbf{0}_N$. To prove that $\boldsymbol{z} = \mathbf{0}_N$ is unique and globally asymptotically stable, we consider the following Lyapunov function:

$$E[\boldsymbol{z}(l)] = \boldsymbol{z}^H(l) \cdot \boldsymbol{D}^2 \cdot \boldsymbol{z}(l) \tag{2.61}$$

$$E[\boldsymbol{z}(l)] : \begin{cases} & 0 : \boldsymbol{z}(l) = \mathbf{0}_N \\ > & 0 \quad \text{otherwise} \end{cases}$$

$E[\boldsymbol{z}(l)]$ is unbounded.
Based on Eq. (2.60),(2.61):

$$\begin{aligned} \Delta E &= E[\boldsymbol{z}(l+1)] - E[\boldsymbol{z}(l)] \\ &= \boldsymbol{z}^H(l+1) \cdot \boldsymbol{D}^2 \cdot \boldsymbol{z}(l+1) - \boldsymbol{z}^H(l) \cdot \boldsymbol{D}^2 \cdot \boldsymbol{z}(l) \\ &= \boldsymbol{\varphi}^{(s),H}[\boldsymbol{z}(l)] \cdot \boldsymbol{W}^H \cdot \boldsymbol{D}^2 \cdot \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(l)] - \boldsymbol{z}^H(l) \cdot \boldsymbol{D}^2 \cdot \boldsymbol{z}(l) \\ &= \boldsymbol{\varphi}^{(s),H}[\boldsymbol{z}(l)] \cdot \boldsymbol{W}^H \cdot \boldsymbol{D}^2 \cdot \boldsymbol{W} \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(l)] - |\boldsymbol{z}(l)|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{z}(l)| \end{aligned}$$

According to Lemma 2.13:

$$\Delta E \leq -|\boldsymbol{z}(l)|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{z}(l)| + \left|\boldsymbol{\varphi}^{(s),H}[\boldsymbol{z}(l)]\right| \cdot |\boldsymbol{W}|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{W}| \cdot \left|\boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(l)]\right|$$

For clarity of presentation, we suppress the dependency on the time index (l). Using Lemma 2.8:

$$\begin{aligned} \Delta E &\leq - |\boldsymbol{z}|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{z}| + |\boldsymbol{z}|^T \cdot \boldsymbol{\Omega}^T \cdot |\boldsymbol{W}|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{W}| \cdot \boldsymbol{\Omega} \cdot |\boldsymbol{z}| \\ &\leq - |\boldsymbol{z}|^T \cdot \left\{\boldsymbol{D}^2 - \boldsymbol{\Omega}^T \cdot |\boldsymbol{W}|^T \cdot \boldsymbol{D}^2 \cdot |\boldsymbol{W}| \cdot \boldsymbol{\Omega}\right\} \cdot |\boldsymbol{z}| \\ &\leq - |\boldsymbol{z}|^T \cdot \boldsymbol{\Omega} \cdot \boldsymbol{Q} \cdot \boldsymbol{\Omega} \cdot |\boldsymbol{z}| \qquad : \qquad \boldsymbol{Q} > 0 \\ &\leq \ 0 \end{aligned}$$

The equality holds if and only if $\boldsymbol{z}(l) = 0$, i.e. $\boldsymbol{u}(l) = \boldsymbol{u}(l+1) = \boldsymbol{u}_{fp}$. ■

Remark 2.31 The original proof [91] has been done for RNN-4. In this case:

$$\boldsymbol{Q} = \left(\frac{b}{1+b} \cdot \boldsymbol{D} - b \cdot \boldsymbol{A} \cdot \boldsymbol{D} \cdot \boldsymbol{A}\right) \cdot \boldsymbol{\Omega}^{-2} - |\boldsymbol{W}|^T \cdot \boldsymbol{D} \cdot |\boldsymbol{W}| \quad > \quad 0 \tag{2.62}$$

where b is a positive number and $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is any diagonal positive definite matrix such that Eq. (2.62) is fulfilled.
We presented in Theorem 2.15 a special case where $b \to \infty$ and $\boldsymbol{A} = \mathbf{0}_{N \times N}$.

2.3 Single Layer High Order Recurrent Neural Networks

Single layer recurrent neural networks are well known to be a very important tool to solve classification and optimization problems without the need for a training process because of their ability to be Lyapunov-stable. This has already been discussed in the previous sections.

However, one of the largest drawbacks of RNNs is their quadratic Lyapunov function.Thus, optimization problems associated with cost functions of higher degree cannot be solved "satisfactorily" by RNNs. *"While preliminary applications using RNNs were encouraging, others revealed limitations that may be helped by increasing the order of the Lyapunov function"* [8].

For this reason, researchers aim to increase the order of the Lyapunov function and thus allowing nonlinear feedback (nonlinear interconnections between the neurons). This increases the class of optimization problems that can be solved by recurrent neural networks [39]. Doing so, we obtain the single layer high order recurrent neural network (HORNN). Therefore, HORNNs can be considered as a generalization of RNNs, that allow nonlinear interaction between the neurons.

It is worth mentioning that the term "high order" refers to the interconnection between the neurons not to the degree of the differential/difference equation, which describes the dynamics. As for RNNs this is still of first order.

In order to apply the HORNNs to solve optimization tasks, their stability has to be investigated. A property without which the behavior of dynamical systems is often suspect [8]. This was the topic of many publications as in [8],[39],[75],[85]. An Example of using HORNNs to solve optimization problems can be found in [89], where it has been focused on the traveling salesman problem.

Depending on the kind of nonlinear interconnection between the neurons, the network possesses different names. Gradient-like (gradient-type) systems (multilinear objective functions) [1],[85], gradient recurrent high order neural networks [39], high order absolutely stable neural networks [8], high order dynamic neural networks [75], high order Hopfield neural networks [2] are just a few examples. However, polynomial interconnection between the neurons is the most common one.

It is worth mentioning that the stability investigation is almost exclusively focused on the continuous-time case. One of the very few publications considering the local stability for the discrete-time "generalized Hopfield model" can be found in [90].

In this section, we focus on real-valued HORNNs. Using the mean value theorem, we connect between RNNs and HORNNs such that global stability results of RNNs can be modified to be valid for HORNNs too. Like RNNs, the dynamical behavior of HORNNs is given by the state-space representation.

2.3.1 Continuous-Time High Order Recurrent Neural Networks

Figure 2.12 shows the continuous-time HORNN. The dynamical behavior is given by:

$$\begin{aligned} \mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} &= -\boldsymbol{u}(t) + \boldsymbol{W} \cdot \boldsymbol{f}[\boldsymbol{v}(t)] + \boldsymbol{W}_0 \cdot \boldsymbol{e} \\ \boldsymbol{v}(t) &= \boldsymbol{\varphi}[\boldsymbol{u}(t)] \end{aligned} \tag{2.63}$$

The parameters in Eq. (2.63) are the same as in Eq. (2.18) but real-valued. $\boldsymbol{f}[\boldsymbol{v}]$ is a real-valued continuously differentiable nonlinear vector function and thus locally Lipschitz function [72]. In addition, $\boldsymbol{f}(\mathbf{0}_N) = \mathbf{0}_N$.

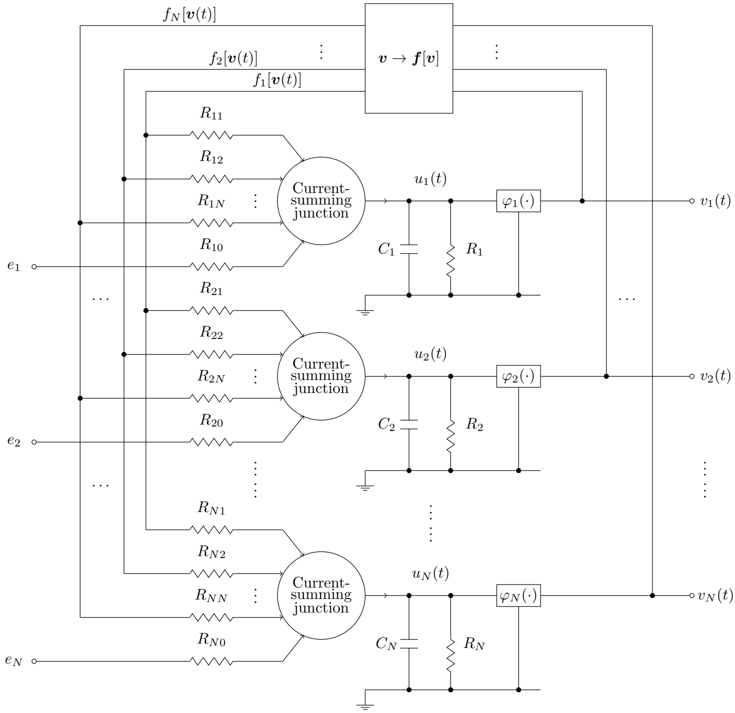


Figure 2.12: Continuous-time high order recurrent neural network. $\boldsymbol{v}(t)$ is the output, $\boldsymbol{u}(t)$ the inner state, $\boldsymbol{e}$ the external input and $\varphi(\cdot)$ the activation function.

2.3.2 Discrete-Time High Order Recurrent Neural Networks

Figure 2.13 shows the discrete-time HORNN. The dynamical behavior is given by:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \boldsymbol{W} \cdot \boldsymbol{f}[\boldsymbol{v}(l)] + \boldsymbol{W}_0 \cdot \boldsymbol{e} \\ \boldsymbol{v}(l) &= \boldsymbol{\varphi}[\boldsymbol{u}(l)] \end{aligned} \tag{2.64}$$

All parameters in Eq. (2.64) are the same as in Eq. (2.63).

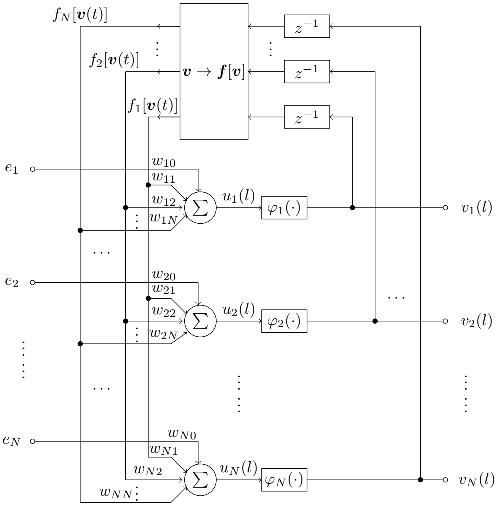


Figure 2.13: Discrete-time high order recurrent neural network. $\boldsymbol{v}(l)$ is the output, $\boldsymbol{u}(l)$ the inner state, $\boldsymbol{e}$ the external input and $\varphi(\cdot)$ the activation function.

Remark 2.32 Depending on Definition 2.8 the discrete- and the continuous-time HORNNs represent twin dynamical systems.

2.3.3 Stability Analysis of High Order Recurrent Neural Networks

Theorem 2.16 The continuous-time HORNN, cf. Fig. 2.12 and Eq. (2.63), reaches an equilibrium point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(4)}$.

ii) There exists a scalar function $F[\boldsymbol{v}]$ such that $\left\{\nabla \cdot F[\boldsymbol{v}]\right\}^T = \boldsymbol{W} \cdot \boldsymbol{f}[\boldsymbol{v}]$.

Proof
Under these assumptions, the continuous-time HORNN possesses the following Lyapunov function:

$$E[\boldsymbol{v}(t)] = -F[\boldsymbol{v}(t)] - \boldsymbol{v}^T(t) \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e} + \sum_{j=1}^{N} \int_0^{v_j(t)} \varphi_j^{-1}(\vartheta)\mathrm{d}\vartheta \tag{2.65}$$

According to the second condition, the dynamical behavior of the continuous-time HORNN Eq. (2.63) can be rewritten as:

$$\mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = -\boldsymbol{u}(t) + \left\{\nabla \cdot F[\boldsymbol{v}(t)]\right\}^T + \boldsymbol{W}_0 \cdot \boldsymbol{e} \tag{2.66}$$

In this case:

$$\begin{aligned}
\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} &= \sum_{j=1}^{N} \frac{\partial E[\boldsymbol{v}]}{\partial v_j} \cdot \frac{\mathrm{d}v_j(t)}{\mathrm{d}t} \\
&= -\sum_{j=1}^{N} \left\{-u_j(t) + \nabla \cdot F[\boldsymbol{v}(t)]_j + w_{jo} \cdot e_j\right\} \cdot \frac{\mathrm{d}v_j(t)}{\mathrm{d}t} \\
&= -\sum_{j=1}^{N} \tau_j \cdot \frac{\mathrm{d}u_j(t)}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_j(t)}{\mathrm{d}t} \\
&= -\frac{\mathrm{d}\boldsymbol{v}^T(t)}{\mathrm{d}t} \cdot \mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} \\
&\leq 0
\end{aligned}$$

Because $\varphi(\cdot) \in g^{(4)}$ is a strictly increasing function.
The equality holds if and only if $\dfrac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t} = \mathbf{0}_N$. This means an equilibrium point has been reached. ■

Remark 2.33 The last proof can be found in the literature in different ways [39],[85]. The most important property in this case is that, the left side of Eq. (2.63) is described by means of the gradient of a scalar function.

Remark 2.34 If $\boldsymbol{f}[\boldsymbol{v}] = \boldsymbol{v}$ & $\boldsymbol{W} = \boldsymbol{W}^T$, the continuous-time HORNN reduces to the real-valued RNN-1. In this case

$$F(v_1, v_2, \cdots, v_N) = \frac{1}{2} \cdot \sum_{j=1}^{N} \sum_{j'=1}^{N} w_{jj'} \cdot v_j \cdot v_{j'}$$

$$\left\{\nabla \cdot F[\boldsymbol{v}]\right\}^T = \boldsymbol{W} \cdot \boldsymbol{v}$$

Theorem 2.17 The continuous-time HORNN, cf. Fig. 2.12 and Eq. (2.63), has a unique and globally asymptotically stable equilibrium point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(4)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$ such that

$$\boldsymbol{Q} = \boldsymbol{D} \cdot \left(\boldsymbol{\Omega}^{-1} - \left|\boldsymbol{W} \cdot \boldsymbol{J}_f[\boldsymbol{v}_0]\right|\right) > 0 \ : \ \forall \boldsymbol{v}_0 \in \mathcal{B}_r^N \tag{2.67}$$

Where $\boldsymbol{J}_f[\boldsymbol{v}_0]$ is the Jacobian matrix of the vector valued function $\boldsymbol{f}$ at the point $\boldsymbol{v}_0$ and $\boldsymbol{\Omega} = \text{diag}\{\eta_1, \eta_2, \ldots, \eta_N\} > 0$, cf. Definition 2.12 and Lemma 2.7.

Proof
The dynamical behavior of the continuous-time HORNN centered at the equilibrium point $\boldsymbol{u}_{ep}$ can be given as:

$$\Upsilon \cdot \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = -\boldsymbol{z}(t) + \boldsymbol{W} \cdot \left\{\boldsymbol{f}[\boldsymbol{v}(t)] - \boldsymbol{f}[\boldsymbol{v}_{ep}]\right\}$$

This can be rewritten considering the mean value theorem on several variables as [70]:

$$\Upsilon \cdot \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = -\boldsymbol{z}(t) + \boldsymbol{W} \cdot \boldsymbol{J}_f[\boldsymbol{v}_0(t)] \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(t)]$$

$\boldsymbol{v}_0(t)$ is between $\boldsymbol{v}(t)$ and $\boldsymbol{v}_{ep}$. $\boldsymbol{\varphi}^{(s)}(\cdot)$ is the same as $\boldsymbol{\varphi}(\cdot)$ but shifted to $(-\boldsymbol{u}_{ep}, -\boldsymbol{v}_{ep})$.

Assume $\boldsymbol{W}(t) = \boldsymbol{W} \cdot \boldsymbol{J}_f[\boldsymbol{v}_0(t)]$, the last relation can be expressed as:

$$\Upsilon \cdot \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = -\boldsymbol{z}(t) + \boldsymbol{W}(t) \cdot \boldsymbol{\varphi}^{(s)}[\boldsymbol{z}(t)] \tag{2.68}$$

which is similar to Eq. (2.57) except the time variation of the weight matrix.

As we found previously, the weight matrix does not appear in the Lyapunov function Eq. (2.58) during the GAS analysis of the RNN-1. This means that the same Lyapunov function Eq. (2.58) can be used and during the time derivation we take into account that the weight matrix is time variant. This leads to the second condition mentioned in this theorem. ■

Theorem 2.18 The discrete-time HORNN, cf. Fig 2.13 and Eq. (2.64), has a unique and globally asymptotically stable fixed point if the following conditions are fulfilled:

i) The activation function $\varphi(\cdot) \in g^{(4)}$.

ii) There exists a diagonal positive definite matrix $\boldsymbol{D} = \text{diag}\{d_1, d_2, \ldots, d_N\}$ such that

$$\boldsymbol{Q} = \boldsymbol{D}^2 \cdot \boldsymbol{\Omega}^{-2} - \left|\boldsymbol{W} \cdot \boldsymbol{J}_f[\boldsymbol{v}_0]\right|^T \cdot \boldsymbol{D}^2 \cdot \left|\boldsymbol{W} \cdot \boldsymbol{J}_f[\boldsymbol{v}_0]\right| > 0 \;:\; \forall \boldsymbol{v}_0 \in \mathcal{B}_r^N \tag{2.69}$$

Where $\boldsymbol{J}_f[\boldsymbol{v}_0]$ is the Jacobian matrix of the vector valued function $\boldsymbol{f}$ at the point $\boldsymbol{v}_0$ and $\boldsymbol{\Omega} = \text{diag}\{\eta_1, \eta_2, \ldots, \eta_N\} > 0$, cf. Definition 2.12 and Lemma 2.7.

Proof
Following the same approach as in Theorem 2.17, the discrete-time HORNN can be reduced to the RNN-3, with a time-variant weight matrix, where the results of Theorem 2.15 can be applied. ■

2.4 Chapter Summary

In this chapter, we introduced the structure and the dynamical behavior of recurrent neural networks and high order recurrent neural networks. They were considered as nonlinear dynamical systems. Trained neural networks and training algorithms have not been considered. A major distinction has been done between discrete-time networks and continuous-time ones.

The stability is one of the most important properties of these networks. So, the main focus of this chapter was the stability investigation in the sense of Lyapunov, both locally and globally. We could extend and generalize many already known stability conditions, especially for complex-valued networks.

One of the most important contributions of this chapter is the stability investigation of the recurrent neural networks with time-variant activation functions. They have been already applied (heuristic approach) to solve the vector equalization problem with good results. In this chapter, we proved that recurrent neural networks with time-variant activation functions stay Lyapunov-stable if some restrictions on the characteristic of the time variations are fulfilled.

Another important contribution is the definition of twin dynamical systems, where a continuous-time dynamical system can be derived from a discrete-time one. Both share the same set of fixed/equilibrium points. This plays an important role, when analog realization of (discrete) iterative methods is required.

Chapter 3

Recurrent Neural Networks for Vector Equalization

In this chapter we relate the vector equalization discussed in Chapter 1 with the RNN structures introduced in Chapter 2. This includes the definition of the RNN (inputs, outputs, weight matrix, activation function etc.) to act as a vector equalizer.

We start this chapter with the parameter estimation problem for general symbol alphabets, which leads to the optimum estimation function $\theta^{(opt)}(\cdot)$, mentioned in Chapter 1. The properties and special form of $\theta^{(opt)}(\cdot)$ for specific symbol alphabets are considered as well. This is a quite known problem [10], [11], therefore we mention only the problem formulation and the result.

We show also the relation between the optimum estimation function $\theta^{(opt)}(\cdot)$ and the classes of g functions in Chapter 2. The evaluation of the optimum estimation function $\theta^{(opt)}(\cdot)$ is investigated afterward. It is shown that for special symbol alphabets the optimum estimation function $\theta^{(opt)}(\cdot)$ can be approximated by a sum of shifted hyperbolic tangent functions [65]. This is important for a numerically stable evaluation of $\theta^{(opt)}(\cdot)$. The vector equalization based on RNNs (VE-RNNs) with all related details embody the last part in this chapter.

Vector equalization based on recurrent neural networks (and their modifications) is not a new idea. In the discrete-time case, they have been considered for example in [12], [56], [58], [59], [64], [73], [80], [81], [82], [83], [88], [93].

In the continuous-time case, to the best of our knowledge, the analytical investigation is restricted to the BPSK [36], [37], [55], [62]. This is generalized in this chapter to square M-QAM. We notice that [58], [59], [62], [64] are our own publications.

In addition, performance improving techniques, like time-variant activation functions, are considered from the stability point of view. Following this way, it can be understood, why such a technique can improve the performance of VE-RNNs.

3.1 The Problem of Parameter Estimation

Consider the estimation problem depicted in Fig. 3.1. A transmit symbol $x \in \Psi = \{\psi_1, \psi_2, \cdots, \psi_M\} : \psi_j \in \mathbb{C} : \forall j \in \{1, 2, \cdots, M\}$ is transmitted over a channel which adds noise and/or interference. Ψ is the symbol alphabet, ζ is the random variable representing x. $\tilde{x}$ is the received symbol and $\tilde{\zeta}$ is the random variable representing $\tilde{x}$. The estimation function $\theta(\cdot)$ has, with given $\tilde{x}$, to deliver $\breve{x}$ (an estimate for x) which has to be as close as possible to x, i.e. minimizing the mean squared error $E_{xp}\left\{|\breve{\zeta} - \zeta|^2\right\}$ where $\breve{\zeta}$ is the random variable representing $\breve{x}$. E_{xp} stands for the mathematical expectation.

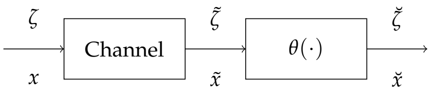


Figure 3.1: Parameter estimation problem.

If x is an interfering symbol, which needs to be estimated in order to cancel its influence on other symbols, then $E_{xp}\left\{|\breve{\zeta} - \zeta|^2 \middle| \tilde{x}\right\}$ is the residual interference power after eliminating the interference.

The function $\theta(\cdot)$ can be analytically derived if the interference added by the channel (other symbols, other users) can be modeled as white Gaussian noise. As a consequence, the channel in Fig. 3.1 is modeled by a zero mean, complex-valued, additive white Gaussian noise where σ_r^2, σ_i^2 are the variances of the statistically independent noise components. In this case, it has been found that the optimum estimation function, taking into account that $\beta_r = \frac{1}{\sigma_r^2}$, $\beta_i = \frac{1}{\sigma_i^2}$, is [10], [11]:

$$\begin{aligned}\breve{x}^{(opt)} &= \theta^{(opt)}(\tilde{x}) \\ &= \frac{\sum_{j=1}^{M} \psi_j \cdot \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}{\sum_{j=1}^{M} \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}\end{aligned} \tag{3.1}$$

We notice that $\breve{x}^{(opt)}$ is not limited to the symbol alphabet Ψ.

Separating the real and imaginary parts in Eq. (3.1) yields:

$$\begin{aligned}\breve{x}_r^{(opt)} &= \theta_r^{(opt)}(\tilde{x}_r, \tilde{x}_i) \\ &= \frac{\sum_{j=1}^{M} \psi_{j,r} \cdot \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}{\sum_{j=1}^{M} \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}\end{aligned} \tag{3.2}$$

$$\begin{aligned}\breve{x}_i^{(opt)} &= \theta_i^{(opt)}(\tilde{x}_r, \tilde{x}_i) \\ &= \frac{\sum_{j=1}^{M} \psi_{j,i} \cdot \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}{\sum_{j=1}^{M} \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}\end{aligned} \tag{3.3}$$

For the residual interference, it has been found that [10]:

$$E_{xp}\left\{|\breve{\zeta}-\zeta|^2\middle|\tilde{x}\right\} = J_r + J_i \tag{3.4}$$

$$J_r = \frac{\sum_{j=1}^{M} \psi_{j,r}^2 \cdot \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}{\sum_{j=1}^{M} \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}} - \breve{x}_r^2 \tag{3.5}$$

$$J_i = \frac{\sum_{j=1}^{M} \psi_{j,i}^2 \cdot \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}}{\sum_{j=1}^{M} \exp\left\{-\frac{1}{2}\cdot\left(\beta_r\cdot\psi_{j,r}^2+\beta_i\cdot\psi_{j,i}^2\right)\right\}\cdot\exp\left\{\beta_r\cdot\psi_{j,r}\cdot\tilde{x}_r\right\}\cdot\exp\left\{\beta_i\cdot\psi_{j,i}\cdot\tilde{x}_i\right\}} - \breve{x}_i^2 \tag{3.6}$$

Special Case
For M-PSK symbol alphabets, it can be easily shown in case of $\beta_r = \beta_i$ that:

$$E_{xp}\left\{|\breve{\zeta}-\zeta|^2\middle|\tilde{x}\right\} = 1 - |\breve{x}|^2$$

3.1.1 Separable Symbol Alphabets

This subsection depends on [10]. Given two real-valued symbol alphabets

$$\Psi^{(1)} = \left\{\psi_1^{(1)}, \psi_2^{(1)}, \cdots, \psi_{M_r}^{(1)}\right\} \quad , \quad \Psi^{(2)} = \left\{\psi_1^{(2)}, \psi_2^{(2)}, \cdots, \psi_{M_i}^{(2)}\right\}$$

$\Psi^{(1)}$ and $\Psi^{(2)}$ fulfill the following conditions:

- $\log_2 M_r, \log_2 M_i \in \mathbb{N}/\{0\}$.
- $\psi_1^{(1)} < \psi_2^{(1)} < \cdots < \psi_{M_r}^{(1)}$ and $\psi_1^{(2)} < \psi_2^{(2)} < \cdots < \psi_{M_i}^{(2)}$.
- $\forall j \in \left\{1, 2, \cdots, \frac{M_r}{2}\right\} : \psi_j^{(1)} = -\psi_{M_r+1-j}^{(1)}$.
- $\forall j \in \left\{1, 2, \cdots, \frac{M_i}{2}\right\} : \psi_j^{(2)} = -\psi_{M_i+1-j}^{(2)}$.

The last two relations indicate the even symmetry of $\Psi^{(1)}$ and $\Psi^{(2)}$. We build now a new symbol alphabet $\Psi^{(sp)} = \left\{\psi_1^{(sp)}, \psi_2^{(sp)}, \cdots \psi_{M_r \cdot M_i}^{(sp)}\right\}$ (sp stands for *separable*) such that:

	$\psi_1^{(1)}$	$\psi_2^{(1)}$	$\cdots$	$\psi_{M_r}^{(1)}$
$\psi_1^{(2)}$	$\psi_1^{(sp)} = \psi_1^{(1)} + \imath\psi_1^{(2)}$	$\psi_2^{(sp)} = \psi_2^{(1)} + \imath\psi_1^{(2)}$	$\cdots$	$\psi_{M_r}^{(sp)} = \psi_{M_r}^{(1)} + \imath\psi_1^{(2)}$
$\psi_2^{(2)}$	$\psi_{M_r+1}^{(sp)} = \psi_1^{(1)} + \imath\psi_2^{(2)}$	$\psi_{M_r+2}^{(sp)} = \psi_2^{(1)} + \imath\psi_2^{(2)}$	$\cdots$	$\psi_{2\cdot M_r}^{(sp)} = \psi_{M_r}^{(1)} + \imath\psi_2^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\psi_{M_i}^{(2)}$	$\psi_{M_r\cdot(M_i-1)+1}^{(sp)} = \psi_1^{(1)} + \imath\psi_{M_i}^{(2)}$	$\psi_{M_r\cdot(M_i-1)+2}^{(sp)} = \psi_2^{(1)} + \imath\psi_{M_i}^{(2)}$	$\cdots$	$\psi_{M_r\cdot M_i}^{(sp)} = \psi_{M_r}^{(1)} + \imath\psi_{M_i}^{(2)}$

In this case, Eq. (3.2), (3.3) can be rewritten as:

$$\breve{x}_r^{(opt)} = \theta_r^{(opt)}(\tilde{x}_r) = \frac{\sum_{j=1}^{M_r} \psi_j^{(1)} \cdot \exp\left\{-\frac{1}{2}\cdot\beta_r\cdot\psi_j^{(1)\,2}\right\}\cdot\exp\left\{\beta_r\cdot\psi_j^{(1)}\cdot\tilde{x}_r\right\}}{\sum_{j=1}^{M_r} \exp\left\{-\frac{1}{2}\cdot\beta_r\cdot\psi_j^{(1)\,2}\right\}\cdot\exp\left\{\beta_r\cdot\psi_j^{(1)}\cdot\tilde{x}_r\right\}} \tag{3.7}$$

$$\breve{x}_i^{(opt)} = \theta_i^{(opt)}(\tilde{x}_i) = \frac{\sum_{j=1}^{M_i} \psi_j^{(2)} \cdot \exp\left\{-\frac{1}{2} \cdot \beta_i \cdot \psi_j^{(2)\,2}\right\} \cdot \exp\left\{\beta_i \cdot \psi_j^{(2)} \cdot \tilde{x}_i\right\}}{\sum_{j=1}^{M_i} \exp\left\{-\frac{1}{2} \cdot \beta_i \cdot \psi_j^{(2)\,2}\right\} \cdot \exp\left\{\beta_i \cdot \psi_j^{(2)} \cdot \tilde{x}_i\right\}} \tag{3.8}$$

Because $\Psi^{(1)}$ and $\Psi^{(2)}$ are even symmetric, Eq. (3.7), (3.8) can be presented in a more compact way:

$$\breve{x}_r^{(opt)} = \theta_r^{(opt)}(\tilde{x}_r) = \frac{\sum_{j=1}^{\frac{M_r}{2}} |\psi_j^{(1)}| \cdot \exp\left\{-\frac{1}{2} \cdot \beta_r \cdot \psi_j^{(1)\,2}\right\} \cdot \sinh\left[\beta_r \cdot |\psi_j^{(1)}| \cdot \tilde{x}_r\right]}{\sum_{j=1}^{\frac{M_r}{2}} \exp\left\{-\frac{1}{2} \cdot \beta_r \cdot \psi_j^{(1)\,2}\right\} \cdot \cosh\left[\beta_r \cdot |\psi_j^{(1)}| \cdot \tilde{x}_r\right]} \tag{3.9}$$

$$\breve{x}_i^{(opt)} = \theta_i^{(opt)}(\tilde{x}_i) = \frac{\sum_{j=1}^{\frac{M_i}{2}} |\psi_j^{(2)}| \cdot \exp\left\{-\frac{1}{2} \cdot \beta_i \cdot \psi_j^{(2)\,2}\right\} \cdot \sinh\left[\beta_i \cdot |\psi_j^{(2)}| \cdot \tilde{x}_i\right]}{\sum_{j=1}^{\frac{M_i}{2}} \exp\left\{-\frac{1}{2} \cdot \beta_i \cdot \psi_j^{(2)\,2}\right\} \cdot \cosh\left[\beta_i \cdot |\psi_j^{(2)}| \cdot \tilde{x}_i\right]} \tag{3.10}$$

We call the group of symbol alphabets $\Psi^{(sp)}$ *separable symbol alphabets*, because the real and imaginary parts are separated in the sense of the optimum estimation function in Fig. 3.1.

Example 3.1 For Quadrature phase shift keying (QPSK) $\Psi^{(1)} = \Psi^{(2)} = \{-1, +1\}$ and $\Psi^{(sp)} = \{-1-\imath, -1+\imath, +1-\imath, +1+\imath\}$

$$\breve{x}_r^{(opt)} = \theta_r^{(opt)}(\tilde{x}_r) = \tanh(\beta_r \cdot \tilde{x}_r)$$
$$\breve{x}_i^{(opt)} = \theta_i^{(opt)}(\tilde{x}_i) = \tanh(\beta_i \cdot \tilde{x}_i)$$

The Real-Valued Case:

For real-valued symbol alphabets $\Psi = \Psi^{(1)}$ & $M = M_r$, Eq. (3.2), (3.3) can be rewritten as:

$$\begin{aligned}\breve{x}_r^{(opt)} &= \theta_r^{(opt)}(\tilde{x}) = \frac{\sum_{j=1}^{M} \psi_j \cdot \exp\left\{-\frac{1}{2} \cdot \beta_r \cdot \psi_j^2\right\} \cdot \exp\left\{\beta_r \cdot \psi_j \cdot \tilde{x}_r\right\}}{\sum_{j=1}^{M} \exp\left\{-\frac{1}{2} \cdot \beta_r \cdot \psi_j^2\right\} \cdot \exp\left\{\beta_r \cdot \psi_j \cdot \tilde{x}_r\right\}} \\ \breve{x}_i^{(opt)} &= 0\end{aligned} \tag{3.11}$$

Because the symbol alphabet is even symmetry:

$$\begin{aligned}\breve{x}_r^{(opt)} &= \theta_r^{(opt)}(\tilde{x}) = \frac{\sum_{j=1}^{\frac{M}{2}} |\psi_j| \cdot \exp\left\{-\frac{1}{2} \cdot \beta_r \cdot \psi_j^2\right\} \cdot \sinh\left[\beta_r \cdot |\psi_j| \cdot \tilde{x}_r\right]}{\sum_{j=1}^{\frac{M}{2}} \exp\left\{-\frac{1}{2} \cdot \beta_r \cdot \psi_j^2\right\} \cdot \cosh\left[\beta_r \cdot |\psi_j| \cdot \tilde{x}_r\right]} \\ \breve{x}_i^{(opt)} &= 0\end{aligned} \tag{3.12}$$

Example 3.2 For BPSK $\Psi = \{-1, +1\}$

$$\begin{aligned}\breve{x}_r^{(opt)} &= \theta_r^{(opt)}(\tilde{x}_r) = \tanh(\beta_r \cdot \tilde{x}_r) \\ \breve{x}_i^{(opt)} &= 0\end{aligned}$$

3.1.2 Evaluation of the Optimum Estimation Function

During the evaluation of Eq. (3.2), (3.3), numerical instability can take place especially for large values of β_r and β_i (very low noise). To solve this problem, an iterative procedure has been described in [10] to evaluate $\theta^{(opt)}(\cdot)$ in a numerically stable way. Because of the discrete-time iterative nature of this procedure, it is not suitable for continuous-time systems. In the following, we introduce a numerical stable method for evaluating $\theta^{(opt)}(\cdot)$, which fits continuous-time systems as well. This subsection is based mainly on [65].

Lemma 3.1 If $\Psi = \Psi^{(sp)}$ and the distance between any two consecutive elements of $\Psi^{(1)}$ and $\Psi^{(2)}$ is two, it holds:

$$\begin{aligned}\theta_r^{(opt)}(\tilde{x}_r) &\approx \sum_{j=1}^{M_r-1} \tanh\left[\beta_r \cdot \left(\tilde{x}_r - \alpha_j^{(1)}\right)\right] \\ \theta_i^{(opt)}(\tilde{x}_i) &\approx \sum_{j=1}^{M_i-1} \tanh\left[\beta_i \cdot \left(\tilde{x}_i - \alpha_j^{(2)}\right)\right]\end{aligned} \tag{3.13}$$

given that:

$$\begin{aligned}\forall j \in \{1, 2, \cdots, M_r - 1\} : \alpha_j^{(1)} &= \frac{\psi_{j+1}^{(1)} + \psi_j^{(1)}}{2} \\ \forall j \in \{1, 2, \cdots, M_i - 1\} : \alpha_j^{(2)} &= \frac{\psi_{j+1}^{(2)} + \psi_j^{(2)}}{2}.\end{aligned}$$

The larger β_r and β_i, the better the approximation.

Proof
We focus on $\theta_r^{(opt)}(\tilde{x}_r)$ and we prove its related statement for $\Psi^{(1)} = \{-3, -1, +1, +3\}$ and we show by simulation that it is valid for $\Psi^{(1)}$ with more elements. The whole analysis is then valid for $\Psi^{(2)}$.

For $\Psi^{(1)} = \{-3, -1, +1, +3\}$, $\alpha^{(1)} = \{-2, 0, +2\}$. From Eq. (3.9):

$$\theta_r^{(opt)}(\tilde{x}_r) = \frac{\exp\{4 \cdot \beta_r\} \cdot \sinh\left[\beta_r \cdot \tilde{x}_r\right] + 3 \cdot \sinh\left[3 \cdot \beta_r \cdot \tilde{x}_r\right]}{\exp\{4 \cdot \beta_r\} \cdot \cosh\left[\beta_r \cdot \tilde{x}_r\right] + \cosh\left[3 \cdot \beta_r \cdot \tilde{x}_r\right]} \tag{3.14}$$

We define according to Eq. (3.13):

$$\tilde{\theta}_r^{(opt)}(\tilde{x}_r) = \tanh\left[\beta_r \cdot (\tilde{x}_r - 2)\right] + \tanh\left[\beta_r \cdot (\tilde{x}_r + 2)\right] + \tanh\left[\beta_r \cdot \tilde{x}_r\right] \tag{3.15}$$

Using the properties of hyperbolic functions, essentially the sum and product of hyperbolic functions, we can rewrite the last relation as follows:

$$\tilde{\theta}_r^{(opt)}(\tilde{x}_r) = \frac{\left[\exp\{4 \cdot \beta_r\} + \exp\{-4 \cdot \beta_r\} + 1\right] \cdot \sinh\left[\beta_r \cdot \tilde{x}_r\right] + 3 \cdot \sinh\left[3 \cdot \beta_r \cdot \tilde{x}_r\right]}{\left[\exp\{4 \cdot \beta_r\} + \exp\{-4 \cdot \beta_r\} + 1\right] \cdot \cosh\left[\beta_r \cdot \tilde{x}_r\right] + \cosh\left[3 \cdot \beta_r \cdot \tilde{x}_r\right]} \tag{3.16}$$

If $\beta_r > 1 : \exp\{4 \cdot \beta_r\} + \exp\{-4 \cdot \beta_r\} + 1 \rightarrow \exp\{4 \cdot \beta_r\}$. In this case

$$\tilde{\theta}_r^{(opt)}(\tilde{x}_r) \approx \frac{\exp\{4 \cdot \beta_r\} \cdot \sinh\left[\beta_r \cdot \tilde{x}_r\right] + 3 \cdot \sinh\left[3 \cdot \beta_r \cdot \tilde{x}_r\right]}{\exp\{4 \cdot \beta_r\} \cdot \cosh\left[\beta_r \cdot \tilde{x}_r\right] + \cosh\left[3 \cdot \beta_r \cdot \tilde{x}_r\right]} \tag{3.17}$$

Comparing the last relation with Eq. (3.14) we conclude:

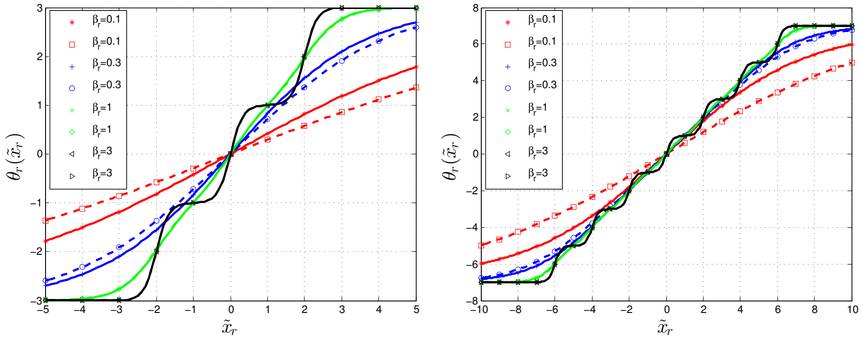
$$\theta_r^{(opt)}(\tilde{x}_r) \approx \tilde{\theta}_r^{(opt)}(\tilde{x}_r) \tag{3.18}$$

The larger β_r, the better the approximation. This is illustrated in Fig. 3.2. This analysis is valid also for $\theta_i^{(opt)}(\tilde{x}_i)$. ■

Remark 3.1 Activation functions with similar structure as in Eq. (3.13) has been considered in [48], [95].

Remark 3.2 If the distance between any two consecutive elements in $\Psi^{(1)}$ and $\Psi^{(2)}$ is different than two, it is still possible to represent $\theta_r^{(opt)}(\tilde{x}_r)$ and $\theta_i^{(opt)}(\tilde{x}_i)$ in Eq. (3.9), (3.10) as a sum of shifted hyperbolic tangent functions.

$$\begin{aligned} \theta_r^{(opt)}(\tilde{x}_r) &\approx \sum_{j=1}^{M_r-1} a_j^{(1)} \cdot \tanh\left[a_j^{(1)} \cdot \beta_r \cdot \left(\tilde{x}_r - \alpha_j^{(1)}\right)\right] \\ \theta_i^{(opt)}(\tilde{x}_i) &\approx \sum_{j=1}^{M_i-1} a_j^{(2)} \cdot \tanh\left[a_j^{(2)} \cdot \beta_i \cdot \left(\tilde{x}_i - \alpha_j^{(2)}\right)\right] \end{aligned} \tag{3.19}$$



(a) $\Psi^{(1)} = \{-3, -1, +1, +3\}$.

(b) $\Psi^{(1)} = \{-7, -5, -3, -1, +1, +3, +5, +7\}$.

Figure 3.2: Solid lines represent Eq. (3.9), dashed lines represent Eq. (3.13) for different values of β_r.

where

$$\Psi^{(1)} = \left\{\psi_1^{(1)}, \psi_2^{(1)}, \cdots, \psi_{M_r}^{(1)}\right\} \quad , \quad \Psi^{(2)} = \left\{\psi_1^{(2)}, \psi_2^{(2)}, \cdots, \psi_{M_i}^{(2)}\right\}$$

$$\forall j \in \{1, 2, \cdots, M_r - 1\} \quad : \quad \alpha_j^{(1)} = \frac{\psi_{j+1}^{(1)} + \psi_j^{(1)}}{2} \quad \text{and} \quad a_j^{(1)} = \frac{\psi_{j+1}^{(1)} - \psi_j^{(1)}}{2}$$

$$\forall j \in \{1, 2, \cdots, M_i - 1\} \quad : \quad \alpha_j^{(2)} = \frac{\psi_{j+1}^{(2)} + \psi_j^{(2)}}{2} \quad \text{and} \quad a_j^{(2)} = \frac{\psi_{j+1}^{(2)} - \psi_j^{(2)}}{2}$$

Fig. 3.3 compares between Eq. (3.9) and Eq. (3.19) for $\Psi^{(1)} = \{-5, -1, +1, +5\}$ and $\Psi^{(1)} = \{-8, -1, +1, +8\}$.

We notice that if the distance between any two consecutive elements in $\Psi^{(1)}$ and $\Psi^{(2)}$ equals two, it holds

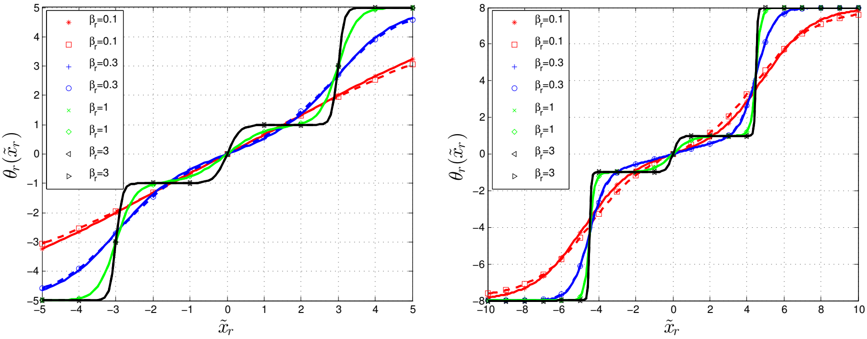
$$\forall j \in \{1, 2, \cdots, M_r - 1\} \quad : \quad \alpha_j^{(1)} = \psi_j^{(1)} + 1 \quad \text{and} \quad a_j^{(1)} = 1 \quad , \quad \alpha_{\frac{M_r}{2}}^{(1)} = 0$$

$$\forall j \in \{1, 2, \cdots, M_i - 1\} \quad : \quad \alpha_j^{(2)} = \psi_j^{(2)} + 1 \quad \text{and} \quad a_j^{(2)} = 1 \quad , \quad \alpha_{\frac{M_i}{2}}^{(2)} = 0$$

In this case Eq. (3.19) reduces to Eq. (3.13).

3.1.3 Properties of the Optimum Estimation Function

Depending on Eq. (3.19) and the properties of the hyperbolic tangent function, it can be shown that $\theta_r^{(opt)}(\tilde{x}_r)$ and $\theta_i^{(opt)}(\tilde{x}_i)$ are:



(a) $\Psi^{(1)} = \{-5, -1, +1, +5\}$.

(b) $\Psi^{(1)} = \{-8, -1, +1, +8\}$.

Figure 3.3: Solid lines represent Eq. (3.9), dashed lines represent Eq. (3.19) for different values of β_r.

- $\theta_r^{(opt)}(0) = \theta_i^{(opt)}(0) = 0$
- continuously differentiable
- bounded: $|\theta_r^{(opt)}(\tilde{x}_r)| \leq \psi_{M_r}^{(1)}$ and $|\theta_i^{(opt)}(\tilde{x}_i)| \leq \psi_{M_i}^{(2)}$
- strictly increasing. If β_r and β_i are large enough, it holds

$$\max\left\{\frac{\mathrm{d}\theta_r^{(opt)}(\tilde{x}_r)}{\mathrm{d}\tilde{x}_r}\right\} \approx \beta_r \cdot \max_j\{a_j^{(1)}\}$$

$$\max\left\{\frac{\mathrm{d}\theta_i^{(opt)}(\tilde{x}_i)}{\mathrm{d}\tilde{x}_i}\right\} \approx \beta_i \cdot \max_j\{a_j^{(2)}\}$$

The larger β_r and β_i, the better the approximation.

If the distance between any two consecutive elements in $\Psi^{(1)}$ and $\Psi^{(2)}$ equals two, the last two relations can be simplified to:

$$\max\left\{\frac{\mathrm{d}\theta_r^{(opt)}(\tilde{x}_r)}{\mathrm{d}\tilde{x}_r}\right\} \approx \beta_r$$

$$\max\left\{\frac{\mathrm{d}\theta_i^{(opt)}(\tilde{x}_i)}{\mathrm{d}\tilde{x}_i}\right\} \approx \beta_i$$

Remark 3.3 Depending on above mentioned properties of $\theta_r^{(opt)}(\tilde{x}_r)$ and $\theta_i^{(opt)}(\tilde{x}_i)$ in case of $\Psi \in \Psi^{(sp)}$ and taking into account Definition 2.11 and Lemma 2.9, we conclude that:

$$\Psi \in \Psi^{(sp)} \Rightarrow \theta^{(opt)}(\cdot) \in g^{(3)}$$

Lemma 3.2

$$\forall j \in \{1, 2, \cdots, M_r\} \;\&\; j' \in \{1, 2, \cdots, M_i\} \;:\; \theta_r^{(opt)}\left(\psi_j^{(1)}\right) \approx \psi_j^{(1)} \;\&\; \theta_i^{(opt)}\left(\psi_{j'}^{(2)}\right) \approx \psi_{j'}^{(2)}$$

Proof
From Eq. (3.7) we have:

$$\theta_r^{(opt)}(\psi_j^{(1)}) = \frac{\psi_j^{(1)} \cdot \exp\left\{\frac{1}{2} \cdot \beta_r \cdot \psi_j^{(1)\,2}\right\} + \sum_{j'=1, j' \neq j}^{M_r} \psi_{j'}^{(1)} \cdot \exp\left\{\beta_r \cdot \left(-\frac{1}{2} \cdot \psi_{j'}^{(1)\,2} + \psi_j^{(1)} \cdot \psi_{j'}^{(1)}\right)\right\}}{\exp\left\{\frac{1}{2} \cdot \beta_r \cdot \psi_j^{(1)\,2}\right\} + \sum_{j'=1, j' \neq j}^{M_r} \exp\left\{\beta_r \cdot \left(-\frac{1}{2} \cdot \psi_{j'}^{(1)\,2} + \psi_j^{(1)} \cdot \psi_{j'}^{(1)}\right)\right\}}$$

The first terms in the nominator and denominator of the last relation are dominating. we conclude:

$$\theta_r^{(opt)}\left(\psi_j^{(1)}\right) \approx \psi_j^{(1)}$$

Following the same approach we conclude that

$$\theta_i^{(opt)}\left(\psi_{j'}^{(2)}\right) \approx \psi_{j'}^{(2)}$$

The larger β_r and β_i, the better the approximation. ■

Remark 3.4 If $\sigma_r^2 \rightarrow 0$ ($\beta_r \rightarrow \infty$) and $\sigma_i^2 \rightarrow 0$ ($\beta_i \rightarrow \infty$) the optimum estimation function $\theta^{(opt)}(\cdot)$ becomes a hard decision device DECI. β_r and β_i effect the slope of the optimum estimation function $\theta^{(opt)}(\cdot)$, therefore they are referred in the following as *slope*.

3.2 Vector Equalization Based on Recurrent Neural Networks (VE-RNNs)

Since the pioneering work of Hopfield on the computational capabilities of RNNs [29], they have been applied to solve classification and optimization problems in many scientific disciplines.

This is usually done by formulating the cost function of the optimization problem to have the same structure as the Lyapunov function of the RNN and by choosing a suitable activation function. Doing so, the parameters of the RNN are defined in terms of the considered optimization problem. In this section, we do this for the vector equalization task [55].

The importance of this approach lies in the ability to avoid the training phase, which is very common for neural networks. Training of neural networks is always associated with computational effort, time, free parameter optimization etc. The ability to avoid it is an interesting feature for applications in the engineering field.

To proceed further we need to recall Fig. 1.7 and Eq. (1.10), (1.19). According to Theorem 2.8, RNN-1, RNN-2 and RNN-3 share the same fixed/equilibrium points. Thus, we decided to deal with the Lyapunov function of RNN-1 in Eq. (2.41).

3.2.1 Determination of the VE-RNNs

The maximum likelihood vector equalization rule is given by Eq. (1.19):

$$c(\boldsymbol{\xi}) = \frac{1}{2} \cdot \boldsymbol{\xi}^H \cdot \boldsymbol{R} \cdot \boldsymbol{\xi} - \Re\left\{\boldsymbol{\xi}^H \cdot \tilde{\boldsymbol{x}}\right\} \tag{3.20}$$

where $\boldsymbol{\xi}$ is a possible vector of transmit symbols. There are M^N possible vectors of transmit symbols. M is the length of the symbol alphabet. N is the length of the vector of transmit symbols. $\boldsymbol{R}$ is the channel matrix. $\tilde{\boldsymbol{x}}$ is the vector of receive symbols. $\boldsymbol{R}$ is decomposed as follows:

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{R}_d + \boldsymbol{R}_{/d} \\ \boldsymbol{R}_d &= \operatorname{diag}\{\operatorname{diag}\{\boldsymbol{R}\}\} \\ \boldsymbol{R}_{/d} &= \boldsymbol{R} - \boldsymbol{R}_d \end{aligned} \tag{3.21}$$

The diagonal elements of $\boldsymbol{R}_d$ are the same as the diagonal elements of $\boldsymbol{R}$ and the off-diagonal elements of $\boldsymbol{R}_d$ are zeros. The off-diagonal elements of $\boldsymbol{R}_{/d}$ are the same as the off-diagonal elements of $\boldsymbol{R}$ and the diagonal elements of $\boldsymbol{R}_{/d}$ are zeros. In addition $\boldsymbol{R}_{/d} = \boldsymbol{R}_{/d}^H$. In this case Eq. (3.20) can be rewritten as:

$$c(\boldsymbol{\xi}) = \frac{1}{2} \cdot \boldsymbol{\xi}^H \cdot \boldsymbol{R}_{/d} \cdot \boldsymbol{\xi} - \Re\left\{\boldsymbol{\xi}^H \cdot \tilde{\boldsymbol{x}}\right\} + \frac{1}{2} \cdot \boldsymbol{\xi}^H \cdot \boldsymbol{R}_d \cdot \boldsymbol{\xi} \tag{3.22}$$

The Lyapunov function of RNN-1 is given as, cf. Eq. (2.41)

$$E[\boldsymbol{v}] = -\frac{1}{2} \cdot \boldsymbol{v}^H \cdot \boldsymbol{D} \cdot \boldsymbol{W} \cdot \boldsymbol{v} - \Re\left\{\boldsymbol{v}^H \cdot \boldsymbol{D} \cdot \boldsymbol{W}_0 \cdot \boldsymbol{e}\right\} + \sum_{j=1}^{N} d_j \cdot \phi_j(v_j) \tag{3.23}$$

Comparing the first two terms in Eq. (3.22), (3.23) we conclude [36]:

$$\begin{array}{rl} \boldsymbol{D}\cdot\boldsymbol{W} = -\boldsymbol{R}_{/d} & \mathrm{DECI}(\boldsymbol{v}) = \boldsymbol{\xi} \quad \boldsymbol{e} = \tilde{\boldsymbol{x}} \\ \boldsymbol{D}\cdot\boldsymbol{W}_0 = \boldsymbol{I} & \qquad \boldsymbol{v} = \breve{\boldsymbol{x}} \end{array} \tag{3.24}$$

This comparison is valid also for RNN-2, RNN-3, RNN-4 and RNN-5 based on Theorem 2.8, 2.9. We notice that $\boldsymbol{D}\cdot\boldsymbol{W} \geq 0$ is not fulfilled. Therefore Theorem 2.4, 2.13 are excluded from further investigations. In addition we assume:

$$\forall j \in \{1, 2, \cdots, N\} : \varphi_j(\cdot) = \theta^{(opt)}(\cdot) \tag{3.25}$$

A quite reasonable assumption to normalize the vector of receive symbols is:

$$\boldsymbol{W}_0 = \boldsymbol{R}_d^{-1} \Rightarrow \boldsymbol{D} = \boldsymbol{R}_d > 0 \tag{3.26}$$

This justifies our proof of the stability for RNN-1 for $\boldsymbol{D} \neq \boldsymbol{I}$, cf. Remark 2.15.

In this case the parameters of the RNNs to act as vector equalizer can be defined as follows:

$$\begin{aligned} \boldsymbol{v} &= \breve{\boldsymbol{x}} \\ \mathrm{DECI}(\boldsymbol{v}) &= \boldsymbol{\xi} \\ \boldsymbol{e} &= \tilde{\boldsymbol{x}} \\ \boldsymbol{W}_0 &= \boldsymbol{R}_d^{-1} \\ \boldsymbol{W} &= \boldsymbol{I} - \boldsymbol{R}_d^{-1}\cdot\boldsymbol{R} \\ \varphi(\cdot) &= \theta^{(opt)}(\cdot) \end{aligned} \tag{3.27}$$

We notice that the diagonal elements of $\boldsymbol{W}$ are forced to be zero. This is important for RNN-2, cf. Theorems 2.2, 2.3, 2.11. Originally, this has been assumed such that the RNN-1 with high slope possesses equilibrium points near the corners of the unite hypercube $[0,1]^N$ [85]. Thus, by operating RNN-1 with high slope, it is possible to minimize the Lyapunov function Eq. (2.41) over the discrete set $\{0,1\}^N$. To proceed further, we recall Eq. (1.10) and define $\tilde{\boldsymbol{n}}_e = \boldsymbol{R}_d^{-1}\cdot\tilde{\boldsymbol{n}}$.

The dynamical behavior of RNN-1 under the above mentioned conditions, cf. Eq. (3.27) can be given as:

$$\boldsymbol{\Upsilon}\cdot\frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = -\boldsymbol{u}(t) + \boldsymbol{x} + \tilde{\boldsymbol{n}}_e + \underbrace{\left(\boldsymbol{R}_d^{-1}\cdot\boldsymbol{R} - \boldsymbol{I}\right)\cdot\left(\boldsymbol{x} - \breve{\boldsymbol{x}}(t)\right)}_{\text{residual interference}} \tag{3.28}$$

If an equilibrium point has been reached

$$\boldsymbol{u}_{ep} = \boldsymbol{x} + \tilde{\boldsymbol{n}}_e + \underbrace{\left(\boldsymbol{R}_d^{-1} \cdot \boldsymbol{R} - \boldsymbol{I}\right) \cdot \left(\boldsymbol{x} - \check{\boldsymbol{x}}_{ep}\right)}_{\text{residual interference}} \tag{3.29}$$

If a correct equalization took place $\check{\boldsymbol{x}}_{ep} = \boldsymbol{x}$:

$$\boldsymbol{u}_{ep} = \boldsymbol{x} + \tilde{\boldsymbol{n}}_e \tag{3.30}$$

The dynamical behavior of RNN-3 considering Eq. (3.27) can be given as:

$$\boldsymbol{u}(l+1) = \boldsymbol{x} + \tilde{\boldsymbol{n}}_e + \underbrace{\left(\boldsymbol{R}_d^{-1} \cdot \boldsymbol{R} - \boldsymbol{I}\right) \cdot \left(\boldsymbol{x} - \check{\boldsymbol{x}}(l)\right)}_{\text{residual interference}} \tag{3.31}$$

If a fixed point has been reached

$$\boldsymbol{u}_{fp} = \boldsymbol{x} + \tilde{\boldsymbol{n}}_e + \underbrace{\left(\boldsymbol{R}_d^{-1} \cdot \boldsymbol{R} - \boldsymbol{I}\right) \cdot \left(\boldsymbol{x} - \check{\boldsymbol{x}}_{fp}\right)}_{\text{residual interference}} \tag{3.32}$$

If a correct equalization took place $\check{\boldsymbol{x}}_{fp} = \boldsymbol{x}$:

$$\boldsymbol{u}_{fp} = \boldsymbol{x} + \tilde{\boldsymbol{n}}_e \tag{3.33}$$

$\forall j \in \{1, 2, \cdots, N\}$ the dynamical behavior of RNN-2 considering Eq. (3.27) can be given as:

$$\boldsymbol{u}_j(\rho+1) = \boldsymbol{x}_j + \tilde{\boldsymbol{n}}_{e,j} + \underbrace{\sum_{j'=1, j'\neq j}^{N} \frac{r_{jj'}}{r_{jj}} \cdot \left(\boldsymbol{x}_{j'} - \check{\boldsymbol{x}}_{j'}(\rho)\right)}_{\text{residual interference}} \tag{3.34}$$

If a fixed point has been reached

$$\boldsymbol{u}_{fp,j} = \boldsymbol{x}_j + \tilde{\boldsymbol{n}}_{e,j} + \underbrace{\sum_{j'=1, j'\neq j}^{N} \frac{r_{jj'}}{r_{jj}} \cdot \left(\boldsymbol{x}_{j'} - \check{\boldsymbol{x}}_{fp,j'}\right)}_{\text{residual interference}} \tag{3.35}$$

If a correct equalization took place $\check{\boldsymbol{x}}_{fp} = \boldsymbol{x}$:

$$\boldsymbol{u}_{fp,j} = \boldsymbol{x}_j + \tilde{\boldsymbol{n}}_{e,j} \tag{3.36}$$

If Eq. (3.30), (3.33), (3.36) are fulfilled, we obtain depending on Lemma 3.2 in the noiseless case $\check{\boldsymbol{x}} \approx \boldsymbol{x}$.

Remark 3.5 For the rest of this chapter we assume $\Psi \in \Psi^{(sp)}$. Depending on Remark 3.3 and Eq. (3.27) the stability of VE-RNNs in Chapter 2, Theorems 2.1-2.3, 2.5-2.7, 2.10-2.12 is fulfilled.

3.2.2 The Slope of the Activation Function for VE-RNNs

Depending on Eq. (3.27), we notice that all parameters of the RNN to act as vector equalizer have been fixed. The only parameter, which can be seen as a free parameter, is the slope of the activation function β_r , β_i for each neuron in the RNN. They can be defined by one of the following methods.

Constant Slope

In this case, it is assumed that β_r and β_i for each neuron in the RNN are constant during the iteration/evolution process. Theorems 2.1-2.3, 2.5-2.7 are applicable. One possibility is to relate the values of β_r and β_i with the covariance matrix of $\tilde{\boldsymbol{n}}_e$ by means of $\beta_r = \beta_i = \frac{1}{\sigma^2}$. Other possibility is to optimize the constant values of β_r and β_i such that a minimum error rate is achieved. However, the optimum value of the slope and the number of iterations (the duration of the evolution time) have to be optimized for every channel matrix $\boldsymbol{R}$. The constant slope of the activation function is especially interesting for RNN-1 because of its continuous-time nature.

Deterministic Time-Variant Slope

In this case, the slope is assumed to be time-variant during the iteration/evolution process according to a given rule. The most common rule is the linear increasing one. We have proven in Chapter 2 that the LAS stays preserved if the slope is nondecreasing [60], [66], [64]. Theorems 2.10-2.12 are applicable. The idea behind this approach is the possibility to skip local minima. This is discussed further in Sec. 3.2.5. However, the optimum increasing step related with a given number of iterations (duration of the evolution process) has to be found by simulation for each channel matrix $\boldsymbol{R}$.

Statistical Time-Variant Slope

This approach is especially interesting for the discrete-time VE-RNNs and uses the power of the residual interference in Eq. (3.31), (3.34), based on (3.4), (3.5), (3.6). For more details we refer to [10]. In this case, the slope can increase and decrease during the iteration process. Thus Theorems 2.10-2.12 are not applicable any more and the stability can not be proven. Nevertheless, simulations show that this approach leads to good results in the serial update case. However, it is computationally more demanding. This is explained further in Sec. 3.2.5.

3.2.3 The Suboptimality of VE-RNNs

The ability to apply RNNs as vector equalizer arises from the comparison between the maximum likelihood vector equalization rule and the RNNs-Lyapunov functions in the LAS case. However, VE-RNNs is a suboptimum scheme for the following reasons:

- For continuous-time RNNs and general symbol alphabets $\Psi \neq \Psi^{(sp)}$, the optimum estimation function $\theta^{(opt)}(\cdot)$ does not necessarily belong to $g^{(1)}$. For discrete-time RNNs, the LAS could not be proven for $\varphi(\cdot) \in g^{(1)}$. An exception is the case where the output of the RNNs lie on the unite circle of the complex plane (phasor model).

- The assumptions under which the optimum estimation function has been derived are not fulfilled. This includes the correlation of the noise $\tilde{\boldsymbol{n}}_e$ as well as modeling the interference from other symbols as complex-valued white Gaussian noise with statistically independent components.

- If the slope of the activation function is updated according to the statistical time-variant rule, Theorems 2.10-2.12 are not fulfilled, because the slope in this case can increase and decrease. However, simulation results show that the VE-RNNs in this case deliver the best performance compared with other slope update rules. More about this observation in Sec. 3.2.5.

- Only the first two terms of the Lyapunov functions in Theorems 2.1-2.3, 2.5-2.7, 2.10-2.12 coincide with the maximum likelihood vector equalization rule, cf. Eq. (3.22). Thus, the VE-RNNs minimizes an approximation of the maximum likelihood vector equalization rule Eq. (3.20).

  For RNN-4 & RNN-5 the inner state feedback $\boldsymbol{A}$ can be controlled such that the nonequivalence because of the above mentioned third term between the Lyapunov functions and Eq. (3.20) is minimized, because $\boldsymbol{A}$ appears in the third term of the Lyapunov functions in case of RNN-4 and RNN-5, cf. Remark 2.22.

- The maximum Likelihood vector equalization rule minimizes Eq. (3.20) globally with respect to $\boldsymbol{\xi}$, whereas the VE-RNNs minimizes an approximation of the maximum Likelihood vector equalization rule locally. More about this point in Sec. 3.2.4.

3.2.4 Globally Stable VE-RNNs

In Theorems 2.14, 2.15 the global asymptotical stability of RNN-1 and RNN-3 has been investigated, which led to the conditions given in Eq. (2.56), (2.59). Taking Eq. (3.27) into account, Eq. (2.56), (2.59) can be fulfilled only by reducing the slope of the activation function β_r, β_i [59].

If the channel matrix $\boldsymbol{R}$ does not include a lot of interference, i.e. small off-diagonal elements, Eq. (2.56), (2.59) can be fulfilled even if the slope is large. We believe that this is one of the cases, where the VE-RNNs deliver near ML-performance.

3.2.5 Time-Variant Slope and Local Minima

Locally asymptotically stable RNNs may get stuck in local minima, where the global one is desired. This is exactly the case for VE-RNNs. On the other side, globally asymptotically stable VE-RNNs may require small slopes such that no interference cancellation takes place. This motivated the idea of the time-variant slope.

At the beginning of the iteration/evolution process, the slopes are small such that the RNNs possess only one globally asymptotically stable fixed/equilibrium point, where the output of the RNN moves toward it. During the iteration/evolution process, the slopes are increased and other local minima arise. Depending on the duration of the first phase, the RNNs may avoid some local minima, which arise in the second phase and the output of the RNN reaches the attraction domain of the global minima.

However, in case of statistical time-variant slopes, the slope can increase and decrease. This enables the RNNs to skip local minima even in the second phase, which can be seen as a kind of simulated annealing [10].

3.3 Simulation Results

Simulations have been done for the following channel impulse responses [67] :

$$\begin{aligned} h_a &= [\ 0.04 \quad -0.05 \quad 0.07 \quad -0.21 \quad -0.5 \quad 0.72 \quad 0.36 \quad 0 \quad 0.21 \quad 0.03 \quad 0.07\ ] \\ h_b &= [\ 0.407 \quad 0.815 \quad 0.407\ ]. \end{aligned}$$

The first channel suffers from a little interference, whereas the second one represents a channel with a strong interference. The following simulation parameters have been assumed:

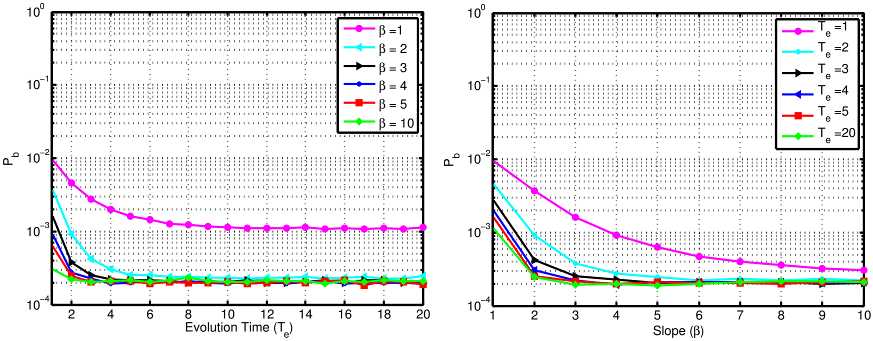
- Symbol alphabet: QPSK
- $N = 32$
- $\mathbf{Y} = \tau \cdot \boldsymbol{I}$

- $\beta_r = \beta_i = \beta$
- Sampling step for the first Euler method $\tau/\Delta t = 10$

VE-RNN-1 with Constant Slope

Fig. (3.4) shows the bit error rate (BER) vs. the slope of the activation function β and the evolution time T_e (multiple of τ) for the channel h_a at $E_b/N_o = 8$ [dB] and VE-RNN-1. We notice from Fig. 3.4 that for enough large slope and long evolution time the BER reaches a minimum and does not change any more. From Fig. 3.4, we expect that VE-RNN-1 for the channel h_a achieves almost maximum likelihood performance.

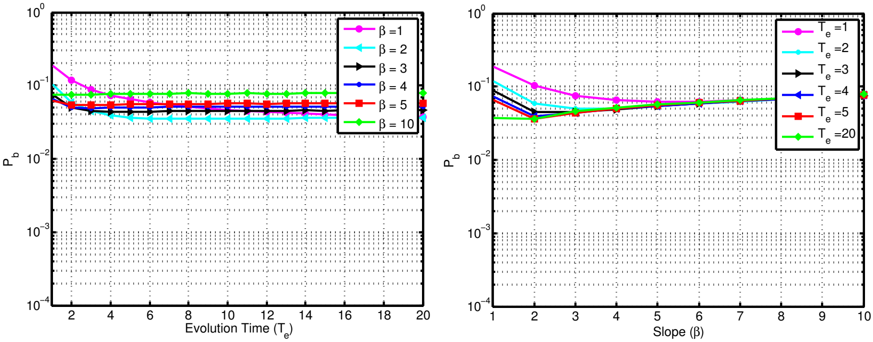
The same simulation is repeated for the channel h_b at $E_b/N_o = 15$ [dB], cf. Fig. 3.5. We observe that the VE-RNN-1 fails for the channel h_b and an error floor occurs.



(a) BER vs. the evolution time T_e (multiple of τ) for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different evolution times T_e (multiple of τ).

Figure 3.4: BER vs. the slope of the activation function β and the evolution time T_e (multiple of τ) for RNN-1, channel h_a and QPSK. $N = 32$, $E_b/N_0 = 8$ [dB].



(a) BER vs. the evolution time T_e (multiple of τ) for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different evolution times T_e (multiple of τ).

Figure 3.5: BER vs. the slope of the activation function β and the evolution time T_e (multiple of τ) for RNN-1, channel h_b and QPSK. $N = 32$, $E_b/N_0 = 15$ [dB]

VE-RNN-2 & VE-RNN-3 with Constant Slope

The BER vs. the slope of the activation function β and the number of iterations for the channel h_a at $E_b/N_o = 8$ [dB] is shown for VE-RNN-2 (serial update) in Fig. (3.6) and for VE-RNN-3 (parallel update) in Fig. (3.7).

The results have a similar behavior as explained for VE-RNN-1. However, we notice here that the improvement of the BER starts to degrade when the slope exceeds some value.

By comparing Fig. (3.6), (3.7) we notice that the difference is only at the first three iterations. In this case the BER of the VE-RNN-2 is better than the one of the VE-RNN-3.

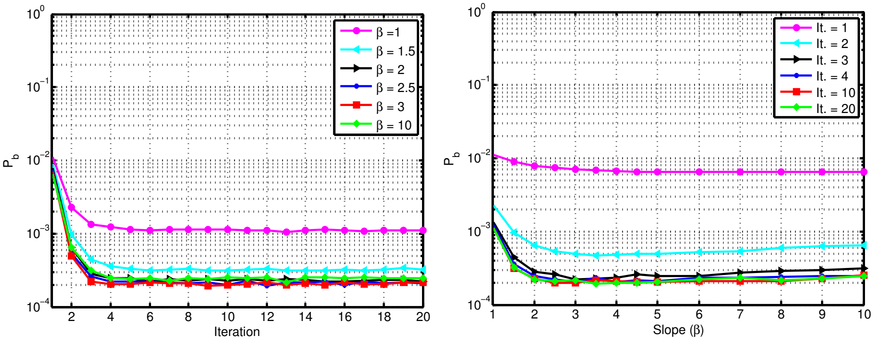
The same simulation has been done for the channel h_b at $E_b/N_o = 15$ [dB]. This is depicted in Fig. (3.8) for VE-RNN-2 and in Fig. (3.9) for VE-RNN-3. In this case, we notice that VE-RNN-3 fails for the channel h_b and a limit cycle behavior can be observed. On the other side, VE-RNN-2 delivers good results. However, the BER is sensitive to small changes of the slope.

VE-RNN-2 & VE-RNN-3 with Linear Increasing Time-Variant Slope

The BER vs. the number of iterations and the maximum time-variant linear increasing slope β_{max} of the activation function for the channel h_a at $E_b/N_o = 8$ [dB] is shown in Fig. 3.10 for VE-RNN-2 and in Fig. 3.11 for VE-RNN-3.

The same simulation has been done for the channel h_b at $E_b/N_o = 15$ [dB]. The results for VE-RNN-2 & for VE-RNN-3 are depicted in Fig. 3.12, 3.13, respectively.

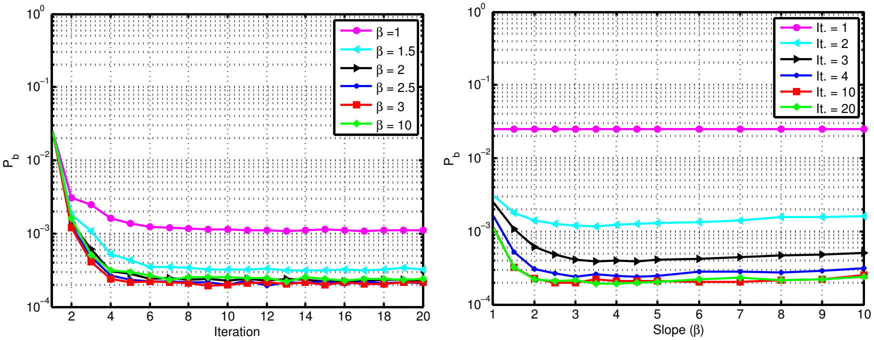
In both cases, the BER of RNN-2 is independent of β_{max} (in the simulated range) and quite few iterations are enough. For RNN-3 optimum β_{max} has been obtained.



(a) BER vs. the number of iterations for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different number of iterations.

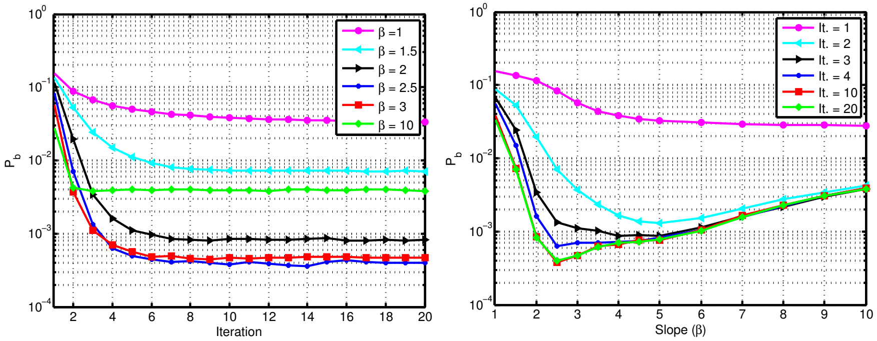
Figure 3.6: BER vs. the slope of the activation function β and the number of iterations for VE-RNN-2, channel h_a and QPSK. $N = 32$, $E_b/N_0 = 8$ [dB].



(a) BER vs. the number of iterations for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different number of iterations.

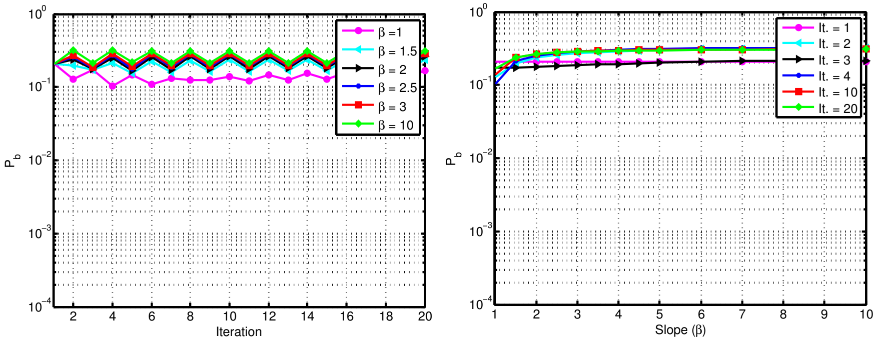
Figure 3.7: BER vs. the slope of the activation function β and the number of iterations for VE-RNN-3, channel h_a and QPSK. $N = 32$, $E_b/N_0 = 8$ [dB].



(a) BER vs. the number of iterations for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different number of iterations.

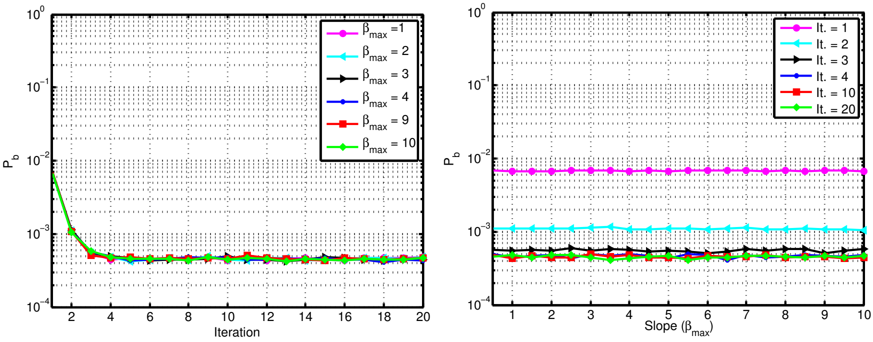
Figure 3.8: BER vs. the slope of the activation function β and the number of iterations for VE-RNN-2, channel h_b and QPSK. $N = 32$, $E_b/N_0 = 15$ [dB].



(a) BER vs. the number of iterations for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different number of iterations.

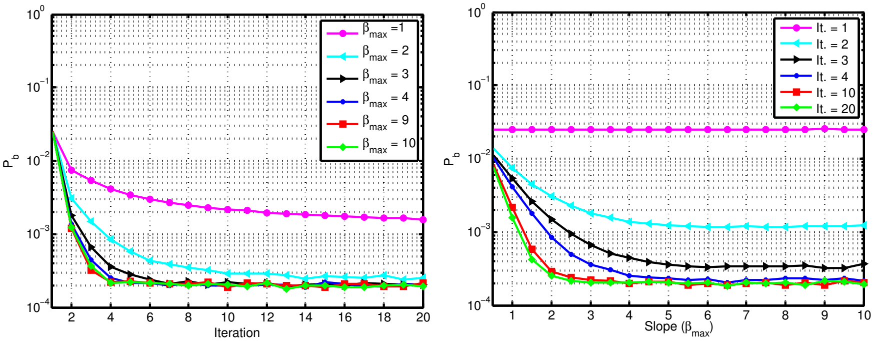
Figure 3.9: BER vs. the slope of the activation function β and the number of iterations for VE-RNN-3, channel h_b and QPSK. $N = 32$, $E_b/N_0 = 15$ [dB].



(a) BER vs. the number of iterations for different maximum time-variant linear increasing slope β_{max} of the activation function.

(b) BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function for different number of iterations.

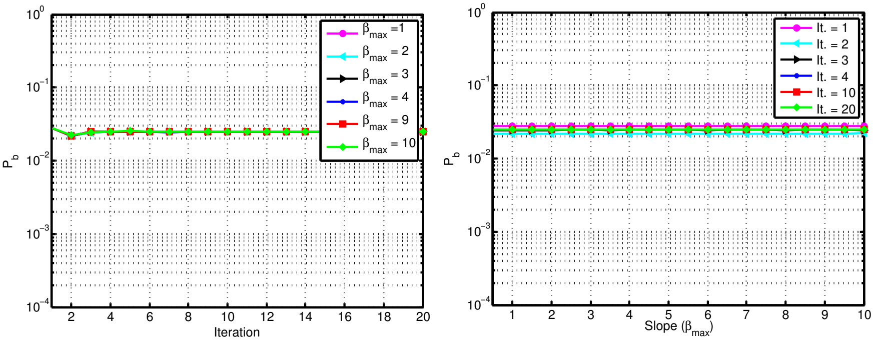
Figure 3.10: BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function and the number of iterations for VE-RNN-2, channel h_a and QPSK. $N = 32$, $E_b/N_0 = 8$ [dB].



(a) BER vs. the number of iterations for different maximum time-variant linear increasing slope β_{max} of the activation function.

(b) BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function for different number of iterations.

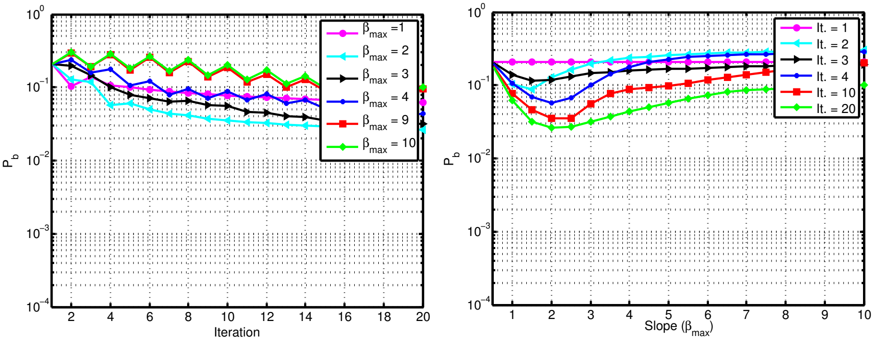
Figure 3.11: BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function and the number of iterations for VE-RNN-3, h_a channel and QPSK. $N = 32$, $E_b/N_0 = 8$ [dB].



(a) BER vs. the number of iterations for different maximum time-variant linear increasing slope β_{max} of the activation function.

(b) BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function for different number of iterations.

Figure 3.12: BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function and the number of iterations for VE-RNN-2, channel h_b and QPSK. $N = 32$, $E_b/N_0 = 15$ [dB].



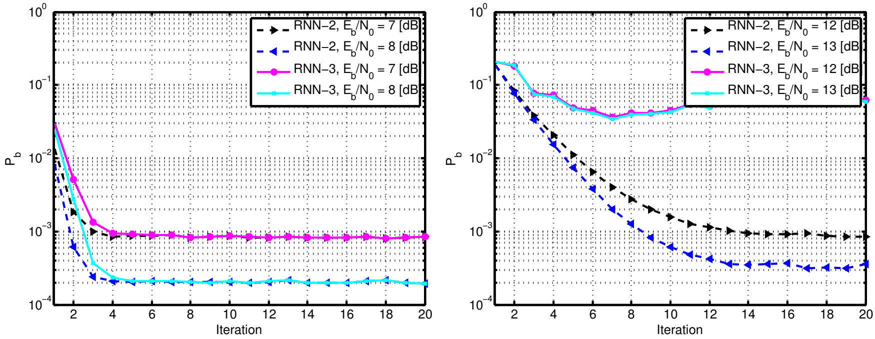
(a) BER vs. the number of iterations for different maximum time-variant linear increasing slope β_{max} of the activation function.

(b) BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function for different number of iterations.

Figure 3.13: BER vs. the maximum time-variant linear increasing slope β_{max} of the activation function and the number of iterations for VE-RNN-3, channel h_b and QPSK. $N = 32$, $E_b/N_0 = 15$ [dB].

VE-RNN-2 & VE-RNN-3 with Statistical Time-Variant Slope

The BER vs. the number of iterations for the channel h_a at $E_b/N_o = 7$ and 8 [dB] and the channel h_b at $E_b/N_o = 12$ and 13 [dB] based on VE-RNN-2 & VE-RNN-3 with statistical time-variant slope is shown in Fig. (3.14). In both cases, we notice that the serial update (VE-RNN-2) performs better than the parallel update (VE-RNN-3). VE-RNN-3 for the channel h_b seems to be unstable.



(a) BER vs. the number of iterations for different values of E_b/N_0 for VE-RNN-2 and VE-RNN-3 and the channel h_a. Statistical time-variant slope.

(b) BER vs. the number of iterations for different values of E_b/N_0 for VE-RNN-2 and VE-RNN-3 and the channel h_b. Statistical time-variant slope.

Figure 3.14: BER vs. the number of iterations for different values of E_b/N_0 for VE-RNN-2 and VE-RNN-3, Channels h_a and h_b and QPSK. Statistical time-variant slope. $N = 32$.

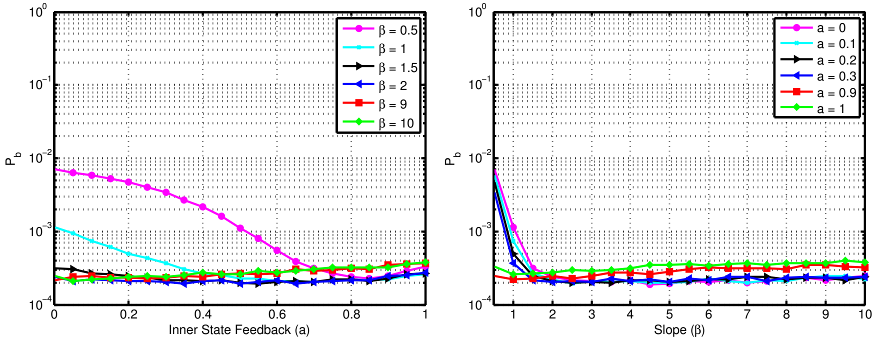
VE-RNN-4 & VE-RNN-5

The BER vs. the (constant) slope of the activation function β and the inner state feedback a for VE-RNN-4 and the channel h_a at $E_b/N_o = 8$ [dB] is depicted in Fig. 3.15. We notice that the inner state feedback for the channel h_a does not improve the results if the slope is larger than some value.

The same simulation has been done for the channel h_b at $E_b/N_o = 15$ [dB]. The results are depicted in Fig. 3.16. In contrast to the channel h_a, the inner inner state feedback improves the BER for the channel h_b, cf. Fig. 3.9. We notice that the limit cycle of length two disappear as well.

In Fig. 3.15, 3.16 the number of iterations equals 20.

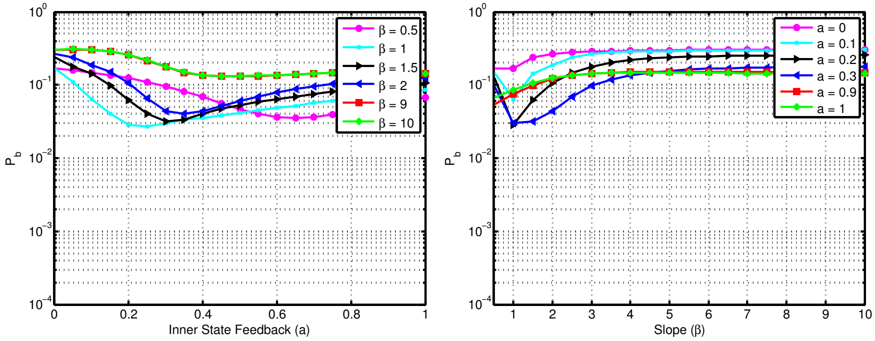
Remark 3.6 VE-RNN-5 delivers similar results as VE-RNN-4. Inner state feedback



(a) BER vs. the inner state feedback a for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different inner state feedback a.

Figure 3.15: BER vs. the slope of the activation function β and inner state feedback a for the channel h_a, QPSK and RNN-4. $N = 32$, $E_b/N_0 = 8$ [dB]. Number of iterations equals 20.



(a) BER vs. the inner state feedback a for different slopes of the activation function β.

(b) BER vs. the slope of the activation function β for different inner state feedback a.

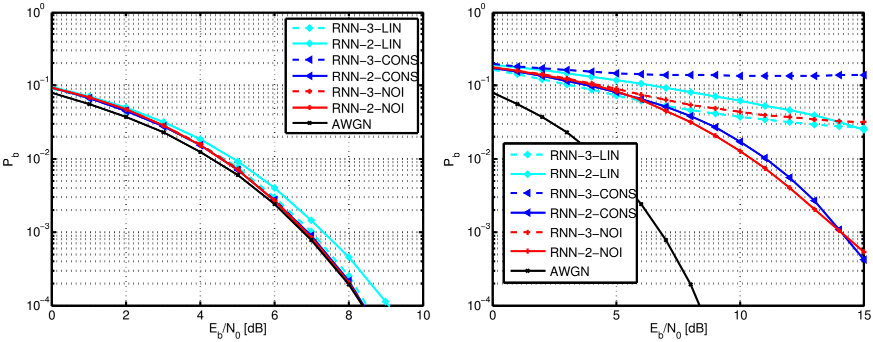
Figure 3.16: BER vs. the slope of the activation function β and inner state feedback a for the channel h_b, QPSK and RNN-4. $N = 32$, $E_b/N_0 = 15$ [dB]. Number of iterations equals 20.

has a similar influence as the time variation of the activation function. Therefore, VE-RNN-4 and VE-RNN-5 are not considered further.

BER vs. E_b/N_0

Using the optimized values for the slope β, number of iterations and duration of the evolution time T_e from the above mentioned simulations, we simulated the BER vs. E_b/N_0 based on the RNN-1, RNN-2 and RNN-3. The results are depicted in Fig. 3.17, 3.18.

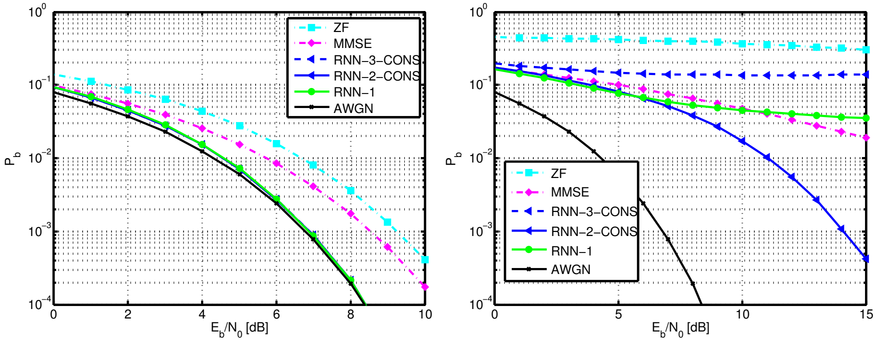
In Fig. 3.17, a comparison between VE-RNN-2 (serial update) and VE-RNN-3 (parallel update) for different types of slopes β has been performed. We notice that for the channel h_a all simulated RNNs, except VE-RNN-3 with linear time-variant slope, performs equally well. For the channel h_b, the serial update is superior to the parallel one.



(a) Channel h_a. (b) Channel h_b.

Figure 3.17: BER vs. E_b/N_0 for VE-RNN2 and VE-RNN-3 and the channels h_a, h_b with different slope types. QPSK and $N = 32$.

The comparison between discrete- and continuous-time RNNs is depicted in Fig. 3.18. As we expected before, the maximum likelihood performance is achieved for the channel h_a. For the channel h_b, an error floor does appear. This result is not surprising, since it is known that VE-RNNs are especially suitable for channels with moderate interference [46]. The continuous-time RNN performs better than discrete-time RNN with parallel update.



(a) Channel h_a.

(b) Channel h_b.

Figure 3.18: BER vs. E_b/N_0 for different RNNs for the channels h_a, h_b. QPSK and $N = 32$.

3.4 Chapter Summary

In this chapter we connected the different structures of recurrent neural networks investigated in Chapter 2 with the vector equalization task introduced in Chapter 1. This has been done by reviewing the problem of parameter estimation and using the resulted optimum estimation function as activation function for the RNNs.

By investigating the properties of the optimum estimation function for separable symbol alphabets, we have shown that the optimum estimation function belongs to the class of functions $g^{(3)}$, introduced in Chapter 2. This enables the stability proofs in Chapter 2.

A novel method to approximate the optimum estimation function in this case as a sum of shifted hyperbolic tangent functions enables a numerically stable evaluation of the optimum estimation function, even in continuous-time systems.

The parameters of the RNNs have been defined in terms of the discrete-time vector-valued transmission model such that the RNN acts as a vector equalizer.

Many methods for updating the slope of the activation function and their interrelation with the local/global minima (local/global stability) have been discussed. In addition, the suboptimality of the RNNs has been discussed. Finally, simulation results have been shown.

Chapter 4

Dynamical Representation of Probabilistic Iterative Decoding Algorithms

TURBO codes [4] and low-density parity-check (LDPC) codes [15] have been shown to achieve an error rate performance very close to the Shannon limit for the AWGN channel. Both are decoded iteratively. To achieve a better in-depth understanding of these powerful error correcting codes, they are considered as dynamical systems to utilize the well-established theory of nonlinear dynamical systems.

Most of the work on the dynamics of iterative decoding schemes is based on their discrete-time representation. However, improving the power-to-speed ratio has motivated the design of iterative decoders in the continuous-time domain (analog decoding). For this reason, several proof-of-concept analog decoders for short codes have been implemented [19], [25].

Far away from proof-of-concept, the continuous-time iterative decoder has been modeled as a first order nonlinear differential equation [22], [23], [24], [26], [27] because of its strong relation to the behavior of a passive first order low pass filter. In [27] as an example it has been shown that the analog min-sum iterative decoder in the LLR domain can be considered as a piecewise linear dynamical system. Nevertheless, the dynamics of iterative decoding algorithms in the continuous-time domain has not been attracting interest, in contrast to the discrete-time case.

In this chapter, we focus on the decoding part of the detection process. We neither construct new codes nor new decoding algorithms but we consider some already known iterative decoding algorithms, particularly belief propagation (BP) and iterative threshold decoding (ITD) as DSs (discrete and continuous-time). We are interested especially in the continuous-time representation. A connection with the high order recurrent neural networks in Chapter 2 is also established. Fig. 1.8 and Eq. (1.12) are applied.

We begin this chapter with a very brief historical overview of coding theory. More details have been already presented in Sec. 1.4. Afterward, we introduce the required tools for dealing with probabilistic iterative decoding algorithms.

The main part of this chapter focuses on formulating BP and ITD as DSs (discrete and continuous-time). This is essential for considering the stability issues and has been already done for discrete-time BP in [69]. We extend it here to the continuous-time case. The major contribution of this chapter is the formulation of ITD as a dynamical system, where close form solutions have been obtained and compared with BP for repetition codes.

The dynamical representation of iterative decoding algorithms in this chapter holds for binary linear block codes with systematic encoding. If the information word is a column vector of length k, the codeword is a column vector of length n, where the first k elements are the information word and the rest $m = n - k$ elements represent the redundancy part. However, an equivalent representation can be obtained for the nonsystematic case, too.

This chapter has been strongly influenced by [22]-[27], [69].

4.1 Introduction

Shannon's statement [74] saying that an arbitrary reliable communication over a noisy channel can be achieved if information are transmitted at a rate smaller than the channel capacity is understood to be the begin of coding theory. This includes two steps [35]. In the first one (encoding at the transmitter) a ”structured” redundancy is added to the original message and both are sent over the noisy channel. In the second step (decoding at the receiver) the redundancy is used to detect/correct possible errors in the original message. Obviously, the research in this field is twofold: How to generate the redundancy at the transmitter ”code construction” and how to use the redundancy at the receiver to detect/correct the errors ”decoding algorithm”. More about the basics of encoding/decoding process has been already introduced in Sec. 1.4. To proceed further we need to introduce Tanner graph.

Tanner Graph

The Tanner graph [79] is a bigraph. It consists of two types of nodes: variable nodes and check nodes, connected to each other by edges according to the parity check equations of the considered code. The Tanner graph visualizes in a very efficient way the parity check equations of binary linear block codes. Furthermore, it gives good insight into the code structure and the structure of iterative decoding algorithms.

Each code symbol (there exist n code symbols) is represented by a variable node and each parity check equation (there exist $m = n - k$ parity check equations) is represented by a check node.

A variable node, representing the j-th code symbol $j \in \{1, 2, \cdots, n\}$, is connected to a check node, representing the j'-th parity check equation $j' \in \{1, 2, \cdots, m\}$, if the j-th code symbol appears in the j'-th parity check equation, i.e. $\boldsymbol{H}_p(j', j) = 1$. Fig. 4.1 shows the Tanner graph of the Hamming code (7,4,3) with the parity check matrix $\boldsymbol{H}_p$:

$$\boldsymbol{H}_p = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right]$$

In the following, we define:

$$\begin{aligned} \boldsymbol{L}_{ch} &= \left[\; L_{ch,1}, L_{ch,2}, \cdots, L_{ch,n} \; \right]^T \\ \boldsymbol{f} &= \left[\; f_1, f_2, \cdots, f_{n_h} \; \right]^T \\ \boldsymbol{L} &= \left[\; L_1, L_2, \cdots, L_{n_h} \; \right]^T \\ \breve{\boldsymbol{x}} &= \left[\; \breve{x}_1, \breve{x}_2, \cdots, \breve{x}_n \; \right]^T \end{aligned} \tag{4.1}$$

n_h is the number of the nonzero elements in $\boldsymbol{H}_p$ which equals the number of the edges in the Tanner graph. In the above mentioned example $n_h = 12$.

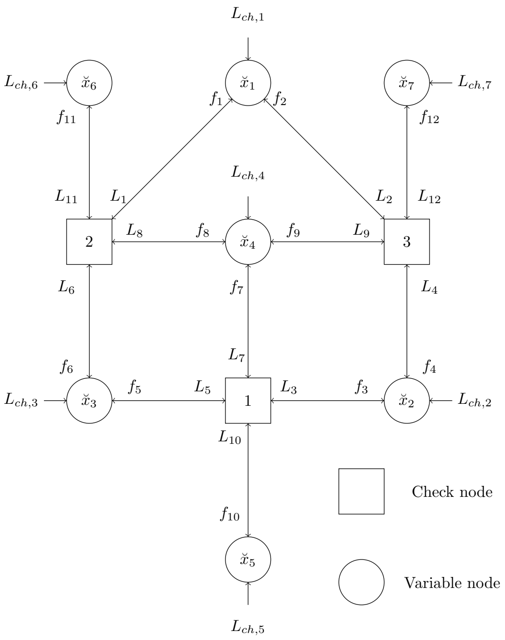


Figure 4.1: Tanner graph of the systematic Hamming code (7,4,3).

$\forall j \in \{1, 2, \cdots, n\}$: $L_{ch,j}$ is the intrinsic L-value of the j-th code symbol, which depends on the transition probability of the channel. For the AWGN-channel and BPSK modulation, cf. Eq. (1.38), (1.39), (1.40).
$\boldsymbol{f}$ is the ″message″ sent from the check nodes to the variable nodes.
$\boldsymbol{L}$ is the ″message″ sent from the variable nodes to the check nodes.
$\breve{\boldsymbol{x}}$ is the soft decided codeword.

Remark 4.1 Enumerating the entries in Eq. (4.1) and Fig. 4.1 has been done in a symbol-ascending order and for each symbol in an ascending order depending on the check nodes. Tanner graph is usually drawn, where all variable nodes are on one side and all check node on the other side. The Tanner graph in Fig. 4.1 does not follow this rule because it can be depicted without crossing edges, which is better for introductory purposes.

Definition 4.1 A *cycle* in the Tanner graph is a sequence of connected nodes which starts and ends at the same node without passing a node twice [35]. The shortest cycle of the Tanner graph of a code affects the performance of iterative decoding algorithms. Codes with cycle-free Tanner graph represent a very special case as we will see later.

4.2 Probabilistic Iterative Decoding Algorithms

Despite intensive research, the Shannon limit was unachievable till the beginning of the nineties. It is the favor of the soft iterative decoding approach introduced by turbo codes [4], which enabled extremely near Shannon limit decoding performance. Later on, it has been realized that the idea of soft iterative decoding has been already introduced in the sixties by Gallager [15]. However, the idea of Gallager sank into oblivion because of its computational complexity for that time. With the introduction of turbo codes and the huge development of computers, Gallager's work has been "rediscovered".

4.2.1 Sum-Product Decoding

The sum-product decoding, also known as belief propagation, has been first introduced by Gallager in 1962 [35]. It aims to approximate the aposteriori probability (APP) of the code symbols, i.e. approximating Eq. (1.43), by iterative exchange of the LLRs on the Tanner graph. BP delivers the exact APPs if the code (Tanner graph of the code) is cycle free [35]. We will see later that this is the case for repetition codes.

BP (and many modifications of it) depends on the iterative exchange of soft values between the variable and the check nodes as illustrated in Fig. 4.2.

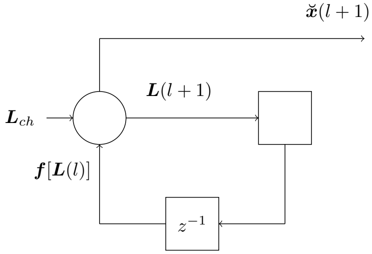

$\breve{\boldsymbol{x}}(l+1)$
$\boldsymbol{L}(l+1)$
$\boldsymbol{L}_{ch}$
$\boldsymbol{f}[\boldsymbol{L}(l)]$
z^{-1}

Figure 4.2: General structure of belief propagation (and many modifications of it).

In the following, we specify the structure in Fig. 4.2. For this purpose we define the following *binary* matrices: $\boldsymbol{S}_{n_h \times n_h}$, $\boldsymbol{P}_{n_h \times n_h}$, $\boldsymbol{B}_{n_h \times n}$. These definitions are slightly different from those in [69]. However, we are following the same approach.

$\boldsymbol{S}_{n_h \times n_h} = \left\{ s_{jj'} \in \{0,1\} : \forall j, j' \in \{1, 2, \cdots, n_h\} \right\}$. $s_{jj'} = 1$ if $j \neq j'$ & L_j, $L_{j'}$ are connected to the same check node. For example, the first row of $\boldsymbol{S}$ for the code given in Fig. 4.1 is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$\begin{aligned} f_j &= 2 \cdot \operatorname{atanh} \left\{ \prod_{j' \in \text{pos}[\boldsymbol{S}(j,:)=1]} \tanh\left(\frac{L_{j'}}{2} \right) \right\} \\ \frac{\partial f_j}{\partial L_j} &= 0 \\ \frac{\partial f_j}{\partial L_{j'}} &\in [-1, +1] \end{aligned} \tag{4.2}$$

pos $[\boldsymbol{S}(j,:) = 1]$ gives the positions of the nonzero elements in the j-th row of $\boldsymbol{S}$. Because of the last row in Eq. (4.2), f_j is a Lipschitz function [20]. Because of the systematic encoding, the low right part of $\boldsymbol{S}$ is $\boldsymbol{0}_{m \times m}$.

$\boldsymbol{P}_{n_h \times n_h} = \left\{ p_{jj'} \in \{0,1\} : \forall j, j' \in \{1, 2, \cdots, n_h\} \right\}$. $p_{jj'} = 1$ if $j \neq j'$ & f_j, $f_{j'}$ are connected to the same variable node. For an example, the seventh row of $\boldsymbol{P}$ for the code given in Fig. 4.1 is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Because of the systematic encoding, the last m-rows of $\boldsymbol{P}$ represent $\boldsymbol{0}_{m \times n_h}$. For codes where each column of the parity check matrix contains two nonzero elements, each row of $\boldsymbol{P}$ contains one nonzero element.

$\boldsymbol{B}_{n_h \times n}$: $\forall j \in \{1, 2, \cdots, n\}$ the number of nonzero elements in the j-th column equals the number of nonzero elements in the j-th column of the parity check matrix. Each row contains only one nonzero element. For an example, the first and second columns of $\boldsymbol{B}$ for the code given in Fig. 4.1 are

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Because of the systematic encoding, the low right part of $\boldsymbol{B}$ is an identity matrix of size $m \times m$.

Using the above mentioned definitions it can be shown that $\boldsymbol{B} \cdot \boldsymbol{B}^T = \boldsymbol{P} + \boldsymbol{I}$.

The Dynamical System of BP

In the discrete-time case:

$$\begin{aligned}\boldsymbol{L}(l+1) &= \boldsymbol{P}\cdot\boldsymbol{f}[\boldsymbol{L}(l)] + \boldsymbol{B}\cdot\boldsymbol{L}_{ch}\\ \breve{\boldsymbol{x}}(l+1) &= \boldsymbol{B}^{T}\cdot\boldsymbol{f}[\boldsymbol{L}(l)] + \boldsymbol{L}_{ch}\end{aligned} \tag{4.3}$$

This has been already described in [69].
In the continuous-time case according to Eq. (2.14), cf. Sec. 2.1.4:

$$\begin{aligned}\mathbf{Y}\cdot\frac{\mathrm{d}\boldsymbol{L}(t)}{\mathrm{d}t} &= -\boldsymbol{L}(t) + \boldsymbol{P}\cdot\boldsymbol{f}[\boldsymbol{L}(t)] + \boldsymbol{B}\cdot\boldsymbol{L}_{ch}\\ \breve{\boldsymbol{x}}(t) &= \boldsymbol{B}^{T}\cdot\boldsymbol{f}[\boldsymbol{L}(t)] + \boldsymbol{L}_{ch}\end{aligned} \tag{4.4}$$

Remark 4.2 Min-sum decoding is a special case of BP and is based on simplifying Eq. (4.2) as:

$$f_j = \left\{ \prod_{j'\in\mathrm{pos}[\boldsymbol{S}(j,:)=1]} \mathrm{sgn}(L_{j'}) \right\} \cdot \left\{ \min_{j'\in\mathrm{pos}[\boldsymbol{S}(j,:)=1]} \Big(|L_{j'}| \Big) \right\} \tag{4.5}$$

However, this will not be investigated further, since the analog implementation of Eq. (4.2) can be obtained directly by bipolar junction transistors [19].

4.2.2 Iterative Threshold Decoding

Threshold decoding is a kind of majority logical decoding. It was first introduced by Massey [53] for convolutional self-orthogonal codes (CSOCs). ITD is the iterative version of the threshold decoding. In this case, it is the soft decision $\breve{\boldsymbol{x}}$, instead of the LLRs $\boldsymbol{L}$, that is exchanged iteratively. In a series of papers Haccoun et. al. [6], [21] have shown that the less complex ITD achieves the same error rate performance as BP for a special class of convolution codes, namely convolution self (and self-doubly) orthogonal codes. Before introducing the dynamical system of ITD we define the following matrices:

$$\begin{aligned}\boldsymbol{w}_0^{(ul)} &= \left[w_1^{(ul)}, w_2^{(ul)} \cdots, w_{n_h-m}^{(ul)}\right]^T\\ \boldsymbol{w}_0^{(br)} &= \left[w_1^{(br)}, w_2^{(br)} \cdots, w_m^{(br)}\right]^T\\ \boldsymbol{w}_1^{(ul)} &= \left[\breve{w}_1^{(ul)}, \breve{w}_2^{(ul)} \cdots, \breve{w}_{n_h-m}^{(ul)}\right]^T\end{aligned} \tag{4.6}$$

$$\begin{aligned}
\boldsymbol{W}_{ITD,0} &= \left[\begin{array}{c|c} \operatorname{diag}\left\{\boldsymbol{w}_0^{(ul)}\right\} & \mathbf{0}_{(n_h-m)\times m} \\ \hline \mathbf{0}_{m\times(n_h-m)} & \operatorname{diag}\left\{\boldsymbol{w}_0^{(br)}\right\} \end{array}\right] \\
\boldsymbol{W}_{ITD,1} &= \left[\begin{array}{c|c} \operatorname{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} & \mathbf{0}_{(n_h-m)\times m} \\ \hline \mathbf{0}_{m\times(n_h-m)} & \mathbf{0}_{m\times m} \end{array}\right]
\end{aligned} \tag{4.7}$$

$\boldsymbol{W}_{ITD,0}$ and $\boldsymbol{W}_{ITD,1}$ are diagonal positive semidefinite matrices.

The Dynamical System of ITD

In the discrete-time case:

$$\begin{aligned}
\boldsymbol{L}(l+1) &= \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{f}[\boldsymbol{L}(l)] + \boldsymbol{W}_{ITD,0} \cdot \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\
\check{\boldsymbol{x}}(l+1) &= \boldsymbol{B}^T \cdot \boldsymbol{f}[\boldsymbol{L}(l)] + \boldsymbol{L}_{ch}
\end{aligned} \tag{4.8}$$

In the continuous-time case according to Eq. (2.14), cf. Sec. 2.1.4:

$$\begin{aligned}
\mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{L}(t)}{\mathrm{d}t} &= -\boldsymbol{L}(t) + \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{f}[\boldsymbol{L}(t)] + \boldsymbol{W}_{ITD,0} \cdot \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\
\check{\boldsymbol{x}}(t) &= \boldsymbol{B}^T \cdot \boldsymbol{f}[\boldsymbol{L}(t)] + \boldsymbol{L}_{ch}
\end{aligned} \tag{4.9}$$

Remark 4.3 Because of the systematic encoding, the structure of $\boldsymbol{W}_{ITD,1}$ prevents the variable nodes of the parity symbols in the corresponding Tanner graph from sending back exactly what they obtained from the check nodes in the previous time step. This has been assumed because it leads to compact close form solution for the stability in case of repetition codes as explained in Sec. 4.3.1. Even if this is not the case (the low right part of $\boldsymbol{W}_{ITD,1}$ is a nonzero diagonal matrix), our experience is that BER are not essentially influenced. This is further discussed in Remark 4.6.

4.3 Stability Analysis

In this section we investigate the stability of BP and ITD depending on their dynamics as described in Eq. (4.3),(4.4) and Eq. (4.8),(4.9). For this purpose we consider at first the linear case, where Eq. (4.3),(4.4) and Eq. (4.8),(4.9) become linear for special codes. After that, the investigation based on the linearization method is done as described in Sec. 2.1.3. The last step is the graphical representation of BP and ITD based on HORNNs as introduced in Chapter 2.

4.3.1 Repetition Codes: The Linear Case

This is based on [63]. If each row of the parity check matrix contains exactly two nonzero elements, then every check node in the Tanner graph of the corresponding code is connected to two variable nodes. In this special case, every row in the matrix $\boldsymbol{S}$ contains only one nonzero element and $\boldsymbol{S}$ becomes a permutation matrix. This is fulfilled in case of repetition codes, as we will see in the following.

The generator and parity check matrices of repetition codes with n code symbols, $k = 1$ information symbol, and $m = n - 1$ parity check symbols are given as follows:

$$\begin{aligned} \boldsymbol{G} &= \left[\begin{array}{c|c} 1 & \mathbf{1}_m^T \end{array} \right] \\ \boldsymbol{H}_p &= \left[\begin{array}{c|c} \mathbf{1}_m & \boldsymbol{I}_{m\times m} \end{array} \right] \end{aligned} \tag{4.10}$$

As illustrated in Fig. 4.3, the Tanner graph of repetition codes is a tree, i.e. it is cycle-free. Important to mention in this case is also

$$n_h = 2 \cdot m \Leftrightarrow n_h - m = m$$

From Fig. 4.3 and based on the definition of $\boldsymbol{P}$, $\boldsymbol{S}$ and $\boldsymbol{B}$ it can be shown that:

$$\begin{aligned} \boldsymbol{P} &= \left[\begin{array}{c|c} (\mathbf{1} - \boldsymbol{I})_{m\times m} & \mathbf{0}_{m\times m} \\ \hline \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \end{array} \right] \\ \boldsymbol{S} &= \left[\begin{array}{c|c} \mathbf{0}_{m\times m} & \boldsymbol{I}_{m\times m} \\ \hline \boldsymbol{I}_{m\times m} & \mathbf{0}_{m\times m} \end{array} \right] \\ \boldsymbol{B} &= \left[\begin{array}{c|c} \mathbf{1}_m & \mathbf{0}_{m\times m} \\ \hline \mathbf{0}_m & \boldsymbol{I}_{m\times m} \end{array} \right] \end{aligned} \tag{4.11}$$

For repetition codes, Eq. (4.2) can be rewritten as: $f_j = L_{j'}$. This has been already found in [69]. In this case, Eq. (4.3),(4.4) and Eq. (4.8),(4.9) can be rewritten as:

BP: Discrete-Time

$$\begin{aligned} \boldsymbol{L}(l+1) &= \boldsymbol{P} \cdot \boldsymbol{S} \cdot \boldsymbol{L}(l) + \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \breve{\boldsymbol{x}}(l+1) &= \boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \boldsymbol{L}(l) + \boldsymbol{L}_{ch} \end{aligned} \tag{4.12}$$

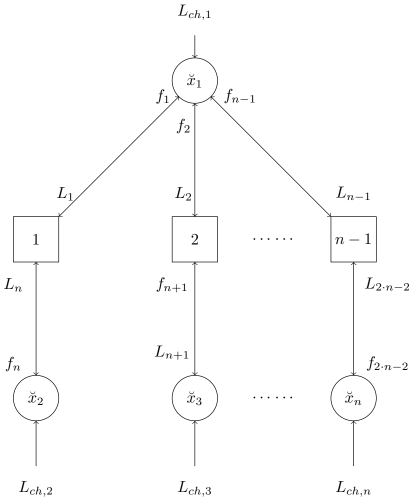


Figure 4.3: Tanner graph of repetition codes.

BP: Continuous-Time

$$\begin{aligned}\mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{L}(t)}{\mathrm{d}t} &= -\boldsymbol{L}(t) + \boldsymbol{P} \cdot \boldsymbol{S} \cdot \boldsymbol{L}(t) + \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \breve{\boldsymbol{x}}(t) &= \boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \boldsymbol{L}(t) + \boldsymbol{L}_{ch}\end{aligned} \tag{4.13}$$

The stability in this case depends on the eigenvalues of $\boldsymbol{P} \cdot \boldsymbol{S}$ where $\boldsymbol{I} - \boldsymbol{P} \cdot \boldsymbol{S}$ is non-singular, cf. Sec. 2.1.4.

BP: Fixed/Equilibrium Points

$$\begin{aligned}\boldsymbol{L}_{fp} &= \boldsymbol{L}_{ep} = [\boldsymbol{I} - \boldsymbol{P} \cdot \boldsymbol{S}]^{-1} \cdot \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \breve{\boldsymbol{x}}_{fp} &= \breve{\boldsymbol{x}}_{ep} = \left\{\boldsymbol{B}^T \cdot \boldsymbol{S} \cdot [\boldsymbol{I} - \boldsymbol{P} \cdot \boldsymbol{S}]^{-1} \cdot \boldsymbol{B} + \boldsymbol{I}\right\} \cdot \boldsymbol{L}_{ch}\end{aligned} \tag{4.14}$$

Depending on Eq. (4.11), (4.14) we find that:

$$\begin{aligned}\boldsymbol{P} \cdot \boldsymbol{S} &= \left[\begin{array}{c|c} \mathbf{0}_{m\times m} & (\mathbf{1} - \boldsymbol{I})_{m\times m} \\ \hline \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \end{array}\right] \\ [\boldsymbol{I} - \boldsymbol{P} \cdot \boldsymbol{S}]^{-1} &= \left[\begin{array}{c|c} \boldsymbol{I}_{m\times m} & (\mathbf{1} - \boldsymbol{I})_{m\times m} \\ \hline \mathbf{0}_{m\times m} & \boldsymbol{I}_{m\times m} \end{array}\right]\end{aligned} \tag{4.15}$$

and

$$\breve{\boldsymbol{x}}_{fp} = \breve{\boldsymbol{x}}_{ep} = \mathbf{1}_{n\times n} \cdot \boldsymbol{L}_{ch} \tag{4.16}$$

The hard decision of $\breve{\boldsymbol{x}}_{fp}$ (or $\breve{\boldsymbol{x}}_{ep}$) in the last relation leads necessarily to a valid codeword.

This has been already shown in [69] for the discrete-time case. Here, we have proved it using the detailed structures of $\boldsymbol{S}$, $\boldsymbol{P}$, and $\boldsymbol{B}$. In addition, we generalized this result to the continuous-time case.

The eigenvalues of a triangular matrix are the elements on the main diagonal [68]. We notice that $\boldsymbol{P} \cdot \boldsymbol{S}$ is a strictly upper triangular matrix, i.e. the eigenvalues of $\boldsymbol{P} \cdot \boldsymbol{S}$ are 0 with multiplicity $2 \cdot m$. This leads to the fact that BP for repetition codes is always (globally) stable in both discrete and continuous-time cases.

ITD: Discrete-Time

$$\begin{aligned} \boldsymbol{L}(l+1) &= \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \boldsymbol{L}(l) + \boldsymbol{W}_{ITD,0} \cdot \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \breve{\boldsymbol{x}}(l+1) &= \boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \boldsymbol{L}(l) + \boldsymbol{L}_{ch} \end{aligned} \tag{4.17}$$

ITD: Continuous-Time

$$\begin{aligned} \mathbf{Y} \cdot \frac{\mathrm{d}\boldsymbol{L}(t)}{\mathrm{d}t} &= -\boldsymbol{L}(t) + \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \boldsymbol{L}(t) + \boldsymbol{W}_{ITD,0} \cdot \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \breve{\boldsymbol{x}}(t) &= \boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \boldsymbol{L}(t) + \boldsymbol{L}_{ch} \end{aligned} \tag{4.18}$$

The stability in this case depends on the eigenvalues of $\boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S}$ where $\boldsymbol{I} - \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S}$ is nonsingular, cf. Sec. 2.1.4.

ITD: Fixed/Equilibrium Points

$$\begin{aligned} \boldsymbol{L}_{fp} = \boldsymbol{L}_{ep} &= \left[\boldsymbol{I} - \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S}\right]^{-1} \cdot \boldsymbol{W}_{ITD,0} \cdot \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \breve{\boldsymbol{x}}_{fp} = \breve{\boldsymbol{x}}_{ep} &= \left\{\boldsymbol{B}^T \cdot \boldsymbol{S} \cdot \left[\boldsymbol{I} - \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S}\right]^{-1} \cdot \boldsymbol{W}_{ITD,0} \cdot \boldsymbol{B} + \boldsymbol{I}\right\} \cdot \boldsymbol{L}_{ch} \end{aligned} \tag{4.19}$$

Depending on Eq. (4.6), (4.7), (4.11), (4.19) we find that:

$$\begin{aligned} \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S} &= \left[\begin{array}{c|c} \mathbf{0}_{m\times m} & \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m} \\ \hline \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \end{array}\right] \\ \left[\boldsymbol{I} - \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S}\right]^{-1} &= \left[\begin{array}{c|c} \boldsymbol{I}_{m\times m} & \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m} \\ \hline \mathbf{0}_{m\times m} & \boldsymbol{I}_{m\times m} \end{array}\right] \end{aligned} \tag{4.20}$$

and

$$\breve{\boldsymbol{x}}_{fp} = \breve{\boldsymbol{x}}_{ep} = \left[\begin{array}{c|c} 1 & \boldsymbol{w}_0^{(br),T} \\ \hline \boldsymbol{w}_0^{(ul)} & \boldsymbol{I}_{m\times m} + \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m} \cdot \mathrm{diag}\left\{\boldsymbol{w}_0^{(br)}\right\} \end{array}\right] \cdot \boldsymbol{L}_{ch} \tag{4.21}$$

The eigenvalues of $\boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{S}$ are 0 with multiplicity $2 \cdot m$ because it is a strictly upper triangular matrix. We conclude that ITD for repetition codes is always (globally) stable for both discrete and continuous-time cases. In contrast to BP, the hard decision of $\breve{\boldsymbol{x}}_{fp}$ (or $\breve{\boldsymbol{x}}_{ep}$) in Eq. (4.21) does not lead necessarily to a valid codeword. This depends on $\boldsymbol{W}_{ITD,0}$, $\boldsymbol{W}_{ITD,1}$.

Remark 4.4 We notice that in the linear case the weight matrices $\boldsymbol{W}_{ITD,1}$, $\boldsymbol{W}_{ITD,0}$ do not play any role for the stability. However, they define the fixed/equilibrium point itself Eq. (4.21).

Remark 4.5 Comparing Eq. (4.16) and Eq. (4.21) we notice that the fixed/equilibrium point of BP does not coincide with the fixed/equilibrium point of ITD. However, If $\boldsymbol{w}_0^{(br),T} = [1, 1, \cdots, 1]$ they coincide for the information symbol.

Remark 4.6 The compact close form solution as in Eq. (4.20), (4.21) owes to the assumption in Remark 4.3. If the low right part of $\boldsymbol{W}_{ITD,1}$ equals $\mathrm{diag}\left\{\boldsymbol{w}_1^{(br)}\right\}$, such that $\boldsymbol{w}_1^{(br)} = \left[\breve{w}_1^{(br)}, \breve{w}_2^{(br)}, \cdots, \breve{w}_m^{(br)}\right]$, Eq. (4.19) leads to:

$$\breve{\boldsymbol{x}}_{fp} = \breve{\boldsymbol{x}}_{ep} = \left[\begin{array}{c|c} \boldsymbol{A}_{ul} & \boldsymbol{A}_{ur} \\ \hline \boldsymbol{A}_{bl} & \boldsymbol{A}_{br} \end{array}\right] \cdot \boldsymbol{L}_{ch} \tag{4.22}$$

such that

$$\begin{aligned} \boldsymbol{A}_{ul} &= 1 \\ &\quad + \mathbf{1}_m^T \cdot \left[\boldsymbol{I}_{m\times m} - \mathrm{diag}\left\{\boldsymbol{w}_1^{(br)}\right\} \cdot \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m}\right]^{-1} \cdot \mathrm{diag}\left\{\boldsymbol{w}_1^{(br)}\right\} \cdot \mathrm{diag}\left\{\boldsymbol{w}_0^{(ul)}\right\} \cdot \mathbf{1}_m \\ \boldsymbol{A}_{ur} &= \mathbf{1}_m^T \cdot \left[\boldsymbol{I}_{m\times m} - \mathrm{diag}\left\{\boldsymbol{w}_1^{(br)}\right\} \cdot \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m}\right]^{-1} \mathrm{diag}\left\{\boldsymbol{w}_0^{(br)}\right\} \\ \boldsymbol{A}_{bl} &= \left[\boldsymbol{I}_{m\times m} - \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m} \cdot \mathrm{diag}\left\{\boldsymbol{w}_1^{(br)}\right\}\right]^{-1} \cdot \boldsymbol{w}_0^{(ul)} \\ \boldsymbol{A}_{br} &= \boldsymbol{I}_{m\times m} \\ &\quad + \left[\boldsymbol{I}_{m\times m} - \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m} \cdot \mathrm{diag}\left\{\boldsymbol{w}_1^{(br)}\right\}\right]^{-1} \cdot \mathrm{diag}\left\{\boldsymbol{w}_1^{(ul)}\right\} \cdot \mathbf{1}_{m\times m} \cdot \mathrm{diag}\left\{\boldsymbol{w}_0^{(br)}\right\} \end{aligned} \tag{4.23}$$

given that $\boldsymbol{I}_{m\times m} - \mathrm{diag}\Big\{\boldsymbol{w}_1^{(br)}\Big\} \cdot \mathrm{diag}\Big\{\boldsymbol{w}_1^{(ul)}\Big\} \cdot \mathbf{1}_{m\times m}$ and $\boldsymbol{I}_{m\times m} - \mathrm{diag}\Big\{\boldsymbol{w}_1^{(ul)}\Big\} \cdot \mathbf{1}_{m\times m} \cdot \mathrm{diag}\Big\{\boldsymbol{w}_1^{(br)}\Big\}$ are invertible.

If $\mathrm{diag}\Big\{\boldsymbol{w}_1^{(br)}\Big\} = \mathbf{0}_{m\times m}$, Eq. (4.22), (4.23) reduce to Eq. (4.21).

4.3.2 Stability Analysis Based on Linearization Method

This subsection is based on Sec. 2.1.3 and Sec. 2.1.4. Depending on Eq. (4.2) we conclude that the Jacobian matrix $\boldsymbol{J}_f[\boldsymbol{L}]$ has the same structure as $\boldsymbol{S}$. Both share the same "zero" entries, however the "one" entries in $\boldsymbol{S}$ are real numbers between "-1" and "+1" in $\boldsymbol{J}_f[\boldsymbol{L}]$.

BP

A fixed point $\boldsymbol{L}_{fp}$ in this case is locally asymptotically stable if $\Big|\mathrm{eig}\Big\{\boldsymbol{P} \cdot \boldsymbol{J}_f[\boldsymbol{L}_{fp}]\Big\}\Big| < 1$. This leads, according to Sec. 2.1.4, to $\Re\Big\{\mathrm{eig}\Big\{\boldsymbol{P} \cdot \boldsymbol{J}_f[\boldsymbol{L}_{ep}]\Big\}\Big\} < 0$, which means that the corresponding equilibrium point $\boldsymbol{L}_{ep}$ in the continuous-time case is also locally asymptotically stable.

ITD

A fixed point $\boldsymbol{L}_{fp}$ in this case is locally asymptotically stable if $\Big|\mathrm{eig}\Big\{\boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{J}_f[\boldsymbol{L}_{fp}]\Big\}\Big| < 1$. This leads, according to Sec. 2.1.4, to $\Re\Big\{\mathrm{eig}\Big\{\boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{J}_f[\boldsymbol{L}_{ep}]\Big\}\Big\} < 0$, which means that the corresponding equilibrium point $\boldsymbol{L}_{ep}$ in the continuous-time case is also locally asymptotically stable.

Remark 4.7 In contrast to the linear case, the weight matrices $\boldsymbol{W}_{ITD,1}$ influences the stability of the fixed/equilibrium points.

Remark 4.8 Even with the knowledge about the structure of $\boldsymbol{P} \cdot \boldsymbol{J}_f[\boldsymbol{L}]$ and $\boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T \cdot \boldsymbol{J}_f[\boldsymbol{L}]$, no close form solution of their eigenvalues have been found.

4.3.3 Stability Analysis Based on HORNNs

This is based on [57], [61]. In this subsection we reformulate Eq. (4.3), (4.4), (4.8), (4.9) such that they possess the same structure as Eq. (2.63), (2.64). After that a stability analysis based on Theorems 2.16-2.18 is done.

Based on Sec. 2.3 we define:

$$\begin{aligned} N &= n_h \\ \boldsymbol{u} &= \boldsymbol{L} \\ \boldsymbol{e} &= \boldsymbol{B} \cdot \boldsymbol{L}_{ch} \\ \varphi(\cdot) &= \tanh\left(\frac{\cdot}{2}\right) \\ \boldsymbol{v} &= \boldsymbol{\varphi}(\boldsymbol{L}) \\ f_j &= 2 \cdot \operatorname{atanh}\left\{ \prod_{j' \in \mathrm{pos}[\boldsymbol{S}(j,:)=1]} v_{j'} \right\} \end{aligned} \tag{4.24}$$

and

$$\boldsymbol{W} = \begin{cases} \boldsymbol{P} & : \text{BP} \quad (4.25\text{a}) \\ \boldsymbol{W}_{ITD,1} \cdot \boldsymbol{B} \cdot \boldsymbol{B}^T & : \text{ITD} \quad (4.25\text{b}) \end{cases}$$

In addition:

$$\boldsymbol{W}_0 = \begin{cases} \boldsymbol{I} & : \text{BP} \quad (4.26\text{a}) \\ \boldsymbol{W}_{ITD,0} & : \text{ITD} \quad (4.26\text{b}) \end{cases}$$

We notice that $\varphi \in g^{(4)}$, cf. Definition 2.12.

By comparing the stability conditions mentioned in Theorem 2.16 with Eq. (4.24)-(4.26) we notice that a scalar function $F[\boldsymbol{v}]$ must exist such that $\left\{\nabla \cdot F[\boldsymbol{v}]\right\}^T = \boldsymbol{W} \cdot \boldsymbol{f}[\boldsymbol{v}]$.

One of the most important questions, arising from this comparison and exposed to future work is: For which codes can the scalar function $F[\boldsymbol{v}]$ be found, which fulfill the condition $\left\{\nabla \cdot F[\boldsymbol{v}]\right\}^T = \boldsymbol{W} \cdot \boldsymbol{f}[\boldsymbol{v}]$ and how much powerful are they?

For Theorems 2.17, 2.18, $\boldsymbol{\Omega}^{-1} = 2 \cdot \boldsymbol{I}$ and the stability is again dependent on the Jacobian matrix of $\boldsymbol{f}[\boldsymbol{v}]$, cf. Eq. (2.67), (2.69).

4.3.4 Stability Analysis: Closing Remarks

We notice from the above mentioned discussion that the LAS of a fixed/equilibrium point depends on the eigenvalues of the Jacobian of some matrix, which is not really meaningful, because no bounds for the eigenvalues could be found, even if the structure of the Jacobian matrix is known. In addition, the convergence to a fixed point is not the sole observed behavior of iterative decoding schemes. In [38] it states:

"All the iterative decoding schemes in use today exhibit similar qualitative dynamics. In particular, a whole range of phenomena known to occur in nonlinear systems, such as existence of multiple fixed points, oscillatory behavior, bifurcation, chaos, and transit chaos are found in iterative decoding algorithms".

However, a common consensus is that the SNR value is crucial for the dynamical behavior of iterative decoding schemes. Fixed points are reached at "relatively" high SNR [38].

4.4 Simulation Results

Simulations have been done for two LDPC codes (called in the following code1 & code2) and further two tail-biting CSOCs (called in the following code3 & code4). For LDPC codes Eq. (4.3), (4.4) have been simulated. For CSOCs Eq. (4.8), (4.9) have been simulated as well. Evolution time T_e in the following is multiple of τ. Moreover, it has been assumed that $\boldsymbol{W}_{ITD,0} = \boldsymbol{I}$ and $\boldsymbol{w}_1^{(ul)} = [w_1, w_1, \cdots, w_1]^T$, cf. Eq. (4.6), (4.7). The parity check matrices of code1 & code2 have been obtained from [51]. Other properties are listed in Table 4.1. Properties of Code3 & code4 are listed in Table 4.2.

	[51]	n	k	r	column weight
code1	96.3.963	96	48	0.5	3
code2	408.33.844	408	204	0.5	3

Table 4.1: Properties of code1 & code2.

	n	k	r	memory	connected places in the shift register
code3	256	128	0.5	6	$0, 1, 4, 6$
code4	256	128	0.5	11	$0, 1, 4, 9, 11$

Table 4.2: Properties of code3 & code4.

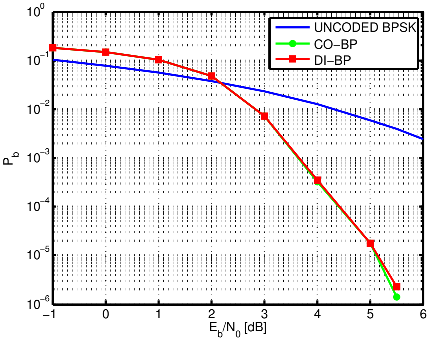
Code1 & Code2

Fig. 4.4 shows the BER performance vs. E_b/N_0 for code1 & code2 depending on Eq. (4.3), (4.4) (discrete- and continuous-time BP) and $T_e = 20 \cdot \tau$. We notice that the discrete- and continuous-time BP lead to the same BER performance. Fig. 4.5 shows the BER performance vs. evolution time/iteration for code1 & code2. We see that the BER performance improves with longer evolution time/iterations. However, the quantity of this improvement decreases with longer evolution time/iteration. Comparing Fig. 4.5(a) with Fig. 4.5(b) we notice that code2 needs longer evolution time/iteration than code1 because the codeword in this case is longer. Thus, messages need longer time to propagate on the Tanner graph.

Code3 & Code4

Simulations have been done to find the optimum value of the weight factor w_1 related with the evolution time/iteration, in both discrete- and continuous-time cases. For code3 this is depicted in Fig. 4.6. (continuous-time) and Fig. 4.7 (discrete-time). For code4 this is depicted in Fig. 4.8. (continuous-time) and Fig. 4.9 (discrete-time). We mention that both cases (discrete- and continuous-time) lead to the same optimum value of the weight factor, which is around $w_1 = 0.7$ for code3 and $w_1 = 0.9$ for code4. In addition, the longer the evolution time/iteration, the lower is the BER. This does not hold after some threshold value of evolution time/iteration.

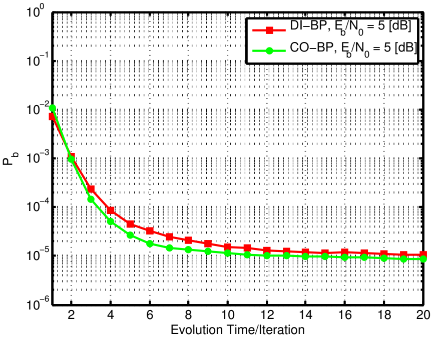
The BER vs. E_b/N_0 for code3 & code4 at the optimized weight factor is depicted in Fig. 4.10. In addition, Fig. 4.10 compares between BP and ITD, where we see that BP performs better than ITD, in both discrete- and continuous-time cases, for the simulated codes.



(a) BER vs. E_b/N_0 for code1.

(b) BER vs. E_b/N_0 for code2.

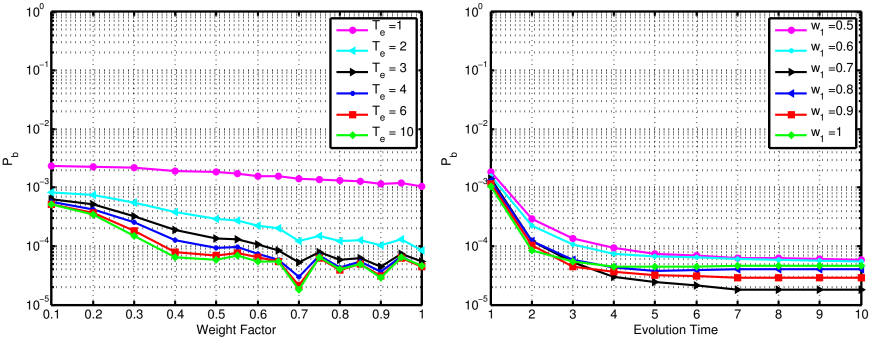
Figure 4.4: BER vs. E_b/N_0 for code1 & code 2 with evolution time equals $20 \cdot \tau$. Discrete- and continuous-time belief propagation.



(a) BER vs. evolution time/iteration for code1.

(b) BER vs. evolution time/iteration for code2.

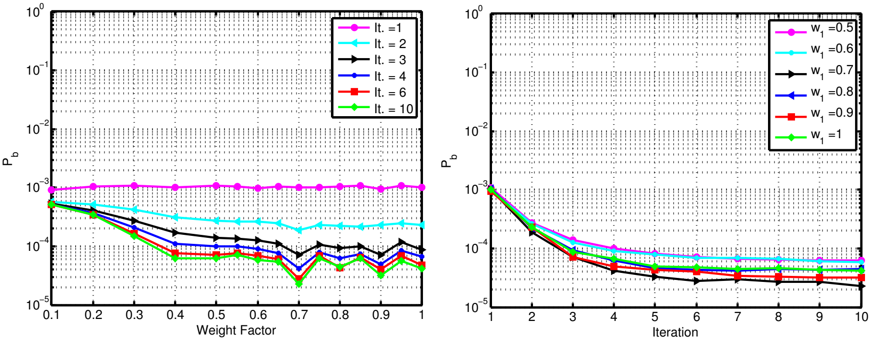
Figure 4.5: BER vs. evolution time/iteration for code1 at $E_b/N_0 = 5$ [dB] and code2 at $E_b/N_0 = 3.5$ [dB]. Discrete- and continuous-time belief propagation.



(a) BER vs. weight factor at different evolution times for code3.

(b) BER vs. evolution time at different weight factors for code3.

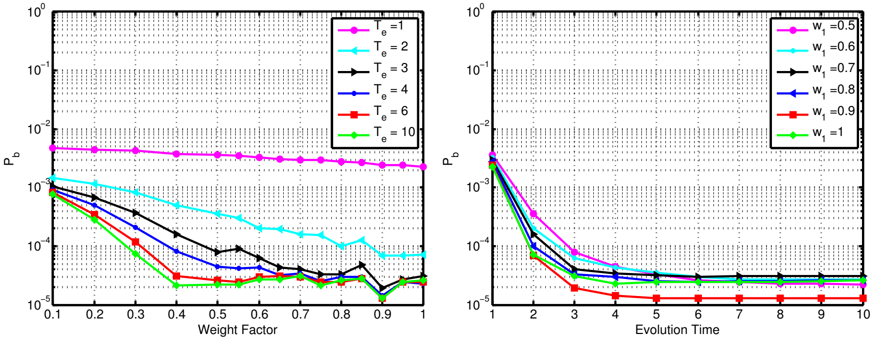
Figure 4.6: BER vs. evolution time and weight factor for code3 at $E_b/N_0 = 5.5$ [dB]. Continuous-time iterative threshold decoding.



(a) BER vs. weight factor at different iterations for code3.

(b) BER vs. number of iterations at different weight factors for code3.

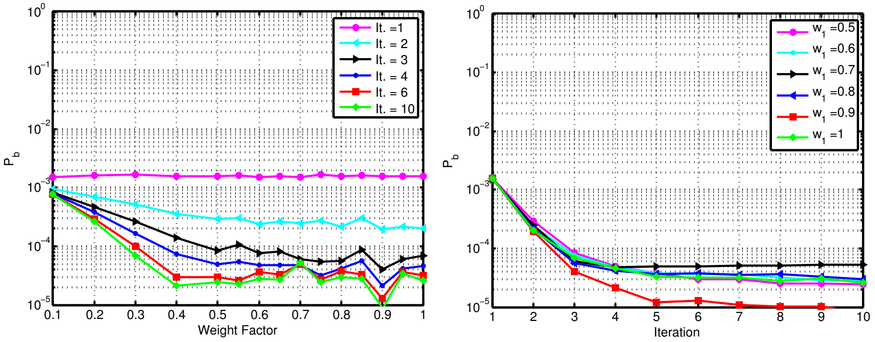
Figure 4.7: BER vs. number of iterations and weight factor for code3 at $E_b/N_0 = 5.5$ [dB]. Discrete-time iterative threshold decoding



(a) BER vs. weight factor at different evolution times for code4.

(b) BER vs. evolution time at different weight factors for code4.

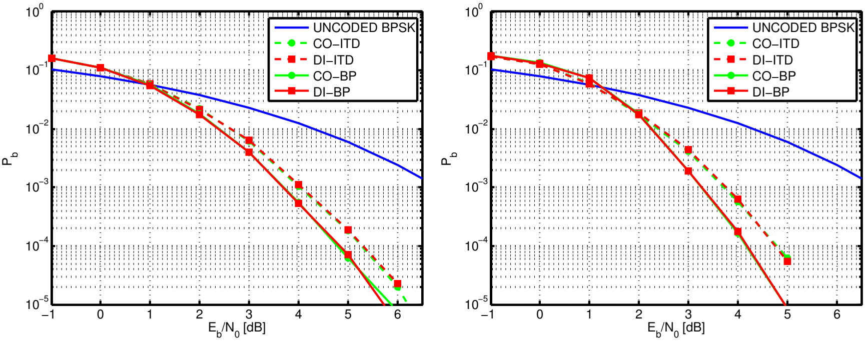
Figure 4.8: BER vs. evolution time and weight factor for code4 at $E_b/N_0 = 5.25$ [dB] Continuous-time iterative threshold decoding.



(a) BER vs. weight factor at different iterations for code4.

(b) BER vs. number of iterations at different weight factors for code4.

Figure 4.9: BER vs. number of iterations and weight factor for code4 at $E_b/N_0 = 5.25$ [dB]. Discrete-time iterative threshold decoding



(a) BER vs. E_b/N_0 for code3.

(b) BER vs. E_b/N_0 for code4.

Figure 4.10: BER vs. E_b/N_0 for code3 & code 4 for evolution time equals $10 \cdot \tau$. Belief propagation and iterative threshold decoding, discrete- and continuous-time.

4.5 Chapter Summary

In this chapter we considered two iterative decoding schemes, namely BP & ITD, as high dimensional nonlinear (discrete- and continuous-time) dynamical systems using the state-space description. A connection with HORNNs has been established as well, which might be useful from the implementation point of view.

For repetition codes, close form solutions have been obtained. In this case, both discrete- and continuous-time representations lead to the same solution.

The major contribution of this chapter is, together with Sec. 2.1.4, showing that modeling continuous-time decoders as first order nonlinear differential equation has, beside its behavior as passive first order low pass filter, important dynamical properties as well.

Chapter 5

Continuous-Time Joint Equalization and Decoding

IN Sec. 1.2 we mentioned that the maximum likelihood detection for coded transmission over nonideal channels can not be generally achieved by separating the detection process into cascade processes, namely equalization and decoding. However, the complexity of the maximum likelihood detection is too high to be implemented in real applications from computational complexity and power consumption point of view. This motivated the idea of joint equalization and decoding. In this case, cf. Fig. 1.10, a ″knowledge″ exchange takes place between the decoder and the equalizer. This improves the performance of the detection process as a whole at the cost of a realistic increase of the complexity. However, almost all research in this area is done for the discrete-time case.

Motivated by the advantages of the continuous-time realization mentioned in the introduction, we aim to give an overview about how a joint equalization and decoding can take place in the continuous-time case.

The goal of this chapter is to stimulate thoughts of how already available knowledge about joint equalization and decoding in the discrete-time case can be used for the extension to the continuous-time case, rather than introducing the whole theoretical background. This can be found in [9], [73].

5.1 Discrete- and Continuous-Time Model

A block diagram of the joint equalization and decoding in the discrete-time case is shown in Fig. 5.1 [9]. We notice that the decoder and the equalizer are exchanging some ″knowledge″ through a (de)interleaver. For the continuous-time case, we ignore the (de)interleaver.

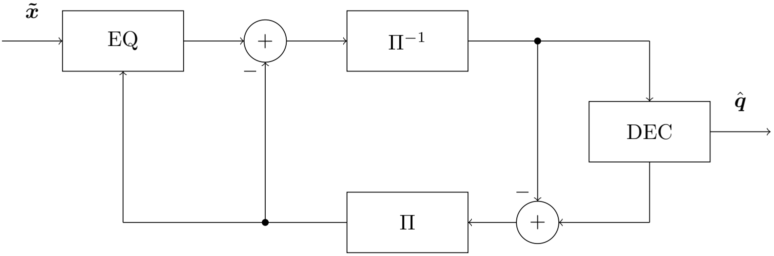


Figure 5.1: Joint equalization and decoding.

In Chapter 3 we introduced vector equalizers based on continuous-time RNNs. In Chapter 4 we introduced vector decoding based on HORNNs. All what we need to perform joint equalization and decoding in the continuous-time domain is a proper connection between the RNN and HORNN (both in continuous-time). This is depicted in Fig. 5.2, 5.3, which must be understood as a single figure.

The following explanation is for BP. However, it can be easily modified for ITD.

Depending on Fig. 5.2, 5.3, we define:

$$\begin{aligned} \boldsymbol{L}_{ch}(t) &= \boldsymbol{\theta}_{\mathrm{S/L}}[\boldsymbol{u}(t)] \\ \check{\boldsymbol{L}}(t) &= \boldsymbol{B}^T \cdot \boldsymbol{f}[\boldsymbol{L}(t)] + \boldsymbol{L}_{ch}(t) \\ \boldsymbol{v}(t) &= \boldsymbol{\theta}_{\mathrm{L/S}}[\check{\boldsymbol{L}}(t)] \end{aligned} \tag{5.1}$$

$\boldsymbol{\theta}_{\mathrm{S/L}}(\cdot)$ converts the soft symbols (inner state of the RNN) to LLRs-values (channel L-values for the decoder) depending on the symbol alphabet Ψ and the noise power (or the noise and residual interference power). The opposite duty is done by $\boldsymbol{\theta}_{\mathrm{L/S}}(\cdot)$. The exact formulation of $\boldsymbol{\theta}_{\mathrm{S/L}}(\cdot)$ and $\boldsymbol{\theta}_{\mathrm{L/S}}(\cdot)$ can be found in [73]. For BPSK, they are ”simple”, cf. Eq. (5.2). However, realizing them for higher symbol alphabets in the continuous-time domain is a challenging task.

$$\begin{aligned} \theta_{s/L}\left(u_j\right) &= \frac{2}{\sigma_n^2} \cdot u_j \\ \theta_{L/s}\left(\check{L}_j\right) &= \tanh\left(\frac{\check{L}_j}{2}\right) \end{aligned} \tag{5.2}$$

Fig. 5.2, 5.3 generalizes the discrete-time joint equalization and decoding principle suggested in [73] to the continuous-time case. The dynamical behavior is described by the following couple of differential equations, taking into account Eq. (5.1):

$$\begin{aligned} \mathbf{Y}_d \cdot \frac{\mathrm{d}\boldsymbol{L}(t)}{\mathrm{d}t} &= -\boldsymbol{L}(t) + \boldsymbol{P} \cdot \boldsymbol{f}[\boldsymbol{L}(t)] + \boldsymbol{B} \cdot \boldsymbol{L}_{ch}(t) \\ \mathbf{Y}_e \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} &= -\boldsymbol{u}(t) + \boldsymbol{W} \cdot \boldsymbol{v}(t) + \boldsymbol{W}_0 \cdot \boldsymbol{e} \end{aligned} \tag{5.3}$$

Assuming that $\mathbf{Y}_d = \tau_d \cdot \boldsymbol{I}$ & $\mathbf{Y}_e = \tau_e \cdot \boldsymbol{I}$, we distinguish:

- $\tau_d > \tau_e$ the equalizer is faster than the decoder
- $\tau_e > \tau_d$ the decoder is faster than the equalizer

The relation τ_d/τ_e is comparable in the discrete-time case to the scheduling issue. Namely, after how many iterations the equalizer (decoder) should feed its output to the decoder (equalizer). This is optimized usually by simulations and is case dependent.

The stability of the continuous-time dynamical system given by Fig. 5.2, 5.3 and Eq. (5.1), (5.3) is a very challenging topic and opposed to future work.

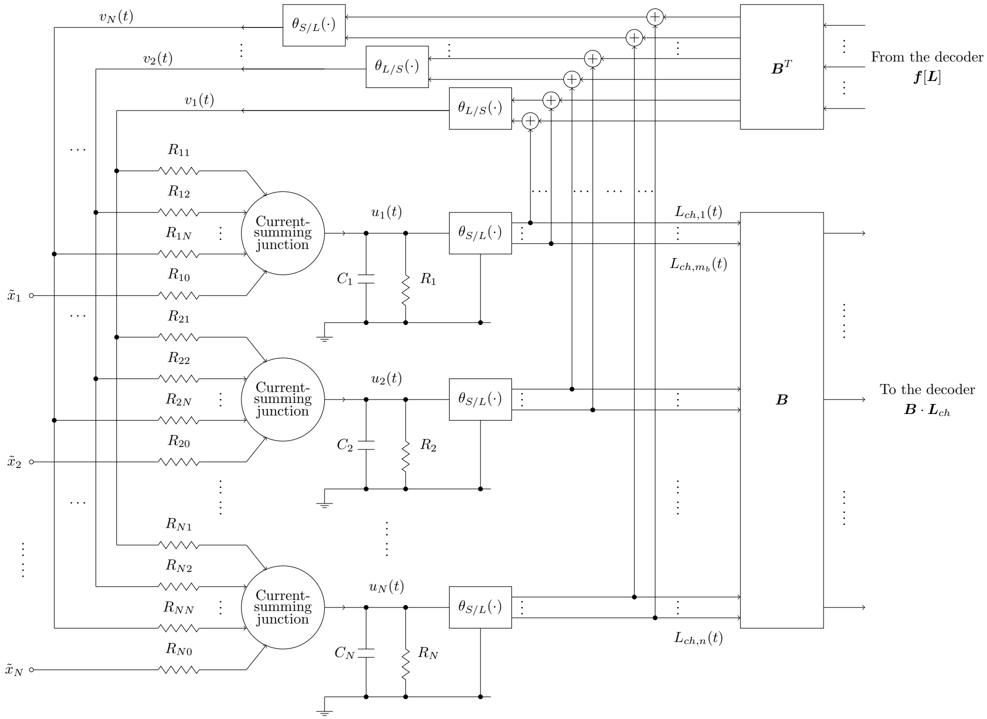


Figure 5.2: Continuous-time joint equalization and decoding. Equalization part.

Remark 5.1 From Fig. 5.2, we notice for an uncoded transmission, cf. Chapter 3, that:

$$\boldsymbol{\theta}^{(opt)}(\boldsymbol{u}) = \boldsymbol{\theta}_{\mathrm{L/S}}[\boldsymbol{\theta}_{\mathrm{S/L}}(\boldsymbol{u})]$$

Based on Eq. (5.2) and for BPSK:

$$\theta^{(opt)}(u_j) = \tanh\left(\frac{u_j}{\sigma_n^2}\right)$$

which is the optimum estimation function (activation function) for BPSK, cf. Example 3.2.

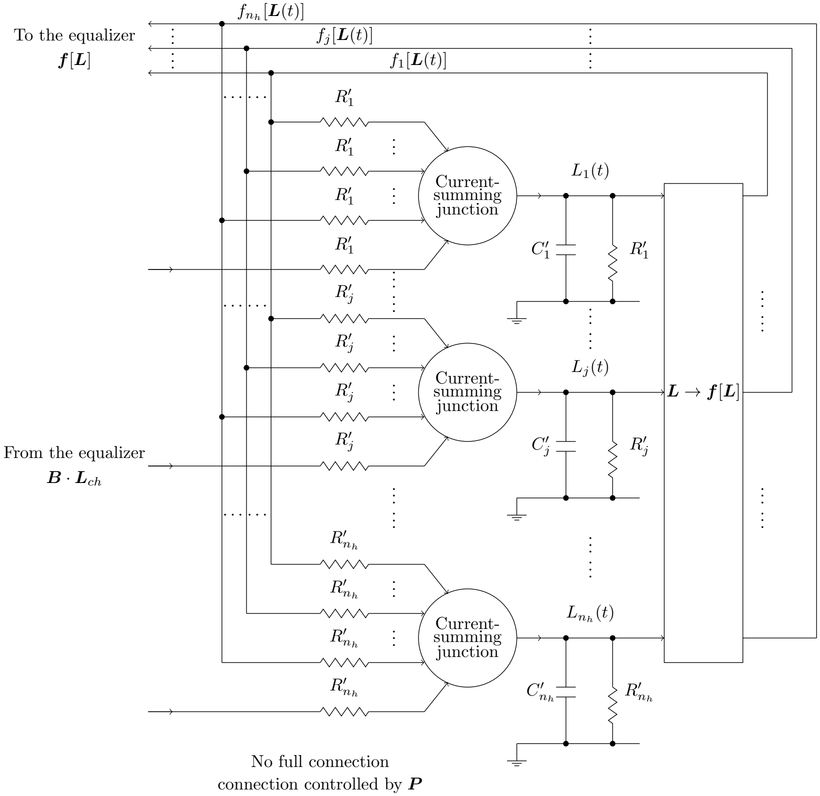


Figure 5.3: Continuous-time joint equalization and decoding. Decoding part.

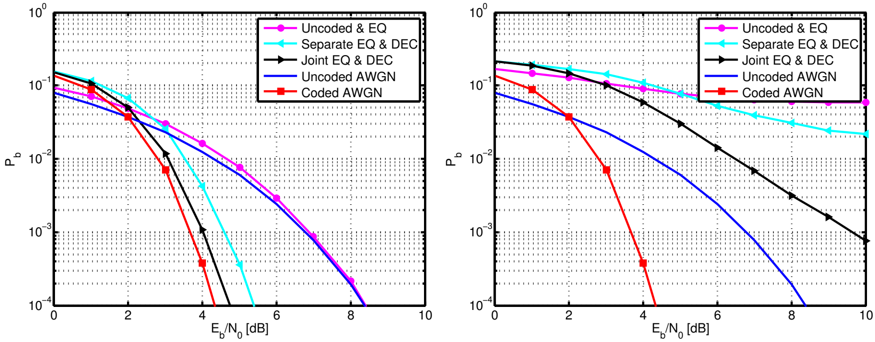
5.2 Simulation Results

We have simulated the dynamical system as in Eq. (5.1)-(5.3) and Fig. 5.2, 5.3. Moreover, the block length equals the codeword length. We assumed that $\mathbf{Y}_d = \mathbf{Y}_e = \tau \cdot \boldsymbol{I}$. We applied code1 as in Table 4.1 with the following channel impulse responses [67] (the same as in Chapter 3) :

$$h_a = [\ 0.04 \quad -0.05 \quad 0.07 \quad -0.21 \quad -0.5 \quad 0.72 \quad 0.36 \quad 0 \quad 0.21 \quad 0.03 \quad 0.07\ ]$$
$$h_b = [\ 0.407 \quad 0.815 \quad 0.407\ ]$$

In addition, we simulated the system for continuous-time separate equalization and decoding. As references we show the performance of the uncoded transmission (only continuous-time equalization) and the (un)coded AWGN-BPSK. The evolution time for the whole system is in all cases $20 \cdot \tau$. For the case of separate equalization and decoding, the evolution time of the equalization equals the evolution time of the decoding and equals $10 \cdot \tau$. The results are shown in Fig. 5.4.

We notice that the joint equalization and decoding overcomes the separate one for the same whole evolution time. For the channel h_a, the BER is close to the coded AWGN-BPSK curve. For the channel h_b, there exists a huge gap between the obtained results and the AWGN-BPSK curve. This reinforces once more the suboptimum nature of the equalization and decoding schemes considered in this work.



(a) BER vs. E_b/N_0 for h_a channel and code 1.

(b) BER vs. E_b/N_0 for h_b channel and code 1.

Figure 5.4: BER vs. E_b/N_0 for evolution time equals $20 \cdot \tau$. Continuous-time joint equalization and decoding

Red curves in Fig. 5.4 must be understood as low bounds. The simulation results in Fig. 5.4 confirm the continuous-time joint equalization and decoding concept as depicted in Fig. 5.2, 5.3. Nevertheless, substantial work is still needed to be done in this field.

Summary and Conclusions

THIS thesis has studied some iterative equalization and decoding schemes. Although this has been done in both domains, discrete- and continuous-time, our major interest was focused on the continuous-time representation because of its advantage for improving the power-to-speed ratio when implemented by an analog VLSI circuit. These advantages have been shown by many previous publications ”proof of concept”.

In this thesis, dynamical neural networks played an essential role. We did a deep analysis of their dynamics. Applying this approach, there is no need for learning (training) strategies.

The main outcomes of this thesis can be summarized as follows:

Stability Analysis

This has been done essentially in Chapter 2. Our contribution includes the definition of twin dynamical systems. In this case, a discrete- and a continuous-time dynamical system are twin if they share the same set of fixed/equilibrium points. In addition, a locally asymptotically stable fixed point of the discrete-time dynamical system represents a locally asymptotically stable equilibrium point of the continuous-time dynamical system. This definition is important, because if these fixed points are the solutions for some task, these solutions can be reached by the continuous-time dynamical system, which is twin with the original discrete-time one. In addition, many local asymptotical stability conditions for discrete-time recurrent neural networks have been extended from the real-valued case to the complex-valued one. For serial update cf. Theorem 2.2,2.3, for parallel update cf. Theorem 2.4,2.5. Additionally, the local asymptotical stability of a discrete-time recurrent neural network with parallel update and inner state feedback of memory equals two has been proven, cf. Theorem 2.6,2.7. This has been done, to the best of our knowledge, for the first time. Furthermore, the local asymptotical stability of recurrent neural networks with time-variant activation functions has been proven. The activation functions in all these cases are complex-valued, invertible and independent with respect to the real and imaginary parts.

Equalization

This has been performed in Chapter 3. The major output comprises modifying and interpreting the local and global asymptotical stability conditions of recurrent neural networks in the light of the vector equalization task. Moreover, it has been shown that the optimum estimation function for square quadrature amplitude modulation is a qualified activation function which serves all the stability conditions related with the activation functions. A numerically stable approach to evaluate the optimum estimation function for square quadrature amplitude modulation has been introduced too. This approach is especially interesting for an analog implementation. Many performance-improving heuristic schemes for vector equalization based on recurrent neural networks, as time-variant slope, have been interpreted based on the stability conditions of the corresponding recurrent neural network as well. As shown by previous publications, the vector equalizer based on recurrent neural networks with serial update has a superior performance compared with the parallel update. The vector equalizer based on continuous-time recurrent neural networks performs as well as the one based on parallel updated recurrent neural networks.

Channel Decoding

This was the topic of Chapter 4. In this case, belief propagation and iterative threshold decoding have been described as dynamical systems (discrete- and continuous-time). This description has been interpreted as a high order recurrent neural network, which might be useful during the implementation. However, the stability of these algorithms could be proven only for repetition codes, where the corresponding dynamical system becomes linear. For all other cases, the question of stability is still open. Using the definition of twin dynamical systems, the simulation results show that the bit error rate performance of belief propagation and iterative threshold decoding in the continuous-time case is as good as in the discrete-time case.

Joint Equalization and Decoding

The first steps towards continuous-time joint equalization and decoding have been performed. We have shown how to connect the continuous-time recurrent neural networks and the continuous-time high order recurrent neural networks, as investigated in Chapter 3,4, for a continuous-time joint equalization and decoding. This includes also defining all needed functions and operations. The first simulation results are encouraging.

Future Work

The work presented in this thesis can be extended for future work in different directions. We suggest:

- The stability investigation for general quadrature amplitude modulation, where the activation function is nonseparable with respect to the real and imaginary parts.

- A deeper understanding of the joint equalization and decoding based on the methods of dynamical systems is required. Utilization of EXIT charts is also envisaged.

- Continuous-time Equalization and decoding for optical transmission because of the high data rate.

Appendix $\mathcal{A}$

Mathematical Derivations

A.1 Proof of Lemma 2.1

We use the properties mentioned in Definition 2.9 and we build the Jacobian matrices of $g(u_r, u_i)$ and $g^{-1}(v_r, v_i)$. For clarity of presentation, we drop the dependency of $g(u_r, u_i)$ and $g^{-1}(v_r, v_i)$ on u_r, u_i, v_r, v_i:

$$J_g = \begin{bmatrix} \dfrac{\partial g_r}{\partial u_r} & \dfrac{\partial g_r}{\partial u_i} \\ \dfrac{\partial g_i}{\partial u_r} & \dfrac{\partial g_i}{\partial u_i} \end{bmatrix} \qquad J_{g^{-1}} = \begin{bmatrix} \dfrac{\partial g_r^{-1}}{\partial v_r} & \dfrac{\partial g_r^{-1}}{\partial v_i} \\ \dfrac{\partial g_i^{-1}}{\partial v_r} & \dfrac{\partial g_i^{-1}}{\partial v_i} \end{bmatrix} \tag{A.1}$$

The determinant of $\boldsymbol{J}_g$ is $\delta_{\boldsymbol{J}_g} = \frac{\partial g_r}{\partial u_r} \cdot \frac{\partial g_i}{\partial u_i} - \frac{\partial g_r}{\partial u_i} \cdot \frac{\partial g_i}{\partial u_r} > 0$, cf. Eq. (2.26).
The determinant of $\boldsymbol{J}_{g^{-1}}$ is $\delta_{\boldsymbol{J}_{g^{-1}}} = \frac{\partial g_r^{-1}}{\partial v_r} \cdot \frac{\partial g_i^{-1}}{\partial v_i} - \frac{\partial g_r^{-1}}{\partial v_i} \cdot \frac{\partial g_i^{-1}}{\partial v_r}$.
From [76], cf. Eq. (2.30):

$$\boldsymbol{J}_g \cdot \boldsymbol{J}_{g^{-1}} = \boldsymbol{I} \quad \Leftrightarrow \quad \boldsymbol{J}_{g^{-1}} = \boldsymbol{J}_g^{-1} \Leftrightarrow \delta_{\boldsymbol{J}_g} \cdot \delta_{\boldsymbol{J}_{g^{-1}}} = 1 \Rightarrow \delta_{\boldsymbol{J}_{g^{-1}}} > 0 \tag{A.2}$$

$$\boldsymbol{J}_g^{-1} = \frac{1}{\delta_{\boldsymbol{J}_g}} \cdot \begin{bmatrix} \frac{\partial g_i}{\partial u_i} & -\frac{\partial g_r}{\partial u_i} \\ -\frac{\partial g_i}{\partial u_r} & \frac{\partial g_r}{\partial u_r} \end{bmatrix} \tag{A.3}$$

Comparing Eq. (A.1) with Eq. (A.3), taking into account Eq. (2.24), (2.25), (A.2), leads to, cf. Eq. (2.28),(2.29),(2.31):

$$\frac{\partial g_r^{-1}}{\partial v_r} = \frac{1}{\delta_{\boldsymbol{J}_g}} \cdot \frac{\partial g_i}{\partial u_i} = \frac{1}{\delta_{\boldsymbol{J}_g} / \frac{\partial g_i}{\partial u_i}} = \left[\frac{\partial g_r}{\partial u_r} - \frac{\frac{\partial g_r}{\partial u_i} \cdot \frac{\partial g_i}{\partial u_r}}{\frac{\partial g_i}{\partial u_i}} \right]^{-1} > 0 \tag{A.4}$$

$$\frac{\partial g_i^{-1}}{\partial v_i} = \frac{1}{\delta_{\boldsymbol{J}_g}} \cdot \frac{\partial g_r}{\partial u_r} = \frac{1}{\delta_{\boldsymbol{J}_g} / \frac{\partial g_r}{\partial u_r}} = \left[\frac{\partial g_i}{\partial u_i} - \frac{\frac{\partial g_r}{\partial u_i} \cdot \frac{\partial g_i}{\partial u_r}}{\frac{\partial g_r}{\partial u_r}} \right]^{-1} > 0 \tag{A.5}$$

$$\frac{\partial g_r^{-1}}{\partial v_i} = \frac{\partial g_i^{-1}}{\partial v_r} \tag{A.6}$$

From Eq. (A.4) and using Eq. (2.24),(2.25),(2.27):

$$\min\left\{\frac{\partial g_r^{-1}}{\partial v_r}\right\} = \left[\max\left\{\frac{\partial g_r}{\partial u_r} - \frac{\dfrac{\partial g_r}{\partial u_i}\cdot\dfrac{\partial g_i}{\partial u_r}}{\dfrac{\partial g_i}{\partial u_i}}\right\}\right]^{-1}$$

$$= \left[\max\left\{\frac{\partial g_r}{\partial u_r}\right\} - \min\left\{\frac{\dfrac{\partial g_r}{\partial u_i}\cdot\dfrac{\partial g_i}{\partial u_r}}{\dfrac{\partial g_i}{\partial u_i}}\right\}\right]^{-1}$$

$$= \left[\max\left\{\frac{\partial g_r}{\partial u_r}\right\} - \frac{\min\left\{\dfrac{\partial g_r}{\partial u_i}\cdot\dfrac{\partial g_i}{\partial u_r}\right\}}{\max\left\{\dfrac{\partial g_i}{\partial u_i}\right\}}\right]^{-1}$$

$$\min\left\{\frac{\partial g_r^{-1}}{\partial v_r}\right\} = \left[\gamma_r - \frac{\min\left\{\dfrac{\partial g_r}{\partial u_i}\cdot\dfrac{\partial g_i}{\partial u_r}\right\}}{\gamma_i}\right]^{-1} \geq \frac{1}{\gamma_r}$$

Following the same way, we find from Eq. (A.5)

$$\min\left\{\frac{\partial g_i^{-1}}{\partial v_i}\right\} = \left[\gamma_i - \frac{\min\left\{\dfrac{\partial g_r}{\partial u_i}\cdot\dfrac{\partial g_i}{\partial u_r}\right\}}{\gamma_r}\right]^{-1} \geq \frac{1}{\gamma_i}$$

Special case: If $g \in g^{(2)}$:

$$\min\left\{\frac{\mathrm{d}g_r^{-1}}{\mathrm{d}v_r}\right\} = \frac{1}{\gamma_r} \quad \& \quad \min\left\{\frac{\mathrm{d}g_i^{-1}}{\mathrm{d}v_i}\right\} = \frac{1}{\gamma_i} \tag{A.7}$$

■

A.2 Proof of Lemma 2.2

We assume:

$$\phi(v_r, v_i) = \int_0^{v_r} g_r^{-1}(\vartheta_r, 0)\mathrm{d}\vartheta_r + \int_0^{v_i} g_i^{-1}(v_r, \vartheta_i)\mathrm{d}\vartheta_i \tag{A.8}$$

$$\begin{aligned}\frac{\partial\phi(v_r,v_i)}{\partial v_r} &= \frac{\partial}{\partial v_r}\int_0^{v_r} g_r^{-1}(\vartheta_r,0)\mathrm{d}\vartheta_r + \frac{\partial}{\partial v_r}\int_0^{v_i} g_i^{-1}(v_r,\vartheta_i)\mathrm{d}\vartheta_i \\ &= \frac{\partial}{\partial v_r}\int_0^{v_r} g_r^{-1}(\vartheta_r,0)\mathrm{d}\vartheta_r + \int_0^{v_i}\frac{\partial g_i^{-1}(v_r,\vartheta_i)}{\partial v_r}\,\mathrm{d}\vartheta_i\end{aligned}$$

Using Eq. (2.31):

$$\begin{aligned}\frac{\partial\phi(v_r,v_i)}{\partial v_r} &= \frac{\partial}{\partial v_r}\int_0^{v_r} g_r^{-1}(\vartheta_r,0)\mathrm{d}\vartheta_r + \int_0^{v_i}\frac{\partial g_r^{-1}(v_r,\vartheta_i)}{\partial \vartheta_i}\,\mathrm{d}\vartheta_i \\ &= g_r^{-1}(v_r,0) + g_r^{-1}(v_r,v_i) - g_r^{-1}(v_r,0) \\ &= g_r^{-1}(v_r,v_i)\end{aligned}$$

Following the same approach yields:

$$\frac{\partial\phi(v_r,v_i)}{\partial v_i} = g_i^{-1}(v_r,v_i)$$

■

A.3 Proof of Lemma 2.3

$$\begin{aligned}I_{nt} &= \Re\left\{\int_{b_1}^{b_2} g^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \\ &= \Re\left\{\int_{b_1}^{b_2}\left[g_r^{-1}(\vartheta_r,\vartheta_i) + \imath g_i^{-1}(\vartheta_r,\vartheta_i)\right]\cdot\left[\mathrm{d}\vartheta_r - \imath\mathrm{d}\vartheta_i\right]\right\} \qquad \text{(A.9)} \\ &= \int_{b_1}^{b_2}\left\{g_r^{-1}(\vartheta_r,\vartheta_i)\mathrm{d}\vartheta_r + g_i^{-1}(\vartheta_r,\vartheta_i)\mathrm{d}\vartheta_i\right\}\end{aligned}$$

Because of Eq. (2.31), the integral I_{nt} is path independent [40] and there is a function $\phi(\theta_r,\theta_i)$, cf. Lemma 2.2 and Eq. (A.8), such that:

$$\nabla\cdot\phi(\vartheta_r,\vartheta_i) = \left[g_r^{-1}(\vartheta_r,\vartheta_i) \quad g_i^{-1}(\vartheta_r,\vartheta_i)\right] \qquad \text{(A.10)}$$

Substituting Eq. (A.10) into Eq. (A.9):

$$\begin{aligned}I_{nt} &= \int_{b_1}^{b_2}\underbrace{\left\{\frac{\partial\phi(\vartheta_r,\vartheta_i)}{\partial\vartheta_r}\,\mathrm{d}\vartheta_r + \frac{\partial\phi(\vartheta_r,\vartheta_i)}{\partial\vartheta_i}\,\mathrm{d}\vartheta_i\right\}}_{\text{Total derivative}} \\ &= \int_{b_1}^{b_2}\mathrm{d}\phi(\vartheta_r,\vartheta_i) = \left[\phi(\vartheta_r,\vartheta_i)\right]_{b_1}^{b_2} = \phi(b_2) - \phi(b_1)\end{aligned}$$

According to the mean value theorem on two variables [70], there exists an intermediate point b_0 such that:

$$I_{nt} = \phi(b_2) - \phi(b_1) = \underbrace{\nabla \cdot \phi(\vartheta_r, \vartheta_i)}_{\vartheta_r = b_{0,r}\ ,\ \vartheta_i = b_{0,i}} \cdot \left[\begin{array}{c} b_{2,r} - b_{1,r} \\ b_{2,i} - b_{1,i} \end{array} \right]$$

Using Eq. (A.10) leads to:

$$I_{nt} = \Re \left\{ (b_2 - b_1)^* \cdot g^{-1}(b_0) \right\}$$

■

A.4 Proof of Lemma 2.4

The Equality holds for $v_1 = v_2$. Otherwise:

$$\Re \{ (v_1 - v_2)^* \cdot (u_1 - u_0) \} = (v_{1,r} - v_{2,r}) \cdot (u_{1,r} - u_{0,r}) + (v_{1,i} - v_{2,i}) \cdot (u_{1,i} - u_{0,i})$$

Depending on Eq. (2.24) and Definition 2.10 we distinguish :

- $v_{1,r} > v_{2,r} \Leftrightarrow u_{1,r} > u_{0,r} > u_{2,r}$
- $v_{1,r} < v_{2,r} \Leftrightarrow u_{1,r} < u_{0,r} < u_{2,r}$

In both cases $(v_{1,r} - v_{2,r}) \cdot (u_{1,r} - u_{0,r}) > 0$.

Using a similar approach based on Eq. (2.25) and Definition 2.10, we conclude that $(v_{1,i} - v_{2,i}) \cdot (u_{1,i} - u_{0,i}) > 0$.

This means $\Re \{ (v_1 - v_2)^* \cdot (u_1 - u_0) \} \geq 0$. ■

A.5 Proof of Lemma 2.5

From Lemma 2.3, Eq. (A.9) we have:

$$I_{nt} = \int_{b_{1,r}}^{b_{2,r}} g_r^{-1}(\vartheta) \mathrm{d}\vartheta + \int_{b_{1,i}}^{b_{2,i}} g_i^{-1}(\vartheta) \mathrm{d}\vartheta \tag{A.11}$$

We define new twice continuously differentiable functions:

$$G_r(v_r) = \int_0^{v_r} g_r^{-1}(\vartheta) \mathrm{d}\vartheta \quad , \quad G_i(v_i) = \int_0^{v_i} g_i^{-1}(\vartheta) \mathrm{d}\vartheta \tag{A.12}$$

Using Eq. (A.7):

$$\begin{aligned}
\min\left[\frac{\mathrm{d}^2 G_r(v_r)}{\mathrm{d}v_r^2}\right] &= \min\left[\frac{\mathrm{d}g_r^{-1}(v_r)}{\mathrm{d}v_r}\right] = \gamma_r^{-1} > 0 \\
\min\left[\frac{\mathrm{d}^2 G_i(v_i)}{\mathrm{d}v_i^2}\right] &= \min\left[\frac{\mathrm{d}g_i^{-1}(v_i)}{\mathrm{d}v_i}\right] = \gamma_i^{-1} > 0
\end{aligned} \tag{A.13}$$

This means $G_r(v_r)$, $G_i(v_i)$ are strongly convex functions [28]. Using Eq. (A.12):

$$\int_{b_{1,r}}^{b_{2,r}} g_r^{-1}(\vartheta)\mathrm{d}\vartheta = G_r(b_{2,r}) - G_r(b_{1,r}) \quad , \quad \int_{b_{1,i}}^{b_{2,i}} g_i^{-1}(\vartheta)\mathrm{d}\vartheta = G_i(b_{2,i}) - G_i(b_{1,i})$$

This leads to:

$$I_{nt} = G_r(b_{2,r}) - G_r(b_{1,r}) + G_i(b_{2,i}) - G_i(b_{1,i}) \tag{A.14}$$

Using the properties of strongly convex functions [28]:

$$\begin{aligned}
G_r(b_{2,r}) - G_r(b_{1,r}) &\leq (b_{2,r} - b_{1,r}) \cdot \left.\frac{\mathrm{d}G_r(v_r)}{\mathrm{d}v_r}\right|_{v_r = b_{2,r}} - \frac{1}{2}\cdot\gamma_r^{-1}\cdot(b_{2,r} - b_{1,r})^2 \\
&\leq (b_{2,r} - b_{1,r}) \cdot g_r^{-1}(b_{2,r}) - \frac{1}{2}\cdot\gamma_r^{-1}\cdot(b_{2,r} - b_{1,r})^2 \\
G_i(b_{2,i}) - G_i(b_{1,i}) &\leq (b_{2,i} - b_{1,i}) \cdot \left.\frac{\mathrm{d}G_i(v_i)}{\mathrm{d}v_i}\right|_{v_i = b_{2,i}} - \frac{1}{2}\cdot\gamma_i^{-1}\cdot(b_{2,i} - b_{1,i})^2 \\
&\leq (b_{2,i} - b_{1,i}) \cdot g_i^{-1}(b_{2,i}) - \frac{1}{2}\cdot\gamma_i^{-1}\cdot(b_{2,i} - b_{1,i})^2
\end{aligned}$$

This means using Eq. (A.14):

$$\begin{aligned}
\Re\left\{\int_{b_1}^{b_2} g^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \leq \Re\left\{(b_2 - b_1)^* \cdot g^{-1}(b_2)\right\} &- \frac{1}{2}\cdot\gamma_r^{-1}\cdot(b_{2,r} - b_{1,r})^2 \\
&- \frac{1}{2}\cdot\gamma_i^{-1}\cdot(b_{2,i} - b_{1,i})^2
\end{aligned}$$

■

A.6 Proof of Lemma 2.6

The equality in Eq. (2.35) holds for $u_1 = u_2$. We focus now on the case $u_1, u_2 \in \mathbb{C}$ with $u_1 \neq u_2$ and we use the properties mentioned in Definition 2.9, Eq. (2.24),(2.25), taking into account Definition 2.10.

$$\begin{aligned}
I_{eq} &= \Re\Big\{(u_1 - u_2)^* \cdot \Big[g(u_1) - g(u_2)\Big]\Big\} \\
&= (u_{1,r} - u_{2,r}) \cdot \Big[g_r(u_{1,r}) - g_r(u_{2,r})\Big] + (u_{1,i} - u_{2,i}) \cdot \Big[g_i(u_{1,i}) - g_i(u_{2,i})\Big] \\
&= (u_{1,r} - u_{2,r})^2 \cdot \underbrace{\frac{g_r(u_{1,r}) - g_r(u_{2,r})}{u_{1,r} - u_{2,r}}}_{>0} + (u_{1,i} - u_{2,i})^2 \cdot \underbrace{\frac{g_i(u_{1,i}) - g_i(u_{2,i})}{u_{1,i} - u_{2,i}}}_{>0} \\
&> 0
\end{aligned}$$

■

A.7 Proof of Lemma 2.7

A differentiable real-valued function with bounded derivative (in absolute value) by k_L is Lipschitz continuous [13]. k_L acts as Lipschitz constant. We mention that $g_r(\cdot)$ & $g_i(\cdot)$ fulfill this condition with Lipschitz constants γ_r & γ_i, respectively.
This means $\forall u_1, u_2 \in \mathbb{C}$:

$$\Big|g_r(u_{1,r}) - g_r(u_{2,r})\Big| \leq \gamma_r \cdot |u_{1,r} - u_{2,r}|$$

$$\Big|g_i(u_{1,i}) - g_i(u_{2,i})\Big| \leq \gamma_i \cdot |u_{1,i} - u_{2,i}|$$

■

A.8 Proof of Lemma 2.8

The equality in Eq. (2.36) holds for $u_1 = u_2$ and any $\eta > 0$. We focus now on the case $u_1, u_2 \in \mathbb{C}$ with $u_1 \neq u_2$ and we use the properties mentioned in Definition 2.9 taking into account Definition 2.10.

By squaring Eq. (2.36):

$$\Big|g(u_1) - g(u_2)\Big|^2 \leq \eta^2 \cdot |u_1 - u_2|^2$$

$$\Big[g_r(u_{1,r}) - g_r(u_{2,r})\Big]^2 + \Big[g_i(u_{1,i}) - g_i(u_{2,i})\Big]^2 \leq \eta^2 \cdot [u_{1,r} - u_{2,r}]^2 + \eta^2 \cdot [u_{1,i} - u_{2,i}]^2$$

The last relation is fulfilled if

$$\Big[g_r(u_{1,r}) - g_r(u_{2,r})\Big]^2 \leq \eta^2 \cdot [u_{1,r} - u_{2,r}]^2$$
$$\Big[g_i(u_{1,i}) - g_i(u_{2,i})\Big]^2 \leq \eta^2 \cdot [u_{1,i} - u_{2,i}]^2$$

Taking into account Lemma 2.7, this leads to:

$$\eta^2 \geq \gamma_r^2 \tag{A.15}$$
$$\eta^2 \geq \gamma_i^2 \tag{A.16}$$

But $\eta, \gamma_r, \gamma_i > 0 \;\Rightarrow\; \eta \geq \max\{\gamma_r, \gamma_i\}$. ■

A.9 Proof of Lemma 2.9

Obviously $\dfrac{\partial\varphi_r(u_r)}{\partial u_i} = \dfrac{\partial\varphi_i(u_i)}{\partial u_r} = 0$

- $\varphi(0) = 0$ and $\varphi(u)$ is bounded
- $\varphi_r(u_r)$, $\varphi_i(u_i)$ are continuously differentiable with respect to u_r and u_i
- $0 < \dfrac{\mathrm{d}\varphi_r(u_r)}{\mathrm{d}u_r} \leq \gamma_r$ and $0 < \dfrac{\mathrm{d}\varphi_i(u_i)}{\mathrm{d}u_i} \leq \gamma_i$ if γ_r and γ_i are larger than some γ_{th}, cf. Fig. 2.9.
- The determinant of the Jacobian matrix of the functions $\varphi(u)$ is positive:

  $$\delta_{J_\varphi} = \frac{\mathrm{d}\varphi_r(u_r)}{\mathrm{d}u_r} \cdot \frac{\mathrm{d}\varphi_r(u_i)}{\mathrm{d}u_i} > 0$$

- $\gamma_{2,r} > \gamma_{1,r}$ and $\gamma_{2,i} > \gamma_{1,i}$ then $\forall v \in \mathcal{B}$ it holds, cf. Fig. 2.11

  $$\int_0^{v_r} \varphi_{1,r}^{-1}(\vartheta)\mathrm{d}\vartheta \geq \int_0^{v_r} \varphi_{2,r}^{-1}(\vartheta)\mathrm{d}\vartheta$$
  $$\int_0^{v_i} \varphi_{1,i}^{-1}(\vartheta)\mathrm{d}\vartheta \geq \int_0^{v_i} \varphi_{2,i}^{-1}(\vartheta)\mathrm{d}\vartheta$$

  This leads to:

  $$\Re\left\{\int_0^{v} \varphi_2^{-1}(\vartheta)\mathrm{d}\vartheta^* - \int_0^{v} \varphi_1^{-1}(\vartheta)\mathrm{d}\vartheta^*\right\} \leq 0 \tag{A.17}$$

■

A.10 Proof of Lemma 2.10

$$\begin{aligned}
\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2 =& \boldsymbol{v}_{1,r}^T \cdot \boldsymbol{C}_r \cdot \boldsymbol{v}_{2,r} + \boldsymbol{v}_{1,i}^T \cdot \boldsymbol{C}_r \cdot \boldsymbol{v}_{2,i} + \boldsymbol{v}_{1,i}^T \cdot \boldsymbol{C}_i \cdot \boldsymbol{v}_{2,r} - \boldsymbol{v}_{1,r}^T \cdot \boldsymbol{C}_i \cdot \boldsymbol{v}_{2,i} \\
&+\imath\Big\{\boldsymbol{v}_{1,r}^T \cdot \boldsymbol{C}_i \cdot \boldsymbol{v}_{2,r} + \boldsymbol{v}_{1,i}^T \cdot \boldsymbol{C}_i \cdot \boldsymbol{v}_{2,i} + \boldsymbol{v}_{1,r}^T \cdot \boldsymbol{C}_r \cdot \boldsymbol{v}_{2,i} - \boldsymbol{v}_{1,i}^T \cdot \boldsymbol{C}_r \cdot \boldsymbol{v}_{2,r}\Big\} \\
\Re\left\{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\right\} =& \boldsymbol{v}_{1,r}^T \cdot \boldsymbol{C}_r \cdot \boldsymbol{v}_{2,r} + \boldsymbol{v}_{1,i}^T \cdot \boldsymbol{C}_r \cdot \boldsymbol{v}_{2,i} + \boldsymbol{v}_{1,i}^T \cdot \boldsymbol{C}_i \cdot \boldsymbol{v}_{2,r} - \boldsymbol{v}_{1,r}^T \cdot \boldsymbol{C}_i \cdot \boldsymbol{v}_{2,i} \\
=& \begin{bmatrix} \boldsymbol{v}_{1,r}^T & \boldsymbol{v}_{1,i}^T \end{bmatrix} \cdot \left[\begin{array}{c|c} \boldsymbol{C}_r & -\boldsymbol{C}_i \\ \hline \boldsymbol{C}_i & \boldsymbol{C}_r \end{array}\right] \cdot \begin{bmatrix} \boldsymbol{v}_{2,r} \\ \boldsymbol{v}_{2,i} \end{bmatrix}
\end{aligned}$$

■

A.11 Proof of Lemma 2.11

$\boldsymbol{C} = \boldsymbol{C}^H \Leftrightarrow \boldsymbol{C}_r = \boldsymbol{C}_r^T$ & $\boldsymbol{C}_i = -\boldsymbol{C}_i^T$. Depending on Lemma 2.10 assuming that $\boldsymbol{v}_1 = \boldsymbol{v}_2 = \boldsymbol{v}$, we notice that:

$$\boldsymbol{v}^H \cdot \boldsymbol{C} \cdot \boldsymbol{v} = \Big(\boldsymbol{v}^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}\Big)^* \Rightarrow \Im\Big\{\boldsymbol{v}^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}\Big\} = 0$$

This can be rewritten as $\boldsymbol{v}^H \cdot \boldsymbol{C} \cdot \boldsymbol{v} = \Re\left\{\boldsymbol{v}^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}\right\}$. ■

A.12 Proof of Lemma 2.12

Depending on Lemma 2.11

$$\begin{aligned}
I_{eq} &= -\frac{1}{2} \cdot \Re\left\{(\boldsymbol{v}_1 - \boldsymbol{v}_2)^H \cdot \boldsymbol{C} \cdot (\boldsymbol{v}_1 - \boldsymbol{v}_2)\right\} - \Re\left\{(\boldsymbol{v}_1 - \boldsymbol{v}_2)^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\right\} \\
&= -\frac{1}{2} \cdot \Re\left\{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2 + \boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1\right\} \\
&= -\frac{1}{2} \cdot \Re\left\{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\right\} - \frac{1}{2} \cdot \Re\left\{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1\right\}
\end{aligned}$$

Depending on Lemma 2.10, the second term in the last relation equals zero for $\boldsymbol{C} = \boldsymbol{C}^H$. This leads to:

$$\begin{aligned}
I_{eq} =& -\frac{1}{2} \cdot \Re\left\{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\right\} \\
=& -\frac{1}{2} \cdot \{\boldsymbol{v}_1^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_1 - \boldsymbol{v}_2^H \cdot \boldsymbol{C} \cdot \boldsymbol{v}_2\}
\end{aligned}$$

■

A.13 Proof of Lemma 2.13

This can be easily proven by taking into account that $\Re\left\{\sum_{j=1}^{N} b_j\right\} = \sum_{j=1}^{N} \Re\left\{b_j\right\}$ and $\forall b_j \in \mathbb{C} : \Re\left\{b_j\right\} \leq |b_j|$. Following the same approach, it can be proven that $\forall \boldsymbol{u} \in \mathbb{C}^N \ \&\ \boldsymbol{C} \in \mathbb{C}^{N\times N} : \Re\left\{\boldsymbol{u}^H \cdot \boldsymbol{C}\right\} \leq |\boldsymbol{u}|^T \cdot |\boldsymbol{C}|$. In this case, $|.|$ is applied elementwise. ■

A.14 Continuous-Time RNNs

Starting from Eq. (2.43) and using Eq. (2.31) we find that:

$$\begin{aligned}
\tilde{E}[\boldsymbol{v}(t)] = & -\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \left(\frac{\mathrm{d}v_{j,i}}{\mathrm{d}t}\right)^2 - \sum_{j=1}^{N} d_j \cdot \tau_j \cdot \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \left(\frac{\mathrm{d}v_{j,r}}{\mathrm{d}t}\right)^2 \cdot \left(\frac{\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}}}{\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}}}\right)^2 \\
& - \sum_{j=1}^{N} d_j \cdot \tau_j \cdot 2 \cdot \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}}}{\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}}} \\
& - \sum_{j=1}^{N} d_j \cdot \tau_j \cdot \frac{\frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,r}} \cdot \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} - \left(\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}}\right)^2}{\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}}} \cdot \left(\frac{\mathrm{d}v_{j,r}}{\mathrm{d}t}\right)^2
\end{aligned}$$

$$\begin{aligned}
\tilde{E}[\boldsymbol{v}(t)] = & -\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \left\{ \frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,i}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} + \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} + \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \left(\frac{\mathrm{d}v_{j,i}}{\mathrm{d}t}\right)^2 \right\} \\
& - \sum_{j=1}^{N} d_j \cdot \tau_j \cdot \left(\frac{\mathrm{d}v_{j,r}}{\mathrm{d}t}\right)^2 \cdot \left\{ \frac{\frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,r}} \cdot \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} - \left(\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}}\right)^2 + \left(\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}}\right)^2}{\frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}}} \right\}
\end{aligned}$$

$$\tilde{E}[\boldsymbol{v}(t)] = -\sum_{j=1}^{N} d_j \cdot \tau_j \cdot \left\{ \frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,r}} \cdot \left(\frac{\mathrm{d}v_{j,r}}{\mathrm{d}t}\right)^2 + \frac{\partial \varphi_{j,r}^{-1}}{\partial v_{j,i}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} \right.$$
$$\left. + \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,r}} \cdot \frac{\mathrm{d}v_{j,r}}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{j,i}}{\mathrm{d}t} + \frac{\partial \varphi_{j,i}^{-1}}{\partial v_{j,i}} \cdot \left(\frac{\mathrm{d}v_{j,i}}{\mathrm{d}t}\right)^2 \right\}$$

By comparing the last relation with Eq. (2.42), we find that:

$$\frac{\mathrm{d}E[\boldsymbol{v}(t)]}{\mathrm{d}t} = \tilde{E}[\boldsymbol{v}(t)]$$

A.15 Serial updated RNNs

$$E[\boldsymbol{v}(\rho)] = -\frac{1}{2} \cdot \sum_{j''=1}^{N} \sum_{j'=1}^{N} d_{j''} \cdot w_{j''j'} \cdot v_{j''}^{*}(\rho) \cdot v_{j'}(\rho) - \Re\left\{ \sum_{j'=1}^{N} d_{j'} \cdot v_{j'}^{*}(\rho) \cdot w_{j'0} \cdot e_{j'} \right\}$$
$$+ \Re\left\{ \sum_{j'=1}^{N} d_{j'} \cdot \int_0^{v_{j'}(\rho)} \varphi_{j'}^{-1}(\vartheta)\mathrm{d}\vartheta^{*} \right\}$$

$$E[\boldsymbol{v}(\rho+1)] = -\frac{1}{2} \cdot \sum_{j''=1}^{N} \sum_{j'=1}^{N} d_{j''} \cdot w_{j''j'} \cdot v_{j''}^{*}(\rho+1) \cdot v_{j'}(\rho+1)$$
$$- \Re\left\{ \sum_{j'=1}^{N} d_{j'} \cdot v_{j'}^{*}(\rho+1) \cdot w_{j'0} \cdot e_{j'} \right\} + \Re\left\{ \sum_{j'=1}^{N} d_{j'} \cdot \int_0^{v_{j'}(\rho+1)} \varphi_{j'}^{-1}(\vartheta)\mathrm{d}\vartheta^{*} \right\}$$

Taking into account that $\boldsymbol{D}\cdot\boldsymbol{W} = \{\boldsymbol{D}\cdot\boldsymbol{W}\}^{H}$

$$\begin{aligned}
E[\boldsymbol{v}(\rho)] = &-\frac{1}{2}\cdot\sum_{\substack{j''=1\\ j''\neq j}}^{N}\sum_{\substack{j'=1\\ j'\neq j}}^{N} d_{j''}\cdot w_{j''j'}\cdot v^{*}_{j''}(\rho)\cdot v_{j'}(\rho) - \Re\left\{\sum_{\substack{j'=1\\ j'\neq j}}^{N} d_{j'}\cdot v^{*}_{j'}(\rho)\cdot w_{j'0}\cdot e_{j'}\right\}\\
&+\Re\left\{\sum_{\substack{j'=1\\ j'\neq j}}^{N} d_{j'}\cdot\int_{0}^{v_{j'}(\rho)}\varphi_{j'}^{-1}(\vartheta)\mathrm{d}\vartheta^{*}\right\}+\Re\left\{d_{j}\cdot\int_{0}^{v_{j}(\rho)}\varphi_{j}^{-1}(\vartheta)\mathrm{d}\vartheta^{*}\right\}\\
&-\Re\left\{v^{*}_{j}(\rho)\cdot d_{j}\cdot\Big[\sum_{\substack{j'=1\\ j'\neq j}}^{N}\cdot w_{jj'}\cdot v_{j'}(\rho)+w_{j0}\cdot e_{j}\Big]\right\}-\frac{1}{2}\cdot d_{j}\cdot w_{jj}\cdot\Big|v_{j}(\rho)\Big|^{2}
\end{aligned}$$

$\rho\rightarrow\rho+1$: Updating the j-th neuron $\Rightarrow v_{j'}(\rho+1)=v_{j'}(\rho),\quad\forall j'\neq j$

$$\begin{aligned}
E[\boldsymbol{v}(\rho+1)] = &-\frac{1}{2}\cdot\sum_{\substack{j''=1\\ j''\neq j}}^{N}\sum_{\substack{j'=1\\ j'\neq j}}^{N} d_{j''}\cdot w_{j''j'}\cdot v^{*}_{j''}(\rho)\cdot v_{j'}(\rho) - \Re\left\{\sum_{\substack{j'=1\\ j'\neq j}}^{N} d_{j'}\cdot v^{*}_{j'}(\rho)\cdot w_{j'0}\cdot e_{j'}\right\}\\
&+\Re\left\{\sum_{\substack{j'=1\\ j'\neq j}}^{N} d_{j'}\cdot\int_{0}^{v_{j'}(\rho)}\varphi_{j'}^{-1}(\vartheta)\mathrm{d}\vartheta^{*}\right\}+\Re\left\{d_{j}\cdot\int_{0}^{v_{j}(\rho+1)}\varphi_{j}^{-1}(\vartheta)\mathrm{d}\vartheta^{*}\right\}\\
&-\Re\left\{v^{*}_{j}(\rho+1)\cdot d_{j}\cdot\Big[\sum_{\substack{j'=1\\ j'\neq j}}^{N}\cdot w_{jj'}\cdot v_{j'}(\rho)+w_{j0}\cdot e_{j}\Big]\right\}\\
&-\frac{1}{2}\cdot d_{j}\cdot w_{jj}\cdot\Big|v_{j}(\rho+1)\Big|^{2}
\end{aligned}$$

$$\begin{aligned}
\Delta E_j =& E[\boldsymbol{v}(\rho+1)] - E[\boldsymbol{v}(\rho)] \\
\Delta E_j =& -\Re\left\{ d_j \cdot \left[v_j^*(\rho+1) - v_j^*(\rho) \right] \cdot \Big[\sum_{\substack{j'=1 \\ j'\neq j}}^{N} w_{jj'} \cdot v_{j'}(\rho) + w_{j0} \cdot e_j \Big] \right\} \\
& + \Re\left\{ d_j \cdot \int_{v_j(\rho)}^{v_j(\rho+1)} \varphi_j^{-1}(\vartheta) \mathrm{d}\vartheta^* \right\} - \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left\{ \left| v_j(\rho+1) \right|^2 - \left| v_j(\rho) \right|^2 \right\} \\
\Delta E_j =& -d_j \cdot \Re\left\{ u_j(\rho+1) \cdot \left[v_j^*(\rho+1) - v_j^*(\rho) \right] - \int_{v_j(\rho)}^{v_j(\rho+1)} \varphi_j^{-1}(\vartheta) \mathrm{d}\vartheta^* \right\} \\
& - \frac{1}{2} \cdot d_j \cdot w_{jj} \cdot \left| v_j(\rho+1) - v_j(\rho) \right|^2
\end{aligned}$$

Appendix B

List of Symbols, Functions and Abbreviations

Abbreviations

APP	Apposteriori probability
AWGN	Additive white Gaussian noise
BER, P_b	Bit error rate
BP	Belief propagation
BPSK	Binary phase shift keying
CAM	Content addressable memory
CMF	Channel matched filter
COD_{cc}	Channel coding
CO-BP	Continuous-time belief propagation
CO-ITD	Continuous-time iterative threshold decoding
CSOC	Convolutional self-orthogonal code

CTDS	Continuous-time dynamical system
DEC	Decoding
DECI	Decision device
DET	Detector
DI-BP	Discrete-time belief propagation
DI-ITD	Discrete-time iterative threshold decoding
DS	Dynamical system
DTDS	Discrete-time dynamical system
EQ	Equalization
ESTI	Estimation function
GAS	Global asymptotical stability
HORNN	High order recurrent neural network
It.	Iteration
ITD	Iterative threshold decoding
LAS	Local asymptotical stability
LDPC	Low-density parity-check
LLR	Log-likelihood ratio
MC-CDM	Multicarrier-code-division multiplexing
MIMO	Multiple-input multiple-output
MMSE	Minimum mean square error
ML	Maximum likelihood
$\mathrm{MOD}_{\mathrm{dig}}$	Digital modulation
M-PSK	M-ary phase shift keying
M-QAM	M-ary quadrature amplitude modulation
OFDM	Orthogonal frequency-division multiplexing
QPSK	Quadrature phase shift keying
RNN	Recurrent neural network
SNK	Sink
SNR	Signal-to-noise ratio
SRC	Source
s/s APP	Symbol-by-symbol apposteriori probability
s/s ML	Symbol-by-symbol maximum likelihood
VE-RNN	Vector equalizer (equlization) based on recurrent neural network
VLSI	Very-large-scale integration
WGN	White Gaussian noise
ZF	Zero forcing

Symbols

$a^{(1)}, a^{(2)}$	Needed parameters to approximate the optimum estimation function
$\mathcal{A}$	Neighborhood of a fixed/equilibrium point
$\boldsymbol{A} = \text{diag}\{a_1, a_2, \cdots, a_N\}$	Inner state feedback of the RNN
b_0, b_1, b_2	Complex numbers
$\mathcal{B}$	Value domain of functions of class $g^{(1)}, g^{(2)}, g^{(3)}$
$\mathcal{B}_r$	Value domain of functions of class $g^{(4)}$
$\boldsymbol{B}$	Needed matrix for iterative decoding
C_j	Capacitance of the j-th neuron
$\boldsymbol{c}$	Codeword
$\mathcal{C}$	Code book
$\boldsymbol{c}_{ls}$	External input of a continuous-time linear dynamical system
$\boldsymbol{C}$	Hermitian matrix
$\boldsymbol{C}_{ls}$	Matrix of a continuous-time linear dynamical system
$\boldsymbol{d}_{ls}$	External input of a discrete-time linear dynamical system
$\boldsymbol{D} = \text{diag}\{d_1, d_2, \cdots, d_N\}$	Diagonal positive definite matrix
$\boldsymbol{D}_{ls}$	Matrix of a discrete-time linear dynamical system
E_b	Energy per information bit
$\boldsymbol{e}$	External input of the RNN/HORNN
$\boldsymbol{G}$	Generator matrix
$\boldsymbol{H}$	Channel impulse response matrix
$\boldsymbol{H}_P$	Parity check matrix
I_{nt}	An integral
I_{eq}	An equality
$\boldsymbol{I}$	Identity matrix of suitable size
j, j' ,j''	Flexible neuron indices
J_r, J_i	Power of the residual interference
$\boldsymbol{J}_{\boldsymbol{s}_c}[\boldsymbol{u}], \boldsymbol{J}_{\boldsymbol{s}_d}[\boldsymbol{u}]$	The Jacobian matrices of the functions $\boldsymbol{s}_c$, $\boldsymbol{s}_d$ at $\boldsymbol{u}$
k	Length of the information word
l, l'	Discrete-time variables (parallel update)
l_c	Length of a limit cycle
$\boldsymbol{L}$	Vector of log-likelihood values
$\boldsymbol{L}_{ch}$	Vector of intrinsic L-values
$\boldsymbol{L}_{ext}$	Vector of extrinsic L-values
m	Number of the parity symbols
M	Length of the symbol alphabet

M_i	Length of the distinct imaginary part of the symbol alphabet
M_r	Length of the distinct real part of the symbol alphabet
M_{tx}	Length of the vector of transmit symbols/signals
M_{rx}	Length of the vector of receive signals
n	Length of the codeword
n_h	Number of the nonzero elements in the parity check matrix
N	Dimension of a dynamical system/neural network
N_b	Block length
N_0	Noise power spectral density
$\boldsymbol{n}$	A sample function of a white Gaussian noise vector process
$\tilde{\boldsymbol{n}}$	A Sample function of a colored WGN vector process
$\boldsymbol{P}$	Needed matrix for belief propagation
$\boldsymbol{q}$	vector of uncoded bits of length k (information word)
$\boldsymbol{q}_c$	vector of coded bits of length n (codeword)
$\hat{\boldsymbol{q}}$	vector of decided bits of length k (information word)
$\boldsymbol{Q}$	Matrix for global stability condition
r	Code rate
R_j	Resistance of the j-th neuron
$\boldsymbol{R}$	Discrete-time channel matrix on symbol basis
$\boldsymbol{s}_{tx}$	Vector of transmit signals
$\boldsymbol{s}_{rx}$	Vector of receive signals
$\boldsymbol{S}$	Needed matrix for iterative decoding
t	Continuous-time variable
T_s	Symbol duration
T_e	Evolution time (multiple of τ)
$\boldsymbol{u}$	State vector of a dynamical system/inner state of a neural network
$\boldsymbol{u}_r$, $\boldsymbol{u}_i$	Real/imaginary part
$\boldsymbol{u}_{fp}$, $\boldsymbol{u}_{eq}$	Fixed/equilibrium point
$\boldsymbol{u}_{ini}$	Initial value of a dynamical system/neural network
$\boldsymbol{U}$	Matrix of basic waveforms
$\boldsymbol{v}$	Output of the RNN/HORNN
$\boldsymbol{V}$	Channel matched filter matrix
$\boldsymbol{W}$	Weight matrix
$\boldsymbol{W}_0$	Diagonal weight matrix for external inputs
$\boldsymbol{W}_{ITD,0}$, $\boldsymbol{W}_{ITD,1}$	Needed matrices for iterative threshold decoding
$\boldsymbol{x}$	Vector of transmit symbols
$\tilde{\boldsymbol{x}}$	Vector of ”receive” symbols
$\check{\boldsymbol{x}}$	Vector of soft estimated symbols

$\hat{\boldsymbol{x}}$	Vector of decided/hard estimated symbols
$\boldsymbol{y}$	Output of the matched filter
$\boldsymbol{z}$	Shifted inner state
$\alpha^{(1)}, \alpha^{(2)}$	needed parameters to approximate the optimum estimation function
β_r, β_i	Slopes of the real and imaginary parts of the activation function, respectively
$\boldsymbol{\Gamma}_r = \mathrm{diag}\left\{\gamma_{1,r}, \gamma_{2,r}, \cdots, \gamma_{N,r}\right\}$	Maximum derivation of the real part of the activation function with respect to the real part of the argument
$\boldsymbol{\Gamma}_i = \mathrm{diag}\left\{\gamma_{1,i}, \gamma_{2,i}, \cdots, \gamma_{N,i}\right\}$	Maximum derivation of the imaginary part of the activation function with respect to the imaginary part of the argument
$\delta_{\boldsymbol{J}}$	The determinant of the matrix J
Δt	Sampling step for the first Euler method
$\zeta, \tilde{\zeta}, \breve{\zeta}$	Random variables representing the transmit, receive and estimated symbol, respectively
η	Lipschitz constant
ϑ	Integration variable
$\lambda^{(j)}_{\boldsymbol{C}_{ls}}, \lambda^{(j)}_{\boldsymbol{D}_{ls}}$	The j-th eigenvalue of the matrices $\boldsymbol{C}_{ls}$, $\boldsymbol{D}_{ls}$, respectively
μ	Bound of the function g
$\boldsymbol{\xi}$	Possible vector of transmit symbols
ρ	Discrete-time variable (serial update)
σ^2	Noise power
σ_r^2, σ_i^2	Variance of the real and imaginary part of the AWGN process, respectively
$\boldsymbol{\Upsilon} = \mathrm{diag}\{\tau_1, \tau_2, \cdots, \tau_N\}$	Relaxation time matrix of a continuous-time dynamical system/neural network
τ_j	Relaxation time of the j-th neuron
$\boldsymbol{\Upsilon}_d$	Relaxation time matrix of the continuous-time decoder
$\boldsymbol{\Upsilon}_e$	Relaxation time matrix of the continuous-time equalizer
$\boldsymbol{\Phi}_{\boldsymbol{nn}}$	Correlation matrix of the stochastic vector-valued process $\boldsymbol{n}$
$\boldsymbol{\Psi}$	Symbol alphabet
$\boldsymbol{\Psi}^{(sp)}$	Separable symbol alphabet
$\boldsymbol{\Omega} = \mathrm{diag}\{\eta_1, \eta_2, \ldots, \eta_N\}$	Matrix of Lipschitz constants
$\boldsymbol{0}_N$	A column vector of zeros of length N
$\boldsymbol{1}_N$	A column vector of ones of length N
$\boldsymbol{0}_{N\times N}$	Matrix of zeros of size $N \times N$
$\boldsymbol{1}_{N\times N}$	Matrix of ones of size $N \times N$

Functions

cosh	Hyperbolic cosine function
exp	Exponential function
E	Lyapunov function
E_{xp}	Expectation
F	A scalar function
$\boldsymbol{f}$	Vector valued function
$g^{(1)}, g^{(2)}, g^{(3)}, g^{(4)}$	Function classes
g	A function belong to $g^{(1)}$, $g^{(2)}$, $g^{(3)}$ or $g^{(4)}$
g_r, g_i	Real/imaginary part
g^{-1}	Inverse function
g_r^{-1}, g_i^{-1}	Real/imaginary part
$G_r(v_r), G_i(v_i)$	Strongly convex functions
ln	Natural logarithm
sgn	sign function
sinh	Hyperbolic sine function
$\boldsymbol{s}_d, \boldsymbol{s}_c$	Function rule of a discrete/continuous-time dynamical system
$\boldsymbol{s}_{d,r}, \boldsymbol{s}_{c,r}$	Real part
$\boldsymbol{s}_{d,i}, \boldsymbol{s}_{c,i}$	Imaginary part
tanh	Hyperbolic tangent function
$\theta^{(opt)}$	Optimum estimation function
$\theta_{S/L}$	Function Converting symbols to L-values
$\theta_{L/S}$	Function Converting L-values to symbols
φ	Activation function of a neural network
ϕ	A scalar function

Notation

$\imath$	Imaginary unit $\sqrt{\imath} = -1$
$(\cdot)^T$	Transpose
$(\cdot)^*$	Conjugate
$(\cdot)^H$	Hermitian
$\underset{x}{\mathrm{argmax}} f(x)$	Denotes the value of x that maximizes $f(x)$
$\Re(\cdot)$	Real part of a complex number
$\Im(\cdot)$	Imaginary part of a complex number
$\max\{\cdot\}$	Maximum of a real-valued variable
$\max\{\cdot,\cdot\}$	The maximum of a two real-valued numbers
$\min\{\cdot\}$	Minimum of a real-valued variable
$p(\cdot)$	Probability density function
Prob	Probability
$\mathbb{N}$	Set of natural numbers
$\mathbb{N}_0$	Set of natural numbers including the zero
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
eig	Eigenvalues
$\lvert \cdot \rvert$	Absolute value
Pos	Position
$\delta(t)/\delta(l)$	Continuous/discrete Dirac delta impulse
∇	Gradient of a function

Bibliography

[1] P.-A. Absil and R. Sepulchre. Continuous dynamical systems that realize discrete optimization on the hypercube. *Systems & Control Letters*, 52(3-4):297–304, July 2004.

[2] M. Atencia, G. Joya, and F. Sandoval. Dynamical analysis of continuous higher-order Hopfield networks for combinatorial optimization. *Neural Computation*, 17(8):1802–1819, August 2005.

[3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2):284–287, March 1974.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes (1). In *IEEE International Conference on Communications ICC*, volume 2, pages 1064–1070, Geneva, Switzerland, 23–26 May 1993.

[5] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*, chapter 19. Verlag Harri Deutsch, 1995.

[6] C. Cardinal, D. Haccoun, and F. Gagnon. Iterative threshold decoding without interleaving for convolutional self-doubly orthogonal codes. *IEEE Transactions on Communications*, 51(8):1274–1282, August 2003.

[7] R. De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18, 2000.

[8] A. Dembo, O. Farotimi, and T. Kailath. High-order absolutely stable neural networks. *IEEE Transactions on Circuits and Systems*, 38(1):57–65, January 1991.

[9] J. Egle. *Detection methods for power and bandwidth efficient single carrier block transmission*. Fortschritt-Berichte: Reihe 10, Informatik, Kommunikation. VDI-Verlag, 2004. Dissertation, Ulm University, Institute of Information Technology.

[10] A. Engelhart. *Vector detection techniques with moderate complexity*. Fortschritt-Berichte: Reihe 10, Informatik, Kommunikation. VDI-Verlag, 2003. Dissertation, Ulm University, Institute of Information Technology.

[11] A. Engelhart, W. G. Teich, and J. Lindner. *Complex valued signal estimation for interference cancellation schemes*. Ulm University, Institute of Information Technology, April 1998. ITUU-TR-1998/04.

[12] A. Engelhart, W. G. Teich, J. Lindner, G. Jeney, S. Imre, and L. Pap. A survey of multiuser/multisubchannel detection schemes based on recurrent neural networks. *Wireless Communications and Mobile Computing, Special Issue on Advances in 3G Wireless Networks*, 2(3):269–284, May 2002.

[13] K. Eriksson, D. Estep, and C. Johnson. *Applied mathematics: Body and soul: Volume 1: Derivatives and geometry in IR3*. Applied Mathematics Series. Springer, 2003.

[14] F. Fogelman-Soulié, C. Mejia, E. Goles, and S. Martínez. Energy functions in neural networks with continuous local functions. *Complex Systems*, 3(3):269–293, 1989.

[15] R. G. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, January 1962.

[16] O. Galor. *Discrete dynamical systems*. Springer Verlag Berlin-Heidelberg, 2010.

[17] S. L. Goh and D. P. Mandic. A complex-valued RTRL algorithm for recurrent neural networks. *Neural Computation*, 16(12):2699–2713, December 2004.

[18] E. Goles, A., and S. Martínez. A short proof on the cyclic behaviour of multithreshold symmetric automata. *Information and Control*, 51(2):95–97, November 1981.

[19] J. Hagenauer, E. Offer, C. Méasson, and M. Mörz. Decoding and equalization with analog non-linear networks. *European Transactions on Telecommunications*, 10(6):659–680, November-Decemer 1999.

[20] S. Haykin. *Neural networks: A comprehensive foundation*. Macmillan College Publishing Company, Inc., 866 Third Avenue, New York, New York 10022, 1994.

[21] Y.-C. He, D. Haccoun, and C. Cardinal. Performance comparison of iterative BP and threshold decoding for convolutional self-doubly-orthogonal codes. In *IEEE 65th Vehicular Technology Conference VTC2007-Spring*, pages 2000–2004, 2007.

[22] S. Hemati and A. H. Banihashemi. Comparison between continuous-time asynchronous and discrete-time synchronous iterative decoding. In *IEEE Global Telecommunications Conference*, volume 1, pages 356–360, November/December 2004.

[23] S. Hemati and A. H. Banihashemi. Dynamics and performance analysis of analog iterative decoding for low-density parity-check (LDPC) codes. *IEEE Transactions on Communications*, 54(1):61–70, January 2006.

[24] S. Hemati and A. H. Banihashemi. Convergence speed and throughput of analog decoders. *IEEE Transactions on Communications*, 55(5):833–836, May 2007.

[25] S. Hemati, A. H. Banihashemi, and C. Plett. A 0.18 μm CMOS analog min-sum iterative decoder for a (32,8) low-density paroty-check (LDPC) code. *IEEE Journal on Solid-State Circuits*, 41(11):2531–2540, November 2006.

[26] S. Hemati and A. Yongacoglu. Dynamics of analog decoders for different message passing representation domains. *IEEE Transactions on Communications*, 58(3):721–723, May 2010.

[27] S. Hemati and A. Yongacoglu. On the dynamics of analog min-sum iterative decoders: An analytical approach. *IEEE Transactions on Communications*, 58(8):2225–2231, August 2010.

[28] J. B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Grundlehren Text Editions. Springer Verlag, Berlin, 2001.

[29] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79(8):2554–2558, April 1982.

[30] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the USA*, 81(10):3088–3092, May 1984.

[31] J. J. Hopfield and D. W. Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985.

[32] S. Hu and J. Wang. Global stability of a class of discrete-time recurrent neural networks. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 49(8):1104–1117, August 2002.

[33] S. Jankowski, A. Lozowski, and J. M. Zurada. Complex-valued multistate neural associative memory. *IEEE Transactions on Neural Networks*, 7(6):1491–1496, November 1996.

[34] S. J. Johnson. *Iterative error correction: Turbo, low-density parity-check and repeat-accumulate codes*. Cambridge University Press, January 2010.

[35] S. J. Johnson and S. R. Weller. Low-density parity-check codes: Design and decoding. In *Wiley Encyclopedia of Telecommunications*, pages 1–18. John Wiley & Sons, Inc., January 2003.

[36] G. I. Kechriotis and E. S. Manolakos. Hopfield neural network implementation of the optimal CDMA multiuser detector. *IEEE Transactions on Neural Networks*, 7(1):131–141, January 1996.

[37] G. I. Kechriotis and E. S. Manolakos. A hybrid digital signal processing-neural network CDMA multiuser detection scheme. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 43(2):96–104, February 1996.

[38] L. Kocarev, F. Lehmann, G. M. Maggio, B. Scanavino, Z. Tasev, and A. Vardy. Nonlinear dynamics of iterative decoding systems: Analysis and applications. *IEEE Transactions on Information Theory*, 52(4):1366–1384, 2006.

[39] E. B. Kosmatopoulos and M. A. Christodoulou. Structural properties of gradient recurrent high-order neural networks. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 42(9):592–603, September 1995.

[40] E. Kreyszig and E. J. Norminton. *Advanced engineering mathematics*. John Wiley & Sons Inc., New York, 7 edition, 1993.

[41] Y. Kuroe, N. Hashimoto, and T. Mori. On energy function for complex-valued neural networks and its applications. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02*, volume 3, pages 1079–1083, 18–22 November 2002.

[42] D. L. Lee. Relaxation of the stability condition of the complex-valued neural networks. *IEEE Transactions on Neural Networks*, 12(5):1260–1262, September 2001.

[43] D.-L Lee. Complex-valued neural associative memories: Network stability and learning algorithms. In A. Hirose, editor, *Complex-valued neural networks: Theories and applications*, volume 5 of *Series on Innovative Intelligence Series*, chapter 3, pages 29–55. World Scientific Publishing Co. Pte. Ltd., 2003.

[44] D. L. Lee and W. J. Wang. A multivalued bidirectional associative memory operating on a complex domain. *Neural Networks*, 11(9):1623–1635, 1998.

[45] J. Lindner. MC-CDMA in the context of general multiuser/ multisubchannel transmission methods. *European Transactions on Telecommunications*, 10(4):351–367, July-August 1999.

[46] J. Lindner. *Informationsübertragung: Grundlagen der Kommunikationstechnik*. Springer-Verlag Berlin Heidelberg, 2005.

[47] D. Liu and A. N. Michel. *Dynamical systems with saturation nonlinearities: Analysis and design*, volume 195 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1994.

[48] Y. Liu, Z. You, and L. Cao. Dynamical behaviors of Hopfield neural network with multilevel activation functions. *Chaos, Solitons & Fractals*, 25(5):1141–1153, September 2005.

[49] H.-A. Loeliger. Decoding in analog VLSI. *IEEE Communications Magazine*, 37(4):99–101, April 1999.

[50] J. D. Logan. *A first course in differential equations*. Springer, New York, Heidelberg, 2006.

[51] D. J. C. MacKay. Encyclopedia of sparse graph codes. `http://www.inference.phy.cam.ac.uk/mackay/codes/data.html#l1`.

[52] C. M. Marcus and R. M. Westervelt. Dynamics of iterated-map neural networks. *Physical Review A*, 40:501–504, July 1989.

[53] J. L. Massey. *Threshold decoding*. M.I.T. Press, Cambridge, Massachusetts, 1963.

[54] C. Mead. *Analog VLSI and neural systems*. Addison-Wesley, 1989.

[55] T. Miyajima, T. Hasegawa, and M. Haneishi. On the multiuser detection using a neural network in code-division multiple-access communications. *IEICE Transactions on Communication, Special Issue on Spread Spectrum Techniques and Applications*, E76-B(8):961–968, August 1993.

[56] T. Miyajima and K. Yamanaka. Phasor model with application to multiuser communication. In A. Hirose, editor, *Complex-valued neural networks: Theories and applications*, volume 5 of *Series on Innovative Intelligence Series*, chapter 12, pages 251–275. World Scientific Publishing Co. Pte. Ltd., 2003.

[57] M. Mostafa, W. G. Teich, and J. Lindner. Analog realization of iterative threshold decoding based on high-order recurrent neural networks. In *4th International Conference on Signal Processing and Communication Systems ICSPCS*, pages 1–6, Gold Coast, Australia, 13-15 December 2010.

[58] M. Mostafa, W. G. Teich, and J. Lindner. A modified discrete recurrent neural network as vector detector. In *IEEE Asia Pacific Conference on Circuits and Systems APCCAS*, pages 620–623, Kuala Lumpur, Malaysia, 6-9 December 2010.

[59] M. Mostafa, W. G. Teich, and J. Lindner. Global vs. local stability of recurrent neural networks as vector equalizer. In *5th International Conference on Signal Processing and Communication Systems ICSPCS*, pages 1–5, Hawaii, USA, 12-14 December 2011.

[60] M. Mostafa, W. G. Teich, and J. Lindner. Stability analysis of recurrent neural networks with time-varying activation functions. In *3rd International Workshop on Nonlinear Dynamics INDS and 16th International Symposiums on Theoretical Electrical Engineering ISTET*, pages 239–244, Klagenfurt, Austria, 25-27 July 2011.

[61] M. Mostafa, W. G. Teich, and J. Lindner. Analysis of high order recurrent neural networks for analog decoding. In *7th International Symposium on Turbo Codes and Iterative Information Processing ISTC*, pages 116–120, Gothenburg, Sweden, 27-31 August 2012.

[62] M. Mostafa, W. G. Teich, and J. Lindner. Vector equalization based on continuous-time recurrent neural networks. In *6th International Conference on Signal Processing and Communication Systems ICSPCS*, Gold Coast, Australia, 12-14 December 2012.

[63] M. Mostafa, W. G. Teich, and J. Lindner. Comparison of belief propagation and iterative threshold decoding based on dynamical systems. In *IEEE International Symposium on Information Theory ISIT*, Istanbul, Turkey, 07-12 July 2013.

[64] M. Mostafa, W. G. Teich, and J. Lindner. Stability analysis of vector equalization based on recurrent neural networks. In K. Kyamakya, W. A. Halang, W. Mathis, J. C. Chedjou, and Z. Li, editors, *Selected topics in nonlinear dynamics and theoretical electrical engineering*, volume 459 of *Studies in Computational Intelligence*, chapter 18, pages 329–349. Springer Berlin Heidelberg, 2013.

[65] M. Mostafa, W. G. Teich, and J. Lindner. Approximation of activation functions for vector equalization based on recurrent neural networks. In *8th International Symposium on Turbo Codes and Iterative Information Processing ISTC*, pages 52–56, Bremen, Germany, 18-22 August 2014.

[66] M. Mostafa, W. G. Teich, and J. Lindner. Local stability analysis of discrete-time, continuous-state, complex-valued recurrent neural networks with inner state feedback. *IEEE Transactions on Neural Networks and Learning Systems TNNLS*, 25(4):830–836, April 2014.

[67] J. G. Proakis. *Digital communications*. McGraw-Hill, Inc., 3 edition, 1995.

[68] D. Roy and J. V. Rao. *Elements of structural dynamics: A new perspective*, chapter 7. John Wiley & Sons Ltd, 2012.

[69] B. S. Rüffer, C. M. Kellett, P. M. Dower, and S. R. Weller. Belief propagation as a dynamical system: The linear case and open problems. *Control Theory & Applications, IET*, 4(7):1188–1200, July 2010.

[70] P. Sahoo and T. Riedel. *Mean value theorems and functional equations*. World Scientific Publishing Co. Inc., Singapore, 1998.

[71] E. R. Scheinerman. *Invitation to dynamical systems*. Prentice Hall, 1996.

[72] D. Schröder. *Intelligente Verfahren: Identifikation Und Regelung Nichtlinearer Systeme*. Springer, 2010.

[73] C. Sgraja, A. Engelhart, W. G. Teich, and J. Lindner. Combined equalization and decoding for general BFDM packet transmission schemes. In *1st International OFDM Workshop*, Hamburg, Germany, 21-22 September 1999.

[74] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[75] W. Shen, J. Gu, and Y. Shen. Stability analysis for high-order dynamic neural networks with time delays. In *IEEE International Conference on Robotics and Biomimetics. ROBIO 2004.*, pages 966–971, Shenyang, China, 22-26 August 2004. IEEE.

[76] M. Spivak. *Calculus on manifolds: A modern approach to classical theorems of advanced calculus*, volume 1 of *Mathematics Monograph Series*. W.A. Benjamin, Inc., 1965.

[77] G. L. Stüber. *Principles of Mobile Communication*. Springer New York, 3 edition, 2011.

[78] D. Tank and J. J. Hopfield. Simple ”neural” optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, 33(5):533–541, May 1986.

[79] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, IT-27(5):533–547, September 1981.

[80] W. G. Teich, J. Egle, M. Reinhardt, and J. Lindner. Detection method for MC-CDMA based on a recurrent neural network structure. In K. Fazel and G. P. Fettweis, editors, *Multi-Carrier spread-spectrum*, pages 135–142. Springer US, 1997.

[81] W. G. Teich, A. Engelhart, W. Schlecker, R. Gessler, and H. J. Pfleiderer. Towards an efficient hardware implementation of recurrent neural network based multiuser detection. In *IEEE 6th International Symposium on Spread Spectrum Techniques and Applications ISSSTA*, volume 2, pages 662–665, 2-4 September 2000.

[82] W. G. Teich and M. Seidl. Code division multiple access communications: multiuser detection based on a recurrent neural network structure. In *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications ISSSTA*, volume 3, pages 979–984, 22-25 September 1996.

[83] W. G. Teich and P. Wallner. Soft iterative cancellation with successive over relaxation for digital transmission schemes based on multiple sets of orthogonal spreading codes. In *6th International Conference on Signal Processing and Communication Systems ICSPCS*, Gold Coast, Australia, 12-14 December 2012.

[84] S. Verdu. *Multiuser Detection*. Cambridge University Press, New York, NY, USA, 1st edition, 1998.

[85] M. Vidyasagar. Minimum-seeking properties of analog neural networks with multilinear objective functions. *IEEE Transactions on Automatic Control*, 40(8):1359–1375, August 1995.

[86] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, 1967.

[87] E. A. Vittoz. Analog VLSI signal processing: Why, where, and how? *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 8(1):27–44, February 1994.

[88] B. Wang, Z. He, and J. Nie. To implement the CDMA multiuser detector by using transiently chaotic neural network. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3):1068–1071, July 1997.

[89] X. Xu and W. T. Tsai. Effective neural algorithms for the traveling salesman problem. *Neural Networks*, 4(2):193–205, 1991.

[90] X. Xu, W. T. Tsai, and N. K. Huang. A generalized neural network model. Technical Report TR 88–30, Computer Science Department, Institute of Technology, University of Minnesota, 136 Lind Hall, Minneapolis, Minnesota 55455, March 1988.

[91] M. Yoshida and T. Mori. Global stability analysis for complex-valued recurrent neural networks and its application to convex optimization problems. In T. Nitta, editor, *Complex-valued neural networks: Utilizing high-dimensional parameters*, chapter 5, pages 104–114. Information Science Reference, 2009.

[92] W. Zhao, W. Lin, R. Liu, and J. Ruan. Asymptotical stability in discrete-time neural networks. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 49(10):1516–1520, October 2002.

[93] W. Zhong and S.-X. Cheng. Multiuser detection using time-varying scaling-parameter transiently chaotic neural networks. *Electronic Letters*, 35(12):987–989, June 1999.

[94] W. Zhou and J. M. Zurada. Discrete-time recurrent neural networks with complex-valued linear threshold neurons. *IEEE Transactions on Circuits and Systems–II: Express Briefs*, 56(8):669–673, 2009.

[95] J. M. Zurada, I. Cloete, and E. van der Poel. Generalized Hopfield networks for associative memories with multi-valued stable states. *Neurocomputing*, 13(2-4):135–149, 1996.

List of Publications

M. Mostafa, W. G. Teich, and J. Lindner. Local stability analysis of discrete-time, continuous-state, complex-valued recurrent neural networks with inner state feedback. IEEE Transactions on Neural Networks and Learning Systems TNNLS, 25(4):830–836, April 2014.

M. Mostafa, W. G. Teich, and J. Lindner. Stability analysis of vector equalization based on recurrent neural networks. In K. Kyamakya, W. A. Halang, W. Mathis, J. C. Chedjou, and Z. Li, editors, Selected topics in nonlinear dynamics and theoretical electrical engineering, volume 459 of Studies in Computational Intelligence, chapter 18, pages 329–349. Springer Berlin Heidelberg, 2013.

M. Mostafa, W. G. Teich, and J. Lindner. Comparison of belief propagation and iterative threshold decoding based on dynamical systems. In IEEE International Symposium on Information Theory ISIT, Istanbul, Turkey, 07-12 July 2013.

M. Mostafa, W. G. Teich, and J. Lindner. Analysis of high order recurrent neural networks for analog decoding. In 7th International Symposium on Turbo Codes and Iterative Information Processing ISTC, pages 116–120, Gothenburg, Sweden, 27-31 August 2012.

M. Mostafa, W. G. Teich, and J. Lindner. Vector equalization based on continuous-time recurrent neural networks. In 6th International Conference on Signal Processing and Communication Systems ICSPCS, Gold Coast, Australia, 12-14 December 2012.

M. Mostafa, W. G. Teich, and J. Lindner. Global vs. local stability of recurrent neural networks as vector equalizer. In 5th International Conference on Signal Processing and Communication Systems ICSPCS, pages 1–5, Hawaii, USA, 12-14 December 2011.

M. Mostafa, W. G. Teich, and J. Lindner. Stability analysis of recurrent neural networks with time-varying activation functions. In 3rd InternationalWorkshop on Nonlinear Dynamics INDS and 16th International Symposiums on Theoretical Electrical Engineering ISTET, pages 239–244, Klagenfurt, Austria, 25-27 July 2011.

M. Mostafa, W. G. Teich, and J. Lindner. Analog realization of iterative threshold decoding based on high-order recurrent neural networks. In 4th International Conference on Signal Processing and Communication Systems ICSPCS, pages 1–6, Gold Coast, Australia, 13-15 December 2010.

M. Mostafa, W. G. Teich, and J. Lindner. A modified discrete recurrent neural network as vector detector. In IEEE Asia Pacific Conference on Circuits and Systems APCCAS, pages 620–623, Kuala Lumpur, Malaysia, 6-9 December 2010.

For the reasons of data privacy,the curriculum vitae has been removed.